Generative AI Use in a First-Year Engineering Design Course: Students' Practices in Design Labs and MATLAB.

Ms. Emmanuella Obiageli Ejichukwu, University of Michigan - Dearborn
Dr. DeLean Tolbert Smith, University of Michigan - Dearborn

DeLean Tolbert Smith is an Associate Professor in the Department of Industrial and Manufacturing Systems Engineering at the University of Michigan-Dearborn. Her work intersects research topics related to Black families, design studies, and informal learning.

Generative AI Use in First-Year Engineering Design Course: Students' Practices in Design Labs and MATLAB.

Abstract

Generative artificial intelligence (GenAI) is increasingly integrated into engineering education, yet limited empirical research examines how students use these tools across distinct instructional contexts in engineering courses. Understanding how engineering students use GenAI is crucial for shaping academic integrity policies, ensuring fundamental skill development, and preparing students for an AI-driven industry. The objective of this mixed-methods study is to investigate how first-year engineering students use GenAI in an open-ended engineering design lab project and a MATLAB-based computational programming project, two distinct instructional settings within the same course. Survey responses and open-ended reflections from 118 students were analyzed using descriptive statistics and inductive thematic coding. Results indicate that students preferred to maintain cognitive and creative ownership of their design ideas in the design project; hence, GenAI use was less frequent and more selective. However, the moderate to high use of GenAI in conducting background research suggests that students viewed it as an acceptable support tool for information gathering. In MATLAB programming, students used GenAI more frequently, primarily for debugging and syntax clarification. Across both contexts, GenAI functioned as a scaffold supporting conceptual understanding and verification. These findings demonstrate that students' use of GenAI is shaped by task structure and cognitive demands, suggesting the need for context-sensitive guidance on GenAI use in engineering courses rather than uniform course policies.

1 Introduction

1.1 Background

Generative artificial intelligence (GenAI) tools are increasingly adopted in higher education, significantly shifting how learning, creativity, and engagement with complex concepts are approached (Granić, 2025; Johri et al., 2023; Kadaruddin, 2023). First-year students enter the course with varying levels of experience. The first-year curriculum focuses on building core concepts and equipping students with foundational skills and knowledge for complex problem-solving through hands-on projects (Foust, 2021). Most students are simultaneously developing foundational technical skills and learning to solve open-ended engineering design problems (Ciolacu et al., 2023; Qadir, 2023). Students with limited experience often disengage, lose interest, or focus on finding the right answer (Atman et al., 2005, Litzinger et al., 2011). First-year engineering students demonstrate strong creativity and idea generation but exhibit weaknesses in information gathering, evaluation, and design documentation (Ejichukwu et al., 2024).

Prior work highlights both opportunities and concerns related to GenAI in engineering education, hence the need to understand how students actually use these tools across different learning contexts (Bharathyuvaraj & Masilamani, 2025). However, there are varied students' and instructors' perceptions, and patterns of adoption of GenAI in computational or procedural programming contexts across specific engineering disciplines. The current discourse on GenAI integration in engineering education offers guidelines for effectively integrating it into engineering courses with a uniform context. Despite the growing body of literature on the use of GenAI in engineering education, few studies examine how first-year engineering students engage differently with GenAI across computational and open-ended design tasks in an introductory design course. This insight is needed for student-focused strategies for GenAI integration in engineering education (Walker et al., 2025).

Programming assignments with well-defined criteria may invite different forms of engagement with GenAI than open-ended design activities that emphasize creativity, judgment, and justification of design choices. Without empirical evidence on how students vary their AI use across structured and open-ended tasks within the same course, educators risk implementing uniform AI policies that may misalign with task-specific learning demands. Addressing this gap requires studies that capture how students vary the frequency and nature of GenAI use across different learning contexts within the same course.

1.2 Purpose of the study

Introduction to engineering design courses combine procedural programming, computational problem-solving, and an open-ended design context, while also emphasizing technical communication, collaboration, career exploration, and hands-on engineering practice (Colgan, 2000; Foust, 2021). Given these multiple objectives, the purpose of this study is to examine how first-year engineering students use GenAI across two distinct learning contexts: an open-ended design activity and a procedural MATLAB-based programming task, both within the same course. Using a mixed-methods approach that combines survey data with qualitative analysis of open-ended responses, this study characterizes what tasks students use GenAI for, the frequency of use, and how the identified patterns differ across contexts.

This paper makes three major contributions. First, it provides evidence that students' use of GenAI is highly context-dependent and differs across programming tasks and open-ended design activities within a course. Second, it integrates quantitative and qualitative findings to illustrate how students interpret and negotiate the use of GenAI. Lastly, it offers implications for designing context-specific AI guidance in engineering courses with multiple contexts.

1.3 Research Questions

The research question guiding this study is: How do first-year engineering students use generative AI across MATLAB-based problem-solving and open-ended design activities?

1.4 Theoretical Framework

This study is grounded in cognitivism as a learning theory which views the learning process as an active process of receiving, storing and retrieving information (McLeod, 2003). The human memory has limited working capacity, hence managing intrinsic, extraneous and germane cognitive load is essential for learning (Plass et al., 2010). Cognitivism as a theoretical framework, enables the understanding of how first-year engineering students engage with GenAI across contexts and integrating it into their project workflows. Students are simultaneously learning MATLAB computational logic, syntax, a new computer aided design (CAD) platform, reasoning through the design process, analyzing tradeoffs and making justifiable decisions. Intrinsic cognitive load arises from inherent project complexity, extraneous cognitive load is from unfamiliar design or programming tools, and germane load is the productive cognitive effort required for learning (DeLeeuw & Mayer, 2008).

GenAI use across design lab and MATLAB based programming context is not uniform but shaped by cognitive demands (Paas & van Merriënboer, 2020; Zeitlhofer et al., 2024). Feedback is essential for student's conceptual understanding and skill development (Cardella et al., 2014). As a scaffold, GenAI provides immediate feedback and facilitates conceptual understanding (Qadir, 2023). In this study, GenAI is conceptualized as a tool to help students manage cognitive load. In structured programming tasks, GenAI may reduce extraneous cognitive load associated with syntax interpretation, debugging, and command recall. In contrast, open-ended design tasks require germane cognitive load for schema construction, tradeoff reasoning, and conceptual development. Gen AI can support these processes; however excessive reliance on AI during these stages may reduce productive cognitive effort necessary for deep learning. This framework guides interpretation of differential AI use across both instructional contexts.

2. Methods

This study used a convergent mixed-methods design to examine first-year engineering students' use of GenAI across the MATLAB-based programming context and the open-ended design task. Quantitative survey data and qualitative reflections were analyzed separately and integrated during interpretation. Quantitative survey data characterize students' use patterns and perceptions, while qualitative open-ended responses provide insight into students' reasons and experiences.

2.1 Course context

The first-year engineering education curriculum focuses on building core concepts and equipping students with foundational skills and knowledge for complex problem-solving through hands-on projects. The introduction to engineering design course (ENGR 100) is a required course for first-year engineering students that integrates two distinct instructional components: the MATLAB-based programming and engineering design component. Students enroll in this course with varied levels of preparedness and previous programming or design experience. The MATLAB component equipped students with computational thinking through structured programming, and the engineering design component allows students to identify needs, generate and evaluate design concepts, build and test prototypes, and communicate design decisions through written reports and oral presentations.

Students worked in teams of three to four members to complete two major course projects with different structures. The MATLAB project was a computational task that focused on modeling steady-state heat transfer in a two-dimensional plate using finite-difference methods. The graphical user interface and starter code provided the overall program structure and visualization components, with students responsible for completing designated code sections for grid initialization, iterative updates, and convergence checks. As a result, the task emphasized procedural implementation, debugging, and verification within a constrained problem space. The design project involved an open-ended optimization challenge where students had to design using CAD, 3D print, assemble and program an autonomous vehicle capable of navigating a defined course while carrying an uneven load over set distance. The constraints were to balance multiple competing objectives, such as speed, line following accuracy, stability of uneven load, and efficiency of the overall system.

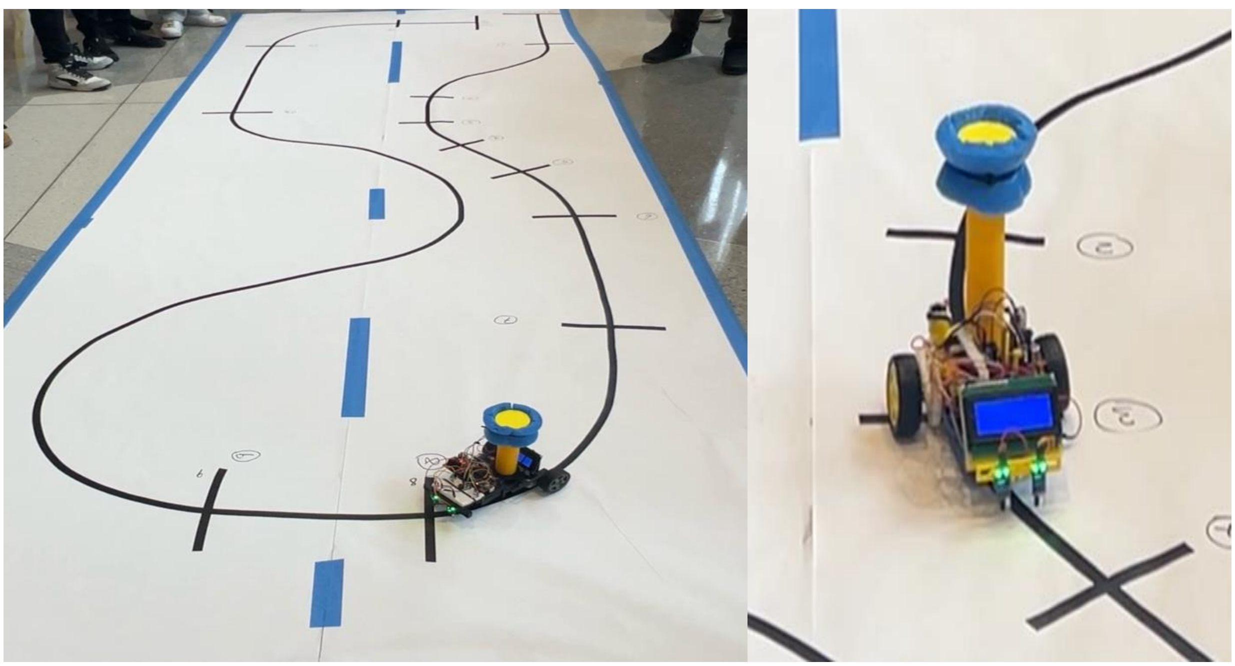

Figure 1: Autonomous vehicle balancing an uneven load and navigating the designated track.

This involved iteratively adjusting design and control parameters, evaluating trade-offs between competing performance objectives, and justifying design decisions based on observed outcomes. Through iterative testing, data collection, debugging, and parameter tuning, teams refined their mechanical design, circuit configuration, and control algorithms to improve overall system performance. The preliminary design review assessed the teams' problem definition, conceptual alternatives, feasibility analysis, initial 3D prototype and part assembly midway through the semester, while the critical design review assessed the finalized prototype performance, validation testing results, optimization trade-offs, technical justification of design decisions, and overall system functionality at the end of the semester. This was done through technical documentation, oral presentation and demonstration of the functioning autonomous system. The final solution reflected a series of data-driven trade-offs across hardware and software components to achieve the best possible performance under given constraints.

2.2 Course Policies and Expectations of GenAI Use

GenAI use was intentionally permitted within a structured instructional framework designed to position AI as a cognitive scaffold. The intended role of GenAI in this course was to serve as an optional scaffolding tool for students with limited knowledge and experience (Wang et al., 2025). The purpose was to help students leverage the potential of GenAI to support interest and engagement in engineering courses. Students were informed about how GenAI outputs may contain errors, incomplete logic, or be misleading. Hence, the need for critical engagement and reflection to ensure correctness, interpretation, and final decision-making. Students were encouraged to first attempt solving the problems independently and, if needed, consult GenAI to gain an understanding of both computational logic and design tasks. They must also critically evaluate GenAI's outputs and not assume correctness. Students were provided with clear guidance on the responsible, transparent, and supportive use of GenAI as a learning aid, not a substitute for independent work or team participation (Quince et al., 2024).

Acceptable use of GenAI for scaffolding during learning includes explaining concepts, clarifying syntax or logic, assisting with debugging, and supporting ideation (Wu et al., 2025). To reinforce responsible use, students were instructed to test AI-generated outputs using edge cases, compare results with course materials, and articulate reasoning prior to implementation. During in-lab check-ins, programming reviews, and design milestones, students explained algorithmic logic, justified implementation decisions, and interpreted debugging steps. These explanation-based validations emphasized understanding and technical reasoning over submission alone. In addition, students documented their AI interactions through structured disclosure prompts requiring them to report what was asked, what was accepted or rejected, and the rationale behind those decisions. These disclosures functioned as metacognitive checkpoints and procedural safeguards that reinforced transparency and critical engagement. Rather than relying on detection technologies, the course embedded assessment redesign and oral validation practices to promote responsible AI use and discourage substitution of learning with automated output.

2.3 Participants and Data Sources

All students enrolled in the first-year engineering design course for the Fall 2025 semester were invited to participate in a voluntary end-of semester survey. Students were informed that their responses would not impact their course grades. A total of 196 students were enrolled in this course, but only 118 students completed the survey. 71% were first-year students, 14% were sophomores, 13% were juniors, and 2% were seniors. The data collected consists of closed-ended survey items measuring the frequency and types of AI use across MATLAB and design lab contexts, Likert-scale items assessing the frequency, perceived benefits, and challenges of AI use, and open-ended responses in which students described how AI supported or hindered their learning. Informed consent was obtained before completing the online survey, in accordance with the institutional review board's requirements.

2.4 Survey Instrument

The semi-structured survey was designed to align with the study's research questions. Survey items were organized in sections corresponding to the two instructional contexts. The survey was administered at the end of the semester, after students had completed all instructional activities and projects, to gather student reflections on their use of GenAI throughout the course. This reduced concerns about grades and allowed students to reflect honestly on their experiences with GenAI across the design lab and MATLAB components of the course. The instructional goals and common challenges faced by first-year engineering students informed the survey design. The survey dimensions are shown in Tables 1 and 2.

The survey also collected data on prior programming experience and familiarity with GenAI. These were used to contextualize the responses and not as predictors, because the analysis does not focus on subgroup comparisons. No identifying information was also collected. The survey measured students' frequency and task-specific use, perceived learning benefits, perceived challenges and risks, cognitive and metacognitive effects, trust in AI outputs, differences in context between MATLAB and design labs, and open-ended reflections on GenAI use.

2.5 Metrics and Data Analysis

2.5.1 Quantitative Analysis

Data analysis focuses on descriptive and qualitative metrics aligned with the post-survey design. Quantitative data were analyzed using descriptive statistics to characterize patterns of AI use and perceptions across contexts. The frequencies and percentages of categorical items were calculated. For comparison, analysis was conducted separately for the MATLAB and design lab contexts.

2.5.2 Qualitative Analysis:

Open-ended responses were analyzed using inductive thematic analysis in NVivo, with codes developed iteratively and refined through constant comparison across responses. A structured qualitative coding process was used to analyze open-ended survey responses to identify recurring patterns in how students described their use of GenAI in the course and their verification practices.

The unit of analysis was the individual student response to each open-ended question. All 118 survey responses were reviewed. Initial descriptive codes were generated inductively based on recurring language patterns related to AI use, verification, and avoidance practices. Codes were refined through constant comparison and grouped into higher-level themes. A subset of responses was independently coded and discrepancies were resolved through discussion to enhance analytic consistency. Coding was conducted using structured spreadsheet analysis. Themes were reviewed in relation to quantitative patterns to support integrated interpretation.

2.5.3 Mixed Methods integration

The analysis emphasizes students' reported experiences in MATLAB and Design Lab rather than changes over time. To integrate the findings, the quantitative results were displayed alongside the qualitative excerpts, linking numerical trends with exploratory narratives and explaining why these patterns emerged. This enabled comparison of quantitative frequency patterns with qualitative narratives explaining why students adopted or avoided AI across contexts. Throughout the analysis, attention was also given to negative cases and divergent perspectives to prevent overgeneralization.

4. Results

This section presents the integrated quantitative and qualitative findings from examining GenAI use across two instructional contexts: the MATLAB-based problem solving and an open-ended design activity in the same course. The results primarily describe the data from the end-of-course survey of a first-year introduction to engineering design course that integrated GenAI as a scaffolding tool to support learning. This section presents findings organized by instructional context to address the research question: How do first year engineering students use generative AI across MATLAB-based problem-solving and open-ended design activities?

4.1 Students' Background and Prior Experience.

The students reported background characteristics related to problem-solving, prior programming experience, familiarity with MATLAB, and proficiency with AI tools. Students were mostly confident in their ability to solve problems. On a 5-point Likert scale, with 1 being the lowest and 5 being the highest, 13% reported a moderate confidence level of 3, 60% reported an above-average confidence level of 4, and 26% reported a very high confidence level of 5 in problem-solving. Most of the students began the course with no programming experience. 36% of the students had no coding experience, 14% had limited experience, 26% had some experience, 11% had significant experience, and 13% had extensive coding experience. This distribution indicates the varying level of experience that students bring to a first-year engineering design course.

The Yes/No question showed that 89% of students had no prior experience with MATLAB, while 11% reported prior experience. Most students were simultaneously learning programming concepts and a new computational platform for the first time. For proficiency working with AI tools, 5% reported very low proficiency, 9% reported low proficiency, 38% reported moderate proficiency, 36% reported above average proficiency, 11% reported high proficiency. This indicates that most students perceive themselves as moderately proficient with GenAI tools, while a few feel unprepared or unfamiliar with them. This highlights a broadly intermediate proficiency in AI. Students were free to choose the GenAI tool that best suited their needs.

4.2 GenAI Use in MATLAB-based Programming Context

Students describe their use of GenAI in MATLAB as just-in-time instructional support. GenAI was mostly used to understand MATLAB concepts or syntax and to debug error messages, with many students reporting 'Sometimes' to 'Often' use. Occasionally, students used it to generate starter code and to check that their solutions meet the assignment requirements. Students with limited programming experience frequently use GenAI to translate problem statements into MATLAB starter code, debug syntax errors, and explain unfamiliar functions, reducing frustration. GenAI was minimally used for optimizing or cleaning code, writing code comments or explanations, or translating pseudocode or logic into MATLAB, suggesting reduced reliance on AI for code refinement and documentation. Students also emphasized a need for explicit guidance on effective prompting. Table 1 presents a breakdown of how students used AI in MATLAB to support learning.

Table 1: GenAI Use Across MATLAB-Based Problem-Solving Context

Survey Items	**Quantitative Patterns of AI use in MATLAB (N = 118)**	**Interpretive Summary**
Understanding concepts or syntax	Never = 15%, Rarely = 15%, Sometimes 32%, Often = 28%, Very often = 8%	AI was frequently used for concept clarification.
Generating starter code or templates	Never = 27%, Rarely = 28%, Sometimes 31%, Often = 11%, Very often = 1%	AI was used occasionally for starter code.
Debugging error messages	Never = 16%, Rarely = 13%, Sometimes 32%, Often = 25%, Very often = 12%	AI was commonly used for debugging support.
Optimizing or cleaning code	Never = 38%, Rarely = 18%, Sometimes 27%, Often = 11%, Very often = 4%	AI played a limited role in code optimization.
Writing explanations or comments in code	Never = 58%, Rarely = 19%, Sometimes 10%, Often = 6%, Very often = 5%	AI was rarely used for code documentation
Translating pseudocode or logic in MATLAB	Never = 51%, Rarely = 21%, Sometimes 16%, Often = 9%, Very often = 1%	AI was infrequently used for logic translation
Checking assignment requirements	Never = 39%, Rarely = 23%, Sometimes 21%, Often = 9%, Very often = 6%	AI was occasionally used for requirement clarification

The interpretive summary column synthesizes qualitative themes and contextualizes quantitative patterns. The students reported using GenAI in the MATLAB context primarily as a procedural support tool for syntax clarification, debugging and understanding MATLAB-specific commands. Students used GenAI occasionally for generating starter code and clarifying assignment requirements, and relied on it far less for code optimization, documentation, or logic translation. Figure 3 below presents a visual summary of the frequency of GenAI use across MATLAB task categories. The bar chart illustrates the relative emphasis on debugging and concept clarification compared to lower frequency uses such as code optimization, documentation, and logic translation.

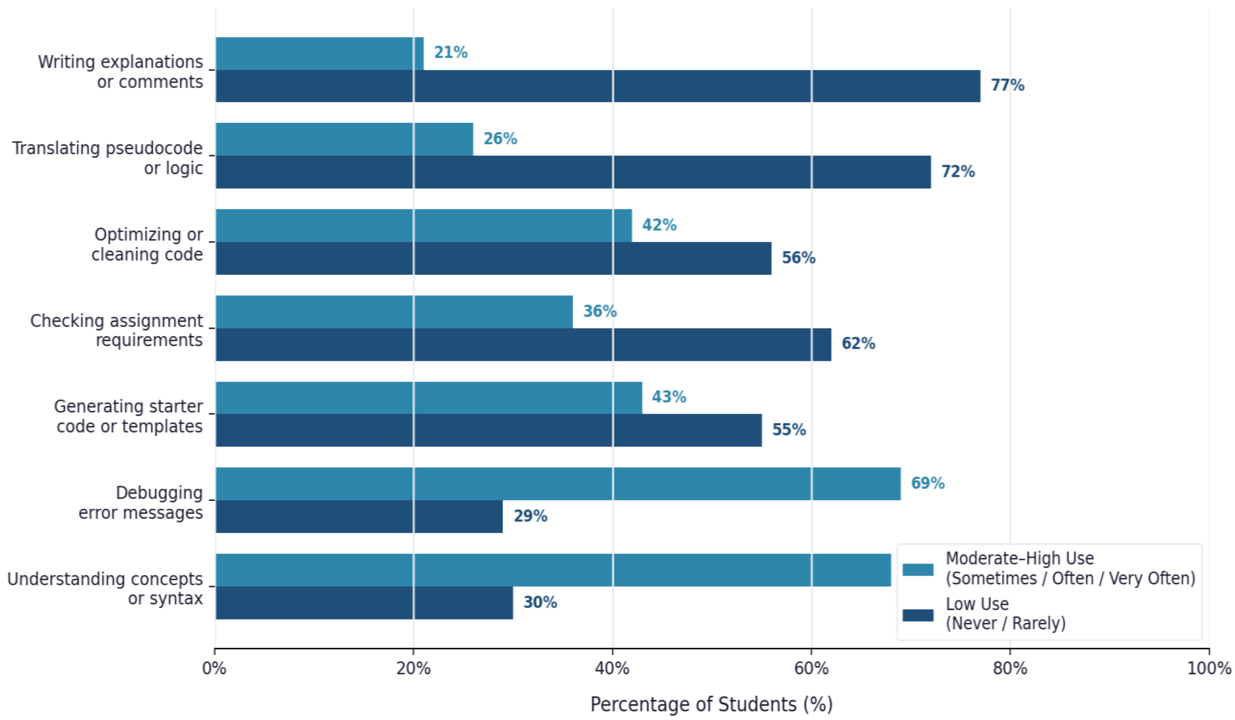

Figure 2:Frequency of GenAI Use Across MATLAB-Based Problem-Solving Context

These patterns in figure 3 suggest that students positioned GenAI as a just-in-time support for overcoming immediate programming challenges rather than as a replacement for core problem-solving processes.

4.3 GenAI Use in an Open-Ended Design Context:

GenAI was used unevenly across stages of the engineering design process in the design lab context. In early-stage design activities, most students reported “never or rarely” using AI for ideation and decision-making. This suggests that teams largely relied on human discussion and judgment when generating and evaluating design concepts. Similarly, GenAI was used only minimally to create visuals, sketches, or diagrams, reinforcing its limited role in the creative and representational aspects of design. When AI was used, the focus was on supporting ideation and brainstorming, verifying trade-off analyses, assisting with testing and validation, presentation-related tasks, and technical documentation towards the end of the design process. Students identified testing and validation as the design stage where AI helped the most, followed by technical documentation and presentation support. Students also reported limited use of GenAI to improve documentation and slides, reinforcing its secondary role in communication. The selective adoption for clarification and validation for specific design stages indicates that GenAI did not drive core design decisions.

GenAI also did not meaningfully alter collaborative practices and team dynamics. Most students indicated that GenAI did not replace team discussions or substantially improve teamwork. Some teams set explicit rules for GenAI use, while most did not, suggesting that AI integration was not strategically planned within teams. Perceptions of outcomes and impact were mixed. Students were largely neutral regarding whether AI improved the overall quality of their final design or accelerated workflow without reducing understanding. However, many students agreed that their teams critically evaluated AI outputs rather than copying them directly, indicating critical engagement and an emphasis on human oversight. These findings show that GenAI use in the design lab is limited for ideation, decision-making, and reflection, and primarily supports documentation and verification in later design stages, with minimal influence on or team interaction. These quantitative patterns and the selective use of GenAI are presented in Table 2 below.

Table 1: Patterns of AI use across Design Problem-Solving

Survey Items	**Quantitative Patterns of AI use in Design Labs (N = 118)**	**Interpretative Summary**
Clarifying the problem requirements	Never = 50%, Rarely = 26%, Sometimes = 24%, Often 14%, Very Often = 3%	AI was used infrequently for problem clarification.
Background Research	Never = 39%, Rarely = 25%, Sometimes = 33%, Often 19%, Very Often = 2%	AI served as an occasional research aid
Brainstorming ideas	Never = 59%, Rarely = 23%, Sometimes = 22%, Often 11%, Very Often = 3%	AI was rarely used for ideation.
Selecting concepts / tradeoff analysis	Never = 65%, Rarely = 27%, Sometimes = 19%, Often 6%, Very Often = 1%	AI played a minimal role in design decisions.
Creating visuals, sketches and diagrams	Never = 85%, Rarely = 17%, Sometimes = 13%, Often 2%, Very Often = 1%	AI was almost never used for visual creation
Writing technical documentation	Never = 76%, Rarely = 23%, Sometimes = 15%, Often 1%	AI use for technical writing was limited
Checking risks and constraints.	Never = 67%, Rarely = 22%, Sometimes = 15%, Often 13%	Limited use of AI for risk evaluation

AI improved the overall quality of our final design.	Strongly disagree = 24%, Disagree = 13%, Neither agree nor Disagree= 57%, Agree = 19%, Strongly agree = 5%	Perceived impact on design quality was neutral.
AI accelerated our workflow without reducing understanding	Strongly disagree = 21%, Disagree = 9%, Neither agree nor Disagree= 48%, Agree = 29%, Strongly agree = 11%	Perceived efficiency gains were mixed.
Our team critiqued AI outputs instead of copying them directly.	Strongly disagree = 11%, Disagree = 10%, Neither agree nor Disagree= 45%, Agree = 34%, Strongly agree = 18%	Many teams critically evaluated AI outputs.

The interpretive summary column synthesizes qualitative themes and contextualizes quantitative patterns. Figure 4 below visualizes the distribution of GenAI use across stages of the engineering design process. The figure highlights the limited use of AI for early-stage ideation and decision-making, compared to its more selective use for background research, validation, and documentation support.

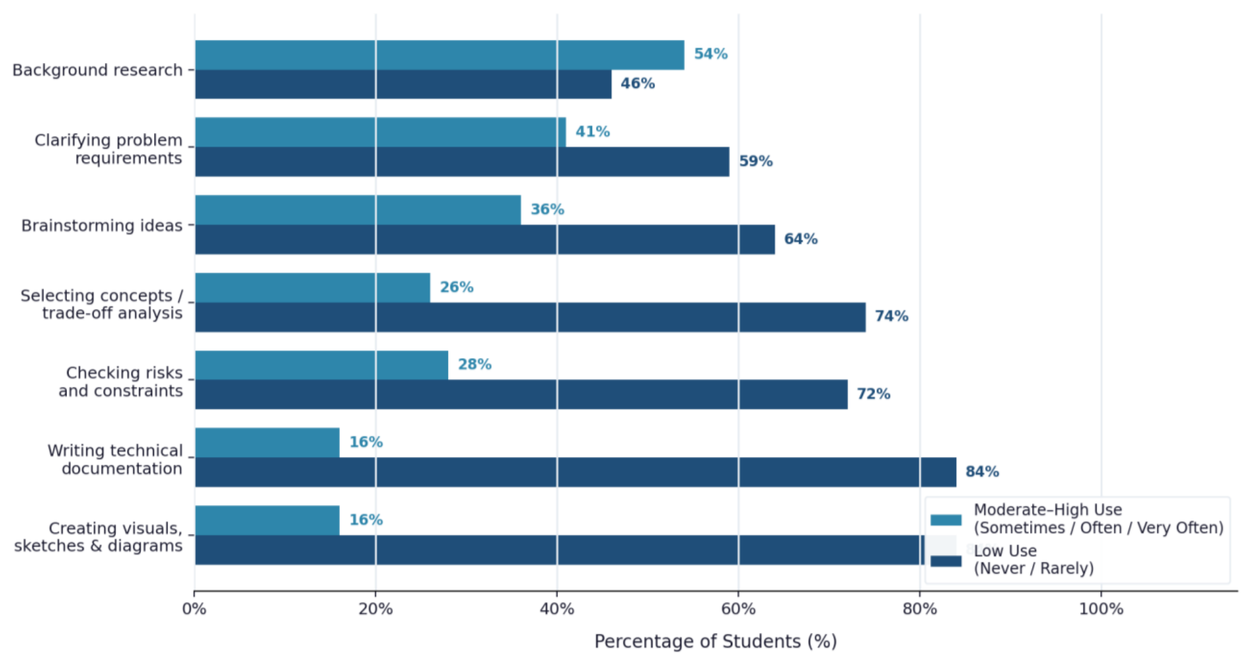

Figure 3: Frequency of AI use Across Design Lab

Generative AI use was minimal during the early stages of the design process, including core design decisions, research, trade-off analysis, and ideation. In the design lab, AI was often described as a supplemental resource for exploration, testing support, validation, and documentation. Students did not find AI helpful for CAD modelling within project constraints. Students cited hallucinations, vague answers, and misleading visuals, which increased their distrust of AI, leading to heavy scrutiny of AI outputs, avoidance, or minimal use. GenAI also required long repeated prompting to gain design context. Students used AI as a tool to verify that design constraints were met, validate code logic, or confirm if the teams were on the right track.

In summary, AI use was more frequent in structured, syntax-dependent programming tasks and less frequent during ambiguity-driven design ideation. This contrast highlights the role of task structure in shaping patterns of GenAI adoption.

5. Discussion

Interpreted through Cognitive Load Theory, GenAI functioned as a context-dependent scaffold. In MATLAB programming tasks, students used GenAI to reduce extraneous load associated with syntax and debugging. In contrast, during germane-intensive design stages requiring ideation, tradeoff reasoning, and schema construction, students limited use of GenAI snd critically evaluation of GenAI output indicate crucial metacognitive monitoring needed to develop technical knowledge. The limited use of GenAI for brainstorming and early-stage ideation is particularly due to concerns about hallucinated or misleading outputs. Students also perceived ideation as a core component of ownership and creativity in engineering design, making them cautious about using GenAI for this. This selective avoidance suggests that students differentiate between procedural assistance and creative authorship, reinforcing the importance of preserving germane cognitive engagement with GenAI use.

The higher frequency of AI use in the MATLAB project reflects the structured and syntax-dependent nature of the task. The heated plate simulation required students to complete designated sections of starter code within a constrained numerical modeling framework. Because most students had no prior MATLAB experience, the task imposed substantial extraneous cognitive load related to syntax interpretation, indexing errors, and debugging. Generative AI was therefore frequently used as a just-in-time procedural tutor to interpret error messages and clarify command structure. In contrast, AI was rarely used for full code generation, optimization, or documentation, likely because the assignment already provided architectural scaffolding and emphasized functional correctness over refinement. This pattern suggests that AI adoption increases when tasks are rule-based and error-driven and decreases when structure and guidance are already embedded within the instructional design.

The observed low frequency of AI use during early-stage design is consistent with the structure of the project. The autonomous vehicle challenge required brainstorming multiple chassis concepts, reasoning through tradeoffs between velocity and balance, and physically prototyping within strict dimensional and component constraints. These germane-load intensive tasks emphasized team discussion, experimentation, and ownership of design decisions. In contrast, later stages of the project shifted toward parameter optimization, debugging, testing, and documentation, which are more structured and procedurally bounded. During these stages, students selectively engaged AI for clarification, validation, and refinement rather than core decision-making. This pattern suggests that AI use increases as tasks become more rule-based and verification-oriented, while remaining limited during ambiguity-driven conceptual design.

Consistent with prior work, first-year engineering students demonstrate creativity, idea generation, and prototyping skills, they exhibited weaker performance in information gathering, problem definition, and design evaluation compared to more advanced students, reflecting the developmental nature of early engineering design knowledge (Ejichukwu et al., 2022). However, students use of GenAI as a tool for background research warrants careful instructional attention. While AI tools can accelerate initial information gathering, reliance on AI-generated summaries may limit direct engagement with primary sources, disciplinary databases, and formal technical documentation. Engineering educators may therefore need to explicitly teach verification practices and source triangulation when AI is used for research-related tasks.

Learners with limited prior experience frequently used GenAI to reduce initial barriers and frustration, particularly in programming contexts. In design, students mostly use GenAI to increase creativity, critical thinking, and problem-solving. Also, students' use of GenAI is influenced by course structure, student factors, and GenAI tool characteristics. The use of GenAI in this study varied substantially as students adapted their use of GenAI to their needs across task types, project contexts, and expectations. In MATLAB tasks, AI was frequently used as a procedural aid to support correctness and efficiency, to translate problem statements into starter code, to debug syntax errors, to explain unfamiliar functions and syntax, and to reduce frustration. This also stayed consistent throughout the semester due to its integration in the programming environment. Whereas GenAI use in the design lab was less frequent in early design stages, more selective, and concentrated in later stages of the design process, such as testing, validation, documentation, and presentation preparation.

Students have varied reasons for using or avoiding GenAI (Dai, 2025). Students' practices varied between MATLAB-based problem-solving tasks and open-ended design lab activities, within the same course. In the MATLAB context, GenAI was primarily valued for improving efficiency, resolving errors, clarifying syntax, and enabling reflection. GenAI was adopted differently depending on whether the task emphasized computational correctness or open-ended design reasoning, emphasizing the importance of context-sensitive approaches to AI integration in engineering education. Many students stated that "AI did not do the work" or was avoided when making decisions. Final decisions were reached through human discussion, consensus, and judgment.

Across both project contexts, students adapted their AI use to align with their individual learning needs and demonstrated behaviors consistent with acceptable uses of GenAI in education as a learning scaffold (Wang et al., 2025). GenAI was used as a secondary resource for verification, validation, and documentation rather than as a driver of decisions. Students described selectively accepting, modifying, or rejecting AI suggestions based on task requirements, constraints, and team consensus. This boundary-setting reflects students' developing understanding of AI's limits and the continued importance of human judgment in engineering work. Clear course policies on AI use reduced ambiguity and discouraged misuse.

Overall, AI use was selective and uneven across design stages. This demonstrates that the use of generative AI among first-year engineering students is not always uniform across project contexts within the same course. This contrast reinforces the importance of examining AI use within specific contexts rather than assuming uniform practices. Open-ended design activities in design lab tasks require students to navigate ambiguity, negotiate constraints, and justify design choices within a team setting. Students engaged with AI more cautiously and reflectively, emphasizing independent creativity, judgment, collaboration, and justification of decisions. For programming tasks, such as those in MATLAB, students positioned AI as a procedural tutor to assist with error resolution, syntax interpretation, and step-by-step reasoning. This reduced frustration, particularly for novice programmers learning foundational programming concepts and a new computational platform. Students consistently evaluated AI-generated outputs and then used other methods, such as verifying them against lecture materials, official documentation, and feedback from peers or instructors, to finalize their solutions.

5.3 Implications for Engineering Education

The differences observed across MATLAB and design lab contexts suggest that task structure plays a critical role in shaping how GenAI is used. In contexts such as MATLAB programming, students' use of GenAI emphasized explanation-focused prompting, debugging support, and verification practices that reinforce conceptual understanding. Students use GenAI to cross-check results against equations, course materials, or documentation, verify whether solutions, code, or designs work, and identify inconsistencies, logic errors, or incorrect assumptions, and selectively accept or reject AI suggestions based on task goals. In design contexts, it was mostly used for validation, documentation, and communication while reinforcing the importance of human-led ideation and decision-making. These patterns of AI use highlight the need to align AI integration with task structure, assessment goals, and developmental stage.

Uniform AI course policies may fail to align task-specific learning demands. In structured programming contexts, instructors may emphasize explanation-based prompting, code walkthroughs, and verification protocols. In open-ended design contexts, educators should require explicit source verification, citation tracing, and triangulation with formal technical references. Designing AI guidance around task structure may better support learning while reducing inappropriate reliance on AI. Assessment practices should reinforce intended AI use.

5.4 Transferability to First Year Engineering Design Courses

Many first-year engineering programs integrate programming and open-ended design within a single course structure. The results suggest that instructions and guidelines on GenAI use should be differentiated based on task structure rather than uniformly applied across course components. To maintain learning integrity in programming contexts, instructors may provide students with guidance on structured prompting and verification protocols whereas in open ended design contexts, emphasize human-led ideation, decision justification, and team based critical evaluation of GenAI outputs allow for design ownership.

7. Limitations and Future Work

The findings of this study describe patterns of AI use in two course projects within the same first-year engineering design course. The data for this study are based on student self-reports collected through end-of-course surveys and open-ended reflections. While these data provide valuable insight into students' reported practices, they do not capture direct measures of real-time AI use. Also, the study was conducted at a single institution within a first-year engineering design course. The project structure and constraints, course policies, and student population may differ from those at other institutions, which limits the generalizability of the findings. Prior programming and AI experience were collected at the end of the semester using retrospective self-report. Although students were instructed to reflect on experience prior to enrollment, retrospective reporting may introduce recall bias. Future studies may collect baseline experience data at the beginning of the semester to reduce potential bias.

While this study focused on describing patterns of generative AI use across instructional contexts, future work will build on this foundation through planned iterations and expansion by examining perceived learning, confidence, and metacognitive awareness. Ongoing work will incorporate artifact-based analyses of student work, including code submissions and design deliverables, to triangulate self-reported practices with observable outcomes. It will also measure student learning and growth quantitatively and qualitatively at strategic points throughout the semester to examine how evolving AI use relates to conceptual understanding, skill development, confidence, and progression in engineering problem-solving. Further analyses will use qualitative methods to examine how first-year student teams negotiate AI use, establish norms for AI use, and balance AI assistance with human judgment in engineering design tasks over time.

8. Conclusion

As generative AI becomes increasingly embedded in engineering practice and education, understanding how students negotiate its use across different task environments is essential for supporting learning, integrity, and professional judgment in engineering education. This study provides empirical evidence that first-year engineering students' use of generative AI is highly context-dependent, varying substantially between procedural MATLAB-based problem-solving and open-ended design lab activities within the same course. By integrating quantitative survey data with qualitative student reflections, the findings show that students adapt their use of AI and selectively engage with it based on task structure, assessment expectations, and self-learning goals. Hence, the need for designing course policies and instructional guidance that account for task structure and learning context, rather than assuming uniform patterns of AI use among first-year engineering students.

References

Atman, C. J., Cardella, M. E., Turns, J., & Adams, R. (2005). Comparing freshman and senior engineering design processes: An in-depth follow-up study. *Design Studies*, *26*(4), 325–357. https://doi.org/10.1016/j.destud.2004.09.005

Bharathyuvaraj, R., & Masilamani, V. (2025). Prospective applications of artificial intelligence (AI) to enhance consistent attention in first-year undergraduate students in the education sector. *IET Conference Proceedings*, *2024*(23), 136–140. https://doi.org/10.1049/icp.2024.4414

Cardella, M., Buzzanell, P., Cummings, A., Tolbert, D., & Zoltowski, C. (2014). A Tale of Two Design Contexts: Quantitative and Qualitative Explorations of Student-Instructor Interactions Amidst Ambiguity. *Design Thinking Research Symposium*. https://docs.lib.purdue.edu/dtrs/2014/Authority/2

Ciolacu, M. I., Alves, G. R., Terkowsky, C., Zoubi, A. Y., Boettcher, K. E. R., Pozzo, M. I., & Kist, A. A. (2023). Developing Future Skills in Engineering Education for Industry 5.0: Enabling Technologies in the Age of Digital Transformation and Green Transition. In M. E. Auer, R. Langmann, & T. Tsiatsos (Eds.), *Open Science in Engineering* (pp. 1019–1031). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-42467-0_94

Colgan, L. (2000). MATLAB in first-year engineering mathematics. *International Journal of Mathematical Education in Science and Technology*, *31*(1), 15–25.

Dai, Y. (2025). Why students use or not use generative AI: Student conceptions, concerns, and implications for engineering education. *Digital Engineering*, *4*, 100019. https://doi.org/10.1016/j.dte.2024.100019

DeLeeuw, K. E., & Mayer, R. E. (2008). A comparison of three measures of cognitive load: Evidence for separable measures of intrinsic, extraneous, and germane load. *Journal of Educational Psychology*, *100*(1), 223–234. https://doi.org/10.1037/0022-0663.100.1.223

Ejichukwu, E., Tolbert Smith, D., & Ayoub, G. (2022). *Assessing Design Process Knowledge In 1st and 3rd-Year Undergraduate Engineers using A Toy Design Project*. Hawaii University International Conferences.

Ejichukwu, E., Tolbert Smith, D., & Ayoub, G. (2024). Assessing design process knowledge in project-based learning: A comparative study in introductory engineering and junior manufacturing courses. *Design Science*, *10*, e37. https://doi.org/10.1017/dsj.2024.29

Foust, E. C. (2021). *Project-Based Learning (PBL) in the First-Year Engineering Design Class*.

Granić, A. (2025). Emerging Drivers of Adoption of Generative AI Technology in Education: A Review. *Applied Sciences*, *15*(13). https://doi.org/10.3390/app15136968

Johri, A., Katz, A. S., Qadir, J., & Hingle, A. (2023). Generative artificial intelligence and engineering education. *Journal of Engineering Education*, *112*(3), 572–577. https://doi.org/10.1002/jee.20537

Kadaruddin, K. (2023). Empowering Education through Generative AI: Innovative Instructional Strategies for Tomorrow's Learners. *International Journal of Business, Law, and Education*, *4*(2), Article 2. https://doi.org/10.56442/ijble.v4i2.215

Litzinger, T., Lattuca, L. R., Hadgraft, R., & Newstetter, W. (2011). Engineering Education and the Development of Expertise. *Journal of Engineering Education*, *100*(1), 123–150. https://doi.org/10.1002/j.2168-9830.2011.tb00006.x

McLeod, G. (2003). *Learning theory and instructional design*. *2*(3), 35–43.

Paas, F., & van Merriënboer, J. J. G. (2020). Cognitive-Load Theory: Methods to Manage Working Memory Load in the Learning of Complex Tasks. *Current Directions in Psychological Science*, *29*(4), 394–398. https://doi.org/10.1177/0963721420922183

Plass, J. L., Moreno, R., & Brünken, R. (2010). *Cognitive Load Theory*. Cambridge University Press.

Qadir, J. (2023). Engineering Education in the Era of ChatGPT: Promise and Pitfalls of Generative AI for Education. *2023 IEEE Global Engineering Education Conference (EDUCON)*, 1–9. https://doi.org/10.1109/EDUCON54358.2023.10125121

Quince, Z., Petkoff, K., Michael, R. N., Daniel, S., & Nikolic, S. (2024). The current ethical considerations of using GenAI in engineering education and practice: A systematic literature review. *Proceedings of the 35th Annual Conference of the Australasian Association for Engineering Education*. https://doi.org/https://search.informit.org/doi/10.3316/informit.T2025032000018100288643600

Walker, C., Krupansky, M., Fowler, R., Alfano, K. M., & Hart, C. (2025, June 22). *Being and Becoming an Engineer: How Generative AI Shapes Undergraduate Engineering Education*. 2025 ASEE Annual Conference & Exposition. https://peer.asee.org/being-and-becoming-an-engineer-how-generative-ai-shapes-undergraduate-engineering-education

Wang, K. D., Wu, Z., Tufts, L., Wieman, C., Salehi, S., & Haber, N. (2025). Scaffold or Crutch? Examining College Students' Use and Views of Generative AI Tools for STEM Education. *2025 IEEE Global Engineering Education Conference (EDUCON)*, 1–10. https://doi.org/10.1109/EDUCON62633.2025.11016406

Wu, J., Wang, J., Lei, S., Wu, F., & Gao, X. (2025). The impact of metacognitive scaffolding on deep learning in a GenAI-supported learning environment. *Interactive Learning Environments*, *33*(9), 5166–5183. https://doi.org/10.1080/10494820.2025.2479162

Zeitlhofer, I., Zumbach, J., & Schweppe, J. (2024). Complexity affects performance, cognitive load, and awareness. *Learning and Instruction*, *94*, 102001. https://doi.org/10.1016/j.learninstruc.2024.102001