software

MDPI

Article

AI Testing for Smart Learning Applications—A Case Study

Tony Li, Quoc Thang Nguyen, Jerry Gao * and Radhika Agarwal

Department of Computer Engineering, College of Engineering, San Jose State University, San Jose, CA 95192, USA; tony.z.li@sjsu.edu (T.L.); quocthang.nguyen@sjsu.edu (Q.T.N.); radhika.agarwal@sjsu.edu (R.A.)
* Correspondence: jerry.gao@sjsu.edu

Abstract

The increasing adoption of artificial intelligence (AI) in smart learning environments has heightened the need for systematic, reliable testing of AI-driven educational applications. Existing studies primarily rely on benchmark accuracy, manual testing, or user-based assessment, offering limited insight into robustness, coverage, and failure behavior. These limitations are driven by the lack of standardized intelligence quality criteria, inadequate test automation support, complex diversity in Q&A tasks, and the difficulty of automatically validating test results in smart learning applications. This paper investigates model-based AI testing for Q&A-based smart learning applications, using ChatGPT (GPT-5) as a case study to evaluate its intelligence quality in college algebra question answering tasks that support student learning. A three-dimensional (3D) AI testing framework structures testing along input, context, and output dimensions to enable model-driven test generation, controlled contextual variation, and consistent validation. College algebra problems selected from a standard undergraduate textbook are used to construct representative test cases. Controlled image-based data augmentation and structured similarity-based validation mechanisms are employed to support automated test execution and result analysis. Empirical results demonstrate that the proposed approach improves intelligence quality coverage and provides more diagnostic insight than ad hoc evaluation methods.

Keywords: artificial intelligence testing; smart learning application testing; model-based testing; educational AI; large language models; robustness evaluation

1. Introduction

The increasing adoption of artificial intelligence (AI) technologies in education has transformed how learning content is delivered, assessed, and supported. In particular, large language model (LLM)-based systems are now widely used in smart learning environments to provide instructional assistance, problem-solving support, and real-time feedback. Market analyses indicate that the global smart learning market is projected to grow from approximately USD 55.9 billion in 2023 to over USD 305 billion by 2033, with a compound annual growth rate of about 18.5% [1]. This growth is driven by demand for scalable, personalized, and technology-enhanced education solutions. As AI-driven systems become integral components of modern learning platforms, ensuring their reliability and trustworthiness has become a critical concern.

Among AI-enabled educational tools, conversational systems such as chatbots and LLM-based assistants have attracted significant attention. Prior studies show that these Q&A-oriented systems are increasingly deployed to support interactive learning experiences, answer student questions, and assist with academic problem-solving [2,3]. Research

Received: 24 February 2026
Revised: 20 May 2026
Accepted: 28 May 2026
Published: 5 June 2026

on smart classrooms and AI adoption in education further underscores that AI systems now operate within integrated learning ecosystems, raising new challenges in system validation and quality assurance [4,5].

Despite their growing use, evaluations of AI-driven educational systems have largely focused on learning outcomes, user satisfaction, and benchmark accuracy. From a software engineering perspective, such evaluation differs from systematic testing, which aims to explore system behavior, identify failure modes, and assess robustness under diverse conditions. Prior work on AI and machine learning testing highlights challenges such as oracle ambiguity, non-determinism, coverage inadequacy, and validation complexity [6–12]. More recent studies on generative AI testing further emphasize the difficulty of validating conversational systems and LLM-based outputs in dynamic application contexts [13–18]. These works collectively indicate the need for structured, model-based AI testing methodologies.

This challenge is particularly evident in mathematical problem-solving. Studies evaluating LLM performance on mathematics tasks report strong results on routine problems but also reveal weaknesses in multi-step reasoning, coherence, and consistency [19–25]. These findings suggest that benchmarking alone is insufficient for assessing reliability in Q&A-based smart learning applications.

The issues discussed in this paper concern the major difficulties in testing AI-enabled smart learning applications, specifically Q&A-based educational systems for solving mathematical problems. The main contributions are as follows: (1) Recognition of key issues in the educational practice of AI-based learning systems, such as inconsistency in the quality of reasoning, absence of standardized requirements of validation, and the problem of robustness in changing conditions of presentation; (2) A model-driven AI testing approach to smart learning applications that incorporates structured input modelling, variation by context and output validation; (3) Three-dimensional (3D) AI testing framework that allows covering the problem types in education, contexts of interaction, and response behaviors systematically; (4) A test workflow with automated testing based on controlled data augmentation and scalable and repeatable evaluation based on similarity; (5) Experimental testing that illustrates the ability of the proposed framework to provide a more insightful diagnosis of AI reasoning errors than other forms of evaluation.

Motivated by these challenges, this paper investigates model-based AI testing for Q&A-based smart learning applications, using ChatGPT (GPT-5) solving college algebra problems as a representative case study to evaluate the quality of intelligence in educational question answering tasks. A three-dimensional (3D) AI testing framework is adopted to structure test generation, contextual variation, and output validation. This work contributes a systematic testing methodology tailored to AI-driven educational problem-solving systems. The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 discusses AI testing challenges in smart learning applications. Section 4 provides the methodology used and the proposed 3D AI testing model. Section 5 describes model-driven test generation and augmentation. Section 6 reports validation results and failure analysis. Section 7 presents a discussion section with a detailed explanation of the contribution, ethical considerations, and threats to validity. Section 8 concludes the paper.

2. Related Work

2.1. Understanding of Smart Learning from an AI Testing Perspective

Smart learning environments represent an evolution of traditional digital learning systems by integrating artificial intelligence, data analytics, and intelligent interaction mechanisms. Rather than emphasizing content delivery alone, smart learning focuses on adaptability, interaction, automation, assessment intelligence, and system-level integration, as illustrated in Figure 1.

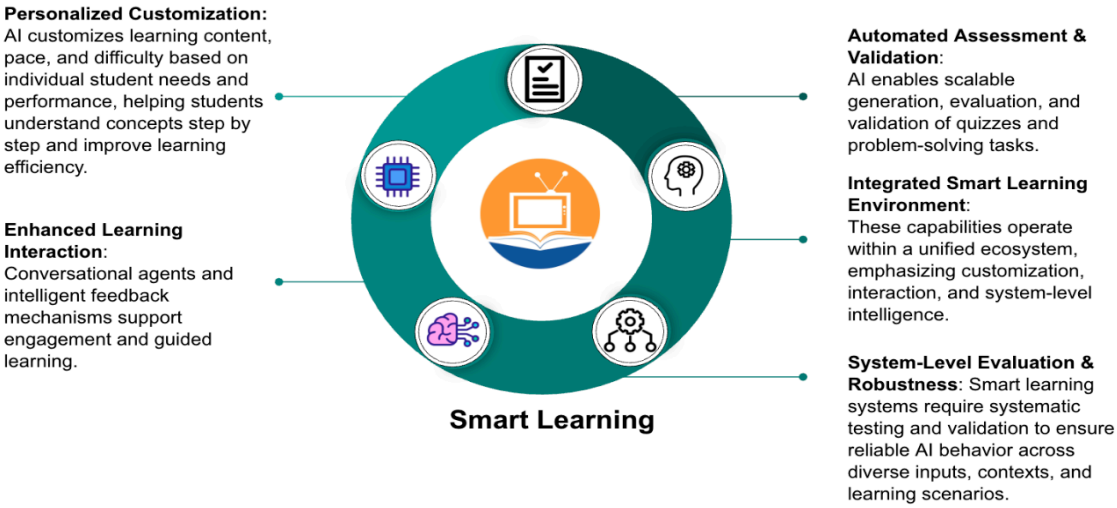


Figure 1. Characteristics of smart learning from five different perspectives.

One important perspective concerns customized teaching materials and learning content. Li [26] proposes an AI-based e-learning intelligence model that dynamically adjusts instructional content and feedback based on learner performance and behavioral analytics. In such systems, the AI model influences not only what content is delivered but also how learning paths are sequenced. Similarly, smart classroom frameworks emphasize adaptive learning mechanisms operating within an integrated digital ecosystem [4]. From a testing standpoint, personalization introduces variability across users, requiring validation of consistency, fairness, and robustness under different learner profiles.

A second perspective involves enhanced interactive learning experiences enabled by conversational agents and educational chatbots. Riza et al. [2] systematically review chatbot-based learning support systems that provide real-time Q&A assistance, while Colace et al. [3] demonstrate a chatbot implementation that supports guided learning interactions. These systems rely on natural language understanding and multi-turn dialogue, where responses may vary depending on context and prior exchanges. Multi-level conversational AI testing research further highlights challenges in validating dialogue flow, reasoning continuity, and non-deterministic response generation [15,16]. Consequently, interaction-driven smart learning applications require testing approaches that account for conversational variability rather than deterministic input–output matching.

A third perspective addresses automated generation and evaluation of quizzes and problem-solving exercises. Studies evaluating LLM-based systems on mathematical tasks show that AI models can generate and solve algebraic problems across diverse difficulty levels [20,21]. For example, comparative evaluations of ChatGPT 4.0 on structured math problems demonstrate the feasibility of automated computational and conceptual Q&A tasks. However, these evaluations often focus on accuracy rates rather than systematic coverage of task types and presentation variations. Testing such automation-oriented smart learning applications, therefore, requires structured modeling of problem categories and generation conditions.

The fourth perspective emphasizes intelligent validation of learning assessment, focusing on how AI-generated responses are evaluated. Hmoud et al. [27] propose rubric-based assessment frameworks to evaluate the quality, completeness, and coherence of explanations in chatbot-generated solutions. While such frameworks improve interpretability compared to simple correctness metrics, they remain primarily evaluation-oriented. Broader AI testing research emphasizes the need for systematic validation frameworks capable of handling ambiguous oracles, non-deterministic outputs, and robustness concerns in

AI systems [6,11–13]. In Q&A-based smart learning applications, intelligent assessment therefore requires structured validation criteria beyond surface-level answer comparison.

Finally, smart learning applications operate within integrated AI-driven ecosystems [4,5], combining personalization, interaction, automation, and assessment intelligence. This integration increases system-level complexity and expands the effective test space. Testing must therefore consider not only isolated model performance but also how AI components behave under diverse learning contexts and interaction scenarios. Among these perspectives, this paper focuses specifically on intelligent validation of learning assessment within Q&A-based smart learning applications. By grounding testing in structured AI test modeling, the proposed approach aims to provide systematic coverage of intelligence quality for AI-driven educational problem-solving systems.

2.2. Comparison of Related Work for AI-Based Mathematical Problem-Solving

Existing research on AI-based mathematical problem-solving has predominantly emphasized performance evaluation rather than systematic AI testing. For example, Plevris et al. [28] compare ChatGPT-3.5, ChatGPT-4, and Google Bard on structured math and logic problems using curated datasets and manual verification of answers. Similarly, Wei [20] evaluates ChatGPT's performance on National Assessment of Educational Progress (NAEP) mathematics problems by reporting correctness rates across predefined question categories. While these studies provide useful accuracy comparisons, they rely on fixed problem sets and manual correctness checking rather than structured coverage modeling.

Comparisons between AI systems and student performance further reinforce this evaluation-oriented paradigm. Hidayatullah et al. [21] assess AI performance relative to secondary school students by comparing final-answer accuracy, but the analysis focuses on outcome comparisons rather than behavioral robustness or failure traceability. Vidal [22] evaluates LLM performance on math word problems, highlighting inconsistencies in multi-step reasoning, while Gandolfi [23] analyzes GPT-4 behavior across calculus tasks and grading scenarios, reporting coherence loss and variability in explanations. Although these studies identify reasoning weaknesses, they employ ad hoc validation strategies and do not define explicit test models that structure input categories, contextual variation, or output classification.

Parallel developments in AI testing research address broader methodological challenges beyond mathematical problem-solving. Foundational studies on AI and ML testing identify difficulties related to oracle design, non-deterministic outputs, robustness under input perturbation, and adequacy of test coverage [6–12,16–19]. For generative AI systems, Tao et al. [13] and Aleti [17] emphasize the complexity of validating LLM-based applications where responses may vary across executions. Multi-level conversational AI testing frameworks propose layered validation approaches that separate functional correctness from dialogue behavior analysis [15], yet they fall short of integrating structured input–context–output modeling tailored to smart learning tasks.

To mitigate oracle ambiguity, datamorphic testing techniques define relational expectations across transformed inputs rather than relying on single-answer matching [11]. Meanwhile, AI-driven software testing surveys examine how AI techniques can support automated test generation and validation workflows [8,9,29]. However, these works primarily focus on improving testing automation pipelines rather than designing domain-specific test models for educational Q&A systems.

Model-based AI testing approaches have been demonstrated in other AI domains, such as computer vision systems, where structured input–context–output modeling enables systematic coverage, reasoning, and failure traceability [30]. Nevertheless, few studies integrate classification-based input modeling, contextual variation control, and structured

output validation into a unified testing framework specifically designed for Q&A-based smart learning applications.

As summarized in Table 1, most existing work emphasizes evaluation using fixed problem sets and manual validation, with limited consideration of coverage reasoning or failure localization within a defined testing space. This gap motivates adopting a model-based three-dimensional AI testing approach that explicitly represents input characteristics, contextual conditions, and output validation criteria for systematic, reproducible evaluation of smart learning applications. All figures in this paper are either created by the authors based on the proposed framework or adapted from cited references, as explicitly indicated in the figure captions.

Table 1. Comparison of evaluation and AI-based testing approaches for smart learning applications.

Ref.	Objective	Automated Test Validation	Test Modeling	Test Generation	Augmentation	AI-Based Testing
[2]	Systematically review the use of chatbots in e-learning contexts	No (literature-based evaluation)	No test model	Not addressed	Not addressed	No
[3]	Demonstrate a chatbot application for e-learning support	No (manual evaluation of responses)	No explicit test model	Manually prepared example intersections	Not addressed	No
[5]	Evaluate a hybrid K-12 e-learning chatbot	No (questionnaire-based and experimental evaluation)	No test model	Manually designed learning scenarios	Not addressed	No
[20]	Compare LLM performance on math and login problems	No (manual correctness judgment using answer keys)	No explicit modeling of input, context, or output	Fixed, manually curated problem set	Repeated execution of identical inputs only	No
[23]	Analyze reliability and coherence of GPT-4 in calculus solving and grading	No (manual grading and qualitative analysis)	Task-oriented experimental setup	Manual problem selection	Not addressed	No
[28]	Develop a rubric for assessing chatbot task-solving quality	No (human-based rubric scoring)	No test model; (evaluation framework only)	Not addressed	Not applicable	No
This paper	Systematic testing of ChatGPT (GPT-5) for college algebra problem-solving	Yes (AI-based) similarity evaluation)	Explicit 3D AI test model	AI-driven test generation	Controlled contextual and presentation augmentation	Yes

3. AI Testing Challenges and Requirements for Q&A-Based Smart Learning Applications

Smart learning applications increasingly rely on large language models (LLMs) to support Q&A-driven instructional services, automated problem-solving, and intelligent learning assessment. In such systems, AI components do not merely produce outputs; they generate reasoning processes, explanations, and contextual responses that directly influence students' understanding. Consequently, testing must move beyond conventional functional validation toward systematic evaluation of Q&A intelligence quality, robustness, and coverage within smart learning environments. In this case study, ChatGPT (GPT-5) is evaluated as the Q&A reasoning engine of a smart learning application for college algebra. Students submit screenshot-based exercises and receive step-by-step explanations. This use case represents a typical Q&A-based smart learning scenario in which the quality of

intelligence depends on correct reasoning, an appropriate explanation structure, contextual interpretation, and robust presentation. Testing, therefore, must operate at the application level, where learning objectives, interaction patterns, and assessment expectations intersect.

3.1. Test Focuses and Requirements for Q&A Smart Learning Applications

Figure 2 presents a taxonomy of test focuses and requirements for Q&A-based smart learning applications. From a learning-task perspective, questions may involve comparison, analysis, or application reasoning, each requiring different cognitive and computational capabilities. From an interaction perspective, Q&A tasks span short conceptual questions, computational exercises, figure-based reasoning, and table-based interpretation. In the context of college algebra, this includes understanding symbolic formulas, performing algebraic manipulation, interpreting graphs, analyzing tabular data, and solving contextualized problems. Effective AI testing must therefore evaluate not only final-answer correctness but also reasoning coherence, explanation completeness, instructional clarity, and alignment with learning objectives. These multi-dimensional requirements define the scope of intelligence quality that must be systematically modeled and validated in Q&A smart learning applications. Figure 2 is derived from established smart learning frameworks and integrates perspectives from prior studies on adaptive learning, AI-driven interaction, and intelligent assessment [4–6].

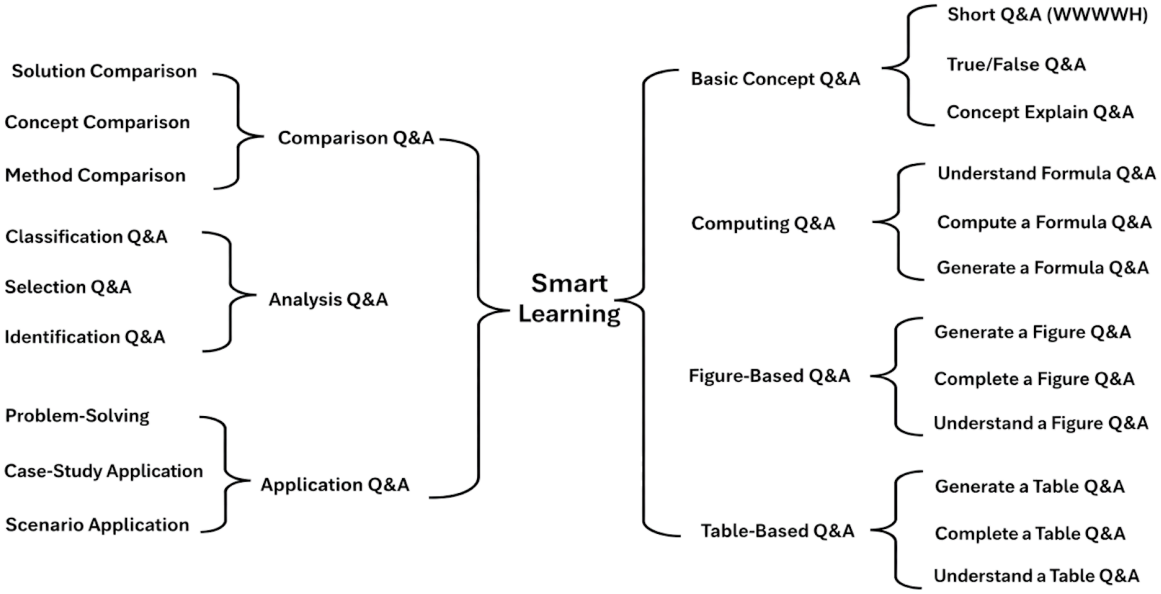


Figure 2. Test focuses and requirements for Q&A smart learning application diagram.

3.2. AI Testing Challenges in Smart Learning Applications

Given the diversity of Q&A task categories described above, Figure 3 summarizes major AI testing challenges for smart learning applications. First, the lack of standardized criteria for intelligence quality makes it difficult to define measurable expectations for reasoning depth, explanation clarity, and pedagogical adequacy. Unlike deterministic educational software, Q&A-based AI systems generate non-unique responses, introducing oracle ambiguity and validation complexity. Second, current test automation platforms provide limited support for validating open-ended LLM outputs. Manual testing or static benchmarking remains a common practice, which restricts reproducibility and systematic coverage analysis. Third, diversity in Q&A formats and interaction patterns significantly expands the effective test space. Smart learning systems must accommodate conceptual questions, computational tasks, graphical interpretation, and contextual application scenarios, each of which interacts with variable input presentation conditions. Finally, challenges related to automated result validation, coverage adequacy, data augmentation, and contin-

uous testing further complicate quality assurance. These factors collectively demonstrate that intelligence quality validation in Q&A-based smart learning applications requires structured modeling of input diversity, contextual variation, and output evaluation criteria.

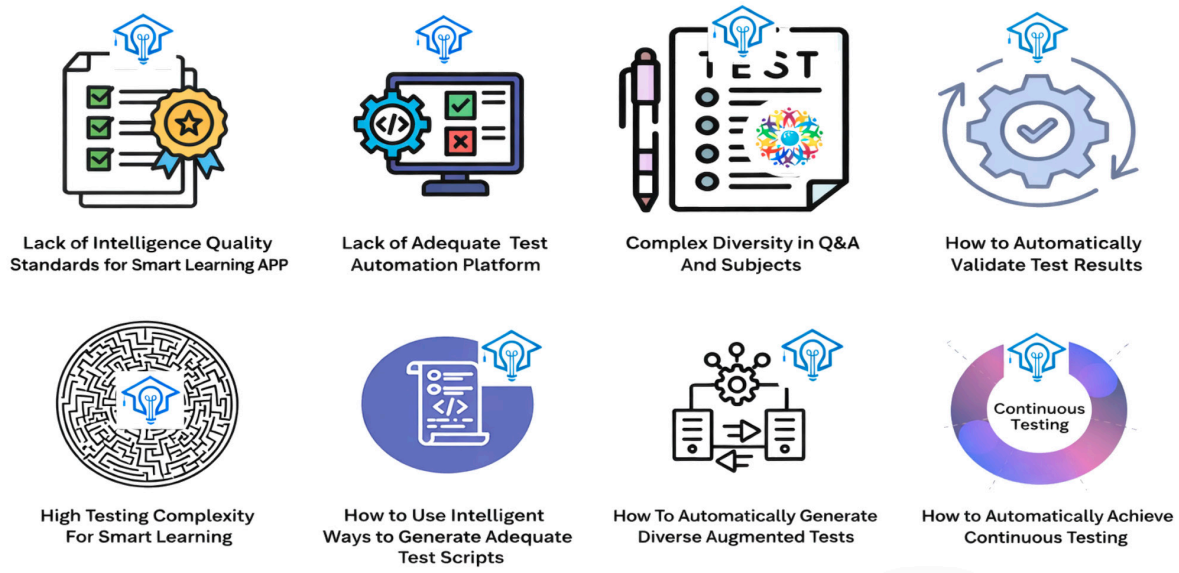


Figure 3. AI testing challenges for smart learning applications.

3.3. Smart Learning Application Test Automation Process

To operationalize these requirements, Figure 4 illustrates a generalized test automation process for smart learning applications. The process begins with identifying Q&A testing requirements, including learning objectives, problem categories, and expected intelligence quality criteria. These requirements inform structured test modeling and analysis, where Q&A task types and contextual factors are explicitly categorized. Subsequently, model-driven test case generation and controlled data augmentation are applied to produce diverse and representative Q&A scenarios. Generated tests are executed through systematic scripting, and outputs are validated using structured correctness and intelligence quality criteria. Finally, test quality evaluation and feedback support iterative refinement and continuous robustness assessment. In the college algebra case study, this process enables systematic exploration of ChatGPT's Q&A intelligence behavior across varied algebraic tasks and presentation contexts rather than relying on isolated examples or fixed benchmark datasets.

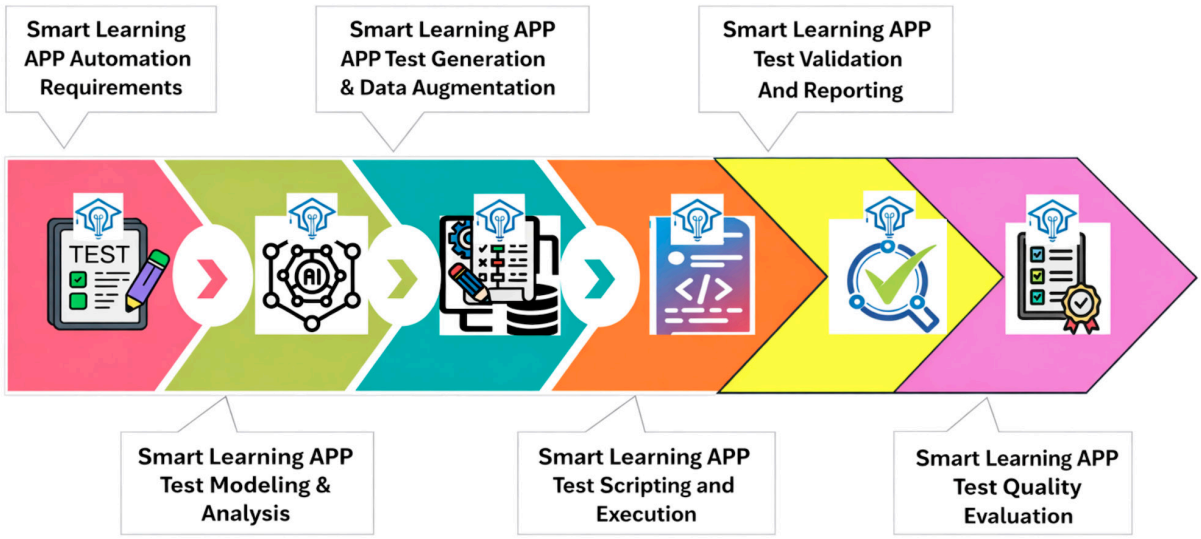


Figure 4. Test automation process for smart learning applications.

4. AI Test Modeling Methodology

This paper adopts a model-driven AI testing approach to systematically analyze the quality of intelligence in Q&A-based smart learning applications. The following steps constitute the methodology:

Step 1: Defining the problem and requirement analysis. Problem-solving educational tasks (college algebra) are studied to determine the main testing requirements, including the soundness of reasoning, the quality of explanation, and robustness across different input conditions.

Step 2: 3D Framework model testing. A three-dimensional testing model is prepared with the following. Input classification tree (ICT): types and forms of problems. Context classification tree (CCT): changes in the environment and presentation. Output classification tree (OCT): correctness and quality of response.

Step 3: Test case generation (model-driving). Test cases are created systematically by mapping combinations of ICT and CCT to executable test conditions.

Step 4: Data augmentation control. Image-based transformations are used to make the educational inputs simulate the variability of the real world.

Step 5: Test implementation and acceptance. The use of responses is assessed through validity in mathematics (oracle validation) and organized output compliance (OCT-based classification).

Step 6: Diagnosis and analysis of failure. ICT CCT conditions are also used to trace failures and establish their root causes, such as perception errors or reasoning inconsistencies. The methodology builds on conventional model-based testing, adding AI-specific model testing challenges, including non-determinism, oracle ambiguity, and contextual variability.

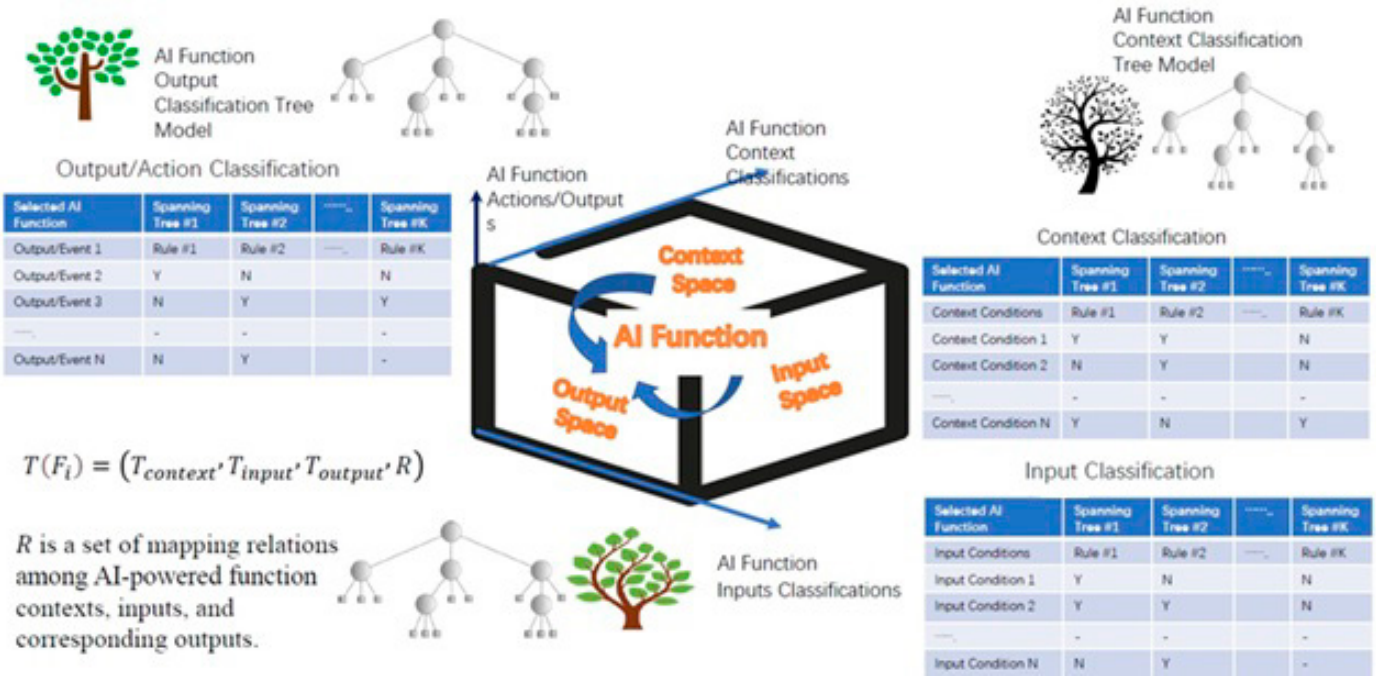
4.1. Rationale for a Three-Dimensional Test Model

Given the testing requirements and challenges identified in Section 3, the evaluation space for Q&A-based smart learning applications must be explicitly structured. In screenshot-based college algebra problem-solving, system behavior cannot be adequately described by a simple input–output mapping. Responses depend on the algebraic task, presentation conditions, and the structural form of the generated explanation. To systematically evaluate Q&A intelligence quality under these interacting factors, we adopt a three-dimensional (3D) AI testing model that separates input, context, and output into independent dimensions. The input dimension represents algebraic task categories and representation formats (printed vs. handwritten). The context dimension captures presentation-related factors such as clarity, lighting, contrast, and background conditions. The output dimension defines observable response classifications, including correctness and reasoning structure. This separation makes the evaluation space explicit, enabling systematic test generation, coverage analysis, and traceable failure characterization under controlled smart learning conditions.

4.2. Construction of the 3D Test Model

The 3D AI testing model is constructed using a classification-tree–based approach. Each dimension—input classification tree (ICT), context classification tree (CCT), and output classification tree (OCT)—is defined independently, with leaf nodes representing testable partitions of the corresponding space. Each test execution is mapped to a tuple consisting of an ICT leaf, a CCT configuration, and an observed OCT label. This mapping forms a three-dimensional classification decision table (3D-CDT), which provides a unified representation of test coverage across inputs, contexts, and outputs. Figures 5 and 6

illustrate the relationship between individual classification trees and the integrated 3D model, including both single-feature and multi-feature (forest) tree models.



Selected AI Function	Spanning Tree #1	Spanning Tree #2	.....	Spanning Tree #K
Output/Event 1	Rule #1	Rule #2	.....	Rule #K
Output/Event 2	Y	N		N
Output/Event 3	N	Y		Y
.....	-	-		-
Output/Event N	N	Y		-

Selected AI Function	Spanning Tree #1	Spanning Tree #2	.....	Spanning Tree #K
Context Conditions	Rule #1	Rule #2	.....	Rule #K
Context Condition 1	Y	Y		N
Context Condition 2	N	Y		N
.....	-	-		-
Context Condition N	Y	N		Y

Selected AI Function	Spanning Tree #1	Spanning Tree #2	.....	Spanning Tree #K
Input Conditions	Rule #1	Rule #2	.....	Rule #K
Input Condition 1	Y	N		N
Input Condition 2	Y	Y		N
.....	-	-		-
Input Condition N	N	Y		-

Figure 5. 3D AI function test model with three classification tables. Reprinted from Ref. [25].

The proposed three-dimensional test model is grounded in the classification-tree method (CTM), a well-established test design technique for systematically partitioning the input space [30]. In this context, the input classification tree (ICT), context classification tree (CCT), and output classification tree (OCT) represent structured partitions of the testing space across different dimensions.

More broadly, this approach aligns with model-based testing principles, in which system behavior is represented by abstract models to enable systematic test generation and coverage analysis [31]. The proposed framework extends these concepts by incorporating AI-specific dimensions such as contextual variability and non-deterministic output evaluation.

4.3. Case Study Instantiation

The case study instantiates the 3D model for ChatGPT (GPT-5), solving college algebra problems presented as screenshots. Algebra exercises are selected from the College Algebra 2e textbook [32] and provided to the system as screenshot-based prompts. To support controlled analysis, the input classification tree (ICT) is defined as chapter topic + representation format, where each algebra topic is exercised under both printed and handwritten formats. This design ensures that topic-level reasoning differences and representation-level perceptual effects can be examined systematically. The context classification tree (CCT) and output classification tree (OCT) are held constant across chapters to ensure comparability. The CCT captures presentation conditions applied uniformly across topics, while the OCT defines a fixed taxonomy for labeling observed response behaviors. This instantiation enables consistent coverage reasoning and failure analysis across multiple chapters and test executions.

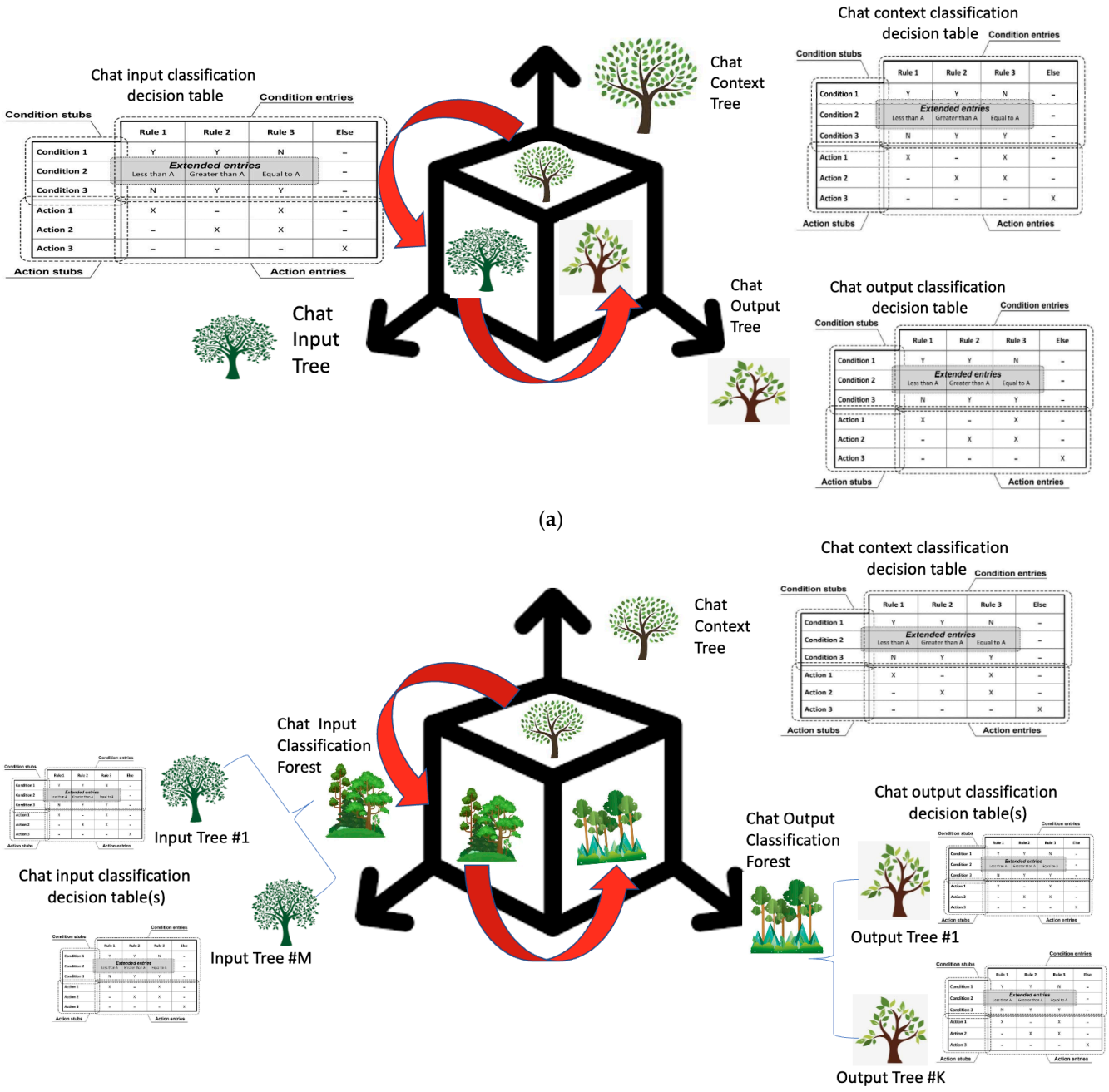


Figure 6. The 3D classification tree models. Reprinted from Ref. [25]. (**a**) Single-feature model. (**b**) Multi-feature (forest) model.

4.3.1. Context Classification Tree Model (CCT)

The context classification tree (CCT) captures perceptual and presentation conditions that may influence the interpretation of screenshot-based problem prompts. It includes four factors: clarity (clear, blurry), lighting (proper, bright, dark), contrast (proper, high, low), and background (plain, cluttered). These factors are defined independently of algebra content and are shared across all selected chapters to enable consistent context-level analysis. Figure 7 presents the CCT used in this case study to represent controlled context conditions for robustness assessment.

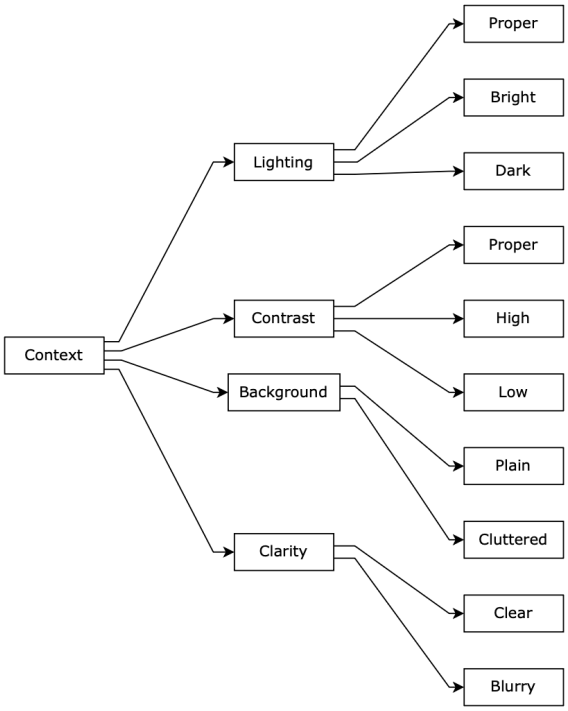


Figure 7. Context classification tree model for the case study.

4.3.2. Output Classification Tree Model (OCT)

The output classification tree (OCT) defines a common outcome taxonomy for labeling model responses consistently across all chapters. It distinguishes between valid and invalid responses and further categorizes outcomes based on correctness, completeness, and explanation structure. By standardizing output labels, the OCT supports consistent aggregation, comparison, and interpretation of observed behaviors across different test conditions. Figure 8 illustrates the OCT applied in this case study.

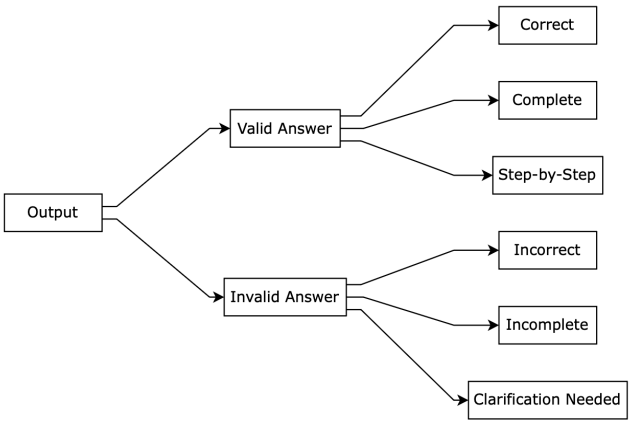


Figure 8. Output classification tree model for the case study.

4.3.3. Input Classification Tree (ICT) Across Four Chapters

Four input classification trees (ICTs) are defined, one for each chapter scope (Ch. 2, Ch. 4, Ch. 6, and Ch. 8). Each ICT partitions chapter content into leaf-level problem families based on topic structure and problem type. The ICT is combined with a format branch (printed vs. handwritten), enabling coverage analysis that accounts for both algebraic content and representation format. Figures 9 and 10 show representative chapter-scoped ICT structures used to support chapter-level coverage reasoning.

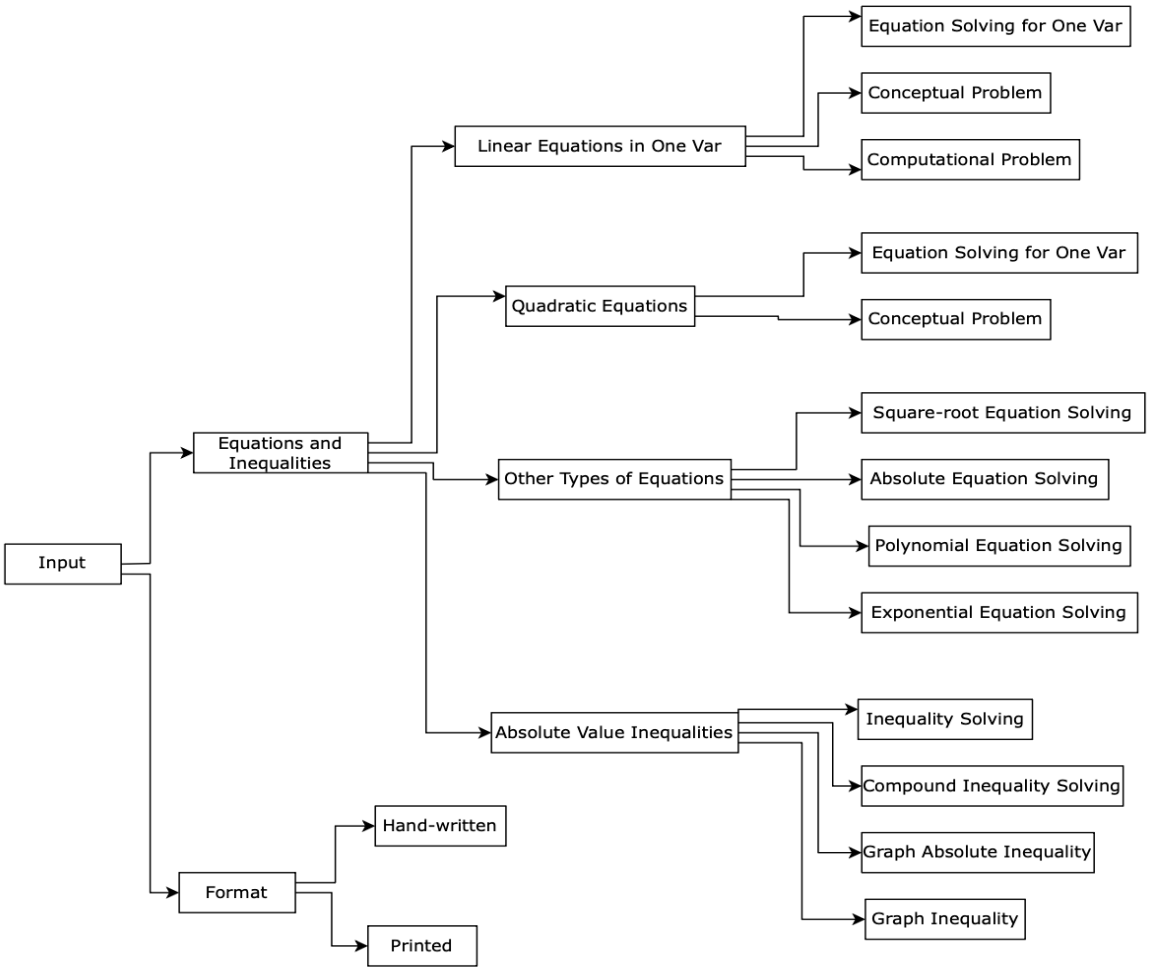


Figure 9. Ch. 2 input classification tree model.

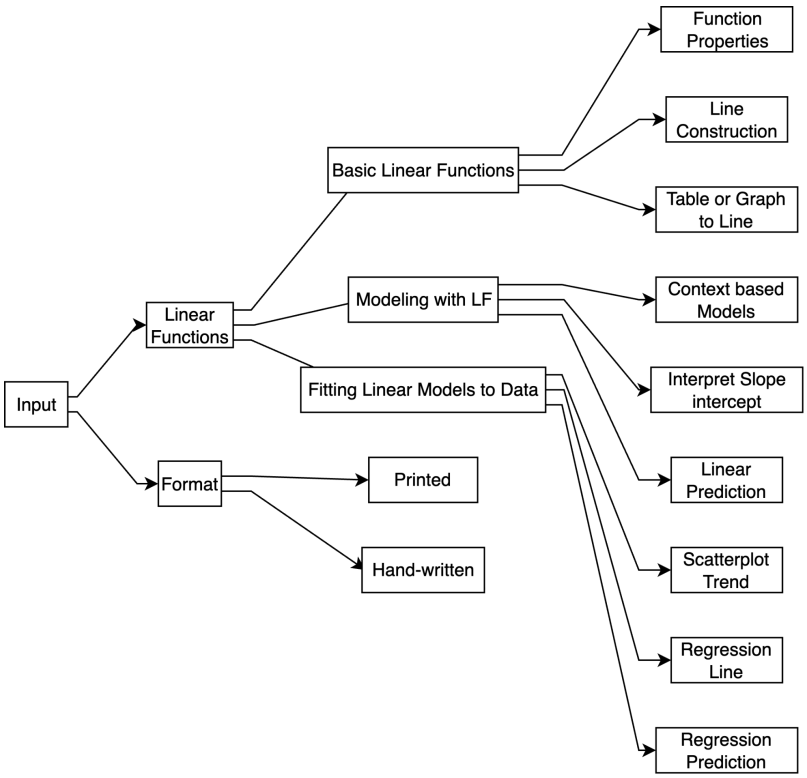


Figure 10. Ch. 4 input classification tree model.

4.4. Integrating the Model: 3D Classification Decision Table (3D-CDT)

The ICT, CCT, and OCT are integrated into a three-dimensional classification decision table (3D-CDT), which serves as the unified test scheme for representing and organizing test conditions. Each test execution is mapped to a specific cell in the three-dimensional space defined by a (ICT leaf, CCT configuration, OCT label) tuple. This mapping enables explicit reasoning about which regions of the model have been exercised and supports traceable coverage analysis. Figures 11 and 12 illustrate the instantiated 3D-CDTs for selected chapter scopes and show how input categories, context conditions, and observed output classifications are combined within a single representation. The 3D-CDT provides a structured view of the testing space without prescribing execution order or validation logic. While the 3D model defines the testing structure, empirical evaluation requires concrete test instances that instantiate model partitions under controlled conditions. The next section operationalizes the 3D-CDT through model-driven test generation and controlled data augmentation.

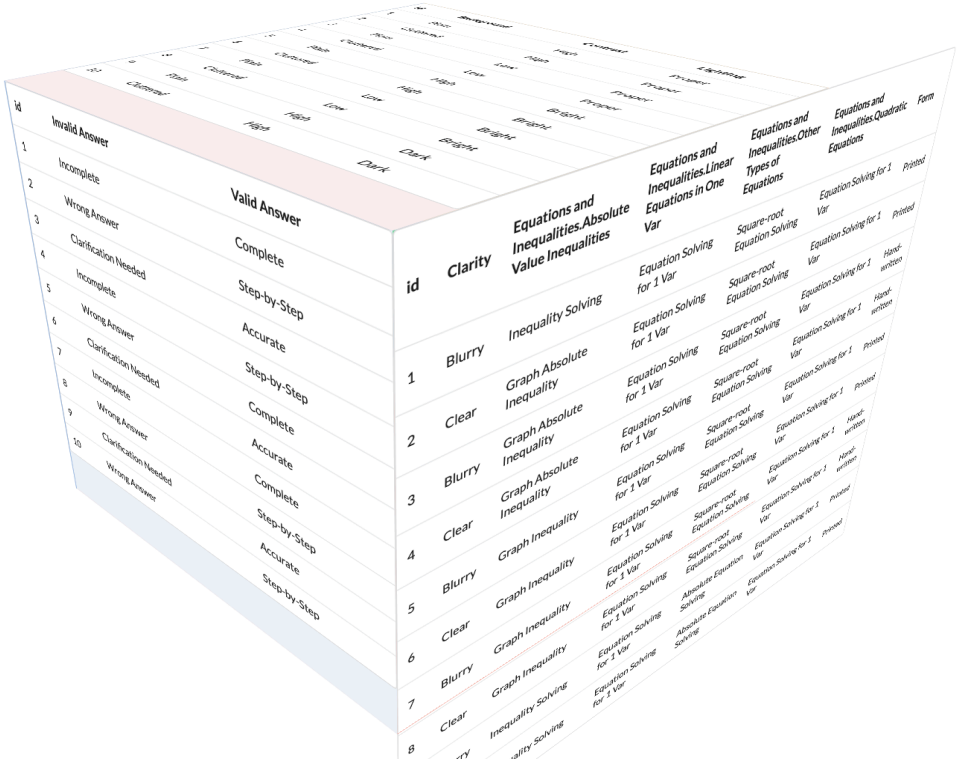


Figure 11. Ch. 2 3D classification decision table.

4.5. Discussion of the 3D AI Testing Model

The three-dimensional (3D) AI testing model presented in this section provides a structured representation of the testing space for Q&A-based smart learning applications. By decomposing the system into input (ICT), context (CCT), and output (OCT) dimensions, the model enables systematic partitioning of both problem characteristics and environmental conditions that influence AI behavior.

This structured separation is particularly important in smart learning scenarios, where system responses depend not only on the problem content but also on presentation factors such as image clarity, contrast, and format. The integration of these dimensions allows the testing process to move beyond simple input–output validation toward coverage-oriented evaluation and traceable failure analysis.

Furthermore, the 3D classification decision table (3D-CDT) provides a unified framework for mapping test cases to specific regions of the testing space, enabling systematic

test generation and reproducibility. This abstraction also facilitates the identification of under-tested combinations of input and context conditions.

The model defined in this section serves as the foundation for the model-driven test generation and data augmentation processes described in Section 5, where the abstract partitions defined by ICT, CCT, and OCT are instantiated into executable test cases.

Figure 12. Ch. 4 3D classification decision table.

5. Test Case Generation and Data Augmentation

5.1. Test Case Generation

Test cases are generated using a model-based strategy derived directly from the three-dimensional classification decision table (3D-CDT). Under this approach, each executable test case corresponds to a specific instantiation of:

(1) An ICT leaf partition (chapter topic + format);
(2) A CCT configuration (clarity, lighting, contrast, background);
(3) An OCT-aligned expected outcome category used for structured validation.

To operationalize this pipeline, base problems are selected from the [32] exercise set and assigned to chapter-scoped ICT partitions. Each problem is instantiated as a concrete test case by pairing the screenshot prompt with explicit ICT and CCT labels together with an OCT-aligned expected outcome definition, the observed chatbot response, and the resulting pass/fail decision. This specification ensures traceability between executed tests and their corresponding positions in the 3D-CDT. Figure 13 presents representative completed test cases from different chapters and illustrates how input format and context conditions are made explicit in the test specification, enabling controlled evaluation of screenshot-based prompts under varied presentation conditions.

5.2. Data Augmentation

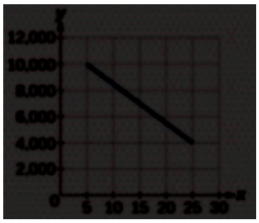
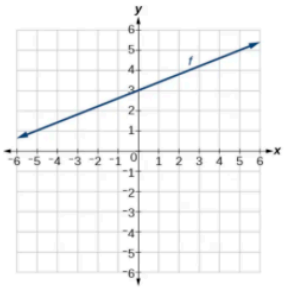
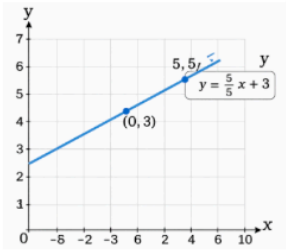
Controlled data augmentation is applied to expand the coverage of presentation conditions while preserving the underlying algebraic task. Starting from an original screenshot prompt, additional prompt instances are generated via image-level transformations that simulate appearance variations without altering mathematical semantics.

Augmentation is implemented using AlbumentationsX, a Python 3.10 library that supports efficient composition of image transformations through a unified API. Using its Compose mechanism, multiple spatial-level transformations are applied consistently across the test set. The study focuses on spatial transformations that affect image geometry and orientation, including horizontal and vertical flips, rotation (with safe rotation to avoid excessive content loss), resizing and scaling, cropping, and geometric distortions such as affine and elastic transformations [33]. Figure 14 illustrates representative spatial-level transformations applied to algebraic expressions from different chapters. All experiments were conducted using Python (version 3.10), with the AlbumentationsX library [34] used for data augmentation. The transformations were applied using the RandomBrightness-Contrast and RandomGamma functions from AlbumentationsX, with $p = 0.5$ for each. To simulate low-contrast conditions, brightness adjustments were sampled from the range $[-0.1, 0.1]$, while contrast adjustment factors were sampled from $[-0.5, -0.2]$ to introduce controlled contrast degradation without obscuring the mathematical expressions. In addition, gamma correction was applied using gamma values sampled from the range [80, 120] to emulate realistic lighting and visibility variations. These parameter ranges were selected to preserve the readability of algebraic symbols while generating sufficient presentation variability for robustness evaluation. The experiments were conducted on a standard computing environment with [brief hardware details, e.g., Intel CPU, 16GB RAM]. These settings ensure reproducibility of the augmentation and testing process.

Problem	Solve the equation: $3x^5 - 6x^3 = 0$	Solve the equation: $\sqrt{3x+7} + \sqrt{x+2} = 1$	Solve the inequality: $\lvert 3x+2 \rvert + 1 \leq 9$	Evaluate the function, round 4 decimal places if necessary. $f(x) = 1.2e^{2x} - 0.3$, for $f(3)$
Input Format	Printed	Hand-written	Printed	Printed
Clarity	Blurry	Clear	Blurry	Clear
Lighting	Proper	Proper	Dark	Bright
Contrast	Low	High	Low	Proper
Background	Plain	Cluttered	Plain	Cluttered
Expected Output	$x = 0, \pm\sqrt{2}$	$x = -2$	$[\frac{-10}{3}, 2]$	$f(3) \approx 483.8146$
Actual Output	$x = 0, \sqrt{2}, -\sqrt{2}$	No real solution.	$\left[-\frac{10}{3}, 2\right]$	$f(3) \approx 483.1846$
Result (Pass/Fail)	Pass	Fail	Pass	Fail

(a)

Figure 13. *Cont.*

Problem	Determine whether this function is increasing or decreasing. $p(x) = \frac{1}{4}x - 5$	Find a linear equation satisfying the conditions, if possible. Passes through $(-1, 4)$ and $(5, 2)$	Find the linear function y, where y depends on x, the number of years since 1980. 	Sketch a line with the given feature. A y-intercept of (0,3) and slope $\frac{2}{5}$.
Input Format	Printed	Printed	Printed	Hand-written
Clarity	Clear	Blurry	Blurry	Clear
Lighting	Proper	Bright	Dark	Proper
Contrast	High	Low	Low	High
Background	Cluttered	Plain	Cluttered	Plain
Expected Output	Increasing	$y = -\frac{1}{3}x + \frac{11}{3}$	$y = -300x + 11{,}500$	
Actual Output	Increasing	$y = -\frac{1}{3}x + \frac{11}{3}$	$y = -300x + 11000$	
Result (Pass/Fail)	Pass	Pass	Fail	Fail

(**b**)

Figure 13. Sample test cases and outcomes. (**a**) Samples from Ch. 2. (**b**) Samples from Ch. 4.

To preserve evaluability, the augmentation policy is constrained to transformations that retain the complete mathematical expression and do not occlude or distort critical symbols. Transformations that render the prompt unreadable or alter its mathematical meaning are excluded. Under these constraints, augmentation systematically increases the diversity of prompt appearances in a controlled and repeatable manner, supporting subsequent analysis of failures attributable to presentation variation rather than problem content.

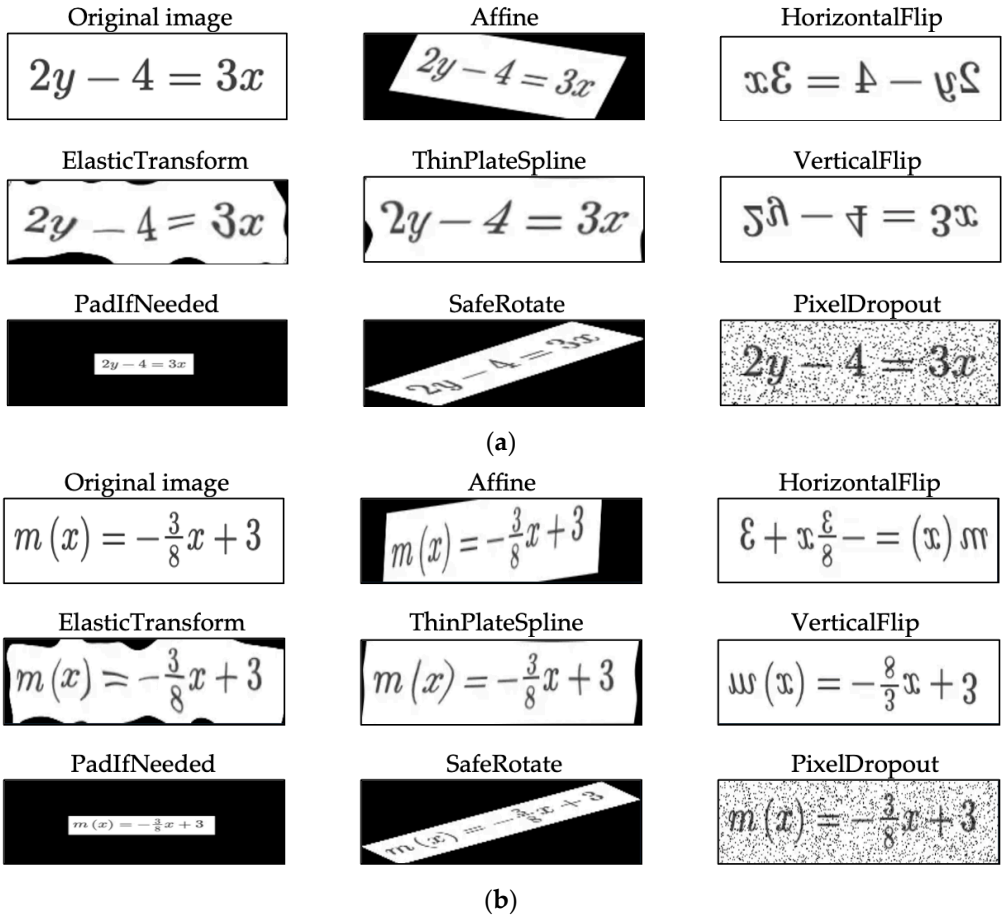


Figure 14. Sample spatial-level transformation outputs. (**a**) Augmented exercise from Ch. 2. (**b**) Augmented exercise from Ch. 4.

5.3. Discussion of Test Generation and Augmentation

The test generation and augmentation process operationalizes the 3D testing model by instantiating ICT–CCT combinations into executable test cases. This enables systematic exploration of diverse problem representations and contextual conditions. The use of controlled augmentation ensures that variations in input presentation can be evaluated independently of underlying problem content. Together, these processes provide a scalable mechanism for extending test coverage and supporting robustness evaluation, forming the basis for the empirical validation presented in Section 6.

6. Test Result Validation and Failure Analysis

Test result validation in smart learning applications must accommodate the non-uniqueness of acceptable learning responses generated by AI-driven systems. For mathematical problem-solving, correct outcomes may be expressed using different algebraic forms, reasoning orders, or explanation styles. Accordingly, validation in this study is designed to assess learning task outcomes produced by an AI component within a smart learning application rather than to compare outputs against a single fixed reference. For screenshot-based tasks, validation further accounts for presentation conditions represented in the context classification tree (CCT). Figure 15 summarizes the validation approaches for intelligent AI systems and situates the layered protocol adopted in this case study.

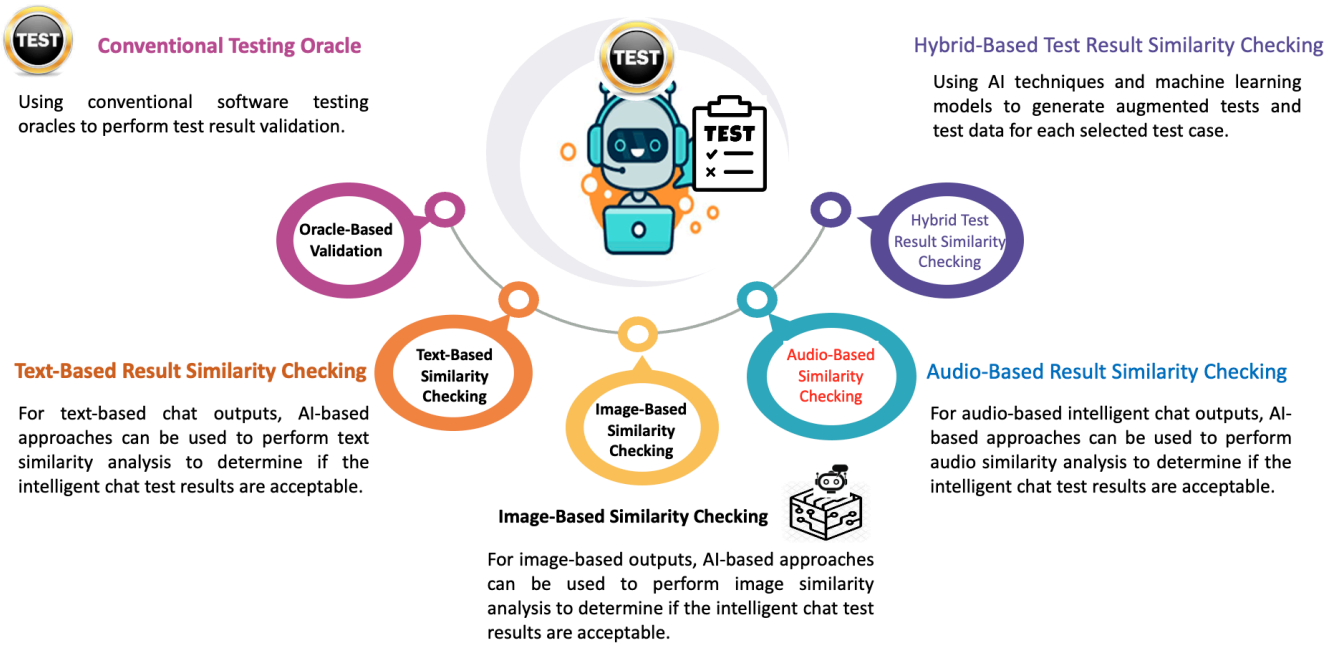


Figure 15. Test result validation approaches for Q&A smart learning applications.

6.1. Validation Protocol and Acceptance Criteria

Validation is conducted at two complementary levels, mathematical correctness and output compliance, reflecting the dual objectives of learning accuracy and instructional quality in smart learning applications.

- Mathematical correctness (oracle layer): Each test case is evaluated to determine whether the AI-generated response yields a mathematically correct solution for the referenced exercise. Correctness is assessed based on algebraic equivalence rather than syntactic form, ensuring that functionally equivalent solutions are treated consistently during testing.
- Output compliance (OCT layer): For responses that satisfy mathematical correctness, additional validation is performed against the OCT-aligned expected outcome category associated with the test case. A test case is labeled "pass" only if it satisfies both correctness and the specified OCT requirement; otherwise, it is labeled "fail" and assigned an OCT-invalid category. For step-structured explanations, similarity-based evaluation is applied to support repeatable acceptance decisions when instructional reasoning content is preserved but phrasing varies. Figure 16 illustrates the mechanisms for text similarity used to operationalize this validation layer.

6.2. Manual Testing Results

Manual testing establishes a baseline set of testing outcomes under the defined ICT–CCT–OCT model, serving as an initial empirical reference for subsequent automated testing. This phase supports rule-consistent oracle checking for mathematical correctness and OCT-aligned labeling at the level of individual algebra problems. Although the prompts in this phase are unaugmented, each test case is executed under an explicitly recorded CCT configuration (clarity, lighting, contrast, background), enabling traceable interpretation of outcomes under realistic presentation variability.

Table 2 summarizes the chapter-scoped minimum ICT coverage implied by defining each chapter's ICT as chapter topic + format (printed, handwritten). Fully enumerating the CCT space over this baseline would yield a substantially larger test space; therefore, the manual suite operationalizes coverage by assigning one CCT combination per

leaf-level ICT obligation, using randomized CCT selections across cases. This strategy preserves minimum ICT coverage while exercising a diverse, though non-exhaustive, set of context conditions.

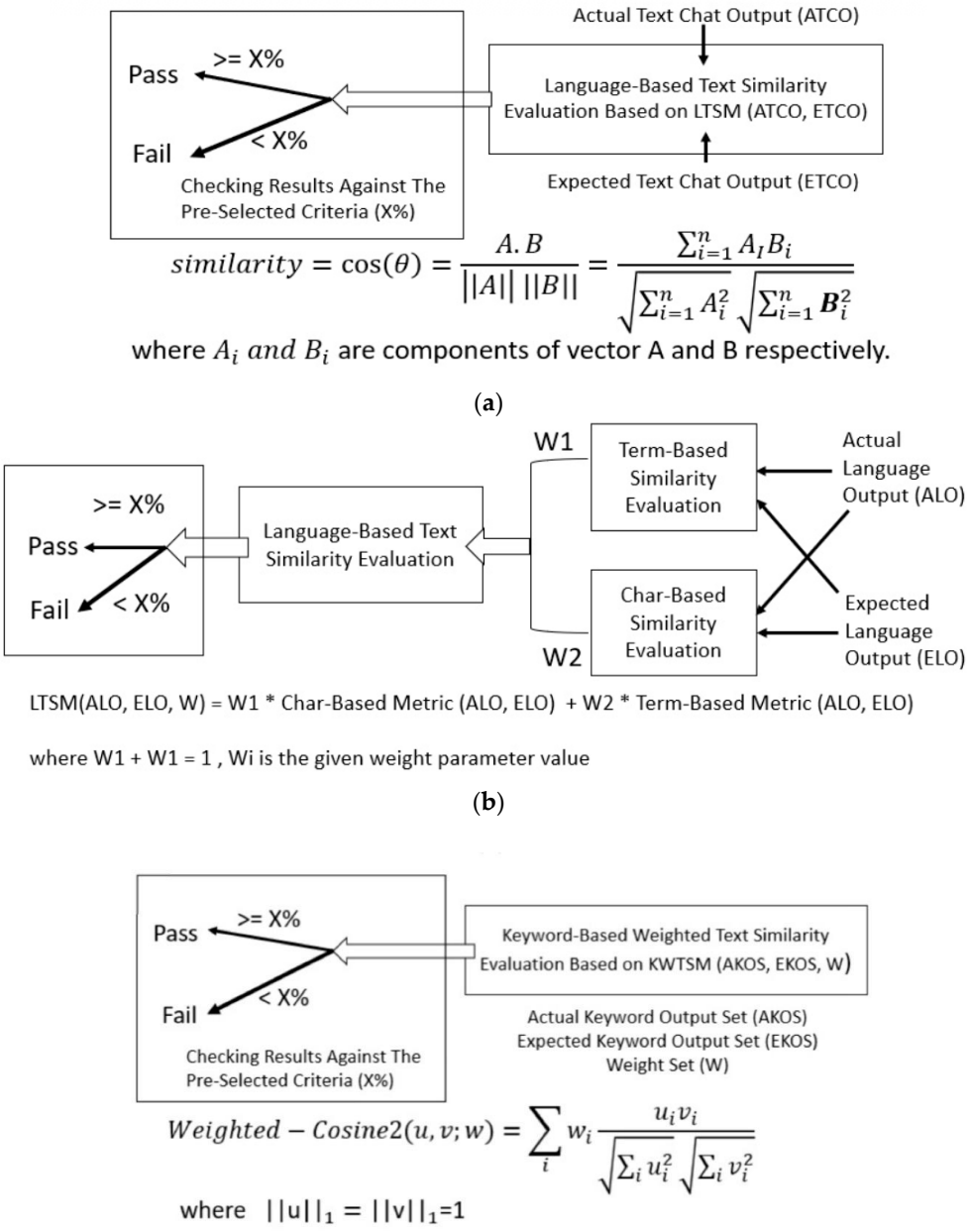


Figure 16. Text similarity evaluation. Reprinted from Ref. [27]. (**a**) Language-based text similarity evaluation. (**b**) Integrated language-based text similarity evaluation. (**c**) Keyword-based weighted text similarity evaluation.

Table 2. Minimum ICT coverage and total theoretical test space.

Chapter	Chapter-Topic Leaf Nodes (ICT)	Format Leaf Nodes (ICT)	Minimum ICT Tests	CCT Combinations	Theoretical Tests per Chapter (ICT × CCT)
Ch. 2	13	2	26	36	936
Ch. 4	9	2	18	36	648

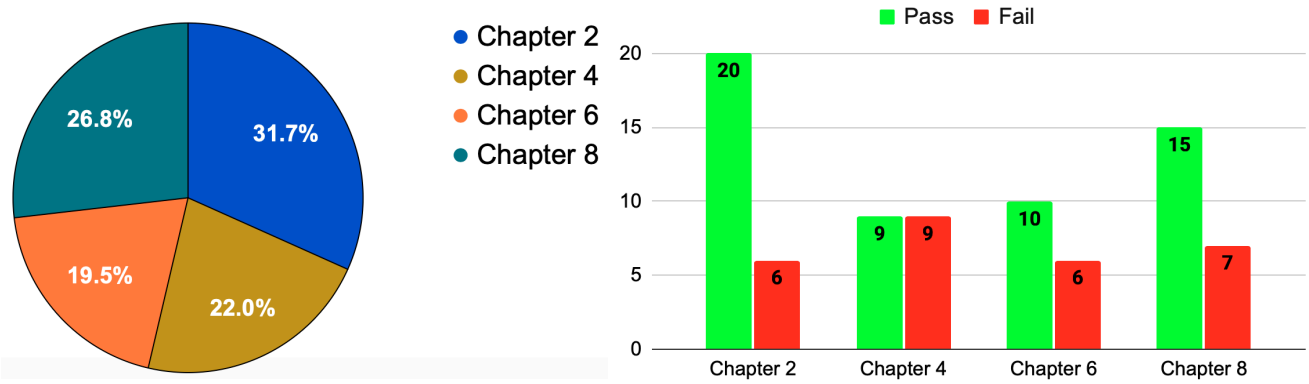
Table 2. *Cont.*

Chapter	Chapter-Topic Leaf Nodes (ICT)	Format Leaf Nodes (ICT)	Minimum ICT Tests	CCT Combinations	Theoretical Tests per Chapter (ICT × CCT)
Ch. 6	8	2	16	36	576
Ch. 8	11	2	22	36	792
Total	41		82		2952

The executed manual suite consists of 82 unaugmented test cases distributed across chapters as follows: 26 (Ch. 2), 18 (Ch. 4), 16 (Ch. 6), and 22 (Ch. 8). Chapter-wise testing outcomes are reported in Table 3, which shows 54 passes and 28 failures, corresponding to an overall pass rate of 65.9%. Figure 17a visualizes the distribution of executed test cases by chapter, while Figure 17b reports the corresponding pass–fail outcomes. Together, these results characterize baseline testing behavior under sampled context conditions and motivate the scaled automated testing in Section 6.3.

Table 3. Chapter-wise results (manual testing).

Chapter	Specified Test (Unaugmented)	Actual Test	Pass	Fail	Pass Rate
Ch. 2	26	26	20	6	76.9%
Ch. 4	18	18	9	9	50.0%
Ch. 6	16	16	10	6	62.5%
Ch. 8	22	22	15	7	68.2%
Total	82	82	54	28	65.9%



(**a**) (**b**)

Figure 17. Manual evaluation set charts. (**a**) Test case distribution by chapter. (**b**) Pass–fail results by chapter.

6.3. Automated Testing Results

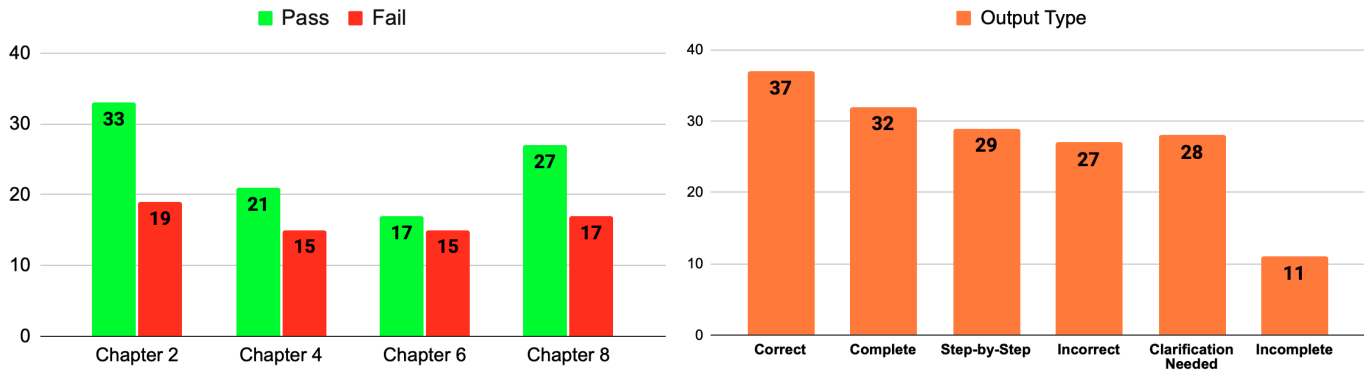
Automated testing extends the manual suite by scaling execution under the same ICT–CCT–OCT model while introducing controlled input perturbations to assess robustness to presentation variation. Starting from the unaugmented test cases in Section 6.2, the automated suite is generated by applying two spatial-level augmentations (randomly selected from the transformation set described in Section 5.2) to each original screenshot. These geometry-level transformations alter visual presentation while preserving the underlying algebra task.

The automated suite increases the total number of test cases from 82 to 164 and is validated using the same two-level protocol described in Section 6.1 (oracle correctness followed by OCT-aligned compliance). Chapter-wise automated testing results are reported in Table 4, showing 98 passes and 66 failures, for an overall pass rate of 59.8%. Relative to the manual baseline, the increased failure rate indicates that the end-to-end smart learning pipeline is sensitive to presentation-level perturbations.

In addition to aggregate pass–fail outcomes, automated testing reveals shifts in observed response categories. In particular, "clarification needed" outcomes become more frequent under augmented conditions, suggesting that spatial perturbations more often disrupt prompt interpretability prior to reasoning. Figure 18a reports pass–fail outcomes by chapter, while Figure 18b summarizes the distribution of output types across all chapters.

Table 4. Chapter-wise results (automated testing).

Chapter	Specified Tests (Augmented)	Actual Tested	Pass	Fail	Pass Rate
Ch. 2	52	52	33	19	63.5%
Ch. 4	36	36	21	15	58.3%
Ch. 6	32	32	17	15	53.1%
Ch. 8	44	44	27	17	61.4%
Total	164	164	98	66	59.8%



(**a**) (**b**)

Figure 18. Automated evaluation set charts. (**a**) Pass–fail results by chapter. (**b**) Output type distribution across four chapters.

6.4. Failure Analysis and Bug Reporting

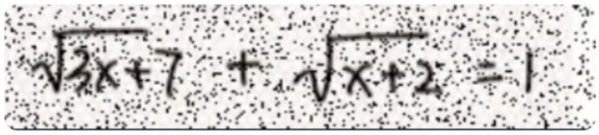
Failure analysis is conducted to ensure that invalid testing outcomes are interpretable and traceable to the modeled testing space. For each failed test case, the observed response is first assigned an OCT-invalid category and then examined alongside its corresponding ICT and CCT descriptors. This enables differentiation between failures consistent with (1) prompt interpretation instability (e.g., symbol or operator misreading under altered presentation conditions) and (2) downstream reasoning divergence following correct extraction of the problem statement.

To support reproducibility and diagnostic analysis, representative failures are documented using structured bug reports that pair the prompt condition, expected output, observed output, and an explanation of the likely failure mechanism. Figure 19 presents example bug reports from Ch. 2 and Ch. 4, illustrating how failures under augmented prompts can arise from misinterpretation of transformed images, leading to incorrect out-

comes despite otherwise coherent reasoning. These artifacts provide concrete evidence for targeted retesting and refinement of testing conditions in smart learning applications.

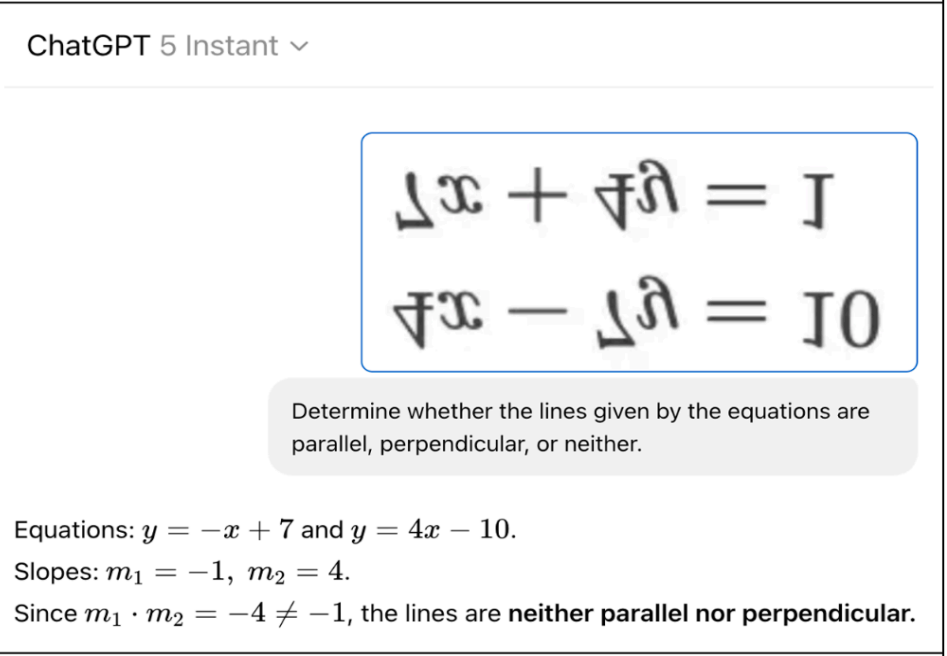
Based on the observed failure patterns, we recommend that users and developers of smart learning applications adopt a structured approach to failure handling. In particular, distinguishing between perception-related errors and reasoning-related errors can support targeted mitigation strategies. Improving input quality, conducting robustness testing across varied conditions, and avoiding reliance on single-run outputs are critical to reducing the impact of incorrect AI-generated responses in educational settings. Structured testing frameworks can further support systematic identification and resolution of such failures.

These examples illustrate how failures arise under different ICT–CCT conditions, demonstrating both perception-related errors due to input distortion and reasoning-related errors despite correct interpretation. The examples confirm that structured failure classification enables clearer diagnosis of system weaknesses and supports targeted improvements in both input handling and reasoning robustness.

Test Description	Testing computational problem from Ch. 2 Equations and Inequalities
Steps to Reproduce	Step-1: Open ChatGPT application; Step-2: Select GPT-5 model; Step-3: Type in prompt "Solve the equation: "; Step-4: Upload the problem image; Step-5: Click the send button.
Attached Proof	ChatGPT 5 Instant  Solve the equation: $\sqrt{3x+7}+\sqrt{x+3}=1$ Isolate one root: $\sqrt{3x+7}=1-\sqrt{x+3}$ Square: $3x+7=1-2\sqrt{x+3}+x+3$ Simplify: $2x+6=-2\sqrt{x+3} \Rightarrow x+3=-\sqrt{x+3}$ Square again: $(x+3)^2=x+3 \Rightarrow x^2+5x+6=0 \Rightarrow (x+2)(x+3)=0$ Check: $x=-2 \Rightarrow LHS=\sqrt{1}+1=2\neq 1;\ x=-3 \Rightarrow LHS=1+0=1$ ✅ $\boxed{x=-3}$
Bug Analysis	ChatGPT (GPT-5) mis-interpreted the equation from the image, therefore, the evaluation result is incorrect.
Expected Output	$x = -2$
Actual Output	No real solution
Test Result	Fail

(**a**)

Figure 19. *Cont.*

Test Description	Testing computational problem from Ch. 4 Linear Functions
Steps to Reproduce	Step-1: Open ChatGPT application; Step-2: Select GPT-5 model; Step-3: Type in prompt "Determine whether the lines given by the equations are parallel, perpendicular, or neither."; Step-4: Upload the problem image; Step-5: Click the send button.
Attached Proof	
Bug Analysis	ChatGPT (GPT-5) mis-interpreted the equations from the augmented image, therefore, the evaluation result is incorrect.
Expected Output	Perpendicular
Actual Output	Neither
Test Result	Fail

(**b**)

Figure 19. Sample bug reports. (**a**) Sample from Ch. 2. (**b**) Sample from Ch. 4.

7. Discussion

7.1. Rationale for a Three-Dimensional Test Model

This paper shows that the input representation and contextual conditions, rather than problem complexity, are the key determinants of an AI system's performance in smart learning applications. Although the manual testing pass rate was 65.9%, performance declined when conditions were augmented, indicating the sensitivity of AI systems to variations in presentation that are typically seen in real-life learning situations.

The findings indicate that there are two main categories of failure: (1) errors in the interpretation of the input due to the difference in images and formatting and (2) errors in judgment in the case of adequate cognition of the problem. This difference provides valuable diagnostic information that cannot be obtained with conventional accuracy-based assessment methods. The proposed 3D AI testing framework can systematically explore these behaviors by organizing the testing space along the three dimensions of input, context, and output. In contrast to other standard assessment tools based on predetermined datasets, this one can facilitate coverage-based testing, reproducibility, and failure analysis that can be traced, which is more appropriate for validating AI-driven educational systems.

Though the framework is model-based, it demonstrates that these methods are tailored to AI-specific features such as non-determinism, oracle ambiguity, and sensitivity to input conditions. Controlled data augmentation also enables enhanced robustness testing by modeling realistic changes in learning environments. In general, the results indicate that well-organized AI testing systems are necessary to achieve reliability and credibility in smart learning applications, especially when the system's output can directly influence student comprehension.

Unlike traditional classification-tree or model-based testing approaches, the proposed 3D AI testing framework introduces four key novelties:

1. Integration of AI-specific dimensions: Combines input diversity, contextual variability, and output intelligence quality into a unified testing space.
2. Explicit handling of oracle ambiguity: Uses similarity-based validation and structured output classification (OCT), which is not addressed in traditional models.
3. Application to educational Q&A systems: Tailors model-based testing to learning-oriented AI systems, where reasoning quality and explanation structure are critical.
4. Support for automated augmentation-based robustness testing: Incorporates controlled input perturbations to evaluate perception-to-reasoning pipelines.

Therefore, the contribution extends beyond repackaging by adapting and enhancing model-based testing for AI-driven educational applications.

7.2. Ethical Considerations in AI-Based Smart Learning

The assessment of AI-based smart learning apps raises several ethical concerns. First, the issues of fairness and bias should be discussed, as AI systems can yield unequal results across different input representations or situations. Second, in the educational context, transparency and explainability are essential, since students depend on AI-generated explanations to learn. Third, excessive reliance on AI systems can undermine students' independent problem-solving unless an appropriate balance is established. Lastly, there is the issue of data privacy dealing with student-generated inputs, especially in actual implementations. Ethical risks in this research are controlled by accessing publicly available educational data, not using personal data, and targeting system-level assessment rather than profiling individual users.

7.3. Threats to Validity

In this study, several threats to validity are considered:

1. Internal validity: Mislabeling the results or determining the accuracy of the results can affect the outcomes. Mitigation: Structured validation criteria and regular OCT classification.
2. External validity: The case study focuses on college algebra problems, which may limit generalizability. Mitigation: The framework should be domain-independent.
3. Construct validity: The correctness and structure of the explanation approximate the quality of intelligence, which is not necessary to describe all aspects of learning.
4. Conclusion validity: A small sample size (82 manual, 164 automated tests) may affect statistical power. Mitigation: Systematic coverage by ICTCCT modeling.

These constraints outline the prospects for future research with larger datasets and expanded learning areas.

8. Conclusions

This paper investigated AI testing for smart learning applications using a model-based three-dimensional (3D) testing framework defined by input, context, and output classification trees (ICT, CCT, and OCT). ChatGPT (GPT-5) solving screenshot-based college algebra

problems was used as the case study to demonstrate how the proposed framework can be instantiated and applied in a realistic smart learning scenario. By partitioning the input domain as chapter topic + format, modeling presentation conditions that affect prompt interpretation, and standardizing outcome labeling through a shared OCT, the framework supports systematic test selection, repeatable validation, and traceable failure diagnosis.

Empirical testing outcomes from both manual and automated execution indicate that robustness is not uniform across the modeled testing space. Manual testing establishes a baseline with unaugment prompts and sampled context conditions, while automated testing shows that spatial-level augmentation within the same ICT–CCT structure can systematically stress the perception-to-reasoning pipeline and reveal sensitivity to presentation variation. Failure analysis further demonstrates that invalid outcomes arise from distinct mechanisms, including upstream prompt interpretation errors and downstream reasoning divergence, highlighting the value of mapping failures back to their ICT–CCT conditions for targeted retesting and robustness improvement. This paper demonstrates that model-based AI testing provides a practical, structured foundation for evaluating smart learning applications, enabling coverage-aware analysis and diagnostic insight beyond ad hoc evaluation (e.g., manual evaluation or fixed datasets). The proposed approach is generalizable to other AI-driven learning tasks and provides a basis for future research on systematic testing and validation of intelligence quality for Q&A-based smart learning applications.

In the future, the work will extend this framework to multiple AI models and broader educational domains, incorporate statistical evaluation of testing outcomes, and explore integration with real-world smart learning platforms to assess pedagogical effectiveness.

Author Contributions: T.L.: writing—original draft preparation, case study, investigation, formal analysis; Q.T.N.: data curation, case study, validation, writing—review and editing; J.G.: conceptualization, methodology, formal analysis, resources, supervision, review, and administration; R.A.: drafting, formal analysis, and review. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting the findings of this study include algebra problem screenshots derived from publicly available OpenStax [32] materials, along with derived test specifications, data augmentation configurations, and labeled testing outcomes generated during evaluation. The original source materials are openly accessible through OpenStax. Derived datasets and testing artifacts generated in this study are available from the corresponding author upon reasonable request for research and replication purposes.

Acknowledgments: The authors acknowledge the OpenStax initiative for providing openly licensed educational materials that enabled the construction of the evaluation corpus used in this study.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study, the collection, analysis, or interpretation of data, the writing of the manuscript or the decision to publish the results.

References

1. Market.Us. *Global Smart Learning Market By Component, By End User, Region and Companies-Industry Segment Outlook, Market Assessment, Competition Scenario, Trends and Forecast 2024–2033*; Report ID: 121838; Market.Us: New York, NY, USA, 2024; pp. 1–231. Available online: https://market.us/report/smart-learning-market (accessed on 28 January 2026).
2. Riza, A.N.I.; Hidayah, I.; Santosa, P.I. Use of Chatbots in E-Learning Context: A Systematic Review. In *Proceedings of the 2023 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 7–10 June 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 0184–0188.

3. Colace, F.; De Santo, M.; Lombardi, M.; Pascale, F.; Pietrosanto, A.; Lemma, S. Chatbot for E-Learning: A Case of Study. *Int. J. Mech. Eng. Robot. Res.* **2018**, *7*, 528–533. [CrossRef]
4. Dimitriadou, E.; Lanitis, A. A critical evaluation, challenges, and future perspectives of using artificial intelligence and emerging technologies in smart classrooms. *Smart Learn. Environ.* **2023**, *10*, 12. [CrossRef]
5. Alnaqbi, A.M.A.; Yassin, A.M. Evaluation of Success Factors in Adopting Artificial Intelligence in E-Learning Environment. *Int. J. Sustain. Constr. Eng. Technol.* **2021**, *12*, 362–369. [CrossRef]
6. Durelli, V.H.S.; Durelli, R.S.; Borges, S.S.; Endo, A.T.; Eler, M.M.; Dias, D.R.C.; Guimarães, M.P. Machine Learning Applied to Software Testing: A Systematic Mapping Study. *IEEE Trans. Reliab.* **2019**, *68*, 1189–1212. [CrossRef]
7. Marijan, D.; Gotlieb, A. Software Testing for Machine Learning. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020*; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 13576–13582.
8. Sajjad, O.; Rehman, W.U.; Numan, M.; Sajjad, Z. Testing Chatbot Systems using Agentic AI Approach. *Int. J. Innov. Sci. Technol.* **2025**, *7*, 1826–1841. [CrossRef]
9. Jawalkar, S.K. Testing AI-Powered Applications: Challenges and Strategies. *Int. J. Innov. Res. Eng. Manag. Pharm. Sci. (IJIRMPS)* **2023**, *11*, 1–8.
10. Bayrı, V.; Demirel, E. AI-Powered Software Testing: The Impact of Large Language Models on Testing Methodologies. In *Proceedings of the 2023 4th International Informatics and Software Engineering Conference (IISEC), Istanbul, Türkiye, 1–2 November 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–4.
11. Zhu, H.; Liu, D.; Bayley, I.; Harrison, R.; Cuzzolin, F. Datamorphic Testing: A Methodology for Testing AI Applications. *arXiv* **2019**, arXiv:1912.04900. [CrossRef]
12. Felderer, M.; Ramler, R. Quality Assurance for AI-Based Systems: Overview and Challenges. *arXiv* **2021**, arXiv:2107.12190. [CrossRef]
13. Tao, C.; Gao, J.; Wang, T. Testing and Quality Validation for AI Software: Perspectives, Issues, and Practices. *IEEE Access* **2019**, *7*, 120164–120175. [CrossRef]
14. Gao, J.; Tao, C.; Jie, D.; Lu, S. What Is AI Software Testing and Why? In *Proceedings of the 2019 IEEE International Conference on Service-Oriented System Engineering (SOSE), San Francisco, CA, USA, 4–9 April 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 27–2709.
15. Masserini, E. Multi-Level Testing of Conversational AI Systems. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering (ICSE-Companion), Rio de Janeiro, Brazil, 12–18 April 2026*; IEEE: Piscataway, NJ, USA, 2026; pp. 1–3.
16. Ginsbourg, S. Testing AI-Based Software Systems—From Theory to Practice. In *Proceedings of the QA&TEST Embedded 2025, Bilbao, Spain*; SQS: Cologne, Germany, 2025; pp. 1–15.
17. Aleti, A. Software Testing of Generative AI Systems: Challenges and Opportunities. In *Proceedings of the 2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE), Melbourne, Australia, 14–15 May 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 4–14.
18. Gao, J.; Agarwal, R.; Vardam, R.; Narang, J. Form-Based Test Modeling, Analysis, and Complexity Evaluation for Smart OCR Apps. *Preprints* **2026**, 2026011280. [CrossRef]
19. Gao, J.; Patil, P.H.; Lu, S.; Cao, D.; Tao, C. Model-Based Test Modeling and Automation Tool for Intelligent Mobile Apps. In *Proceedings of the 2021 IEEE International Conference on Service-Oriented System Engineering (SOSE), Oxford, UK, 23–26 August 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–10.
20. Wei, X. Evaluating chatGPT-4 and chatGPT-4o: Performance insights from NAEP mathematics problem solving. *Front. Educ.* **2024**, *9*, 1452570. [CrossRef]
21. Hidayatullah, E.; Untari, R.; Fifardin, F. Effectiveness of AI in solving math problems at the secondary school level: A comparative study with student performance. *Union J. Ilm. Pendidik. Mat.* **2024**, *12*, 350–360. [CrossRef]
22. Vidal, J. Evaluation of the Performance of State-of-the-Art Large Language Models (LLMs) in Solving Math Word Problems. *SSRN* **2024**. [CrossRef]
23. Gandolfi, A. GPT-4 in Education: Evaluating Aptness, Reliability, and Loss of Coherence in Solving Calculus Problems and Grading Submissions. *Int. J. Artif. Intell. Educ.* **2025**, *35*, 367–397. [CrossRef]
24. Spreitzer, C.; Straser, O.; Zehetmeier, S.; Maaß, K. Mathematical Modelling Abilities of Artificial Intelligence Tools: The Case of ChatGPT. *Educ. Sci.* **2024**, *14*, 698. [CrossRef]
25. Gao, J.; Agarwal, R.; Garsole, P. AI Testing for Intelligent Chatbots—A Case Study. *Software* **2025**, *4*, 12. [CrossRef]
26. Li, G. E-Learning Intelligence Model with Artificial Intelligence to Improve Learning Performance of Students. *J. Comput. Allied Intell.* **2023**, *1*, 14–26. [CrossRef]
27. Hmoud, M.; Swaity, H.; Anjass, E.; Aguaded-Ramírez, E.M. Rubric Development and Validation for Assessing Tasks' Solving via AI Chatbots. *Electron. J. e-Learn.* **2024**, *22*, 1–17. [CrossRef]
28. Plevris, V.; Papazafeiropoulos, G.; Jiménez Rios, A. 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]

29. Lima, R.; da Cruz, A.M.R.; Ribeiro, J. Artificial Intelligence Applied to Software Testing: A Literature Review. In *Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), Seville, Spain, 24–27 June 2020*; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
30. Grochtmann, M.; Grimm, K. Classification Trees for Partition Testing. *Softw. Test. Verif. Reliab.* **1993**, *3*, 63–82. [CrossRef]
31. Gao, J.; Agarwal, R. AI Test Modeling for Computer Vision System—A Case Study. *Computers* **2025**, *14*, 396. [CrossRef]
32. Abramson, J. *College Algebra 2e*; OpenStax: Houston, TX, USA, 2021. Available online: https://openstax.org/details/books/college-algebra-2e?Book%20details (accessed on 28 January 2026).
33. Utting, M.; Legeard, B. *Practical Model-Based Testing: A Tools Approach*; Springer: Berlin, Germany, 2010.
34. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. Available online: https://github.com/albumentations-team/AlbumentationsX?tab=readme-ov-file (accessed on 28 January 2026). [CrossRef]