2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #53538

WIP: First-Year Students' Use of AI-Assisted Programming in Open-Ended Robotics Design Problems

Dr. Sandra Walter Huffman, Tufts University

Sandy is a postdoctoral scholar at Tufts University working on Engineering Education research. She is interested in student modeling practices, knowledge practices, and sense-making activities in undergraduate engineering courses. She looks at the ways traditional classroom contexts influence student problem-solving methodologies, and ways task design and facilitation strategies can help students practice authentic engineering. She received her undergraduate and masters degrees in the Department of Mechanical Engineering at MIT, focusing in product design for emerging markets and engineering education research, accordingly. Her interdisciplinary PhD, also based in the Department of Mechanical Engineering at MIT, centered the sociocultural ways students engage with technical engineering material and suggested directions for a more integrated, authentic approach to helping students develop technical modeling practices that utilize course material in unfamiliar contexts.

Sarah Kaczynski, Tufts University

Sarah is a PhD student studying STEM Education at Tufts University. Her research focuses on the design, development, use, and assessment of educational technologies in undergraduate science and engineering courses. She works with Professor Milo Koretsky to support, maintain, and develop tools for teaching and learning, including the Concept Warehouse and a collection of virtual labs.

Dr. Ethan E Danahy, Tufts University

Dr. Ethan Danahy is a Research Associate Professor at the Center for Engineering Education and Outreach (CEEO) with secondary appointment in the Department of Computer Science within the School of Engineering at Tufts University. Having received his graduate degrees in Computer Science and Electrical Engineering from Tufts University, he continues research in the design, implementation, and evaluation of different educational technologies. With particular attention to engaging students in the STEAM content areas, he focuses his investigations on enhancing creativity and innovation, supporting better documentation, and encouraging collaborative learning.

Dr. Merredith D Portsmore, Tufts University

Dr. Merredith Portsmore is the Director for Tufts Center for Engineering Education and Outreach (www.ceeo.tufts.edu).

Dr. Milo Koretsky, Tufts University

Milo Koretsky is the McDonnell Family Bridge Professor in the Department of Chemical and Biological Engineering and in the Department of Education at Tufts University. He received his B.S. and M.S. degrees from UC San Diego and his Ph.D. from UC Berkeley,

WIP: First-Year Students' Use of AI-Assisted Programming in Open-Ended Robotics Design Problems

In this work-in-progress paper, we describe initial results from undergraduates' responses to a survey about their programming/coding experience and use of a custom AI tool within a first-year engineering design course.

AI in University Classrooms

The recent rise in availability and use of generative AI has greatly impacted the educational ecosystem. In response to AI, many educators have begun creating policies for its use, implementing existing generative AI solutions into their classrooms, and even creating and piloting custom generative AI tools. Recent studies have shown that custom AI chatbots can provide more accurate results than generic large language modes (LLMs) such as ChatGPT, Gemini, or Claude. Custom AI chatbots can be tailored to support specific instructional goals [1], [2]. Students report perceiving these tools as helpful to their learning and express using them in a variety of ways. For example, students report using AI as a personal tutor, explaining code and concepts, and creating new ways of studying [2], [3], [4], [5].

Though previous work has investigated the use of custom AI tools for coding tasks within computer science courses and non-coding tasks in other disciplines, the use of custom AI tools for coding tasks in non-computer science courses has not been researched as extensively. A previous study including the AI tool discussed in this paper showed that students can develop coding skills and experience shifts in their attitudes towards AI simultaneously throughout an engineering design course [6]. This work-in-progress paper builds upon that work and provides preliminary insights into how students in an introductory engineering design course use a custom AI tool to write MicroPython code for LEGO Education SPIKE Prime robots.

Course Description and Goals

This study was conducted in a first-year robotics engineering course at Tufts University. The Tufts first-year program consists of 10-12 diverse engineering courses per year that each focus on providing a cornerstone engineering experience to entering students. The robotics class, called EN-1: Simple Robotics, has been offered for the past 15 years using LEGO robotics hardware platforms and primarily text-based coding environments. The primary goal of the course is to provide students with a formative engineering design experience while introducing foundational concepts across multiple domains, including:

- *General Engineering:* engineering design process, engineering disciplines, engineering education pathways, and engineering ethics
- *Mechanics:* structural integrity, gears and gear ratios, and introductory control theory
- *Electronics:* sensors, motors, computer architecture, and communication protocols
- *Computer Science*: program flow and runtime behavior, mathematics and order of operations, functions, variables, data types and structures, loops, and parallel processing

Over the course of the semester students complete 10 one week projects ranging from creating a robotic animal to a Rube Goldberg robotic arm challenge where multiple groups work to pass a ping pong ball from group to group. The course culminates with a multi-week final client-focused project where students develop novel robotic play experiences for young children (age 4-7) that are enrolled at the laboratory school on campus. Students submit documentation of their physical builds using photo, video, descriptions, and code for all projects. They also submit weekly reflections on their experiences.

The primary motivation for the design and implementation of the EN-1 AI chatbot is to support all students' engagement in these robotics design challenges. Students enter the first-year robotics course with a wide variety of coding experiences. Some students have completed AP CS courses while others have never coded. These diverse backgrounds present a challenge in course design. The chatbot was first implemented in Fall 2023 [6] and redesigned for Fall 2025. The EN-1 AI Chatbot is a browser-based code editor for LEGO Education SPIKE Prime robots with an integrated LLM coding assistant. By referencing educator- defined context, documentation, and instructions, the LLM is able to write hardware- compatible code adapted to the user's self-identified experience level.

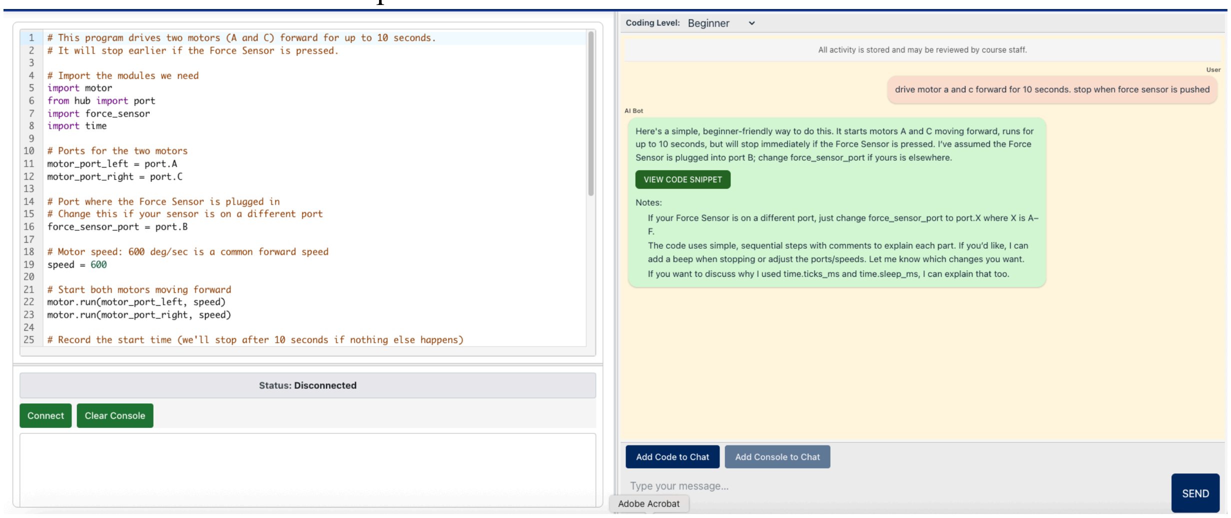

Figure 1: An sample prompt and code snippet from the EN-1 AI Editor

Users can edit and customize the code within the browser allowing students to interact with the code at their ability level. The code can also be sent back to the AI chatbot to ask it for help with explaining or debugging. As coding was not a learning goal of the class, students were encouraged to use the AI chatbot and other AI resources to support their progress.

Research Questions

- Are there differences between the reported coding habits of students with low and high initial coding experience? Do the differences change over time?
- Are there differences in how students with low and high initial coding experience rank their feelings about their coding experience over the semester?
- Are students' feelings about their coding experiences and coding habits related?

Methods

Participants and Data Collection

Nineteen out of thirty enrolled students consented through an IRB-approved process. Course submissions (assignments, surveys, reflections, etc.) and usage logs of the course AI tool were collected for all consenting students. For group projects, video/audio recordings of several groups were captured during in-class worktime. Additionally, five students kept weekly journals where they reflected on the class, and participated in three semi-structured, ~45 min-long interviews over the semester.

For this Work-In Progress paper, we focus on course surveys. Start-of-semester, mid-semester, and end-of -semester surveys were conducted as part of normal class activities. The first survey asked students about their prior experiences (coding, LEGO, design, etc), some of their thoughts regarding AI, and what they are excited and nervous about looking ahead to the semester. The mid-semester survey asked several questions regarding students' coding, AI use, LEGO, and engineering design process experiences, checked in about the course as a whole, and asked what students were excited and nervous about for the second half of the semester. The end-of-semester survey once again asked about coding, AI use, LEGO, and engineering design process experiences. Additionally, it repeated some questions from the intro survey regarding AI use, and asked general questions about the course.

Data analysis

We choose to focus on survey questions related to AI use and coding practices. Using two questions from the first survey (*How much programming/coding experience (any language) do you have?* and *Please list any programming languages you've used in the past (and if appropriate, the context or depth)),* we grouped the students into "high" and "low" initial general coding experience groups. This split students into two evenly sized groups, 9 students with high initial coding experience, and 10 with low initial coding experience.

For each student and group, we traced self-reported coding methods over the course of the semester using the multi-select question *What is your approach to programming/coding?* from the mid-semester survey and the multi-select question *What approach did you use for your final project?* from the end-of-semester survey. Additionally, we analyzed the Likert Scale question *How are you feeling about your programming/coding experience at this point in the course?* from both the mid-semester and end-of-semester surveys for each student, group, and coding method. We used Fisher's Exact, Chi Squared tests, and confidence intervals to determine the statistical significance of any differences, depending on group size. Each of these analyses allowed us to uncover trends and interesting exceptions to those trends. Conclusions made from this survey data will help direct future analysis of coding logs, other course data, interview, and journal analysis for an expanded work.

Findings

Coding Methods vs Initial Coding Experience

Figure 2 shows each students' reported coding method in mid-semester and end-of-semester surveys. For the mid-semester survey, all students claimed that they wrote their code from scratch, wrote code based on sample code from class, or used the class AI tool with some level of modification. None selected "other" or indicated using another source to help them code. The most popular coding method was *using the course AI* with 58% of students selecting that they *used the AI tool and modified the code a bunch* and 63% saying that they *used the AI code without much modification* (6 students selected both options). There does not appear to be a statistically significant difference (Fisher's Exact Test, $p = 0.33$) between the coding approaches of students who cited high levels of coding experience and those who cited low levels of coding experience.

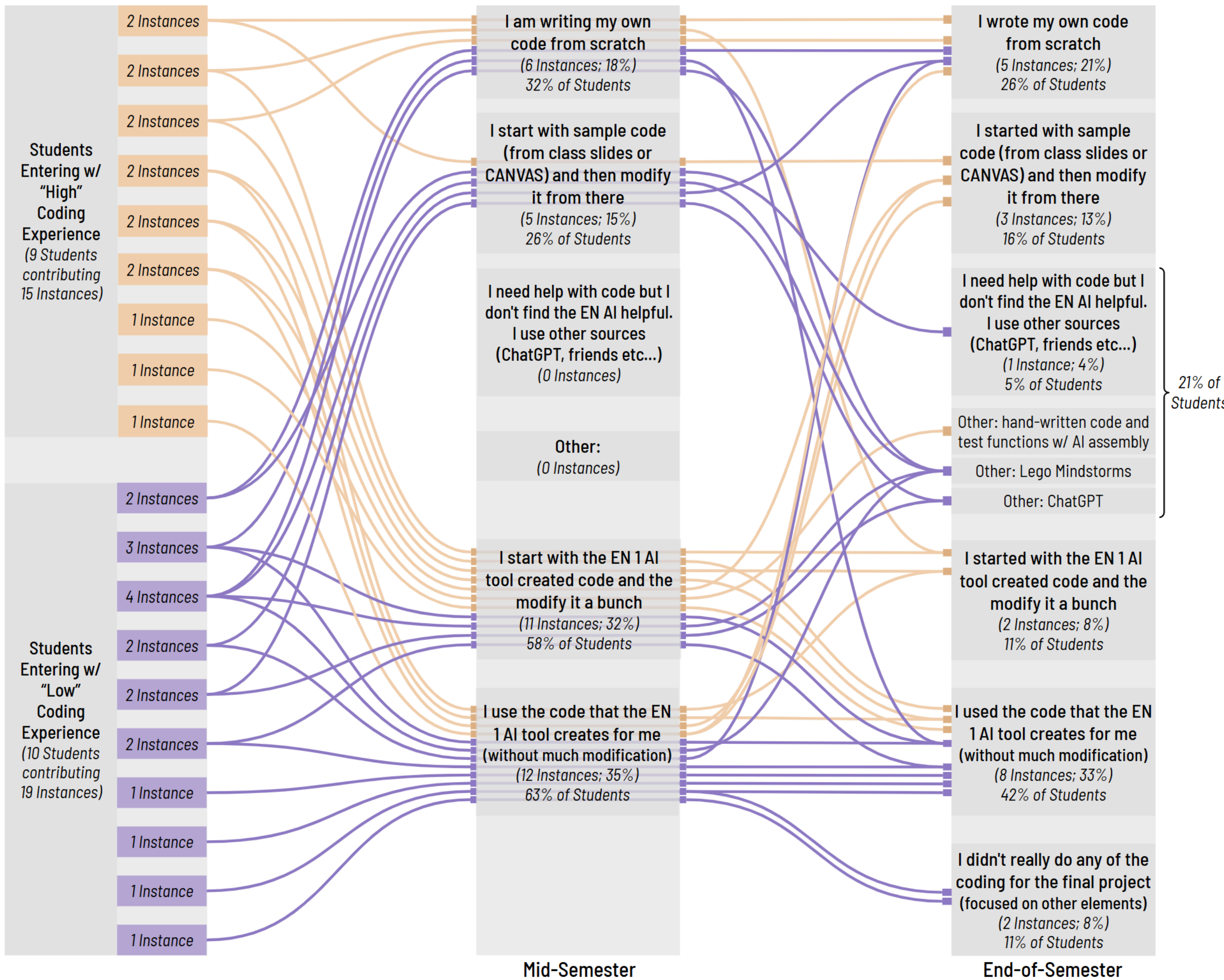

Figure 2: students' reported coding method(s) in mid-semester and end-of- semester surveys

However, $p = 0.09$ using a Fisher's Exact Test between these groups for the end-of-semester survey, indicating a possible difference. We plan to investigate this possible link further in future work. Additionally, there is a statistically significant difference (Chi-Squared Test with "other" "*I need help with code but…*" and "*I didn't really do any of the..*" grouped together $p = 0.016$) between the way students reported coding in the mid-semester survey and the end-of-semester survey. These results suggest that prior coding experience may be connected to how students chose to code for the final project, even though the connection did not appear mid-semester.

The end-of-semester survey also showed an increase in the total number of response types. There were three "other" responses, one "*I need help with the code but don't find…*" responses, and two "*I didn't really do any of the coding…*" responses. There was a notable decrease (58% of students to 11% of students) in the number of students who reported starting with the course AI's code and modifying it a bunch. This decrease was especially true among students with low prior coding experience, none of whom selected this option in the end-of-semester survey. This trend could indicate a change in how students are interacting with the course AI; further analysis of other data sources is needed to understand this change.

The composition of students who selected "started with sample code…" changed from mid- semester to end-of-semester. Mid-semester, five out of six instances were students with little to no prior coding experience, while at the end-of-semester, all three instances were students with significant prior coding experience. These students included only one from the initial group, suggesting changes in approach occurred. Further analysis of coding logs related to this notable shift could help inform an iterative improvement of course resources.

Student Ratings of their Coding Experience

In addition to their coding method, we report students' rating of their feelings about coding by analysing the 5-point Likert-scale question "How are you feeling about your programming/coding experience at this point in the course?" from mid-semester and end-of-semester surveys. Figure 3 shows individual student trajectories for students with "high" and "low" prior coding experience.

No clear trends emerge from these trajectories. This finding is promising, as it shows no obvious distinction between students with high and low prior coding experience. This observation is confirmed by the averages for each group, which are not significantly different at 95% confidence for either the mid-semester or end-of semester rating;

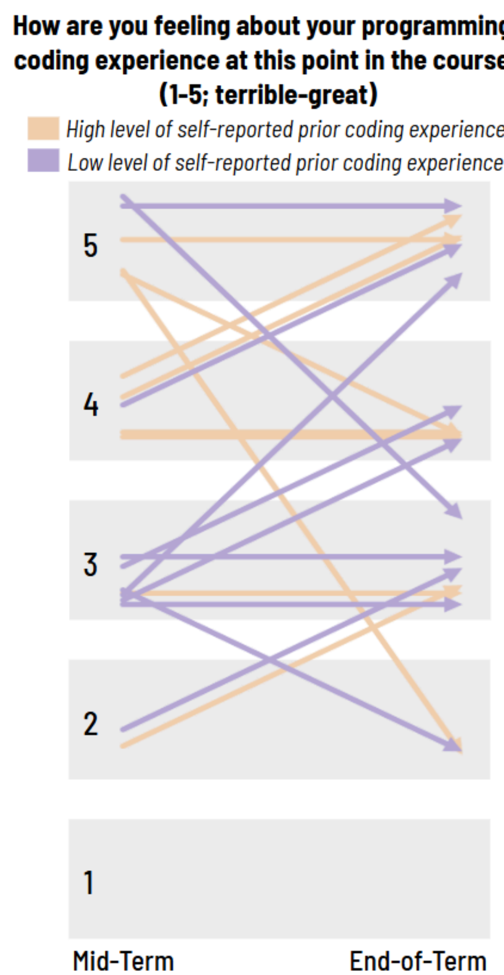

Figure 3: Self-reported rating of students' coding feelings from mid-semester to end-of-semester by "high" and "low" prior level of coding experience

however, this lack of significance may also be due to the small sample size. In future work, we will investigate this further by pairing these ratings with open-ended survey questions, interviews, journal entries, and classroom recordings to determine if any more latent differences emerge between the two groups.

The individual trajectories highlight some students of note. For instance, there is a student with high levels of prior coding experience that dropped from a “5” to a “2” over the course of the semester and a student with low levels of coding experience that dropped from a “5” to a “3.” Additionally, there is a student with low levels of coding experience that rose from a “3” to a “5.” Investigating these students' trajectories further may help reveal particular strengths or pitfalls of the AI chatbot or other class elements.

Student Feelings about Coding Experience vs Coding Method

Lastly, we looked at students’ rating of how they feel about their course coding experience and their reported coding method(s). Figure 4 shows a heat map of feelings about coding rating and coding methods from the mid-semester survey, end-of-semester survey, and combined grouped into “high” and “low” initial coding experience, and all students. All responses other than the primary four were grouped in the last row.

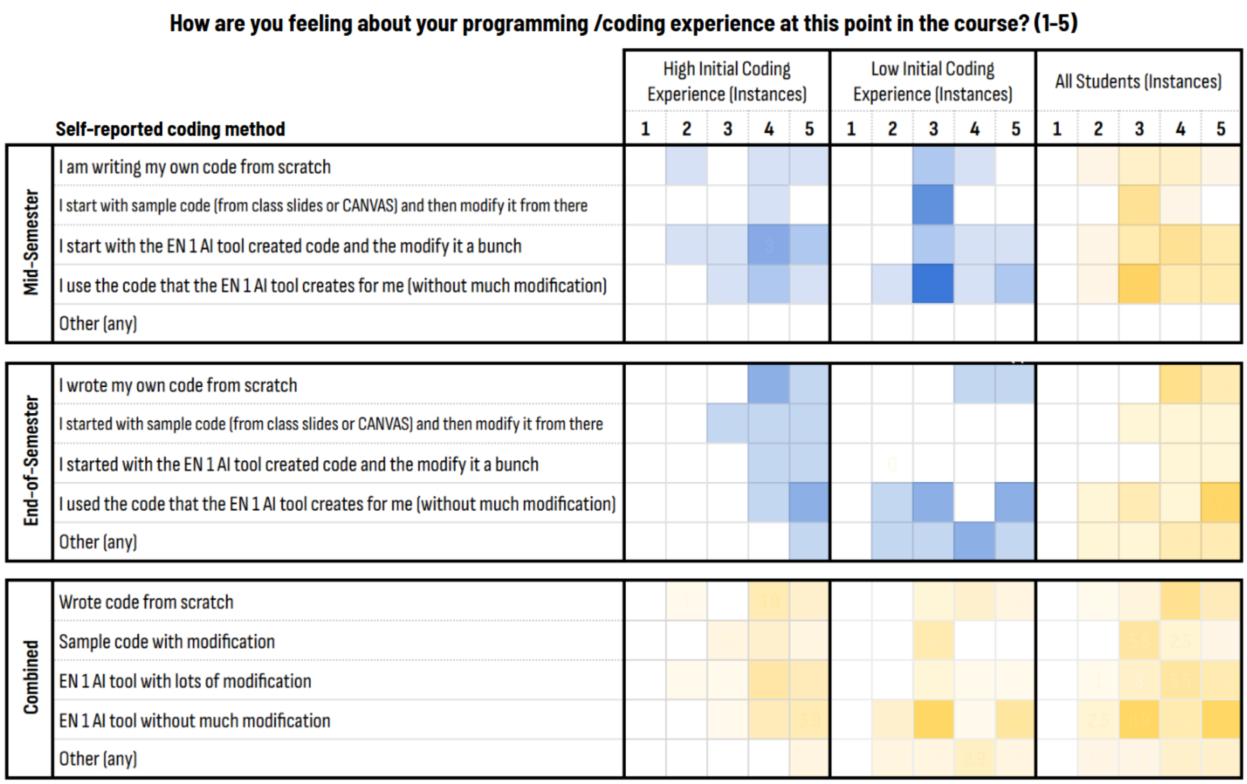

Figure 4: Heat map of self-reported coding method vs self-reported feelings about coding rating for “high” and “low”initial self-reported coding experience groups, and combined. Darker colors represent more instances. “Other (any)” groups all responses shown in Figure 2 outside of the four listed options.

Due to the low number of instances spread across so many categories, no statistical analysis was conducted. However, some trends can be noted. For example, within students who used the course AI tool's code without much modification, most students ranked their feelings about their coding experience as either a "3" or a "5," with many students ranking their experience as "3" in the mid-semester survey and "5" in end-of semester ratings. This shift could be attributed to students getting better at using the tool, improvements in the tool over the course of the semester, students trying out different coding tools early on and then choosing their preferred tool for the final project, or another explanation. Further investigation into other data sources may help explain this trend.

We also observed a difference between groups of students using the AI tool without much modification in the final survey. Students with high prior programming experience who used AI tool without much modification rated their coding experience as a "4" or "5", with more students ranking as "5." Students with low prior programming experience who used the AI tool without much modification rated their coding experience lower, with several "2"s and "3"s represented in addition to some "5"s. While these are small number observations, they raise questions about different students' relationships with the programming tools, and what they are able to use. Further investigation into these students' coding logs, and other qualitative data sources, may reveal more answers.

Table 1 shows the average feeling rating for each coding method per group (high and low prior coding experience). The combined averages for each group are similar, all between 3.5 and 4. There does not appear to be a correlation between coding method and high or low feelings about coding rating. Again, due to low sample sizes, no statistical tests were conducted.

Table 1: Average feelings about coding rating per coding method per group followed by count of number of students in each category.

	High	***Count***	**Low**	***Count***	**Combined**	***Count***
Wrote code from scratch	4.00	*6*	3.80	*5*	3.91	*11*
Sample code with modification	4.00	*4*	3.00	*4*	3.50	*8*
EN-1 AI tool with lots of modification	4.00	*9*	3.75	*4*	3.92	*13*
EN-1 AI tool without much modification	4.29	*7*	3.50	*14*	3.76	*21*
Other (any)	5.00	*1*	3.60	*4*	3.83	*5*

Conclusions and Future Work

There is frequent debate among instructors if students *should* or *should not be* allowed to use AI in their courses. Through our initial analysis, we see a more nuanced story. There were no clear differences between groups' prior programming experience and students' rating of their feelings about course coding experience, or between student rating of their feelings about course coding experience and their coding method(s). These are positive results from the perspective of the teaching team because they suggest students from both groups may have similar experiences in these regards despite their difference in initial coding experience, implying that the AI chatbot is helping the teaching team achieve their goals. Our expanded work will explore other types of data to expand upon this initial conclusion.

Students with high and low prior coding experiences were not stratified in their coding preferences in the first half of the semester; however, differences between the groups emerged in their final project coding preferences. We saw significant shifts in the way students choose to code between the mid-semester and end-of-semester surveys. Notably, students moved away from using the course AI chatbot's code without much modification and towards a wider range of coding methods. Maybe the course AI chatbot helps put students on an even playing field when it comes to the initial projects, but not the final project. Maybe students with less prior coding experience are just more likely to try out more methods as they explore the space. Maybe different resources appeal to different types of students.

We are reluctant to make any conclusions from survey data alone, especially given the small sample size. Next, we plan to investigate other data sources: open-response survey questions, student coding logs, interviews, student journals, course data, and classroom recordings. Coding data logs and course data have the potential to provide details about how students are using AI and the coding practices they are engaging in alongside that tool. The logs contain the students' prompts to the AI chatbot, the content returned by the AI, the students' code and console output, and when they click copy, save, or run. Through that data we can see, for example, how frequently students are engaging with the AI, whether they are replacing their code with that generated by the AI or modifying it (and if so, in what ways), and potential insights into their processes for debugging. Open-response survey questions, interviews, student journals, and classroom recordings will provide a qualitative lens to how these practices connect to student experiences. With this future work, we seek to explore how the AI chatbot can provide rich robotic engineering design experiences even for first-year students with low prior coding experience.

Acknowledgments:
The first author received funding from the McDonnell Family Foundation though the Tufts Institute for Research on Learning and Instruction.

We would like to thank the Koretsky Lab group for their support and thoughtful advice throughout this project. Additionally, we would like to thank Duncan Johnson for his work on and continuous improvement of the course AI platform. Finally, we would like to thank all the students who participated in this study. This work is not possible without you.

References

[1] G. Arun *et al.*, “ChatGPT versus a customized AI chatbot (Anatbuddy) for anatomy education: A comparative pilot study,” *Anatomical Sciences Ed*, vol. 17, no. 7, pp. 1396–1405, Oct. 2024, doi: 10.1002/ase.2502.

[2] R. Liu, C. Zenke, C. Liu, A. Holmes, P. Thornton, and D. J. Malan, “Teaching CS50 with AI: Leveraging Generative Artificial Intelligence in Computer Science Education,” in *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, Portland OR USA: ACM, Mar. 2024, pp. 750–756. doi: 10.1145/3626252.3630938.

[3] 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.

[4] C. Papakostas, C. Troussas, A. Krouska, and C. Sgouropoulou, “A Rule-Based Chatbot Offering Personalized Guidance in Computer Programming Education,” in *Generative Intelligence and Intelligent Tutoring Systems*, vol. 14799, A. Sifaleras and F. Lin, Eds., in Lecture Notes in Computer Science, vol. 14799. , Cham: Springer Nature Switzerland, 2024, pp. 253–264. doi: 10.1007/978-3-031-63031-6_22.

[5] J. Qadir, “Generative AI in Undergraduate Classrooms: Lessons from Implementing a Customized GPT Chatbot for Learning Enhancement,” in *2025 IEEE Global Engineering Education Conference (EDUCON)*, London, United Kingdom: IEEE, Apr. 2025, pp. 1–10. doi: 10.1109/EDUCON62633.2025.11016479.

[6] M. Vora and E. Danahy, “A Student Classification and Characterization Model of Generative AI Use in First-Year Engineering Design,” in *2025 ASEE Annual Conference & Exposition Proceedings*, Montreal, Quebec, Canada: ASEE Conferences, Jun. 2025, p. 55398. doi: 10.18260/1-2--55398.