International Journal of Evaluation and Research in Education (IJERE)
Vol. 15, No. 3, June 2026, pp. 2478~2486
ISSN: 2252-8822, DOI: 10.11591/ijere.v15i3.38919

Instructional scaffolding in dialogue-based programming tutoring

Julieto Perez[1], January Naga[1], Salma Naga-Marohombsar[2]
[1]College of Computer Studies, Mindanao State University-Iligan Institute of Technology, Iligan City, Philippines
[2]College of Natural Science and Mathematics, Mindanao State University-Main Campus, Marawi City, Philippines

Article Info

Article history:

Received Jan 31, 2026
Revised Apr 24, 2026
Accepted May 14, 2026

Keywords:

Artificial intelligence tutoring
Educational evaluation
Instructional scaffolding
Learning technology
Programming education

ABSTRACT

This study examines how instructional scaffolding is enacted in dialogue-based artificial intelligence (AI) tutoring systems for programming education and evaluates the levels of cognitive demand they support. While AI tutors can guide novice learners through programming tasks, it remains unclear whether they promote meaningful higher-order thinking or primarily support procedural task completion. Using a mixed-methods approach, 1,255 tutor utterances from 36 tutoring sessions were analyzed using a dual-layer coding framework grounded in instructional scaffolding theory and Bloom's revised taxonomy. Results show that instructional support is concentrated at the understanding and applying levels, with prompting and explaining as dominant strategies. Higher-order cognitive scaffolding (analyzing, evaluating, creating) was rare or absent. Sequential patterns revealed repetitive prompting–explaining cycles with limited scaffold progression. These findings indicate that AI tutoring effectively supports foundational learning but lacks mechanisms for deeper cognitive engagement. This study highlights the need for pedagogically informed AI tutor design and provides actionable insights for educators and system developers to integrate AI tools in ways that promote higher-order thinking and independent problem-solving.

Corresponding Author:

January Naga
College of Computer Studies, Mindanao State University-Iligan Institute of Technology
Iligan City, Philippines
Email: january.naga@g.msuiit.edu.ph

1. INTRODUCTION

Dialogue-based tutoring systems are a promising approach for supporting learning in complex domains such as programming in higher education. Recent studies on artificial intelligence (AI) in education highlight the growing use of AI-driven tutoring systems for adaptive learning, personalized feedback, and intelligent assessment [1]. These systems support learning through natural language interaction, allowing tutors to guide reasoning and provide timely feedback. Such environments encourage learners to explain their thinking and engage in multi-turn dialogue, which can improve learning outcomes, especially in tasks that require reasoning. Research also shows that integrating diagnostic assessment and student modeling allows instruction to adapt to learner responses, enabling more personalized learning [2], [3]. Compared with static materials, dialogue-based systems promote active engagement through questioning, explanation, and feedback, which can support deeper learning [4].

Programming education presents challenges for novice learners. Beginners often struggle to translate problems into correct program structures, develop mental models, and reason about program behavior. These difficulties lead to misconceptions, logical errors, and high cognitive load, especially when working with variables, conditionals, and loops [5], [6]. Instructional scaffolding helps address these challenges by providing guidance that is gradually reduced as learners gain competence [7]‒[9]. This aligns with the zone of proximal development (ZPD), where learners perform tasks with support before achieving independence [8]. Bloom's revised taxonomy provides a framework for examining cognitive demand, from basic recall to higher-order thinking such as analyzing, evaluating, and creating [10].

The increasing use of AI tutoring systems in higher education raises questions about instructional quality. While prior research highlights their potential for personalization and feedback [11], [12], most studies focus on system performance or learning outcomes. Less attention has been given to how scaffolding is implemented in tutoring dialogue and how it shapes cognitive engagement [13]‒[16]. A key limitation is that many AI programming tutors emphasize procedural knowledge, such as syntax and step-by-step problem solving, with limited support for higher-order thinking. This can lead to surface-level learning, where students complete tasks but struggle with abstraction, evaluation, and creative problem solving. These skills are essential for real-world programming. Without careful design, reliance on AI tutoring may limit critical thinking and independence. This issue is important in programming education, where learning requires both correct solutions and deep understanding. Without systematic evaluation of scaffolding and cognitive demand, educators lack guidance on whether AI tutors support or constrain deeper learning.

Research on instructional dialogue shows that interaction can improve understanding when learners actively explain and respond to questions [16], [17]. Scaffolding theory also shows that effective support adapts to learner needs and promotes independence [7], [18]. However, recent studies suggest that AI tutoring systems often rely on directive or surface-level guidance unless pedagogical principles are embedded in their design [13], [19], [20].

This study addresses the question: how is instructional scaffolding enacted in dialogue-based AI programming tutoring, and what levels of cognitive demand does it promote during problem solving? A mixed-methods approach was used to analyze 1,255 tutor utterances from 36 sessions using a dual-layer coding framework based on scaffolding theory and Bloom's revised taxonomy. This study provides a structured way to evaluate instructional quality in AI tutoring and offers insights for educators, instructional designers, and system developers.

2. METHOD

2.1. Research design

This study adopted a mixed-methods research design to evaluate instructional scaffolding in dialogue-based tutoring interactions. The analysis combined quantitative coding of tutor utterances with qualitative examination of instructional dialogue. A dual-layer coding framework was used, integrating instructional scaffolding strategies with Bloom's revised taxonomy [10] to evaluate both the type of instructional support and the level of cognitive demand. The evaluation focused on instructional behavior rather than learner performance outcomes or technical system accuracy. This methodological approach is particularly important for educational evaluation because it moves beyond measuring system performance or learner outcomes and instead examines the instructional quality embedded within AI tutoring dialogue. By analyzing how scaffolding strategies align with cognitive demand, the study provides a more comprehensive understanding of whether AI systems support meaningful learning processes rather than merely facilitating task completion.

2.2. Research setting and participants

The study was conducted in an introductory programming course at a public university. Participants were novice programming students who voluntarily engaged with a dialogue-based tutoring system as a supplementary learning tool during problem-solving activities. The tutoring system supported learners as they worked on structured programming tasks involving fundamental programming concepts such as input handling, conditional statements, and loops. A total of 36 tutoring sessions were included in the analysis. Sessions were selected based on completeness, defined as sustained interactions containing multiple instructional exchanges related to a programming task. All interaction data were anonymized before analysis to protect participant privacy and ensure ethical compliance.

2.3. Data collection

Tutoring interaction logs were collected automatically by the learning platform. Each log contained sequential dialogue between the tutor and the learner, including instructional prompts, explanations, questions, and feedback provided during problem-solving activities. Sessions with minimal interaction or incomplete dialogue were excluded. The final dataset consisted of 1,255 tutor utterances across the 36 sessions. Tutor utterances were selected as the unit of analysis because the study focused on evaluating the instructional support provided during tutoring interactions, rather than learner responses or overall session outcomes.

2.4. Coding framework
Instructional support was evaluated using a dual-layer coding framework grounded in educational theory. The first coding layer classified tutor utterances according to instructional scaffolding strategies, including prompting, questioning, explaining, instructing, modeling, and feedback. These categories reflect established forms of instructional support commonly used in tutoring and classroom instruction. The second coding layer classified each tutor's utterance according to it is cognitive demand using Bloom's revised taxonomy. Cognitive levels included remembering, understanding, applying, analyzing, evaluating, and creating. Each utterance was assigned a single dominant cognitive level based on the instructional intent conveyed in the dialogue. The instructional scaffolding strategies and cognitive levels used in the analysis are summarized in Table 1, which presents the dual-layer coding framework applied to tutor utterances. This framework enabled examination of how instructional scaffolding strategies were distributed across different levels of cognitive complexity.

Table 1. Dual-layer coding framework for instructional scaffolding and cognitive levels

Bloom's cognitive level	Cognitive objective	Instructional scaffolding strategies	Illustrative tutor actions
Remembering	Recall facts, definitions, or syntax	Cueing, direct instruction, corrective feedback	Reminding learners of programming keywords or syntax rules
Understanding	Explain concepts or interpret meaning	Explaining, paraphrasing, clarification questions, concept checking	Clarifying the purpose of a variable or explaining how a condition works
Applying	Use knowledge to complete tasks	Prompting, worked examples, guided practice, feedback	Guiding learners to apply rules when writing or modifying code
Analyzing	Differentiate, organize, or identify relationships	Questioning, error identification, comparison prompts	Asking learners to locate logic errors or compare solution approaches
Evaluating	Justify decisions or assess solutions	Justification prompts, reflective questioning, feedback	Encouraging learners to explain why a solution is efficient or correct
Creating	Generate new or original solutions	Open-ended prompting, minimal guidance, scaffold fading	Encouraging learners to design their own functions or alternative solutions

2.5. Coding procedure and reliability
All tutor utterances were coded using a combination of rule-based preprocessing and manual review. Initial coding was conducted by the research team using a predefined coding guide. To support consistency, utterances were first organized using instructional cues to assist the manual coding process. All classifications were subsequently verified through human review to ensure accuracy and reliability. To ensure reliability, a subset of tutor utterances was independently reviewed by more than one researcher. Discrepancies in coding decisions were discussed and resolved through consensus, and the coding guide was refined as needed. This iterative process helped maintain internal consistency across the dataset.

2.6. Data analysis
Quantitative analysis focused on descriptive statistics, including frequency distributions of scaffolding strategies and cognitive levels. Transition patterns between instructional strategies were also examined to explore how instructional support evolved during tutoring interactions. Inferential statistical testing was not applied, as the purpose of the study was to evaluate instructional behavior rather than to test causal relationships or generalize findings to broader populations. Qualitative analysis involved close examination of representative dialogue excerpts to illustrate common instructional patterns and notable deviations. These excerpts were used to contextualize quantitative findings and to examine how instructional scaffolding supported or constrained learner cognitive engagement during problem-solving activities.

3. RESULTS AND DISCUSSION
3.1. Results
This section presents the findings from the analysis of instructional scaffolding in dialogue-based programming tutoring interactions. A total of 1,255 tutor utterances from 36 tutoring sessions were examined using a dual-layer coding framework.

3.1.1. Distribution of instructional scaffolding strategies
The distribution of instructional scaffolding strategies is shown in Figure 1. Prompting was the most frequently used strategy. It accounted for the largest share of tutor utterances and was used to guide learners through procedural steps, such as focusing on code segments or encouraging iterative testing. Explaining and

questioning were the next most common strategies. Explaining clarified programming concepts and syntax, while questioning elicited reasoning or checked understanding. In contrast, instructing, modeling, and feedback appeared less often. These results show that instructional support was mainly directive and focused on task completion rather than demonstrating expert reasoning or encouraging reflection.

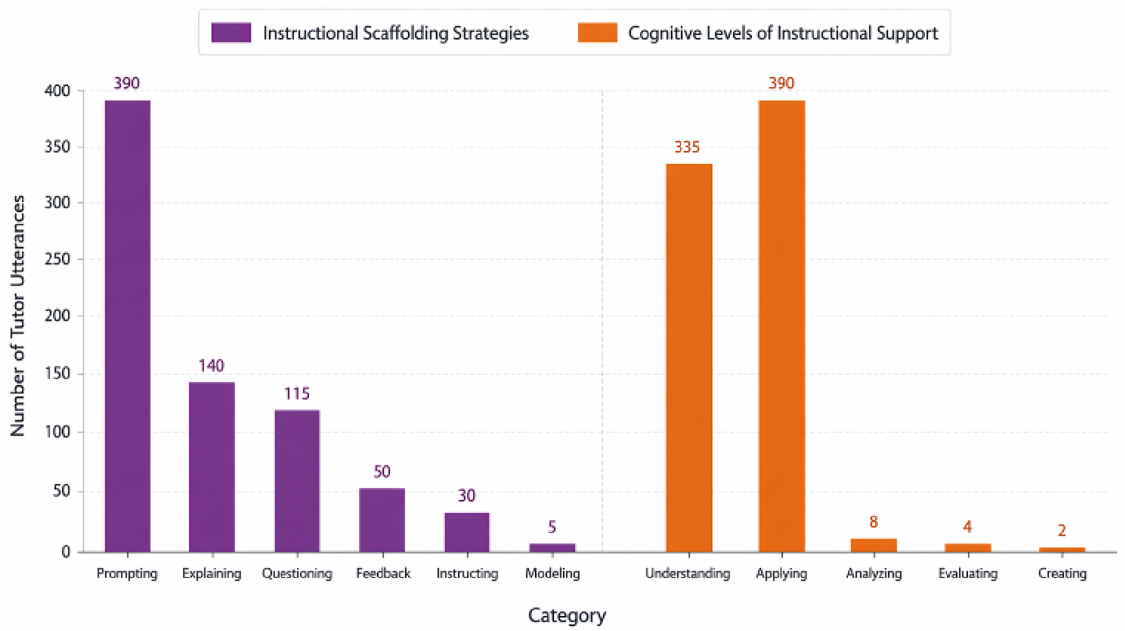


Figure 1. Distribution of scaffolding strategies and cognitive levels in tutor utterances

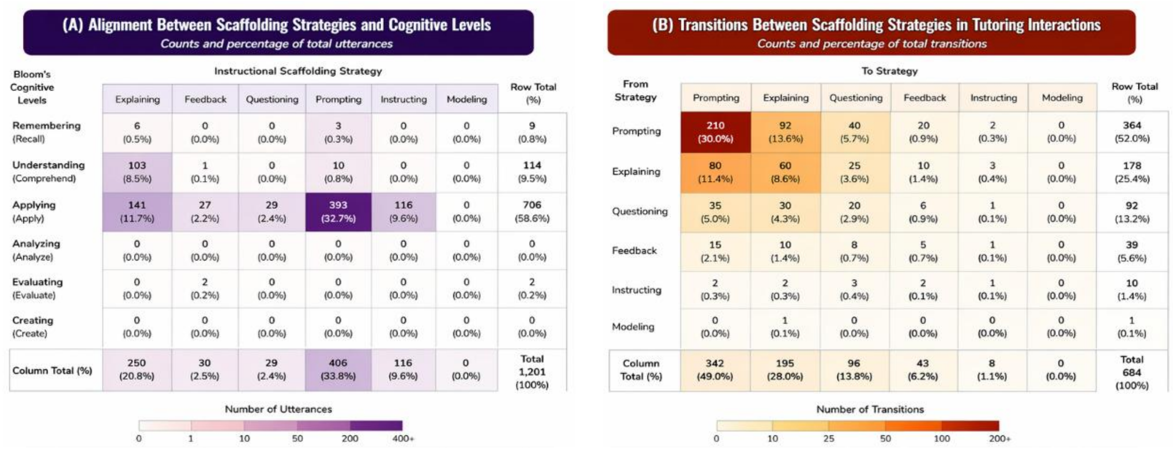
3.1.2. Cognitive levels of instructional support

The cognitive levels of instructional support are also shown in Figure 1. Most utterances were at the understanding and applying levels. These focused on concept clarification, syntax explanation, and rule-based problem solving. Only a few utterances reached the analyzing and evaluating levels. These involved error detection, comparison of approaches, or justification of solutions. No utterances were classified at the creating level. This shows a lack of prompts that encourage original solutions or alternative strategies. Overall, the cognitive demand was limited to lower and mid-level processes.

3.1.3. Analysis of scaffolding alignment and instructional transitions

Figure 2 presents an integrated view of instructional scaffolding by combining the alignment between scaffolding strategies and Bloom's cognitive levels, as in Figure 2(a), and the transition patterns between strategies during tutoring, as in Figure 2(b). Together, these show both the cognitive depth of instruction and how scaffolding unfolds over time. The alignment heatmap shows that scaffolding is concentrated at lower to mid-level cognitive demands. Prompting is most strongly linked to the applying level (393 utterances, about 33%), indicating a focus on procedural tasks. Explaining aligns with the understanding level and supports concept clarification. Questioning appears across understanding and applying, with few instances at higher levels. The remembering level is rare and mainly associated with explaining and questioning. Higher-order levels such as analyzing and evaluating are limited across all strategies, and creating is absent. This pattern shows that scaffolding focuses on procedural guidance rather than deeper cognitive engagement.

The transition heatmap shows how scaffolding develops during interaction. The most common pattern is prompting → prompting, which reflects repeated directive support. This is followed by transitions between prompting and explaining, showing cycles of instruction and clarification. Transitions involving questioning, feedback, and instructing are less frequent. Modeling is rare and does not appear in sustained sequences. There is little evidence of progression to more complex strategies or scaffold fading. The figure shows instructional stability rather than progression. Tutoring interactions remain at lower to mid-level scaffolding and do not advance toward higher-order thinking or adaptive shifts in instruction.



(a) (b)

Figure 2. Integrated alignment of (a) instructional scaffolding strategies with Bloom's cognitive levels and (b) transition dynamics in tutoring interactions

3.1.4. Qualitative illustrations of scaffolding patterns

Representative dialogue excerpts are shown in Table 2. These examples illustrate common scaffolding patterns. Instructional support mainly guides procedural task completion and concept clarification. Fewer instances promote higher-order reasoning or evaluation. Qualitative analysis shows that learners were effectively guided through syntax corrections and logical steps, which supported task completion. However, even when learners seemed ready for deeper reasoning, the system often returned to procedural prompts. Opportunities for abstraction, comparison, and evaluation were limited. The strategy and cognitive level patterns in Table 2 align with the results in Figure 2. Prompting at the applying level and explaining at the understanding level were most common. This consistency between quantitative and qualitative findings strengthens the overall results.

Table 2. Representative dialogue excerpts illustrating instructional scaffolding patterns

Scaffolding strategy	Bloom's cognitive level	Representative tutor excerpt	Instructional purpose
Prompting	Applying	"Try checking the condition inside the loop again. What happens when the value changes?"	Guides learners to apply programming rules to complete a task
Explaining	Understanding	"An *int* is used for whole numbers, while a *char* stores a single character."	Clarifies conceptual understanding of programming constructs
Questioning	Understanding	"What do you think this line of code is supposed to do?"	Encourages learners to articulate conceptual understanding
Prompting	Understanding	"Look at the variable declaration first before moving to the condition."	Directs attention to relevant components of the task
Questioning	Analyzing	"How is this loop different from the one you used earlier?"	Prompts comparison and identification of relationships
Feedback	Applying	"That condition works correctly now. You can move to the next step."	Confirms correct application and reinforces learning
Explaining	Remembering	"Remember that if statements always need a condition inside the parentheses."	Reinforces recall of syntax rules
Prompting	Applying	"Try running the code again after changing that value."	Encourages iterative testing and task completion
Questioning	Evaluating	"Why do you think this solution is more efficient?"	Promotes justification of programming choices
Prompting	Applying	"Add the break statement once the score reaches the limit."	Guides procedural execution

3.2. Discussion

The findings of this study are supported by empirical analysis of 1,255 tutor utterances collected from 36 authentic tutoring sessions. This allowed a systematic examination of scaffolding strategies and their cognitive levels in dialogue-based tutoring. The findings show how these systems support programming learning and reveal both strengths and limitations of AI-supported scaffolding.

3.2.1. Interpretation of instructional patterns

The frequent use of prompting and explaining shows that the system supports procedural tasks and basic concept understanding. Prior studies show that novice programmers benefit from clear guidance and explicit explanations, especially when AI systems provide structured scaffolding and feedback during problem solving [7], [8], [21]. From a sociocultural perspective, such instructional support aligns with scaffolding within the learner's ZPD, where guidance enables task completion that learners may not yet achieve independently [8].

However, the limited use of modeling and reflective feedback suggests that learners are rarely exposed to demonstrations of expert reasoning or opportunities for metacognitive reflection. Research on cognitive apprenticeship emphasizes that modeling plays a critical role in making expert thinking visible, thereby supporting conceptual transfer and strategic problem solving [22]. Feedback studies also show that effective feedback promotes reflection, self-monitoring, and autonomy, not just error correction [23]. Its impact depends on task complexity, which highlights the need for well-designed feedback to support deeper learning [24].

3.2.2. Cognitive demand and depth of learning

The concentration of instructional support at the understanding and applying levels of Bloom's revised taxonomy shows a focus on lower to mid-level cognitive processes. This pattern is consistent with prior studies in programming education, where instruction often emphasizes rule application and syntax mastery [5], [25]. While this is suitable for novice learners, the limited support for higher-order thinking raises concerns. Opportunities for analysis, evaluation, and creative problem solving appear restricted. Higher-order engagement is important for developing transferable skills and long-term learning. It supports deeper understanding, sustained problem solving, and persistence in real tasks [26], [27]. The absence of creating-level scaffolding suggests that dialogue-based tutoring systems need more deliberate design to support cognitive progression beyond procedural skills. Without such support, AI tutoring may reinforce surface-level learning rather than deeper understanding. These findings do not mean the system is ineffective. Instead, they show that its pedagogical scope is limited. The system is effective for procedural mastery but less suited for promoting higher-order cognitive engagement.

3.2.3. Scaffolding progression and instructional design

Effective scaffolding involves not only providing instructional support but also gradually shifting responsibility to the learner as competence increases [24]. The observed repetition of prompting and explaining without consistent progression toward open-ended questioning or scaffold fading suggests that instructional support may plateau rather than adapt to learner readiness. Similar limitations have been reported in studies of AI-driven scaffolding systems, where instructional dialogue may remain directive unless adaptive scaffolding mechanisms are intentionally embedded in the system design [28]–[30]. These findings underscore the importance of embedding pedagogical intelligence into dialogue-based tutoring systems to support sustained cognitive development.

3.2.4. Implications for AI tutor design

The findings have important implications for the design of AI-supported tutoring systems. Current systems support procedural learning but need stronger pedagogical features to promote deeper thinking. AI tutors should include adaptive scaffolding that increases in cognitive complexity as learners improve. They should also use higher-order questions that prompt learners to analyze, evaluate, and justify their answers. Systems should apply scaffold fading to gradually reduce support and build learner independence. Metacognitive prompts can help learners reflect and monitor their thinking, while modeling expert reasoning can make problem-solving strategies clearer. Together, these features can move AI tutoring beyond procedural support and promote cognitive development, higher-order thinking, and learner autonomy.

3.2.5. Implications for educational practice

From an educational evaluation perspective, the findings suggest that AI-supported tutoring systems effectively reinforce foundational and procedural skills. However, consistent with prior research, reliance on these systems without pedagogical oversight may limit higher-order thinking and self-regulated learning [9], [11], [22], [31]. Thus, dialogue-based tutors should be used as complementary tools rather than replacements for instruction targeting reasoning, reflection, and creativity. Educators can integrate AI tutoring for practice and reinforcement, while reserving class time for analysis, evaluation, and creative problem-solving. This blended approach supports both automated guidance and human-facilitated cognitive development. Additionally, teacher training should include guidance on interpreting AI-generated dialogue to identify when further scaffolding is needed.

3.2.6. Limitations and future research directions

This study has several limitations. It analyzed interaction logs from a single tutoring system in a specific programming context, limiting generalizability. It also focused on tutor utterances without examining learner responses or outcomes. Future research should explore how adaptive dialogue adjusts scaffolding based on learner responses and readiness, compare AI and human scaffolding approaches, and conduct longitudinal studies on cognitive development. Applying this framework across domains and AI platforms would further test generalizability.

4. CONCLUSION

This study examined instructional scaffolding in a dialogue-based AI tutoring system for novice programming learners. It aimed to determine whether AI tutoring promotes cognitive engagement beyond procedural tasks. A dual-layer coding framework based on scaffolding theory and Bloom's revised taxonomy was used to analyze 1,255 tutor utterances from 36 sessions. The findings show that scaffolding is concentrated at the understanding and applying levels. Prompting and explaining were the main strategies. These support concept clarification and rule-based problem solving. However, higher-order processes such as analyzing, evaluating, and creating were rarely observed. The analysis also shows that interactions often remain in directive guidance and do not progress toward scaffold fading or learner autonomy.

These results suggest that AI tutoring systems support foundational knowledge and procedural skills. However, their impact is limited without design features that promote deeper cognitive engagement. Adaptive scaffolding, reflective questioning, and higher-order prompts may help support more advanced thinking. AI tutors should not be treated as complete instructional solutions. They should be used as part of a broader pedagogical approach. Combining AI support with teacher guidance can better promote meaningful learning and cognitive development.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Julieto Perez	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
January Naga	✓	✓		✓	✓	✓		✓	✓	✓	✓		✓	
Salma Naga-Marohombsar				✓	✓		✓			✓	✓	✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Informed consent was obtained from all participants. Participation was voluntary, and all data were anonymized with no personally identifiable information retained.

ETHICAL APPROVAL

This study complied with relevant national regulations and institutional policies. Data handling and analysis adhered to the Philippine Data Privacy Act of 2012 (Republic Act No. 10173) and were approved by the University Ethics Review Board (UERB-2025-00536).

DATA AVAILABILITY

The data supporting this study are available from the corresponding author, [JN], upon request but are not publicly available due to privacy restrictions.

REFERENCES

[1] S. Wang, F. Wang, Z. Zhu, J. Wang, T. Tran, and Z. Du, "Artificial intelligence in education: a systematic literature review," *Expert Systems with Applications*, vol. 252, p. 124167, Oct. 2024, doi: 10.1016/j.eswa.2024.124167.

[2] M. Park, S. Kim, S. Lee, S. Kwon, and K. Kim, "Empowering personalized learning through a conversation-based tutoring system with student modeling," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, May 2024, pp. 1–10, doi: 10.1145/3613905.3651122.

[3] R. Schmucker, M. Xia, A. Azaria, and T. Mitchell, "Ruffle&Riley: from lesson text to conversational tutoring," in *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, Jul. 2024, pp. 547–549, doi: 10.1145/3657604.3664719.

[4] Y. Li *et al.*, "An innovative Socratic method-based artificial intelligence platform for healthcare education: a quasi-experimental study," *Nurse Education in Practice*, vol. 92, p. 104770, Mar. 2026, doi: 10.1016/j.nepr.2026.104770.

[5] R. Weeda, S. Smetsers, and E. Barendsen, "Unraveling novices' code composition difficulties," *Computer Science Education*, vol. 34, no. 3, pp. 414–441, Jul. 2024, doi: 10.1080/08993408.2023.2169067.

[6] M. Bastian and A. Mühling, "Misconceptions in programming: intuitive reasoning and tracing task performance across experience levels," in *Proceedings of the 2025 ACM Conference on International Computing Education Research*, Aug. 2025, pp. 141–154, doi: 10.1145/3702652.3744209.

[7] B. Allagui, "A scaffolding intervention to improve self-efficacy in source-based argumentative writing," *Frontiers in Psychology*, vol. 15, p. 1454104, Nov. 2024, doi: 10.3389/fpsyg.2024.1454104.

[8] L. S. Vygotsky, *Mind in society*. Cambridge, MA: Harvard University Press, 1980, doi: 10.2307/j.ctvjf9vz4.

[9] B. R. Belland, C. Kim, and M. J. Hannafin, "A framework for designing scaffolds that improve motivation and cognition," *Educational Psychologist*, vol. 48, no. 4, pp. 243–270, Oct. 2013, doi: 10.1080/00461520.2013.838920.

[10] L. W. Anderson, D. R. Krathwohl, and B. S. Bloom, *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. New York, NY: Longman, 2001.

[11] E. Kasneci *et al.*, "ChatGPT for good? On opportunities and challenges of large language models for education," *Learning and Individual Differences*, vol. 103, p. 102274, Apr. 2023, doi: 10.1016/j.lindif.2023.102274.

[12] L. Labadze, M. Grigolia, and L. Machaidze, "Role of AI chatbots in education: systematic literature review," *International Journal of Educational Technology in Higher Education*, vol. 20, no. 1, p. 56, Oct. 2023, doi: 10.1186/s41239-023-00426-1.

[13] A. Al-Abri, "Exploring ChatGPT as a virtual tutor: a multi-dimensional analysis of large language models in academic support," *Education and Information Technologies*, vol. 30, no. 12, pp. 17447–17482, Aug. 2025, doi: 10.1007/s10639-025-13484-x.

[14] S. Beltozar-Clemente, E. Díaz-Vega, J. Zapata-Paulini, and R. E. Tejeda-Navarrete, "We can rely on ChatGPT as an educational tutor: a cross-sectional study of its performance, accuracy, and limitations in university admission tests," *International Journal of Engineering Pedagogy (iJEP)*, vol. 14, no. 1, pp. 50–60, Jan. 2024, doi: 10.3991/ijep.v14i1.46787.

[15] A. Létourneau, M. D. Martineau, P. Charland, J. A. Karran, J. Boasen, and P. M. Léger, "A systematic review of AI-driven intelligent tutoring systems (ITS) in K-12 education," *npj Science of Learning*, vol. 10, no. 1, p. 29, May 2025, doi: 10.1038/s41539-025-00320-7.

[16] A. Nugraha, T. Inoue, T. A. Salim, and M. H. Inamullah, "A dialogue-like video created from a monologue lecture video provides better learning experience," *International Journal of Distance Education Technologies*, vol. 21, no. 1, pp. 1–21, Nov. 2023, doi: 10.4018/IJDET.334012.

[17] M. T. H. Chi, "Active-constructive-interactive: a conceptual framework for differentiating learning activities," *Topics in Cognitive Science*, vol. 1, no. 1, pp. 73–105, Jan. 2009, doi: 10.1111/j.1756-8765.2008.01005.x.

[18] D. Wood, J. S. Bruner, and G. Ross, "The role of tutoring in problem solving," *Journal of Child Psychology and Psychiatry*, vol. 17, no. 2, pp. 89–100, Apr. 1976, doi: 10.1111/j.1469-7610.1976.tb00381.x.

[19] R. Kasepalu, L. P. Prieto, T. Ley, and P. Chejara, "Teacher artificial intelligence-supported pedagogical actions in collaborative learning coregulation: a wizard-of-oz study," *Frontiers in Education*, vol. 7, p. 736194, Feb. 2022, doi: 10.3389/feduc.2022.736194.

[20] D. R. Krathwohl, "A revision of Bloom's taxonomy: an overview," *Theory Into Practice*, vol. 41, no. 4, pp. 212–218, Nov. 2002, doi: 10.1207/s15430421tip4104_2.

[21] A. S. Omar, M. Mgala, and F. Mwakondo, "AI-driven visual scaffolding in education: a comprehensive literature review," *International Journal of Research and Scientific Innovation*, vol. 12, no. 3, pp. 740–750, 2025, doi: 10.51244/IJRSI.2025.12030055.

[22] K. K. Maurya, K. A. Srivatsa, K. Petukhova, and E. Kochmar, "Unifying AI tutor evaluation: an evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language* Technologies, 2025, pp. 1234–1251, doi: 10.18653/v1/2025.naacl-long.57.

[23] J. Campbell and R. E. Mayer, "Questioning as an instructional method: does it affect learning from lectures?" *Applied Cognitive Psychology*, vol. 23, no. 6, pp. 747–759, Sep. 2009, doi: 10.1002/acp.1513.

[24] J. Sweller, P. Ayres, and S. Kalyuga, *Cognitive load theory*. New York, NY: Springer New York, 2011, doi: 10.1007/978-1-4419-8126-4.

[25] S. Grover and R. Pea, "Computational thinking in K–12," *Educational Researcher*, vol. 42, no. 1, pp. 38–43, Jan. 2013, doi: 10.3102/0013189X12463051.

[26] A. Collins, J. S. Brown, and S. E. Newman, "Cognitive Apprenticeship: Teaching the Craft of Reading, Writing, and Mathematics," Technical Report No. 403, Center for the Study of Reading, University of Illinois, 1987. [Online]. Available: https://eric.ed.gov/?id=ED284181

[27] E. H. S. Y. Elim, "Promoting cognitive skills in AI-supported learning environments: the integration of Bloom's taxonomy," *Education 3-13*, vol. 54, no. 3, pp. 612–622, Apr. 2026, doi: 10.1080/03004279.2024.2332469.

[28] V. Aleven *et al.*, "Example-tracing tutors: intelligent tutor development for non-programmers," *International Journal of Artificial Intelligence in Education*, vol. 26, no. 1, pp. 224–269, Mar. 2016, doi: 10.1007/s40593-015-0088-2.
[29] K. Holstein, B. M. McLaren, and V. Aleven, "Co-designing a real-time classroom orchestration tool to support teacher–AI complementarity," *Journal of Learning Analytics*, vol. 6, no. 2, pp. 27–52, Jul. 2019, doi: 10.18608/jla.2019.62.3.
[30] J. Stamper, R. Xiao, and X. Hou, "Enhancing LLM-based feedback: insights from intelligent tutoring systems and the learning sciences," in *International Conference on Artificial Intelligence in Education*, 2024, pp. 32–43, doi: 10.1007/978-3-031-64315-6_3.
[31] C. Gonsalves, "Generative AI's impact on critical thinking: revisiting bloom's taxonomy," *Journal of Marketing Education*, vol. 48, no. 1, pp. 4–19, Apr. 2026, doi: 10.1177/02734753241305980.

BIOGRAPHIES OF AUTHORS

Julieto Perez is a faculty member in the Department of Computer Science at MSU-Iligan Institute of Technology. He is a Ph.D. candidate in Computer Science from De La Salle University-Manila. His research interest revolves around artificial intelligence in education, with a specific focus on the innovative use of natural language processing in building educational applications. He has authored peer-reviewed Scopus-indexed papers related to AI in education. He can be contacted at email: julieto.perez@g.msuiit.edu.ph.

January Naga is an associate professor at MSU-Iligan Institute of Technology and a Doctor in Information Technology (DIT) candidate at De La Salle University-Manila. Her research interests include digital sociology, digital health, information and communication technologies for development (ICT4D), and information systems. She can be contacted at email: january.naga@g.msuiit.edu.ph.

Salma Naga-Marohombsar is a mathematician and peace education scholar at Mindanao State University (MSU), Marawi City, Philippines, where she serves as Chairperson of the Department of Mathematics. She holds a doctoral degree in mathematics and has extensive teaching experience in mathematics, statistics, and interdisciplinary peace education. Her research focuses on algebra, graph theory, and mathematical modeling, particularly applying quantitative and systems-based approaches to peace, conflict, and post-conflict recovery in Mindanao. She can be contacted at email: salma.marohombsar@msumain.edu.ph.