

AI in Programming Education: A Narrative Review and Framework for Hybrid Instruction

Allen Spector, DSc

Independent Scholar and Cybersecurity Practitioner

Manuscript prepared for the International Journal of Culture and Education (IJCAE)

Abstract: Artificial intelligence (AI) has renewed debate about the relationship between personalized assistance and programming education led by instructors. Comparisons are complicated, however, by the treatment of adaptive learning, generative tutoring, student satisfaction, and demonstrated learning as interchangeable constructs. The article presents a narrative review of selected programming and contextual education literature published through 2025, supported by targeted verification of bibliographic details and individual claims. The synthesis examines student engagement, academic performance, educator perceptions, scalability, and implementation challenges while distinguishing evidence from programming contexts from contextual and conceptual contributions. Three recurring interpretive problems emerge: equating technology use with cognitive engagement, inferring independent competence from performance supported by AI, and presenting hybrid instruction as effective without specifying the coordination of instructional responsibility. Drawing on dynamic personalization and distributed scaffolding, the article proposes a framework for hybrid instruction organized around evidence of understanding. Movement between automated support, human dialogue, and independent work depends on observable learner performance. The framework connects diagnosis, adaptive practice, explanation, collaborative review, unaided transfer, and instructional adjustment. Four research propositions and a comparative evaluation agenda address learning, teacher agency, and equitable participation. The contribution is an analytical and design framework rather than a pooled effectiveness estimate or a validated intervention. Educational value should be assessed through independent learning and accountable instructional coordination, not automation alone.

Index Terms: Adaptive learning, artificial intelligence in education, educational equity, hybrid instruction, programming education, scaffolding.

I. INTRODUCTION

Programming education requires more than the production of executable code. Learners must interpret a problem, construct a solution, diagnose errors, explain decisions, and apply concepts when the task changes. Artificial intelligence (AI) tools capable of generating explanations and code complicate the relationship between an observable product and underlying understanding. Becker et al. [1] frame AI code generation as an educational opportunity accompanied by substantial assessment and instructional challenges. Cambaz and Zhang [2] similarly identify assistive uses and risks in a systematic review of code generation models in programming education.

The resulting debate is often framed as a comparison between adaptive learning driven by AI and traditional teaching. Yet the categories contain materially different practices. Adaptive programming activities described by Troussas et al. [3] are not equivalent to conversational generative tutors. Likewise, the approach to improving the CS50 AI tutor through human feedback described by Liu et al. [4] concerns the pedagogical behavior of a particular system, not the comparative effectiveness of all instruction supported by AI. A useful analysis must therefore specify technological functions, learner practices, continuing teacher responsibilities, and measured outcomes.

Traditional instruction is also an imprecise comparator. Instruction may involve lectures, live coding, laboratory work, peer programming, individualized discussion, or combinations of instructional activities. Srivatanakul's account of an introductory programming course [5] addresses blended delivery and participant perceptions, rather than an experiment

comparing AI with teachers. Similarly, the review by Almusaed et al. [6] provides secondary synthesis concerning AI and hybrid education rather than original controlled evidence of programming examination gains. The distinctions matter because a persuasive pedagogical rationale alone does not establish a causal learning advantage.

The educational problem is consequently not whether AI or teachers are universally better. The problem concerns the coordination of instructional functions without confusing immediate assistance with durable competence. Institutional context is integral to instructional coordination. Mirata et al. [7] identify organizational, technical, and pedagogical challenges in implementing adaptive learning across two university settings. Yan et al. [8] also emphasize learning opportunities and questions about model reliability, assessment, and informed use associated with generative AI. Technology adoption must therefore be examined alongside teacher agency, social participation, and conditions governing access to assistance.

Three research questions organize the review. RQ1 examines how adaptive learning driven by AI and programming education led by instructors compare in relation to student engagement and academic performance. RQ2 examines the evidence concerning educators' perceptions of effectiveness and implementation. RQ3 examines the technical, organizational, ethical, and resource challenges shaping adoption. Scalability and equitable participation are treated as concerns across all three questions rather than as automatic consequences of providing a digital tool.

The article makes three bounded contributions. First, the review differentiates technologies, instructional arrangements, and outcome measures frequently combined in broad claims about AI effectiveness. Second, the synthesis examines five interrelated pedagogical and implementation themes while making the evidentiary limits of individual studies explicit. Third, the proposed framework for hybrid instruction specifies when automated support should be supplemented, challenged, or withdrawn. The framework and associated research propositions are conceptual contributions developed from the synthesis, not reported findings from a newly conducted intervention.

II. CONCEPTUAL AND PEDAGOGICAL FOUNDATIONS

A. Distinguishing Adaptivity, Generation, and Delivery Mode

For the present review, adaptive learning refers to the adjustment of instructional content, task selection, pacing, or support in relation to information about a learner. Generative assistance refers to the production of responses, explanations, examples, or code. A system can combine adaptive and generative functions, but an apparently personalized answer does not establish the presence of an accurate model of a learner's developing knowledge. Tetzlaff et al. [9] offer a dynamic account of personalization connecting repeated information about learner characteristics with instructional adaptation.

Instruction led by teachers is defined by human responsibility for pedagogical decisions, not by the absence of technology. A teacher can adapt an explanation, reorganize a group, or modify an activity without an AI system. Conversely, a platform can provide extensive automation while offering little meaningful adaptation. The relevant comparison is therefore between specified instructional arrangements, not between an inherently flexible technology and an inherently inflexible teacher.

Hybrid instruction in the present article means the deliberate coordination of activities supported by AI with teacher and peer participation. Hybrid instruction does not simply mean mixing online and classroom delivery. A course can be blended in delivery mode without using AI, and an entirely online course can combine automated feedback with substantial human dialogue. Preserving the distinction prevents evidence about blended course delivery from being interpreted as direct evidence for hybrid instruction involving AI.

B. Dynamic Personalization and Distributed Scaffolding

Dynamic personalization provides a rationale for revisiting a learner's needs rather than assigning a permanent category at the beginning of a course [9]. Distributed scaffolding provides a complementary account of how support can be supplied through tools, teachers, and other learners. Puntambekar [10] distinguishes scaffolding processes from the mere availability of instructional scaffolds and addresses the transfer of responsibility to learners. Together, dynamic personalization and distributed scaffolding direct attention to the relationship

between assistance, observed understanding, and the eventual reduction of support.

The framework developed below applies dynamic personalization and distributed scaffolding as design principles. Assistance should respond to a current instructional need; the learner should remain responsible for explaining and testing a solution; and instructional support should change when evidence of understanding changes. The principles represent proposed design commitments, not verified capabilities of a particular AI product. Continuous availability of an explanation generator is not equivalent to a pedagogically contingent support process.

Social participation should also be treated as a design variable. AI does not inherently eliminate collaboration: Ouyang and Zhang [11] examine learning analytics driven by AI in collaborative learning supported by computers. Equally, adding a discussion board or assigning partners does not establish meaningful dialogue. The relevant issue is whether an instructional arrangement creates opportunities to articulate reasoning, challenge interpretations, and negotiate understanding. The present article therefore avoids attributing deep learning or social isolation to a technology category alone.

III. REVIEW DESIGN AND ANALYTICAL PROCEDURE

A. Review Type and Scope

The study is a narrative review and conceptual synthesis, not a systematic review, statistical research synthesis, or new empirical investigation. The purpose is to clarify constructs, evaluate the scope of selected claims, and develop an instructional framework. Narrative reviews can make an integrative contribution when the purpose and interpretive boundaries are explicit [12]. The distinction from a systematic review concerns the present selection and reporting procedures, not a restriction of systematic reviews to quantitative studies [13].

The review uses the bibliography and thematic structure of an earlier 2025 research manuscript as the starting corpus. The earlier manuscript reports searches of Google Scholar, EBSCOhost, IEEE Xplore, and the ACM Digital Library. However, complete search dates, search strings for individual databases, screening records, and an auditable selection flow were not available in the supplied record. Accordingly, the reported searches are not represented as a reproducible systematic protocol, and no retrospective screening counts are provided.

Publications were retained purposively for relevance to the three research questions, contribution to a clearly identified theme, and the ability to establish the scope of a cited claim. Publications specific to programming form the focal evidence. Broader higher education studies, research on online interaction, and theoretical works provide contextual or conceptual support rather than interchangeable evidence concerning programming interventions. The selected literature extends through 2025, with earlier methodological works

retained where relevant. The review does not provide a comprehensive update of subsequently published research.

B. Verification and Evidence Classification

Targeted verification during the revision examined publication identity, study purpose, educational context, and the relationship between cited evidence and manuscript claims. Publisher records, publications hosted by authors, and institutional repositories were used where available. Some sources were accessible only through abstracts or publication records. Interpretation of partially accessible sources is restricted to the available scope and design information; unavailable methodological details and numerical effects are not inferred.

Sources were distinguished as designs or evaluations specific to programming, contextual empirical studies, research reviews, and conceptual or methodological contributions. Within the source categories, the analysis separated reported usage and perceptions from assessed learning and distinguished observational evidence from claims requiring controlled comparison. Table I provides a selective evidence map of influential sources. The map is an analytical illustration, not a complete record of a systematic screening process.

C. Synthesis and Limits of Inference

The thematic comparison follows the source manuscript's concerns: engagement, academic performance, educator perceptions, scalability, and implementation. For each theme, the analysis examines the described outcome or process, represented perspective, relevance of the setting to programming education, and inferences supported by the research design. Contradictory implications are retained rather than resolved by counting publications favorable to a technology.

No pooled effect size, formal assessment score for risk of bias, or statistic measuring agreement between coders is reported. The article also does not claim a documented qualitative coding procedure with six stages when the underlying coding materials are unavailable. The synthesis is interpretive and generates propositions. Observations from published studies are distinguished from the proposed framework and future evaluation design, preventing conceptual recommendations from being presented as completed empirical findings.

IV. THEMATIC SYNTHESIS

A. Student Engagement: Participation Is Not Understanding

Feedback supported by AI can be examined through several different forms of engagement. Behavioral participation concerns whether a student accesses a tool or persists with an activity. Cognitive engagement concerns the reasoning invested in an activity. Relational engagement concerns participation with teachers and peers. The distinctions serve an analytical purpose: a high volume of interaction with an automated tutor does not establish learner explanation, evaluation, or generalization of the information received.

TABLE I
SELECTIVE EVIDENCE MAP AND INFERENCE BOUNDARIES

Source and type	Contribution and inference limit
Troussas et al. [3] Programming design	Describes adaptive programming activities. Distinguishes adaptation from generative tutoring; no comparative effect estimate is extracted.
Liu et al. [4] Tutor evaluation	Examines human feedback for CS50 tutor improvement. Response alignment, deployment reach, and knowledge gains remain distinct outcomes.
Srivatanakul [5] Course account	Addresses blended programming through instructor reflection and student perceptions, not a trial of hybrid instruction involving AI.
Cambaz and Zhang [2] Systematic review	Synthesizes code generation uses and risks. Supports analysis of accuracy and dependence, not universal causal judgments about AI.
El-Sabagh [14] Contextual comparison	Examines engagement in adaptive online learning. A setting focused on learning skills does not establish an effect specific to programming.
Miao et al. [15] Contextual survey	Connects interaction, social presence, and online engagement at one university. Not a controlled AI comparison.
Kaplan-Rakowski et al. [16] Teacher survey	Reports teachers' generative AI perspectives. Attitudes and associations with use frequency do not establish instructional effectiveness.
Mirata et al. [7] Delphi study	Identifies implementation challenges across two universities. Supports organizational analysis, not causal claims about learner outcomes.
Almusaed et al. [6] Review analysis	Provides secondary synthesis on AI and hybrid education, not original experimental evidence of programming examination gains.

Note: The map is a selective analytical summary, not an inventory from a systematic review.

El-Sabagh [14] reports a quasiexperimental study of an adaptive online learning environment involving 118 university students in the first year, with 60 in the experimental group and 58 in the comparison group. The reported engagement advantage is relevant contextual evidence, but the setting concerns learning skills rather than programming. The study therefore cannot establish a general effect on learning programming. Adaptation based on learning styles in the study is also not treated as validation of all approaches matching instruction to learning styles. The finding illustrates a particular engagement intervention under particular conditions.

The CS50 example requires a different interpretation. Liu et al. [4] describe extensive tutor use and report earlier survey figures concerning frequency of use and perceived helpfulness. The usage indicators have different denominators and purposes from the paper's evaluation of tutor responses. The indicators support discussion of uptake, perceived utility, and system improvement, but do not establish improved engagement, examination performance, or independent programming proficiency among all users. A deployment count is evidence of reach, not an educational effect size.

Human interaction provides another relevant but bounded source of evidence. Miao et al. [15] examine relationships among interaction between teachers and students, interaction among students, social presence, and engagement using data

from 354 undergraduates at one Chinese university. The study supports attention to the social organization of online learning, but provides neither an AI comparison specific to programming nor a basis for universal causal claims about mentorship. Inclusion of the study broadens the interpretation of engagement without substituting contextual evidence for direct intervention evidence.

The synthesis suggests a measurement priority rather than a technology ranking. Evaluations should pair usage traces with evidence of students' ability to describe a misconception, explain a correction, or challenge an unsuitable recommendation. Similarly, participation in group work should be examined through contributions and reasoning, not attendance alone. A hybrid design may create conditions for richer engagement, but the achievement of richer engagement remains an empirical question beyond the scope of availability or satisfaction measures alone.

B. Academic Performance: Assisted Success and Independent Competence

Academic performance must be interpreted in relation to the conditions of observation. Correct code produced with unrestricted assistance, a solution explained after receiving hints, and an unfamiliar problem solved without assistance are different outcomes. Treating assisted and independent outcomes as equivalent risks crediting the learner with work performed by the tool. For programming education, the distinction is especially important when automated support can supply a solution and a plausible explanation.

Troussas et al. [3] provide an example of adaptive learning activities specific to programming. The source is relevant to the adaptation of instructional activity, but the present review neither extracts an unavailable comparative effect estimate nor equates the design with unrestricted generative code completion. Cambaz and Zhang [2] identify accuracy limitations, potential overreliance, and risks for novice users within the literature on code generation. The synthesis motivates closer examination of assistance practices without justifying a universal assertion of learner dependence on AI.

A second distinction concerns the level of evaluation. In Liu et al. [4], human feedback is used to improve the instructional alignment of an AI tutor. Better alignment of responses with teaching goals is a meaningful system outcome. Improved alignment nevertheless differs from evidence of conceptual retention or skill transfer after support is removed. A pedagogically improved tool may be a necessary component of a successful intervention without providing sufficient evidence of learning effectiveness.

Concerns about scaffolding are therefore better expressed conditionally. AI systems do not inherently lack scaffolding, nor do teachers automatically provide effective scaffolding. The design question is whether assistance responds to observed difficulty and is subsequently reduced when the learner can proceed independently. Drawing on distributed scaffolding [10], the proposed framework treats explanation and unaided transfer as evidence for changing responsibility rather than assuming mastery from repeated access to hints.

For future comparison, immediate task completion should be reported separately from code comprehension, debugging rationale, delayed retention, and performance on a structurally related but unfamiliar task. Separating the outcomes enables detection of a particularly important pattern: improved assisted performance without corresponding improvement in independent competence. The possibility of a divergence between assisted and independent performance is an evaluation requirement, not an empirical finding asserted by the review.

C. Educator Perceptions: Agency, Trust, and Work

Educator perspectives should not be reduced to enthusiasm versus resistance. Questions about curriculum alignment, erroneous feedback, assessment responsibility, or workload may represent legitimate professional judgments. Kaplan-Rakowski et al. [16] surveyed 147 teachers and reported generally positive perspectives on generative AI, with more frequent use associated with more positive views. The findings provide evidence about attitudes in the surveyed group, not a demonstration of a causal relationship between use and acceptance or a predictive relationship between acceptance and effective programming instruction.

Perceptions and implementation outcomes should consequently be analyzed separately. An educator can value a tool while rejecting use for a particular assessment. Reservations can also coexist with effective use under a restricted policy. A meaningful adoption study should examine delegated teaching functions, inspected outputs, and intervention decisions. Operational evidence is more informative than an assumption of direct correspondence between agreement with broad statements about AI and classroom practice.

The instructor's proposed role in the hybrid framework is not to approve every automated response. Instead, the teacher defines acceptable assistance, selects the evidence required for progression, and reviews cases involving problematic learner explanations or system behavior. The allocation is intended to preserve professional judgment while avoiding an impractical requirement for universal manual checking. Improvements in trust or workload must be investigated rather than assumed.

Faculty development should likewise be framed as an intervention to evaluate. Relevant activities might include analyzing misleading feedback, designing assessments requiring explanation, interpreting learning records, and adjusting escalation rules. Training attendance alone would not demonstrate readiness. More useful evidence would include the quality of teachers' decisions in representative instructional cases and the capacity to revise a workflow when the technology or learner population changes.

D. Scalability: Reach Does Not Establish Sustainability

Scalability has at least three dimensions in the present review: the number of learners able to access assistance, the institution's capacity to maintain the service, and the ability to preserve educational quality as participation grows. The dimensions should not be collapsed into a single claim

equating automation with inexpensive personalized teaching. An intervention can have broad reach while transferring substantial work to system configuration, monitoring, accessibility support, and assessment redesign.

Mirata et al. [7] used a Delphi process with four stages and experts from universities in Switzerland and South Africa. Differences in implementation priorities across the settings make the study useful for examining the influence of context on adoption. The study does not establish a universal ranking of barriers or a numerical return on investment. The contribution to the present synthesis concerns the importance of organizational and teaching conditions in evaluating adaptive learning, rather than treating institutional conditions as background details.

Sustainable implementation should therefore be assessed through a workload and resource account. Proposed measures include preparation time, monitoring time, escalation volume, service interruptions, and the availability of assistance without AI. The measures would reveal whether automation reduces routine work, relocates existing responsibilities, or creates additional tasks. A headline cost per interaction cannot substitute for the total resources needed to sustain an instructional arrangement within a course.

Human review creates a particular design tension. Reviewing every interaction can erode the scalability sought through automation; omitting review can leave inaccurate or pedagogically unsuitable guidance unchallenged. A design organized around evidence checkpoints addresses the tension by specifying review triggers and sampling selected interactions. The efficiency and sensitivity of the procedures remain hypotheses for evaluation, especially when class sizes, assessment demands, and support staffing differ.

E. Implementation, Educational Culture, and Equity

Software reliability, curriculum integration, privacy, algorithmic fairness, financial access, and educator readiness are treated as connected implementation challenges. The concerns should be examined as conditions of participation rather than as a checklist appended after deployment. Yan et al. [8] situate generative AI within broader questions about ethical use, model limitations, and competencies needed for informed engagement with AI. The present framework translates the concerns into proposed instructional and governance responsibilities.

At the course level, the proposed safeguards include defined purposes for collecting interaction records, limited access to the records, an explicit retention policy, and a route for students to question automated guidance. Automated recommendations should not become unreviewable decisions about a student's ability or progression. The safeguards are design recommendations, not determinations of applicable institutional laws or claims of compliance with legal or security standards.

Educational culture enters the analysis through the organization of help, authority, and participation. A learner may encounter expectations about whether to ask questions publicly, challenge an explanation, or acknowledge receiving

assistance. The review does not establish variation in expectations across countries or cultural groups. Rather, language of instruction, local practices for seeking help, and expectations of teacher authority are proposed contextual variables for future study. Local investigation should replace inference from national labels.

Equitable access is similarly more demanding than distributing the same software. A proposed evaluation should examine reliable access to the service, comprehension of explanations, usability of the interface, and availability of human assistance during service failures. Evaluation should also examine whether unsuccessful attempts are interpreted as lack of ability despite reflecting barriers to participation. The potential mechanisms require empirical testing and do not represent subgroup effects demonstrated in the selected literature.

Taken together, the thematic synthesis supports a conditional conclusion. The literature supplies reasons to investigate coordinated AI and human instruction, but does not establish a universally superior hybrid model. The practical and research task is to specify support allocation, evidence of understanding, and accountability to learners and educators.

V. A FRAMEWORK FOR HYBRID INSTRUCTION

A. Rationale and Components

The proposed framework addresses a specific coordination problem: a learner can move quickly from requesting assistance to submitting a correct solution without making understanding visible. Evidence gating refers to instructional transitions contingent on observable learner performance rather than tool use alone. The framework combines six phases: diagnosis, adaptive practice, explanation and testing, human dialogue, unaided transfer, and instructional adjustment. Table II specifies the proposed evidence and responsibility associated with each phase.

The contribution is not the invention of diagnosis, feedback, collaboration, or assessment as separate practices. The contribution lies in organizing established practices around explicit decisions about when responsibility should remain with the system, move to a teacher or peer, or return fully to the learner. The organization applies dynamic personalization and distributed scaffolding to the transition problem identified in the thematic synthesis [9], [10]. No claim of empirical validation or priority over all existing frameworks is made.

Three decisions give the sequence instructional meaning. The support decision concerns assistance justified by the learner's current difficulty. The explanation decision concerns the learner's ability to account for the proposed correction and test the correction against the task. The transfer decision concerns the learner's ability to apply the relevant idea when assistance is reduced. The decisions need not be implemented as automated scores. A teacher can make instructional judgments from short demonstrations, code traces, or structured discussions.

TABLE II
PROPOSED INSTRUCTIONAL SEQUENCE AND EVIDENCE
CHECKPOINTS

Phase	Evidence and instructional responsibility
1. Diagnose	Learner predicts, traces, or attempts a task before assistance. Teacher defines the target concept; initial errors guide support.
2. Adapt practice	System or teacher supplies bounded help matched to observed difficulty. Records identify the kind of support, not only completion.
3. Explain and test	Learner explains a correction and constructs relevant tests. A correct product without a coherent account does not establish understanding.
4. Discuss and review	Peer or instructor examines unresolved reasoning or questionable feedback. Human dialogue addresses ambiguity remaining after assistance.
5. Check transfer	Learner attempts a related unfamiliar task under reduced or no assistance. Teacher evaluates the specified concept independently.
6. Adjust support	Teacher or approved adaptive process revises the next task and level of help using evidence collected across the sequence.

Note: The sequence is a conceptual proposal. Effects on learning and workload have not been evaluated in the present study.

B. Illustrative Programming Sequence

A hypothetical introductory exercise involves a loop omitting the final item in a collection. The student first predicts the output and identifies where the behavior differs from the requirement. An approved tutor then offers a bounded hint about the loop condition rather than supplying a complete rewritten program. The learner proposes a correction and explains why the original condition failed.

Before accepting the solution, the activity requires the learner to construct tests for an empty collection, a single item, and a longer sequence. When the explanation remains inconsistent with the test results, the case is routed to a peer discussion or instructor review. When the explanation is coherent, the learner attempts a related problem involving boundary conditions without the same hint. The next practice task is selected from the collected evidence rather than from successful execution of the initial program alone.

The example illustrates a proposed teaching sequence, not an observation from the reviewed studies. The example also exposes failure conditions. A generated explanation copied without comprehension should not satisfy the explanation requirement; successful reproduction of a familiar example should not be mistaken for transfer. Teachers must therefore choose prompts and subsequent tasks capable of making the relevant reasoning visible rather than merely eliciting a polished answer.

C. Responsibilities and Boundary Conditions

The instructor retains responsibility for learning objectives, assistance policies, and the interpretation of ambiguous cases. The system supplies approved support and records information needed for instructional decisions. The learner remains responsible for testing and explaining a solution. Peers provide opportunities to compare interpretations rather than acting as a

substitute for unresolved instructor support. The allocation of responsibilities should be made explicit in the course design.

The framework is not a requirement to gate every minor coding action. Excessively frequent checks could fragment practice and create unnecessary work. Implementation should begin with selected concepts or tasks requiring particular attention to unaided competence. Review triggers and the frequency of transfer checks should then be adjusted on the basis of evaluation. The appropriate degree of control remains a design question, not a universal threshold established by the article.

VI. RESEARCH PROPOSITIONS AND EVALUATION AGENDA

A. Propositions for Empirical Testing

P1: Explanation and transfer. Under comparable task and access conditions, practice supported by AI and requiring learners to explain and test corrections is expected to produce stronger unaided transfer than practice supported by AI and emphasizing completed solutions alone. Improved assisted completion without a transfer advantage would not support the proposition. The proposed mechanism is active verification of reasoning, rather than the mere presence of an AI tool.

P2: Contingent withdrawal of support. Adjusting and reducing assistance in response to demonstrated understanding is expected to support subsequent independent performance more effectively than maintaining the same level of assistance throughout practice. A study should examine learning and the cost of premature withdrawal. The prediction therefore concerns appropriately contingent support, not minimizing help irrespective of learner need.

P3: Teacher agency and implementation. Teacher participation in defining assistance boundaries and review procedures is expected to be associated with more sustained and instructionally coherent implementation than deployment centered only on technical orientation. Relevant outcomes include enacted classroom practice, intervention quality, and workload, not satisfaction alone. Comparative evidence would be needed to distinguish the contribution of teacher agency from prior enthusiasm or institutional support.

P4: Context and equitable participation. The effects of a hybrid arrangement are expected to vary with access conditions, prior programming experience, language demands, and the availability of human support. A favorable overall average may therefore coexist with uneven participation or learning. The proposition concerns measurable contextual mechanisms and must not be interpreted as a claim about fixed characteristics of cultural groups.

B. Comparative Study Design

A future evaluation could compare a clearly specified course led by an instructor, a course supported by AI under a defined assistance policy, and a course implementing the proposed framework with evidence checkpoints. The condition led by an instructor should include credible existing teaching practices rather than an artificially weak comparator.

Curriculum objectives, task difficulty, instructional time, and planned human contact should be documented and made as comparable as the research question permits.

Where feasible, allocation by course section could reduce contamination between instructional conditions. Where random allocation is impractical, a matched or carefully adjusted quasiexperimental design could be used, with explicit acknowledgment of causal limitations. Planning the sample size should follow the primary outcome, anticipated clustering, and the smallest educationally meaningful difference. The article provides no predetermined sample requirement or expected effect size. The comparison and analysis plan should be specified before outcome inspection.

The primary outcome should be an independently assessed transfer task completed under clearly stated assistance conditions. Secondary outcomes could include immediate performance, code comprehension, debugging explanations, and delayed retention. Engagement measures should combine participation traces with evidence of reasoning. Human assessment of explanations should use a common rubric, and assessors should be unaware of condition where feasible. Assistance available during each assessment must be reported rather than assumed.

Analysis should account for baseline proficiency and the nesting of students within sections or instructors. Attrition, missing observations, and deviations from assigned support should be reported transparently. Estimates with uncertainty are more informative than isolated significance statements. Contextual comparisons should be planned and adequately supported by the design; an absence of a detected subgroup difference should not automatically be interpreted as evidence of equal benefit.

C. Process, Workload, and Qualitative Evidence

Outcome comparison alone would not explain why the framework succeeds or fails. A process evaluation should examine the kinds of hints delivered, the proportion of cases escalated, teacher responses to escalated cases, and whether independent checks actually occur. Records of model version, configuration, assistance policy, and substantive changes during a course are needed to identify the delivered intervention.

Interviews or structured reflections with students and educators could examine how support is interpreted, why recommendations are accepted or rejected, and how classroom norms influence practices for seeking help. Sampling should include nonusers, struggling learners, and participants experiencing access problems, rather than only satisfied adopters. Qualitative findings would be used to explain mechanisms and boundary conditions, not to manufacture general prevalence estimates.

A complementary resource analysis should examine preparation, maintenance, monitoring, and review time alongside service costs and learning outcomes. Evaluation over longer periods could then assess whether the same arrangement remains useful across successive courses and changing tools. The research program would test the

framework's educational and organizational claims without assuming sustainable improvement from initial enthusiasm.

VII. LIMITATIONS

The review is limited by purposive selection seeded from an existing manuscript. Because the original search log and full screening record were unavailable, completeness cannot be established and publication or selection bias cannot be quantified. The selected publications extend through 2025 and should not be read as a comprehensive account of later developments. Some claims were verifiable only through abstracts or publication records. Partially accessible sources are therefore not used to reconstruct unavailable results.

The evidence also spans different technologies, study designs, learning settings, and outcome measures. Contextual findings from online education or teacher surveys do not establish effects in programming courses. Reviews and conceptual papers contribute interpretation but are not additional independent intervention samples. Differences across the sources preclude a defensible pooled estimate or a universal ranking of instructional models within the present study.

The proposed framework for hybrid instruction remains conceptual. The coordination rules, workload implications, and potential benefits have not been tested in the present study. Explanation tasks can also be assisted, and transfer checks may capture only selected aspects of competence. The proposed cultural and equity considerations identify questions for investigation rather than demonstrated effects across cultures. Generalization therefore depends on future empirical work in clearly described instructional and institutional settings.

VIII. CONCLUSION

A research comparison of programming education supported by AI and traditional programming education must distinguish system functions, demonstrated learner competence, and continuing teacher responsibilities. The selected literature supports investigation of adaptive assistance, human feedback, and coordinated instruction. The literature does not justify treating usage as cognitive engagement, assisted correctness as independent competence, or hybrid delivery as a guaranteed improvement.

The review contributes a differentiated interpretation of evidence and a proposed framework for coordinating diagnosis, adaptive practice, explanation, human dialogue, unaided transfer, and instructional adjustment. The central design principle requires changes in assistance to follow evidence of understanding. Human guidance and automation are consequently treated as instructional resources requiring explicit coordination and evaluation, rather than as mutually exclusive alternatives.

The next step is a comparative evaluation of learning, teacher agency, and equitable participation under transparent assistance conditions and clearly specified institutional contexts, not a stronger assertion of technological superiority.

Future research would establish whether hybrid instruction organized around evidence checkpoints offers advantages, identify the learners benefiting from particular arrangements, and determine the institutional work required for sustained implementation.

DATA AVAILABILITY

No new data from participants were collected or analyzed. The review draws on the publications cited in the reference list; the instructional sequence and research propositions are conceptual proposals.

REFERENCES

- [1] B. A. Becker, P. Denny, J. Finnie-Ansley, A. Luxton-Reilly, J. Prather, and E. A. Santos, "Programming is hard—or at least it used to be: Educational opportunities and challenges of AI code generation," *arXiv:2212.01020*, 2022, doi: 10.48550/arXiv.2212.01020.
- [2] D. Cambaz and X. Zhang, "Use of AI-driven code generation models in teaching and learning programming: A systematic literature review," *Proc. 55th ACM Tech. Symp. Comput. Sci. Educ. V. 1*, 2024, pp. 172–178, doi: 10.1145/3626252.3630958.
- [3] C. Troussas, A. Krouska, and C. Sgouropoulou, "A novel teaching strategy through adaptive learning activities for computer programming," *IEEE Trans. Educ.*, vol. 64, no. 2, pp. 103–109, 2021, doi: 10.1109/TE.2020.3012744.
- [4] R. Liu, J. Zhao, B. Xu, C. Perez, Y. Zhukovets, and D. J. Malan, "Improving AI in CS50: Leveraging human feedback for better learning," *Proc. 56th ACM Tech. Symp. Comput. Sci. Educ. V. 1*, 2025, pp. 715–721, doi: 10.1145/3641554.3701945.
- [5] T. Srivatanakul, "Emerging from the pandemic: Instructor reflections and students' perceptions on an introductory programming course in blended learning," *Educ. Inf. Technol.*, vol. 28, no. 5, pp. 5673–5695, 2023, doi: 10.1007/s10639-022-11328-6.
- [6] A. Almusaed, A. Almssad, I. Yitmen, and R. Z. Homod, "Enhancing student engagement: Harnessing "AIED"'s power in hybrid education—a review analysis," *Educ. Sci.*, vol. 13, no. 7, Art. no. 632, 2023, doi: 10.3390/educsci13070632.
- [7] V. Mirata, F. Hirt, P. Bergamin, and C. van der Westhuizen, "Challenges and contexts in establishing adaptive learning in higher education: Findings from a Delphi study," *Int. J. Educ. Technol. High. Educ.*, vol. 17, Art. no. 32, 2020, doi: 10.1186/s41239-020-00209-y.
- [8] L. Yan, S. Greiff, Z. Teuber, and D. Gašević, "Promises and challenges of generative artificial intelligence for human learning," *Nat. Hum. Behav.*, vol. 8, no. 10, pp. 1839–1850, 2024, doi: 10.1038/s41562-024-02004-5.
- [9] L. Tetzlaff, F. Schmiedek, and G. Brod, "Developing personalized education: A dynamic framework," *Educ. Psychol. Rev.*, vol. 33, no. 3, pp. 863–882, 2021, doi: 10.1007/s10648-020-09570-w.
- [10] S. Puntambekar, "Distributed scaffolding: Scaffolding students in classroom environments," *Educ. Psychol. Rev.*, vol. 34, no. 1, pp. 451–472, 2022, doi: 10.1007/s10648-021-09636-3.
- [11] F. Ouyang and L. Zhang, "AI-driven learning analytics applications and tools in computer-supported collaborative learning: A systematic review," *Educ. Res. Rev.*, vol. 44, Art. no. 100616, 2024, doi: 10.1016/j.edurev.2024.100616.
- [12] R. F. Baumeister and M. R. Leary, "Writing narrative literature reviews," *Rev. Gen. Psychol.*, vol. 1, no. 3, pp. 311–320, 1997, doi: 10.1037/1089-2680.1.3.311.
- [13] A. P. Siddaway, A. M. Wood, and L. V. Hedges, "How to do a systematic review: A best practice guide for conducting and reporting narrative reviews, meta-analyses, and meta-syntheses," *Annu. Rev. Psychol.*, vol. 70, pp. 747–770, 2019, doi: 10.1146/annurev-psych-010418-102803.
- [14] H. A. El-Sabagh, "Adaptive e-learning environment based on learning styles and its impact on development students' engagement," *Int. J. Educ. Technol. High. Educ.*, vol. 18, Art. no. 53, 2021, doi: 10.1186/s41239-021-00289-4.
- [15] J. Miao, J. Chang, and L. Ma, "Teacher–student interaction, student–student interaction and social presence: Their impacts on learning engagement in online learning environments," *J. Genet. Psychol.*, vol. 183, no. 6, pp. 514–526, 2022, doi: 10.1080/00221325.2022.2094211.
- [16] R. Kaplan-Rakowski, K. Grotewold, P. Hartwick, and K. Papin, "Generative AI and teachers' perspectives on its implementation in education," *J. Interact. Learn. Res.*, vol. 34, no. 2, pp. 313–338, 2023.