WIP: Low Effort, High Grades? Benchmarking LLMs on Various Engineering Assignments

Mr. Yuxuan Chen, University of Illinois Urbana-Champaign

Yuxuan Chen is a Master's student in Computer Science at UIUC. His primary research interests focus on computer science education and artificial intelligence. He is dedicated to enhancing student learning experiences and accessibility in computing education through both innovative technology and research-driven teaching practices.

Siegfried Eggl, University of Illinois Urbana-Champaign

Siegfried Eggl (Ph.D. '13 Univ. of Vienna, Austria) is Assistant Professor in the Grainger College of Engineering, Department of Aerospace Engineering at the University of Illinois Urbana-Champaign, where he is leading the Astrodynamics and Planetary EXploration (APEX) group at UIUC conducting research in Space Situational Awareness, Celestial Navigation, Planetary Science and Planetary Defense. As part of the NSF-Simons Foundation-funded SkAI institute, he is engaged in AI research across astronomy and aerospace engineering.

Dr. Abdussalam Alawini, University of Illinois Urbana-Champaign

Dr. Alawini is a Teaching Associate Professor in the School of Computing and Data Science at the University of Illinois at Urbana-Champaign. He holds a bachelor's degree from the University of Tripoli and master's and doctoral degrees in Computer Science from Portland State University. His research focuses on data management systems and computing education. Dr. Alawini is passionate about building data-driven, AI-based systems for improving teaching and learning.

Prof. Mariana Silva, University of Illinois Urbana-Champaign

Mariana Silva is a Teaching Associate Professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign. Silva is known for her teaching innovations and educational studies in large-scale assessments and collaborative learning. She has participated in two major overhauls of large courses in the College of Engineering: she played a key role in the re-structure of the three Mechanics courses in the Mechanical Science and Engineering Department, and the creation of the new computational-based linear algebra course, which was fully launched in Summer 2021. Silva research focuses on the use of web-tools for class collaborative activities, and on the development of online learning and assessment tools. Silva is passionate about teaching and improving the classroom experience for both students and instructors.

Max Fowler, University of Illinois Urbana-Champaign

Max Fowler is a Teaching Assistant Professor in the Siebel School of Computing and Data Science at the University of Illinois Urbana-Champaign. Much of Max's research is in the space of skills and assessment: specifically identifying the core competencies of programming (and, increasingly, skills for working with generative AI) while developing strategies for fair, secure assessment-at-scale. His work in this space includes leveraging GenAI for autograding and feedback provision, intersecting with "LLM-as-judge" methodologies.

Abhishek Umrawal, University of Illinois Urbana-Champaign

Dr. Abhishek Umrawal is a Teaching Assistant Professor of Electrical and Computer Engineering (ECE) at the University of Illinois Urbana–Champaign, with affiliations in the Coordinated Science Laboratory (CSL) and the National Center for Supercomputing Applications (NCSA). He earned his Ph.D. in Industrial Engineering (Operations Research) from Purdue University, where his doctoral work developed efficient machine learning algorithms for influence maximization on social networks. His research integrates machine learning, operations research, and causal inference to design intelligent and trustworthy decision

making systems, with applications in algorithmic marketing, digital platforms, and safe and controllable generative AI. His scholarship also includes teaching-integrated work in computing and algorithms education, including the responsible use of AI in instruction and assessment. He also holds an M.S. in Economics from Purdue University's Daniels School of Business and an M.S. in Statistics from the Indian Institute of Technology (IIT) Kanpur.

Melkior Ornik, University of Illinois Urbana-Champaign

Melkior Ornik (mornik@illinois.edu) is an Assistant Professor with the Dept. of Aerospace Engineering and the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL, USA. He received a Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2017. In the context of education, he is interested in exploring the impact of generative artificial intelligence on the postsecondary enterprise. He is the principal investigator of the effort "Artificially Successful? Investigating and Responding to Generative AI's Capability to Solve Course Assignments" supported by the Strategic Instructional Innovations Program of the University of Illinois Urbana-Champaign's Grainger College of Engineering. His technical research interests include developing theory and algorithms for learning and planning of autonomous systems operating in uncertain, complex, and changing environments, as well as in scenarios where only limited knowledge of the system is available.

WIP: Low Effort, High Grades? Benchmarking LLMs on Various Engineering Assignments

1 Introduction

With the rapid advancement of large language models (LLMs) [1, 2], students are increasingly adopting such tools in their academic work [3]. This has raised growing concerns among educators regarding academic integrity [4, 5] as well as the potential impact of LLM use on student learning outcomes [3, 6].

Recent LLM systems support expanded capabilities, including explicit reasoning through chain-of-thought prompting [7], multimodal understanding of files and images [8], programming code generation [9], and complex document production (e.g., slide decks, spreadsheets, and PDFs) [10]. These advances have enhanced the adoption of LLMs in educational contexts [11], where they are increasingly used for automated assessment [12] and conversational tutoring [13, 14]. Instructors have expressed concern that students may rely on LLMs to minimize effort on coursework, potentially diminishing meaningful learning. In this paper, we specifically examine whether currently widely used LLMs can complete various authentic engineering coursework end-to-end (from prompt to final solution without human intervention).

Systematic evidence remains limited regarding LLM performance across the range of assignment formats commonly used in engineering education. Wang et al. [15] examine ChatGPT's success in a calculus-based physics course, but is limited to homework questions with numerical answers. Kortemeyer [16] experiments with a more varied list of assignments, also in a physics course, but allowed a significant amount of human interaction. Borges et al. [17] tested ChatGPT's performance on a large bank of mathematical questions across courses in their university. However, these questions were all text-based and largely had closed, quantitative answers. Puthumanaillam et al. [18] evaluate ChatGPT across an entire undergraduate aerospace engineering course, examining its ability to independently complete a broad range of coursework tasks. Building on prior work, this work-in-progress study explores the end-to-end performance of multiple LLMs across various authentic engineering assignment types, including both open-ended and closed-form problems, with the goal of providing preliminary insights to inform future comprehensive evaluations.

We evaluate six state-of-the-art LLMs commonly used by students on 24 authentic assignments drawn from three undergraduate engineering courses at a public research university, covering introductory programming, numerical methods, and computational algorithms. Courses range from introductory to upper-division. To assess the effects of prompt input modality, we compare image-based prompts versus simplified-text prompts. To capture assignment-type diversity, we

organize tasks into multiple-choice questions, programming code-writing tasks, and constructed-response questions, which we further divide into long-form algorithm design questions and short-form conceptual analysis questions.

2 Methods

We approximate an archetype of a student who wants to get a "good grade" without putting in the effort required to achieve such a grade. Therefore, we design an evaluation pipeline in which assignment materials are directly provided to LLMs, and the model's output is submitted with minimum human intervention. In this section, we describe the types of assignments considered, the selection of LLMs evaluated, and the prompts used in our study.

2.1 Assignment Types

We collected 24 authentic assignment questions in total, sampled from three undergraduate engineering courses at a large public research university. These questions represent a subset of the assignment content within each course. The courses cover introductory programming, numerical methods, and computational algorithms, and range from lower-division to upper-division levels, allowing us to assess model performance across varying levels of conceptual difficulty. The introductory programming and numerical methods courses used Python as their programming language, while the computational algorithms course required students to write pseudocode.

We categorized assignments by assessment type: multiple-choice questions, programming code-writing tasks, and constructed-response questions. We further divided constructed-response questions into long-form algorithm design questions, which require extended written responses describing how to solve an algorithmic problem (including algorithm selection, correctness justification, and pseudocode), and short-form conceptual analysis tasks, which require brief written responses focused on reasoning and explanation of numerical concepts.

Multiple-choice submissions and code-writing submissions were both auto-graded through an online assessment platform used by the university. Algorithm-design responses were graded by two engineering graduate students with advanced training in algorithms using a predefined rubric; any discrepancies were reviewed together and resolved. Graders were not the subjects of this study and thus their quality was not evaluated; the grades that the LLM received from the grader were considered as-is, without further analysis. Conceptual-analysis responses were added to each course's regular grading pool, mixed with student submissions, and evaluated by the course teaching assistant without disclosure of authorship to minimize bias. These grading formats were defined by the course instructors based on learning goals and course pedagogy, and had been used in prior semesters independent of this study; the only change in the present work was the inclusion of AI-generated responses among the graded submissions. Table 1 provides an summary of the 24 assignments used in this study.

2.2 LLM Selection

Our goal is to evaluate whether LLMs that are currently widely used by students can complete authentic engineering coursework in a manner that reflects realistic student use. We approximate

Assignment Type	Course	Grading Method	Count
Multiple-choice	Intro Programming	Autograder	4
Programming code-writing	Intro Programming	Autograder	4
Algorithm design	Computational Algorithms	Human	4
Conceptual analysis	Numerical Methods	Human	12

Table 1: Overview of the 24 authentic coursework assignments evaluated in this study, showing assignment type, course context, grading method, and the number of assignments in each category.

scenarios in which students rely on readily available generative AI tools to complete assignments end-to-end with minimal additional effort. Therefore, we restrict our study to models that are accessible through chat-based interfaces, are free to use or available at relatively low cost, and require no specialized technical setup.

Many authentic engineering assignments include LaTeX-formatted equations, diagrams, or other visual elements. Based on our instructional experience across multiple engineering courses, we observed that some students submitted such materials directly (e.g., as screenshots or supplementary files) rather than manually transcribing them into text. Therefore we further require that all selected models support image-based and supplementary file inputs. We also systematically compare model performance under plain natural-language text prompting and image-based prompting conditions, described further in Section 2.3.

Based on the above criteria, we selected six state-of-the-art LLMs that were widely accessible to students at the time of this study [19]: GPT-5-Thinking, GPT-5-Instant, GPT-4o, Gemini-2.5-Pro, Claude Sonnet 4, and DeepSeek-V3.1 DeepThink.

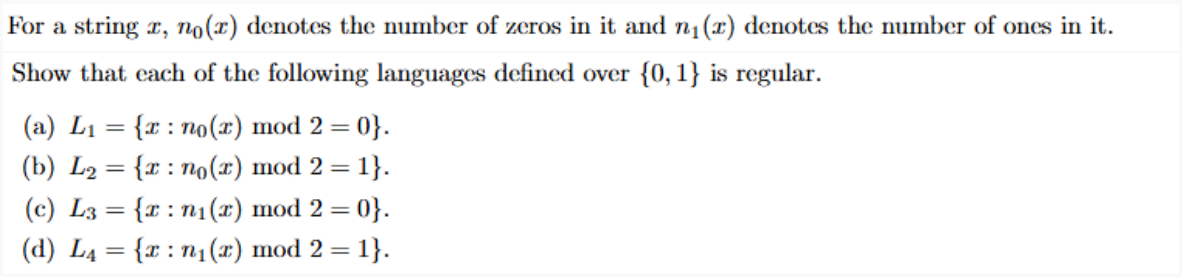
2.3 LLM Prompts

We adopt prompting strategies similar to those proposed by Puthumanaillam et al. [18]. Specifically, we evaluate two prompting techniques intended to reflect realistic minimum-effort student behavior when submitting coursework to LLMs. All model generations are performed without human intervention beyond prompt preparation.

Image-based prompting. In the first approach, assignment questions were provided to the LLMs as direct screenshot uploads, including any mathematical equations and diagrams (Figure 1). This approach closely mirrors how students may interact with LLMs in practice by submitting photographed or screenshot assignment materials without any manual transcription.

Simplified-text prompting. In the second approach, assignment questions were translated into plain natural-language text (Figure 2). When questions contained LaTeX expressions, mathematical notation, or tables, these elements were converted into equivalent textual descriptions. This prompting method serves as a text-only baseline and allows us to compare performance against image-based inputs while preserving the problem content.

LLM input prompt (screenshot)



You are given a screenshot of a question. Provide the final solution in LaTeX format.

Figure 1: Example input prompt using image-based prompting, where the original assignment is provided as a screenshot.

LLM input prompt (simplified text)

Description:
For a string x, n0(x) denotes the number of zeros in it and n1(x) denotes the number of ones in it.
Show that each of the following languages defined over {0,1} is regular.
a) L1 = {x : n0(x) mod 2 = 0}
b) L2 = {x : n0(x) mod 2 = 1}
c) L3 = {x : n1(x) mod 2 = 0}
d) L4 = {x : n1(x) mod 2 = 1}

You are given a question description. Provide the final solution in LaTeX format.

Figure 2: Example LLM input prompt using simplified-text prompting, where any mathematical notation and formatting are converted into plain natural-language text.

3 Results and Discussion

We report preliminary results on the performance of six LLMs across question types and prompting modalities. Performance is computed by averaging the percentage scores obtained on individual questions within each category. We first summarize overall model performance aggregated across all assignments, and then present detailed analyses by question type and prompting modality.

Overall performance. GPT-5-Thinking achieves the highest overall performance with a performance score of 98.47% with Gemini-2.5-Pro following closely at 97.08%. DeepSeek-V3.1 DeepThink, GPT-5-Instant, Claude Sonnet 4, and GPT-4o obtain performance scores of 96.94%, 95.65%, 94.61%, and 93.08%, respectively.

Performance by question type. Figure 3 reports performance by question category. Across all six models, performance is near ceiling on multiple-choice questions and programming code-writing tasks. All models achieve 100% on programming questions. For multiple-choice questions, five of the six models score 100%, with the only deviation coming from GPT-4o

(87.5%). These results suggest that currently widely used LLMs are capable of reliably generating correct solutions for introductory programming coding and multiple-choice assignments, as expected.

The models achieve near-perfect scores on advanced algorithm-design questions, contrary to common assumptions about the relative difficulty of long-form responses. LLMs may struggle with long text generation, especially as required length and structural complexity increase [20]. However, we discover here that despite requiring extended written responses, correctness justification, and pseudocode, all models achieve mean scores above 96%.

In contrast, short-form conceptual analysis questions achieve slightly lower accuracy than long-form algorithm-design questions for every model evaluated in this study, although still considered high overall. Scores range from 86.94% (Claude Sonnet 4) to 96.67% (GPT-5-Thinking). Errors in this category frequently involve incorrect mathematical reasoning, misuse of methods not covered in the course, or incomplete or inconsistent explanations.

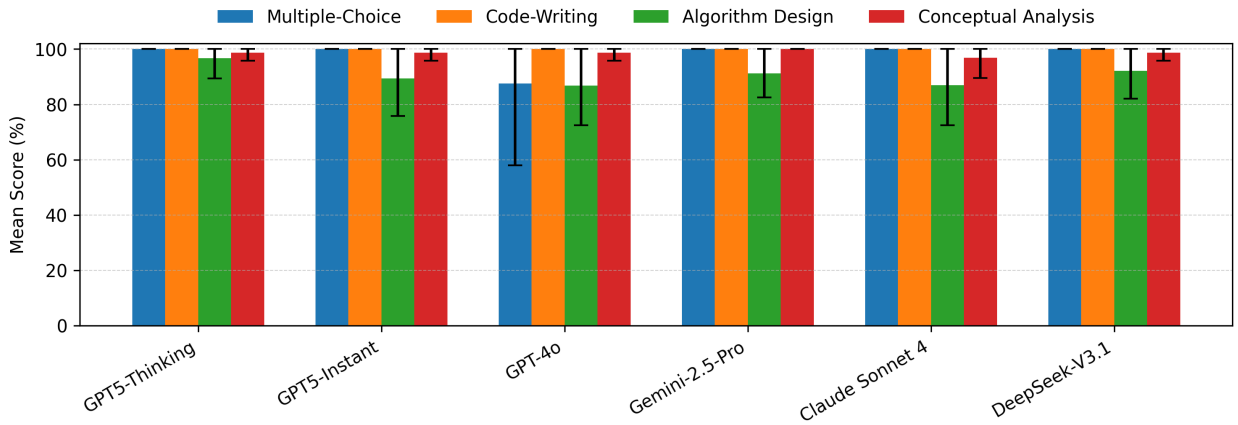


Figure 3: Performance of six LLMs across the four question categories: multiple-choice, programming code-writing, short-form conceptual analysis, and long-form algorithm design. Error bars indicate 95% confidence intervals around the mean scores.

Performance by input modality. Figure 4 compares model performance under simplified-text prompting and image-based prompting modalities. This analysis is restricted to two courses for which identical questions were evaluated under both modalities; the others courses were excluded because comparable paired inputs were not available.

Across most models, simplified-text and image-based prompting yield comparable performance, and no consistent advantage of one modality over the other is observed. In some cases, image-based prompting performs slightly worse, while in others performance is marginally higher, suggesting that modality effects are model-dependent rather than systematic.

Exceptions include Claude Sonnet 4 and GPT-4o, which exhibit lower performance under image-based prompting in specific settings. Manual inspection of errors reveal that these failures come from misinterpreted information in the image input (unread equations and numerical values), suggesting limitations in optical character recognition rather than deficiencies in problem-solving ability. More recent models appear less affected by such issues.

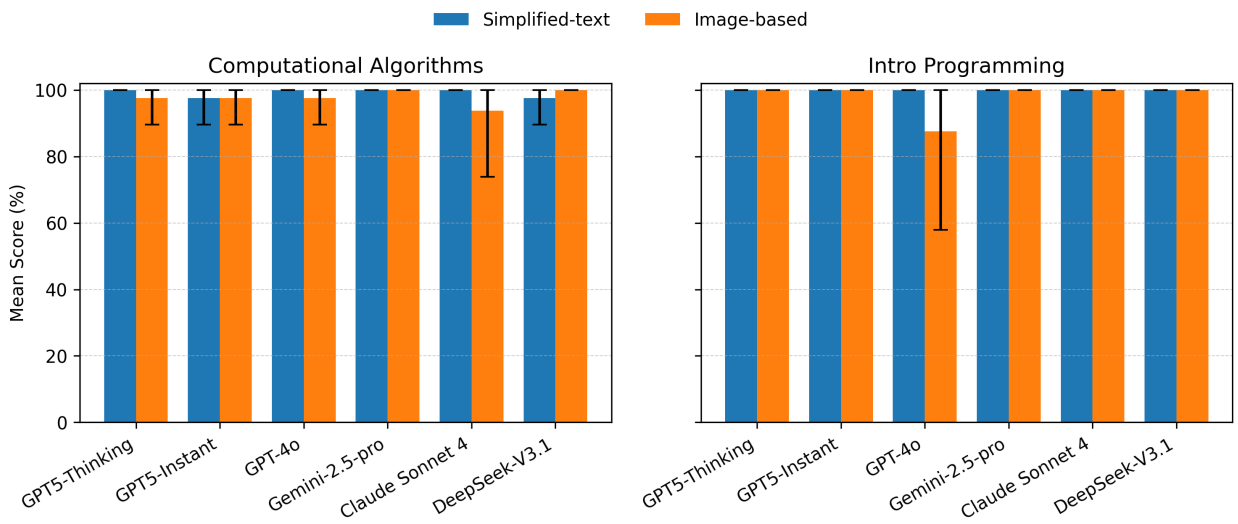


Figure 4: Performance of six LLMs under simplified-text and image-based prompting across two courses. Error bars indicate 95% confidence intervals around the mean scores. A third course is excluded because assignment questions were not evaluated under both prompting approaches.

4 Conclusion

In this paper, we present a preliminary evaluation of six widely used LLMs on a range of authentic engineering coursework, focusing on their ability to generate end-to-end solutions under minimal human intervention. Our study aims to extend prior research by examining multiple widely used LLMs across both open-ended and closed-form assignment types.

We observe near-ceiling performance on multiple-choice, code-writing, and long-form algorithm design questions across all six models. Performance on conceptual analysis questions is more variable, though generally high. Text-based and image-based prompts perform similarly overall; image-based prompts exhibit occasional errors due to optical character recognition limitations. These preliminary findings provide insight into the capabilities and limitations of current LLMs when applied to authentic engineering coursework.

This work serves as an initial step toward our ongoing research on more comprehensive, longitudinal evaluations of LLM performance across complete engineering courses and multiple semesters. We aim to understand the implications of LLM use for assessment design and student learning outcomes in engineering education.

Acknowledgments

This work was supported through the Strategic Instructional Innovations Program (SIIP) by The Grainger College of Engineering at the University of Illinois Urbana-Champaign. We thank Olga Mironenko, Keilin Jahnke, and Kellie Halloran for their ongoing support throughout the project. We thank Abrar Murtaza for assistance with assignment grading. Finally, we acknowledge the many others whose contributions helped make this research possible.

References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[2] Y. Liu and H. Wang, "Who on Earth is using generative AI?" *World Development*, vol. 199, p. 107260, 2026.

[3] H. Pearson, "Universities are embracing AI: will students get smarter or stop thinking?" *Nature*, vol. 646, pp. 788–791, 2025. [Online]. Available: https://doi.org/10.1038/d41586-025-03340-w

[4] S. Lau and P. Guo, "From 'Ban It Till We Understand It' to 'Resistance is Futile': How University Programming Instructors Plan to Adapt as More Students Use AI Code Generation and Explanation Tools such as ChatGPT and GitHub Copilot," in *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 1*, ser. ICER '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 106–121. [Online]. Available: https://doi.org/10.1145/3568813.3600138

[5] B. Chen, C. M. Lewis, M. West, and C. Zilles, "Plagiarism in the Age of Generative AI: Cheating Method Change and Learning Loss in an Intro to CS Course," in *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, ser. L@S '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 75–85. [Online]. Available: https://doi.org/10.1145/3657604.3662046

[6] J. Wang and W. Fan, "The effect of ChatGPT on students' learning performance, learning perception, and higher-order thinking: insights from a meta-analysis," *Humanities and Social Sciences Communications*, vol. 12, p. 621, 2025. [Online]. Available: https://doi.org/10.1057/s41599-025-04787-y

[7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, ser. NIPS '22. Red Hook, NY, USA: Curran Associates Inc., 2022.

[8] S. Schulhoff, M. Ilie, N. Balepur, K. Kahadze, A. Liu, C. Si, Y. Li, A. Gupta, H. Han, S. Schulhoff, P. S. Dulepet, S. Vidyadhara, D. Ki, S. Agrawal, C. Pham, G. Kroiz, F. Li, H. Tao, A. Srivastava, H. D. Costa, S. Gupta, M. L. Rogers, I. Goncearenco, G. Sarli, I. Galynker, D. Peskoff, M. Carpuat, J. White, S. Anadkat, A. Hoyle, and P. Resnik, "The Prompt Report: A Systematic Survey of Prompt Engineering Techniques," 2025. [Online]. Available: https://arxiv.org/abs/2406.06608

[9] X. Jiang, Y. Dong, L. Wang, Z. Fang, Q. Shang, G. Li, Z. Jin, and W. Jiao, "Self-Planning Code Generation with Large Language Models," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 7, Sep. 2024. [Online]. Available: https://doi.org/10.1145/3672456

[10] Gemini Team, "Gemini: A Family of Highly Capable Multimodal Models," 2025. [Online]. Available: https://arxiv.org/abs/2312.11805

[11] H. Xu, W. Gan, Z. Qi, J. Wu, and P. S. Yu, "Large Language Models for Education: A Survey," 2024. [Online]. Available: https://arxiv.org/abs/2405.13001

[12] C. Zhao, M. Silva, and S. Poulsen, "Language Models are Few-Shot Graders," in *Artificial Intelligence in Education*. Cham: Springer Nature Switzerland, 2025, pp. 3–16.

[13] J. Batista, A. Mesquita, and G. Carnaz, "Generative AI and higher education: Trends, challenges, and future directions from a systematic literature review," *Information*, vol. 15, no. 11, p. 676, 2024.

[14] A. AlRabah, Z. Li, M. Blumthal, S. Koloutsou-Vakakis, V. Kindratenko, T. Kozlowski, and A. Alawini, "Data-Driven Insights into AI-Powered Learning: Analyzing Student Interactions with AI-bot in Engineering Education," in *2025 ASEE Annual Conference & Exposition*, no. 10.18260/1-2–56204. Montreal, Quebec, Canada: ASEE Conferences, June 2025, https://peer.asee.org/56204.

[15] K. D. Wang, E. Burkholder, C. Wieman, S. Salehi, and N. Haber, "Examining the potential and pitfalls of ChatGPT in science and engineering problem-solving," *Frontiers in Education*, vol. Volume 8 - 2023, 2024. [Online]. Available: https://www.frontiersin.org/journals/education/articles/10.3389/feduc.2023.1330486

[16] G. Kortemeyer, "Could an artificial-intelligence agent pass an introductory physics course?" *Phys. Rev. Phys. Educ. Res.*, vol. 19, p. 010132, May 2023. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevPhysEducRes.19.010132

[17] B. Borges, N. Foroutan, D. Bayazit, A. Sotnikova, S. Montariol, T. Nazaretsky, M. Banaei, A. Sakhaeirad, P. Servant, S. P. Neshaei, J. Frej, A. Romanou, G. Weiss, S. Mamooler, Z. Chen, S. Fan, S. Gao, M. Ismayilzada, D. Paul, P. Schwaller, S. Friedli, P. Jermann, T. Käser, and A. Bosselut, "Could ChatGPT get an engineering degree? Evaluating higher education vulnerability to AI assistants," *Proceedings of the National Academy of Sciences*, vol. 121, no. 49, Nov. 2024. [Online]. Available: http://dx.doi.org/10.1073/pnas.2414955121

[18] G. Puthumanaillam, T. Bretl, and M. Ornik, "The Lazy Student's Dream: ChatGPT Passing an Engineering Course on Its Own," in *Proceedings of the 14th IFAC Symposium on Advances in Control Education*, 2025, pp. 213–218. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896325006184

[19] S. Sarkar, "The AI 'Big Bang' Study 2025: Best AI Chatbots and What 55.88 Billion Visits Reveal," 2025, last updated August 8, 2025; accessed January 21, 2026. [Online]. Available: https://onelittleweb.com/data-studies/best-ai-chatbots/

[20] S. Wu, Y. Li, X. Qu, R. Ravikumar, Y. Li, T. Loakman, S. Quan, X. Wei, R. Batista-Navarro, and C. Lin, "LongEval: A Comprehensive Analysis of Long-Text Generation Through a Plan-based Paradigm," 2025. [Online]. Available: https://arxiv.org/abs/2502.19103