Beyond the Perfect Assistant: Provoking Learning with Flawed AI Partners

Vaisiya Balakrishnan*
College of Computing and Data Science
Nanyang Technological University
Singapore, Singapore, Singapore
vaisiya001@e.ntu.edu.sg

Ler Koon Kway*
College of Computing and Data Science
Nanyang Technological University
Singapore, Singapore, Singapore
lkway001@e.ntu.edu.sg

Advika Harvande Sandeep
College of Computing and Data Science
Nanyang Technological University
Singapore, Singapore, Singapore
advika001@e.ntu.edu.sg

Eva Faith Jie Ting Wong
College of Computing and Data Science
Nanyang Technological University
Singapore, Singapore, Singapore
evaf0001@e.ntu.edu.sg

Ong Chin Ann
College of Computing and Data Science
Nanyang Technological University
Singapore, Singapore, Singapore
chinann.ong@ntu.edu.sg

Owen Noel Newton Fernando
College of Computing and Data Science
Nanyang Technological University
Singapore, Singapore, Singapore
ofernando@ntu.edu.sg

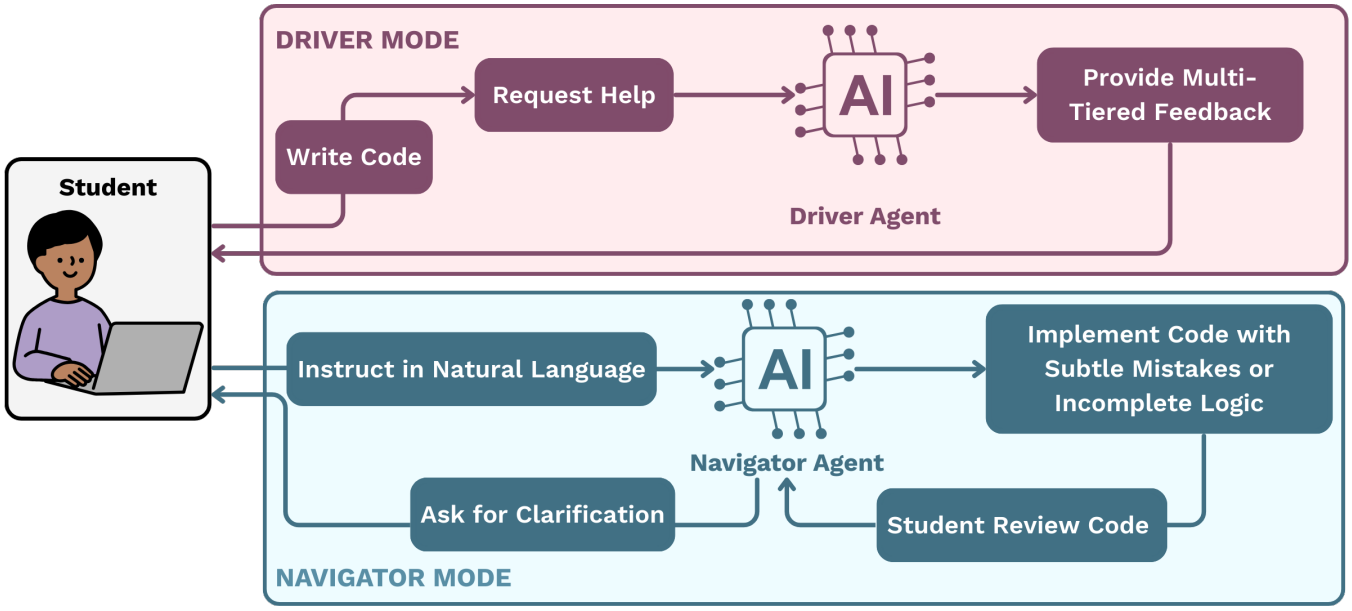

Figure 1: The core interaction loop within AlgoGPT. The architecture operationalizes pedagogical friction through two role-reversing interaction loops: (Top) Driver Mode, where the student writes code and receives non-directive, multi-tiered Socratic feedback from the Driver Agent; and (Bottom) Navigator Mode, where the student provides natural language instructions and the Navigator Agent intentionally generates code with subtle mistakes or incomplete logic to provoke student review and critical reflection.

Abstract

Generative AI systems in education are commonly designed as flawless, omniscient assistants optimized for correctness, speed, and convenience. While effective for information retrieval, this paradigm risks encouraging cognitive offloading and passive learning, particularly in abstract domains such as computer science education. This work-in-progress challenges that assumption by presenting AlgoGPT, a model of AI-mediated pair programming in which the AI is intentionally constrained. Rather than acting as an oracle, the AI functions as a pedagogical partner that withholds direct solutions and, in one mode, introduces deliberate imperfections. Through two role-reversing interaction modes alongside a chat-based baseline, the system provokes learners to articulate reasoning,

*Equal contribution.

DIS Companion '26, Singapore, Singapore

ACM ISBN 979-8-4007-2632-3/26/06
https://doi.org/10.1145/3802974.3809428

critique AI-generated code, and engage in metacognitive reflection. Findings from a large-scale deployment with 907 students and an SRL survey of 155 participants suggest the design is associated with sustained engagement and elevated metacognitive activity, while also surfacing affective risks that call for adaptive scaffolding.

CCS Concepts

• **Human-centered computing** → **Interactive systems and tools**; • **Applied computing** → *Education*.

Keywords

AI Pair Programming; Cognitive Load Theory; Self-regulated Learning

ACM Reference Format:
Vaisiya Balakrishnan, Ler Koon Kway, Advika Harvande Sandeep, Eva Faith Jie Ting Wong, Ong Chin Ann, and Owen Noel Newton Fernando. 2026. Beyond the Perfect Assistant: Provoking Learning with Flawed AI Partners. In *Designing Interactive Systems Conference (DIS Companion '26), June 13–17, 2026, Singapore, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3802974.3809428

1 Introduction

The rapid adoption of Large Language Models (LLMs) in education has been driven by a narrative of optimization: faster feedback, fewer errors, and frictionless access to solutions [11, 17]. In programming education, this has manifested as AI assistants that generate correct code on demand, explain algorithms fluently, and debug with near-instant accuracy [1, 16].

However, this pursuit of "perfect assistance" creates a pedagogical tension [7, 10]. Learning theories such as Cognitive Load Theory (CLT) and Self-Regulated Learning (SRL) emphasize that durable understanding requires effortful engagement and active knowledge construction [18, 19]. When AI systems provide immediate, correct solutions, they risk short-circuiting this process, shifting learners toward solution retrieval rather than reasoning [11, 14].

This tension is particularly pronounced in abstract domains such as computer science education [8, 13]. In response, we present AlgoGPT, an intelligent Data Structures and Algorithms (DSA) tutor built around an alternative design principle: that productive friction, deliberately embedded in the interaction architecture, may better support learning than frictionless assistance. AlgoGPT's two pair programming modes constrain the AI in different ways—one through Socratic withholding, the other through intentional code imperfection—while a chat-based Normal Mode serves as a within-system baseline.

2 Related Work

AI-mediated pair programming. LLM-based pair programming systems generate code, explanations, and fixes in real time, improving productivity and reducing perceived effort [3, 9]. Prior work introduces structured prompting and staged assistance to encourage user involvement [12, 15]. However, these systems are typically evaluated in short-term settings and are not explicitly grounded in learning theories like CLT or SRL, leaving open questions about their effects on sustained learning.

Cognitive friction in AI-mediated interaction. A growing body of work has examined how interaction design can counteract the tendency toward uncritical acceptance of AI outputs. Techniques such as requiring users to commit to judgments before seeing AI outputs or delaying responses (*cognitive forcing functions*) have been shown to reduce over-reliance, though at the cost of increased effort and lower user preference [2]. Danry et al. [4] show that framing AI feedback as questions rather than statements encourages independent reasoning and improves error detection.

Prior work treats friction as an occasional intervention (e.g., a forcing prompt or a reframed output). AlgoGPT extends this by embedding friction as a *structural* property of the interaction: Driver Mode enforces Socratic withholding across every help request; Navigator Mode makes imperfection the default output of every code generation. This also grounds the approach explicitly in CLT and SRL frameworks.

3 AlgoGPT: AI as a Pedagogical Partner

Our central hypothesis is that the most effective AI pedagogical agent may be a deliberately constrained one—one whose behavior is optimized not for answer-correctness, but for learner engagement. As shown in Figure 1, AlgoGPT operationalizes this through a dual-mode AI pair programming environment that aims to emulate traditional human-human pair programming.

Driver Mode. Using the interface shown in Figure 2, students write code and the AI intervenes only upon request. Rather than offering direct fixes, the agent provides non-directive, Socratic nudges that draw attention to potential edge cases, control-flow issues, or mismatches between problem requirements and implementation. Friction here takes the form of *strategic withholding*: the agent deliberately refrains from supplying complete solutions, functioning more like a senior human instructor who scaffolds through questions. This design encourages learners to plan, monitor, and reflect on their own problem-solving processes, resisting cognitive offloading while providing timely scaffolding at impasses.

Navigator Mode. In Navigator Mode, the roles are reversed. As illustrated by Figure 3, students articulate their reasoning in natural language while the AI writes code in a locked editor, forcing students to externalize their mental model. The Navigator agent in this mode is intentionally designed to behave as a diligent but imperfect collaborator. It is prompted to introduce subtle logical oversights or incomplete reasoning proportionate to the complexity of the student's instruction. This is distinct from ordinary model error: the prompt instructs the agent to produce *plausibly wrong* rather than *randomly wrong* outputs, targeting errors a student might plausibly make themselves. Students must then Accept or Reject the generated code, operationalizing the "evaluate before accepting" principle of cognitive forcing functions [2]. We acknowledge the limitation that not all Navigator outputs can be verified as intentional flaws versus model misunderstanding.

Normal Mode (Baseline). Normal Mode provides a conversational AI interface available across all lab sessions. Students may ask questions about concepts or problem requirements, but the AI withholds direct code solutions and step-by-step implementations.

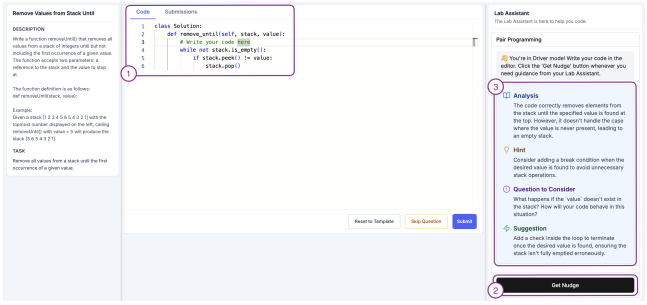

Figure 2: Driver Interface. The workspace demonstrates: (1) the active code editor; (2) the student-initiated "Get Nudge" trigger; and (3) the multi-tiered feedback panel which decomposes support into high-level analysis, conceptual hints, questions, and suggestions to guide the student toward independent resolution without providing direct code fixes.

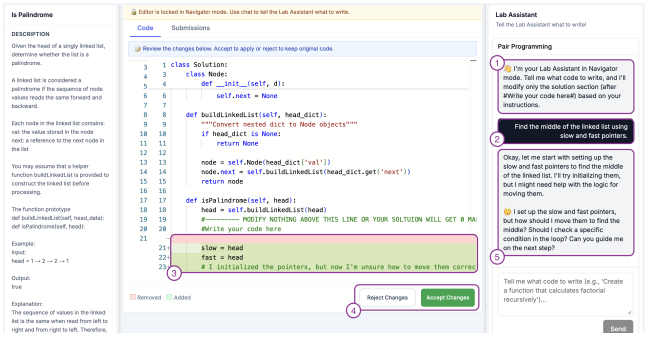

Figure 3: Navigator Interface. The interaction sequence highlights: (1) the agent establishing its role; (2) natural language instructions from the student; (3) the resulting code injection within the locked editor, featuring intentional logical flaws or omissions; (4) the validation UI where students must "Accept" or "Reject" changes; and (5) the agent's follow-up to prompt further logic refinement.

This serves as a within-system baseline, controlling for AI availability while isolating the incremental effects of Socratic withholding and flaw injection.

4 Methodology

We conducted an in-situ deployment of AlgoGPT with 907 undergraduate students in an introductory DSA course at Nanyang Technological University across four lab sessions. Students completed 27 DSA problems, with each problem pre-assigned to one of the three modes. Students were not required to transition between modes within a problem. We recorded and analyzed 14,958 submissions, examining pass rates and attempt frequency per mode per lab. To supplement log data, a 12-item Likert-scale SRL instrument (1 = Strongly Disagree, 5 = Strongly Agree) was administered to 155 students. The 12 items map onto five key dimensions:

- **Metacognitive Planning:** Goal setting before sessions (Q01) and pre-planning implementation steps (Q02).
- **Cognitive Scaffolding:** The system's role in organizing learning in a structured way (Q03).
- **Metacognitive Monitoring:** Checking understanding during tasks (Q04), adapting strategy when stuck (Q05), and tracking individual progress (Q06).
- **Reflective Self-Evaluation:** Post-session reflection (Q07), using AI feedback for evaluation (Q08), and leveraging mistakes for improvement (Q09).
- **Affective Outcomes:** Sustained focus and persistence (Q10), motivation to continue (Q11), and perceived sense of control (Q12).

Two key limitations bound interpretation. First, the Normal Mode baseline does not represent a fully permissive AI. Comparisons between modes therefore reflect *incremental* friction rather than a contrast with unconstrained AI assistance. Second, problem difficulty is not controlled across mode assignments, which complicates attribution of performance differences to interaction design alone.

5 Results

5.1 Engagement and Performance Patterns

Students averaged 3.61 attempts per question, and 74.1% of sessions ran to time expiry with only 6.6% explicitly skipped, indicating sustained willingness to persist under challenging conditions.

As seen in Figure 4, mode-level pass rates revealed a consistent pattern. Normal Mode achieved the highest pass rates in every lab (76.5%–91.5%). Driver Mode was generally lowest (23%–57%), consistent with the demand for independent coding under Socratic withholding. Navigator Mode occupied an intermediate, more variable position (39%–78%), with one exception: in Lab 4 (Binary Search Trees), Navigator Mode underperformed Driver Mode (62% vs. 70%), possibly because the increased abstraction of BST operations made natural-language instruction more difficult. Although problem difficulty is not controlled, the pattern is directionally consistent with the hypothesis that both friction modes impose greater cognitive demand than the baseline.

At the cohort level, 62.2% of students fell into a struggling group (<30% success) and 30.9% into an intermediate group, with only 10 students achieving high performance (>80%). This is consistent with the system's design intention of promoting productive difficulty, but should not be interpreted as evidence of learning gains without pre/post assessments.

5.2 Self-Regulated Learning Survey Results

The overall mean SRL score was 3.39/5.0, above neutral across all five dimensions. One-sample t-tests confirmed all items significantly exceeded the midpoint as shown in Figure 5 ($\alpha = 0.05$, one-tailed, $t_{\text{crit}} = 1.645$, $N = 155$).

Reflective Self-Evaluation was the strongest dimension ($M = 3.46$), driven by Planning Improvement ($t = 5.980$) and Post-session Reflection ($t = 5.136$). Metacognitive Monitoring was similarly strong ($M = 3.41$), with Checking Understanding showing the largest effect ($t = 6.968$). Cognitive Scaffolding ($M = 3.24$) and Affective Outcomes ($M = 3.23$) were weakest; Motivation to Continue ($t = 1.869$) only marginally exceeded the significance threshold.

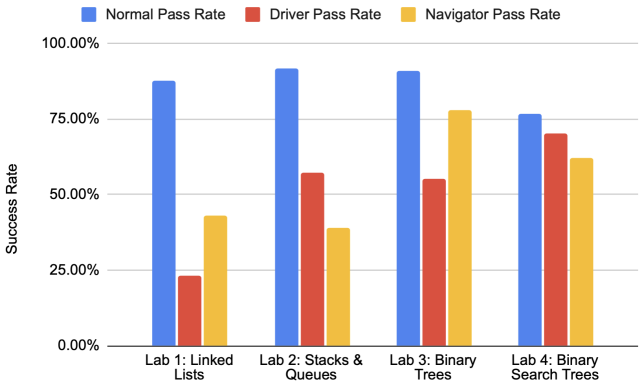

Figure 4: Success rates by interaction mode across labs

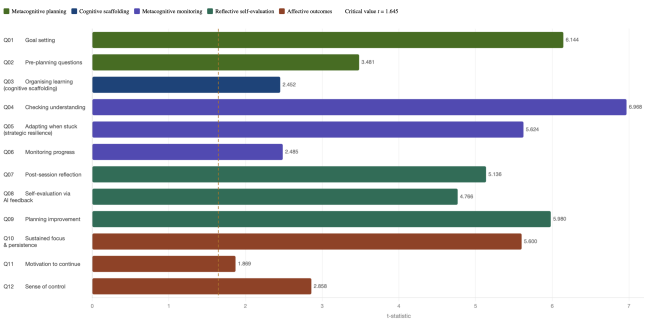

Figure 5: One-sample t-test: Is each survey question significantly above the neutral midpoint?

Student engagement varied substantially: 48.4% fell into a High Agency tier (mean ≥ 3.5), 38.7% into Moderate Adoption, and 12.9% into Low Engagement. Within the High Agency group, 81.3% reported adapting strategy when stuck and 83.9% reported actively reflecting after sessions.

6 Discussion

Engagement despite difficulty. A key concern with introducing friction into learning systems is the risk of frustration or abandonment [5]. High attempt frequency, long session durations, and low skip rates collectively suggest that students were not broadly deterred. However, persistence alone does not confirm productive learning [6]: repeated attempts could equally reflect confusion or trial-and-error behavior. Distinguishing the two will require qualitative analysis of interaction logs and direct outcome measures such as pre/post assessments or transfer tasks. The current results are best understood as characterizing *engagement under friction*, not learning gains.

Metacognitive engagement. The SRL results offer a more direct signal. Reflective Self-Evaluation and Metacognitive Monitoring emerged as the strongest dimensions, aligning with the specific cognitive demands of each mode: Driver Mode's withholding targets self-monitoring, while Navigator Mode's flaw injection targets post-hoc critique. From a CLT standpoint, this pattern is consistent with effort being redirected toward germane load [18]. The affective picture is more cautionary: Q11's borderline result ($t = 1.869$) indicates that sustaining intrinsic motivation under repeated failure is a distinct design challenge not yet resolved. Taken alongside the 62.2% struggling cohort, this raises a genuine risk of learned helplessness and misconception reinforcement if friction is left unmodulated.

Uneven impact across learners. Driver Mode's consistently low pass rates and Navigator Mode's wider variance suggest that both modes impose meaningful cognitive demand beyond the Normal baseline—though uncontrolled problem difficulty limits causal claims. Crucially, the large struggling cohort (62.2%) and the Low Engagement segment (12.9% of survey respondents) indicate that friction is not uniformly beneficial. Without adaptive scaffolding, repeated failure risks reinforcing misconceptions or triggering learned helplessness. The current system applies uniform friction regardless of learner state. Future iterations could explore performance-triggered mode switching and calibrated flaw severity to better match individual needs.

7 Conclusion

This work challenges the assumption that educational AI should prioritize correctness and convenience. We instead present a model in which constraint and imperfection are used to provoke reasoning, reflection, and self-regulation. Findings from a large-scale deployment suggest that learners remain engaged and report increased metacognitive activity under this friction-based design, even in the presence of frequent failure. However, the current evidence characterizes engagement under friction rather than learning gains, and the affective data caution that a non-trivial proportion of students may experience the system as motivationally discouraging. Future work will focus on addressing remaining limitations through controlled comparisons with fully assistive AI systems, direct pre/post learning measures, and qualitative analysis of student-AI exchanges. Further investigation is also needed to refine the design of intentional flaws and to develop adaptive friction mechanisms that better support diverse learners.

Acknowledgments

This study was funded by Nanyang Technological University (NTU), Singapore (Edex Grant #025734-00001) and administered by NTU. Any opinions, conclusions or implications in this material are those of the author(s) and do not necessarily reflect the views of NTU. The study was approved by the NTU IRB (number: IRB-2023-735).

References

[1] Patrick Bassner, Ben Lenk-Ostendorf, Ramona Beinstingel, Tobias Wasner, and Stephan Krusche. 2026. Less stress, better scores, same learning: The dissociation of performance and learning in AI-supported programming education. *Computers and Education: Artificial Intelligence* 10 (2026), 100537. doi:10.1016/j.caeai.2025.100537

[2] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. 2021. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW1, Article 188 (April 2021), 21 pages. doi:10.1145/3449287

[3] Xi Chen and Jingsai Liang. 2024. Pair Programming with ChatGPT. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 2* (Portland, OR, USA) *(SIGCSE 2024)*. Association for Computing Machinery, New York, NY, USA, 1600–1601. doi:10.1145/3626253.3635600

[4] Valdemar Danry, Pat Pataranutaporn, Yaoli Mao, and Pattie Maes. 2023. Don't Just Tell Me, Ask Me: AI Systems that Intelligently Frame Explanations as Questions

Improve Human Logical Discernment Accuracy over Causal AI explanations. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 352, 13 pages. doi:10.1145/3544548.3580672

[5] Anique de Bruin. 2023. Dealing with Desirable Difficulties: Supporting Students to Accept, Reduce, or Silence Effort. *Medical Science Educator* 33 (10 2023). doi:10.1007/s40670-023-01911-y

[6] Anique de Bruin, Felicitas Biwer, Luotong Hui, Erdem Onan, Louise David, and Wisnu Wiradhany. 2023. Worth the Effort: the Start and Stick to Desirable Difficulties (S2D2) Framework. *Educational Psychology Review* 35 (03 2023). doi:10.1007/s10648-023-09766-w

[7] Iris Delikoura, Yi. R Fung, and Pan Hui. 2025. From Superficial Outputs to Superficial Learning: Risks of Large Language Models in Education. arXiv:2509.21972 [cs.CY] https://arxiv.org/abs/2509.21972

[8] Said Elnaffar, Farzad Rashidi, and Abedallah Zaid Abualkishik. 2026. Teaching with AI: A Systematic Review of Chatbots, Generative Tools, and Tutoring Systems in Programming Education. *International Journal of Learning, Teaching and Educational Research* 25, 1 (Jan. 2026), 1–28. doi:10.26803/ijlter.25.1.1

[9] Guangrui Fan, Dandan Liu, Rui Zhang, and Lihu Pan. 2025. The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair programming and individual approaches. *International Journal of STEM Education* 12 (03 2025). doi:10.1186/s40594-025-00537-3

[10] Qian Fu, Yaning Zhao, Zixi Jia, and Yafeng Zheng. 2025. Large Language Models (LLMs) in Programming Learning: The Current Research State and Agenda. *IEEE Transactions on Learning Technologies* PP (01 2025), 1–22. doi:10.1109/TLT.2025.3622043

[11] Wenhan Lyu, Yimeng Wang, Tingting (Rachel) Chung, Yifan Sun, and Yixuan Zhang. 2024. Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Study. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale (L@S '24)*. ACM, 63–74. doi:10.1145/3657604.3662036

[12] Shuai Ma, Junling Wang, Yuanhao Zhang, Xiaojuan Ma, and April Yi Wang. 2025. DBox: Scaffolding Algorithmic Programming Learning through Learner-LLM Co-Decomposition. arXiv:2502.19133 [cs.HC] https://arxiv.org/abs/2502.19133

[13] Adam Mtaho and Leonard Mselle. 2024. Difficulties in Learning the Data Structures Course: Literature Review. *The Journal of Informatics* 4, 1 (May 2024). doi:10.59645/tji.v4i1.136

[14] Griffin Pitts, Anurata Prabha Hridi, and Arun-Balajiee Lekshmi-Narayanan. 2025. A Survey of LLM-Based Applications in Programming Education: Balancing Automation and Human Oversight. arXiv:2510.03719 [cs.CY] https://arxiv.org/abs/2510.03719

[15] Peter Robe and Sandeep Kuttal. 2022. Designing PairBuddy—A Conversational Agent for Pair Programming. *ACM Transactions on Computer-Human Interaction* 29 (08 2022), 1–44. doi:10.1145/3498326

[16] Sergio Rojas-Galeano, Julian Tejada, and Fernando Marmolejo-Ramos. 2025. Between Tool and Trouble: Student Attitudes Toward AI in Programming Education. arXiv:2508.05999 [cs.ET] https://arxiv.org/abs/2508.05999

[17] Yuhong Shi, Kun Yu, Yifei Dong, and Fang Chen. 2026. Large language models in education: a systematic review of empirical applications, benefits, and challenges. *Computers and Education: Artificial Intelligence* 10 (2026), 100529. doi:10.1016/j.caeai.2025.100529

[18] John Sweller. 2011. Cognitive Load Theory. In *Psychology of Learning and Motivation*, Jose P. Mestre and Brian H. Ross (Eds.). Psychology of Learning and Motivation, Vol. 55. Academic Press, 37–76. doi:10.1016/B978-0-12-387691-1.00002-8

[19] Barry Zimmerman. 2002. Becoming a Self-Regulated Learner: An Overview. *Theory Into Practice* 41 (06 2002), 64–70. doi:10.1207/s15430421tip4102_2

Appendix

SRL Survey Instrument

Table 1: SRL survey instrument: item-to-dimension mapping (N = 155; 1 = Strongly Disagree, 5 = Strongly Agree).

Item	Dimension	Question text	*M*	*SD*	*t*
Q01	Metacognitive Planning	I set clear goals for what I wanted to learn before using AlgoGPT.	3.52	1.06	6.14
Q02	Metacognitive Planning	I thought about the steps I would take before starting a problem.	3.27	0.97	3.48
Q03	Cognitive Scaffolding	AlgoGPT helped me organise my learning in a structured way.	3.24	1.21	2.45
Q04	Metacognitive Monitoring	I checked whether I actually understood what I was doing while solving problems.	3.58	1.04	6.97
Q05	Metacognitive Monitoring	When I was stuck, I tried different strategies before giving up or asking for help.	3.48	1.07	5.62
Q06	Metacognitive Monitoring	I tracked my own progress while working through problems.	3.23	1.13	2.49
Q07	Reflective Self-Eval.	After a session, I thought about what went well and what I could improve.	3.44	1.06	5.14
Q08	Reflective Self-Eval.	The feedback from AlgoGPT helped me evaluate my own understanding.	3.45	1.16	4.77
Q09	Reflective Self-Eval.	I used what I learned from mistakes to plan how to improve.	3.50	1.03	5.98
Q10	Affective Outcomes	I stayed focused and persistent even when problems were difficult.	3.50	1.12	5.60
Q11	Affective Outcomes	AlgoGPT kept me motivated to keep learning.	3.18	1.20	1.87
Q12	Affective Outcomes	I felt in control of my learning process when using AlgoGPT.	3.28	1.24	2.86

One-tailed one-sample t-test vs. midpoint 3.0; $t_{\text{crit}} = 1.645$, $\alpha = .05$. All items significant ($p < .05$).