scientific reports

OPEN

A comparison of human, GPT-3.5, and GPT-4 performance in a university-level coding course

Will Yeadon✉, Alex Peach & Craig Testrow

This study evaluates the performance of ChatGPT variants, GPT-3.5 and GPT-4, both with and without prompt engineering, against solely student work and a mixed category containing both student and GPT-4 contributions in university-level physics coding assignments using the Python language. Comparing 50 student submissions to 50 AI-generated submissions across different categories, and marked blindly by three independent markers, we amassed $n = 300$ data points. Students averaged 91.9% (SE:0.4), surpassing the highest performing AI submission category, GPT-4 with prompt engineering, which scored 81.1% (SE:0.8)—a statistically significant difference ($p = 2.482 \times 10^{-10}$). Prompt engineering significantly improved scores for both GPT-4 ($p = 1.661 \times 10^{-4}$) and GPT-3.5 ($p = 4.967 \times 10^{-9}$). Additionally, the blinded markers were tasked with guessing the authorship of the submissions on a four-point Likert scale from 'Definitely AI' to 'Definitely Human'. They accurately identified the authorship, with 92.1% of the work categorized as 'Definitely Human' being human-authored. Simplifying this to a binary 'AI' or 'Human' categorization resulted in an average accuracy rate of 85.3%. These findings suggest that while AI-generated work closely approaches the quality of university students' work, it often remains detectable by human evaluators.

Keywords ChatGPT, GPT-4, Coding, Benchmark

Coding courses are now ubiquitous in university curricula globally, signifying the increasing recognition of programming as a critical skill within the digital economy. The emergence of advanced Large Language Models (LLMs), such as Codex that powers GitHub Copilot[1], prompts a reevaluation of coding assessments' efficacy and integrity within educational settings. Whilst the coding abilities of LLMs have been well-explored through benchmarks like MBPP and MathQA-Python[2], our investigation focuses on AI's potential effects on university coding courses. Rather than evaluating AI's capacity to solve coding puzzles, as seen in work by Tian et al.[3], this study looks at AI's potential impact on a practical coding curriculum. Specifically, we consider a 10-week coding course within a physics degree at Durham University, where students engage in lectures and complete assignments on an nb-grader powered Jupyter notebook server.

Physics degrees offer a broad spectrum of assessments, ranging from lab-based experiments and presentations to written exams, essays, and coding assignments. This variety distinguishes physics from disciplines that rely more heavily on particular types of assessments, such as written exams within mathematics, a greater focus on lab work in chemistry, a more essay-centric approach in English literature, or computer science where looking at student's coding abilities is a key form of assessment. This multifaceted approach within a single subject provides a good opportunity to assess AI's impact on various forms of academic assessment. Arguably, the tangible, hands-on nature of lab work and presentations largely precludes AI's influence. While AI's role in written physics exams has yet to match students[4], the integrity of essay-based assessments is being increasingly challenged by AI's ability to produce work that is both indistinguishable from, and comparable in quality to, that of humans[5]. Further, a critical aspect of a physics degree involves how students grasp, interpret, and apply core physics principles. Recent studies probing LLMs' comprehension of physics indicate their performance is improving significantly, with results increasingly approximating human levels; however, peculiar errors persist[6–9]. This nuanced progress highlights the necessity to continuously reassess the role and effectiveness of coding assignments as AI technologies advance. By examining physics coding courses, this study seeks to determine the ongoing validity, utility and integrity of such assignments in accurately assessing student performance in an era of rapid technological development. To ensure transparency and enable replication, the code used in this research is openly available on GitHub (https://github.com/WillYeadon/AI-Exam-Completion).

Department of Physics, Durham University, Durham DH1 3LB, UK. ✉email: will.yeadon@durham.ac.uk

Methods

Overview

This study aims to assess the effectiveness of contemporary Large Language Models (LLMs) in performing coding tasks typically assigned to university students, using a blinded marking approach to evaluate code written by both students and AI. In the context of physics, coding is primarily utilized for simulations and data analysis, including plotting. A critical skill is therefore the creation of clear, well-labeled plots that elucidate the underlying physics of a scenario. Consequently, this physics coding assessments place special emphasis on the quality of the produced plots and the performance of the code used to create them, particularly in terms of runtime. This approach contrasts with Computer Science scenarios, where students are expected to explain and justify their coding choices, with readability and maintainability being key evaluation criteria. To determine whether ChatGPT is an effective coding tool for physics education, 14 plots, documented in a 16-page report, were evaluated against a specific marking scheme for both AI and student-authored submissions. After blinded marking, the evaluators assigned the probably authorship on a 4-point Likert scale from 'Definitely AI' to 'Definitely Human'. This project received ethical approval from the Durham University physics ethics committee ref: `EDU-2023-03-14T14_02_18-hvxg44`. All methods were performed in accordance with Durham University's Research Integrity Policy and Code of Good Practice. All students who participated in this study completed a statement of informed consent.

Coding assignment

The coding assignments using in this study come from the 'Laboratory Skills and Electronics' module at Durham University, UK, designed for physics and natural sciences students in their second and third years. This module encompasses essential laboratory practices, electronics, and a coding component. The Python coding segment spans 10 weeks, featuring weekly lectures and a total of 8 weekly assignments, excluding the first and last week. Topics covered include finite difference methods, numerical integration, solving first and second order differential equations, Monte Carlo methods, and random walks. We solicited submissions from the 2023/24 student cohort, and from 103 consenting participants, 55 were randomly selected to represent the body of student work. A limitation here is that there is a chance the submitted student work may be in part itself be generative AI. This was explicitly against the course rules and warnings against the use of generative AI were made multiple times during the course. However, this remains a risk.

The 8 assignments are structured as a series of online Jupyter notebooks with short tasks that account for 30–60% of the total marks based on the completion of functions (graded automatically using assert statements) and the remainder assessed through manually graded plotting tasks. The course involves producing a total of 14 plots across the eight assignments: one plot for each of the first three assignments, two plots for each of the following four assignments, and three plots for the final assignment. Each assignment is worth one-eighth of the total coding mark and are scored out of 20. The plots themselves have varying marks assigned to them so for the purpose of this study each plot was scored out a five giving a max score of 70.

Generating the AI code

While students complete the assignments on an online nbgrader-powered Jupyter notebook server, enabling easy extraction of their plots as images, simulating AI completion of these notebooks requires converting them into a text format for input into an LLM. However, without explicit instructions such as '*Please complete this assignment*', there is no guarantee that an LLM will interact with the submitted text appropriately. Within many of the assignment notebooks students are often given a small amount of starter code, without specific guidance this can lead to the AI redefining variables or even data that has already been provided. Additionally, as previously mentioned, the assignments allocate 40–60% of the marks to assert statements, including hidden cells that are irrelevant when evaluating plots. Moreover, although there are 8 assignments yielding 14 plots, if the LLM produces code that throws errors, it should not automatically result in a zero score for the other plots within an assignment. These issues pose significant challenges for ensuring a fair comparison between AI-generated and student-generated submissions. Therefore, the assignment notebooks were pre-processed when converted into input for the LLM to align with AI processing capabilities. However, this act somewhat complicates a direct comparison as it means assisting the LLM, thus possibly altering the 'human versus AI' dynamic into 'human versus AI with human assistance.' Given the manner in which pre-processing is conducted has been found to influence the performance of GPT-4[10], we implemented only those changes that were absolutely necessary to get the AI to consistently complete the assignments. These changes are detailed in Table 1.

After the application of any form of pre-processing, there is a slippery slope for how much is apt as with enough pre-processing, exceptional performance can be achieve despite it not being reflective of actual AI capabilities. Nonetheless, despite the pre-processing steps detailed in Table 1, the inputs fed into the LLM still contained a fair amount of clutter. Instructions intended to guide humans through the notebook, such as '*Now implement a function*', could potentially confuse an LLM. The significant enhancement of performance through prompt engineering is well-documented[11]. Consequently, we prepared a second set of inputs in addition to the minimal pre-processing outlined in Table 1. These inputs incorporate prompt engineering focused enhancements, as detailed in Table 2. Finally, to test if potential idiosyncratic characteristics in AI-generated plots flag work as being AI, we merged human and AI work 50:50 into single submissions to see if these submissions more closely resemble human-only or AI-only work. Thus we created the following six categories:

- **GPT-3.5 raw**: This entry includes the assignment text with the minimal adjustments specified in Table 1, submitted directly to OpenAI's API using the gpt-3.5-turbo model for processing.

Change	Description
1	All graded cells tested via assert statements up to the plotting tasks were completed with correct answers, these cells represent 30-60% of the marks per notebook
2	At the beginning of the script, the text '*# Please read this script and then complete the plot described by* ' : : : Task : : : ' *writing your code where the script indicates 'HERE HERE HERE*" was inserted. The ' : : : Task : : : ' and 'HERE HERE HERE' were strategically placed to direct the LLM to the task and the specific location for code input, respectively. This approach was adopted to ensure that the AI consistently responds to the script rather than ignoring it or asking for more information
3	Cells containing assert statements for autograding were excluded from API submissions since the necessary code to pass them was already provided. Moreover, some of these cells were hidden from students anyway. This was done as the details of the assert statements are only pertinent for behind-the-scenes operation and are not relevant for the students or the AI
4	Some cells contained tables that the API couldn't parse in their original form, so these tables were converted into Python dictionaries. This alteration was solely for formatting purposes to facilitate parsing and did not involve any change in the data or addition of new information
5	Course-related content present in the notebooks, such as links to the course website, copied material, or submission guidelines for the internal server, was removed. Additionally, instructions like 'click on the "+" button to create new cells' were omitted for being considered unnecessary and potentially confusing to the AI. These edits were made to focus the content more directly on the study's objectives
6	In instances where multiple plots were required in a single workbook, AI-generated solutions were inserted at the appropriate places. Surrounding text was minimally altered for clarity, changing phrases like 'Create a plot' to 'Below is a simulation' to better reflect the content and context of the workbook

Table 1. Summary of the pre-processing changes to the raw .ipynb files to transfer them into Python scripts that could be completed by LLMs to compare to student work.

Change	Description
1	Function definitions within the notebooks were rewritten for clarity. For example, textual descriptions within the notebooks such as '*Define the function 'f', such that* $\mathrm{f}(x) \equiv x^2 \cos(2x)$. *This is the function that we will be integrating.*' were simplified to '*definition of the function* $f(x) = x^2 * \cos(2x)$'
2	All non-task-related information was removed to focus solely on the assignment requirements and to avoid potential confusion or distraction
3	An enhanced preamble was added to clearly outline the task and instructions for completion. This included explaining the task, providing objectives, and offering suggestions for successfully completing the assignment. Additionally, explicit locations for code insertion were marked with '*HERE HERE HERE*', guiding the AI to the expected areas of input
4	Post-task details were elaborated to further guide the completion process. This involved clarifying the task's aim, specifying objectives such as plotting differences between analytical and numerical derivatives, and offering suggestions to enhance the clarity and effectiveness of the plots

Table 2. Prompt engineering steps used for the GPT-3.5 with prompt engineering, GPT-4 with prompt engineering, and Mixed student and GPT-4 categories.

- **GPT-3.5 with prompt engineering**: Here, the assignment text is modified following the guidelines in Table 2 to optimize interaction with the GPT-3.5 model, aiming for improved responses.
- **GPT-4 raw**: Adopts the same approach as the 'GPT-3.5 Raw' but employs the more advanced gpt-4-1106-preview model.
- **GPT-4 with prompt engineering**: Mirrors the process used for the 'GPT-3.5 with prompt engineering', but uses the gpt-4-1106-preview model.
- **Mixed student and GPT-4**: This category amalgamates contributions from both students and GPT-4 enhanced with prompt engineering. Out of 10 mixed submissions, 14 plots were selected at random from five student entries using Python's random.choice method to fill half the slots, while the remaining plots were generated by the GPT-4 with prompt engineering.
- **Student only**: This group contained 50 student submissions from the randomly selected 55 of the 2023/24 cohort.

For each of these categories, we generated 10 submissions (each a PDF document containing 14 plots), resulting in a total of 50 AI submissions. Along with the 50 student submissions previously mentioned, this yielded a total of 100 submissions. These were blindly evaluated by three independent markers, providing $n = 300$ data points. The AI submissions were crafted by sending the input text—either in its raw form or after prompt engineering—to OpenAI's API. The response was then appended to the input text, executed as a Python script, and any resulting plots were extracted as images. At times, the AI's responses included written text or non-Python code, leading to errors where no image could be extracted. To achieve 10 submissions per category, the scripts were run in a while loop until the target number was met. Furthermore, in some cases, actual blank figures were generated. These were excluded as they provided no valuable information for this study.

Results

Score comparison

A combined dataset of the scores from the three markers for all submissions, evaluated blindly, is shown in Fig. 1. Here we see Student only achieved an average of 91.1% which is in line with the typical average for the actual coding component of 'Laboratory Skills and Electronics' at Durham University. In comparison the best performing AI category, GPT-4 with prompt engineering, scores 81.1%. A t-test between these groups produced a t-statistic of −8.193, with a p-value of 2.482×10^{-10}. This result shows that although GPT-4 exhibits remarkable capabilities, when it comes to physics coding assignments, it still often isn't as proficient as university students.

Examining the impact of prompt engineering reveals statistically significant improvements: GPT-4's scores increased from 71.9 (SE:1.3) to 81.1% (SE:0.8) with a p-value of 1.661×10^{-4} from a t-test, and GPT-3.5's scores improved from 30.9 (SE:1.2) to 48.8% (SE:1.4) with a t-test giving a p-value of 4.967×10^{-9}. Thus, as expected, there is are clear and significant benefits to prompt engineering. Interestingly, the mixed submissions, comprising both student and GPT-4 work, scored lower (76.0% with SE: 1.3) than GPT-4 submissions alone. This may be attributed to variability in the quality of student work sampled, given that the mixed group included plots from five student submissions compared to the 50 in the student-only group.

Author identification

After reviewing each submission, the evaluators assigned authorship scores on a Likert scale, the findings of which are depicted in Fig. 2. This demonstrates that genuine student submissions are more often recognized as student-authored. Converting the Likert scale to a numerical range—assigning 'Definitely AI' a value of 0 and 'Definitely human' a value of 3—we arrive at the average scores: 0.033 for GPT-3.5 with raw input, 0.200 for GPT-3.5 with prompt engineering, 0.467 for GPT-4 with raw input, 1.167 for GPT-4 with prompt engineering, 1.300 for the Mixed student and GPT-4 category, and 2.367 for Student only. Therefore all work with an AI-authored component to it has an average categorization closest to either 'Definitely AI' (0) or 'Probably AI' (1). Furthermore, by designating any submission not solely by students as 'AI-authored' to a degree, and comparing it to 'human-authored' submissions, we applied a Cochran-Armitage test. This test resulted in a p-value of 0.025 and a positive trend of 0.302, statistically verifying that as we move from 'Definitely AI' to 'Definitely Human' on the scale, the proportion of human-authored submissions increases.

These findings indicate that AI-generated content can be identified with a reasonable degree of accuracy. When categorizing the Likert scale responses into a binary system of 'AI' or 'Human,' the reviewers managed an average accuracy rate of 85.3%, with scores of 89% for Marker #1, 77% for Marker #2, and 90% for Marker #3. We found that content produced by the more advanced GPT-4 model is closer to that created by humans, especially when enhanced through prompt engineering techniques. This observation is contextualized by Fig. 1, which shows that content generated by humans is generally of higher quality than that produced by any form of AI, implying that work of superior quality is more frequently categorized as human.

Characteristics of AI work

Given the disparities in percentage scores across the six categories, we can trace their origins by examining performance on individual plots. Figure 3 shows the average of the three markers' scores for each plot across all

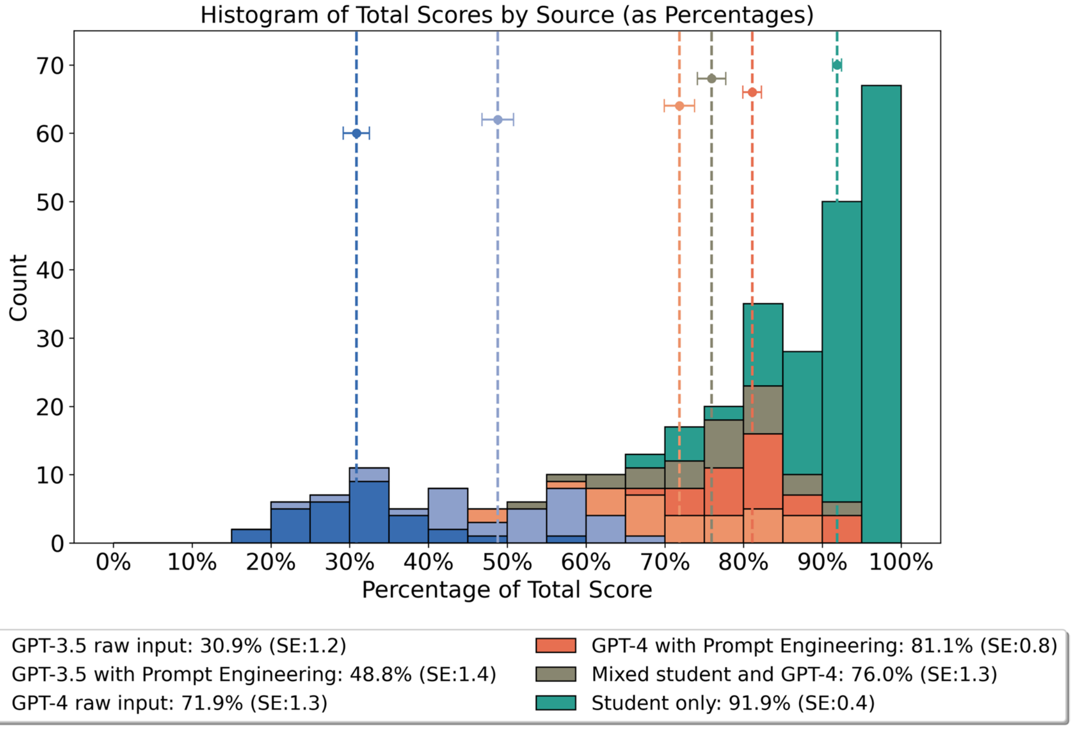

Fig. 1. Percent scores for each of the six categories of submission. Student submissions score the best thou they are closely followed by GPT-4 with prompt engineering and the Mixed student and AI work. GPT-3.5 performs strictly worse than GPT-4.

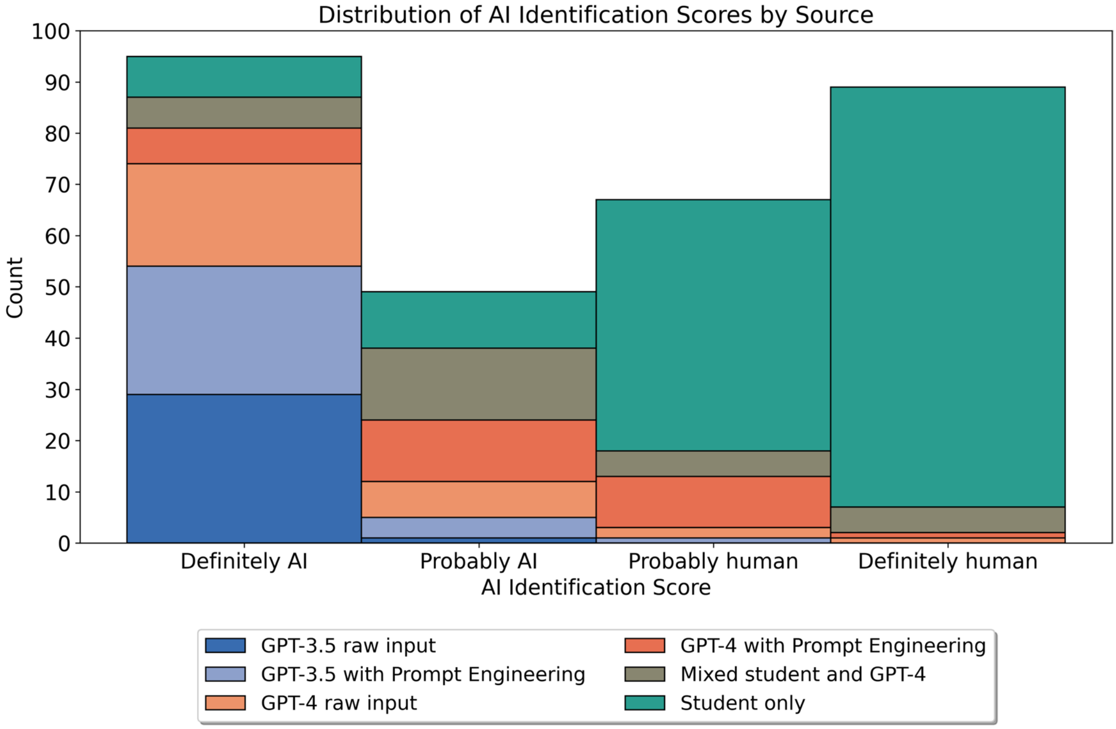

Fig. 2. Histogram showing the markers' assigned authorship versus actual authorship of the 300 assessed submissions. The amount of actual human-authored code in 'Definitely human' is 92.1% , then 73.1% in 'Probably human' followed by 22.4% in 'Probably AI' and 8.4% in 'Definitely AI'.

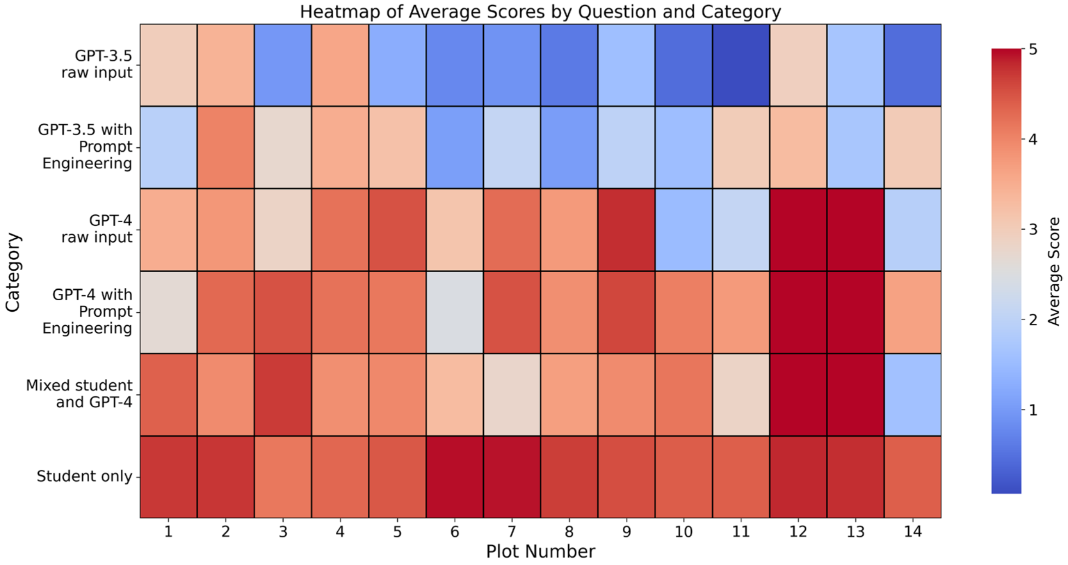

Fig. 3. Heatmap showing the average scores from three markers for each plot across the authorship categories. The colour scale transitions from blue (lower scores) to red (higher scores). Plot 10 exhibited the highest variability in scores across authorship categories, while Plot 4 showed the least.

categories. This reveals that plots 10 and 4 have the highest and lowest standard deviations of 1.727 and 0.343, respectively.

Plot 10, the first task in Assignment 7, requires comparing the convergence of the Newton-Raphson, Secant, and Bisection root-finding methods for the function $f(x) = x - \tanh(2x)$ to find the root at $x \approx 0.96$. Interestingly, visually plotting $f(x)$ reveals a second root at the origin, [0,0]—it is possible to converge towards this point instead of $x \approx 0.96$ if an inappropriate starting position is used. The structure of the Jupyter notebooks given to students meant they likely saw this plot of $f(x)$ rather than just the code defining it which may have led them to avoid this mistake. In fact, there is a noticeable underperformance of GPT-4 raw input, which scored an average of 1.5/5, on plot 10 compared to 4.067/5 for GPT-4 with prompt engineering. While not the primary focus of this work and thus based on a small sample, it is interesting that the prompt engineering detailed in Table 2 led to significantly different results.

In contrast, plot 4, the first task in Assignment 4, has the lowest standard deviation. It required plotting the trajectories of a cannonball launched at various angles between 0 and 90° found using Euler's method for solving ordinary differential equations. A possible explanation for the lower deviation is that this task, given at the start of the course, was relatively simple enabling all categories to perform well on it; the mean score across all categories here was a respectable 3.438/5. Support for this view is found in the next lowest standard deviation of 0.444 from plot 2, another relatively simple task of plotting the fractional error between results from numerical and analytical integration this time with a strong mean of 3.525/5. However, these interpretations are preliminary, as identifying strengths and weaknesses of question formats was not the main objective of this study.

Discussion

Overview and recommendations

The findings of this study indicate that, in the short term, the most sophisticated AI models might not rival human expertise in university-level physics coding assignments. However, our research highlights a distinct improvement of GPT-4 over GPT-3.5, a result also found across diverse disciplines from medicine[12] to university entrance exams[13]. This pattern showcases the steady advancement of AI capabilities, suggesting an evolving landscape where AI's potential to match or surpass human performance becomes increasingly plausible. Given this potential trajectory, it is important for educators to reassess the role of coding assignments and, more broadly, the objectives of their educational strategies. Coding, inherently a practical skill, involves regular consultation of documentation and the reuse of existing code. Therefore, the integration of AI into educational practices, akin to a pair programming setup exemplified by tools like GitHub Copilot, should not be viewed negatively[14,15]. This approach is an exciting opportunity for innovation in physics education. It prompts the exploration of whether AI could surpass traditional teaching methods as a more effective tutor for coding[16].

The lower performance in raw input categories indicates that, from an academic integrity perspective, students are likely to achieve better results by completing the 'Laboratory Skills and Electronics' assignments themselves. The extensive prompt engineering for the other AI categories, detailed in Table 2, necessitates a level of engagement with the course materials that there is ambiguity regarding its severity from an academic integrity standpoint. In fact, the aforementioned limitation that we could not guarantee the student category was 100% human written, given it is from 2023/24, may not be a cause of too much worry as the student work clearly has a different—and higher scoring—distribution in Fig. 1 than both Mixed student and GPT-4 and GPT-4 with Prompt Engineering.

While whether a particular amount of 'AI use' counts as academic misconduct might vary on an individual basis, a potential remedy involves implementing barriers that make the pre-processing described in Table 1 more demanding than the assignments themselves. For example, incorporating plots into the Jupyter notebooks for data interpretation, rather than using tables, could compel the AI to analyze visual data—which it may not do accurately—or necessitate a textual plot description, a process found to adversely affect GPT-4's performance[10].

Contrary to earlier studies examining physics essays[5], which found that human evaluators could not distinguish AI-generated content from human work better than random chance, our findings reveal that for coding assignments markers can quite successfully identify AI work. This difference is not merely due to the lower scores of GPT-3.5 inputs; all submissions with AI contributions were predominantly classified as either 'Definitely AI' or 'Probably AI'. Evaluators particularly noted the AI-generated plots' tendency to appear slightly askew or misaligned, attributes easily spotted by humans, such as unusual font size choices and or positioning. This said, a distinctive feature of student work highlighted by markers was the unique, sometimes bold, design choices students made, including unconventional colour schemes in their plots. This contrasted sharply with the AI's preference for default colours in matplotlib. These observations suggest that the nuanced, creative decisions by students serve as a clear differentiator from AI-generated content. Furthermore, as the markers in this study all have PhDs in physics and teach or have taught Computational Physics methods at the university level, they likely have a better understanding of physics than the students in a previous study[17], who struggled to distinguish between correct and incorrect AI responses for more complex physics, enabling them to evaluate the work more accurately.

Looking specifically at the Mixed student and GPT-4 category, there is a contradiction whereby the overall group average is lower than that of its two constituent categories in Fig. 1, but it is simultaneously rated above all the other AI categories yet below the Student only category in Fig. 2. Beyond sample-size effects, this may be due to the characteristic idiosyncrasies of student work not being correlated to their quality. For example, there are no marks awarded for the unconventional colour schemes students sometimes opt for, but this still identifies the work as student-authored. Regardless, investigating this further with a larger sample size could be an avenue for further study.

Limitations

An important methodological concern of this study is the effect of the pre-processing (see Table 1) on the AI's output quality. Although we deliberately limited the pre-processing to essentials, it played a crucial role in preparing the AI to comprehend and execute the given tasks effectively. By employing prompt engineering techniques, we noted a marked improvement in the AI's performance that was statistically significant. Considering further refinement of prompt engineering, such as incorporating detailed, step-by-step instructions, might boost the LLM's effectiveness even further. However, as previously discussed, this approach shifts the assessment focus from the inherent abilities of the AI to the efficiency of human-augmented AI interaction. Given that prior studies (e.g. Ghassemi et al.[18]) have shown how user involvement can significantly vary the performance enhancements achieved with AI, this introduces an additional layer of complexity to any analysis.

Another limitation is the selection effect within the Mixed student and GPT-4 category. A larger sample size could have mitigated this, as the $n = 5$ sample for the student portion seemed to include relatively weaker students. With a larger sample size the results could change.

Conclusion

This study found that the latest LLMs have not surpassed human proficiency in physics coding assignments. Nonetheless, we observed a strict superiority in the capabilities of GPT-4 over GPT-3.5, and identified that prompt engineering can significantly enhance performance. Should the improvement trajectory from GPT-3.5 to GPT-4 continue, LLMs may soon outperform student capabilities. Additionally, our analysis revealed that plots generated by LLMs are distinguishable from student-created ones due to their often misaligned or skewed layouts and a tendency to utilize default colour schemes. In contrast, student plots occasionally feature more unique, albeit sometimes garish, design choices. This distinction underscores the potential for human markers to identify AI-generated content. Crucially, this research highlights the importance of accounting for the degree of human intervention in evaluating the effectiveness of any LLM. As we move forward, these findings prompt a reevaluation of how we measure AI performance and the role of human collaboration in harnessing AI's full potential.

Data availability

The datasets generated and analysed during the current study are available in the figshare repository at doi.org/10.6084/m9.figshare.25673799. This repository includes the scores awarded by each marker and all AI work. The student work is not publicly available due to data privacy regulations.

Appendix A Breakdown of marks by marker

As shown in Fig. 4, plotting the results in Fig. 1 but by the scores from each of the three markers reveals strong similarity among them. To further investigate potential differences, we conducted an analysis of variance (ANOVA) on the three averages. The ANOVA yielded an F-value of 2.72 and a p-value of 0.067, which exceeds the conventional significance threshold of 0.05. Consequently, we fail to reject the null hypothesis, leading us to conclude that there is no significant evidence to suggest that the group means are different.

Beyond the ANOVA test, we employed the Intraclass Correlation Coefficient (ICC) analysis to evaluate the consistency of grading across multiple markers. The ICC1 model was used to assess the absolute agreement among markers and the ICC2 model was used to examine the agreement among markers on the relative ranking of submissions, rather than focusing on the exact scores assigned. The ICC ranges from −1 to 1, with a value of 1 indicating perfect agreement among markers, and −1 signifying complete disagreement. As shown in Table 3, both ICC1 and ICC2 yielded a high value of 0.932, indicating an excellent agreement and consistency in the ratings provided by the markers. Furthermore, the high F values (41.942 for ICC1 and 68.293 for ICC2) and

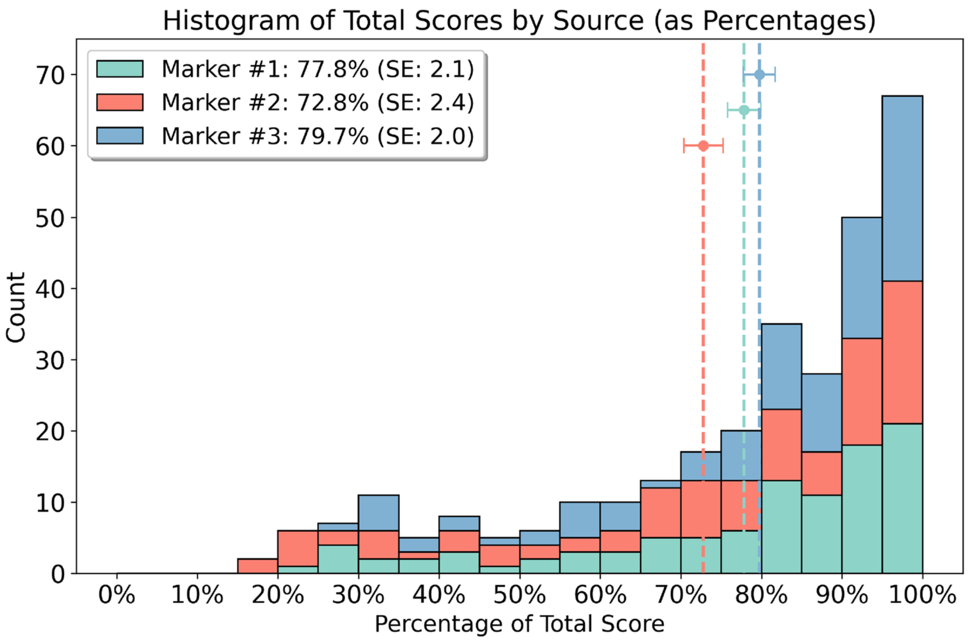

Fig. 4. Stacked histogram of the scores awarded by the three independent markers. Both the ANOVA and ICC models used find that the markers are consistent in their evaluations.

Model	ICC	F	df1	df2	p-value	95% CI
ICC1	0.932	41.942	99	200	$< 10^{-10}$	[0.91, 0.95]
ICC2	0.932	68.293	99	198	$< 10^{-10}$	[0.83, 0.97]

Table 3. Intraclass Correlation Coefficient (ICC) Analysis Results

extremely low p-values (≈ 0) suggest that the observed variances are statistically significant, providing strong evidence for the reliability of the grading process.

Received: 12 April 2024; Accepted: 19 September 2024
Published online: 07 October 2024

References

1. Chen, M. *et al.* Evaluating large language models trained on code. *arXiv preprint*[SPACE] arXiv:2107.03374 (2021).
2. Austin, J. *et al.* Program synthesis with large language models. *arXiv preprint*[SPACE] arXiv:2108.07732 (2021).
3. Tian, H. *et al.* Is chatgpt the ultimate programming assistant–How far is it? *arXiv preprint*[SPACE] arXiv:2304.11938 (2023).
4. Yeadon, W. & Hardy, T. The impact of AI in physics education: A comprehensive review from gcse to university levels. *Phys. Educ.* **59**, 025010. https://doi.org/10.1088/1361-6552/ad1fa2 (2024).
5. Yeadon, W., Agra, E., Inyang, O.-o., Mackay, P. & Mizouri, A. Evaluating ai and human authorship quality in academic writing through physics essays. *arXiv preprint*[SPACE] arXiv:2403.05458 (2024).
6. West, C. G. Ai and the fci: Can chatgpt project an understanding of introductory physics? *arXiv preprint*[SPACE] arXiv:2303.01067 (2023).
7. Kortemeyer, G. Could an artificial-intelligence agent pass an introductory physics course?. *Phys. Rev. Phys. Educ. Res.* **19**, 010132 (2023).
8. Polverini, G. & Gregorcic, B. Performance of chatgpt on the test of understanding graphs in kinematics. *Phys. Rev. Phys. Educ. Res.* **20**, 010109 (2024).
9. Polverini, G. & Gregorcic, B. How understanding large language models can inform the use of chatgpt in physics education. *Eur. J. Phys.* **45**, 025701 (2024).
10. Feng, T. H., Denny, P., Wuensche, B., Luxton-Reilly, A. & Hooper, S. More than meets the AI: Evaluating the performance of gpt-4 on computer graphics assessment questions. In *Proceedings of the 26th Australasian Computing Education Conference* 182–191 (2024).
11. OpenAI. Best practices for prompt engineering with openai api (2023). https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api.
12. Rosoł, M., Gąsior, J. S., Łaba, J., Korzeniewski, K. & Młyńczak, M. Evaluation of the performance of gpt-3.5 and gpt-4 on the polish medical final examination. *Sci. Rep.* **13**, 20512 (2023).
13. Nunes, D., Primi, R., Pires, R., Lotufo, R. & Nogueira, R. Evaluating gpt-3.5 and gpt-4 models on brazilian university admission exams. *arXiv preprint*[SPACE] arXiv:2303.17003 (2023).
14. Bird, C. *et al.* Taking flight with copilot: Early insights and opportunities of AI-powered pair-programming tools. *Queue* **20**, 35–57 (2022).
15. Moradi Dakhel, A. *et al.* Github copilot AI pair programmer: Asset or liability?. *J. Syst. Softw.* **203**, 111734. https://doi.org/10.1016/j.jss.2023.111734 (2023).
16. Wu, T., Koedinger, K. *et al.* Is ai the better programming partner? human-human pair programming vs. human-ai pair programming. *arXiv preprint*[SPACE] arXiv:2306.05153 (2023).
17. Dahlkemper, M. N., Lahme, S. Z. & Klein, P. How do physics students evaluate artificial intelligence responses on comprehension questions? A study on the perceived scientific accuracy and linguistic quality of chatgpt. *Phys. Rev. Phys. Educ. Res.* **19**, 010142 (2023).
18. Dell'Acqua, F. *et al.* Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality. in *Harvard Business School Technology & Operations Mgt. Unit Working Paper* (2023).

Author contributions

W.Y. conceived the experiment, analysed the results and served as a Marker, A.P. and C.T. served as Markers. All authors reviewed the manuscript. We confirm the corresponding author has read the journal policies and submit this manuscript in accordance with those policies.

Competing interests

We declare that the authors have no competing interests as defined by Nature Research, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Additional information

Correspondence and requests for materials should be addressed to W.Y.