

SocratiQ: A Generative AI-Powered Learning Companion for Personalized Education and Broader Accessibility

Jason Jabbour[†] Kai Kleinbard[†] Olivia Miller Robert Haussman Vijay Janapa Reddi

Harvard University
Boston, Massachusetts, United States
{jasonjabbour,kak2594,omiller01,rhaussman,vj}@g.harvard.edu

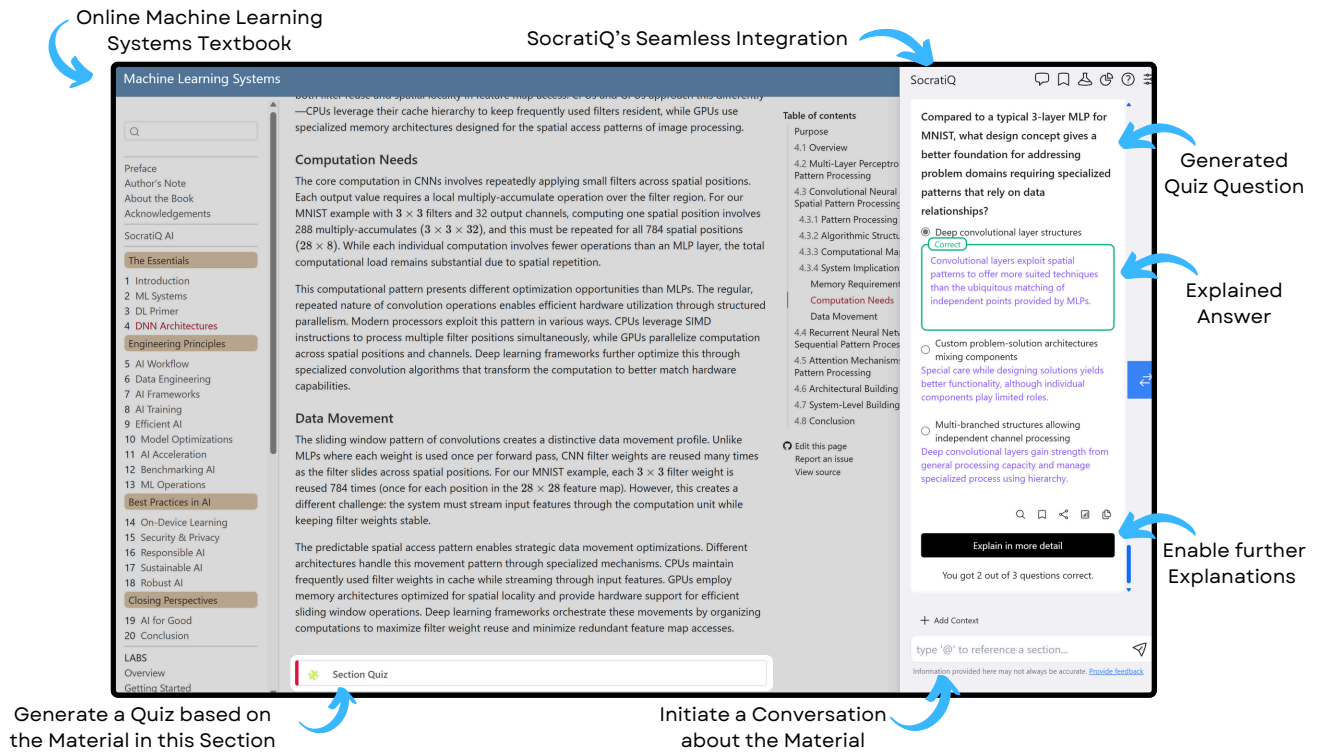


Figure 1: Overview of SocratiQ’s integration into the online machine learning textbook, showcasing how students can generate quiz questions and engage in natural language conversations for further explanations. You can try it at <https://mlsysbook.ai>.

Abstract

Traditional educational approaches often struggle to provide personalized and interactive learning experiences on a scale. In this paper, we present SocratiQ, an AI-powered educational assistant that addresses this challenge by implementing the Socratic method through adaptive learning technologies. The system employs a novel Generative AI-based learning framework that dynamically creates personalized learning pathways based on student responses and comprehension patterns. We provide an account of our integration methodology, system architecture, and evaluation framework, along with the technical and pedagogical challenges encountered

during implementation and our solutions. Although our implementation focuses on machine learning systems education, the integration approaches we present can inform similar efforts across STEM fields. Through this work, our goal is to advance the understanding of how generative AI technologies can be designed and systematically incorporated into educational resources.

1 Introduction

Education has long grappled with the challenge of meeting the unique needs of individual learners. Traditional methods often rely on standardized approaches that struggle to accommodate differences in prior knowledge, learning pace, and cognitive strengths, leaving many students without the personalized support they need

[†]Jason Jabbour and Kai Kleinbard contributed equally to this research.

to succeed. Evidence suggests that teaching styles significantly influence both student achievement [1], [2] and attitudes toward subjects [3], with teaching behavior and instructional strategies playing a critical role in shaping learning outcomes [4]. Although educators may aspire to tailor instruction to every student, the scale of such efforts makes it impractical in conventional classroom settings.

Recent advances in artificial intelligence (AI) have introduced groundbreaking possibilities in education. By leveraging the increasing sophistication of AI-driven tools, we can reimagine teaching and learning to deliver personalized and interactive educational experiences across a range of fields. Large language models (LLMs) have rapidly advanced in their ability to reason through complex problems and articulate solutions [5]. They now excel at tackling challenging subjects like mathematics, physics, chemistry, biology, and law with graduate-level rigor [6]–[10]. Through fine-tuning, these models have been adapted to emulate human-like communication, becoming proficient conversationalists capable of natural dialogue and effective question answering [11]–[13]. As a result, LLMs have become popular tools for learners, often serving as chatbots where students can seek clarification or explore topics.

While these tools hold significant potential to enhance education, the challenge lies in moving beyond their current ad hoc use and integrating them meaningfully into structured learning environments. Achieving this integration requires addressing gaps in delivering personalized, interactive, and course-specific educational experiences. To address this challenge, this paper introduces *SocratiQ*, an AI-powered learning companion that evolves with the learner’s progress. At its core is the principle of “Generative Learning”, a cognitive science framework where learners actively connect new information to prior knowledge by integrating new experiences with existing knowledge structures [14]. This framework emphasizes engaging cognitively through modes such as interactive dialogue (e.g., asking and answering questions). This approach contrasts with passive learning, which treats students as mere recipients of information. *SocratiQ* operationalizes Generative Learning through personalized explanations, adaptive assessments, and thought-provoking interactive conversations, fostering deeper comprehension, application of concepts, and higher-order thinking.

We have integrated *SocratiQ* into our online *Machine Learning Systems* textbook, developed at Harvard University for the CS249R course and available at [MLSysBook.ai](https://mlsysbook.ai). As a course focused on machine learning systems, this textbook is an ideal testbed for *SocratiQ* because it requires students to understand how different technical domains interact in ML applications, ranging from the fundamentals of neural networks to understanding their design and implementation in computer system design. Students engaging with this college-level textbook must draw on these diverse areas of prior knowledge to understand how ML systems work in practice, making standardization less effective for meaningful learning.

SocratiQ allows students to engage with interactive quizzes, receive real-time feedback, request in-depth explanations of advanced concepts, and tailor the conversation to their own level of understanding. Fully integrated into the reading experience of [MLSysBook.ai](https://mlsysbook.ai) as shown in Figure 1, it creates an engaging learning environment for a textbook. The primary objectives of this integration are threefold: first, to enhance student engagement by

transforming passive reading into an active, participatory experience; second, to provide personalized learning pathways that adapt to individual student needs and learning styles; and third, to provide broader accessibility to high-quality machine learning education.

We anticipate that the integration of *SocratiQ* into [MLSysBook.ai](https://mlsysbook.ai) has the potential to transform how students interact with and learn from textbooks, particularly in complex, multidisciplinary fields. Having implemented this prototype in our CS249R course, we share our experiences and insights with the broader community. We address several key questions that are likely to interest educators, researchers, and policymakers working in AI-enhanced education.

- **For educators:** *What are the practical steps to implement an AI learning assistant in an online textbook or course resource?* We share our hands-on experience integrating *SocratiQ* into the CS249R course textbook, focusing on the practical challenges encountered and the solutions developed.
- **For researchers:** *How does the integration of AI technology in the classroom affect learning outcomes and what insights can be gained to improve student assessment systems?* We present a set of characteristics that focus on improving comprehension, retention, and critical thinking skills. We explore strategies for adaptive assessments to better guide students in identifying and addressing gaps in their knowledge.
- **For policymakers:** *How can AI be integrated into classroom settings while ensuring that students remain active participants in the learning process?* We highlight that AI tools, such as *SocratiQ*, complement rather than replace critical learning activities. We demonstrate how AI can guide students to actively engage with the material, ensuring mastery of concepts rather than passive reliance on AI-generated answers.

The remainder of this paper is organized as follows. We first review related work on AI-enabled educational tools, examining both their capabilities and limitations. We then compare traditional and AI-enhanced learning approaches, analyzing their respective strengths and areas for improvement. Next, we present *SocratiQ*’s key features and system architecture, followed by detailed operational strategies for implementing such a system in practice. We conclude with an evaluation of *SocratiQ*’s effectiveness in generating educational content and fostering student engagement, providing insights for future development of AI learning companions.

2 The Evolving Educational Landscape

Artificial intelligence has transformed education by reinventing traditional teaching methods. AI-driven tools improve personalization, streamline administrative tasks, and promote engagement, enriching the educational experience. However, they also pose unique challenges, such as overreliance. In this section, we trace the evolution of AI tools from early adaptive systems to modern language models, examining their strengths, limitations, and transformative impact on education, which sets the stage for *SocratiQ*.

2.1 The Rise of AI-Enabled Educational Tools

The evolution of AI in education has progressed through several distinct phases, as summarized in Table 1. Early advances in AI laid

Table 1: Evolution of AI in education, from adaptive learning systems to large language models.

Wave	Period	Representative Technologies	Key Capabilities	Impact on Education
First Wave(Adaptive Learning)	1990s-2010s	ITS (MATHia, ALEKS), Early adaptive platforms (DreamBox)	Rule-based personalization, Performance tracking, Automated assessments	Pioneered adaptive learning, Demonstrated value of personalization, Limited to structured domains.
Second Wave (Specialized AI)	2010s-2020	Language platforms (Duolingo, Rosetta Stone), STEM tools (Wolfram Alpha, GeoGebra), Writing assistants (Grammarly, Turnitin)	Task-specific AI, Interactive features, Automated feedback	Enhanced domain expertise, Improved engagement, Created specialized support systems.
Third Wave (Generative AI)	2020-Present	LLMs (ChatGPT, Gemini), AI tutors (Khanmigo), Enhanced platforms (Duolingo Max)	Natural conversation, Context understanding, Dynamic adaptation	Enabled open-ended learning, Broadened accessibility, Changed student-AI interaction.

the foundation for personalized and adaptive learning tools. Intelligent Tutoring Systems (ITS), such as Carnegie Learning’s MATHia [15], ALEKS [16], and DreamBox Learning [17], provided tailored instruction using techniques such as rule-based reasoning, data mining, and Bayesian networks. They offered personalized pathways for students by adapting lessons to individual progress and needs, particularly in mathematics and science education [18], [19]. Language platforms powered by AI such as Duolingo [20], Rosetta Stone [21], and Babbel [22] used Automatic Speech Recognition (ASR) to improve pronunciation and give feedback [23].

The second wave brought specialized AI tools that targeted specific educational needs. AI-powered platforms such as Wolfram Alpha [24], GeoGebra [25], and Labster [26] supported STEM education with interactive features, self-directed learning, and collaborative opportunities [27]. Furthermore, AI writing assistants such as Grammarly [28], Turnitin [29] and RefME [30] improved academic writing by providing grammar correction, plagiarism detection, and citation assistance [31], [32]. However, these tools were not conversationalists; they were designed to perform specific, predefined tasks using models trained for their respective functionalities.

The third wave, driven by advances in natural language processing (NLP), has led to the development of more versatile LLMs such as OpenAI’s ChatGPT [33] and Google’s Gemini [34], which show strong capabilities in question answering and maintaining conversational interactions. These generative AI platforms are transforming education by offering instant answers, personalized explanations, and interactive dialogues that enhance learning and student engagement [35], [36]. The integration of generative AI technologies such as GPT-4 has significantly expanded the capabilities of educational tools. For example, Khan Academy’s Khanmigo [37] evolved from a platform for instructional videos and exercises into a virtual tutor powered by generative AI, offering contextual guidance and adaptive support to enhance student understanding and autonomy [38]. Similarly, Duolingo Max [39], leveraging GPT-4, now provides interactive role-play scenarios and contextual practice, creating realistic and immersive language learning experiences [40].

Looking at this progression (Table 1), we can see how educational AI has evolved from narrow domain-specific applications to more flexible, interactive learning companions. Although early systems excelled at structured tasks like problem solving in mathematics,

and specialized tools brought focused expertise to specific subjects, modern AI systems are beginning to bridge multiple domains and adapt to diverse learning contexts. This evolution reflects broader changes in educational needs, where the ability to integrate knowledge across disciplines and participate in open-ended learning has become increasingly important.

However, while these AI systems offer powerful capabilities, their effective integration into existing educational frameworks remains a significant challenge, requiring careful consideration of how to preserve and enhance traditional teaching approaches.

2.2 Balancing AI & Human-Centered Education

The evolution of education has seen significant advances in both traditional and AI-assisted learning models, each contributing distinct benefits to the development of well-rounded students. Traditional educational models offer irreplaceable advantages in the cultivation of interpersonal relationships and a vibrant sense of community among students. Collaborative activities, such as group projects and discussions, facilitate learning from diverse perspectives, thus enhancing critical thinking and problem solving skills.

In addition, experiential learning opportunities, including research projects and laboratory activities, establish crucial connections between classroom instruction and real-world applications, thus improving knowledge retention and comprehension. In particular, the presence of human instructors provides essential mentorship, individualized guidance, and support for the development of vital soft skills such as communication and presentation abilities - skills that remain critical for professional development.

Concurrently, AI’s capabilities have revamped educational possibilities through unprecedented personalization and scalability. The capacity of AI to deliver customized learning experiences represents one of its most significant contributions to education: generating customized materials, creating adaptive quizzes, and providing writing assistance that improves student engagement and study time [41], [42]. These tools have demonstrated particular efficacy in STEM fields, where computer-assisted instruction significantly improves performance compared to traditional methods [43].

The ability of AI to provide immediate feedback enables students to identify strengths and areas for improvement in real time, while its automation of routine tasks such as grade allows instructors

to focus on meaningful student interactions [42]. Importantly, AI contributes to the democratization of education by offering high-quality learning resources to students who might otherwise lack access to traditional educational opportunities [44].

Despite the complementary potential of traditional and AI-enhanced approaches, their integration presents challenges that require careful consideration. Traditional learning models are constrained by temporal, spatial, and resource limitations, which impede the scalability of personalized attention, particularly in large educational settings. In contrast, AI-assisted models introduce their own complexities, ranging from privacy and security concerns arising from extensive data collection to implementation barriers such as limited understanding of data-driven systems and questions of data sovereignty [45]. A key challenge lies in mitigating the potential erosion of human connections; while AI excels in content delivery, it cannot replicate the empathetic and nuanced support that human educators provide for students' emotional and psychological needs [46]. Furthermore, there exists a risk of overreliance on AI tools, which could potentially impede the development of critical thinking skills if students default to AI-generated answers rather than engaging deeply with the material [47]. Consequently, we must navigate the delicate balance between leveraging AI's capabilities and preserving the essential human elements of learning.

The creation of an effective educational environment requires the thoughtful integration of traditional and AI-assisted approaches while preserving their respective strengths. In traditional education, various roles collectively improve the learning experience, from peer-to-peer support among classmates to teaching assistants that bridge gaps between students and instructors, to course coordinators who supervise curriculum quality (Table 2). AI systems should be designed to complement rather than replace these roles, addressing specific gaps such as providing personalized support in large courses or augmenting peer learning where additional assistance is required. The solution lies in leveraging AI's capabilities for personalization and efficiency while maintaining the human connections and community support essential for meaningful learning experiences. This balanced approach can foster both immediate engagement and long-term development: while AI provides immediate feedback and adaptive pathways tailored to student skill levels, traditional methods build self-efficacy through collaborative projects and substantive human interactions.

The rapid evolution of AI technology emphasizes the need for systematic frameworks to guide this integration. Such frameworks must carefully balance AI's capabilities with traditional pedagogical strengths while addressing the challenges of implementation. This objective motivated the development of our SocratiQ AI learning companion system, which demonstrates how AI can enhance rather than replace the human elements central to effective education.

3 SocratiQ System Design

In this section, we introduce SocratiQ as a learning companion designed for an online textbook on machine learning systems, focusing on how it addresses the specific challenges of teaching such a complex subject. SocratiQ addresses the unique challenges inherent in teaching complex and multidisciplinary subjects by integrating advanced AI capabilities with traditional pedagogical approaches.

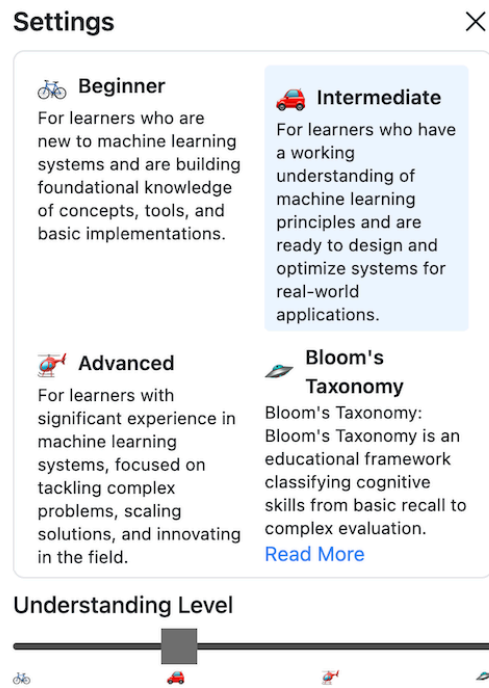


Figure 2: Students can dynamically adjust the academic level to match their learning preferences.

The design of SocratiQ is based on the framework presented in Table 2, with a primary focus on fulfilling the role of the learning companion. This approach allows us to leverage AI's strengths in personalization and scalability while preserving the irreplaceable aspects of human-centered education. We identified four key features critical to an effective AI learning companion: personalized explanations, adaptive assessments, bounded learning, and gamification. These features are carefully designed to address the specific challenges of teaching machine learning systems, a field that requires students to synthesize knowledge from diverse areas such as algorithms, linear algebra, and computer architecture.

In the following subsections, we explore each of these features in depth. We discuss the rationale behind their inclusion (based on our experience in teaching CS249r) and detail their structure and implementation. This overview hopes to provide educators with a practical framework for incorporating similar AI-enhanced elements into their own courses, regardless of the subject matter.

3.1 Personalized Explanations

Students enter courses with varying levels of prior knowledge, a challenge that is particularly evident in advanced college-level curricula that integrate concepts from multiple disciplines. This variation is especially pronounced in certain types of classes, which require thorough understanding across several foundational areas.

The CS249r Machine Learning Systems class at Harvard University exemplifies this challenge, illustrating the complex interplay of knowledge required in this field. The course demands not only theoretical knowledge, but also practical application skills. Students are required to design and implement efficient and scalable machine

Table 2: A framework for understanding the AI roles that are possible in a classroom setting, along with the various design considerations, ranging from individual learning to institutional implementation, that need consideration or are required. Green checkmarks indicate features that are incorporated in the current version of SocratiQ.

Dimensions ↓	AI Roles →		
	Learning Companion	Teaching Assistant (AI-TA)	Assessment Coordinator
<i>Foundation & Purpose</i>			
Target Users	✓ Individual learners	✓ Students and instructors	✓ Administrators and faculty
Primary Objective	✓ Individual mastery of concepts	✗ Facilitate group learning	✓ Course-wide learning assessment
Key Use Cases	✓ Self-paced learning, concept review	✗ Office hours, group discussions	✓ Curriculum planning, performance tracking
<i>Capabilities & Features</i>			
Generative AI Features	✓ Personalized explanations, adaptive hints	✗ Answer synthesis, discussion prompts	✓ Assessment generation, feedback analysis
Interactive Capabilities	✓ Real-time dialogue, concept mapping	✗ Group facilitation, query handling	✓ Performance visualization, reporting
Personalization Level	✓ High (individual learning paths)	✗ Medium (group-aware responses)	✗ Low (standardized assessment)
<i>Technical Architecture</i>			
Infrastructure Needs	✓ Edge computing, local processing	✗ On-premises database	✗ Data warehouses, analytics engines
Data Architecture	✓ Local-first, encrypted storage	✗ Distributed, session-based	✗ Centralized, hierarchical
Integration Points	✓ E-readers	✗ LMS plugins	✗ Grade books, admin dashboards
<i>Implementation & Deployment</i>			
Privacy Requirements	✓ Individual data protection	✗ Group interaction privacy	✗ Institutional data governance
Deployment Model	✓ Client-side focused	✓ Hybrid client-server	✗ Server-side focused
Key Challenges	✓ Context retention, personalization	✗ Group dynamics, response timing	✓ Assessment fairness, data quality
<i>Evaluation Framework</i>			
Learning Metrics	✓ Knowledge retention, engagement	✓ Participation, query resolution	✓ Course completion, outcomes
System Metrics	✓ Response time, adaptation accuracy	✓ Concurrent users, availability	✓ Processing speed, data accuracy
Success Indicators	✓ Student progress, satisfaction	✓ Reduced instructor load, accessibility	✓ Curriculum effectiveness, ROI

learning systems that can handle large datasets and complex models. This requires a deep understanding of the principles of system design and the ability to optimize algorithms for various hardware configurations. For instance, they must grasp the underlying algorithms of machine learning, such as backpropagation and stochastic gradient descent, while simultaneously comprehending linear algebra principles, including matrix multiplication used in neural networks. Furthermore, they need to possess familiarity with computer architecture, understanding how numerical representations are handled at a low level and how different hardware architectures impact machine learning model quality and performance.

The breadth of these prerequisites results in a student body with widely varying degrees of expertise across these areas. Some students may excel in theoretical machine learning concepts but struggle with systems-level implementation. Others might have a strong background in computer architecture but find the intricacies of advanced machine learning algorithms challenging. This disparity in background knowledge often leads to significant gaps that can impede a student’s ability to fully engage with the material.

The challenge in CS249r lies not just in teaching individual concepts, but in helping students bridge these diverse areas of knowledge, creating a holistic understanding of machine learning systems that spans from theoretical foundations to practical, efficient implementations. This multifaceted nature of the course highlights the complexity of teaching advanced, interdisciplinary subjects and the need for adaptive, comprehensive educational approaches.

In such scenarios, a one-size-fits-all approach to explanation—delivering content in a single, standardized manner—is often ineffective. Effective comprehension requires adapting explanations to align with a student’s background knowledge and addressing specific areas of difficulty. Adjusting the complexity and depth of explanations enables learners to integrate new information with their existing understanding, improving retention and engagement.

To address this, SocratiQ provides four difficulty levels that allow the AI learning companion to tailor its explanations to the student’s current level of understanding. This personalization is achieved through a user interface that enables learners to adjust their knowledge level and style using a slider menu as displayed in Figure 2. The difficulty levels are defined as follows:

- (1) **Beginner:** Focus on foundational concepts, definitions, and straightforward applications in machine learning systems, suitable for learners with little to no prior knowledge.
- (2) **Intermediate:** Emphasize problem-solving, system design, and practical implementations, targeting learners with a basic understanding of machine learning principles.
- (3) **Advanced:** Challenge learners to analyze, innovate, and optimize complex machine learning systems, requiring deep expertise and a holistic grasp of advanced techniques.
- (4) **Expert (Bloom’s Taxonomy):** Create responses that progress through Bloom’s levels: remember, understand, apply, analyze, evaluate, and create. Guide my learning.

These difficulty levels are implemented as system prompts provided to the language model, enabling users to adjust the platform

to align with their learning needs. The corresponding prompts for each difficulty level, simplified here for brevity, are as follows:

Beginner Prompt:

“You are conversing with a Beginner learner: Focus on foundational concepts, definitions, and straightforward applications in machine learning systems, suitable for learners with little to no prior knowledge.”

Intermediate Prompt:

“You are conversing with an Intermediate learner: Emphasize problem-solving, system design, and practical implementations, targeting learners with a basic understanding of machine learning principles.”

Advanced Prompt:

“You are conversing with an Advanced learner: Challenge learners to analyze, innovate, and optimize complex machine learning systems, requiring deep expertise and a holistic grasp of advanced techniques.”

Expert Prompt:

“You are an expert ML teacher using Bloom’s Taxonomy: Create responses that progress through Bloom’s levels: remember, understand, apply, analyze, evaluate, and create. Guide my learning.”

While these prompts are tailored to the specific challenges of teaching machine learning systems, they provide us with a model that can be easily customized for other domains outside of this subject. The core principle of identifying and addressing the diverse knowledge backgrounds of students, bridging interdisciplinary gaps, and creating adaptive learning experiences is universally applicable. Therefore, by adapting this framework, educators in various disciplines can create more inclusive and effective learning environments that cater to the diverse needs of their students.

3.2 Adaptive Assessments

Traditional static assessments often struggle to capture whether a student has truly understood the material or provide opportunities to address gaps in their knowledge. Although giving a student one chance to answer a question provides some indication of their comprehension, it does little to help them identify and address areas of misunderstanding. In addition, instructors are typically limited in their ability to create a wide variety of assessments, making it challenging to offer the repeated and varied testing opportunities necessary for comprehensive learning.

In CS249r, for example, students tackle complex tasks such as optimizing neural networks for GPU execution. A traditional assessment might ask all students to implement this optimization, regardless of their background. However, this approach does not address the diverse starting points of students. For instance, a student

with strong machine learning theory but limited GPU knowledge would struggle differently than one with extensive systems experience but less algorithmic understanding. An adaptive assessment could initially gauge each student’s strengths, then provide tailored subtasks and resources, ensuring a more effective and personalized learning experience for all.

An effective alternative is an adaptive assessment approach that provides students with multiple opportunities to test their knowledge, focus on areas of weakness, and revisit the material in a meaningful way [48], [49]. This approach is particularly valuable in courses like CS249r, where students’ diverse backgrounds requires personalized learning paths. This process should not rely on repeated exposure to the same questions, as that would merely test rote memorization or random chance rather than true understanding. Instead, adaptive assessments should dynamically adjust to focus on the concepts the student has yet to master, ensuring a more targeted and effective learning experience.

To realize such an adaptive assessment approach, we leverage an LLM to dynamically generate assessments, providing learners with frequent opportunities to check their understanding and track their progress. This feature is integrated into our online machine learning systems textbook, which includes chapters on a variety of topics such as AI Acceleration, Model Optimizations, and Machine Learning Operations. Each chapter contains extensive content with which students are expected to engage and the information in a chapter is divided into sections like any typical textbook.

When a student visits the website, the text in the chapter is indexed into discrete sections based on HTML heading tags (H1, H2, etc.), and the corresponding text is saved in memory. A button is also added after each section. When a button is clicked, indicating that the learner would like to test their understanding of the material, the associated text is passed to the LLM with the following instructions:

Quiz Generation Prompt:

```
Create a quiz from a CHAPTER SECTION.
The quiz should have 3 questions in JSON format:
- Q1 \& Q2: Directly related to the quote's content.
- Q3: Requires deeper understanding.
Use this JSON template, modifying it as needed:
{"questions": [
  {"question": "Q1 here?",
   "answers": [
     {"text": "A1", "correct": true/false,
      "explanation": "explanation"},
     {"text": "A2", "correct": false,
      "explanation": "explanation"},
     {"text": "A3", "correct": false,
      "explanation": "explanation"},
   ]},
  {"question": "Q2 here?", "answers": [/* options */]},
  {"question": "Q3 here?", "answers": [/* options */]}
]}
QUOTE: ${<insert quote here>}
CHAPTER SECTION ${<insert name and section number of chapter>}$
```

The LLM returns a JSON object containing the questions, multiple-choice options, and detailed answer explanations embedded in the response. This JSON object is rendered for the learner to interact with, providing them with immediate feedback on their quiz.

As students progress through the sections of the textbook and complete quizzes, a knowledge graph is constructed to outline



Figure 3: As students progress through the textbook, a knowledge graph is built, tracking their reading progress and quiz performance for each section.

various topics and subtopics relevant to their coursework, as shown in Figure 3. This graph tracks the content a student has engaged with, their quiz proficiency, and the types of questions they have answered incorrectly. The knowledge graph provides students with an interactive tool to select sections for assessment, enabling them to focus on areas of strength or weakness. Students can also use the graph to seek AI-driven advice on their learning, identify patterns, and receive guidance on what to study next.

3.3 Bounded Learning

Large language models are trained on vast corpora of text that span much of the Internet, equipping them with strong reasoning and comprehension capabilities [50]. However, the specific datasets used during training are often unknown to the public, and the completeness or accuracy of the information they generate cannot be guaranteed. Thus, relying solely on the pre-trained knowledge of the language model introduces the risk of generating content that is inconsistent or not related to the course content curated.

Consider a scenario in CS249r where students are learning about tensor core optimizations for GPU acceleration. A general LLM might provide broad information about GPUs and tensors, but it may not capture the specific techniques and best practices taught in the course. For instance, when asked about optimizing matrix multiplication on tensor cores, the LLM might suggest generic CUDA programming tips rather than the specific tiling and data layout strategies covered in the class lectures and textbook.

Algorithm 1 Fuzzy Paragraph Matching Overview

Require: Query Q , precomputed fingerprints $\text{textMap} = \{(f_i, \text{id}_i, T_i)\}$

Ensure: Top k most similar paragraphs to Q

- 1: **Text Preprocessing:** Prepare T by removing punctuation and converting to lowercase.
- 2: **Fingerprinting:** Compute the fingerprint for T as the average ASCII value of its characters:

$$\text{fingerprint}(T) = \frac{\sum_{c \in T} \text{ascii}(c)}{|T|}$$

- 3: **Precomputed Fingerprint Map:** Store fingerprints f_i for paragraphs T_i :

$$\text{textMap} = \{(f_i, \text{id}_i, T_i)\}$$

- 4: **Finding Candidate Paragraphs:** Given a query text Q with fingerprint f_Q , perform a binary search on textMap to find the closest fingerprints. Select neighboring fingerprints as candidates:

$$\text{candidates}(Q) = \text{findClosest}(f_Q, \text{textMap})$$

- 5: **Refined Similarity (Chunked Levenshtein Distance):** For each candidate C , calculate the Levenshtein distance to Q , representing the minimum edits to transform one string into another. Compute similarity as:

$$\text{similarity}(Q, C) = 1 - \frac{\text{Levenshtein}(Q, C)}{\max(|Q|, |C|)}$$

- 6: **Ranking and Selection:** Rank candidates by similarity and return the top k .
-

Using AI’s unbounded knowledge as a tool may not fully leverage the advantages of the carefully curated content of our textbook. Although pre-trained knowledge of the language model offers valuable information, it is essential to balance this capability with a focus on curated learning resources that were specifically designed for the course. This approach ensures that learners benefit from the accuracy, depth, and coherence of the primary textbook while still having access to broader supplementary information when needed.

Our solution emphasizes textbook material as the primary focus of the language model through in-context prompts, while retaining access to its broader pre-trained knowledge. These curated prompts explicitly bind the LLM to the information in the textbook, providing a prior that biases the model to draw primarily from curated content while still allowing it to explore related concepts as needed. In the tensor core optimization example, this means the LLM would prioritize information from the course materials about specific CUDA programming patterns for tensor cores, while still being able to provide general context about GPU architectures.

To determine which information to include within the in-context prompt, we quickly extract text relevant to any user query. To achieve this efficiently, we use Algorithm 1, which identifies similar paragraphs using fingerprinting and Levenshtein distance.

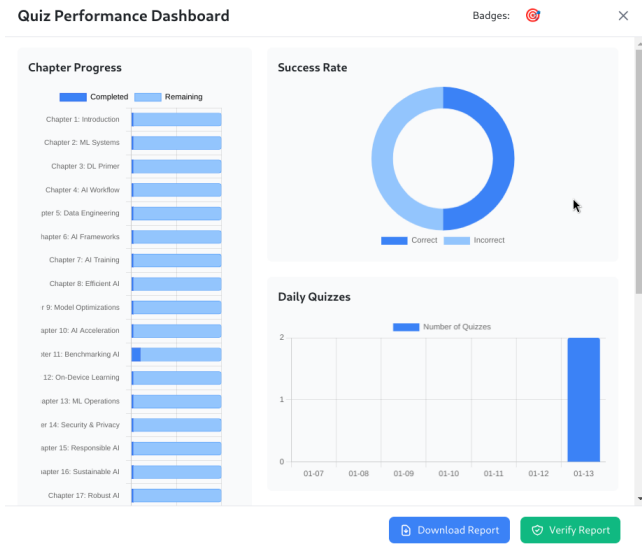


Figure 4: Dashboard showcasing gamification elements, including badges for quiz streaks and progress bars, to enhance engagement and provide learning insights.

Each page that a student visits for the first time undergoes pre-computation, where every diagram (e.g., charts, figures) and paragraph (<p> element in HTML) is assigned a unique fingerprint attribute. This allows us to efficiently find the most relevant paragraphs for any query based on these fingerprints.

By balancing the curated textbook content with the supplementary knowledge from the language model’s training data, this approach ensures that learners receive accurate, focused, and contextually relevant information [51], [52].

3.4 Gamification

An effective AI learning companion should incorporate features that actively engage students and sustain their motivation throughout the learning process. Gamification has been shown to be an effective strategy for increasing enjoyment in educational contexts [53]. Research has also highlighted the mass appeal of gamification in stimulating learner motivation and promoting active engagement [54]. With these elements, educators can create engaging learning experiences that encourage both participation and persistence.

SocratiQ uses gamification and progress tracking to enhance user engagement and provide opportunities for self-assessment. We achieve this by implementing five features designed to motivate learners:

- (1) **Progress:** Tracks the learner’s completion of sections within a chapter as shown in Figure 4. The number of required sections to pass a chapter and the threshold score for a quiz are parameters set by the administrator.
- (2) **Streaks:** The number of consecutive days a user has taken a quiz.
- (3) **Passing quiz attempts:** The total number of quiz attempts a user has passed.

Badge	Name	How to Earn
	First Steps	Complete your first quiz
	On a Streak	Maintain a streak of perfect scores
	Quiz Medalist	Complete 10 quizzes
	Quiz Champion	Complete 20 quizzes
	Quiz Legend	Complete 30 quizzes
	Quiz AGI Super Human	Complete 40 or more quizzes

Figure 5: Overview of achievement badges students can earn through quizzes.

- (4) **Badges:** Users earn badges for every c number of quiz attempts that pass, where c is a parameter chosen by the administrator, as shown in Figure 5.
- (5) **Engagement Heatmap:** Visualizes user activity, with darker squares for more engagement as shown in Figure 6.

These elements provide users with a sense of accomplishment and motivation to continue using the platform.

4 SocratiQ Implementation

The implementation of the SocratiQ system can be organized into four main stages: Initial Setup, Learning Flow, Quiz Generation, and Progress & Gamification as illustrated in Figure 7. These stages encapsulate the key phases of user interaction and system functionality. Each stage integrates multiple components, spanning a web interface, local database, Azure, and a language model. The following sections detail the purpose and implementation of each stage, highlighting their roles within the overall architecture.

4.1 Initial Setup

The SocratiQ platform is implemented as a client-side application, delivered as a single JavaScript file. It is designed to integrate into any webpage, making it a versatile solution to embed AI-driven learning tools on various educational platforms. When the user enables SocratiQ, the JavaScript code injects itself into the Shadow DOM. This creates an isolated environment, ensuring that SocratiQ’s styles and functionality remain independent of the host webpage. Using Shadow DOM, SocratiQ can integrate into any online environment without disrupting existing website elements or being affected by external styles or scripts. During the initialization process, the JavaScript code extracts text from the webpage, in our case, chapters from our online machine learning systems textbook. Indexes the content into discrete sections and dynamically injects quiz buttons after each section of the textbook chapter.

Once the system is enabled, the user can personalize their academic level using a slider, as described in Section 3.1. The platform stores these preferences locally in the web browser’s built-in database, IndexedDB, ensuring both privacy and fast access without relying on server-side storage. Once this setup is complete, the SocratiQ learning companion is fully operational and ready to deliver a personalized and engaging learning experience.

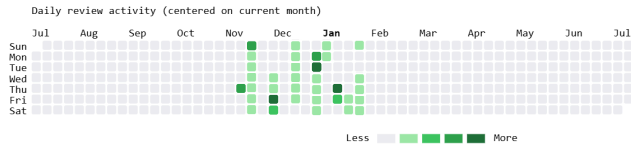


Figure 6: Socratiq displays a visualization of the frequencies of one’s engagement with the app, inspired by the Github Contribution heatmap.

4.2 Learning Flow

In our use case of an online textbook, the AI learning companion comes to life when the student simply highlights text on the page that they need clarification on. This intuitive interaction keeps the student engaged with the material, seamlessly integrating the companion into the learning process without redirecting them to another page or disrupting their focus. For instance, in CS249r, a student might highlight a passage about cache coherence protocols in multicore systems, seeking a more detailed explanation.

When a student highlights a section of text, the selected content is sent through an Azure Function. Our architecture is designed to be serverless to minimize complexity and relieve instructors of the need to manage infrastructure. Azure Functions automatically handle server provisioning, scaling, and maintenance, providing a highly scalable and efficient solution for processing requests. This design ensures that the system can handle multiple simultaneous user interactions without requiring dedicated server resources.

The Azure Function forwards the request to a language model via an API call. Using the stored preferences set during the initial setup (e.g. academic level), the language model generates an explanation tailored to the learner’s needs. In our CS249r example, the model might provide a more in-depth explanation of how cache coherence protocols maintain consistent shared memory state across multiple cores, adjusting the complexity based on the student’s background.

Note that these language models are hosted in the cloud and that we access them via API calls rather than hosting the models on our own servers. While this offloading reduces the complexity of maintaining high-performance infrastructure, it incurs a cost per API call, which will be discussed further in Section 5.3. Once the language model returns the generated explanation, it is formatted for readability and displayed to the student.

4.3 Quiz Generation

At the end of a chapter section, students can select the embedded quiz button to initiate a quiz. For example, in CS249r, after studying a section on GPU architecture and CUDA programming, a student might choose to test their understanding using a generated quiz. A similar pipeline is followed as when a student highlights text: the quiz request is sent through an Azure function.

The response is then formatted and displayed to the user. A key aspect of this process is the use of the prompt structure described in Section 3.2, which ensures that the language model generates questions relevant to the selected content, at the appropriate academic level, and in the correct format. In the CS249r course, this might result in questions about different quantization techniques (such as post-training quantization or quantization-aware training), the

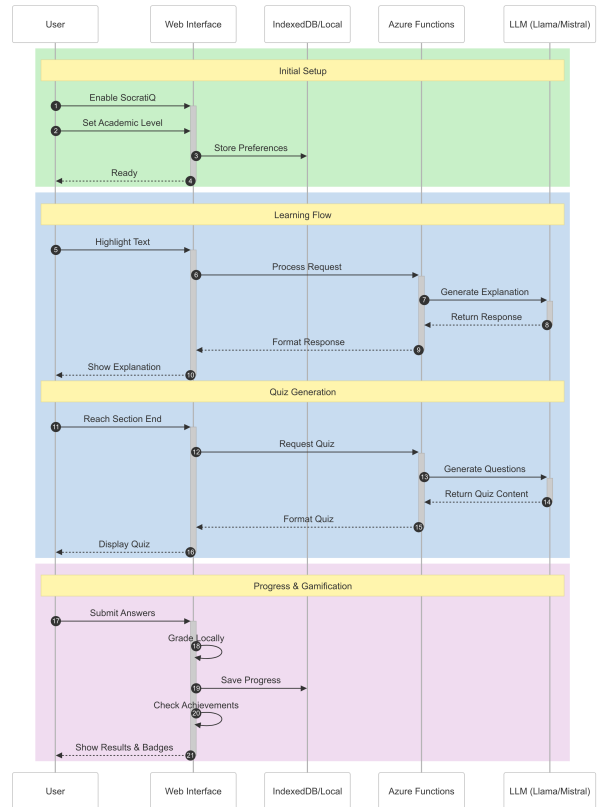


Figure 7: SocratiQ system architecture.

trade-offs between model size and accuracy in pruning, or specific strategies for structured and unstructured pruning, all tailored to the student’s current level of understanding.

The amount of information sent to the language model for generating questions is constrained by token limits, and the strategies used to optimize this process are discussed in Section 5.2. These optimizations are particularly crucial for complex topics in CS249r, where a concise yet comprehensive context is necessary to generate meaningful questions about advanced concepts such as the impact of quantization on different neural network architectures or the iterative process of pruning and fine-tuning.

4.4 Progress & Gamification

Upon completing a quiz, the student’s grade is saved in the browser’s IndexedDB, which ensures that their performance data remain accessible without requiring server-side storage. This local storage approach also records detailed quiz statistics, current progress, and recent learning streaks, enabling comprehensive tracking of user engagement and achievement.

For CS249r students, this might include specific metrics on their understanding of model quantization techniques or pruning strategies. For instance, the system could track a student’s proficiency in post-training quantization versus quantization-aware training, or their ability to analyze the trade-offs between model size reduction and accuracy preservation in various pruning scenarios.

These statistics are combined with gamification elements to provide students with a clear and motivating display of their progress. Beyond simple badges, the system incorporates challenges tailored to each student’s learning trajectory. For example, a student who excels in theoretical aspects of model optimization might be challenged to apply these concepts in practical scenarios, such as optimizing a machine learning model for edge device deployment.

5 Operationalizing SocratiQ

This section provides insights and practical considerations for operationalizing the SocratiQ AI learning companion. We discuss various language model options, practical optimizations for handling API call limitations, cost analysis, and how the system can be integrated into the broader structure of a course.

5.1 Language Models

We use Groq [55], a cloud-based service, to run inference on language models for SocratiQ. It offers several key advantages: fast inference times, needed for maintaining real-time interactivity in our educational platform; elimination of the need for local deployment, avoiding the substantial costs and maintenance associated with server-grade GPUs; and scalability to handle varying loads without infrastructure management on our part. However, Groq, much like any other cloud-based service, imposes specific API call limitations, including constraints on Requests per Minute, Requests per Day, Tokens per Minute, and Tokens per Day.

A unique challenge we face is that our CS249r online textbook is open to the public as it is a website; it has been visited by over 30,000 users in the last 6 months alone. Therefore, it can potentially generate a large volume of requests from users around the world. This open access policy, aligned with our goal of democratizing education, requires a flexible and cost-effective approach to handling model inference. To address this, we use a combination of language models, including Mixtral-8x7b, Gemma 7b, and Llama 3.2, as our primary AI services. By distributing requests across these models, we optimize performance while staying within the free-tier’s limits.

In the event of Groq unavailability or when demand exceeds the free tier limits, we rely on other services such as Google Gemini as a backup service to ensure uninterrupted functionality. This multi-model, multi-service strategy not only provides resilience but also allows us to leverage the strengths of different models for various educational tasks while managing costs effectively.

Note, however, that when SocratiQ is used within a controlled classroom setting, we have the option to switch to more dedicated services with higher capacity and fewer restrictions. However, for the public-facing version of our textbook, our strategy optimizes cost efficiency by prioritizing the use of free tiers and switching to paid instances only when absolutely necessary. This approach enables us to maintain high-quality AI-driven educational experiences for a global audience while also managing operational costs.

5.2 Optimizations

The largest amount of text passed to the language models occurs during quiz generation, which requires loading information about an entire section of a chapter. However, passing an entire chapter to the language model is impractical due to token limitations and

real-time requirements, which constrain the amount of text that can be processed in a single request. This challenge is significant given the length of CS249r’s chapters, which average approximately 10,508 tokens, with some chapters exceeding 40,000 tokens.

5.2.1 Token Management and Vectorization. To keep the total tokens within a limit of $l = 5000$ during quiz generation, our system employs a selective text inclusion strategy. This approach keeps the input concise while preserving the context necessary for generating meaningful questions. Specifically, we vectorize only the first k sentences from each section of the chapter, where k is dynamically tuned to balance the trade-off between capturing sufficient context and adhering to the token budget.

The value of k is adjusted based on the complexity and length of the chapter. For instance, in more dense chapters like “AI Training” (41,434 tokens) or “Robust AI” (32,548 tokens), k might be smaller to ensure coverage of all critical sections. Conversely, for shorter or more introductory chapters like “Introduction” (14,307 tokens), a larger k value can be used to capture more detailed context.

We achieve this optimization efficiently within the browser by implementing a simplified yet fast Word Co-Occurrence Matrix to vectorize the text. This computationally lightweight solution operates entirely on the client-side, aligning with the constraints of a browser environment while ensuring robust performance.

The details are outlined in Algorithm 2. This method reduces text size while retaining the most relevant information for quiz generation. It allows us to generate meaningful quizzes even for our most extensive chapters, such as those covering complex topics like privacy and security (29,728 tokens) or data engineering (26,237 tokens), without overwhelming the language model’s token limit. The adaptive nature of our approach makes sure that we can maintain consistent quiz quality across our diverse range of topics, from fundamental concepts in the deep learning primer to advanced discussions in hardware acceleration and sustainable AI.

5.2.2 Question Caching and Reuse Strategy. We also implemented a question caching and reuse strategy to further optimize system performance and reduce costs. This approach is particularly effective given the substantial length of our chapters, which average approximately 7,506 tokens, with some chapters exceeding 29,000 tokens. As students engage in quizzes, our AI generates 3 to 5 questions tailored to the selected sections of the knowledge graph. These questions are saved, gradually building a comprehensive repository for each section. Once ten quizzes are accumulated per section, we initiate a balanced approach of recycling previous questions while continuing to generate new ones. If the repository reaches a predefined threshold of n questions, we primarily utilize the saved questions, reducing the need for frequent AI-generated content.

This strategy significantly decreases the number of API calls to language models over time, leading to substantial cost savings and reduced computational load. The system remains responsive to user needs; if a student requests regeneration or downvotes a quiz, we discard those specific items and generate new ones, maintaining content quality and relevance. The large pool of questions enables more diverse and tailored assessments, supporting individualized learning paths even for our most extensive chapters. This dynamic and evolving repository not only enhances the learning experience, but also contributes to the long-term sustainability and scalability

Table 3: Cost analysis for serverless functions using various providers and ai models.

Provider	Model Host	Model	Students	API Calls/Day	Days/Week	Weeks	Total Calls	In Cost	Out Cost	Total Cost
<i>Estimated Monthly Costs (4 Weeks)</i>										
Azure	Groq	Mixtral-8x7b	20	30	5	4	12000	2.88	2.88	5.76
Azure	OpenAI	GPT-4	20	30	5	4	12000	30.00	120.00	150.00
Azure	Google	Gemini	20	30	5	4	12000	0.84	3.60	4.44
AWS	Groq	Mixtral-8x7b	20	30	5	4	12000	3.36	3.36	6.72
AWS	OpenAI	GPT-4	20	30	5	4	12000	36.00	144.00	180.00
AWS	Google	Gemini	20	30	5	4	12000	1.00	4.32	5.32
Cloudflare	Groq	Mixtral-8x7b	20	30	5	4	12000	2.40	2.40	4.80
Cloudflare	OpenAI	GPT-4	20	30	5	4	12000	25.00	100.00	125.00
Cloudflare	Google	Gemini	20	30	5	4	12000	0.70	2.88	3.58
<i>Estimated Semester Costs (16 Weeks)</i>										
Azure	Groq	Mixtral-8x7b	20	30	5	16	38400	9.12	9.12	18.24
Azure	OpenAI	GPT-4	20	30	5	16	38400	96.00	384.00	480.00
Azure	Google	Gemini	20	30	5	16	38400	2.64	11.52	14.16
AWS	Groq	Mixtral-8x7b	20	30	5	16	38400	10.72	10.72	21.44
AWS	OpenAI	GPT-4	20	30	5	16	38400	115.20	460.80	576.00
AWS	Google	Gemini	20	30	5	16	38400	3.20	13.76	17.00
Cloudflare	Groq	Mixtral-8x7b	20	30	5	16	38400	7.68	7.68	15.36
Cloudflare	OpenAI	GPT-4	20	30	5	16	38400	80.00	320.00	400.00
Cloudflare	Google	Gemini	20	30	5	16	38400	2.24	9.12	11.36

of the SocratiQ platform, providing valuable content for potential future updates to the textbook.

The combination of our token management strategies and this question caching system allows us to efficiently handle the varying lengths and complexities of our chapters. For instance, in more dense chapters like "Training" (29,596 tokens) or "Robust AI" (23,249 tokens), the caching system is particularly beneficial, reducing the need for frequent, token-heavy question generation. Similarly, for shorter or more introductory chapters like "Introduction" (10,219 tokens), the system can maintain a diverse question set without overreliance on the language model.

This multifaceted optimization approach ensures that SocratiQ can deliver high-quality, personalized learning experiences across our diverse range of topics, from fundamental concepts in the deep learning primer to advanced discussions in hardware acceleration and sustainable AI, while maintaining efficiency in both computational resources and costs. By intelligently balancing the use of AI-generated content with cached questions, we create a system that is both responsive to individual student needs and scalable for widespread use in various educational contexts.

5.3 Cost Analysis

The SocratiQ AI learning companion is highly dependent on API calls to language models, which introduces an associated cost. It is important to provide an estimate of these costs to help instructors budget for deploying such a system in their classrooms.

To estimate costs, we consider the following scenario: a class of 20 students, each making 30 API calls per day, 5 days a week. Table 3 provides a detailed breakdown of costs between different platforms (Azure, AWS, and Cloudflare) and various language models. The

table evaluates two deployment scenarios: the cost for a single month and the cost for an entire semester (16 weeks).

From the analysis, Mixtral-8x7b and Gemini emerge as the most cost-effective options in all scenarios. Semester costs for Mixtral-8x7b range from \$15.36 to \$21.44, while Gemini costs range from \$11.36 to \$17.00, depending on the provider. However, these cost savings come with trade-offs in performance. For instance, Mixtral-8x7b is a 7-billion-parameter model, which often has inferior performance compared to larger models like Gemini, estimated to have over a trillion parameters. However, GPT-4, while likely providing the highest reasoning capabilities, incurs the highest costs. Semester costs for GPT-4 typically exceed \$400.00 in most providers. This highlights a critical trade-off between cost and performance when selecting a language model for use in a classroom setting.

In particular, Cloudflare provides the lowest costs across all models, followed by Azure and AWS. It is important to emphasize that these cost estimates are based on the prices provided by the cloud providers and the AI model vendors. Actual costs may vary due to factors such as location, usage patterns, and additional expenses. Additionally, pricing and performance characteristics may change over time as new models are released and pricing evolves.

Additional cost savings are achieved through our question caching and reuse strategy, detailed in Section 5.2.2, which significantly reduces the number of API calls required over time.

5.4 Scalability

The open source nature of our CS249r textbook presents scalability challenges, as it potentially attracts a global audience with unpredictable usage patterns. To address this, SocratiQ is designed for efficient scalability, utilizing a "local first" approach that minimizes back-end demands. This design is crucial to maintain performance

Algorithm 2 Cryptographic Hashing of PDF

Require: Quiz progress stats P **Require:** Unique user ID U **Require:** Secret key S (stored on server)**Ensure:** Cryptographically hashed PDF H

- 1: Generate a unique UUID $u \leftarrow \text{UUID}()$
 - 2: Hash the quiz progress stats and UUID: $h \leftarrow \text{Hash}(P, u)$
 - 3: Combine the hashed quiz progress stats with the secret key: $k \leftarrow \text{Hash}(h, S)$
 - 4: Insert the unique code k at the end of the PDF content
 - 5: Compute the cryptographic hash of the PDF content with the unique code: $H \leftarrow \text{Hash}(\text{PDF Content}, k)$
 - 6: Store the hashed PDF content H on the server, associated with the secret key S
 - 7: **return** PDF with unique code k
-

and cost effectiveness, regardless of whether the system is used by individual learners worldwide or in structured classroom settings.

Most of the compute-intensive tasks, such as tracking quiz progress, generating adaptive feedback, and performing quick content searches, are handled directly on the user's device. This approach significantly reduces the strain on centralized resources and ensures that the platform remains responsive, even with a large and geographically diverse user base. By pushing these operations to the client side, we can support a virtually unlimited number of simultaneous users without proportional increases in server load or costs.

For tasks requiring advanced processing, such as generating quizzes or handling complex queries, SocratiQ utilizes a serverless architecture hosted on Azure. This architecture scales automatically on demand, allowing it to accommodate spikes in user activity without requiring dedicated server infrastructure. This is particularly beneficial for an open-source textbook, where usage can fluctuate dramatically, for instance, during exam periods or when the textbook is featured in online learning communities. When usage decreases, the system seamlessly "descales," ensuring efficient resource use and lower operational costs.

The combination of local device computing and serverless scalability makes it possible to provide advanced AI assistance to a wide audience, from individual self-learners across the globe to structured institutional settings, while keeping costs low and performance high. This scalability strategy not only supports the open-access philosophy of our textbook but also ensures that the quality of the learning experience remains consistent, whether the user is a single student exploring machine learning systems or part of a large cohort in a university course.

In addition, this approach to scalability aligns with the evolving landscape of educational technology, where learning is increasingly happening both inside and outside traditional classroom boundaries. Designing SocratiQ to be highly scalable and adaptable allows us to support various learning scenarios. A prime example of this versatility is how SocratiQ supports both CS249r, offered locally at Harvard, and its online counterpart, the Professional Certificate in Tiny Machine Learning (TinyML) program available through HarvardX [56]. While CS249r students benefit from SocratiQ in a traditional classroom setting, learners enrolled in the online TinyML program can

Algorithm 3 Verification of PDF

Require: PDF with unique code k **Require:** Secret key S (stored on server)**Ensure:** Verification result V

- 1: Extract the unique code k from the end of the PDF content
 - 2: Compute the cryptographic hash of the PDF content with the extracted unique code: $H' \leftarrow \text{Hash}(\text{PDF Content}, k)$
 - 3: Send the hashed PDF content H' and the extracted unique code k to the server
 - 4: On the server, compute the cryptographic hash of the PDF content with the stored secret key S and the extracted unique code k : $H'' \leftarrow \text{Hash}(\text{PDF Content}, k)$
 - 5: **if** $H' == H''$ **then**
 - 6: **return** $V = \text{Verified}$
 - 7: **else**
 - 8: **return** $V = \text{Not Verified}$
 - 9: **end if**
-

take advantage of the same resource remotely, demonstrating how our platform seamlessly adapts to different educational contexts. This dual application showcases SocratiQ's ability to maintain its interactive and personalized learning experience across various scales, from a focused classroom environment to a massive open online course (MOOC) setting with potentially thousands of concurrent users worldwide. By supporting both local and global learning initiatives, SocratiQ shows its capacity to scale effectively while preserving the quality and personalization of the learning experience, regardless of the educational format or number of users.

5.5 Integration with Course

While SocratiQ can function as a standalone learning tool, its true potential is realized when integrated into the broader course structure. This integration is particularly valuable for instructors seeking to monitor and support student progress effectively. SocratiQ facilitates this by allowing students to securely share their progress with instructors through a system of cryptographically hashed PDFs.

This secure sharing mechanism serves multiple purposes. It allows instructors to gain insight into individual student engagement and performance, enabling timely interventions or support when needed. The cryptographic hash ensures the authenticity of the shared data, providing a reliable mechanism for instructors to verify the accuracy of progress reports. While enabling progress sharing, the system maintains student privacy by giving them control over what information is shared and when. Furthermore, with access to detailed progress data, instructors can adapt their teaching strategies to address common challenges or misconceptions identified through SocratiQ interactions.

The process of generating these secure PDFs is described in Algorithm 2, which demonstrates how hashing is employed to create tamper-evident progress reports. This approach not only ensures data integrity, but also streamlines the progress monitoring process, allowing instructors to focus on providing targeted support and enhancing the overall learning experience.

To ensure that the authenticity of the PDF file can be verified or to prevent students from altering the quiz scores offline, Algorithm

3 is used. This verification process guarantees that the PDF shared by the student has not been tampered with. The unique code k acts as a digital signature, and the server verifies its authenticity by recomputing the hash and comparing it with the stored hash. This secure system provides instructors with confidence in the integrity of the data, while maintaining a seamless integration between the SocratiQ learning companion and the broader course infrastructure.

5.6 Privacy

Security and privacy are central to the SocratiQ platform, embodied in its “local-first” architecture. This design prioritizes user privacy and minimizes external data dependencies, ensuring that sensitive information remains under the user’s control.

The privacy-centric approach is the local storage of user data. All chats are securely stored in the browser’s IndexedDB, while quiz statistics are maintained in local storage. This means that information such as progress tracking and assessment results resides on the learner’s device rather than on centralized servers. By keeping data local, we significantly reduce the risk of large-scale data breaches and offer users greater control over their personal information.

When interaction with external systems is necessary, such as for section quiz requests, the data transmitted contains no personal information, maintaining user anonymity. In cases where users ask specific questions, requests do pass through our servers, but we adhere to a strict no-storage policy for these interactions.

To balance privacy with functionality and cost efficiency, we have implemented a selective centralized caching system. User-upvoted questions are saved in a centralized database, allowing us to improve the quality of generated quizzes and reduce operational costs. This approach enhances the learning experience while still safeguarding individual user data. The platform also facilitates secure progress sharing through cryptographically hashed PDFs. This feature allows learners to share their progress with instructors without compromising data integrity, striking a balance between privacy and the need for academic oversight.

By adhering to these privacy-first principles, SocratiQ ensures a strong balance between functionality, cost efficiency, and user data protection. This approach not only safeguards user information, but also fosters trust, encouraging open and honest engagement with the platform. As educational technology continues to evolve, SocratiQ’s commitment to privacy positions it as a responsible and user-centric solution in the AI-assisted learning landscape.

6 Evaluation

We examine SocratiQ’s effectiveness in supporting self-paced learning through two main approaches: an analysis of AI-generated questions and a limited case study in a sprint-style machine learning course. In the first part of our evaluation, we focus on assessing the quality of AI-generated questions. We analyze questions across various topics, using Bloom’s Taxonomy to evaluate their cognitive depth and relevance to the course material. This analysis provides insights into the system’s ability to generate diverse and meaningful assessments. The second part of our evaluation presents findings from a small-scale implementation of SocratiQ in a real learning environment. While limited in scope, this case study offers

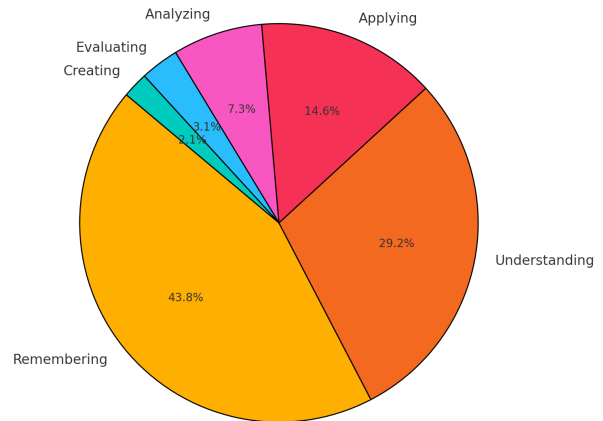


Figure 8: Distribution of generated questions by cognitive level based on Bloom’s Taxonomy.

preliminary insights into how students interact with the system and its potential impact on engagement and learning outcomes.

6.1 Experimental Setup

Our experimentation was conducted in a seven-week machine learning systems course designed to maximize the benefits of self-paced learning while maintaining structured guidance. Each week featured a two-hour lecture that introduced core ML concepts, followed by student-driven exploration periods where SocratiQ provided personalized support through adaptive assessments and targeted resources. This hybrid approach blended traditional teaching methods with self-directed learning, allowing students to progress at their own pace while ensuring coverage of essential material.

The course consisted of a small cohort of five undergraduate students with varying technical backgrounds, making self-paced learning particularly valuable. The students came from diverse academic disciplines, including computer science, engineering, and measurement technology, with most possessing intermediate programming skills. This diversity in technical proficiency was intended to showcase the ability of SocratiQ to adapt to individual learning needs while maintaining consistent educational outcomes.

SocratiQ played a central role in enabling self-paced learning through its quiz generation and resource recommendation capabilities. By aligning personalized assessments with preassigned readings, the platform allowed students to test their understanding and progress at their own pace.

6.2 Content Quality Analysis

To evaluate the quality and depth of AI-generated questions, we employed Bloom’s Taxonomy [57], [58], which categorizes educational objectives into six hierarchical levels: Remembering, Understanding, Applying, Analyzing, Evaluating, and Creating. This analysis focuses on questions generated at the beginner level, aligning with our goal of providing adaptive, personalized learning experiences

Table 4: Examples of AI-generated questions categorized by bloom’s taxonomy cognitive levels.

Cognitive Level	Example	Generated Question
Remembering	Example 1	What is the definition of a machine learning system?
	Example 2	List three advantages of edge AI over cloud AI.
Understanding	Example 1	Explain the difference between supervised and unsupervised learning.
	Example 2	Summarize the key components of a neural network.
Applying	Example 1	Identify the best algorithm for training a classification model.
	Example 2	Use the gradient descent formula to calculate the next step for the given parameters.
Analyzing	Example 1	Compare and contrast embedded AI frameworks with traditional AI frameworks.
	Example 2	Identify potential bottlenecks in the given machine learning pipeline.
Evaluating	Example 1	Assess the ethical implications of deploying facial recognition technology in public spaces.
	Example 2	Evaluate the trade-offs between model accuracy and computational efficiency.
Creating	Example 1	Design an end-to-end workflow for deploying an edge AI application.
	Example 2	Propose a novel algorithm for handling missing data in time-series datasets.

as outlined in Section 3. We randomly analyzed 100 questions from the small cohort across various topics and chapters, aiming to evaluate the AI system’s support for critical thinking and deep learning, identify gaps in question types, and gain insights for improving the system’s question generation capabilities. This approach allows us to assess how effectively SocratiQ fulfills its role in adaptive assessment for novice learners, as described in Section 3.

Figure 8 illustrates the distribution of beginner-level questions on Bloom’s cognitive levels, revealing a significant imbalance. Lower-order cognitive skills dominate, with remembering at 42% and understanding at 28%. In contrast, higher-order thinking skills are underrepresented: Applying (14%), Analyzing (7%), Evaluating (3%), and Creating (2%). This distribution is consistent with what can be expected for beginner-level content, focusing on building foundational knowledge and basic comprehension. The prevalence of lower-order questions at the beginner level aligns with the system’s goal of providing comprehensive coverage of fundamental course material. However, it also highlights potential areas for improvement in gradually introducing higher-order thinking skills.

It is important to note that this analysis is specific to the beginner difficulty setting of SocratiQ. As shown in Table 5, the system generates questions of varying complexity across different difficulty levels. In higher difficulty settings, we would expect to see a higher proportion of questions that target higher-order cognitive skills.

These findings have implications for SocratiQ’s adaptive capabilities. Although the system provides appropriate questions for beginners, it could be improved to gradually introduce more challenging cognitive tasks even in the early stages of learning. This could help prepare learners for the more complex questions they will encounter at higher difficulty levels. Moving forward, improving SocratiQ’s ability to incorporate a broader range of cognitive skills, even at the beginner level, could be beneficial. This enhancement would support a smoother transition to higher difficulty levels and potentially increase student engagement in self-paced learning environments, addressing the challenges mentioned in Section 2.

6.3 AI Capability Analysis

While our previous analysis focused on beginner-level questions, SocratiQ’s AI capabilities extend beyond this single difficulty level. To provide an evaluation of the system’s adaptability, we evaluated SocratiQ’s performance in generating questions across various difficulty levels: Beginner, Intermediate, Advanced, and Expert, as discussed in Section 3.1. This analysis complements our earlier Bloom’s Taxonomy assessment by showing how SocratiQ adapts question complexity to suit different levels of learner expertise.

Table 5 gives examples of AI-generated questions categorized by chapters and difficulty levels. This table highlights SocratiQ’s ability to generate questions tailored to specific difficulty levels, ranging from introductory prompts for beginners to complex queries for advanced and expert learners. Our analysis shows several key insights about the system’s performance. As the difficulty level increases from Beginner to Expert, we observe a clear progression in the complexity of questions. This progression aligns with the system’s design goal of providing adaptive learning experiences. The questions show the SocratiQ’s ability to adapt not only to difficulty levels but also to specific topic areas within the course. For instance, questions in the “DNN Architectures” chapter progress from basic definitions to intricate details of neural network operations.

Regarding cognitive skill targeting, we found that while beginner and intermediate levels tend to focus more on foundational knowledge (aligning with the lower levels of Bloom’s taxonomy), advanced and expert levels provide greater opportunities for critical thinking and problem solving. This shift corresponds to higher levels of Bloom’s taxonomy, such as Analyzing, Evaluating, and Creating. The language and concepts used in the questions become progressively more sophisticated and domain-specific as the difficulty level increases. For example, in the “Efficient AI” chapter, the beginner question asks about the basic terminology of floating-point representation, while the expert question requires learners to explain the benefits of BF16 over FP16 and its implications for deep learning applications. Similarly, in the “Model Optimizations”

Table 5: Examples of AI-generated questions categorized by chapters and difficulty levels.

Chapter	Difficulty Level	Generated Question
DNN Architectures	Beginner	What is the core operation in a Convolutional Neural Network (CNN)?
	Intermediate	What are the two outer loops in the given nested loops used for in Convolutional Neural Networks (CNNs)?
	Advanced	Memory requirements can be reduced using which type of neural network under appropriate convolutional configurations given spatially structured computations?
	Expert	Which compute pattern allows for processing a local 3x3 region of input values across all input channels through the actual convolution operation, moving across all input and output spatial positions?
Efficient AI	Beginner	Floating point: Known as a single-precision floating point, FP32 utilizes 32 bits to represent a number, incorporating its sign, _____ and mantissa.
	Intermediate	Why is reduced numeric precision important for hardware acceleration?
	Advanced	What are the advantages of using lower precision integer numbers during the inference phase with accelerators and GPUs optimized for those operations?
	Expert	Explain the benefits of using BF16 over regular FP16. Why is it considered an efficient alternative for deep learning applications?
Model Optimizations	Beginner	What is the first step in the illustrated process of network pruning?
	Intermediate	How is computational efficiency increased through structured pruning?
	Advanced	What is the main difference between iterative pruning and one-shot pruning?
	Expert	Choose a fitting statement for situations that can lead to better overall results when using one-shot pruning compared to iterative pruning.

chapter, the progression moves from identifying the first step in network pruning at the beginner level to comparing one-shot and iterative pruning techniques at the expert level. This progression helps learners develop a more nuanced understanding of the subject matter. In addition, higher difficulty levels often require learners to apply their knowledge to specific scenarios or compare different concepts, fostering a deeper understanding of the material.

Despite these strengths, we also identified areas for improvement in SocratiQ’s question generation capabilities. Although the system effectively differentiates between difficulty levels, there could be more gradual transitions, especially between intermediate and advanced levels. The degree of difficulty progression varies somewhat between different chapters, and standardizing this progression could enhance the overall learning experience. Furthermore, even at beginner and intermediate levels, there is potential to incorporate more questions that encourage critical thinking.

This analysis demonstrates SocratiQ’s capability to generate diverse, difficulty-appropriate questions across various topics. The system’s ability to adapt question complexity aligns with our goal of providing personalized learning experiences, as outlined in Section 3. However, there remains room for refinement to ensure a smoother progression of difficulty and to incorporate higher-order thinking skills more consistently across all levels. These insights will guide future improvements to SocratiQ, enhancing its ability to support learners at various stages of their educational journey.

6.4 Student Feedback and Experiences

To complement our quantitative analysis of SocratiQ’s AI capabilities and question quality, we gathered feedback (in a systematic

manner) from students across multiple topics who used the system during the sprint-style machine learning course. These qualitative data provide us with important insights into the student experience and the effectiveness of SocratiQ as a learning tool.

The *ease of use and the design of the interface* of SocratiQ received mixed feedback. Although one student found it straightforward, stating:

“ It’s easy to open it and to complete a quiz.

Others encountered some confusion with the interface:

“ I’m confused about why all levels of the quiz are displayed in a section, but I can only submit one specific level.

In terms of *learning enhancement*, students found SocratiQ helpful. One student noted its effectiveness in reinforcing knowledge:

“ It can help me deepen my memory of conceptual knowledge points.

Another student highlighted how the system encouraged active engagement with the material:

“ Questions do get you to think about what you have read.

The *AI interaction capabilities* of SocratiQ were particularly appreciated. One student shared a specific example of how the AI provided relevant information for them.

“ When I asked him about the use of robustai, he gave me many examples.

Moreover, SocratiQ’s ability to foster *critical thinking* was emphasized by several students. One particularly insightful comment highlighted the interactive nature of the system:

“ Yes, with the help of SocratiQ’s Q&A system, I can ask questions whenever they come up, instead of the one-way input of traditional reading. This interactive approach allows me to think more critically about the book’s content.

When comparing SocratiQ to traditional study methods, students found the system more engaging:

“ SocratiQ provided an interactive experience, making studying more engaging compared to passive reading or note-taking.

The *gamification features* of SocratiQ received mixed responses. Although some students found them motivating:

“ This would be very effective for me because I really enjoy collecting badges.

Others had a more nuanced view:

“ I liked the visual stats, not really interested in the badges, but the students might appreciate them.

The students also provided valuable suggestions *for improvement*. One student proposed enhancing the credibility of the system:

“ If the related section includes not only internal answers but also external links, it might be more credible and less likely to be dismissed as an AI hallucination.

Another suggested adding more interactive features:

“ Maybe add a section that can let the students or readers discuss for each chapter, such as questions and related information.

These student experiences highlight both the strengths of SocratiQ and areas for potential improvement. They underscore the system’s effectiveness in providing interactive, engaging learning experiences while also pointing to opportunities for enhancing user interface design, question diversity, and customization options to better cater to individual learning preferences.

The qualitative feedback largely aligns with our quantitative analysis of SocratiQ’s AI capabilities. Students’ appreciation of the interactive Q&A system and immediate feedback corroborates our findings on the system’s ability to generate relevant and helpful responses. However, the comments about user interface confusion and the need for more diverse content types suggest areas where further refinement could enhance the system’s effectiveness in supporting self-paced learning.

7 Discussion

Our implementation of SocratiQ provides insight into how future educational systems might evolve, particularly as language models continue to advance at an unprecedented pace. The development of more capable models, from GPT-4 to Gemini and beyond, suggests that the capabilities shown in this paper are only the beginning of what will be possible in AI-enhanced education.

The next generation of educational AI systems will likely be shaped by several key advances. The increasing sophistication of language models will enable a more nuanced understanding of student responses and the generation of sophisticated questions, moving beyond current constraints in token management and prompt engineering. These advances could transform the way students interact with educational content, enabling more natural, conversational learning experiences. The evolution of multi-modal AI capabilities will enable seamless integration of visual, auditory, and interactive elements, particularly valuable in technical education, where understanding often requires engaging with diagrams, equations, and dynamic visualizations.

Also, we believe that cost considerations will likely shift as language model deployment becomes more efficient, with significant implications for educational access in developing nations. While our current multi-model approach using Mixtral-8x7b and Gemini helps manage expenses, future optimizations could make sophisticated AI tutoring more widely accessible. Our experiments with local model deployment and aggressive caching strategies point toward solutions for regions with limited connectivity or financial resources. Future systems could implement hybrid approaches where smaller and more efficient models run locally while maintaining access to cloud-based models when needed. The increasing availability of multilingual models further opens possibilities for delivering personalized education in local languages, addressing crucial barriers to educational access in many developing regions.

Our experience in developing SocratiQ for CS249r has already highlighted challenges in teaching complex technical content. Machine learning systems courses require students to simultaneously grasp concepts that span data engineering, model architecture, and computational infrastructure. When discussing neural network optimization, for example, students must understand the interplay between model architecture choices, hardware capabilities, and performance requirements. Future AI learning companions will need sophisticated mechanisms to dynamically adjust their depth of explanation. When a student asks about quantization, the system must determine whether to focus on mathematical principles, hardware implications, or practical implementation considerations—a challenge that becomes more complex across different institutions and student populations with varying technical backgrounds.

Alas, the integration of AI into education will only likely accelerate dramatically in the coming years. While the architecture and implementation strategies presented here provide a foundation for future development, the field must evolve rapidly to harness new capabilities. Ultimately, the goal remains to enhance rather than replace traditional educational approaches, as discussed in Section 2. The next generation of AI-powered learning companions must navigate the complex intersection of technological capability and pedagogical effectiveness, ensuring advances in AI technology translate into meaningful improvements in educational outcomes.

8 Conclusion

Education faces a persistent challenge in addressing the unique needs of individual learners. Traditional teaching methods, while effective in delivering foundational knowledge, often fail to provide personalized and interactive learning experiences. In this paper, we introduce SocratiQ, an AI-powered learning companion rooted in the principles of personalized education, to address these gaps. Through personalized explanations, gamification features, and adaptive assessments, SocratiQ fosters deeper comprehension, active participation, and higher-order thinking. Its integration into the CS249R Machine Learning Systems course demonstrated its potential to bridge the gap between traditional and AI-based education.

Acknowledgments

The authors thank the Harvard LLnc Faculty Fellowship program for its support in pedagogical design and curriculum development. The authors also acknowledge the National Science Foundation for supporting student research through the NSF Graduate Research Fellowship. The authors are grateful to the Harvard Extension School and the Harvard Data Science Initiative (HDSI) for their financial support in the development of this project.

References

- [1] M. Aitkin and R. Zuzovsky, "Multilevel interaction models and their use in the analysis of large-scale school effectiveness studies," *School effectiveness and school improvement*, vol. 5, no. 1, pp. 45–73, 1994.
- [2] H. Ebmeier and T. L. Good, "The effects of instructing teachers about good teaching on the mathematics achievement of fourth grade students," *American Educational Research Journal*, vol. 16, no. 1, pp. 1–16, 1979.
- [3] J. V. Ebenezer and U. Zoller, "Grade 10 students' perceptions of and attitudes toward science teaching and school science," *Journal of research in science teaching*, vol. 30, no. 2, pp. 175–186, 1993.
- [4] M.-C. Opendakker and J. Van Damme, "Teacher characteristics and teaching styles as effectiveness enhancing factors of classroom practice," *Teaching and teacher education*, vol. 22, no. 1, pp. 1–21, 2006.
- [5] J. Wei, X. Wang, D. Schuurmans, et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [6] L. Phan, A. Gatti, Z. Han, et al., "Humanity's last exam," *arXiv*, 2025.
- [7] T. Sawada, D. Paleka, A. Havrilla, et al., "Arb: Advanced reasoning benchmark for large language models," *arXiv preprint arXiv:2307.13692*, 2023.
- [8] D. Rein, B. L. Hou, A. C. Stickland, et al., "Gpqa: A graduate-level google-proof q&a benchmark," *arXiv preprint arXiv:2311.12022*, 2023.
- [9] E. Glazer, E. Erdil, T. Besiroglu, et al., "Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai," *arXiv preprint arXiv:2411.04872*, 2024.
- [10] D. Hendrycks, C. Burns, S. Kadavath, et al., "Measuring mathematical problem solving with the math dataset," *arXiv preprint arXiv:2103.03874*, 2021.
- [11] M. Safdari, G. Serapio-García, C. Creppy, et al., "Personality traits in large language models," *arXiv preprint arXiv:2307.00184*, 2023.
- [12] J. Wei, M. Bosma, V. Y. Zhao, et al., "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.
- [13] D. M. Ziegler, N. Stiennon, J. Wu, et al., "Fine-tuning language models from human preferences," *arXiv preprint arXiv:1909.08593*, 2019.
- [14] L. Fiorella and R. E. Mayer, "Eight ways to promote generative learning," *Educational psychology review*, vol. 28, pp. 717–741, 2016.
- [15] C. Learning, *Mathia - personalized math learning software*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.carnegielearning.com/solutions/math/mathia/>.
- [16] A. Corporation, *Aleks - adaptive learning and assessment for k-12, higher education, and continuing education*. Accessed: 2025-01-24, n.d. [Online]. Available: https://www.aleks.com/?_s=7790627437647913.
- [17] D. Learning, *Dreambox learning - personalized math and reading solutions*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.dreambox.com/>.
- [18] W. Ma, O. O. Adesope, J. C. Nesbit, and Q. Liu, "Intelligent tutoring systems and learning outcomes: A meta-analysis," *Journal of educational psychology*, vol. 106, no. 4, p. 901, 2014.
- [19] E. Mousavinasab, N. Zarifsanaiy, S. R. Niakan Kalhori, M. Rakhshan, L. Keikha, and M. Ghazi Saedi, "Intelligent tutoring systems: A systematic review of characteristics, applications, and evaluation methods," *Interactive Learning Environments*, vol. 29, no. 1, pp. 142–163, 2021.
- [20] Duolingo, *Duolingo - learn a language for free*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.duolingo.com/>.
- [21] R. Stone, *Rosetta stone - language learning software*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.rosettastone.com/>.
- [22] Babbel, *Babbel - learn languages online*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.babbel.com/>.
- [23] E. M. Golonka, A. R. Bowles, V. M. Frank, D. L. Richardson, and S. Freynik, "Technologies for foreign language learning: A review of technology types and their effectiveness," *Computer assisted language learning*, vol. 27, no. 1, pp. 70–105, 2014.
- [24] W. Alpha, *Wolfram alpha - computational intelligence*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.wolframalpha.com/>.
- [25] GeoGebra, *Geogebra - dynamic mathematics for everyone*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.geogebra.org/>.
- [26] Labster, *Labster - virtual labs for science education*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.labster.com/>.
- [27] D. A. Huffaker and S. L. Calvert, "The new science of learning: Active learning, metacognition, and transfer of knowledge in e-learning applications," *Journal of Educational Computing Research*, vol. 29, no. 3, pp. 325–334, 2003.
- [28] Grammarly, *Grammarly - ai writing assistant*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.grammarly.com/>.
- [29] Turnitin, *Turnitin - integrity matters*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.turnitin.com/>.
- [30] C. T. F. Me, *Cite this for me - free reference generator*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.citethisforme.com/>.
- [31] Y. Dong and L. Shi, "Using grammarly to support students' source-based writing practices," *Assessing Writing*, vol. 50, p. 100 564, 2021.
- [32] R. O'Neill and A. Russell, "Stop! grammar time: University students' perceptions of the automated feedback program grammarly," *Australasian Journal of Educational Technology*, vol. 35, no. 1, 2019.
- [33] J. Achiam, S. Adler, S. Agarwal, et al., "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [34] G. Team, R. Anil, S. Borgeaud, et al., "Gemini: A family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.
- [35] C. K. Y. Chan and W. Hu, "Students' voices on generative ai: Perceptions, benefits, and challenges in higher education," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, p. 43, 2023.
- [36] A.-P. Correia, S. Hickey, and F. Xu, "Beyond the virtual classroom: Integrating artificial intelligence in online learning," *Distance Education*, pp. 1–11, 2024.
- [37] K. Academy, *Khanmigo - ai-powered virtual tutor by khan academy*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://www.khanmigo.ai/>.
- [38] N. Kshetri, "The economics of generative artificial intelligence in the academic industry," *Computer*, vol. 56, no. 8, pp. 77–83, 2023.
- [39] Duolingo, *Introducing duolingo max - powered by gpt-4*. Accessed: 2025-01-24, n.d. [Online]. Available: <https://blog.duolingo.com/duolingo-max/>.
- [40] B. Marr, "The amazing ways duolingo is using ai and gpt-4," *Forbes*. April, vol. 28, 2023.
- [41] I. Pesovski, R. Santos, R. Henriques, and V. Trajkovic, "Generative ai for customizable learning experiences," *Sustainability*, vol. 16, no. 7, p. 3034, 2024.
- [42] L. Chen, P. Chen, and Z. Lin, "Artificial intelligence in education: A review," *Ieee Access*, vol. 8, pp. 75 264–75 278, 2020.
- [43] I. Morgil, S. Yavuz, Ö. Ö. Oskay, and S. Arda, "Traditional and computer-assisted learning in teaching acids and bases," *Chemistry Education Research and Practice*, vol. 6, no. 1, pp. 52–63, 2005.
- [44] S. Baum and M. McPherson, "The human factor: The promise & limits of online education," *Daedalus*, vol. 148, no. 4, pp. 235–254, 2019.
- [45] A. Renz and R. Hilbig, "Prerequisites for artificial intelligence in further education: Identification of drivers, barriers, and business models of educational

- technology companies,” *International Journal of Educational Technology in Higher Education*, vol. 17, no. 1, p. 14, 2020.
- [46] C. Dhanapal, N. Asharudeen, and S. Y. Alfaruque, “Impact of artificial intelligence versus traditional instruction for language learning: A survey,” *World Journal of English Language*, vol. 14, no. 2, pp. 182–182, 2024.
- [47] A. M. Al-Zahrani and T. M. Alasmari, “Exploring the impact of artificial intelligence on higher education: The dynamics of ethical, social, and educational implications,” *Humanities and Social Sciences Communications*, vol. 11, no. 1, pp. 1–12, 2024.
- [48] J. Hattie and H. Timperley, “The power of feedback,” *Review of educational research*, vol. 77, no. 1, pp. 81–112, 2007.
- [49] V. J. Shute, “Focus on formative feedback,” *Review of educational research*, vol. 78, no. 1, pp. 153–189, 2008.
- [50] T. Brown, B. Mann, N. Ryder, *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
- [51] E. Wenger, *Communities of practice: Learning, meaning, and identity*. Cambridge university press, 1999.
- [52] H. P. Learn, “Brain, mind, experience, and school,” *Committee on Developments in the Science of Learning*, 2000.
- [53] J. Looyestyn, J. Kernot, K. Boshoff, J. Ryan, S. Edney, and C. Maher, “Does gamification increase engagement with online programs? a systematic review,” *PloS one*, vol. 12, no. 3, e0173403, 2017.
- [54] Z. Zainuddin, S. K. W. Chu, M. Shujahat, and C. J. Perera, “The impact of gamification on learning and instruction: A systematic review of empirical evidence,” *Educational research review*, vol. 30, p. 100 326, 2020.
- [55] Groq, *Groq official website*, <https://groq.com/>.
- [56] *Tiny machine learning (tinymtl) professional certificate*, [Online; accessed 2025-01-30]. [Online]. Available: <https://www.edx.org/certificates/professional-certificate/harvardx-tiny-machine-learning>.
- [57] M. Forehand, “Bloom’s taxonomy,” *Emerging perspectives on learning, teaching, and technology*, vol. 41, no. 4, pp. 47–56, 2010.
- [58] D. Krathwohl, “A revision bloom’s taxonomy: An overview,” *Theory into Practice*, 2002.