2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #51299

Integration of Generative AI Coding into a Course on Computational Biomedical Engineering

Kaitlyn Wintruba, University of Virginia

Kaitlyn Wintruba (she/her) is a PhD student in Biomedical Engineering at the University of Virginia.

Lavie Ngo, University of Virginia
Dr. Jeffrey J Saucerman, University of Virginia

Integration of Generative AI Coding into a Course on Computational Biomedical Engineering

Generative artificial intelligence (GenAI) tools are increasingly used by biomedical engineering students, yet formal instruction and guidance on their effective use are often absent from biomedical engineering curricula. In this study, we investigated the impact of integrating structured GenAI instruction into an undergraduate Computational Biomedical Engineering course. We hypothesized that GenAI instruction and optional use would not hinder student learning outcomes. Early in the semester, students received instruction on the fundamentals of GenAI and its application for coding support, including how these tools work, common pitfalls such as inaccurate code generation and hallucinations, and strategies for responsible use. Students were required to use GenAI tools on the first problem set and then given the option to use it on the subsequent ten coding assignments. Learning of non-AI course content, self-reported completion time, and student perceptions were evaluated through pre- and post-course assessments and anonymous surveys. Students who used GenAI ($n = 15$) and those who did not ($n = 8$) demonstrated comparable improvements in conceptual understanding, indicating no negative impact on non-AI course content. Students who used GenAI reported shorter average completion times at the aggregate level, though significant time reductions were limited to select assignments and results varied among individual students. Student interest in biomedical engineering and confidence in foundational computational topics were consistently positive across groups. Overall, these findings suggest that intentional GenAI instruction and optional use can be integrated into computational biomedical engineering courses without compromising learning or student interest in biomedical engineering.

Introduction

Computational methods and coding are fundamental skills for biomedical engineers, particularly in preparation for careers in biotechnology, data science, medical devices, and related fields [1]. As a result, developing programming competency is critical for biomedical engineering students. An emerging challenge to this goal is the rapid and expanding use of generative artificial intelligence (GenAI) tools. The increasing presence of GenAI in engineering education has sparked significant discussion regarding its appropriate role and potential challenges to student learning. Prior studies indicate that GenAI tools can assist students with procedural coding tasks, but that ineffective use may also constrain opportunities for meaningful learning [2], [3], [4], [5], [6]. Consequently, biomedical engineering educators continue to grapple with how to integrate GenAI tools responsibly while preserving meaningful learning and disciplinary skill development [1].

Evidence evaluating the impact of GenAI use on student learning remains limited across STEM education [7]. Within BME courses, several recent studies have incorporated GenAI tools in to lectures, assignments, or laboratory activities, and reported generally positive student perceptions [8], [9], [10], [11]. However, many of these studies emphasize qualitative feedback or self-reported attitudes and do not assess learning outcomes related to programming-intensive course content. Moreover, no studies have yet examined undergraduate computational biomedical engineering courses, where coding is typically used as a means to support disciplinary learning. As GenAI tools continue to become increasingly capable and accessible, there is a need for to evaluate how their use intersects with learning outcomes, coding efficiency, and student perceptions in this specific educational context.

To address this need, we examined how structured instruction on GenAI and optional use of these tools influenced student learning outcomes in an introductory undergraduate Computational Biomedical Engineering course. Prior work suggests that concerns about GenAI hindering learning primarily arise from difficulties in its effective integration into teaching materials, [6], [7], whereas structured instructional approaches are more likely to support student engagement and critical thinking [12]. Accordingly, we hypothesized that GenAI instruction and optional use would not hinder student learning outcomes. This hypothesis was examined through three research questions (RQs): RQ1) How does GenAI use on coding assignments affect conceptual learning of non-AI course content? RQ2) How does GenAI use affect completion time of coding assignments? RQ3) How does GenAI use on coding assignments affect student interest in BME?

Methods

Course and study design

All study procedures were reviewed and approved by the Institutional Review Board for Social and Behavioral Sciences (IRB-SBS) at the University of Virginia. This study was conducted in a sophomore-level Computational Biomedical Engineering course required for all undergraduate biomedical engineering majors at a large, public R1 research university. Data was collected from 23 students enrolled in the course during the Fall 2023 semester. The course introduced techniques for analyzing complex biomedical data and developing computational models of biological systems. This course has long taught the computational foundations of what is now considered AI, how to use computational methods to solve biomedical problems, and show the pitfalls that can occur when doing computational analysis. The learning objectives are:

1. Describe the basic programming control structures (for, while, if/then) and how to use them together to solve specific problems (computational thinking).
2. Define the basic vocabulary of numerical methods and select the most appropriate numerical method for a given problem.
3. Implement and validate numerical methods for analyzing biomedical signals using differentiation, integration, and interpolation.

To evaluate overall learning gains, an eight-question multiple-choice concept quiz was administered at the beginning and end of the semester.

Early in the semester, students received a dedicated lecture introducing AI and GenAI tools and its application for coding support. The session covered how GenAI models function, example workflows for code generation and debugging, common pitfalls including hallucinated outputs and syntactically incorrect code, and recommended practices for responsible use.

To ensure that all students received initial exposure to GenAI-assisted coding, the use of a GenAI tool was required for the first assigned problem set. For the remaining ten coding assignments, GenAI use was optional. Although students were free to choose which GenAI tool to use, nearly all students used ChatGPT. We did not collect information on whether students used free or paid versions of ChatGPT. For each problem set, students were provided with a unit test. The unit tests were instructor-written python scripts that evaluated student code against predefined inputs and expected outputs, enabling students to iteratively revise their code and verify correctness prior to submission. Following each assignment, students were asked to complete an anonymous brief survey indicating 1) whether they used GenAI, 2) estimated time spent on the coding problem, and 3) optional reflections on their experience. The free-response question at the end of each survey allowed students to elaborate on benefits, challenges, or strategies they employed.

Students were categorized into two comparison groups based on self-reported usage: a GenAI group, consisting of students who reported using GenAI on at least one-third of assignments, and a non-GenAI group, consisting of students who reported using GenAI rarely or not at all. This pre-determined threshold was selected as an inclusive criterion to distinguish students who engaged with GenAI beyond occasional use, while maintaining sufficient group sizes for comparison in a small cohort. Pre- and post-course concept quiz scores, along with changes in student interest in computational biomedical engineering, were compared across groups to evaluate the effects of GenAI integration on learning gains and student perceptions.

Student perceptions of biomedical engineering topics

Student perceptions of the course were assessed through an anonymous end-of-semester survey administered via the course learning management system. The survey included Likert-type items measuring students' interest in biology, medicine, and engineering; perceived usefulness of course material for future coursework and careers; and self-reported confidence in foundational computational and numerical methods covered in the course. Responses were recorded on a five-point scale (1 = strongly disagree, 2 = disagree, 3 = neutral/undecided, 4 = agree, 5 = strongly agree). The survey concluded with open-ended questions inviting students to provide additional qualitative feedback on their learning experiences and on the integration of GenAI in the course.

Statistics

Concept quiz scores and student self-reported assignment completion times were reported as mean ± standard deviation (mean ± SD). Likert-type survey responses are reported as median with interquartile range (IQR) for each comparison group. Nonparametric tests were used to account for non-normal distributions and unequal sample sizes (GenAI: $n = 15$ students; non-GenAI: $n = 8$ students). Pre- to post-course changes in concept quiz scores were analyzed using a Wilcoxon signed-rank test. Between-group differences in self-reported completion times and Likert-scale responses were assessed using a Mann-Whitney U-test. For interpretive purposes, a median Likert score of ≥ 4.0 was considered indicative of agreement or a positive perception of the survey item. Statistical significance was defined as $p \leq 0.05$.

Results

RQ1: Conceptual learning gains did not differ between students who used GenAI or not

Students who reported regular optional use of GenAI and those who did not achieved comparable conceptual learning gains over the duration of the course (**Figure 1**). GenAI users improved from 30.83% ± 18.80 on the pre-course concept quiz to 86.67% ± 13.74 post-course ($p = 0.00122$, Wilcoxon signed-rank test). Similarly, the non-GenAI group increased from 28.13% ± 16.02 pre-course to 90.13% ± 8.83 post-course ($p = 0.00781$, Wilcoxon signed-rank test). Pre-course ($p = 0.633$, Mann-Whitney U-test) and post-course scores ($p = 0.791$, Mann-Whitney U-test) showed no significant difference between GenAI and non-GenAI groups, and multiple students from both groups achieved a score of 100% (8/8 questions) on the post-course quiz. These results for RQ1 indicate that use of GenAI on coding assignments neither hindered nor enhanced conceptual understanding in this Computational Biomedical Engineering course.

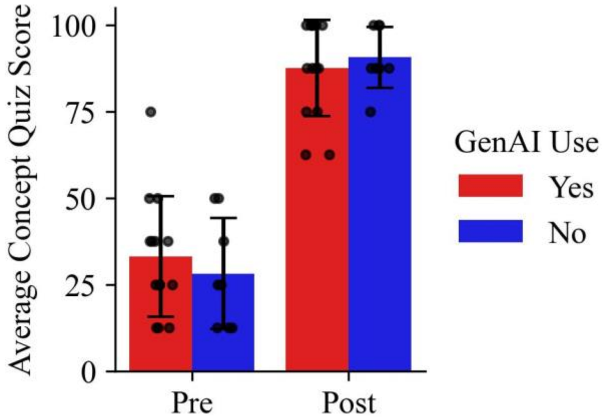

Figure 1. Students in both groups improved from pre- to post-course concept quiz scores. Pre- and post-course concept quiz scores for students who used GenAI (red) and those who did not (blue). Bars represent mean ± SD. Both groups showed significant improvement from pre- to post-course (GenAI: $p = 0.00122$; non-GenAI: $p = 0.00781$, Wilcoxon signed-rank test).

RQ2: Students using GenAI complete coding assignments more quickly than students not using GenAI

Across ten coding assignments, students who used GenAI reported shorter overall completion times compared to students who did not use GenAI. Average completion time for GenAI users was 45.3 ± 61.74 minutes, while non-users averaged 56.82 ± 64.74 minutes ($p = 0.0108$, Mann-Whitney U test). Although this finding was apparent at the aggregate level, assignment-by-assignment comparisons revealed that statistically significant time reductions occurred only for two coding assignments: numerical differentiation (GenAI: 14.82 ± 13.76 minutes; non-GenAI: 45.56 ± 44.68 minutes, $p = 0.0139$) and multi-trapezoidal rule (GenAI: 11.33 ± 7.73 minutes; non-GenAI: 27.50 ± 10.69 minutes, $p = 0.00238$) (**Figure 2**). These results for RQ2 suggest that GenAI use may be associated with shorter reported completion times for a subset of coding problems at the aggregate level. However, this pattern was not consistent across all assignments or individual students. Because completion times were self-reported rather than objectively measured, these findings should be interpreted cautiously and do not imply uniform or definitive efficiency gains, particularly as they may not capture additional time spent verifying, debugging, or correcting AI-generated outputs.

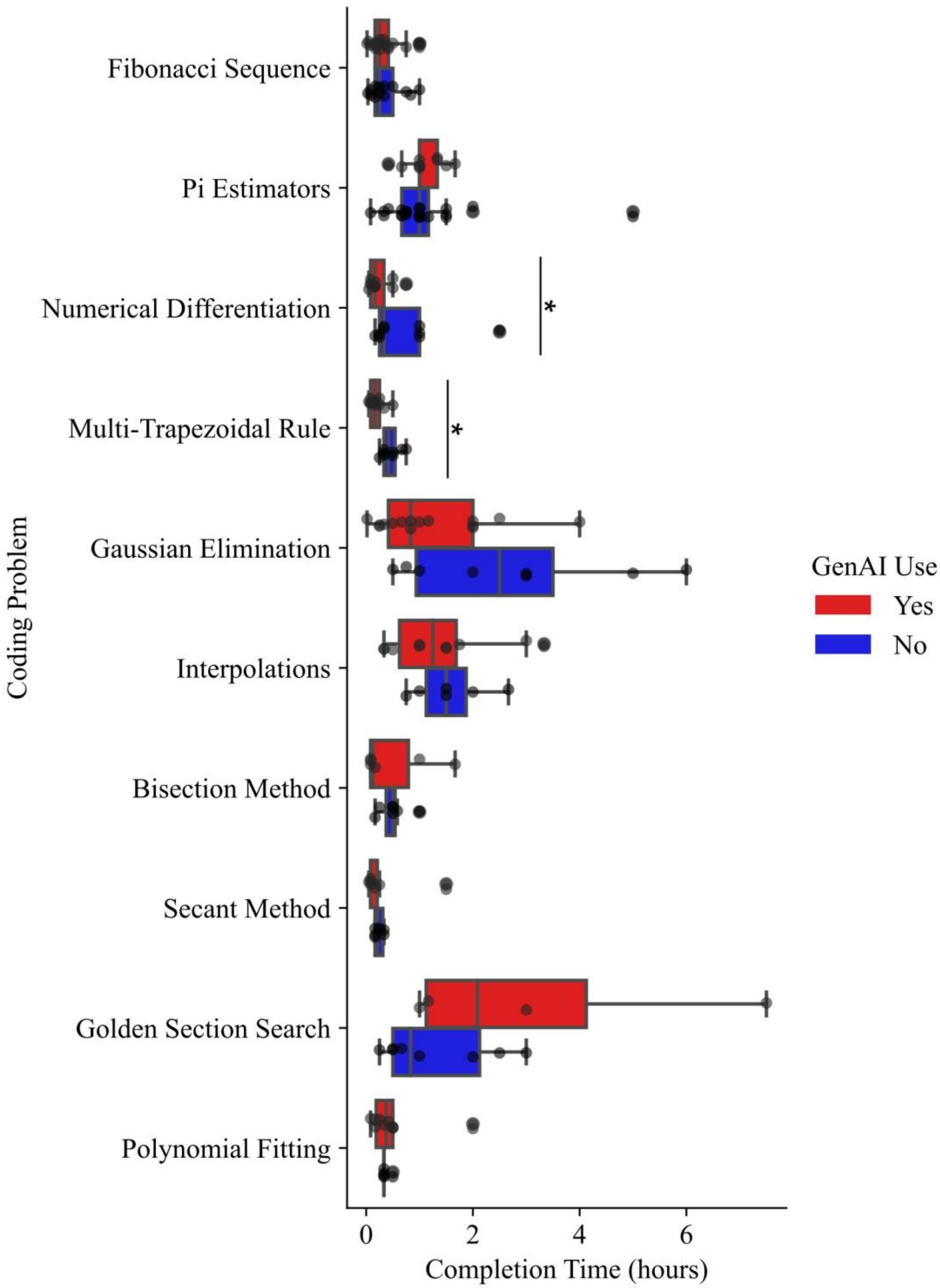

Figure 2. Students using GenAI take less time on average to complete some coding assignments. Box plots show median completion time with 25th and 75th percentiles for each assignment, separated by GenAI users (red, n=4-18) and non-users (blue, n=5-21). Significant differences in completion time occurred only on the numerical differentiation ($p = 0.0139$) and multi-trapezoidal rule ($p = 0.00238$) assignments where GenAI users took less time on average assessed by Mann-Whitney U-test. *$p<0.05$.

RQ3: Student perceptions of biomedical engineering did not differ between students who used GenAI or not

Student perceptions of interest, perceived utility, and self-reported confidence were consistently positive across both GenAI users and non-GenAI users (**Table 1**). Survey questions were intentionally designed to assess general attitudes toward biomedical engineering and confidence in foundational computational concepts, with comparisons made between GenAI and non-GenAI groups. All survey items yielded median Likert scores ≥ 4.0, indicating general agreement or high confidence regardless of GenAI usage. While no statistically significant differences were observed between groups assessed by Mann-Whitney U-tests, these group-level comparisons were limited by small sample sizes and ceiling effects in the response distributions. Despite this limitation, several trends were observed. GenAI users reported slightly higher interest in biology, engineering, and medicine, as well as higher perceived career relevance of the course material, while non-GenAI users reported marginally higher confidence in specific computational skills, including ordinary differential equation solving, Gaussian elimination, and programming control structures. Confidence in fundamental numerical methods such as matrix multiplication and numerical integration and differentiation was high and comparable across both groups. Collectively, these findings for RQ3 suggest that GenAI use neither diminished students' interest in the field nor their confidence in foundational computational biomedical engineering skills, though larger studies will be needed to more conclusively evaluate potential differences in student perceptions attributable to GenAI use.

Table 1. End-of course student perceptions on career interests and confidence in technical topics. Likert-type scores were encoded on a scale from 1 (strongly disagree) to 5 (strongly agree). Median (IQR) reported for each group with p-value from Mann-Whitney U-test for each survey question.

Survey Question	**GenAI**	**non-GenAI**	**p-value**
I am interested in biology.	5.0 (1.0)	4.0 (0.25)	0.133
I am interested in engineering.	5.0 (1.0)	4.5 (1.0)	0.572
I am interested in medicine.	5.0 (1.0)	4.0 (1.0)	0.330
I expect that the material from BME 2315 will be useful to me in future courses.	4.0 (1.0)	5.0 (1.0)	0.502
I expect that the material from BME 2315 will be useful to me in my career.	5.0 (1.0)	4.0 (1.25)	0.322
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in binary numbers?	4.0 (1.0)	4.0 (1.25)	0.972
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in computational methods for solving odes?	4.0 (1.0)	5.0 (1.0)	0.451
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in gaussian elimination?	4.0 (1.0)	5.0 (1.0)	0.251

On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in matrix multiplication?	5.0 (1.0)	5.0 (1.0)	0.490
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in numerical integration and differentiation?	5.0 (1.0)	5.0 (0.25)	0.455
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in programming control structures (if/then, for/while loops)?	4.0 (1.0)	5.0 (0.25)	0.115
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in programming functions?	4.0 (1.0)	4.5 (1.0)	0.330
On a scale of 1-5 (5 being highly confident), how confident are you in your knowledge and skills in Taylor series?	4.0 (0.0)	4.0 (0.25)	0.848

Student reflections on the optional use of GenAI for coding assignments

In addition to the three RQs addressed in this study, voluntary student reflections were collected to examine patterns in students' perceptions of and experiences with the optional use of GenAI tools in the course. For each problem set, students were given the opportunity to provide unstructured feedback on their use of GenAI for coding. Students also had the option to add comments in the course evaluations under any of these three questions:

- Tell us briefly how any of the above learning activities contributed to your learning in this course.
- What would you like the instructor and university administrators to know about your experience in this course?
- What constructive suggestions do you have to help [the professor] improve this course for future students?

Selected student responses from both sources are summarized in **Table 2**. Student comments were first categorized as reflecting either positive or negative attitudes toward the use of GenAI on coding assignments. Within the positive category, a primary theme centered on increased interest in or improved understanding of the GenAI tool itself, with several students noting its ability to generate coding solutions. One student extended this perspective by commenting on the potential usefulness of GenAI tools for future careers. Collectively, these responses suggest a general acceptance of, or favorable disposition toward, integrating GenAI into the course.

In contrast, negative comments primarily focused on challenges associated with using GenAI for coding assignments. A commonly reported issue was encountering hallucinations or incorrect AI-generated outputs, which often resulted in increased time spent on assignments or required students to independently debug their code. While these comments reflected dissatisfaction with GenAI performance, they may also indicate deeper engagement in problem-solving and programming practices as students worked to identify and correct errors.

Student reflections highlight substantial variability in experiences with GenAI use. Together with self-reported completion time data, these qualitative accounts suggest its impact on efficiency is highly context- and user-dependent. Overall, student responses reflected both enthusiasm and frustration toward GenAI integration in lectures and coding assignments. However, these comments represent a subset of students and may not capture all perspectives. Despite these limitations, the reflections provide insight into how students balance perceived learning benefits against the challenges of GenAI use.

Table 2. Selected open-ended student responses collected from assignment surveys and course evaluations regarding GenAI use.

Positive comments on GenAI-aided learning or discussion
• "It was very interesting learning about ChatGpt and other AI tools and how they do not have very good computational skills" • "This was my first time using generative AI and I thought it was really interesting." • "Using chatgpt definitely helps in doing the problems. There are a lot of hallucinations generated but logically working through is helps me learn the concepts and understanding the reasoning better." • "It's interesting how one chat bot can do a problem correctly while another can't. I know we've talked about this, but it's just really surprising answers can be so unique." • "I also think the new content about ChatGPT and other generative AI was super informative and helpful for many contexts that will be applicable to our futures in BME"
Challenges with GenAI use on coding problems
• "Even though I used Chat GPT, I had to edit the code to make the test case work." • "I used ChatGPT for question 2, but it did not help for the approximation function at all, and I wasn't sure how to do it." • "The code given from ChatGPT was not correct, had to edit and figure out myself for the code to work." • "ChatGPT gave very comprehensive code, but it wouldn't work so time was spent debugging the code." • "I initially attempted to use chatgpt to see what it would do. The code chatgpt gave had lots of errors, so I decided to code it myself instead of debugging chatgpt's code." • "It was difficult. I generated my own code and then used ChatGPT to debug it." • "ChatGPT is really hit or miss for these coding problems" • "I tried to use ChatGPT but the code didn't end up working no matter how much I tried to change things. It ended up just wasting time because I had to code #2 myself."

Discussion

GenAI tools are now widely used for computer programming in industry, and it is increasingly likely that students will encounter and use these tools both during their education and in their future careers [1], [13]. Despite this reality, there is concern among educators that GenAI may paradoxically increase efficiency while limiting meaningful learning, particularly if students become over-reliant on AI-generated solutions. As a result, careful consideration is required regarding where and how GenAI tools are introduced within engineering curricula. In this study, we assessed three research questions (RQs) to evaluate whether optional use of GenAI tools for coding assignments affected conceptual learning and disciplinary interests of students enrolled in an undergraduate Computational Biomedical Engineering course. Overall, we found that optional GenAI use did not significantly affect learning gains measured by a pre- and post-course concept quiz (RQ1), nor did it negatively impact time taken to complete coding assignments (RQ2). Additionally, GenAI use did not significantly influence students' post-course interest in biomedical engineering or confidence in computational methods (RQ3).

These findings should be interpreted in light of several limitations. First, both GenAI use and time spent on assignments were self-reported, which minimized course disruption but introduces the potential for recall bias or inconsistencies in how students estimated effort. Second, students self-selected whether to use GenAI on coding assignments, raising the possibility that those who chose to use GenAI differed systematically from non-users in motivation, prior experience, or confidence with programming. While no significant differences in learning outcomes were observed between groups, such unmeasured factors may still influence patterns of GenAI adoption and engagement. Finally, GenAI tools are rapidly evolving, and differences in model behavior or updates over time may affect student interactions with these tools, limiting the generalizability of findings beyond the study context.

Within these constraints, our results align with prior work suggesting that GenAI tools may be most appropriate in courses where coding supports disciplinary learning rather than serving as the primary learning objective [14]. In the context of the Computational Biomedical Engineering course examined here, coding was used to analyze biomedical systems and data rather than to teach foundational coding skills. In this applied context, optional GenAI use did not detract from conceptual learning outcomes, indicating that GenAI can be integrated without undermining course objectives when its role is appropriately defined.

Student feedback further suggested that access to GenAI did not eliminate active engagement with the course material. Many students described approaching AI-generated outputs critically, identifying errors or limitations and validating results through debugging or independent reasoning. These behaviors reflect key elements of self-regulated learning [15], including monitoring, evaluation, and adaptive strategy use rather than passive acceptance of AI-generated solutions. This behavior indicates that students did not uniformly accept GenAI responses as authoritative. In addition, a subset of students chose not to use GenAI for coding tasks at all, despite its availability. This pattern is consistent with prior findings showing a decline in students' perceived usefulness of GenAI over time as confidence and experience increase [16]. Furthermore it aligns with constructs from the Technology Acceptance Model [17], in which perceived usefulness and task relevance shape voluntary adoption.

When students did use GenAI, it was often described as a supportive learning aid rather than a replacement for effort. Prior work has shown that students frequently view GenAI tools as accessible, responsive resources that can provide explanations, suggest alternative approaches, and encourage experimentation [18]. These features may help reduce barriers to entry in computational courses, particularly for students with less prior experience. However, existing evidence also suggests that the benefits of GenAI are not automatic and depend heavily on how its use is framed and supported within the course structure. A recent study in a first-year engineering course demonstrated that students who receive explicit instruction on how to interact with GenAI tools, such as guidance on prompting strategies and evaluating outputs, outperformed students who used GenAI without guidance and those who relied on traditional internet searches [19]. This underscores the importance of instructional scaffolding in GenAI-enabled learning environments, as simply permitting access to GenAI tools may be insufficient to support effective learning. A related consideration is that students may have unequal access to GenAI tools, which could influence learning experiences. Such disparities may occur within a course or across institutions with different resources, underscoring the importance of intentional instructional design and potential institutional support to promote equitable outcomes.

Rather than discouraging GenAI use, instructors can promote productive engagement by emphasizing practices such as verification, iteration, and reflection. These approaches support both effective technology use and the development of self-regulated learning skills [15], [17]. Encouraging students to test AI-generated code, question assumptions, and refine outputs aligns with established engineering problem-solving practices. This approach supports not only technical learning but also the development of professional judgment and awareness, which are essential for navigating real-world biomedical engineering careers.

References

[1] R. Khojah, A. Werth, K. W. Broadhead, L. W. Dobrucki, C. Geiger, and D. A. Rubenstein, "Integrating Generative Artificial Intelligence Tools and Competencies in Biomedical Engineering Education," *Biomed Eng Education*, vol. 5, no. 2, pp. 135–151, Jul. 2025, doi: 10.1007/s43683-025-00175-9.

[2] 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*, in ICER '23, vol. 1. New York, NY, USA: Association for Computing Machinery, Sep. 2023, pp. 106–121. doi: 10.1145/3568813.3600138.

[3] C. Zastudil, M. Rogalska, C. Kapp, J. Vaughn, and S. MacNeil, “Generative AI in Computing Education: Perspectives of Students and Instructors,” Aug. 08, 2023, *arXiv*: arXiv:2308.04309. doi: 10.48550/arXiv.2308.04309.

[4] J. Leinonen *et al.*, “Comparing Code Explanations Created by Students and Large Language Models,” in *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*, in ITiCSE 2023. New York, NY, USA: Association for Computing Machinery, Jun. 2023, pp. 124–130. doi: 10.1145/3587102.3588785.

[5] J. Prather *et al.*, “The Robots Are Here: Navigating the Generative AI Revolution in Computing Education,” in *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*, in ITiCSE-WGR ’23. New York, NY, USA: Association for Computing Machinery, Dec. 2023, pp. 108–159. doi: 10.1145/3623762.3633499.

[6] Y. Luo, E. Chen, H. Liu, K. Riener, and S.-H. Yang, “Generative AI in Engineering Education: A Survey of Student and Instructor Usage and Attitudes,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 69, no. 1, pp. 272–277, Sep. 2025, doi: 10.1177/10711813251358792.

[7] V. Chamola, D. Dave, I. Goyal, and S. Sharma, “Generative Artificial Intelligence in STEM Education: A Review of Applications, Benefits and Challenges,” *Expert Systems*, vol. 43, no. 2, p. e70201, 2026, doi: 10.1111/exsy.70201.

[8] M. R. King *et al.*, “Incorporation of ChatGPT and Other Large Language Models into a Graduate Level Computational Bioengineering Course,” *Cell Mol Bioeng*, vol. 17, no. 1, pp. 1–6, Feb. 2024, doi: 10.1007/s12195-024-00793-3.

[9] A. Lai and K. Karrobi, “Characterizing student adoption of generative AI in technical communication courses,” presented at the 2025 ASEE Annual Conference & Exposition, Jun. 2025. Accessed: Feb. 12, 2026. [Online]. Available: https://peer.asee.org/characterizing-student-adoption-of-generative-ai-in-technical-communication-courses

[10] X. Wang, T. M. Chan, and A. A. Tamura, “A learning module for generative AI literacy in a biomedical engineering classroom,” *Front. Educ.*, vol. 10, Mar. 2025, doi: 10.3389/feduc.2025.1551385.

[11] V. S. Vasudevan, S. D. Archer, and J. T. Butcher, “Exploring the Efficacy of Generative AI and ChatGPT in BME Instructional Labs: A Case Study on GABA Receptors and Synaptic Potentials,” presented at the 2025 ASEE Annual Conference & Exposition, Jun. 2025. Accessed: Feb. 12, 2026. [Online]. Available: https://peer.asee.org/exploring-the-efficacy-of-generative-ai-and-chatgpt-in-bme-instructional-labs-a-case-study-on-gaba-receptors-and-synaptic-potentials

[12] C. E. King and B. A. Lopour, “Fostering Critical Thinking During Use of Generative AI: A Novel Learning Module for Ideation in Biomedical Engineering Design,” *Biomed Eng Education*, vol. 5, no. 2, pp. 397–413, Jul. 2025, doi: 10.1007/s43683-025-00192-8.

[13] N. Rane, “ChatGPT and Similar Generative Artificial Intelligence (AI) for Smart Industry: Role, Challenges and Opportunities for Industry 4.0, Industry 5.0 and Society 5.0,” May 31, 2023, *Social Science Research Network, Rochester, NY*: 4603234. doi: 10.2139/ssrn.4603234.

[14] H. Keuning *et al.*, “Students’ Perceptions and Use of Generative AI Tools for Programming Across Different Computing Courses,” in *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*, in Koli Calling ’24. New

York, NY, USA: Association for Computing Machinery, Nov. 2024, pp. 1–12. doi: 10.1145/3699538.3699546.

[15] E. Panadero, “A Review of Self-regulated Learning: Six Models and Four Directions for Research,” *Front Psychol*, vol. 8, p. 422, 2017, doi: 10.3389/fpsyg.2017.00422.

[16] L. E. Margulieux *et al.*, “Self-Regulation, Self-Efficacy, and Fear of Failure Interactions with How Novices Use LLMs to Solve Programming Problems,” in *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, in ITiCSE 2024. New York, NY, USA: Association for Computing Machinery, Jul. 2024, pp. 276–282. doi: 10.1145/3649217.3653621.

[17] D. Marikyan and S. Papagiannidis, “Technology Acceptance Model: A review,” *TheoryHub Book*, 2024, Accessed: Feb. 18, 2026. [Online]. Available: https://eprints.ncl.ac.uk

[18] S. Groothuijsen, A. van den Beemt, J. C. Remmers, and L. W. van Meeuwen, “AI chatbots in programming education: Students’ use in a scientific computing course and consequences for learning,” *Computers and Education: Artificial Intelligence*, vol. 7, p. 100290, Dec. 2024, doi: 10.1016/j.caeai.2024.100290.

[19] A. Garg, K. Nisumba Soodhani, and R. Rajendran, “Enhancing data analysis and programming skills through structured prompt training: The impact of generative AI in engineering education,” *Computers and Education: Artificial Intelligence*, vol. 8, p. 100380, Jun. 2025, doi: 10.1016/j.caeai.2025.100380.

Appendix A: Concept quiz questions for Computational Biomedical Engineering course.
Correct answers are shown in *italics*.

1. What is the goal of nonlinear regression on a function f(t,p) with parameter p?
    a. To maximize f(t,p)
    b. To minimize f(t,p)
    c. To find a value of p that is a root of the sum of squares error
    d. To find a value of p that maximizes the sum of squares error
    e. To find a value of p that minimizes the sum of squares error

2. Which of the following methods is considered a bracketing or closed method?
    a. Forward-divided difference method
    b. Euler's Method
    c. Simpson's 1/3 rule
    d. Bisection method

3. In the vicinity of a function's maximum, which optimization method converges the fastest?
    a. Golden section
    b. Steepest ascent
    c. Newton's method
    d. Newton-Raphson method

4. Which method can be used to find the maximum of a function when you cannot take its derivative?
    a. Newton-Raphson
    b. Newton's method
    c. Steepest ascent
    d. Golden-section search
    e. Secant method

5. What type of error causes unexpected results when you type 1 - 3*(4/3 - 1) in Python?
    a. Truncation error
    b. Roundoff error
    c. Approximate error
    d. Syntax error

6. Which of the following approximation methods is NOT derived from a Taylor Series expansion?
    a. Newton's method
    b. Gaussian Elimination
    c. Interpolation
    d. Simpson's 1/3 rule

7. Which of the following is the binary version of the smallest non-negative number that can be represented in an 8-bit floating point number (sign, 3 bits for the signed exponent, and 4 bits for the mantissa)?
    a. *11101000*
    b. 01111000
    c. 01101001
    d. 11101001

8. Which of the following methods can be applied to data points, without an explicit equation (e.g. for f(x) or f'(x))?
    a. Adaptive Euler's method
    b. Linear interpolation
    c. Finite difference method for PDEs
    d. Newton-Raphson method