ChatGPT

Research on Tools and Methods for Automated Assessment and Error Diagnosis in Handwritten Linear-Algebra Assignments (August 2025)

1. Introduction

Grading handwritten solutions in introductory linear-algebra courses requires recognising mathematical expressions, understanding the student's reasoning, comparing each step against expected methods, and giving constructive feedback. Modern computer algebra systems (CAS) can check whether an answer is algebraically equivalent to the correct one, but they do not examine the *reasoning process* or locate errors. Recent advances in vision-language models (VLMs), large language models (LLMs), and interactive theorem provers promise to improve automated grading, yet research shows that current models still struggle with handwritten mathematics. This report summarises existing tools and datasets, research on automated grading and error diagnosis, and developments in formal proof verification that could inform an application for marking handwritten linear-algebra solutions.

2. Tools for handwriting recognition and automated grading

Tool	Purpose & Key features	Evidence/limitations
Mathpix Snip	Commercial OCR service that converts handwritten/printed mathematical expressions into LaTeX. Widely used to pre-process student work before feeding it into another grading system.	In a study on GPT-4 grading of handwritten math solutions, Mathpix and GPT-4V were used for OCR because sequence-to-sequence HME-recognition models struggle with complex two-dimensional notation [1]. Mathpix is a digitisation tool only – it does not grade or locate errors [2].
Graded.Pro	Paid AI grading platform for middle- and high-school mathematics that accepts scanned handwritten or typed work. Supports step-by-step solutions, equations, graphs and word problems. Teachers upload photos, and the system provides AI-generated scores and voice/text annotations; teachers can override scores.	The platform advertises support for scanned handwritten work and rubrics, and emphasises teacher control [3]. Evidence of accuracy is marketing-based; the system does not claim to understand proofs or abstract reasoning.

Tool	Purpose & Key features	Evidence/limitations
Gradescope (Turnitin)	Widely adopted tool in universities for uploading scanned exams and grading via a dynamic rubric. Uses optical character recognition and AI to group similar answers so that instructors can grade one representative and propagate the score. Supports written responses, mathematical equations, diagrams and code.	The University of Florida notes that Gradescope "uses OCR and AI assistance to group similar student responses," accelerating grading of free-response questions and mathematical equations [4]. Favourite features include automatic grouping of similar answers, dynamic rubrics and analytics [5]. Gradescope does not automatically detect logical errors—it clusters and scores answers based on similarity.
Recursive AI Grading Assistant	Custom AI grading system (Japan, 2024) for elementary-school maths. Students upload photos of workbook pages; the assistant recognises handwritten numbers, computes the correct answer, applies government scoring guidelines and returns a score.	The case study describes converting handwritten input to a machine-readable format, computing the correct solution, and providing instant scores [6]. The system was trained on a large, labelled dataset to ensure high accuracy and strict adherence to scoring formats [7]. It demonstrates that accurate grading is feasible when problems involve basic arithmetic and fixed formats.
Virtual AI Teacher (VATE)	System used by Squirrel AI to analyse student drafts and maintain an "error pool." It engages in multi-round dialogue to point out mistakes and suggest corrections.	According to the preprint, VATE achieved **78.3 % accuracy** in diagnosing elementary-level mathematical errors and was positively received by students [8]. It shows that combining error tracking with conversational feedback can improve learning.
ScribeSense / Quick Comments / Conker AI	Commercial products for automated grading of multiple-choice or short-answer mathematics homework. They can scan answer sheets and apply grading rules quickly.	Public descriptions emphasise time savings but provide few technical details. They are designed for basic calculations or multiple-choice formats and are not suited for long proofs.

Observations

1. **OCR remains a bottleneck for complex notation.** Handwritten-mathematical-expression recognition is difficult because of ambiguous two-dimensional notation. Sequence-to-sequence models such as WAP and TAP perform poorly on complex expressions, so commercial tools like Mathpix or GPT-4V are used to extract LaTeX [1].
2. **Current grading platforms prioritise efficiency over deep error analysis.** Tools like Graded.Pro and Gradescope group similar answers and apply rubrics but do not detect conceptual or logical

errors. They are appropriate for final-answer checking or structured tasks but not for assessing the reasoning process.

3. **Specialised systems for elementary arithmetic show that AI grading can be reliable when tasks are well-defined.** Recursive's AI assistant achieved fast and accurate scoring for basic operations [6]. Extending this to linear algebra, where solutions involve vector proofs and multi-step reasoning, is far more challenging.

3. Benchmarks and datasets for evaluating automated mathematics grading

Dataset/Benchmark	Description & Purpose	Key findings
Fermat benchmark (2025) [9]	Contains ~2,200 handwritten solutions with synthetic perturbations across four error dimensions (computational, conceptual, notational, presentation). Designed to test VLMs on error detection, localization and correction.	Nine VLMs were evaluated; the best (Gemini-1.5-Pro) corrected only **77 %** of errors. Models performed significantly better when the handwritten input was replaced with printed text, highlighting difficulties with handwriting [9].
CHECK-MAT (2024–25) [10]	122 scanned handwritten solutions from Russia's national high-school math exam with expert grades and a detailed rubric. Evaluates seven VLMs on error identification and alignment with the rubric.	The benchmark emphasises understanding student solutions rather than just checking the final answer. Early automated grading systems using CAS (STACK, LON-CAPA) verified algebraic equivalence but could not assess reasoning [11].
MathCCS (Mathematical Classification and Constructive Suggestions) dataset (2025) [12]	Contains real problems and student solutions with expert-annotated error categories and longitudinal data. Designed for multi-agent systems that classify errors and provide tailored feedback.	State-of-the-art models (Qwen2-VL, LLaVA-OV, Claude-3.5-Sonnet, GPT-4o) achieved < 30 % classification accuracy, and suggestions were low quality. The authors propose sequential error analysis and a multi-agent framework to improve performance [12].
Middle-School Algebra Misconceptions benchmark (2024) [13]	Captures 55 algebra misconceptions and 220 diagnostic examples. Evaluated with GPT-4 constrained by topic and teacher feedback.	Achieved precision/recall up to **83.9 %**, but performance varied across topics. Teachers considered the dataset valuable and emphasised the need for human expertise [13].

Dataset/Benchmark	Description & Purpose	Key findings
MathLP/MLP (2015) [14]	Framework that clusters student solutions into correct/partial/ incorrect groups using features from the solution. Allows minimal instructor grading and identifies error locations when a multistep solution deviates from the correct cluster.	Demonstrated that data-driven clustering can reduce instructor workload and provide targeted feedback, but requires typed solutions and cannot handle complex proofs.
FATE-M (Formal Algebra Theorem Evaluation—Medium) (2025) [15]	141 undergraduate-level abstract-algebra problems formalised in Lean4, extracted from 12 textbooks. Problems are paired with Lean proofs and natural-language comments. Used to evaluate formal theorem provers.	Provides the first Lean benchmark targeting college-level algebra. When evaluated with REAL-Prover (retrieval-augmented Lean prover), the best success rate (Pass@64) was **56.7 %**, surpassing other 7B-parameter models [16].
PeanoBench (2025) [17]	371 Peano-arithmetic proofs derived from the Natural Number Game. Each natural-language proof step is paired with the corresponding Lean tactic. Used to evaluate the LeanTutor system.	LeanTutor's autoformalizer correctly formalised 57 % of tactics in correct proofs and identified the incorrect step in 30 % of incorrect proofs [17].
Probability exam dataset for GPT-4o grading (2024) [18]	Real handwritten responses from 18 students' probability exams with rubrics. Used to evaluate GPT-4o's ability to grade using scanned images and rubrics.	When provided the correct answer and rubric, GPT-4o achieved the best alignment with human grades (mean absolute error 0.0766). However, scores were still off by ~7.7 % on average, showing substantial room for improvement [19].

Insights from benchmarks

- **Handwriting is a major challenge for VLMs and LLMs.** The Fermat benchmark shows that VLMs detect and correct errors better with printed text than with handwriting [9]. GPT-4o's performance on a probability exam similarly improved when given the rubric and correct answer but still lagged behind human graders [19].
- **Accurate error diagnosis requires more than answer verification.** Datasets like CHECK-MAT and MathCCS highlight the importance of analysing the solution process and aligning feedback with a rubric [10] [12]. Simple clustering or CAS equivalence checks are insufficient for proofs and complex reasoning.
- **Formal-theorem benchmarks (FATE-M, PeanoBench) bridge the gap between human and machine reasoning.** These datasets formalise problems in Lean and provide proof scripts, enabling evaluation of automated theorem provers and autoformalization pipelines.

4. Research on automated grading and error diagnosis

4.1 Vision- and language-model based grading

- **Fermat, CHECK-MAT and MathCCS** (see above) show that current VLMs are far from human performance at error detection, localization and correction. Models such as Gemini-1.5-Pro achieved ~77 % correction on synthetic errors [9], while classification accuracy in MathCCS remained below 30 % [12].
- **AI-assisted grading of handwritten university-level maths exams (ETH Zurich, 2024)** used GPT-4 combined with Mathpix to grade semi-open responses. Results showed that GPT-4 gave reliable initial grades but still required human verification; improvements in OCR and grading rules were recommended [20]. The background section emphasises that HME recognition (using WAP/TAP or Mathpix) is still challenging [1].
- **GPT-4o for grading probability-theory exams (2024)**: providing the rubric and correct answer improved alignment (MAE $\approx$ 0.0766), but the model still over- or under-estimated scores depending on context [19]. Thus, rubric-aware prompting helps but does not fully solve the problem.
- **Middle-School Algebra Misconceptions study (2024)**: GPT-4, when constrained by topic and guided by educator feedback, achieved up to 83.9 % precision/recall in diagnosing misconceptions [13]. Educator involvement and topic constraints were key to high performance.
- **VATE (Virtual AI Teacher)**: A multi-round dialogue system that maintains an error pool and guides students to self-correct. Achieved 78.3 % accuracy in elementary maths [8], indicating that conversational feedback can complement grading systems.

4.2 Data-driven methods and CAS-based grading

- **Mathematical Language Processing (MLP)**: a 2015 framework that converts open-response solutions into feature vectors, clusters them into correctness categories and requires minimal instructor grading [14]. Tracking when a multistep solution deviates from the correct cluster allows error localization.
- **STACK and other CAS-based systems (MapleTA, LON-CAPA)**: open-source systems for online mathematics assessment. STACK evaluates answers by algebraic equivalence using the Maxima CAS and provides specific feedback while separating validation from assessment [21]. These systems are effective for algebraic manipulation but cannot grade reasoning or proofs [11].

4.3 Formal proof verification and theorem proving

Proof assistants and educational tools

- **Lean proof assistant**: an interactive theorem prover and programming language used to formalise mathematics. Lean's proofs are rigorous because the kernel checks every inference; users can script proofs using tactics and can rely on an extensive library (mathlib). The Duke University "linear algebra game" demonstrates using Lean as a teaching aid: students write formal proofs in Lean to solve 64 linear-algebra puzzles, thereby learning the logical structure of proofs [22]. The project contributed new definitions and lemmas to mathlib, including orthogonal complements of subspaces [23].
- **Teaching mathematics with Lean**: a pilot study at the University of Zurich taught freshman students logic foundations using Lean and compared their performance to students in a traditional course. The paper emphasises that Lean acts as a bridge between automated theorem provers and

proof assistants, offering mathematician-friendly syntax and a supportive community [24]. Teaching with Lean requires careful design and often yields positive learning effects [25].

- **Lean educational ecosystem**: The Lean project offers interactive games (Natural Number Game), textbooks, university courses and tutorials [26]. The Lean Focused Research Organisation envisions children using Lean as a playground for learning mathematics, where they receive instantaneous feedback similar to coding [26].
- **LeanTutor (2025)**: an LLM-based tutoring system that interacts with students in natural language, autoformalises their proofs in Lean, verifies correctness, generates next tactics and provides hints [17]. Its autoformalizer correctly formalised 57 % of tactics in correct proofs and identified the incorrect step in 30 % of incorrect proofs [17]. The PeanoBench dataset accompanies the system.
- **Verbose Lean** and **Natural-Language Games**: educational libraries enabling students to write proofs in a controlled natural language and receive feedback. These tools have been used to teach set theory and algebra but require knowledge of Lean; accessible citations were limited.

Automated theorem proving with LLMs

- **DeepSeek-Prover (2024)**: generated a synthetic dataset of 8 million Lean proofs from high-school and undergraduate competition problems. After fine-tuning on this dataset, their 7B-parameter model achieved whole-proof accuracy of **46.3 %** with 64 samples (compared with GPT-4's 23.0 % on the same benchmark) and solved five problems from the FIMO benchmark, while GPT-4 solved none [27]. The method demonstrates that large synthetic datasets can substantially improve LLM-based theorem proving.
- **REAL-Prover (Retrieval-Augmented Lean Prover, 2025)**: integrates a fine-tuned LLM with a retrieval system (LeanSearch-PS) that selects relevant theorems from mathlib. REAL-Prover achieved a **23.7 %** success rate on the ProofNet benchmark and **56.7 %** on the FATE-M algebra benchmark [16]. The FATE-M dataset contains 141 undergraduate-level abstract-algebra problems formalised in Lean, each paired with Lean proofs and natural-language comments [15].
- **Lean and AI for proof checking**: An ACM article notes that large proofs like Fermat's Last Theorem are being formalised via Lean with help from volunteer communities. Lean's library now contains the equivalent of an undergraduate-level mathematics course [28]. AI models may assist by working on modular proof components and by learning from crowdsourced proof data [28].

4.4 Challenges and open problems

1. **Gap between handwritten solutions and formal proofs.** Converting students' handwritten reasoning into a formal representation remains unsolved. OCR errors, ambiguous notation and informal reasoning make autoformalization difficult [1]. LeanTutor's autoformalizer achieves only 57 % accuracy [17].
2. **Understanding reasoning vs. final answers.** Datasets like CHECK-MAT and MathCCS highlight the need to analyse the entire solution process. VLMs currently misclassify or miss errors, especially conceptual ones [10] [12].
3. **Generalisation to advanced mathematics.** Synthetic-data provers (DeepSeek-Prover) perform well on high-school competition problems but struggle with undergraduate algebra; retrieval-augmented models like REAL-Prover improve success rates but still fail on many problems [16].
4. **Reliance on human experts.** Even sophisticated LLM-based graders require rubrics and human oversight to achieve acceptable accuracy [19]. Teachers remain essential for verifying grades and providing nuanced feedback.

5. Implications for a linear-algebra grading application

- **Pipeline for recognising and evaluating handwritten proofs.** A practical system should combine high-quality OCR (e.g., Mathpix or GPT-4V) to convert handwriting into LaTeX, a parser to identify algebraic expressions and reasoning steps, and a grading engine that compares each step to expected methods. Existing tools like Graded.Pro or Gradescope can handle uploading, annotation and rubric-based scoring but do not understand logical reasoning.
- **Error classification using benchmark insights.** The Fermat benchmark's four error dimensions (computational, conceptual, notational, presentation) provide a useful taxonomy [9]. Training models to classify errors along these dimensions could improve feedback quality. However, VLMs still perform poorly on handwriting; thus, the system may need to encourage students to type solutions or provide clearer scanned images.
- **Incorporating proof assistants.** Lean can verify formal proofs of linear-algebra statements; the Duke "linear-algebra game" demonstrates that students can learn to write proofs in Lean [22]. For grading handwritten proofs, one could explore autoformalization techniques (e.g., LeanTutor) to translate student reasoning into Lean and then verify correctness. Current autoformalizers are only partially accurate [17], so manual correction or simplified formal language (e.g., controlled natural language) may be necessary.
- **Leveraging AI theorem provers for hints.** Retrieval-augmented provers like REAL-Prover and LLM-based provers trained on synthetic data show promise in generating next proof steps and could be used to suggest corrections or hints when a student's proof deviates from the expected path [16]. However, these models are not yet reliable enough to grade proofs autonomously.
- **Future research directions.** To build a robust grading assistant for linear algebra, research should focus on: (i) better OCR and LaTeX conversion for two-dimensional notation; (ii) datasets of real handwritten linear-algebra proofs with expert annotations; (iii) hybrid systems that combine CAS equivalence checking with reasoning-process analysis; and (iv) integration of interactive theorem provers and LLMs for autoformalization and feedback.

6. Conclusion

There is rapid progress in AI-assisted grading and automated theorem proving, but no existing tool fully addresses the challenge of grading handwritten linear-algebra proofs. Commercial platforms like Gradescope and Graded.Pro streamline the logistics of collecting and scoring submissions but rely on rubrics and do not analyse reasoning. Vision-language models can classify certain types of errors yet struggle with handwriting and conceptual understanding, as shown by benchmarks such as Fermat, CHECK-MAT and MathCCS. Formal proof assistants like Lean guarantee correctness but require formal input; research efforts such as LeanTutor, DeepSeek-Prover and REAL-Prover aim to bridge this gap. A successful application will likely need to combine these approaches—using high-quality OCR, error-classification models, CAS for algebraic checks, and proof assistants for verifying reasoning—while keeping human teachers in the loop to ensure fairness and provide pedagogically meaningful feedback.

---

[1] [20] AI-assisted Automated Short Answer Grading of Handwritten University Level Mathematics Exams
https://arxiv.org/html/2408.11728v1

[2] [3] AI Marking and Feedback Platform for Teachers | Graded Pro
https://graded.pro/pages/best-ai-grading-tool-for-math

4 5 Gradescope | e-Learning | University of Florida
https://elearning.ufl.edu/supported-services/gradescope/

6 7 Enhancing Education Efficiency with Automated Grading
https://recursiveai.co.jp/case-studies/enhancing-education-efficiency-with-automated-grading

8 AI-Driven Virtual Teacher for Enhanced Educational Efficiency: Leveraging Large Pretrain Models for Autonomous Error Analysis and Correction
https://arxiv.org/html/2409.09403v1

9 Can Vision-Language Models Evaluate Handwritten Math?
https://arxiv.org/html/2501.07244v2

10 11 CHECK-MAT: Checking Hand-Written Mathematical Answers for the Russian Unified State Exam
https://arxiv.org/html/2507.22958v1

12 [2502.13789] From Correctness to Comprehension: AI Agents for Personalized Error Diagnosis in Education
https://arxiv.org/abs/2502.13789

13 A Benchmark for Math Misconceptions: Bridging Gaps in Middle School Algebra with AI-Supported Instruction
https://arxiv.org/html/2412.03765v1

14 [1501.04346] Mathematical Language Processing: Automatic Grading and Feedback for Open Response Mathematical Questions
https://arxiv.org/abs/1501.04346

15 16 REAL-Prover: Retrieval Augmented Lean Prover for Mathematical Reasoning
https://arxiv.org/html/2505.20613v1

17 [2506.08321] LeanTutor: A Formally-Verified AI Tutor for Mathematical Proofs
https://arxiv.org/abs/2506.08321

18 19 Evaluating GPT-4 at Grading Handwritten Solutions in Math Exams
https://arxiv.org/html/2411.05231v1

21 STACK
https://stack-assessment.org/

22 23 Automated theorem proving and proof verification | Department of Mathematics
https://math.duke.edu/mathplus/2023/automated-theorem-proving-and-proof-verification

24 25 Teaching “Foundations of Mathematics” with the LEAN Theorem Prover
https://arxiv.org/html/2501.03352v2

26 How the Lean language brings math to coding and coding to math - Amazon Science
https://www.amazon.science/blog/how-the-lean-language-brings-math-to-coding-and-coding-to-math

27 DeepSeek-Prover: Advancing Theorem Proving in LLMs through Large-Scale Synthetic Data
https://arxiv.org/html/2405.14333v1

28 Feedback Loops Guide AI to Proof Checking – Communications of the ACM
https://cacm.acm.org/news/feedback-loops-guide-ai-to-proof-checking/