Evaluating LLM-Generated Formative Feedback for Undergraduate Mathematics Through the Lens of Feedback Theory

Aron Gohr[1] **Marie-Amelie Lawn**[2] **Kevin Gao**[2] **Stephen Heslip**[2] **Inigo Serjeant**[2]
[1]Independent Researcher [2]Imperial College London
aron.gohr@gmail.com m.lawn@imperial.ac.uk

Abstract

Large language models can generate feedback on free-form student writing, but there is still limited evidence on whether such feedback is mathematically correct, pedagogically useful, and aligned with expert judgement across varied undergraduate proof-writing tasks. We evaluate LLM-generated feedback on 65 undergraduate proof-writing exercises using Hattie and Timperley's feedback framework and a grade-agreement metric, comparing two models (GPT-4.1, GPT-5) under two workflow configurations graded by two independent LLM evaluators. GPT-5 produces higher-quality feedback across all dimensions. A mark-scheme-augmented workflow improves grade correlation with human experts for both models, and precomputed mark schemes allow instructors to audit the system before deployment. While the present work cannot supply evidence of benefit in downstream learning tasks, as no controlled trials suitable to show such were run, we do show automated detection and explanation of errors on a wide variety of typical undergraduate-level proof tasks that is aligned with expert judgement on the mathematical issues. However, providing meaningful self-regulation support and controlled studies with students remain to be done. These results show that feedback theory provides a useful lens for evaluating automated mathematical feedback.

1 Introduction

Proof writing is a core skill in mathematics, yet students entering university typically have little experience with it (Moore, 2016). Frequent, detailed formative feedback is important in the acquisition of this skill, but producing it at scale is labour-intensive and inconsistent across graders. Unlike most high-school mathematics tasks, undergraduate proof exercises require assessing the validity, relevance, and level-appropriateness of open-ended arguments. A student may give a correct solution that is unexpected to the grader; an incorrect argument may still deserve substantial credit if it contains the main ideas; a small computational error may invalidate the solution if the subsequent argument depends on it; and a mathematically correct proof may still be poor if it relies on machinery outside the course, takes large unnecessary detours, or is poorly written. Recently, large language models (LLMs) have achieved strong performance on mathematical reasoning benchmarks (OpenAI, 2024) and have been trialled as tutors (Jurenka et al., 2024; Miller and DiCerbo, 2024). It has also been shown that at least at a high computational budget, they can be used to grade complex proofs (Ma et al., 2025). Educational applications, however, must produce not only grades but *pedagogical feedback* while operating under tight budget and latency constraints. To our knowledge, LLM-generated feedback quality for free-form undergraduate proof exercises has not been previously studied.

We evaluate LLM-generated feedback on 65 undergraduate proof-writing exercises along four dimensions derived from the Hattie–Timperley feedback framework (Hattie and Timperley, 2007): correctness, task-level clarity, process-level guidance, and self-regulation support. We compare two models (GPT-4.1, GPT-5) under two workflow configurations (a direct baseline and a mark-scheme-augmented pipeline), graded by two independent LLM evaluators, and use grade agreement with human experts as a complementary screening metric. Code, prompts, and data are available at `https://github.com/agohr/llm_proof_feedback`.

Contributions.

1. We operationalise the Hattie–Timperley framework by deriving a rubric for automated evaluation of LLM-generated mathematical feedback.

2. We show that evaluation using this frame-

Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications, pages 813–818

work reveals quality differences invisible to grade agreement alone. For instance, grade agreement shows no advantage for GPT-5 over GPT-4.1, despite GPT-5 scoring substantially higher on all feedback dimensions.

3. We stress test our tutoring system in several ways: we show that LLMs can effectively find errors in exemplary solutions of our exercises, we analyse the behaviour of our workflows for questions designed to trick the models, we show that both graders prefer the GPT-5 output despite self-serving bias (Panickssery et al., 2024).

Related work. Intelligent tutoring systems have provided automated mathematical feedback for decades, from Andes (Gertner and VanLehn, 2000) to proof-assistant-based systems like Waterproof (Wemmenhove et al., 2022) and LeanTutor (Patel et al., 2026), but these cannot process natural free-form writing. LLM-based systems can (Miller and DiCerbo, 2024; Jurenka et al., 2024), but exhibit failure modes such as sycophancy and hallucination. LLM-as-judge approaches are widely used in the LLM capabilities literature (Zheng et al., 2023); we adopt a cross-grader design to control for self-preference bias (Panickssery et al., 2024).

Feedback theory. Hattie and Timperley (2007) proposed a model distinguishing four feedback levels: *task* (FT), *process* (FP), *self-regulation* (FR), and *self* (FS). Task-level feedback identifies what was correct or incorrect; process-level feedback addresses strategies and reasoning; self-regulation feedback prompts metacognitive monitoring and planning; self-level feedback ("you are smart") is generally ineffective. We operationalise this framework as a grading scheme for LLM-generated feedback.

2 Methods

Dataset. We develop a synthetic dataset consisting of 65 question–solution pairs from a first-year transition-to-proof course covering number systems, vector geometry, and calculus. Solutions were authored by experienced students familiar with the course and include deliberate errors. The solution authors were the student co-authors in their second or third year who had taken the course. Given a workload of 21 to 22 problems per student they had to work quickly, and were encouraged to write down partial arguments where they could not finish a solution, and, where natural, to include plausible errors that would be useful test cases. The dataset is therefore controlled and error-rich, but not claimed to represent the full distribution of real student submissions, which we expect will in any case not be uniform across deployments. Three expert markers (two of the authors, one mathematics PhD student) independently graded each item on a 0–5 rubric; grades were reconciled by manual review by one of the authors, who is a lecturer of the course.

Feedback Generation. We compare two feedback-generation models and two workflow configurations, yielding four conditions:

- **Models:** GPT-4.1 and GPT-5.
- **Workflows:**
  - *Baseline-concise*: the model receives the question and student answer with a 200-word limit and generates feedback directly.
  - *MS-w-example*: a multi-step pipeline that first specialises a generic mark scheme to the question, then generates feedback informed by the specialised mark scheme. This workflow was selected based on its strong performance in grade-agreement pre-evaluations.

GPT-5 and GPT-4.1 were selected for this experiment because they represent modern reasoning and non-reasoning language models, respectively. Baseline-concise represents a simple workflow that just asks the model to provide *short* feedback.[1]

Hattie–Timperley Evaluation. Each feedback item is scored on four dimensions, operationalised as rubrics with detailed level descriptors (0–5):

1. **Correctness (D1):** Mathematical accuracy of the feedback content.
2. **Task-level clarity (D2/FT):** Whether the feedback specifically identifies what was correct or incorrect, with reference to the exercise's goals.
3. **Process-level guidance (D3/FP):** Whether the feedback engages with the student's reasoning strategies and suggests alternatives.

[1]We found in early experiments that directly asking the models for feedback tended to result in feedback that in our judgment was clearly too long.

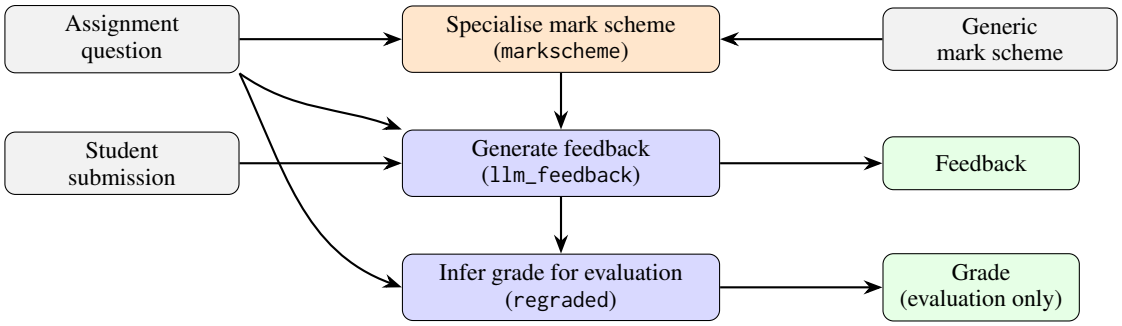

Figure 1: High-level view of the `MS-w-example` workflow. The mark-scheme step is precomputable because it depends on the assignment and generic rubric, but not on any student submission. In deployment, the system returns only feedback; the grade node is used for evaluation.

Role	Model/settings
Feedback generation	GPT-4.1; GPT-5 with medium reasoning. Defaults otherwise.
HT feedback evaluation	GPT-5.4 and GPT-4.1 as independent graders.
Grade screening	GPT-4.1-nano infers a grade from question and feedback only.
Workflow preselection	Earlier grade-correlation sweeps used GPT-4o workflows and GPT-4o grading.

Table 1: Compact model-role summary. Reasoning models do not expose a temperature parameter; non-reasoning calls used temperature zero. We note that OpenAI models do not behave deterministically even at temperature zero.

4. **Self-regulation support (D4/FR):** Whether the feedback prompts self-monitoring, identifies next steps, and attributes outcomes to controllable factors.

We distinguish two roles: *feedback generation* (the model producing feedback for students) and *meta-evaluation* (the model scoring that feedback). Two LLM meta-evaluators, GPT-5.4 and GPT-4.1, independently score all $4 \times 65 = 260$ feedback instances on all four dimensions. For instance, GPT-5, which acts as a feedback-generation model in one experimental condition, is thus evaluated by GPT-5.4 in its meta-evaluator role. Each meta-evaluator first produces a brief chain-of-thought justification and then assigns an integer score, yielding paired scores that support cross-grader reliability analysis. The justifications were retained for human inspection and are part of our code and data repository.

Grade Screening. As a complementary metric, we regrade each feedback instance using a lightweight model (GPT-4.1-nano) that infers a 0–5 grade from the question and feedback alone (student solution hidden), and compute Kendall τ with human grades. The use of a *lightweight* grader is motivated here by two points. First, feedback should be as unambiguous as possible. Second, a mathematically capable grader could correctly set a grade based on feedback that is completely incompetent but leaks enough information about the original submission.

Manually Checking a Sample To check whether GPT-5.4 was grading the model feedback items too generously, we manually graded all 65 question/answer/feedback triples from the MS-w-example workflow with GPT-4.1 as the feedback-generation model. On this sample, LLM ratings were on average *more critical* than human ratings across all four HT categories, except for the GPT-4.1 meta-rating on feedback correctness, which was slightly more optimistic than the corresponding human grade.

Qualitative tasks. We additionally ran two qualitative checks. First, we asked a solve-then-comment workflow to proofread all 65 exemplary solutions and the workflow indeed found several errors which we confirmed, with no hallucinations.[2] Second, we constructed five stress-test problems designed to expose mathematical or contextual failures, including overpowered methods, surprising elementary arguments, advanced material, and multilingual input.

Examples of Feedback generated. We illustrate our feedback generation system with a stress-test example, using the ms-w-example workflow and GPT-5 as the underlying language model. Table 2

[2]There were false positives, but these were due to the model missing course-specific context.

shows one of our stress tests on which the workflow generates correct and useful feedback. The main difficulty in this example is that when given the problem, the models we tested themselves tend to suggest solutions that have the same shortcomings as the student solution, and will only find elementary arguments if specifically instructed to do so.

3 Results

Grade correlation. Table 3 shows Kendall τ between regraded and human grades. The ms-w-example workflow with GPT-4.1 achieves $\tau = 0.561$, substantially exceeding the human inter-grader baseline from our two independent gradings of the dataset ($\tau = 0.387$). The same workflow consistently outperforms the *baseline_concise* workflow across both models.

When developing the workflows, we optimised for grade correlation under GPT-4o workflows with GPT-4o grading. The data we collected on the influence of model and grader choice on grade correlations are included in the GitHub repository at `https://github.com/agohr/llm_proof_feedback`.

GPT-5 feedback is substantially better under the HT rubric. Table 4 presents mean HT scores across all four conditions, averaged over both graders. GPT-5 outscores GPT-4.1 on every dimension, most dramatically on correctness (D1: 4.40–4.78 vs. 3.48–4.54) and process guidance (D3: 4.14–4.46 vs. 3.29–3.57). Obviously, the conclusions that can be drawn from this are weaker than conclusions from observed learning outcomes would be.

Self-regulation is hard to grade. Across all conditions and graders, self-regulation support clusters around low values. Indeed, under the GPT-5.4 grader, most items receive a score of exactly 3. We attribute this mainly to the fact that LLMs have difficulty separating this feedback dimension from the others; when manually producing meta-feedback, we found that we faced the same problem.

Cross-grader agreement and bias. Both graders rate GPT-5 feedback substantially higher than GPT-4.1 feedback across all dimensions. The consistency of the GPT-5 advantage across graders lends some support to the validity of the evaluation, as under self-serving bias (Panickssery et al., 2024) each model is expected to rate its own outputs higher.

Qualitative evaluation. We also evaluated our system qualitatively on the two additional tasks defined above.

The first involved proofreading the exemplary solutions to our exercises. We used a solve-then-comment workflow in which the model (o4-mini) first generated a solution to each problem and then evaluated the proposed exemplary solution. Except where the model was unaware of course-specific notation, conventions, or disallowed tools, the feedback was uniformly useful, surfacing several clerical and minor technical errors.

The second was a stress test: five problems that are mathematically accessible but constructed to challenge current LLMs. Performance differed substantially between non-reasoning and reasoning models. While even GPT-5 failed on 3/5 items, targeted hints substantially improved its performance. This suggests that lightweight teacher interventions can meaningfully improve model performance at considerably less effort than writing full question-specific rubrics. Full details are included in the code and data repository.

4 Discussion

Our results support three main findings relevant to automated feedback evaluation and design.

Precomputation of parts of a workflow can help prevent failures. In our stress tests, we were able to trace failure cases to errors in the model-generated mark schemes. Since this stage of feedback generation does not depend on student submissions, it is natural to precompute it at the beginning of a course. Instructors can thus audit system performance on course content before deployment and intervene with suitable hints.

Grade correlation is an insufficient but useful metric for workflow evaluation. On its own, correlation of grades derived from feedback generated by our system is insufficient to select a good workflow for a given application. Our work shows that at least for our undergraduate exercise setting, it fails to show any advantage of the use of reasoning models, when several other metrics we consider, as well as our stress testing, do show such advantage. We still regard grade correlation as a useful screening tool for the development of mathematics tutoring applications, however, as it allows the reuse of graded proof datasets such as (Frieder et al., 2023; Ma et al., 2025).

Question	Show that there are infinitely many numbers that are divisible by 6 and that can be written as the sum of two prime numbers. (Early undergraduate elementary number theory course.)
Student answer	By Dirichlet's theorem on primes in arithmetic progressions, there are infinitely many primes both of the form $6k+1$ and $6k+5$. Let S_5 and S_7 denote two such infinite sets. Then $p+q$ with $p \in S_5$ and $q \in S_7$ is always divisible by 6, and there are infinitely many such pairs; this yields the desired result.
Generated feedback	*What you did well:* You correctly identified the relevant residue classes (primes > 3 are $\equiv 1$ or $5 \pmod 6$), and $1+5 \equiv 0 \pmod 6$), and the overall pairing strategy is appropriate. *Main issues: (i)* You invoke Dirichlet's theorem without proof. For this course and problem, relying on such a deep result is not acceptable; an elementary argument is expected. *(ii)* Even if Dirichlet were allowed, "there are infinitely many such pairs" does not by itself rule out repeated sums—you need infinitely many *distinct* multiples of 6. *How to fix:* Replace Dirichlet with an elementary Euclid-style argument: assume finitely many primes $\equiv 5 \pmod 6$, form $N = 6\prod r_i - 1$, show $2 \nmid N$ and $3 \nmid N$, and deduce N has a new prime factor $q \equiv 5 \pmod 6$. Then fix $r = 7$ ($\equiv 1 \pmod 6$) and note the sums $q+7$ are distinct and divisible by 6. *Notation:* S_5 and S_7 do not match the stated sets of primes of the form $6k+1$ and $6k+5$; there is no residue class "7" mod 6.

Table 2: Stress-test example: GPT-5 with the ms-w-example workflow. The student invokes Dirichlet's theorem. Their argument is mathematically correct, but uses far heavier machinery than the course level allows, and introduces an ambiguous notation (S_5, S_7) mismatched with the stated residue classes. The feedback correctly flags all problems and sketches a possible elementary fix via a Euclid-like argument.

Model	Workflow	τ	n
GPT-4.1	Baseline	+0.306	65
GPT-4.1	MS-w-example	+0.561	65
GPT-5	Baseline	+0.136	65
GPT-5	MS-w-example	+0.296	65
Human inter-grader		+0.387	65

Table 3: Grade correlation: Kendall τ between grades regraded by GPT-4.1-nano from feedback alone and reconciled human grades. Human baseline shown for reference.

Model	Workflow	D1 Corr.	D2 Task	D3 Proc.	D4 Self-R.
GPT-5.4 grader					
GPT-4.1	Baseline	3.62	3.42	3.29	3.02
GPT-4.1	MS-w-ex.	3.48	3.71	3.40	2.88
GPT-5	Baseline	4.40	4.14	4.14	3.05
GPT-5	MS-w-ex.	4.63	4.29	4.25	3.06
GPT-4.1 grader					
GPT-4.1	Baseline	4.37	3.69	3.57	2.78
GPT-4.1	MS-w-ex.	4.54	3.89	3.46	2.32
GPT-5	Baseline	4.77	4.09	4.28	3.62
GPT-5	MS-w-ex.	4.78	4.25	4.46	3.51

Table 4: Mean HT scores (0–5) by condition and grader. D1–D4 correspond to correctness, task clarity, process guidance, and self-regulation.

Metacognitive feedback is still a gap. In our test cases, our system generally produces correct and task-level relevant feedback but qualitatively, we see that it often fails or does not attempt to diagnose *why* a certain mistake was made. This is also visible to some extent in the self-regulation scores produced by our evaluation, although that part of the evaluation probably suffered from difficulties our meta-grader had in distinguishing this dimension from the others.

We expect that closing this gap would require longer interaction histories and more agentic model scaffolds.

Limitations

Our dataset consists of 65 controlled, author-written items from a single institution and course. This allows useful comparisons to be drawn between configurations, but the frequency and variety of errors in authentic student work may be different. The HT evaluation is itself performed by LLMs, and only the GPT-4.1/MS-w-example feedback ratings were cross-checked by the authors. We evaluate only two feedback-generation models and two workflows in the main experiment, although our supplementary code and data contain grade-correlation and stress-testing data for additional combinations. Finally, we do not measure learning outcomes or student uptake of the feedback, which would be the gold-standard test for pedagogical effectiveness. While we deployed our tutoring system in several real courses using the lambda feedback platform at Imperial College, and received informal feedback on its use from students, no data from these deployments was used in this paper for ethics reasons. We leave this as future work.

References

Simon Frieder, Luca Pinchetti, Chevalier Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023. Mathematical Capabilities of ChatGPT. *Advances in neural information processing systems*, 36:27699–27744.

Abigail S. Gertner and Kurt VanLehn. 2000. Andes: A Coached Problem Solving Environment for Physics. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, pages 133–142.

John Hattie and Helen Timperley. 2007. The power of feedback. *Review of Educational Research*, 77(1):81–112.

Irina Jurenka, Markus Kunesch, Kevin R McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, and 1 others. 2024. Towards responsible development of generative AI for education: An evaluation-driven approach. *arXiv preprint arXiv:2407.12687*.

Wenjie Ma, Andrei Cojocaru, Neel Kolhe, Bradley Louie, Robin Said Sharif, Haihan Zhang, Vincent Zhuang, Matei Zaharia, and Sewon Min. 2025. Reliable fine-grained evaluation of natural language math proofs. *arXiv preprint arXiv:2510.13888*.

Pepper Miller and Kristen DiCerbo. 2024. LLM Based Math Tutoring: Challenges and Dataset. Khan Academy.

Robert C. Moore. 2016. Mathematics Professors' Evaluation of Students' Proofs: A Complex Teaching Practice. *International Journal of Research in Undergraduate Mathematics Education*, 2(2):246–278.

OpenAI. 2024. Learning to Reason with LLMs.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. LLM Evaluators Recognize and Favor Their Own Generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

Manooshree Patel, Rayna Bhattacharyya, Thomas Lu, Arnav Mehta, Niels Voss, Narges Norouzi, and Gireeja Ranade. 2026. LeanTutor: Towards a Verified AI Mathematical Proof Tutor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 40670–40678.

Jelle Wemmenhove, Dick Arends, Thijs Beurskens, Maitreyee Bhaid, Sean McCarren, Jan Moraal, Diego Rivera Garrido, David Tuin, Malcolm Vassallo, Pieter Wils, and 1 others. 2022. Waterproof: Educational Software for Learning How to Write Mathematical Proofs. *arXiv preprint arXiv:2211.13513*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in neural information processing systems*, 36:46595–46623.