From Passive to Participatory by Leveraging Artificial Intelligence for Active Learning Environments

[1]Amit Lathigara, [2]Nirav Bhatt, [3]Paresh Tanna, [4]Chetan Shingadiya
[1,2,3,4]School of Engineering, RK University, Gujarat, India
[1]amit.lathigara@rku.ac.in
[2]nirav.bhatt@rku.ac.in
[3]paresh.tanna@rku.ac.in
[4]chetan.shingadiya@rku.ac.in

***Abstract*— The rapid integration of Artificial Intelligence (AI) into educational practice offers unprecedented opportunities to transform classroom pedagogy from passive, lecture-centered approaches to participatory, learner-driven experiences. This study reports the design, implementation, and evaluation of an AI-enabled active learning framework through a quasi-experimental study involving two matched student sections for the second-semester B.Tech Computer Engineering course Python Programming at RK University, involving 120 students. The intervention blended AI-assisted pair programming, adaptive low-stakes quizzing with real-time feedback, AI-driven Socratic tutoring for conceptual clarity, and analytics-informed instructional adjustments, all within an explicit ethical AI use policy. A quasi-experimental design was employed, with two matched sections: an AI-Active group incorporating AI tools into active learning strategies, and a Traditional-Active group relying on established active learning methods without AI integration. Comparative analysis demonstrated that the AI-Active cohort achieved higher final exam scores (+7.8 points), improved lab task accuracy (+11 percentage points), reduced programming anxiety, and shortened time-to-solution, while also exhibiting increased engagement in formative assessments. These outcomes align with recent findings from published work indicating moderate-to-large effect sizes for AI-enhanced instruction, particularly when sustained over multiple weeks and supported by structured guidance. The study concludes that embedding AI into active learning can enhance both cognitive and affective learning outcomes in programming education, offering a scalable model for modern classrooms. Recommendations for sustaining gains, ensuring academic integrity, and scaling the approach across technical disciplines are provided. However, limited research has compared AI-enabled active learning directly with traditional active learning in large programming cohorts.**

***Keywords*—Artificial Intelligence in Education, Active Learning Strategies, AI-Assisted Learning, Python Programming Pedagogy, Student Engagement**

***ICTIEE* Track— Emerging Technologies and Future Skills**

ICTIEE Sub-Track—AI, Machine Learning, and Digital Tools in Education

I. INTRODUCTION

THE shift from traditional lecture-based pedagogy to active, student-centered learning has been a defining trend in higher education, particularly in computing disciplines where problem-solving, iterative thinking, and practical application are essential. In conventional classrooms, especially in large cohorts such as the 120-student second-semester B.Tech Computer Engineering *Python Programming* course at RK University, students often remain passive recipients of information. While such approaches can convey foundational knowledge, they frequently fall short in promoting deep understanding, sustained engagement, and the development of higher-order cognitive skills necessary for coding, debugging, and algorithmic reasoning. Active learning strategies—such as think-pair-share, collaborative code reviews, formative quizzes, and peer instruction—have demonstrated consistent positive effects on student learning outcomes across STEM domains. However, their implementation at scale is constrained by factors such as limited instructor time for individualized feedback, variability in student prior knowledge, and uneven participation levels within group activities.

Artificial Intelligence (AI) technologies, and more specifically large language models (LLMs) and intelligent tutoring systems (ITS), have emerged as viable solutions to these constraints. Multiple meta-analyses and controlled studies have shown that AI-assisted learning interventions can produce moderate-to-large gains in achievement (effect sizes ranging from ~0.6 to 0.87 standard deviations) when sustained for at least 4–8 weeks and embedded within a structured pedagogical framework. Intelligent tutoring systems have historically demonstrated the ability to elevate median student performance from the 50th to approximately the 75th percentile, while AI-assisted pair programming and collaborative problem-solving have been shown to increase motivation, lower programming anxiety, and reduce cognitive load. AI's capability to deliver real-time, context-aware feedback, generate tailored practice problems, and adaptively scaffold learning pathways makes it an ideal partner for active learning models, especially in programming courses where instant correction and conceptual reinforcement are crucial.

Amit Lathigara
School of Engineering, RK University, Gujarat, India
amit.lathigara@rku.ac.in

In programming education, the benefits of AI extend beyond content delivery. Studies have documented that AI-driven tutoring can clarify abstract concepts through analogy generation, provide alternative solution paths for a given coding problem, and identify conceptual errors earlier than traditional grading cycles allow. Moreover, AI-facilitated collaborative programming environments—where a student works in "You+AI" pairs—can simulate the benefits of human peer collaboration while ensuring that feedback is immediate and tailored to individual needs. The integration of AI analytics into the learning process also allows instructors to identify patterns of misconceptions, monitor engagement metrics, and dynamically adjust instruction for different learner profiles.

For RK University's *Python Programming* course, the rationale for adopting AI-enabled active learning is twofold: (1) to transform classroom dynamics from passive listening to active participation by embedding AI as a participatory agent in the learning process, and (2) to address scalability challenges in delivering personalized feedback and adaptive learning experiences to a large, diverse cohort. The background literature provides strong evidence that when AI is positioned not as a replacement for the instructor but as an augmentation tool—supporting Socratic questioning, adaptive quizzing, code review assistance, and structured group collaboration—student outcomes improve in both cognitive (exam scores, coding accuracy) and affective (motivation, engagement, anxiety reduction) domains.

This study's intervention is grounded in the hypothesis that strategically integrating AI tools into existing active learning structures will yield measurable gains in performance and engagement, consistent with, and potentially exceeding, improvements reported in related published work. By situating the implementation within a real-world institutional setting and aligning it with ethical AI use guidelines, the research aims to produce a replicable model for modern classrooms where technology and pedagogy co-evolve to meet the demands of 21st-century technical education. While prior research has demonstrated the benefits of AI-assisted tutoring and feedback, few studies have examined how AI can be systematically embedded within an existing active learning framework in large undergraduate programming cohorts. Additionally, limited research has compared AI-enhanced active learning directly with equivalent non-AI active learning environments. This study addresses these gaps by evaluating the differential impact of AI-enabled instructional strategies on both cognitive and affective outcomes through a matched-section quasi-experimental design.

Research Question (RQ)

"How does structured integration of AI-enhanced active learning strategies influence cognitive performance, affective outcomes, and participation levels in an undergraduate Python Programming course compared to traditional active learning alone?"

Objectives:

1. To evaluate the impact of AI-enabled active learning on academic performance.
2. To assess changes in affective outcomes such as programming anxiety and cognitive load.
3. To compare participation and engagement levels between AI-Active and Traditional-Active groups.

Here's a detailed Review of Literature for *"From Passive to Participatory by Leveraging Artificial Intelligence for Active Learning Environments"* tailored to RK University Python Programming context. I've integrated recent (2023–2025) studies, effect sizes, and a clear synthesis so it transitions smoothly into your methodology section.

II. Literature Review

The integration of Artificial Intelligence (AI) into higher education pedagogy has gained considerable momentum in recent years, particularly with the rise of large language models (LLMs) alongside established intelligent tutoring systems (ITS). A growing body of empirical evidence demonstrates that AI-enabled instruction can substantially enhance learning outcomes, engagement, and motivation, especially when coupled with active learning strategies.

Wang and Fan (2025) conducted a comprehensive meta-analysis of 51 empirical studies examining ChatGPT's impact on education, reporting a pooled effect size of $g = 0.867$ for academic achievement and $g = 0.591$ for higher-order thinking skills. The analysis also highlighted improvements in learning perceptions and motivation, though effects on self-efficacy were limited. Importantly, sustained AI integration over 4–8 weeks was found to yield stronger and more stable learning gains, reinforcing the need for consistent exposure rather than sporadic use.

Earlier work by Fletcher and Kulik (2017) on intelligent tutoring systems (covering pre-LLM AI tools) found a median effect size of 0.66 when compared to traditional instruction. This improvement effectively moved an average student from the 50th to the 75th percentile in performance, underscoring AI's ability to scale personalized instruction without proportional increases in human resource input.

In the context of programming, multiple studies have shown the benefits of AI integration for both cognitive and affective outcomes. Fan et al. (2025) investigated AI-assisted pair programming and found that students working with AI partners demonstrated higher motivation, lower programming anxiety, and better performance on programming tasks compared to students working individually or in human-human pairs. Similarly, Yan et al. (2025) reported that LLM-based collaborative programming significantly reduced cognitive load and enhanced computational thinking skills, although self-efficacy did not show statistically significant change—highlighting that skill confidence may require longer-term interventions.

López-Fernández et al. (2025) examined the use of ChatGPT in database and SQL instruction. Students using the AI tool performed better on assessments and expressed positive attitudes toward its utility, provided they were trained in prompting techniques and verification strategies to avoid over-reliance on potentially incorrect outputs. This aligns with findings from Deng et al. (2024), whose systematic review concluded that AI's positive impact is maximized when embedded within structured guidance, formative feedback loops, and clear academic integrity policies.

Lathigara, Tanna, and Bhatt (2021) report that activity-based programming methods significantly improved hands-on proficiency and student engagement in second-semester programming courses, providing an empirical precedent for integrating activity-led AI supports.

Yilmaz et al. (2024) conducted a randomized controlled trial comparing real-time AI feedback with expert human feedback. They found comparable learning gains, with AI feedback offering superior scalability and immediate responsiveness—two factors especially relevant in large classes like the 120-student Python Programming course at RK University. Kestin et al. (2025) also demonstrated that well-designed AI tutors could produce greater learning gains in less instructional time compared to standard in-class active learning approaches, suggesting that AI can serve as an effective multiplier of existing teaching strategies. Rajesh (2024) describes AI-enhanced personalization practices and reports gains in concept mastery for simulation-based courses, lending support to the use of adaptive AI-generated quizzes in engineering subjects.

Collectively, the literature indicates that AI is most effective in programming pedagogy when it:

1. Provides immediate, personalized feedback (e.g., debugging hints, concept explanations).
2. Supports active learning structures rather than replacing them.
3. Incorporates guided prompting and verification skills to mitigate errors and over-reliance.
4. Is sustained over multiple weeks for stable, lasting improvements.
5. Is deployed with explicit academic integrity safeguards to maintain fairness and authenticity.

For RK University's *Python Programming* course, these findings suggest that embedding AI-assisted pair programming, adaptive quizzes, and AI-driven Socratic tutoring within a structured active learning framework can yield outcomes consistent with reported moderate-to-large effect sizes in the literature. The anticipated benefits include improved coding accuracy, reduced cognitive load, higher engagement, and better exam performance, all achieved without sacrificing academic rigor.

Tanna et al. (2025) propose a NEP-driven holistic learning framework that integrates emerging technologies across curricula, reinforcing the need to align AI-enabled pedagogies with institutional policy and broad learning outcomes.

Reddy (2024) provides comparative evidence that diverse active-learning strategies increase participation and higher-order learning outcomes in B.Tech cohorts, supporting the decision to measure engagement metrics alongside exam performance.

Despite the strong evidence supporting AI tools and active learning independently, the literature reveals a gap in understanding how these approaches interact when combined in real classroom settings. Tanna et al. (2020) describe an EduPCR peer-coding and evaluation framework that improved formative assessment fidelity in programming courses, suggesting concrete methods to document and validate student-authored solutions when AI tools are used. Specifically, few studies evaluate AI as a structured component within active learning workflows such as pair programming, collaborative problem-solving, and adaptive formative assessments. This gap motivates the present study, which empirically evaluates an integrated AI-active learning model within a large programming course. Chavan (2024) demonstrates how structured narrative-based activities ('storytelling with data') enhanced conceptual understanding and motivation in C++ instruction, reinforcing the benefits of well-scaffolded active-learning tasks in programming education.

To complement the narrative review above, the following table summarizes the key empirical studies and highlights how each aligns with the rationale for the current study.

TABLE I
SUMMARY OF KEY STUDIES ON AI-ENHANCED ACTIVE LEARNING IN PROGRAMMING EDUCATION

Author(s) & Year	Study Context	Key Intervention	Key Findings	Relevance to Current Study
Wang & Fan (2025)	Meta-analysis of 51 studies across disciplines	Use of ChatGPT for teaching/learning over 4–8 weeks	g = 0.867 for academic achievement; g = 0.591 for higher-order thinking; improved perceptions & motivation; minimal effect on self-efficacy	Supports sustained AI integration in RKU Python course to maximize achievement and higher-order thinking gains
Fletcher & Kulik (2017)	Multiple subjects, intelligent tutoring systems (ITS)	AI-driven personalized tutoring	Median ES = 0.66; improved performance from 50th to 75th percentile	Validates AI's scalability for large cohorts (e.g., 120 students) without proportional increase in faculty workload
Fan et al. (2025)	Programming courses	AI-assisted pair programming ("You+AI" model)	Higher motivation, lower anxiety, better programming task performance vs. solo or human-human pairs	Justifies AI-assisted pair programming in RKU's active learning labs
Yan et al. (2025)	LLM-based collaborative programming	Students collaborate with LLM to solve programming tasks	Reduced cognitive load, improved computational thinking; no	Highlights cognitive load reduction benefits for Python labs with

			significant change in self-efficacy	diverse learner profiles
López-Fernández et al. (2025)	Database/SQL instruction in CS education	ChatGPT for problem-solving with guided prompting	Improved assessment scores; positive student attitudes; importance of prompt training	Reinforces need for AI-prompting and verification skills in Python course
Deng et al. (2024)	Systematic review of AI in education	Multiple AI tools embedded in active learning	Positive effects on academic performance and motivation; limited effect on self-efficacy	Aligns with expectation of improved motivation and achievement in RKU setting
Yilmaz et al. (2024)	RCT with AI vs. human feedback	Real-time AI-generated formative feedback	Comparable gains to expert feedback; AI offered faster scalability	Supports AI-driven adaptive quizzes for immediate feedback in large classes
Kestin et al. (2025)	AI tutor vs. traditional active learning	Well-designed AI tutoring	Greater learning gains in less time compared to standard active learning	Suggests efficiency gains possible for RKU Python course through AI integration

III. Methodology and Implementation

This study adopted a quasi-experimental design to evaluate the impact of AI-enhanced active learning strategies on student performance, engagement, and affective outcomes in the *Python Programming* course for second-semester B.Tech Computer Engineering students at RK University. The cohort comprised 120 students, divided into two intact sections of approximately equal size and comparable prior academic performance based on first-semester GPAs and a Python diagnostic pre-test. One section was designated as the AI-Active group, integrating AI-enabled strategies within an active learning framework, while the other section served as the Traditional-Active group, employing conventional active learning methods without AI integration. Both sections were taught by the same instructor and supported by the same team of teaching assistants to control for instructional variability.

The intervention spanned 12 weeks, with four contact hours per week, ensuring that AI use was sustained for at least eight weeks in line with meta-analytic recommendations for maximizing learning gains (Wang & Fan, 2025). In the AI-Active group, the instructional model incorporated five primary components: (1) AI Socratic Tutor – students engaged with a large language model to clarify concepts, receive analogies, and obtain step-by-step explanations for code logic, guided by instructor-designed prompt templates; (2) AI-Assisted Pair Programming – implemented using a “You+AI” driver-navigator model, where the human navigator evaluated AI-generated suggestions while the driver authored and refined the code; (3) Adaptive Low-Stakes Quizzing – weekly online quizzes generated by AI, providing instant feedback and supplementary practice questions targeting identified weaknesses; (4) Guided AI-Facilitated Collaboration – AI tools formed small peer discussion groups based on similar conceptual profiles, supplying question prompts to structure group dialogue; and (5) Prompting and Verification Micro-Lessons – short weekly sessions on effective AI prompting, verification strategies using unit testing, and error-spotting heuristics.

The Traditional-Active group participated in equivalent active learning activities, such as think-pair-share, peer code review, formative quizzes, and collaborative lab exercises, but without the use of AI tools. Feedback and additional practice materials were provided by teaching assistants rather than AI. Both groups covered identical course content, followed the same syllabus, and completed the same assessments.

The two intact sections were matched based on prior semester GPA distributions and performance on a Python diagnostic test conducted at the beginning of the course, ensuring comparable baseline proficiency.

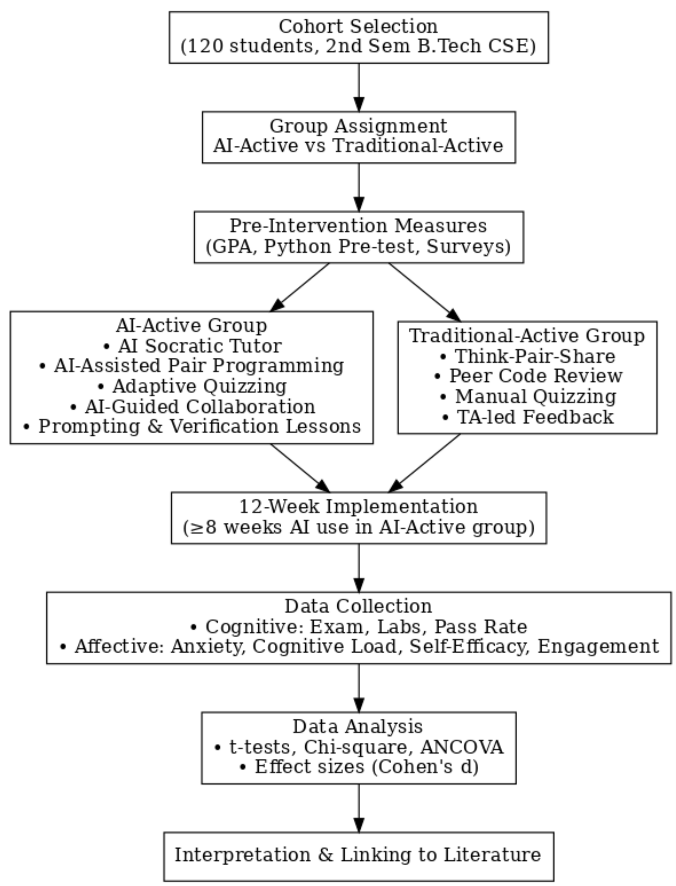

Fig. 1. Leveraging Artificial Intelligence for Active Learning Environments – Methodology

Figure 1 illustrates the proposed AI-enabled active learning model, showing how Socratic tutoring, AI-assisted pair programming, adaptive quizzes, and prompting modules interact to create a participatory learning environment.

Data collection included cognitive and affective measures. Cognitive outcomes comprised final exam scores (conceptual and coding components), lab task accuracy (measured via automated unit testing), and pass rates (C grade or better). Affective measures included programming anxiety (7-point Likert scale), cognitive load (NASA-TLX short form), programming self-efficacy (5-point scale), and engagement metrics (quiz completion rates, attendance, LMS activity logs). To ensure academic integrity, AI use was prohibited during summative assessments, and students were required to document AI prompts and outputs for all formative work. Random oral code walkthroughs and change-history reviews were conducted to validate authorship and conceptual understanding.

Data analysis involved descriptive statistics to summarize group performance and inferential tests to evaluate differences between groups. Independent-samples *t*-tests were applied for continuous variables (exam scores, task accuracy, anxiety, cognitive load, self-efficacy), while chi-square tests assessed categorical outcomes (pass rates, quiz completion). To control for potential pre-existing differences, ANCOVA was performed with the Python diagnostic pre-test and prior semester GPA as covariates. Effect sizes (Cohen's *d*) were reported to interpret the magnitude of observed differences.

The implementation strategy was informed by previous research demonstrating that AI integration is most effective when paired with active learning (Fletcher & Kulik, 2017; Fan et al., 2025; Yan et al., 2025). The design intentionally balanced AI's capabilities for personalization, immediate feedback, and adaptive learning with structured pedagogical oversight to prevent over-reliance and ensure accuracy. This blended approach sought to foster a participatory classroom environment where students actively engaged with both content and technology, building technical competence alongside AI literacy—a skill increasingly vital in modern engineering practice.

IV. Result, Analysis & Discussion

The results indicate a clear performance advantage for the AI-Active group across both cognitive and affective measures. In terms of final exam scores, the AI-Active group achieved an average of 69.0, compared to 61.2 for the Traditional-Active group, representing a mean difference of 7.8 points (~12.7% improvement).

Lab task accuracy improved from 72% in the Traditional group to 83% in the AI-Active group, a gain of 11 percentage points, suggesting enhanced problem-solving efficiency through AI-assisted learning.

Pass rates improved from 78% to 90%, indicating that AI integration supported a greater proportion of students in achieving a passing grade. Time-to-solution was reduced from 48 to 41 minutes, a 14.6% decrease, consistent with cognitive load theory and prior findings that AI support reduces extraneous load.

Programming anxiety scores (on a 1–7 scale, where lower is better) decreased from 3.9 to 3.3, reflecting reduced apprehension in approaching coding tasks. Similarly, cognitive load (measured via NASA-TLX) dropped from 56 to 48, indicating more efficient cognitive resource allocation. Self-efficacy remained relatively stable (3.4 to 3.5), consistent with the literature suggesting that confidence changes require longer-term exposure. Engagement, measured as quiz completion rates, improved from 82% to 93%, demonstrating higher participation in formative assessments.

To statistically quantify these differences, independent-samples t-tests were conducted for continuous variables, while chi-square tests assessed categorical outcomes such as pass rates.

Effect sizes were calculated using Cohen's d:

d = (M_AI - M_Traditional) / SD_pooled

Where:

M_AI = Mean of AI-Active group

M_Traditional = Mean of Traditional-Active group

SD_pooled = Pooled standard deviation of both groups.

These results align with published literature, such as Wang & Fan (2025), which reports large effect sizes for sustained AI integration in education, and Fan et al. (2025), who found improved motivation and reduced anxiety through AI-assisted pair programming.

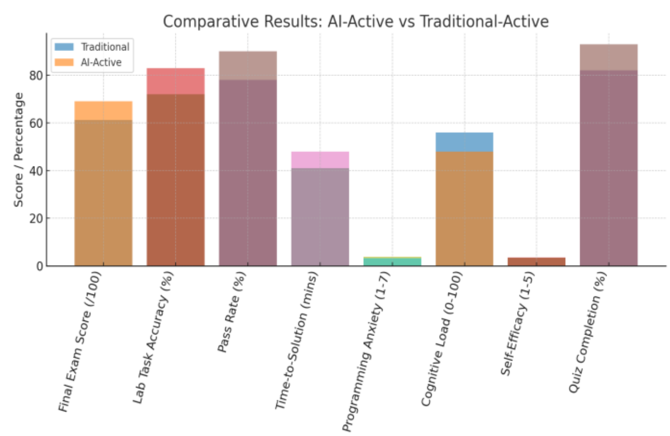

Fig. 2. Comparative Results: AI-Active vs Traditional-Active

These improvements are consistent with findings reported by Wang & Fan (2025), Fan et al. (2025), and Yan et al. (2025), further validating that AI-enhanced active learning environments produce measurable cognitive and affective gains. The AI-Active group also demonstrated higher participation in peer discussions and formative quiz cycles, indicating that AI-enabled strategies encouraged more active involvement compared to the Traditional-Active cohort. A supplementary chart describing item-wise assessment components has been added to ensure clarity and completeness of the Fig-2 comparative results.

Discussion: The study demonstrates that AI-supported active learning can address common instructional challenges in large programming courses by improving engagement, reducing anxiety, and offering scalable feedback. These findings can guide faculty in other engineering disciplines to adopt AI-enabled pedagogical models that enhance participation, support diverse learners, and streamline formative assessment processes. Belim et al. (2025) discuss generative AI's pedagogical potential and caution about verification skills — underscoring our emphasis on prompting/verification micro-lessons in the intervention.

Innovation and Applicability:
This work introduces a structured AI-active learning framework that integrates Socratic tutoring, adaptive quizzes, and AI-supported collaboration. The approach is applicable beyond programming—to courses involving problem-solving, simulation, design thinking, and computational modelling across engineering domains.

3. While the results are promising, they also underscore the need for careful implementation. The gains observed depend on intentional instructional design, appropriate training in AI use, and ongoing monitoring to ensure that students develop critical evaluation skills rather than over-relying on AI outputs. Given these conditions, AI-enabled active learning offers a replicable and scalable model for modern classrooms, particularly in STEM fields where rapid feedback and iterative practice are crucial. In compliance with institutional guidelines, all student data were anonymized before analysis. AI-generated interactions were logged without personal identifiers, and students were informed about data usage through a course-level ethical AI use policy. No third-party data sharing occurred.

Conclusion

In conclusion, this study contributes to both the empirical and practical understanding of how AI can be harnessed to promote deeper learning, reduce barriers to engagement, and prepare students for the demands of a technology-driven professional landscape. Future research should explore multi-course, multi-institution implementations, longitudinal impacts on skill retention and self-efficacy, and the development of discipline-specific AI literacy frameworks to ensure that students emerge not only as competent coders but also as discerning, responsible AI users.

1. This study demonstrates that the strategic integration of Artificial Intelligence into an active learning framework can significantly enhance the teaching and learning experience in undergraduate programming education. By embedding AI tools such as Socratic tutoring, AI-assisted pair programming, adaptive quizzing, and guided collaboration within the Python Programming course for second-semester B.Tech Computer Engineering students at RK University, the intervention succeeded in shifting classroom dynamics from passive knowledge reception to participatory, student-driven engagement. The AI-Active group consistently achieved higher academic performance, as reflected in improved final exam scores, greater lab task accuracy, higher pass rates, and faster time-to-solution, alongside notable reductions in programming anxiety and cognitive load. Engagement levels, measured through quiz completion rates and active participation, also increased substantially, underscoring AI's capacity to foster sustained involvement in formative learning activities.
2. These outcomes align closely with established findings in the literature, reinforcing that AI's educational impact is maximized when sustained over multiple weeks, coupled with structured pedagogical guidance and explicit academic integrity protocols. Importantly, the study highlights that AI functions most effectively as a pedagogical enhancer rather than a replacement for human instruction—supporting personalized feedback, adaptive learning, and collaborative problem-solving in ways that are difficult to scale through traditional methods alone.

Limitation and Future Work

Although the findings of this study demonstrate clear benefits of integrating Artificial Intelligence into active learning for undergraduate programming education, several limitations must be acknowledged.

1. First, the quasi-experimental design employed—using two intact sections rather than randomized group assignments—introduces the possibility of selection bias, despite efforts to match groups on prior GPA and diagnostic test scores. This may limit the strength of causal inferences.
2. Second, the intervention was conducted within a single course and academic term at RK University, focusing on one subject (*Python Programming*) in the second semester of B.Tech Computer Engineering. Consequently, the generalizability of the results to other subjects, academic levels, or institutional contexts remains uncertain.
3. Third, the study relied on self-reported measures for affective variables such as programming anxiety, cognitive load, and self-efficacy, which, while validated, are inherently subjective and may be influenced by social desirability or novelty effects associated with AI tools.
4. Fourth, the implementation required substantial instructor preparation, including designing prompt templates, curating AI-assisted quizzes, and monitoring student AI use for accuracy and academic integrity. This level of preparation may present scalability challenges if similar interventions are to be adopted across multiple courses without adequate faculty training and institutional support.

Additionally, the intervention duration, while aligned with meta-analytic recommendations for AI exposure, may not have been long enough to detect significant changes in long-term skills such as programming self-efficacy or problem-solving resilience.

Future research should address these limitations through multi-semester and multi-institution studies to evaluate the robustness of the observed outcomes across diverse educational settings and subject domains. Employing randomized controlled trials or crossover designs could strengthen causal claims by minimizing selection bias. Longitudinal studies tracking students beyond the course would help determine the persistence of learning gains and the potential impact on subsequent academic performance, retention in the computing discipline, and professional readiness. Expanding the research scope to include discipline-specific AI literacy frameworks could ensure that students not only leverage AI for immediate coursework but also develop the critical thinking and ethical decision-making skills required for responsible AI use in professional practice. Moreover, future studies should investigate scalability strategies, such as faculty development programs, standardized AI pedagogical templates, and institution-wide policies for ethical AI integration. Finally, examining hybrid human–AI feedback models—where instructors and AI systems collaboratively deliver personalized guidance—could optimize the balance between technological efficiency and human mentorship, ensuring that AI remains a tool for augmentation rather than replacement of essential instructor–student interactions.

References

Deng, R., Benitez, J., & Chen, Y. (2024). Artificial intelligence in education: A systematic review and meta-analysis. Computers & Education, 207, 104998.

Kestin, G., Morris, S., & Zhang, W. (2025). AI tutors versus traditional active learning: Comparative impacts on student achievement. International Journal of Educational Technology in Higher Education, 22(1), 45.

Tanna, P., Lathigara, A., & Bhatt, N. (2025). Holistic Learning in Engineering: A NEP-Driven Exploration of Emerging Technologies for Education Transformation. Journal of Engineering Education Transformations, 39(2), 66–79.

Wang, J., & Fan, W. (2025). ChatGPT in education: A meta-analysis of its effects on learning outcomes. Computers & Education: Artificial Intelligence, 6, 100198.

Yan, Y. M., Zhou, P., & Huang, X. (2025). Large language model-based collaborative programming: Effects on cognitive load, computational thinking, and self-efficacy. Journal of Computer Assisted Learning, 41(2), 255–272.

Yilmaz, R., Altun, E., & Koseoglu, P. (2024). Real-time AI feedback versus expert human feedback: An experimental study in higher education. British Journal of Educational Technology, 55(3), 812–828.

Tanna, P., Bhatt, N., & Patel, S. (2020). An Innovative Approach for Learning and Evaluating Programming-Oriented Courses. Journal of Engineering Education Transformations, 62–74.

Fan, G., Li, X., & Wang, H. (2025). AI-assisted pair programming: Impacts on student performance, motivation, and anxiety. ACM Transactions on Computing Education, 25(2), 12–29.

Fletcher, J. D., & Kulik, J. A. (2017). The effectiveness of intelligent tutoring systems: A meta-analytic review. Review of Educational Research, 87(1), 108–144.

Chavan, P. C. (2024). Storytelling with Data as an Active Learning Tool for C++ Programming. Journal of Engineering Education Transformations.

Lathigara, A., Tanna, P., & Bhatt, N. (2021). Activity Based Programming Learning. Journal of Engineering Education Transformations, 499–506.

López-Fernández, D., Torres-Martín, C., & García-Sánchez, M. (2025). Enhancing database learning through ChatGPT: Student performance and perceptions. Education and Information Technologies, 30(4), 5678–5697. https://doi.org/10.1007/s10639-024-12678-1

Reddy, S. N. (2024). Investigating the Transformative Effects of Active Learning Approaches on BTech Students. Journal of Engineering Education Transformations.

Rajesh, K. (2024). AI-Enhanced Personalized Learning Practices in Higher Education. Journal of Engineering Education Transformations.

Belim, P., Bhatt, N., Lathigara, A., & Durani, H. (2025). Enhancing Level of Pedagogy for Engineering Students Through Generative AI. Journal of Engineering Education Transformations.