

Article

SPARK_AI: A Prompt-Orchestrated Architecture for Stateful, Process-Oriented Reasoning with Large Language Models

Marija Kaplar ^{1,*}, Sebastijan Kaplar ¹, Miloš Vučić ², Lidija Ivanović ³, Aleksandra Stevanović ⁴, Aleksandar Milenković ⁵ and Nemanja Vučičević ⁵

¹ University of Novi Sad, Dr Zorana Đinđića 1, 21000 Novi Sad, Serbia; kaplar@uns.ac.rs

² Faculty of Mechanical Engineering, University of Belgrade, Kraljice Marije 16, 11120 Belgrade, Serbia; mvucic@mas.bg.ac.rs

³ Faculty of Education, University of Novi Sad, Podgorička 4, 25101 Sombor, Serbia; lidija.ivanovic@uns.ac.rs

⁴ Faculty of Information Technology, University Metropolitan, Tadeuša Košćuška 63, 11000 Belgrade, Serbia; aleksandra.stevanovic@metropolitan.ac.rs

⁵ Faculty of Science, University of Kragujevac, Radoja Domanovića 12, 34000 Kragujevac, Serbia; aleksandar.milenkovic@pmf.kg.ac.rs (A.M.); nemanja.vucicevic@pmf.kg.ac.rs (N.V.)

* Correspondence: marija.kaplar@uns.ac.rs

Abstract

This paper presents SPARK_AI, a prompt-orchestrated system architecture for governing how large language models (LLMs) conduct structured and adaptive reasoning in human–AI interaction. The framework mitigates ad hoc LLM use by replacing direct answer generation with a process-oriented, step-by-step reasoning workflow. We focus on SPARK_AI_MATH, a domain module that supports learners in solving non-routine problem-solving tasks by operationalizing well-established problem-solving phases and guided questioning dialog strategies (Socratic-style prompts), with an optional tool-mediated visualization layer (e.g., GeoGebra). The module implements a five-phase conversational protocol consisting of problem interpretation, analysis of givens, planning, execution, and reflection, together with a controlled hint policy. This design is realized through a stateful system architecture in which each problem instance is maintained as an independent interaction track with a persistent reasoning state. User acceptance was evaluated by first-year mechanical engineering students ($N = 108$) using an expanded Technology Acceptance Model instrument, and the results were analyzed via PLS-SEM. The findings indicate overall favorable perceptions, with perceived usefulness and learning support emerging as key predictors of intention for continued use. Beyond this specific domain, the SPARK_AI framework enables efficient domain adaptation through localized prompt strategies while preserving a shared cognitive control layer for reasoning-centered human–LLM interaction.

Keywords: modular system architecture; orchestration layer; educational chatbot; stateful interaction; workflow control



Academic Editor: Antony Bryant

Received: 13 February 2026

Revised: 13 April 2026

Accepted: 14 April 2026

Published: 17 April 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Large language model-based conversational systems are now widely used across multiple domains, including education [1–3]. As a result, artificial intelligence has moved from an experimental technology to an influential sociotechnical infrastructure, including a growing presence in formal and informal education [2–7]. Importantly, this transition has shifted attention away from model capabilities alone toward system-level questions

concerning how reasoning processes are structured, controlled, and sustained during human–AI interaction [2]. In domains requiring transparent, incremental reasoning, the lack of explicit workflow and reasoning-control mechanisms remains a key limitation of general-purpose LLM applications [8,9].

In educational settings, the question is no longer whether LLMs will be used, but how interactive LLM-based systems should be designed to support meaningful reasoning. Recent research has examined a range of interaction paradigms in which LLMs function as tutors, assistants, or feedback providers, with a focus on their influence on learning processes [5,6,10,11]. From a systems perspective, the challenge is not only generating fluent responses but regulating how outputs are produced and integrated into user workflows. Their cross-domain applicability and low configuration requirements [6,12–14] have accelerated adoption, yet they expose interaction-design limitations, particularly in controlling reasoning depth, sequencing, and transparency.

Within this broader context of system-level adoption, several benefits of LLM-based conversational systems have been reported, particularly for text-centric tasks such as drafting, revision, and summarization [5]. More recent studies also indicate potential value in instructional settings, including mathematics learning across multiple educational levels [15–17]. Despite these benefits, substantial concerns have been raised regarding the pedagogical implications of unrestricted LLM usage [2,6,11,18,19]. A recurring issue is the tendency of general-purpose chatbots to provide final answers or complete solutions without engaging learners in intermediate problem-solving steps [13,19–21]. This can lead to superficial task completion by bypassing essential processes such as understanding the task, planning a strategy, and executing it stepwise [22,23]. From a pedagogical perspective, skipping these steps undermines the development of conceptual understanding and problem-solving skills, even when the final answer is correct [24,25]. Another challenge is learners' increasing reliance on LLM-based chatbots, particularly when they are unable to solve a task independently [19,20,26,27]. When users lack sufficient domain knowledge, they are less capable of identifying errors, inconsistencies, or hallucinated responses produced by the model [2,28]. This increases the risk of uncritical acceptance of generated solutions and further amplifies the negative effects of step skipping and overreliance on automated assistance [20,23,26,28,29].

Motivated by these challenges, this paper presents SPARK_AI (Socratic–Polya Adaptive Reasoning Kit), an adaptable, prompt-orchestrated system architecture that prioritizes process-oriented reasoning over direct answer generation. SPARK_AI enforces explicit interaction control through stepwise phase progression and a bounded hint policy, while maintaining stateful, per-problem sessions with persistent dialog traces and structured logs to support reproducible interaction trajectories and fine-grained analysis. We instantiate the approach in SPARK_AI_MATH, a domain module that operationalizes Pólya-inspired problem-solving phases and Socratic dialog with optional tool-mediated scaffolding (e.g., GeoGebra). We also report a preliminary user-acceptance evaluation with students, combining descriptive statistics, qualitative analysis, and PLS-SEM (Partial Least Squares-Structural Equation Modeling). The technical solution is implemented as a web-based system that separates a persistence layer from an LLM orchestration service and remains provider-agnostic through an adapter-based provider abstraction, enabling the underlying LLM backend to be swapped without altering the core reasoning protocol; the same control and logging structure also supports efficient domain adaptation through localized prompt strategies.

2. Literature Review

2.1. LLM-Based Conversational Systems for Structured Human–AI Interaction

Recent work reports effective use of LLM-based chatbots for educational support by enabling human-like, on-demand dialog that reduces temporal and spatial constraints and supports self-paced learning [30–33]. Studies also highlight that pedagogical value depends more on conversational continuity and persistent context than on isolated question–answer exchanges, with such dialog structures being used to scaffold time and task management and to support constructs such as grit and growth mindset [6,34]. Given these reported benefits, it is unsurprising that LLM-based chatbots are increasingly adopted as task-support tools across educational domains, with their scaffolding function commonly structured into three phases: pre-task preparation, task implementation, and post-task review [6]. In the preparation phase, they should support brainstorming, planning, outlining, and rapid draft prototyping [35]. During task implementation, they should provide real-time guidance through explanations, pseudocode-level support, progressive hints with varying levels of assistance, and personalization [36]. In the review phase, they should assist with revision and refinement, including language polishing, code debugging, and manuscript improvement while aiming to preserve students’ original ideas and writing style [21]. Also, from a cognitive computing perspective, effective AI systems are expected to integrate perception, reasoning, learning, and adaptation within transparent and accountable system-level architectures, rather than relying solely on opaque, end-to-end model behavior [37]. However, ad hoc use of general-purpose LLM chatbots often reduces support to immediate answer delivery, which learners may accept uncritically and thus bypass the learning process [11]. From a system design perspective, these findings emphasize the importance of conversational continuity, persistent context, and explicit interaction structuring as foundational components of LLM-based cognitive systems [21]. This motivates our contribution: a process-first, stateful tutoring design that enforces phase-based problem solving and controlled hinting to sustain metacognitive engagement while preserving the accessibility benefits of LLM support.

2.2. Reasoning Control and Scaffolding in LLM-Based Mathematics Systems

In a related study, a generative AI learning companion (TALPer) was integrated into the Taiwan Adaptive Learning Platform (TALP) to support mathematics learning through Socratic questioning with Polya’s problem-solving steps to scaffold mathematics learning [15]. The system was evaluated on a fifth-grade “Ratio and Percentage” unit and reports benefits of structured, platform-integrated chatbot support [15]. While this work informed our direction, SPARK_AI targets a broader scope and learner population (high-school and university) and differs in its interaction design, including a distinct hint policy and optional tool-mediated scaffolding (e.g., visualization) to support process-oriented problem solving.

Benchmarking studies of general-purpose LLMs as mathematics tutoring assistants provide further insight into tutoring tasks such as hint generation, step-by-step solution generation, and exercise creation [16,38,39]. The results suggest strong task-dependence and expose a key failure mode in ad hoc tutoring: hints may violate “concealment” by revealing the solution rather than scaffolding progress, motivating explicit control of hint policies [15,16,39,40]. Moreover, geometry is excluded due to the lack of visual input, highlighting the need for tool-mediated support when visualization is required [16,38,41].

Also, several studies indicate that the effectiveness of using LLMs for mathematics learning depends on careful pedagogical design [15,38,39,42,43]. When students first attempt to solve a problem independently and only then consult LLM-generated explanations, solution accuracy increases from 50% to over 67%, suggesting that premature exposure to

complete solutions may undermine deeper understanding and knowledge transfer [44]. At the same time, excessive reliance on AI support, as well as affective and attitudinal biases in LLMs toward mathematics, can reduce engagement in reasoning and weaken critical reflection [45–47]. In addition, LLM-based tutoring remains limited for tasks that require visual presentation (e.g., geometry), is susceptible to the generation of incorrect information, and is hindered by the lack of unified evaluation standards for assessing educational effectiveness [4,15]. Taken together, these studies highlight the limitations of ad hoc LLM deployment and motivate the need for system-level mechanisms that govern hint generation, reasoning progression, and the integration of external tools within conversational workflow [7,15,38,39,44].

2.3. Structured Reasoning Strategies and Tool-Mediated Support

Socratic, dialog-based instruction has long been used in mathematics education to emphasize reflection and problem exploration rather than immediate answer delivery [15]. Instead of presenting complete solutions, this approach relies on guided questioning to make learners' reasoning explicit, surface misconceptions, and prompt consideration of alternative strategies, which has been shown to support deeper mathematical understanding in both self-study and remedial settings [15].

A second major approach is closely associated with Pólya-inspired problem solving, a well-established instructional framework that structures learners' activity into phases such as problem comprehension, planning, execution, and reflection. Aligning support with these phases has been linked to more systematic mathematical thinking and improved performance [15,48]. Prior evidence also suggests that structured problem-solving guidance can yield more efficient learning outcomes than conventional instruction, including stronger conceptual understanding and better transfer to novel tasks [49]. Collectively, these findings motivate process-oriented tutoring designs that sustain computational thinking, mathematical reasoning, self-regulated learning, and longer-term learning effectiveness [15,49].

Beyond dialog-based scaffolding, visualization is often required to support mathematical sensemaking, especially in tasks where learners must coordinate algebraic and graphical representations [50]. Dynamic mathematics software such as GeoGebra enables learners to inspect formulas and graphs in parallel and to visualize intermediate steps, supporting step verification and the formation of general conclusions from concrete examples [51]. Empirical evidence further indicates that integrating GeoGebra into collaborative calculus learning can yield significantly higher achievement than comparable instruction without GeoGebra [50,51]. More broadly, tool-mediated visualization can strengthen learners' reasoning by helping them connect mathematical properties with evolving solution ideas. Interaction techniques such as dragging additionally support the exploration of invariant properties and the coordination of geometric and algebraic interpretations in ways that are difficult to achieve in purely text-based tutoring environments [51,52]. Accordingly, integrating optional tool-mediated visualization (where appropriate) as a complement to dialog-based guidance can substantially support conceptual understanding and the formation of effective solution strategies [41,53]. Viewed through a cognitive computing lens, these approaches point toward the value of explicit reasoning control policies and multimodal interaction components that extend beyond purely text-based model outputs [15,44,54,55]. This motivates our contribution, which integrates optional tool-mediated visualization to support tasks requiring visual reasoning.

3. Spark_AI High Level Architecture Overview

3.1. Application Structure and Persistence Model

The persistence model (Figure 1) is implemented in a Ruby on Rails application backed by a relational database. The model separates user ownership, configuration, content artifacts, and interaction traces. A User represents an authenticated account and stores basic profile metadata for personalization and reporting. A User owns one or more Project instances. A Project is the main organizational unit. Each project contains multiple Post objects and stores the configuration parameter `ai_prompt_type`. This parameter determines the tutoring module used for AI interactions associated with the project. A Post represents a tutoring session artifact and serves as the container for a stateful interaction. Each post belongs to a project, references a Snapshot, and stores configuration and execution metadata, including the selected tutoring mode (via the project's `ai_prompt_type`) and the `thread_id` used to resume stateful LLM conversations across requests. A Snapshot groups posts into higher-level collections (e.g., sessions or daily digests). Although only the reference is shown in the schema excerpt, the entity is designed to support additional provenance and evaluation metadata. A Note represents an individual conversational turn (message) exchanged between the learner and the AI assistant. Notes are associated with a specific post and collectively form the chat transcript for that tutoring session. Each note is linked to its author (User) and to its session container (Post), enabling persistent storage of multi-turn dialogs for later inspection, evaluation, and analysis.

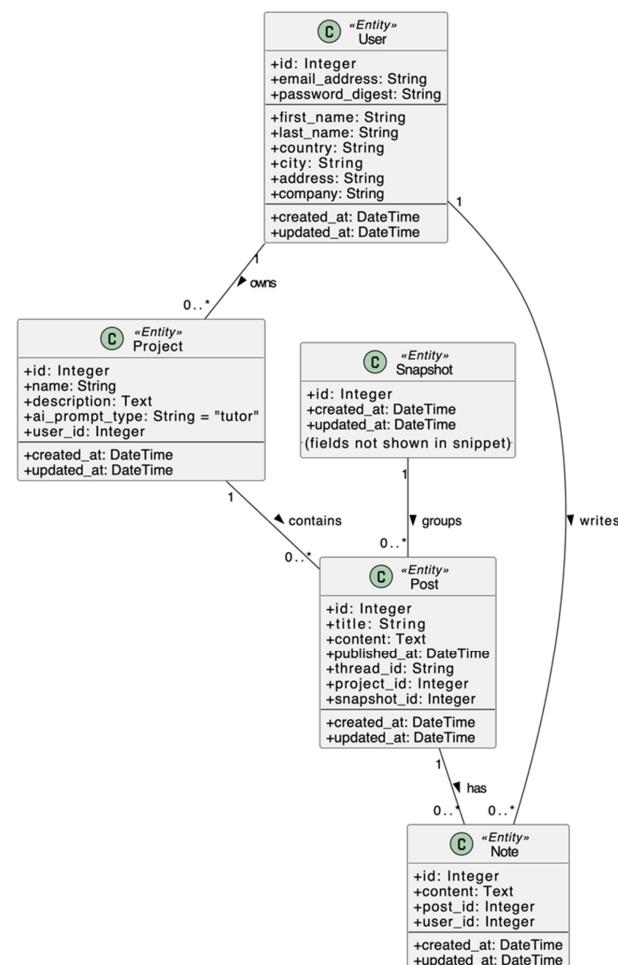


Figure 1. UML diagram of the SPARK_AI Data Model: User, Project, Snapshot, Post, Note.

3.1.1. Prompt Modularity and Extension

SPARK_AI is designed as a modular tutoring platform. Prompt modularity is implemented via the `ai_prompt_type` attribute in Project (Figure 1). The current implementation supports a baseline tutor mode and two specialized modules: SPARK_AI_MATH and SPARK_AI_PROGRAMMING. At runtime, `ai_prompt_type` is resolved into a prompt strategy that provides the system-level instruction template. This follows a strategy-style design: new tutoring modules can be introduced by adding new prompt types and corresponding prompt templates, without changing the orchestration logic. This approach supports controlled experimentation. For example, different pedagogical protocols (e.g., Socratic–Polya scaffolding vs. PRIMM-based programming guidance) can be switched at the project level, while the data model and service interface remain unchanged. At present, SPARK_AI_MATH is in active use, whereas SPARK_AI_PROGRAMMING is still in the testing phase. Accordingly, this paper provides a detailed presentation of SPARK_AI_MATH, while the programming module is left for future work.

3.1.2. AI Orchestration Service

The AI interaction is encapsulated in an application service, `AiContentProcessor` (Figure 2). The service is responsible for (i) building the model input, (ii) invoking the LLM backend, (iii) extracting the output text, and (iv) updating conversation state. The processor enforces a bounded input size via `MAX_CHARS` to ensure predictable runtime behavior. If the input exceeds the threshold, the current implementation truncates content, which can be replaced by chunking and hierarchical summarization in future work. Prompt injection and history import occur only during initialization. When no `thread_id` is available, the processor sends the selected system prompt and optionally imports prior conversation history from persistent storage. This bootstraps the LLM context in one request. Thereafter, continuation mode sends only the new user message while the provider preserves state, reducing prompt tokens and avoiding per-request history reconstruction.

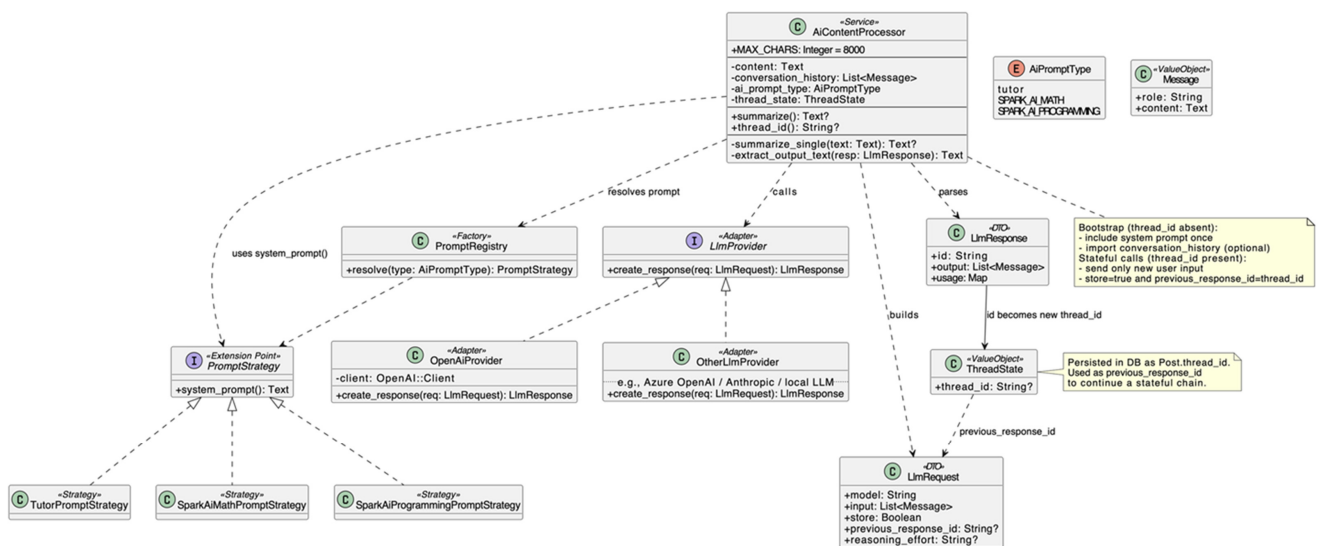


Figure 2. UML Representation of the SPARK_AI LLM Integration Architecture: Prompt Strategy Resolution and Stateful Thread Continuation.

3.1.3. Stateful LLM Interaction via Thread_id

SPARK_AI explicitly supports stateful LLM APIs. The system persists a conversation handle (`thread_id`) in Post and uses it to continue multi-turn interactions (Figure 2). The first request creates a stored conversation state at the provider. Subsequent requests include

previous_response_id = thread_id, enabling the provider to restore context server-side. After each call, the provider returns a new response identifier, which becomes the updated thread_id and persists for the next turn. This design provides two practical benefits. First, it reduces payload size and token overhead by avoiding repeated transmission of the full prompt and history. Second, it improves interaction consistency by relying on provider-managed state rather than application-side history reconstruction.

3.1.4. Platform Independence Through Provider Abstraction

Although the current implementation uses the OpenAI Ruby client, the architecture is designed to remain platform-independent. The LLM access is modeled through an LlmProvider abstraction (Figure 2), which defines a single operation for generating responses from a provider-agnostic request format. Concrete implementations (e.g., OpenAiProvider and alternative provider adapters) can be substituted without modifying the persistence model or the orchestration flow. This supports portability across providers and enables comparative evaluations under a fixed application protocol. In the current codebase, provider-specific details are localized to the service. A straightforward refactoring can further strengthen this separation by injecting the provider dependency into AiContentProcessor, rather than instantiating it internally.

3.1.5. Implementation and Operational Considerations

The system includes structured logging for request payloads, response identifiers, and usage metadata. This supports traceability of multi-turn conversations, monitoring of token usage, and experimental measurement of cost and latency. Errors in provider calls are handled defensively. Exceptions are logged, re-raised in development environments for debugging, and degraded to safe failures in production to prevent application-wide disruptions. Overall, the architecture separates persistence concerns from AI interaction concerns, high level system architecture is shown in Figure 3. This separation simplifies maintenance and facilitates experimental methodology, since artifacts, prompts, and state identifiers can be systematically linked and analyzed.

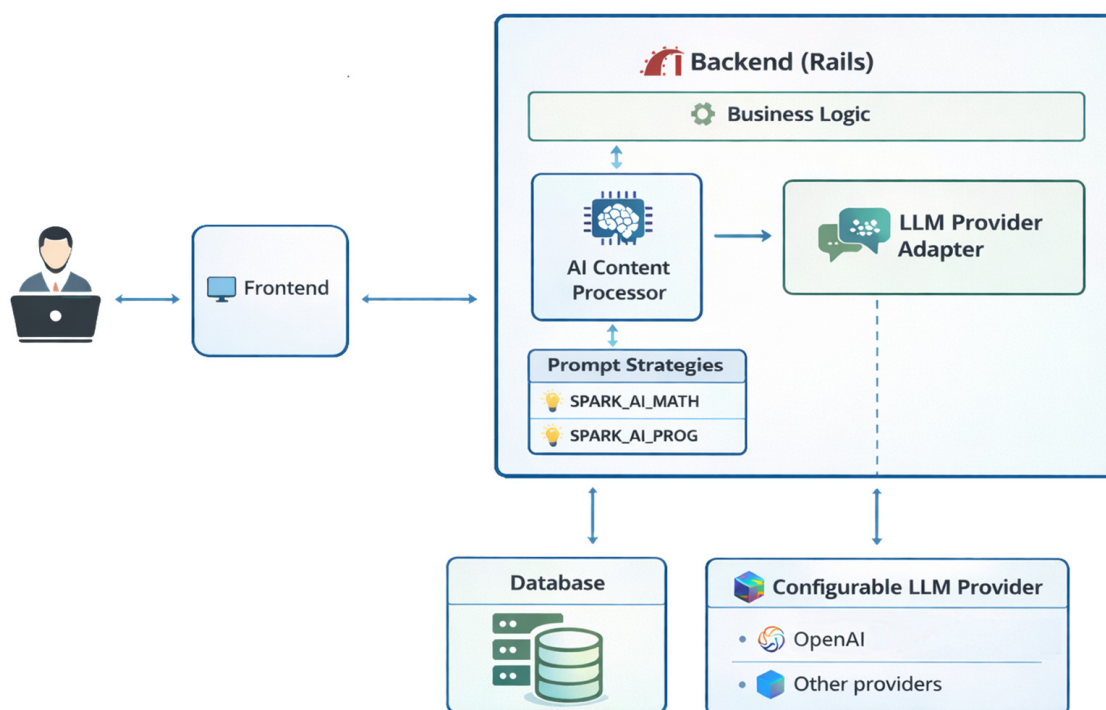


Figure 3. High level system architecture of the SPARK_AI framework.

3.2. Structure and Development of SPARK_AI_MATH

SPARK_AI_MATH is a domain module within the SPARK_AI framework designed to support secondary-school learners and early-year university students in solving non-routine mathematics problems (and can be used with younger learners when appropriate). The module targets scenarios in which learners are unable to make progress independently and would otherwise rely on general-purpose LLM chatbots (e.g., ChatGPT), which often optimize for rapid answer delivery with limited pedagogical structure. In contrast, SPARK_AI_MATH is designed to preserve process-oriented learning by guiding users through a structured problem-solving dialog rather than directly providing final solutions. A session begins when a learner creates a project, selects SPARK_AI_MATH, and submits a problem statement. The interaction then follows a Socratic style: the system asks targeted questions, elicits intermediate reasoning, and advances only after evaluating each learner response. The conversational tone is intentionally supportive to sustain engagement during extended problem-solving episodes (Figure 4).

Methodologically, SPARK_AI_MATH operationalizes Pólya-inspired problem solving through phases covering (i) problem comprehension, (ii) plan construction, (iii) plan execution, and (iv) reflection and verification (Figure 4). In the comprehension phase, the system probes understanding of requirements, given information, and notation. In the planning phase, it prompts the learner to articulate a viable solution strategy. During execution, it supports stepwise completion of the plan. When the task benefits from visualization (e.g., functions, graphs, or geometry-related reasoning), the system optionally routes the learner to GeoGebra Classic (Version 6.0.920.0) with task-specific input instructions and interpretation prompts, encouraging coordination between algebraic work and graphical evidence prior to finalizing an analytic solution. In the current implementation, support for GeoGebra was provided at the prompt level rather than through direct software integration. When visualization was pedagogically relevant, the system instructed students to use GeoGebra and provided explicit guidance on what to construct and what conclusions to draw from the resulting representation.

After completing the solution, the system guides the learner through a structured “look-back,” presents a consolidated solution trace that highlights points of divergence from the expected approach, invites clarification questions, and optionally proposes a structurally similar follow-up task.

To manage impasses while avoiding premature solution disclosure, SPARK_AI_MATH implements a controlled help policy (Figure 4). An initial design used three escalating hint levels (from high-level cues to partial worked steps), followed by a full solution when needed. A pilot study with 22 information-technology students indicated that multi-level hinting increased interaction overhead and time-on-task. Based on these observations, the help mechanism was simplified to a single partial-solution hint; if the learner still cannot proceed, the system provides the corresponding step with a brief explanation and continues to the next stage.

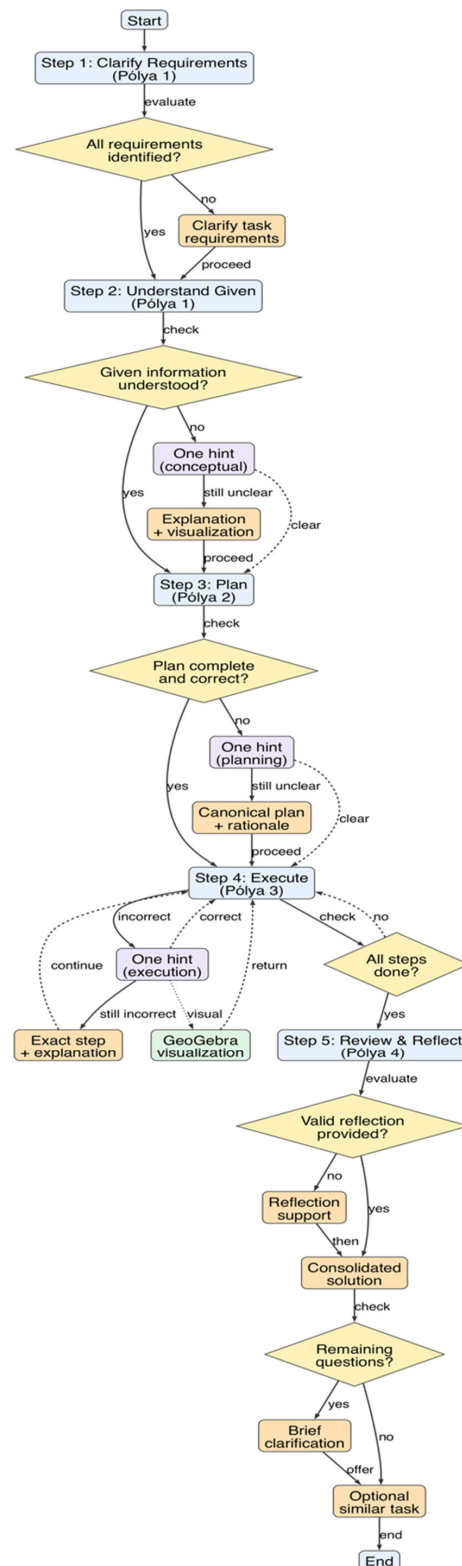


Figure 4. Structured Socratic–Pólya tutoring flow implemented in SPARK_AI_MATH.

Prompt-Specified Tutoring Policy in SPARK_AI_MATH

To ensure pedagogical controllability and reproducible tutoring trajectories, SPARK_AI_MATH encodes its tutoring protocol directly at the prompt level as an explicit policy specification (Table 1). The prompt defines the tutor role, target population, and a strict multi-step interaction contract aligned with Pólya-style problem solving. In our implementation, the protocol is expressed as five labeled steps which operationalize

and refine the four Pólya phases by separating problem interpretation from analysis of givens for improved error checking and clearer interaction logging. Concretely, the prompt enforces (i) strict step progression and step separation (each step appears in a new message with explicit labeling), (ii) response gating, where the tutor evaluates the learner's input before advancing, and (iii) concise outputs with bounded length to prevent "solution dumping." The assistance policy is also bounded: after an incorrect response, the system provides exactly one hint; if the learner still cannot proceed, it provides the required step with a brief explanation and continues. In addition, execution is constrained to one step per message, which prevents collapsing the entire solution into a single response (Figure 5).

Table 1. Prompt-specified pedagogical principles and resulting behaviors in SPARK_AI_MATH.

Pedagogical Principle	Prompt-Level Constraint/Rule	Resulting Behavior
Socratic guidance	Prefer questions over direct answers; elicit reasoning before advancing.	Learner articulates intermediate steps; solution is constructed via dialog.
Pólya-aligned protocol	Enforce labeled phases: interpret/analyze → plan → execute → reflect/verify.	Consistent, process-oriented tutoring trajectory across sessions.
Plan gating	Do not enter execution until an explicit, complete plan is stated.	Reduces step-skipping; promotes strategic planning before computation.
Stepwise pacing	One step per message; keep responses concise (length-bounded).	Limits "solution dumping"; lowers cognitive load during execution.
Bounded help policy	After an error: provide one hint; if unresolved, provide the needed step + brief rationale.	Balances autonomy and progress; minimizes premature full-solution exposure.
Tool-mediated visualization	During execution, perform a "GeoGebra check"; if relevant, provide concrete GeoGebra inputs and interpretation prompts.	Couples algebraic reasoning with interactive graphical evidence when needed.
Reflection and consolidation	Require a look-back; summarize solution path and highlight error points/pitfalls.	Supports verification and metacognitive reflection; improves transfer to similar tasks.
Supportive, non-misleading feedback	Encourage effort; never confirm incorrect answers.	Maintains engagement while preserving instructional correctness.

Finally, tool-mediated visualization is operationalized as a first-class mechanism through an explicit "GeoGebra check" during execution; whenever graphical reasoning is relevant, the prompt requires concrete GeoGebra Classic instructions (what to input and what to observe), thereby coupling dialog-based scaffolding with tool-mediated support (Figure 5). Formatting constraints (e.g., MathJax-only expressions) further improve the readability and consistency of generated solution traces. By specifying these constraints explicitly at the prompt level, SPARK_AI_MATH shifts part of the pedagogical design

from implicit model behavior to an auditable, reproducible tutoring policy, enabling consistent interaction patterns across sessions and facilitating fine-grained analysis of tutoring trajectories (Table 1).

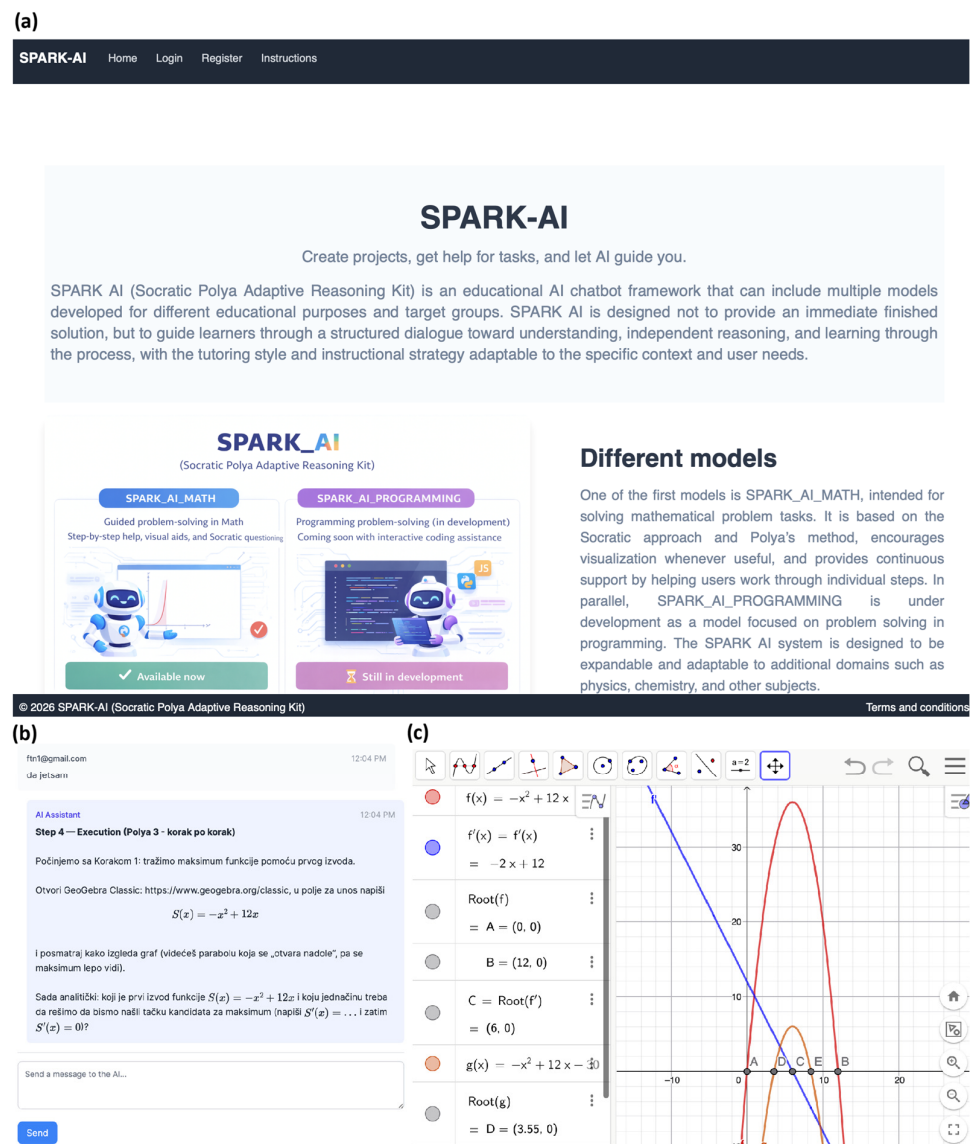


Figure 5. Overview of the SPARK_AI_MATH application. (a) The SPARK_AI home page depicting project-based interaction and module selection; (b) stepwise chatbot interaction following a Socratic-Polya guided problem-solving protocol; (c) Example of a user-generated graphical representation following the chatbot's guidance during task execution.

3.3. Preliminary Evaluation of SPARK_AI

For the evaluation, first-year mechanical engineering students used SPARK_AI during a regular class session to solve three real-world, problem-oriented mathematics tasks. The study protocol was approved by the faculty research ethics committee (No. MF 742/2). A total of 143 students participated, and informed consent was obtained from all participants. Students were first asked to attempt solving the tasks independently on paper; 108 of 143 students were unable to solve the tasks without support and were subsequently guided to use SPARK_AI, forming the main study sample for the chatbot-based phase.

After interacting with the chatbot, students completed an expanded Technology Acceptance Model (TAM) questionnaire [56–58], adapted to the SPARK_AI_MATH context. The current study examined five constructs: intrinsic motivation (IM), perceived ease of use

(PEOU), perceived usefulness (PU), behavioral intention (BI), and learning support (LS). Consistent with prior extended TAM research on LLM adoption, IM refers to the extent to which using the system is experienced as enjoyable, pleasant, and interesting; PEOU captures the degree to which the system is perceived as easy to use and understandable during interaction; PU reflects the extent to which the system is perceived as useful for supporting learning and problem solving; and BI refers to students' intention to continue using the system in the future [56]. IM was grounded in prior work on intrinsic motivation in technology use [56,59], PEOU and PU were based on the original TAM framework [60], and BI was aligned with prior acceptance research on continued intention to use educational technologies [56]. LS was added as a context-specific construct to capture the pedagogical support provided by SPARK_AI_MATH. In our study, IM, PEOU, PU, BI were adapted to the context of SPARK_AI_MATH, and an additional construct, LS, was introduced to capture the perceived pedagogical value of the system, particularly in terms of step-by-step guidance, support for understanding, and contribution to the learning process. Together, these constructs enabled us to assess both general technology acceptance and the domain-specific educational value of SPARK_AI_MATH.

Finally, the questionnaire included three open-ended prompts addressing (i) potential improvements to SPARK_AI_MATH, (ii) the system's main perceived advantage, and (iii) features students expect from an "ideal" chatbot for learning and solving mathematical content.

Data Analysis

Data analysis was conducted in three stages. First, descriptive statistics were calculated for all quantitative items to summarize central tendency and variability, using Python (version 3.11).

PLS-SEM was applied using SmartPLS (version 4.1.1.6). The measurement model was evaluated by examining indicator loadings, internal consistency reliability (Cronbach's alpha and composite reliability), convergent validity (average variance extracted), discriminant validity (Fornell–Larcker criterion), and collinearity (variance inflation factors). The structural model was assessed using path coefficients, coefficients of determination (R^2), effect sizes (f^2), and nonparametric bootstrapping with 5000 resamples to test the significance of structural relationships.

Finally, responses to open-ended questions were analyzed qualitatively using an inductive categorization procedure. Two researchers independently read all student responses and classified them into response categories based on their content. New categories were created when a response expressed an idea not represented in the existing coding scheme, whereas conceptually similar responses were assigned to already established categories. Disagreements between the two researchers occurred in fewer than 5% of responses and were resolved through discussion and consensus, resulting in the final set of categories used in the analysis. The categorized responses were then summarized by reporting the most frequent categories and illustrative translated comments.

4. Results

4.1. Descriptive Statistics

Descriptive statistics for the 19 quantitative items indicated generally positive evaluations of SPARK_AI. Item means ranged from 3.38 (PU2) to 4.52 (MA3), with most modes at 4 or 5, suggesting responses clustered toward agreement (Appendix A, Table A1). Standard deviations were moderate overall, with the largest dispersion observed for PU2 ($SD = 1.07$) and the smallest for MA3 ($SD = 0.65$). At the construct level, mean scores were consistently high, spanning from PU ($M = 3.88$, $SD = 0.66$) and BI ($M = 3.93$, $SD = 0.81$) to LS

($M = 4.36$, $SD = 0.59$), with IM ($M = 4.01$, $SD = 0.61$) and PEOU ($M = 4.04$, $SD = 0.65$) also above 4. Internal consistency was acceptable to good across constructs, with Cronbach's alpha ranging from 0.758 (LS) to 0.886 (BI) (Table 2) [61].

Table 2. Questionnaire construct scores and internal consistency.

Construct	Items	Mean	SD	Cronbach's α
Intrinsic Motivation (IM)	4	4.009	0.606	0.819
Perceived Ease of Use (PEOU)	4	4.037	0.646	0.821
Perceived Usefulness (PU)	4	3.884	0.662	0.804
Behavioral Intention (BI)	4	3.926	0.809	0.886
Learning Support (SPARK-specific)	3	4.359	0.593	0.758

4.2. Measurement and Structural Model Assessment

4.2.1. Assessment of the Measurement Model

Indicator reliability was assessed using outer loadings, and all indicators exceeded the 0.70 threshold, indicating that the items adequately represent their underlying latent constructs [61] (Table 3). Internal consistency reliability was supported by Cronbach's alpha and composite reliability (CR), with all values above 0.70 [62] (Table 4). Convergent validity was established because all constructs achieved average variance extracted (AVE) values greater than 0.50 [61] (Table 4). Discriminant validity was examined using the Fornell and Larcker criterion [61] and was satisfied, as the square roots of average variance extracted exceeded the corresponding inter construct correlations (Table 4).

Table 3. Measurement model assessment (factor loadings, AVE, CR, and Cronbach's alpha).

Construct	Item	Outer Loadings	AVE	CR	Cronbach α
Intrinsics Motivation	im1	0.857	0.648	0.880	0.819
	im2	0.812			
	im3	0.750			
	im4	0.798			
Perceived Ease of Use	peou1	0.778	0.649	0.883	0.821
	peou2	0.854			
	peou3	0.793			
	peou4	0.796			
Perceived usefulness	pu1	0.813	0.630	0.872	0.804
	pu2	0.786			
	pu3	0.810			
	pu4	0.765			
Behavioral intention	bi1	0.898	0.745	0.921	0.886
	bi2	0.894			
	bi3	0.791			
	bi4	0.867			
Learning support	ls1	0.829	0.676	0.862	0.758
	ls2	0.868			
	ls3	0.766			

Table 4. Discriminant validity (Fornell–Larcker criterion).

Construct	BI	IM	LS	PEOU	PU
BI	0.863				
IM	0.541	0.805			
LS	0.622	0.707	0.822		
PEOU	0.327	0.573	0.640	0.806	
PU	0.637	0.747	0.689	0.532	0.794

Note: Diagonals indicate the square root of AVE, whereas other entries indicate correlations.

4.2.2. Structural Model Assessment

Before testing the hypothesized relationships, collinearity among predictor constructs was assessed using inner variance inflation factor (VIF) values (ranged from 1.00 to 2.76). All VIF values were below common critical thresholds (3.3) [63], indicating that collinearity is unlikely to bias the estimation of the structural paths. The SRMR value was 0.088 for the saturated model and 0.108 for the estimated model, which indicates an acceptable approximation of model fit in PLS-SEM; therefore, interpretation focuses primarily on the model's explanatory results and bootstrapped significance testing.

The model's explanatory power was evaluated using coefficients of determination. The model explained 49.1% of the variance in BI ($R^2 = 0.491$), 47.5% in LS ($R^2 = 0.475$), 57.5% in PU ($R^2 = 0.575$), and 32.8% in PEOU ($R^2 = 0.328$), suggesting moderate to substantial explained variance across endogenous constructs (Figure 6).

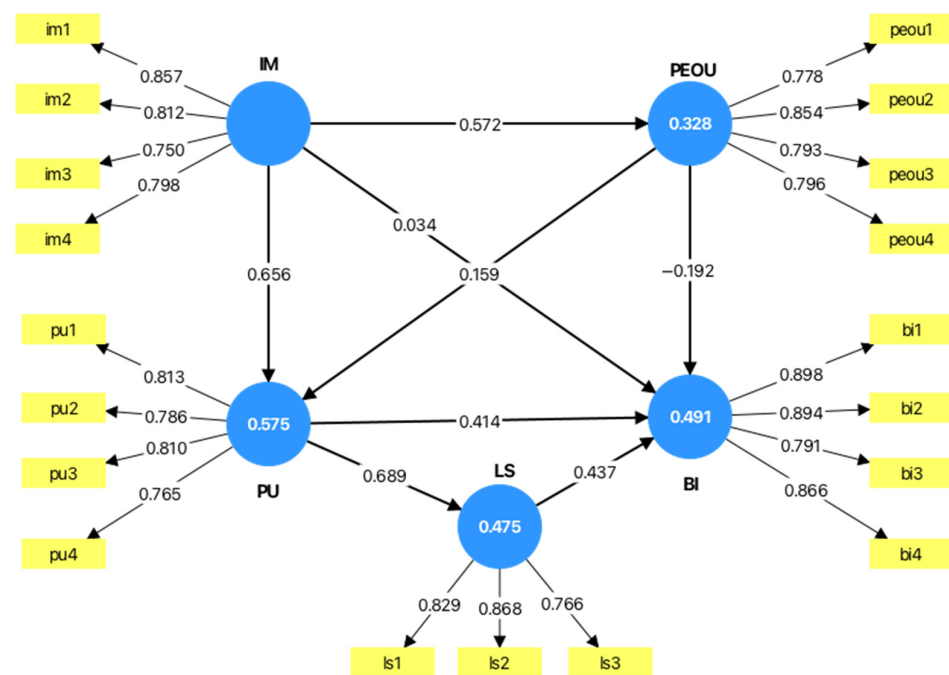


Figure 6. PLS-SEM Results for the Expanded TAM in the SPARK_AI_MATH Evaluation.

The hypotheses tested in this study were formulated with the aim of providing a more detailed explanation of students' attitudes toward the system. They were based on the Technology Acceptance Model (TAM) and its motivational extensions, while also being informed by prior research on the roles of perceived usefulness, perceived ease of use, intrinsic motivation, and behavioral intention in educational technology adoption [56,64–66]. In addition, the Learning Support construct was introduced as a context-specific pedagogical extension of this framework, reflecting the extent to which students perceive the system as supporting their learning process. Based on this theoretical background, the following hypotheses were formulated:

- H1.** *IM positively affects BI.*
- H2.** *IM positively affects PEOU.*
- H3.** *IM positively affects PU.*
- H4.** *LS positively affects BI.*

H5. *PEOU positively affects BI.*

H6. *PEOU positively affects PU.*

H7. *PU positively affects BI.*

H8. *PU positively affects LS.*

Effect sizes (f^2) indicated that $PU \rightarrow LS$ had a very large effect ($f^2 = 0.903$), and $IM \rightarrow PU$ ($f^2 = 0.681$) and $IM \rightarrow PEOU$ ($f^2 = 0.488$) also showed large effects. The effects of $PU \rightarrow BI$ ($f^2 = 0.130$) and $LS \rightarrow BI$ ($f^2 = 0.143$) were small-to-moderate, whereas $PEOU \rightarrow PU$ ($f^2 = 0.040$) and $PEOU \rightarrow BI$ ($f^2 = 0.041$) were small. The direct effect $IM \rightarrow BI$ was negligible ($f^2 = 0.001$). Model fit was additionally assessed using SRMR as an approximate fit indicator.

Bootstrapping results supported most of the hypothesized direct effects (Figure 6, Table 5). Specifically, $IM \rightarrow PEOU$ ($\beta = 0.572, p < 0.001$), $IM \rightarrow PU$ ($\beta = 0.656, p < 0.001$), $PEOU \rightarrow PU$ ($\beta = 0.159, p = 0.045$), $PU \rightarrow LS$ ($\beta = 0.689, p < 0.001$), $LS \rightarrow BI$ ($\beta = 0.437, p < 0.001$), and $PU \rightarrow BI$ ($\beta = 0.414, p = 0.001$) were significant. In contrast, the direct path $IM \rightarrow BI$ was not significant ($\beta = 0.034, p = 0.781$), and $PEOU \rightarrow BI$ was not significant at the 0.05 level ($\beta = -0.192, p = 0.056$) (Figure 6, Table 5). Finally, the mediation pattern indicates that the influence of IM on BI is primarily indirect. The total indirect effect $IM \rightarrow BI$ was significant ($0.424, p < 0.001$), while the direct effect was not, consistent with (near) full mediation through downstream constructs (notably via PU and LS).

Table 5. Direct effects in the structural model (bootstrapping).

Hypotheses	Paths	Path Coefficients	T-Values	p-Values	Results
H1	$IM \rightarrow BI$	0.034	0.278	0.781	Not Supported
H2	$IM \rightarrow PEOU$	0.572	8.375	0.000	Supported
H3	$IM \rightarrow PU$	0.656	9.508	0.000	Supported
H4	$LS \rightarrow BI$	0.437	3.778	0.000	Supported
H5	$PEOU \rightarrow BI$	-0.192	1.909	0.056	Not Supported
H6	$PEOU \rightarrow PU$	0.159	2.006	0.045	Supported
H7	$PU \rightarrow BI$	0.414	3.227	0.001	Supported
H8	$PU \rightarrow LS$	0.689	13.346	0.000	Supported

4.3. Qualitative Findings

The qualitative analysis of the open-ended responses revealed several recurring themes across all prompts. Regarding potential improvements (Table 6), participants most frequently emphasized the need for more flexible input and output options, such as mathematical input tools and code editors, as well as faster response times. When asked about the primary advantage of SPARK_AI_MATH (Table 6), respondents predominantly highlighted the clarity and depth of explanations and the overall learning support provided by the system. Step-by-step guidance and explicit problem decomposition were also frequently mentioned, while visualization was identified as a valuable feature in a smaller number of responses. Finally, descriptions of an ideal mathematics chatbot largely mirrored the existing design of SPARK_AI_MATH (Table 6), particularly its emphasis on detailed, step-by-step explanations, while additionally underscoring the importance of rapid feedback, enhanced input modalities, and adaptability to different learning contexts.

Table 6. Summary of themes from open-ended feedback (translated).

Open-Ended Prompt	Top Themes (Count)	Example Comment (Translated)
How could SPARK_AI_MATH be improved? (<i>n</i> = 92)	More flexible input and output options (45); Faster response time and greater efficiency (24); Improved communication and interaction quality (13); Alternative problem-solving approach (10)	"It could be significantly faster." "It would be helpful to add a built-in code editor and mathematical input to make entering problems easier."
What is the greatest advantage of SPARK_AI_MATH? (<i>n</i> = 86)	Clear, detailed explanations and learning support (46); Step-by-step guidance and problem decomposition (34); Visualizations (6)	"It checks our way of thinking and encourages us to solve the problem independently. It provides detailed explanations when I do not know something."/"I like that it solves step by step and breaks the task into key steps."
What features should an ideal math chatbot include? (<i>n</i> = 88)	High speed and better input options such as photo upload and mathematical entry (37); SPARK_AI features such as detailed, step-by-step solutions (35); Alternative methodological approaches, for example, integration into MOOCs and different solution strategies (16)	"It should allow me to take a photo of the task and send it, so I do not waste time typing."/"Clear and detailed step-by-step explanations."/"An ideal chatbot should be available for every course and adapt to my learning style."

5. Discussion

The descriptive analysis provides evidence that SPARK_AI_MATH was positively perceived by first-year mechanical engineering students when used for mathematical problem solving. Mean scores across all TAM constructs were consistently high, with particularly strong ratings for the SPARK-specific learning support items (Table 2). These results suggest that students valued the system's instructional scaffolding, including visualization support, perceived improvement in understanding upon task completion, and structured step-by-step guidance in contrast to receiving an immediate solution. Among the TAM constructs, PEOU also emerged as one of the most strongly rated dimensions, indicating that students were able to adapt quickly to the chatbot-based tutoring interaction without experiencing notable usability barriers. Overall, all questionnaire items exhibited relatively positive evaluations, with mode values predominantly at 4 or 5 on the Likert scale (Appendix A, Table A1). This distribution indicates that the majority of participants expressed favorable attitudes toward the system not only in terms of learning support and perceived ease of use, but also with respect to intrinsic motivation, perceived usefulness, and behavioral intention to use such tool in the future. From a system design perspective, these descriptive patterns suggest that users positively respond to interaction paradigms that explicitly structure reasoning processes and make intermediate steps visible, rather than treating the language model as a black-box answer generator.

The structural model results indicate that intrinsic motivation functions as a primary upstream driver of technology-related perceptions. Intrinsic motivation exhibits strong positive effects on both perceived ease of use and perceived usefulness, suggesting that more intrinsically motivated participants are more likely to perceive the system as easier to use and, more importantly, as useful. This pattern is consistent with findings from prior studies examining students' adoption of generative AI tools, including research on the use of ChatGPT in higher education contexts [56]. This finding indicates that motivational predispositions shape how users engage with the system's reasoning workflow, reinforcing the role of adaptive interaction design in supporting positive usability and usefulness perceptions.

In turn, perceived ease of use contributes more modestly to perceived usefulness, supporting the notion that usability primarily facilitates the formation of usefulness beliefs rather than directly driving adoption. Similar relationships have been reported in studies investigating student acceptance of educational platforms and digital learning technologies [67,68].

Perceived usefulness emerges as a central mechanism translating these perceptions into learning-related outcomes and usage intentions. Perceived usefulness strongly predicts learning support and exerts a direct effect on behavioral intention, while learning support further contributes to behavioral intention. This pattern is consistent with findings reported in prior studies on the adoption of AI-based and educational technologies [56,67,68]. Taken together, these results suggest that users' intentions to continue using the system are driven primarily by perceived instructional value and the learning support provided by the system, rather than by usability considerations alone. In the context of SPARK_AI_MATH, perceived usefulness appears to be closely tied to the system's ability to regulate reasoning flow and scaffold intermediate steps, rather than to surface-level usability features.

Notably, the direct effect of perceived ease of use on behavioral intention was not statistically significant and was slightly negative, a finding that is consistent with prior research on student adoption of generative AI tools [56]. This pattern suggests that ease of use alone may be insufficient to sustain engagement in reasoning-centered systems, underscoring the importance of explicitly communicating and delivering cognitive value through structured interaction. In addition, the direct path from intrinsic motivation to behavioral intention was also non-significant, diverging from the results reported by Lai (2023) [56]. However, intrinsic motivation exhibited a significant overall effect on behavioral intention through indirect pathways, indicating that motivation influences intention primarily via perceived usefulness and learning support. The absence of a direct relationship should not be interpreted as evidence that the link between intrinsic motivation and behavioral intention is unimportant. Rather, it may suggest that, in this context, intrinsic motivation toward the tool was expressed less as a direct driver of intention and more through students' beliefs about the system's usefulness and its capacity to support learning. One possible interpretation is that first-year mechanical engineering students approached SPARK_AI_MATH primarily as a practical learning support tool; accordingly, their intention to continue using it depended more strongly on its perceived instructional value than on whether the system itself was intrinsically motivating to use. At the same time, this interpretation remains tentative and should be treated with caution, as the present quantitative design does not permit firm conclusions about the underlying reasons for this pattern.

Qualitative feedback complemented the quantitative results by clarifying how students experienced SPARK_AI_MATH. Participants highlighted clear explanations, step-by-step guidance, and, in some cases, visualization as key strengths, which is consistent with prior research showing that students value educational chatbots and intelligent tutoring systems that provide explanatory feedback, structured guidance, and learning support [69–71]. In particular, the emphasis on step-by-step problem decomposition aligns with the broader literature on intelligent tutoring systems, in which guided support is regarded as a central mechanism for promoting understanding and problem-solving [70]. At the same time, students reported practical limitations affecting interaction efficiency, including response latency, limited multimodal input, and the inability to edit previous turns. Such issues are highly relevant for the overall user experience and align with prior findings emphasizing the importance of response time in conversational systems [72] and the value of flexible input modalities, including handwriting-based mathematical input, in tutoring environments [73]. Overall, the findings reinforce that the system's primary value lies in its reasoning control mechanisms

rather than raw answer generation, and they motivate concrete requirements: latency aware response handling and multimodal input support for diagram or handwriting-based work.

From a system design perspective, these findings highlight the importance of explicit reasoning control, guided problem solving, and sustained engagement in LLM-based conversational systems. The strong perceptions of learning support align with the design choice to enforce stepwise reasoning and integrate visualization into the tutoring workflow. However, these results are preliminary and reflect short term, first use perceptions. Future work should complement acceptance evidence with objective task and learning measures, such as solution accuracy, quality of intermediate problem-solving steps, time to solution, and retention or transfer. Larger samples would also enable more reliable PLS SEM analyses to examine indirect effects, group differences, and changes over sustained use. Overall, perceived value appears to depend more on interaction architecture that supports structured and transparent reasoning than on generative fluency alone.

In comparison with other similar systems, SPARK_AI can be situated in relation to several recent LLM-based tutoring architectures. Similar to Khanmigo, it is intended to support learning through guided interaction rather than direct answer delivery; however, whereas Khanmigo is embedded in Khan Academy's broader multi-subject learning ecosystem and is presented as an AI tutor that prompts students to think critically without simply providing answers, SPARK_AI places stronger emphasis on explicit control of reasoning progression in mathematics-focused dialog (Khan Academy, 2023) [74]. In contrast to systems embedded within predefined content structures, SPARK_AI supports more flexible, user-initiated problem exploration. SPARK_AI also shares important pedagogical ground with TALPer, which combines Socratic dialog and Pólya's problem-solving strategy within the Taiwan Adaptive Learning Platform, but differs in targeting older learners and in framing support through a more explicit reasoning-control architecture with a distinct hint policy and optional tool-mediated visualization support [15]. A further point of comparison is SocraticLM, which advances a Socratic, thought-provoking teaching paradigm through multi-round tutoring dialogs; compared with such approaches, SPARK_AI is less centered on model-level teaching-style generation and more on reproducible interaction design that regulates how support unfolds across phases of problem solving [75]. Finally, unlike recent LLM tutoring architectures that emphasize personalization, learned pedagogical response generation, or model-level optimization of student outcomes, SPARK_AI foregrounds reproducible reasoning control at the interaction-design level through pedagogically constrained orchestration rather than personalization or model fine-tuning alone [76]. This distinction is particularly important in mathematical problem solving, where the structure and progression of reasoning play an important role in learning. From this perspective, the main contribution of SPARK_AI lies not in claiming general superiority over other AI tutors, but in offering a transparent and controllable architecture for process-oriented mathematical reasoning support.

6. Conclusions

This work presents SPARK_AI as an extensible, modular, and stateful system architecture for process-oriented reasoning across domains, using explicit mechanisms to control LLM interaction behavior. Within this architecture, SPARK_AI_MATH instantiates a structured interaction protocol that mitigates the direct-answer tendency of general-purpose LLMs. The system enforces a predefined conversational workflow comprising problem interpretation, analysis of givens, planning, execution, and reflection, supported by a bounded hint strategy and optional visualization. An in-class deployment with first-year mechanical engineering students solving mathematical tasks indicated strong perceptions

of learning support, corroborated by qualitative feedback on explanation clarity and step-by-step guidance.

From a modeling perspective, the PLS-SEM analysis accounted for substantial variance in perceived usefulness, learning support, and behavioral intention. Intrinsic motivation exerted strong direct effects on perceived usefulness and perceived ease of use, as well as indirect effects on behavioral intention mediated through perceived usefulness and learning support. Perceived usefulness and learning support further functioned as key downstream predictors of behavioral intention. These findings suggest that, in reasoning-centric conversational systems, design choices that make cognitive value and structured reasoning support explicit play a more decisive role in sustained use than interface usability alone.

Future work will extend the evaluation to larger and more diverse samples, refine the ease-of-use indicators, and incorporate objective, performance-based measures of problem-solving quality and learning outcomes. This is particularly important, as objective outcome indicators would make it possible to evaluate the system's educational impact more directly than perception-based measures alone. In addition, user feedback collected during system evaluation will be systematically used to guide iterative improvements in interaction efficiency and input flexibility. Future research should also examine the relationship between intrinsic motivation and behavioral intention in greater depth, as the present findings suggest that this link warrants further investigation. Together, these extensions will enable a more rigorous assessment of the system's impact and support the generalization of the proposed cognitive computing and reasoning-control principles to other LLM-based conversational systems.

Author Contributions: Conceptualization, M.K., A.S. and S.K.; methodology, M.K.; software, S.K.; validation, M.K., L.I. and A.S.; formal analysis, M.K.; investigation, M.K.; resources, M.V. and N.V.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, L.I., A.M. and N.V.; visualization, S.K. and M.V.; supervision, M.K.; project administration, M.K., M.V. and N.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, and approved by the Ethics Committee of Mechanical faculty, University of Belgrade (MF 742/2; 7 May 2024).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The anonymized dataset is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

LLM	Large Language Model
SPARK_AI	Socratic–Polya Adaptive Reasoning Kit Artificial Intelligence
SPARK_AI_MATH	Socratic–Polya Adaptive Reasoning Kit Artificial Intelligence for Mathematics
SPARK_AI_PROGRAMMING	Socratic–Polya Adaptive Reasoning Kit Artificial Intelligence for Programming
TALP	Taiwan Adaptive Learning Platform
TAM	Technology Acceptance Model
IM	Intrinsic Motivation

PEOU	Perceived Ease of Use
PU	Perceived Usefulness
BI	Behavioral Intention
LS	Learning Support
PLS-SEM	Partial Least Squares Structural Equation Modeling

Appendix A

Table A1. Item-level descriptive statistics for the extended TAM questionnaire adapted to SPARK_AI_MATH ($N = 108$). Scale: 5-point Likert (1 = Strongly disagree, 5 = Strongly agree).

Construct	Item	Statement	N	Mean	SD	Mode
IM	IM1	I find using Spark_AI_Math enjoyable.	108	4.185	0.725	4
	IM2	The actual process of using Spark_AI_Math was pleasant.	106	4.075	0.752	4
	IM3	I had fun using Spark_AI.	107	3.785	0.753	4
	IM4	Using Spark_AI for solving tasks is interesting.	108	3.991	0.791	4
PEOU	PEOU1	Learning how to use Spark_AI_Math is easy for me.	106	4.434	0.717	5
	PEOU2	I find SPARK_AI_Math easy to use for problem-solving tasks.	107	3.879	0.843	4
	PEOU3	I find it easy for me to become skillful at asking Spark_AI_Math for solving tasks.	108	3.861	0.836	4
	PEOU4	My interaction with Spark_AI_Math is clear and understandable when solving tasks.	108	3.981	0.796	4
PU	PU1	I find Spark_AI_Math useful for solving tasks.	108	4.028	0.836	4
	PU2	Using Spark_AI_Math helps me to solve task more quickly.	108	3.380	1.065	4
	PU3	Using Spark_AI_Math for solving tasks would increase my academic performance.	108	4.028	0.689	4
	PU4	Using Spark_AI_Math for solving tasks would enhance my effectiveness of learning.	108	4.102	0.735	4
BI	BI1	Over the next few weeks, I intend to use Spark_AI_Math in solving tasks.	108	3.896	0.945	4
	BI2	Over the next few weeks, I plan to use Spark_AI_Math for solving tasks.	108	3.926	0.903	4
	BI3	Over the next few weeks, I predict I will use Spark_AI_Math for solving tasks.	108	4.213	0.897	5
	BI4	I plan to continue to use Spark_AI_Math for solving tasks.	108	3.796	0.944	4
LS	LS1	The visualization provided by Spark_AI_Math helped me solve the tasks.	108	4.296	0.739	5
	LS2	I now understand the task I was solving.	108	4.259	0.766	4
	LS3	I find it useful to solve a task step by step, rather than receiving an instant, ready-made solution.	107	4.523	0.649	5

References

1. Hu, K. ‘ChatGPT Sets Record for Fastest-Growing User Base—Analyst Note,’ Reuters. 1 February 2023. Available online: <https://www.reuters.com/technology/chatgpt-sets-record-fastest-growing-user-base-analyst-note-2023-02-01/> (accessed on 28 August 2025).
2. Wölfel, M.; Shirzad, M.B.; Reich, A.; Anderer, K. Knowledge-Based and Generative-AI-Driven Pedagogical Conversational Agents: A Comparative Study of Grice’s Cooperative Principles and Trust. *Big Data Cogn. Comput.* **2024**, *8*, 2. [CrossRef]
3. Stamatakis, A.; Logothetis, I.; Chatzea, V.E.; Papadakis, A.; Vidakis, N. Implementing Educational Innovation in LMSs: Hackathons, Microcredentials, and Blended Learning. *Appl. Syst. Innov.* **2025**, *8*, 175. [CrossRef]

4. Zhang, D.W.; Boey, M.; Tan, Y.Y.; Jia, A.H.S. Evaluating large language models for criterion-based grading from agreement to consistency. *npj Sci. Learn.* **2024**, *9*, 79. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Deng, R.; Jiang, M.; Yu, X.; Lu, Y.; Liu, S. Does ChatGPT enhance student learning? A systematic review and meta-analysis of experimental studies. *Comput. Educ.* **2025**, *227*, 105224. [\[CrossRef\]](#)
6. Shi, Y.; Yu, K.; Dong, Y.; Chen, F. Large language models in education: A systematic review of empirical applications, benefits, and challenges. *Comput. Educ. Artif. Intell.* **2026**, *10*, 100529. [\[CrossRef\]](#)
7. Zhang, P.; Tur, G. A systematic review of ChatGPT use in K-12 education. *Eur. J. Educ.* **2024**, *59*, e12599. [\[CrossRef\]](#)
8. Wu, T.; Terry, M.; Cai, C.J. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. In *Conference on Human Factors in Computing Systems—Proceedings*; Association for Computing Machinery: New York, NY, USA, 2022. [\[CrossRef\]](#)
9. Chen, B.; Zhang, Z.; Langrené, N.; Zhu, S. Unleashing the potential of prompt engineering for large language models. *Patterns* **2025**, *6*, 101260. [\[CrossRef\]](#)
10. Dwivedi, N.K.; Kshetri, N.; Hughes, L.; Slade, E.L.; Jeyaraj, A.; Kar, A.K.; Baabdullah, A.M.; Koohang, A.; Raghavan, V.; Ahuja, M.; et al. Opinion paper: ‘so what if chatgpt wrote it?’ multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy. *Int. J. Inf. Manag.* **2023**, *71*, 102642. [\[CrossRef\]](#)
11. Asperti, A.; Naibo, A.; Coen, C.S. Thinking Machines: Mathematical Reasoning in the Age of LLMs. *Big Data Cogn. Comput.* **2026**, *10*, 38. [\[CrossRef\]](#)
12. İpek, Z.H.; Gözü, A.İ.C.; Papadakis, S.; Kalogiannakis, M. Educational Applications of the ChatGPT AI System: A Systematic Review Research. *Educ. Process Int. J.* **2023**, *12*, 26–55. [\[CrossRef\]](#)
13. Mustafa, M.Y.; Tlili, A.; Lampropoulos, G.; Huang, R.; Jandrić, P.; Zhao, J.; Salha, S.; Xu, L.; Panda, S.; Kinshuk; et al. A systematic review of literature reviews on artificial intelligence in education (AIED): A roadmap to a future research agenda. *Smart Learn. Environ.* **2024**, *11*, 59. [\[CrossRef\]](#)
14. Faith, L.; Zaugg, T.; Stolys, N.; Szabo, M.; Haghi, F.; Badlis, C.; Olmedo, S.L. Persona, Break Glass, Name Plan, Jam (PBNJ): A New AI Workflow for Planning and Problem Solving. *AI* **2025**, *6*, 310. [\[CrossRef\]](#)
15. Kuo, B.C.; Bai, Z.E.; Lin, C.H. Developing an AI learning companion for mathematics problem solving in elementary schools. *Comput. Educ.* **2026**, *240*, 105463. [\[CrossRef\]](#)
16. Ramanathan, H.; Palaniappan, R. Comparison of three large language models as middle school math tutoring assistants. *J. Emerg. Investig.* **2024**, *7*, 1–6. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Plevris, V.; Papazafeiropoulos, G.; Rios, A.J. Chatbots Put to the Test in Math and Logic Problems: A Comparison and Assessment of ChatGPT-3.5, ChatGPT-4, and Google Bard. *AI* **2023**, *4*, 949–969. [\[CrossRef\]](#)
18. Lo, C.K.; Hew, K.F.; Jong, M.S.Y. The influence of ChatGPT on student engagement: A systematic review and future research agenda. *Comput. Educ.* **2024**, *219*, 105100. [\[CrossRef\]](#)
19. Vargas-Murillo, A.R.; de la Asuncion, I.N.M.; de Jesús Guevara-Soto, F. Challenges and opportunities of AI-assisted learning: A systematic literature review on the impact of ChatGPT usage in higher education. *Int. J. Learn. Teach. Educ. Res.* **2023**, *22*, 122–135. [\[CrossRef\]](#)
20. Sánchez-Ruiz, L.M.; Moll-López, S.; Nuñez-Pérez, A.; Moraño-Fernández, J.A.; Vega-Fleitas, E. ChatGPT Challenges Blended Learning Methodologies in Engineering Education: A Case Study in Mathematics. *Appl. Sci.* **2023**, *13*, 6039. [\[CrossRef\]](#)
21. Fan, Y.; Tang, L.; Le, H.; Shen, K.; Tan, S.; Zhao, Y.; Shen, Y.; Li, X.; Gašević, D. Beware of metacognitive laziness: Effects of generative artificial intelligence on learning motivation, processes, and performance. *Br. J. Educ. Technol.* **2025**, *56*, 489–530. [\[CrossRef\]](#)
22. Suriano, R.; Plebe, A.; Acciai, A.; Fabio, R.A. Student interaction with ChatGPT can promote complex critical thinking skills. *Learn. Instr.* **2025**, *95*, 102011. [\[CrossRef\]](#)
23. Krupp, L.; Steinert, S.; Kiefer-Emmanouilidis, M.; Avila, K.E.; Lukowicz, P.; Kuhn, J.; Küchemann, S.; Karolus, J. Challenges and Opportunities of Moderating Usage of Large Language Models in Education. *arXiv* **2023**. [\[CrossRef\]](#)
24. Chan, C.K.Y.; Hu, W. Students’ voices on generative AI: Perceptions, benefits, and challenges in higher education. *Int. J. Educ. Technol. High. Educ.* **2023**, *20*, 43. [\[CrossRef\]](#)
25. Lee, P.-L.; Hung, S.-T.; Chang, P.-H.; Chang, C.-Y.; Bao, L.; Yeh, T.-K.; Lee, L.-C. Exploring Problem-Solving Strategies in Gifted and Regular Students: Education Insights from Eye-Tracking Analysis. *Appl. Syst. Innov.* **2026**, *9*, 38. [\[CrossRef\]](#)
26. Athar, M.E. The constructive, overreliant, and irresponsible use of artificial intelligence tools in academia: Personality correlates and implications for academic integrity. *Comput. Hum. Behav. Rep.* **2025**, *18*, 100679. [\[CrossRef\]](#)
27. Kwon, K.; Yang, S.; Kale, U.; Park, J. Generative AI in a High School English Career Preparation Units: Student Interactions, Perceptions, and Ethical Concerns. *Comput. Educ. Artif. Intell.* **2026**, *10*, 100588. [\[CrossRef\]](#)
28. Klingbeil, A.; Grützner, C.; Schreck, P. Trust and reliance on AI—An experimental study on the extent and costs of overreliance on AI. *Comput. Hum. Behav.* **2024**, *160*, 108352. [\[CrossRef\]](#)

29. Cai, Q.; Lin, Y.; Yu, Z. Factors Influencing Learner Attitudes Towards ChatGPT-Assisted Language Learning in Higher Education. *Int. J. Hum. Comput. Interact.* **2024**, *40*, 7112–7126. [\[CrossRef\]](#)
30. Chen, B.-H.; Chen, C.-C. Invention of Line-ChatBot: An Innovative Application of ChatGPT API and LINE Bot for Enhanced Student Learning. In *2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII)*; IEEE: New York, NY, USA, 2023; pp. 452–455. [\[CrossRef\]](#)
31. Mohammed, I.A.; Bello, A.; Ayuba, B. Effect of large language models artificial intelligence chatgpt chatbot on achievement of computer education students. *Educ. Inf. Technol.* **2025**, *30*, 11863–11888. [\[CrossRef\]](#)
32. Looi, C.K.; Jia, F. Personalization capabilities of current technology chatbots in a learning environment: An analysis of student-tutor bot interactions. *Educ. Inf. Technol.* **2025**, *30*, 14165–14195. [\[CrossRef\]](#)
33. Lin, Y.T.; Ye, J.H. Development of an Educational Chatbot System for Enhancing Students' Biology Learning Performance. *J. Internet Technol.* **2023**, *24*, 275–281. [\[CrossRef\]](#)
34. Chen, C.-Y.; Juan, Y.-S.; Wang, J.-H.; Yang, S.-H.; Chen, G.-D. Integrate an AI Chatbot-Based Learning Butler Digital system to enhance Students' Grit and Growth Mindset for Improving Learning Outcomes. In *2024 IEEE International Conference on Advanced Learning Technologies (ICALT)*; IEEE: New York, NY, USA, 2024; pp. 21–25. [\[CrossRef\]](#)
35. Tsao, C.C.; Lin, Y.H.; Chou, C.H.; Chang, K.N.; Han, P.H. Design of an Assisted Learning System Based on ChatGPT. In *AICCC'23: Proceedings of the 2023 6th Artificial Intelligence and Cloud Computing Conference*; ACM International Conference Proceeding Series; Association for Computing Machinery: New York, NY, USA, 2023; pp. 247–254. [\[CrossRef\]](#)
36. Zhu, W.; Xing, W.; Lyu, B.; Li, C.; Zhang, F.; Li, H. Bridging the Gender Gap: The Role of AI-Powered Math Story Creation in Learning Outcomes. In *15th International Conference on Learning Analytics and Knowledge, LAK 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 918–923. [\[CrossRef\]](#)
37. Ao, S.I.; Hurwitz, M.; Palade, V. Cognitive Computing and Business Intelligence Applications in Accounting, Finance and Management. *Big Data Cogn. Comput.* **2025**, *9*, 54. [\[CrossRef\]](#)
38. Tonga, J.C.; Clement, B.; Oudeyer, P.-Y. Automatic Generation of Question Hints for Mathematics Problems using Large Language Models in Educational Technology. *arXiv* **2014**, arXiv:2411.03495.
39. Macina, J.; Daheim, N.; Hakimi, I.; Kapur, M.; Gurevych, I.; Sachan, M. MathTutorBench: A Benchmark for Measuring Open-ended Pedagogical Capabilities of LLM Tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2025.
40. Li, S.; Wong, K.W.; Wang, G.; Duong, T.T. A systematic review of multi-modal large language models on domain-specific applications. *Artif. Intell. Rev.* **2025**, *58*, 383. [\[CrossRef\]](#)
41. Gao, J.; Pi, R.; Zhang, J.; Ye, J.; Zhong, W.; Wang, Y.; Hong, L.; Han, J.; Xu, H.; Li, Z.; et al. G-LLaVA: Solving Geometric Problem with Multi-Modal Large Language Model. *arXiv* **2025**. [\[CrossRef\]](#)
42. Xu, S.; Luo, Y.; Shi, W. Geo-LLaVA: A Large Multi-Modal Model for Solving Geometry Math Problems with Meta In-Context Learning. In *LGM3A 2024—Proceedings of the 2nd Workshop on Large Generative Models Meet Multimodal Applications*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 11–15. [\[CrossRef\]](#)
43. Shen, Z.; Chen, Y.; Zhang, J.; Chen, H. How explanatory features of AI and time frame reshape adolescents' decision-making. *Comput. Educ.* **2026**, *248*, 105563. [\[CrossRef\]](#)
44. Kumar, H.; Rothschild, D.M.; Goldstein, D.G.; Hofman, J.M. Math Education With Large Language Models: Peril or Promise? In *Artificial Intelligence in Education. AIED 2025. Lecture Notes in Computer Science*; Cristea, A.I., Walker, E., Lu, Y., Santos, O.C., Isotani, S., Eds.; Springer: Cham, Switzerland, 2025; pp. 60–75. [\[CrossRef\]](#)
45. Almarashdi, H.S.; Jarrah, A.M.; Khurma, O.A.; Gningue, S.M. Unveiling the potential: A systematic review of ChatGPT in transforming mathematics teaching and learning. *Eurasia J. Math. Sci. Technol. Educ.* **2024**, *20*, em2555. [\[CrossRef\]](#)
46. Pepin, B.; Biehler, R.; Gueudet, G. Mathematics in Engineering Education: A Review of the Recent Literature with a View towards Innovative Practices. *Int. J. Res. Undergrad. Math. Educ.* **2021**, *7*, 163–188. [\[CrossRef\]](#)
47. Abramski, K.; Citraro, S.; Lombardi, L.; Rossetti, G.; Stella, M. Cognitive Network Science Reveals Bias in GPT-3, GPT-3.5 Turbo, and GPT-4 Mirroring Math Anxiety in High-School Students. *Big Data Cogn. Comput.* **2023**, *7*, 124. [\[CrossRef\]](#)
48. Gulam, A.-J.B.; Arenas, J.C. Mathematics Performance and Polya's Method in Problem Solving. *World J. Adv. Res. Rev.* **2024**, *23*, 2156–2162. [\[CrossRef\]](#)
49. Lee, C.I. An appropriate prompts system based on the Polya method for mathematical problem-solving. *Eurasia J. Math. Sci. Technol. Educ.* **2017**, *13*, 893–910. [\[CrossRef\]](#)
50. Takači, D.; Stankov, G.; Milanovic, I. Efficiency of learning environment using GeoGebra when calculus contents are learned in collaborative groups. *Comput. Educ.* **2015**, *82*, 421–431. [\[CrossRef\]](#)
51. Kaplar, M.; Radović, S.; Veljković, K.; Simić-Muller, K.; Marić, M. The Influence of Interactive Learning Materials on Solving Tasks That Require Different Types of Mathematical Reasoning. *Int. J. Sci. Math. Educ.* **2022**, *20*, 411–433. [\[CrossRef\]](#)
52. Olsson, J. Relations Between Task Design and Students' Utilization of GeoGebra. *Digit. Exp. Math. Educ.* **2019**, *5*, 223–251. [\[CrossRef\]](#)

53. Cai, S.; Bao, K.; Guo, H.; Zhang, J.; Song, J.; Zheng, B. GeoGPT4V: Towards Geometric Multi-modal Large Language Models with Geometric Image Generation. *arXiv* **2024**. [[CrossRef](#)]
54. Wang, S.; Hew, K.F. Enhancing a large language model with a chain-of-metacognitive reasoning approach increases argumentative writing evaluation accuracy, student writing outcomes, and mental effort. *Comput. Educ.* **2026**, *250*, 105621. [[CrossRef](#)]
55. CLin, H.; Zhou, K.; Li, L.; Sun, L. Integrating generative AI into digital multimodal composition: A study of multicultural second-language classrooms. *Comput. Compos.* **2025**, *75*, 102895. [[CrossRef](#)]
56. Lai, C.Y.; Cheung, K.Y.; Chan, C.S. Exploring the role of intrinsic motivation in ChatGPT adoption to support active learning: An extension of the technology acceptance model. *Comput. Educ. Artif. Intell.* **2023**, *5*, 100178. [[CrossRef](#)]
57. Al-Gahtani, S.S.; Hubona, G.S.; Wang, J. Information technology (IT) in Saudi Arabia: Culture and the acceptance and use of IT. *Inf. Manag.* **2007**, *44*, 681–691. [[CrossRef](#)]
58. Wut, T.M.; Lee, S.W.; Xu, J.; Kwok, M.L.J. Do Trusting Belief and Social Presence Matter? Service Satisfaction in Using AI Chatbots: Necessary Condition Analysis and Importance-Performance Map Analysis. *Informatics* **2025**, *12*, 91. [[CrossRef](#)]
59. Dysvik, A.; Kuvaas, B. The relationship between perceived training opportunities, work motivation and employee outcomes. *Int. J. Train. Dev.* **2008**, *12*, 138–157. [[CrossRef](#)]
60. Koufaris, M. Applying the technology acceptance model and flow theory to online consumer behavior. *Inf. Syst. Res.* **2002**, *13*, 205–223. [[CrossRef](#)]
61. Fornell, C.; Larcker, D.F. Structural Equation Models With Unobservable Variables and Measurement Error: Algebra and Statistics. *J. Mark. Res.* **1981**, *18*, 382–390. [[CrossRef](#)]
62. Bagozzi, R.R.; Yi, Y. On the Evaluation of Structural Equation Models. *J. Acad. Mark. Sci.* **1988**, *16*, 74–94. [[CrossRef](#)]
63. Kock, N. Common method bias in PLS-SEM: A full collinearity assessment approach. *Int. J. E-Collab.* **2015**, *11*, 1–10. [[CrossRef](#)]
64. Davis, F.D.; Bagozzi, R.P.; Warshaw, P.R. Extrinsic and Intrinsic Motivation to Use Computers in the Workplace¹. *J. Appl. Soc. Psychol.* **1992**, *22*, 1111–1132. [[CrossRef](#)]
65. Davis, F.D.; Bagozzi, R.P.; Warshaw, P.R. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Manag. Sci.* **1989**, *35*, 982–1003. [[CrossRef](#)]
66. Davis, F.D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Q.* **1989**, *13*, 319–340. [[CrossRef](#)]
67. Zhou, L.; Xue, S.; Li, R. Extending the Technology Acceptance Model to Explore Students' Intention to Use an Online Education Platform at a University in China. *Sage Open* **2022**, *12*, 21582440221085259. [[CrossRef](#)]
68. Chang, C.C.; Liang, C.; Yan, C.F.; Tseng, J.S. The Impact of College Students' Intrinsic and Extrinsic Motivation on Continuance Intention to Use English Mobile Learning Systems. *Asia-Pac. Educ. Res.* **2013**, *22*, 181–192. [[CrossRef](#)]
69. Okonkwo, C.W.; Ade-Ibijola, A. Chatbots applications in education: A systematic review. *Comput. Educ. Artif. Intell.* **2021**, *2*, 100033. [[CrossRef](#)]
70. Noraset, T.; Supratak, A.; Ragkhitwetsagul, C.; Worathong, N.; Tuarob, S. Evaluating lab assistant chatbot on student learning and behaviors in a programming short course. *Comput. Educ. Artif. Intell.* **2026**, *10*, 100527. [[CrossRef](#)]
71. Trocado, A.; Santos, J.M.D.; Saimon, M.; Lavicza, Z. Learning Quadratic Functions with ChatGPT: An Innovative Experience in High School Mathematics Education. *Int. J. Educ. Math. Sci. Technol.* **2026**, *14*, 296–314. [[CrossRef](#)]
72. Wang, Y.L.; Lo, C.W. The effects of response time on older and young adults' interaction experience with Chatbot. *BMC Psychol.* **2025**, *13*, 150. [[CrossRef](#)] [[PubMed](#)]
73. Anthony, L.; Yang, J.; Koedinger, K.R. Evaluation of multimodal input for entering mathematical equations on the computer. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*; Association for Computing Machinery: New York, NY, USA, 2005; pp. 1184–1187.
74. Khan Academy. *Khan Academy's AI-Powered Teaching Assistant*; Khan Academy Meet Khanmigo: Khan Academy's AI-Powered Teaching Assistant; Khan Academy: Mountain View, CA, USA, 2023.
75. Chen, E.; Huang, Z.; Liu, J.; Liu, Q.; Sha, J.; Wang, S.; Wu, J.; Xiao, T. SocraticLM: Exploring Socratic Personalized Teaching with Large Language Models. In *Advances in Neural Information Processing Systems*; Neural Information Processing Systems Foundation: La Jolla, CA, USA, 2024. [[CrossRef](#)]
76. Scarlatos, A.; Liu, N.; Lee, J.; Baraniuk, R.; Lan, A. Training LLM-based Tutors to Improve Student Learning Outcomes in Dialogues. In *Artificial Intelligence in Education*; Springer: Cham, Switzerland, 2025. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.