

Pedagogical Noise in GenAI-Supported Algorithmisation: Scaffolding and Substitution in Upper-Secondary Informatics

Martin Žáček* [0000-0002-9062-4233]

Department of Informatics and Computers, Faculty of Science, University of Ostrava, Ostrava, Czech Republic

e-mail: martin.zacek@osu.cz

Abstract. Generative artificial intelligence (GenAI) is increasingly used in school informatics, yet technically correct assistance can still interfere with the reasoning through which learners develop algorithmic competence. This conceptual paper focuses on upper-secondary algorithmisation, defined here as the staged design and representation of algorithms from problem interpretation and decomposition through pseudocode or flowcharts to debugging and verification. It proposes pedagogical noise as an integrative, task-specific diagnostic lens for identifying misalignment between an AI contribution and the competence-forming work that should remain learner-owned at a particular stage. A structured and traceable conceptual synthesis compares this lens with cognitive offloading, productive struggle, scaffolding failure, automation bias, performance-learning dissociation, feedback overload, reduced epistemic agency, overreliance, hallucination, and academic misuse. The synthesis develops seven proposed forms of pedagogical noise, each tied to a primary diagnostic dimension: completeness, timing, transparency, learner judgement, causal debugging, volume/actionability, and ownership/accountability. Worked cases show how technically correct GenAI support may function as scaffolding or substitution depending on learner state, timing, output granularity, transparency, and agency. The contribution is not a new theory of learning or a claim that GenAI is inherently harmful; it is a bounded diagnostic framework and a set of design propositions for preserving learner-owned reasoning in upper-secondary algorithmisation.

Keywords: generative artificial intelligence; pedagogical noise; algorithmisation; upper-secondary informatics; computational thinking; AI literacy.

1. Introduction

Generative artificial intelligence (GenAI) can explain code, generate pseudocode, translate between representations, diagnose errors, and provide dialogue-based feedback. These affordances are relevant to upper-secondary informatics because novice learners often strug-

* Corresponding author

gle with abstraction, decomposition, algorithm design, debugging, and movement from informal reasoning to formal representations (Brennan & Resnick, 2012; Grover & Pea, 2013; Lye & Koh, 2014; Pears et al., 2007; Shute et al., 2017). School-level studies already indicate that structured GenAI use can support learning performance, computational thinking, critical thinking, or engagement under particular instructional designs, including reflective debugging, dialogic feedback, and collaborative programming (Fang et al., 2025; Gong et al., 2025; Hussein et al., 2026; Zhao et al., 2025).

At the same time, the educational value of GenAI cannot be inferred from task completion alone. Overreliance, understood here as excessive dependence on AI support, may reduce independent attempts and verification; hallucination refers to generated content that is false or unreliable; weakened self-regulation refers to reduced planning, monitoring, and revision of one's own work. Programming studies report difficulties correcting AI errors, patterns of solution seeking, and uncertainty about independent understanding (Akçay et al., 2025; Kohen-Vacs et al., 2025; Liu et al., 2026; López-Pernas et al., 2025). These risks matter even when the generated artefact is technically plausible or correct.

This creates a specific problem for algorithmisation. Algorithmisation is treated here as the staged design and representation of algorithms: learners interpret a problem, identify inputs and constraints, decompose it, formulate a procedure, move among natural language, pseudocode, flowcharts, and code, debug, verify, and justify the solution. The educational target is therefore not merely a correct artefact but the formation of reasoning that learners can later reproduce, explain, transfer, and evaluate independently.

The established performance-learning distinction already warns that better immediate task performance does not necessarily imply durable learning (Soderstrom & Bjork, 2015). Recent programming evidence makes this concern concrete: Bassner et al. (2026) found that AI-supported students achieved higher task scores without corresponding gains on measured learning outcomes, while Liu et al. (2026) described recurring movement between scaffolding and offloading in AI-assisted programming. Structured integration can, however, support higher-order engagement when it explicitly requires decomposition, debugging, validation, and reflection (Nathaniel et al., 2026). The unresolved issue is therefore not whether performance and learning can diverge; it is how educators can diagnose which AI contribution is misaligned with the competence-forming work of a particular algorithmisation phase.

This paper addresses that narrower gap by proposing pedagogical noise as a relational diagnostic lens. Pedagogical noise is defined as misalignment between an AI contribution and the learner-owned reasoning that should remain visible, accountable, and practised at a specific stage of algorithmisation. The construct is intended to integrate, not replace, established concepts such as cognitive offloading, scaffolding failure, automation bias, feedback overload, and reduced epistemic agency. Its proposed increment is task-specific diagnosis: what learner-owned reasoning step is being displaced, at what phase, and by what form of AI contribution.

Accordingly, the paper asks one conceptual question: What does a pedagogical-noise lens allow educators and researchers to diagnose in GenAI-supported upper-secondary algorithmisation that is not sufficiently specified by neighbouring constructs alone? The contribution is a structured and traceable conceptual synthesis, a seven-form diagnostic taxonomy with explicit primary dimensions, a relational scaffolding-substitution model, illustrative worked cases, and bounded design propositions for upper-secondary informatics.

2. Background and related work

2.1 Algorithmisation and competence-forming reasoning

In school informatics, algorithmic competence involves more than language syntax. Computational-thinking scholarship emphasises decomposition, abstraction, algorithmic design, debugging, generalisation, and evaluation (Brennan & Resnick, 2012; Grover & Pea, 2013; Shute et al., 2017; Weintrop et al., 2016). Bebras-related work similarly demonstrates that algorithmic and computational ideas can be made visible through tasks that require reasoning rather than full programming environments (Dagienė & Futschek, 2008; Izu et al., 2017; Mannila et al., 2014). Curriculum-oriented work in school computer science stresses understanding the nature and logic of programs rather than treating code production as the sole endpoint (Brodnik et al., 2021; Sentance & Csizmadia, 2017; Sentance et al., 2018).

For upper-secondary learners, important competence-forming work occurs in transitions: from a verbal task to a decomposition, from a decomposition to pseudocode or a flowchart, from a representation to executable code, and from observed failure to causal debugging and verification. GenAI can assist at every transition, but assistance may also hide the transition. A learner can therefore possess a polished artefact without having enacted the reasoning normally used to construct it.

2.2 Performance support, learning, and structured GenAI use

The distinction between immediate performance and learning is established in educational psychology (Soderstrom & Bjork, 2015). GenAI makes the distinction especially salient because fluent code, pseudocode, explanations, and repairs can improve visible products quickly. In an undergraduate programming experiment, Bassner et al. (2026) reported markedly better exercise performance under AI support but no corresponding advantage in pre-post knowledge gains or code comprehension; a scaffolded hint-first tutor preserved motivational benefits better than unrestricted answer-giving assistance. This finding supports the performance-learning problem but does not itself provide an upper-secondary diagnostic taxonomy.

Liu et al. (2026) offer the closest programming-specific conceptual neighbour. Their grounded theory describes a tension between domain mastery and tool mastery, with recurring scaffolding and offloading loops and problems of verification and metacognitive calibration. The present paper does not claim the general scaffolding-offloading problem as novel. Instead, it narrows the unit of analysis to a particular AI contribution within a particular algorithmisation phase and asks whether that contribution displaces learner-owned competence-forming work.

A complementary line of work concerns structured integration. Nathaniel et al. (2026), in higher education, reported that a structured GenAI programming framework increased pedagogical alignment, cognitive engagement, and selected learning outcomes compared with unguided use. Direct upper-secondary studies likewise suggest benefits under structured designs, including reflective debugging (Fang et al., 2025), staged dialogic feedback (Gong et al., 2025), and human-computer collaborative programming (Zhao et al., 2025). These studies support the importance of pedagogical design, but they do not establish the seven-form taxonomy proposed here.

2.3 Digital competence, AI literacy, and prompt literacy

Digital competence and AI literacy shape whether learners can use GenAI without surrendering judgement. DigComp 2.2 treats digital competence as including information and data literacy, content creation, safety, and problem solving (Vuorikari et al., 2022), while DigCompEdu emphasises educators' capacity to select and orchestrate digital learning environments (Redecker, 2017). AI-literacy frameworks add critical evaluation, ethical awareness, and informed use of AI systems (Long & Magerko, 2020; Ng et al., 2021; UNESCO, 2024a, 2024b).

Prompt literacy is relevant because the form of a request influences the granularity and transparency of AI support. Federiakin et al. (2024) position prompt engineering as an emerging competence; this paper extends that idea to algorithmisation by arguing that prompts can be designed to request hints, traces, counterexamples, or critiques rather than full solutions. This is an argument of the present framework, not a demonstrated causal effect of prompt literacy.

2.4 Neighbouring constructs and the conceptual gap

Pedagogical noise overlaps with several established constructs. Cognitive offloading concerns the use of external resources to reduce internal cognitive demand (Risko & Gilbert, 2016). Productive failure and productive struggle concern the learning value of effortful problem engagement before consolidation (Kapur, 2008, 2016). Scaffolding concerns calibrated support that structures and problematises learner activity and is progressively faded (Pea, 2004; Reiser, 2004; Wood et al., 1976). Automation bias concerns inappropriate reliance on automated suggestions (Parasuraman & Manzey, 2010; Parasuraman & Riley, 1997). Feedback research highlights the need for feedback to be interpretable,

actionable, and appropriately bounded (Hattie & Timperley, 2007; Narciss, 2008; Shute, 2008). Epistemic-agency research concerns responsibility for knowledge construction and judgement (Damşa et al., 2010; Markauskaite & Goodyear, 2017).

GenAI-specific concerns add overreliance, hallucination, and academic misuse. Overreliance describes excessive dependence on AI assistance and may involve repeated outsourcing or uncritical acceptance (Kohen-Vacs et al., 2025; Liu et al., 2026). Hallucination concerns unreliable or fabricated output (Kasneci et al., 2023). Academic misuse concerns violations of authorship, disclosure, or assessment rules; related high-school pseudocode work demonstrates why provenance and verification matter in AI-mediated programming tasks (Liu et al., 2025).

None of these concepts is rejected or superseded here. The proposed gap is finer-grained. Existing constructs can tell us that offloading, poor scaffolding, excessive trust, or agency loss may be occurring; pedagogical noise asks what AI contribution is misaligned with what learner-owned reasoning step in what algorithmisation phase. Table 1 summarises this incremental positioning and explicitly source-supports each neighbouring construct.

Table 1. Pedagogical noise in relation to neighbouring constructs

Neighbouring construct	Primary concern	Overlap in GenAI-supported algorithmisation	What the proposed pedagogical-noise lens adds	Key sources
Cognitive offloading	External tools reduce internal cognitive demand.	AI may offload decomposition, debugging, or verification.	Asks whether the offloaded work is specifically the competence-forming work the task is intended to preserve.	Risko & Gilbert (2016); Liu et al. (2026)
Loss of productive struggle	Beneficial effort or problem engagement is removed.	AI can remove difficulty before learners attempt interpretation or solution design.	Locates the displaced effort in a named algorithmisation phase and learner-owned reasoning step.	Kapur (2008, 2016)
Scaffolding failure	Support is poorly calibrated, excessive, insufficient, or not faded.	AI support may be too complete, opaque, or badly timed.	Treats misalignment as a task-specific diagnostic event rather than a general failure of support.	Pea (2004); Reiser (2004); Wood et al. (1976)
Automation bias	Automated suggestions receive excessive trust.	Learners may accept fluent code or pseudocode without verification.	Also covers technically correct output that is pedagogically misaligned even when the learner does not blindly trust it.	Parasuraman & Riley (1997); Parasuraman & Manzey (2010)
Performance-learning dissociation	Immediate performance can improve without durable learning.	AI can improve artefacts or scores while independent competence remains unchanged.	Identifies which AI contribution and which displaced reasoning step may explain the dissociation in an algorithmisation task.	Soderstrom & Bjork (2015); Bassner et al. (2026)

Neighbouring construct	Primary concern	Overlap in GenAI-supported algorithmisation	What the proposed pedagogical-noise lens adds	Key sources
Feedback overload	Feedback is excessive, complex, or insufficiently actionable.	GenAI can produce too many explanations or alternatives.	Treats quantity/actionability as one diagnostic dimension within a broader task-output alignment framework.	Hattie & Timperley (2007); Narciss (2008); Shute (2008)
Reduced epistemic agency	Learners lose responsibility for knowledge construction and judgement.	Correctness, verification, or authorship may be delegated to AI.	Locates possible agency loss in concrete activities such as tracing, testing, and justification.	Damşa et al. (2010); Markauskaite & Goodyear (2017)
Overreliance	Learner dependence on AI assistance.	Repeated solution-seeking or acceptance may replace independent attempts.	Diagnoses the particular task contribution that invites substitution rather than defining dependence globally.	Kohen-Vacs et al. (2025); Liu et al. (2026)
Hallucination	Generated content is false or unreliable.	AI may invent a rule, condition, or explanation.	Covers technically correct output that still displaces competence-forming reasoning.	Kasneci et al. (2023)
Academic misuse	Authorship, disclosure, or assessment rules are violated.	AI-generated pseudocode or code may be submitted inappropriately.	Also applies where AI use is permitted and disclosed but pedagogically misaligned.	Liu et al. (2025)

3. Method: conceptual synthesis procedure

3.1 Conceptual article type and scope

This is a conceptual/theoretical Research Article. It does not estimate the effectiveness of GenAI, and it is not a systematic review. Its purpose is to develop and delimit a proposed diagnostic construct for upper-secondary algorithmisation. The evidence serves four roles: defining competence-forming work, positioning the construct against neighbouring concepts, identifying recurrent GenAI-related tensions, and bounding the resulting propositions.

The substantive scope is upper-secondary informatics, especially decomposition, pseudocode, flowcharts, procedural design, debugging, verification, and justification. Literature from broader programming education, higher education, psychology, and the learning sciences is used as transferred conceptual or empirical evidence and is labelled accordingly rather than presented as direct upper-secondary evidence.

3.2 Literature fields, period, and source selection

The synthesis covered four literature fields: (1) school informatics, computational thinking, and algorithmic reasoning; (2) GenAI in programming and school informatics; (3) learning sciences and educational psychology on scaffolding, productive struggle, cognitive offloading, automation bias, feedback, epistemic agency, and learning versus performance; and (4) digital competence, AI literacy, and prompt literacy. Foundational learning literature from 1976 onward was retained where conceptually load-bearing, while GenAI literature was concentrated in 2023–2026 because the field is recent.

Sources were selected purposively, not exhaustively. Priority was given to peer-reviewed studies and established theoretical work that (a) described competence-forming reasoning in algorithmisation or programming; (b) documented opportunities or risks of GenAI-supported programming; (c) defined a construct that could be confused with pedagogical noise; or (d) supplied evidence for boundary conditions. Policy frameworks were retained only where they directly supported AI-literacy or digital-competence framing.

3.3 Derivation and traceability of the taxonomy

The taxonomy was developed iteratively rather than produced by frequency counting. First, a deductive pass used scaffolding, productive struggle, cognitive offloading, feedback, automation, and epistemic-agency literature to identify general dimensions along which assistance can become misaligned with learning. Second, a comparative pass examined recent programming and school-informatics evidence for recurrent situations such as direct solution seeking, shallow debugging repair, difficulty evaluating generated output, excessive feedback, and uncertainty about responsibility. Third, an integrative pass retained a category only when it could be tied to a distinct primary diagnostic dimension and a concrete learner-owned activity in algorithmisation.

Positive and limiting evidence were both used. Structured interventions were treated as evidence of potentially low-noise conditions, whereas reports of offloading, correction difficulty, or unguarded solution seeking were used to identify high-noise conditions. The resulting categories are proposed diagnostic distinctions, not empirically validated latent variables and not an exhaustive taxonomy. Categories may co-occur within a single interaction. Table 2 documents the synthesis path so that the derivation is structured and traceable without implying a reproducible systematic-review protocol.

Table 2. Structured and traceable conceptual-synthesis procedure

Step	Purpose	Decision rule	Result in this paper
1. Field mapping	Identify literature needed to define algorithmisation, GenAI support, learning support, and AI literacy.	Retain fields only when they directly define the problem, a neighbouring construct, or a boundary condition.	Four literature fields and a bounded upper-secondary target context.

Step	Purpose	Decision rule	Result in this paper
2. Construct comparison	Test whether the proposed construct merely renames an established concept.	Compare focus, unit of analysis, and diagnostic question against closest neighbours.	Relational, task-specific misalignment retained as the proposed increment.
3. Category generation	Identify recurrent forms of task-output misalignment.	Retain a form only if it maps to a distinct primary diagnostic dimension.	Seven proposed forms tied to completeness, timing, transparency, judgement, causal debugging, volume/actionability, and ownership/accountability.
4. Case operationalisation	Check whether the construct can diagnose concrete algorithmisation situations.	Specify intended competence, learner-owned reasoning, AI work, evidence of scaffolding/substitution, and risk.	Three illustrative worked cases.
5. Claim bounding	Prevent conceptual claims from being presented as empirical findings.	Classify statements as established background, proposed framework, design proposition, or validation need.	Evidence-bounded Discussion and explicit limitations.

3.4 Relationship to prior work, search dataset, and separate review work

The present article is thematically related to the earlier conference contribution Algorithmization with AI (Žáček & Konečná, 2024). That two-page extended abstract introduced algorithmic thinking, listed common algorithm representations, and discussed practical affordances of then-available AI tools for generating, explaining, modifying, and debugging algorithms in school contexts. It did not introduce pedagogical noise, a relational definition of AI-learning misalignment, the seven-form taxonomy, diagnostic dimensions, the worked cases used here, or the present scaffolding-substitution framework.

The literature orientation for this conceptual synthesis was also informed by bibliographic search material deposited in Mendeley Data (Žáček, 2026; DOI 10.17632/d246s355tz.1). The dataset contains Web of Science and Scopus exports originally assembled for a separate review project on generative AI in upper-secondary algorithmisation and programming education. Some source overlap is therefore expected. The dataset itself is not treated as empirical evidence in this article, and no screening counts or review findings are reported as results here. A separate review manuscript was developed from that dataset but has not been published; it is distinct from the present conceptual article in aim, method, material, and contribution.

The unique contribution of the present paper is therefore not the discovery of the general performance-learning problem, which is already established, nor a reuse of the earlier tool-oriented discussion. It is the proposed upper-secondary algorithmisation-specific relational diagnostic framework: a definition of pedagogical noise, comparison with the closest constructs, seven proposed forms with primary diagnostic dimensions, worked cases, and bounded design propositions.

4. Conceptual synthesis: the pedagogical-noise framework

4.1 *Relational definition and construct boundary*

Pedagogical noise is proposed as the relational misalignment between an AI contribution and the competence-forming learner work that should remain visible, accountable, and learner-owned at a particular stage of algorithmisation. It is not a property of an output in isolation. The same technically correct output can be low-noise in one situation and high-noise in another because the learning goal, learner state, timing, or accountability structure differs.

The construct is therefore best understood as an integrative diagnostic lens, not an empirically established psychological mechanism. It can coexist with cognitive offloading, scaffolding failure, automation bias, overreliance, or reduced epistemic agency. Those constructs may describe causes, behaviours, or consequences; pedagogical noise names the task-specific misalignment to be diagnosed. The paper no longer treats epistemic risk as an independent framework component. Reduced epistemic agency is instead treated as one possible consequence when responsibility for interpreting, testing, or justifying a solution is displaced.

Four questions operationalise the relational definition: What competence is the task intended to form? What reasoning has the learner already performed? What work does the AI contribution perform? What learner-owned work remains visible in explanation, testing, or revision? These questions provide the bridge from a broad concern about AI dependence to diagnosis in a specific algorithmisation phase.

4.2 *Seven proposed forms and their primary diagnostic dimensions*

The seven forms are deliberately distinguished by a primary diagnostic dimension. Over-completion is primarily about completeness: the AI performs more of the solution than the learner's difficulty requires. Premature solutioning is primarily about timing: a solution arrives before the learner has completed the intended interpretation or decomposition. The two can overlap, but they are not identical; a full answer can be well timed after an independent attempt, whereas even a partial answer can be premature if it arrives before the target reasoning phase.

Representational opacity is primarily about transparency across representations. False fluency is primarily about learner judgement: polished output may be mistaken for correctness or understanding. Shallow repair is primarily about causal debugging: the artefact is corrected without diagnosis of why it failed. Feedback flooding is primarily about volume and actionability. Responsibility diffusion is primarily about ownership and accountability for decisions, verification, and authorship. Because these are diagnostic lenses on an interaction, multiple forms may be present simultaneously; the primary dimension indicates the main reason for assigning a category, not an exclusive causal mechanism.

Table 3 summarises the primary dimension, competence risk, and a corresponding noise-reducing response for each proposed form.

Table 3. Proposed forms of pedagogical noise and primary diagnostic dimensions

Proposed form	Primary diagnostic dimension	Description	Potential risk for competence	Noise-reducing response
Over-completion	Completeness	AI provides more of the solution than the learner's partial difficulty requires.	Sequencing and procedural design may be skipped.	Use hints, questions, or missing-step prompts before full answers.
Premature solutioning	Timing	AI introduces a solution before the intended interpretation or decomposition phase.	Constraint analysis and decomposition may be bypassed.	Require a learner restatement or plan before solution-oriented AI use.
Representational opacity	Transparency	AI converts among verbal descriptions, pseudocode, flowcharts, or code without exposing the mapping.	Relationships among representations may remain invisible.	Require mapping and justification across representations.
False fluency	Learner judgement	Polished and confident output is taken as evidence of correctness or understanding.	Verification and evaluative judgement may weaken.	Use traces, edge cases, and critique.
Shallow repair	Causal debugging	AI repairs an error without requiring diagnosis of the causal mechanism.	Debugging may become artefact repair rather than reasoning.	Require bug hypothesis, trace, and repair justification.
Feedback flooding	Volume/actionability	AI provides too much feedback, too many alternatives, or insufficient prioritisation.	Attention may become fragmented or passive.	Limit feedback to the next actionable step or a small number of alternatives.
Responsibility diffusion	Ownership/accountability	Responsibility for correctness, verification, or authorship becomes unclear.	Learner judgement and accountability may be delegated.	Require disclosure, verification, and explicit decision ownership.

4.3 *Scaffolding, substitution, and the conceptual model*

Scaffolding and substitution describe two ends of the pedagogical function of support. Scaffolding helps learners participate in work they cannot yet complete independently while preserving meaningful cognitive and epistemic participation (Reiser, 2004; Wood et al., 1976). Substitution, as used here, occurs when assistance performs the competence-forming reasoning on the learner’s behalf. The distinction is relational rather than tool-specific.

Four conditions are especially useful for diagnosing movement between scaffolding and substitution: timing, granularity, transparency, and agency. Timing concerns when AI en-

ters the task; granularity concerns how much work it performs; transparency concerns whether the reasoning behind transformations and repairs is inspectable; agency concerns whether the learner remains responsible for decisions, testing, and revision. These conditions are consistent with programming evidence showing scaffolding-offloading dynamics (Liu et al., 2026), performance-learning dissociation under unrestricted assistance (Bassner et al., 2026), and benefits of structured pedagogical integration (Nathaniel et al., 2026).

Table 4 turns these conditions into a task-level diagnostic model.

Table 4. Relational model for analysing pedagogical noise

Framework element	Diagnostic question	Lower-noise condition	Higher-noise condition
Learning goal	What competence is being formed?	AI support targets a reasoning step without replacing it.	AI output primarily targets completion rather than the intended competence.
Learner state	What has the learner already attempted?	AI enters after visible interpretation, decomposition, diagnosis, or testing.	AI enters before the learner-owned reasoning phase begins.
Timing	When does AI enter?	After an attempt or at a designed checkpoint.	Before the intended reasoning opportunity.
Granularity	How much work does AI perform?	Hint, question, trace, contrast, or bounded explanation.	Complete solution or repair when partial support would suffice.
Transparency	Can the learner inspect the transformation?	Reasoning, mapping, or causal explanation is visible and discussable.	Output is opaque or accepted as a black box.
Agency	Who decides and verifies?	Learner justifies, tests, and revises.	Learner accepts, copies, or delegates responsibility.
Assessment alignment	What is evaluated?	Process, explanation, verification, and transfer are visible.	Only final artefact correctness is evaluated.

4.4 Worked cases: applying the framework

The following cases are illustrative applications of the conceptual framework, not empirical findings. Unless stated otherwise, they assume that the GenAI output is technically correct so that the analysis isolates pedagogical alignment rather than hallucination. The cases were chosen to represent three common algorithmisation activities - procedural representation, debugging, and flowchart/decision representation - and to foreground timing, granularity, transparency, and agency. Their plausibility is supported by evidence that learners use GenAI for coding and debugging, that reflective debugging can improve learning conditions, and that unstructured assistance can promote offloading or difficulty evaluating AI output (Akçay et al., 2025; Fang et al., 2025; Kohen-Vacs et al., 2025; Liu et al., 2026).

Case 1: Natural-language task to pseudocode

A class is given a list of exam scores and a pass threshold of 50. Students must count the scores below 50 and report that remedial support is needed when at least 25% of the scores are below the pass threshold. A learner asks GenAI for pseudocode and receives a technically correct loop-based solution with a counter and the final 25% decision rule. The primary diagnostic dimension is timing: the same complete output can support comparison after an independent attempt or prematurely replace decomposition if requested first.

Table 5 applies the framework to Case 1.

Table 5. Worked case 1: same technically correct output as scaffolding or substitution

Element	Analysis
Intended competence	Interpret a verbal rule, identify inputs/outputs, decompose the counting procedure, and express it in pseudocode.
Learner-owned reasoning	Identify the list, pass threshold, counter, repeated comparison, proportion calculation, and the final rule: remedial support if <code>below_count / number_of_scores >= 0.25</code> .
Work performed by AI	Generate complete pseudocode containing the loop, conditional increment, proportion calculation, and final decision.
Scaffolding condition	The learner has already drafted a procedure and uses the AI version to compare ordering, variable choices, and the 25% decision rule.
Substitution condition	The learner requests the solution before attempting decomposition and adopts the AI pseudocode as the first representation of the task.
Evidence distinguishing the two	In scaffolding, the learner can explain the mapping from the verbal rule to each pseudocode step and revise their own draft. In substitution, the learner cannot justify why the proportion is computed or why the counter is updated inside the conditional.
Potential educational risk	Visible pseudocode quality improves while decomposition and representation skills remain uncertain.
Primary diagnostic dimension	Timing, with possible overlap with completeness and representational transparency.
Nearest evidence	Performance-learning dissociation and scaffolding/offloading dynamics provide the closest transferable support (Bassner et al., 2026; Liu et al., 2026).

Case 2: Debugging a loop boundary

A student writes a loop intended to process values from 1 to n, but the program omits the final value because the condition uses `< n` rather than `<= n`. Here the comparison uses two AI-response variants rather than claiming identical output. In the scaffolding variant, the AI asks the learner to trace the final iterations and identify which value is never reached before explaining the boundary issue. In the substitution variant, the AI immediately supplies the corrected condition. The primary diagnostic dimension is causal debugging, with granularity as the key design condition. Reflective debugging research provides direct school-level support for preserving diagnosis before repair (Fang et al., 2025).

Table 6 applies the framework to Case 2.

Table 6. Worked case 2: debugging as diagnosis or repair substitution

Element	Analysis
Intended competence	Trace loop execution, diagnose a boundary error, and test edge cases.
Learner-owned reasoning	Identify the missing iteration, connect it to the loop condition, and test $n = 1$, $n = 2$, and a typical case.
Work performed by AI	Variant A prompts a trace and then explains the boundary mechanism; Variant B immediately provides the corrected condition $\leq n$.
Scaffolding condition	The AI first elicits a trace or bug hypothesis, and the learner explains why the final iteration is omitted before accepting the repair.
Substitution condition	The AI supplies the corrected condition immediately and the learner pastes it without causal diagnosis.
Evidence distinguishing the two	The learner can reproduce the bug mechanism and test the repair independently in the scaffolding condition; only artefact correctness changes in the substitution condition.
Potential educational risk	The program is repaired while debugging competence remains unexercised.
Primary diagnostic dimension	Causal debugging, with overlap with granularity and learner judgement.
Nearest evidence	Reflective debugging in senior high school supports diagnosis-oriented use; difficulties correcting AI-generated code illustrate the importance of verification (Fang et al., 2025; Kohen-Vacs et al., 2025).

Case 3: Flowchart representation from a verbal rule

Students must design a decision process that classifies a number as negative, zero, positive even, or positive odd. A learner asks GenAI for a textual flowchart description and receives a technically correct sequence of decisions. The output is held constant; what varies is the learner state and required accountability. The primary diagnostic dimension is transparency: the question is whether the learner can explain why the decision order works and how the verbal conditions map to the flowchart branches.

Table 7 applies the framework to Case 3.

Table 7. Worked case 3: representation support or representational opacity

Element	Analysis
Intended competence	Order conditional decisions and map verbal rules into a flowchart-like representation.
Learner-owned reasoning	Decide which test comes first, why zero is handled separately, and why parity is tested only for positive values.
Work performed by AI	Generate a technically correct textual sequence for the decision structure and branch outcomes.
Scaffolding condition	The learner has produced an initial decision tree and uses the AI output to identify missing branches or compare ordering.
Substitution condition	The AI-generated sequence becomes the learner's first and only representation and is copied into a flowchart without explanation.
Evidence distinguishing the two	In scaffolding, the learner can map every branch to the original verbal rule and defend the ordering. In substitution, the learner cannot justify the representation beyond stating that the AI produced it.

Element	Analysis
Potential educational risk	A clean representation masks whether representational translation has been learned.
Primary diagnostic dimension	Transparency, with overlap with timing and ownership/accountability.
Nearest evidence	School-informatics work establishes the importance of representational and algorithmic reasoning; GenAI studies support the need for structured explanation and verification (Dagienė & Futschek, 2008; Zhao et al., 2025).

5. Discussion

5.1 Conceptual question and synthesis result

The conceptual question was what a pedagogical-noise lens permits educators and researchers to diagnose beyond neighbouring constructs. The synthesis suggests a bounded answer: it makes the unit of analysis a particular AI contribution in relation to a particular competence-forming step, learner state, and algorithmisation phase. This is more specific than identifying offloading, poor scaffolding, excessive trust, or performance-learning dissociation in general. The result is a proposed relational diagnostic framework rather than evidence that pedagogical noise is an independently measurable causal mechanism.

The seven forms make that diagnosis actionable by assigning a primary dimension to each recurring misalignment. Completeness and timing distinguish over-completion from premature solutioning; transparency distinguishes representational opacity; learner judgement distinguishes false fluency; causal debugging distinguishes shallow repair; volume/actionability distinguishes feedback flooding; and ownership/accountability distinguishes responsibility diffusion. Overlap is expected because a single interaction can be both premature and over-complete or both opaque and responsibility-diffusing.

5.2 Relation to the closest literature

The framework is closest to three recent programming strands. First, Liu et al. (2026) describe scaffolding and offloading loops, domain versus tool mastery, and verification problems in undergraduate programming. Pedagogical noise does not replace that process theory; it adds a task-level diagnostic vocabulary for identifying what work was displaced in an algorithmisation episode. Second, Bassner et al. (2026) provide direct evidence that improved AI-supported task performance can coexist with unchanged measured learning and that scaffolded, hint-first support differs from unrestricted answer giving. Pedagogical noise uses this established dissociation as background and proposes categories for analysing how such dissociation might arise in specific algorithmisation tasks. Third, Nathaniel et al. (2026) show that structured pedagogical integration can improve alignment and cognitive engagement in higher-education programming. This supports the importance of orchestration while remaining an indirect contextual source for upper-secondary claims.

Direct school evidence is narrower but important. Reflective GenAI debugging with senior high-school students has been associated with improved learning performance under a structured reflective strategy (Fang et al., 2025). Other high-school studies report benefits from structured human-computer collaborative programming or differences across collaboration designs (Hussein et al., 2026; Zhao et al., 2025). These findings justify treating design conditions as consequential, but they do not empirically validate the proposed taxonomy.

5.3 Alternative interpretations and overlap with neighbouring constructs

A plausible alternative interpretation is that pedagogical noise is simply scaffolding failure renamed for GenAI. That interpretation is partly correct in the sense that several noisy interactions are also failures of calibration. The proposed increment is not a new universal mechanism but a narrower diagnostic lens: it distinguishes whether the misalignment concerns completeness, timing, transparency, judgement, causal debugging, feedback actionability, or accountability, and ties that diagnosis to an algorithmisation phase. Whether this added granularity produces better research coding or instructional decisions remains an empirical question.

A second alternative is that the construct is reducible to cognitive offloading or reduced epistemic agency. Again, overlap is real. Offloading may be adaptive when AI removes routine syntax burden so that learners can focus on decomposition or evaluation; it becomes problematic only when the offloaded work is itself the learning target. Reduced epistemic agency may be a downstream consequence when learners cease to verify or justify. The relational definition is intended to keep these distinctions visible rather than to treat all AI assistance as harmful.

5.4 Limitations and boundary conditions

The framework has four principal limitations. First, it is conceptual and has not been validated as a coding scheme, measurement instrument, or predictor of learning outcomes. Second, the seven forms are proposed and non-exhaustive; category overlap is a feature of complex interactions, not evidence that the categories are statistically independent. Third, the evidence base mixes direct upper-secondary studies with transferred evidence from higher education, programming education, educational psychology, and learning sciences. Claims are therefore strongest as design hypotheses for upper-secondary algorithmisation rather than general conclusions about all education.

Fourth, the framework intentionally analyses technically correct or plausible AI support as well as incorrect support. Hallucination can compound pedagogical noise, but factual accuracy is analytically separate from pedagogical fit. The worked cases are illustrative and were not observed classroom episodes. They demonstrate application logic, not prevalence or causal effect.

5.5 *Direct upper-secondary evidence and transferred evidence*

Evidence is treated in layers. Fang et al. (2025), Hussein et al. (2026), Liu et al. (2025), Zhao et al. (2025), Akçay et al. (2025), and Sánchez Muñoz et al. (2025) provide direct or pre-university evidence relevant to school programming, pseudocode, computational thinking, or GenAI use. Bassner et al. (2026), Liu et al. (2026), Nathaniel et al. (2026), Kohen-Vacs et al. (2025), Jacobs et al. (2025), and López-Pernas et al. (2025) are programming or AI-supported learning studies outside the target context and are used for conceptual transfer, not as direct evidence of upper-secondary effects. Foundational scaffolding, feedback, offloading, automation, and epistemic-agency literature provides theoretical framing. This layered treatment limits the inference level while preserving useful comparison across adjacent fields.

5.6 *Bounded design and teacher implications*

The framework supports design propositions, not prescriptions with demonstrated universal effects. In upper-secondary algorithmisation, teachers may reduce potential pedagogical noise by delaying full solutions until learners have attempted interpretation or decomposition; requesting hints, traces, or contrastive examples rather than complete answers; requiring students to explain mappings across natural language, pseudocode, flowcharts, and code; and designing debugging as diagnosis before repair. These propositions are consistent with scaffolding theory and with recent structured GenAI studies, but their causal effect on long-term algorithmic competence requires classroom testing.

Verification and accountability are equally important. Learners can be required to test edge cases, justify corrections, document accepted and rejected AI suggestions, and retain responsibility for final decisions. Feedback volume can be constrained to the next actionable step. Support can also be faded as competence grows. These practices target the diagnostic dimensions in Table 3 rather than treating AI access itself as the intervention.

Table 8 summarises the design propositions and their evidential status.

Table 8. Noise-aware design propositions and evidence boundaries

Design proposition	Primary dimension targeted	Rationale	Evidence boundary
Delay full solutions until after learner interpretation/decomposition.	Timing; completeness	Preserves an opportunity for learner-owned problem formulation.	Conceptual proposition; not a universal causal rule.
Request hints, traces, questions, or contrastive examples before complete answers.	Completeness; transparency	Keeps more reasoning visible and learner-performed.	Consistent with scaffolding and structured-tutor evidence; context dependent.
Require explicit mapping across representations.	Transparency	Makes transformations among verbal rules, pseudocode, flowcharts, and code inspectable.	Design recommendation requiring classroom validation.

Design proposition	Primary dimension targeted	Rationale	Evidence boundary
Diagnose before repair.	Causal debugging	Separates bug understanding from artefact correction.	Supported indirectly and by school reflective-debugging evidence.
Verify technically correct-looking output with tests and explanations.	Learner judgement	Counters false fluency and automation bias.	Strong theoretical rationale; effect size unknown.
Limit feedback to a small number of actionable next steps.	Volume/actionability	Reduces feedback flooding and prioritisation burden.	Framework-derived application of feedback literature.
Document AI contribution and learner decisions.	Ownership/accountability	Keeps authorship, verification, and responsibility visible.	Design recommendation; does not itself prove improved learning.
Fade AI support as competence develops.	Completeness; ownership	Aligns assistance with increasing learner independence.	Scaffolding-based proposition requiring longitudinal validation.

5.7 Empirical validation priorities

Future research should test whether the proposed categories can be identified reliably in authentic learner-AI interactions and whether they predict later independent performance, transfer, debugging quality, or explanation quality. Process data such as prompt logs, screen recordings, traces, think-aloud protocols, and learner explanations are particularly useful because final artefacts alone may conceal whether reasoning was performed or outsourced (Jacobs et al., 2025; López-Pernas et al., 2025).

Intervention studies could compare unrestricted assistance with hint-first prompting, explanation gates, mandatory verification, representation-mapping requirements, or faded support. The proposition that AI role and task design may matter more than tool brand should be treated as a hypothesis requiring direct comparative testing, not as a finding of this paper. Prompt literacy should likewise be tested as a possible moderator rather than assumed to reduce pedagogical noise automatically.

Table 9 separates established background evidence from the paper’s proposed framework, design recommendations, and validation needs.

Table 9. Status of central claims

Claim	Status	Basis	Boundary
Immediate performance can diverge from durable learning.	Established background principle with programming-specific evidence.	Soderstrom & Bjork (2015); Bassner et al. (2026).	Not novel to this paper.

Claim	Status	Basis	Boundary
Pedagogical noise is a relational misalignment between AI contribution and learner-owned competence-forming work.	Proposed conceptual construct.	Integrative comparison with neighbouring constructs and algorithmisation tasks.	Requires empirical validation as a diagnostic category.
Seven forms can be distinguished by primary diagnostic dimensions.	Proposed taxonomy.	Conceptual synthesis and worked-case operationalisation.	Non-exhaustive; categories may overlap.
Structured debugging and dialogic/reflective support can be beneficial in programming-related learning.	Empirically supported background claim.	Emerging school and programming studies.	Effects are context- and design-dependent.
Delay full solution generation in early algorithmisation.	Design proposition.	Competence-preservation logic plus scaffolding/productive-struggle literature.	Not established as a universal causal rule.
AI role/task design may matter more than tool brand.	Future empirical hypothesis.	Suggested by variation across structured and unrestricted support designs.	Requires direct comparative testing.
Noise-aware design improves long-term algorithmic competence.	Future empirical hypothesis.	Plausible implication of the framework.	Requires delayed, transfer, and independent-performance measures.

6. Conclusion

Established literature shows that immediate task performance and durable learning can diverge and that the pedagogical design of GenAI-supported programming matters. Direct upper-secondary studies further suggest that structured debugging, collaboration, and feedback can support learning under particular conditions. These points are background evidence; they are not discoveries of the present article.

The contribution proposed here is narrower: pedagogical noise as an upper-secondary algorithmisation-specific relational diagnostic lens. It asks whether a particular AI contribution is aligned with the learner-owned reasoning that should remain active at a particular phase. The seven proposed forms organise that question by completeness, timing, transparency, learner judgement, causal debugging, volume/actionability, and ownership/accountability. Worked cases illustrate how technically correct output may scaffold or substitute depending on learner state and instructional conditions.

The framework therefore does not establish that pedagogical noise causes learning loss, nor that the taxonomy is exhaustive or empirically validated. It provides a bounded vocabulary for designing and studying GenAI-supported algorithmisation while keeping learner reasoning visible. Future work should validate the categories in classroom interaction data, test whether noise-reducing designs improve transfer and independent performance, and examine how learner expertise changes the boundary between scaffolding and substitution. The educational question is thus not only whether GenAI is used in algorithmisation, but

also how its contribution is orchestrated so that support does not inadvertently replace the reasoning the lesson is intended to develop.

Declarations

Ethics approval. Not applicable. This is a conceptual article and did not involve human participants, human data, or animal subjects.

Consent to participate. Not applicable.

Data and materials availability. No new empirical data were generated or analysed for this conceptual paper. Bibliographic search exports that informed the literature orientation are publicly available in Mendeley Data (Žáček, 2026; DOI 10.17632/d246s355tz.1). The dataset was originally assembled for a separate review project and contains Web of Science and Scopus bibliographic exports rather than participant-level data or full texts.

Code availability. Not applicable.

Author contributions. Martin Žáček: conceptualization, literature synthesis, writing - original draft, writing - review and editing.

Use of generative AI tools. Generative AI tools were used only for language proofreading, consistency checking, and checking the manuscript against journal requirements. They were not used to generate the conceptual contribution, select or interpret sources, or produce empirical data. The author reviewed and approved the final text and takes full responsibility for the manuscript.

References

- Akçay, A., Avan, Ç., & Karabulut Coşkun, B. (2025). You're not alone while programming! Challenges faced by high school students, generative AI solutions and issues. *Interactive Learning Environments*, 34(7), 4643–4661. <https://doi.org/10.1080/10494820.2025.2608191>
- Bassner, P., Lenk-Ostendorf, B., Beinstingel, R., Wasner, T., & Krusche, S. (2026). Less stress, better scores, same learning: The dissociation of performance and learning in AI-supported programming education. *Computers and Education: Artificial Intelligence*, 10, 100537. <https://doi.org/10.1016/j.caeai.2025.100537>
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association*.
- Brodnik, A., Csizmadia, A., Futschek, G., Kralj, L., Lonati, V., Micheuz, P., & Monga, M. (2021). Programming for all: Understanding the nature of programs. *arXiv*. <https://arxiv.org/abs/2111.04887>
- Dagienė, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In R. T. Mittermeir & M. M. Sysło (Eds.), *Informatics education - Supporting computational thinking (Lecture Notes in Computer Science, Vol. 5090, pp. 19–30)*. Springer. https://doi.org/10.1007/978-3-540-69924-8_2
- Damşa, C. I., Kirschner, P. A., Andriessen, J. E. B., Erkens, G., & Sins, P. H. M. (2010). Shared epistemic agency: An empirical study of an emergent construct. *Journal of the Learning Sciences*, 19(2), 143–186. <https://doi.org/10.1080/10508401003708381>

- Fang, J.-W., Chen, J., Weng, Q.-L., Tu, Y.-F., Hwang, G.-J., & Xia, Y.-C. (2025). Effects of a GenAI-based debugging approach integrating the reflective strategy on senior high school students' learning performance and computational thinking. *Educational Technology & Society*, 28(3), 66–81. [https://doi.org/10.30191/ETS.202507_28\(3\).SP06](https://doi.org/10.30191/ETS.202507_28(3).SP06)
- Federiakin, D., Molerov, D., Zlatkin-Troitschanskaia, O., & Maur, A. (2024). Prompt engineering as a new 21st century skill. *Frontiers in Education*, 9, 1366434. <https://doi.org/10.3389/educ.2024.1366434>
- Gong, X., Li, Z., & Qiao, A. (2025). Impact of generative AI dialogic feedback on different stages of programming problem solving. *Education and Information Technologies*, 30, 9689–9709. <https://doi.org/10.1007/s10639-024-13173-1>
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43. <https://doi.org/10.3102/0013189X12463051>
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81–112. <https://doi.org/10.3102/003465430298487>
- Hussein, M. H., Aldeeb, F. H. A., Sulaiman, N. L., & Elaiash, M. M. (2026). Comparing the impact of individual- and team-based ChatGPT collaboration on high school students' cognitive competencies in programming education. *Journal of Computer Assisted Learning*, 42, e70226. <https://doi.org/10.1002/jcal.70226>
- Izu, C., Mirolo, C., Settle, A., Mannila, L., & Stupurienė, G. (2017). Exploring Bebras tasks content and performance: A multinational study. *Informatics in Education*, 16(1), 39–59. <https://doi.org/10.15388/infedu.2017.03>
- Jacobs, S., Haas, J., & Kiesler, N. (2025). Student engagement with GenAI's tutoring feedback: A mixed methods study. In *Proceedings of the 25th Koli Calling International Conference on Computing Education Research (Koli Calling '25)* (pp. 1–12). ACM. <https://doi.org/10.1145/3769994.3770034>
- Kapur, M. (2008). Productive failure. *Cognition and Instruction*, 26(3), 379–424. <https://doi.org/10.1080/07370000802212669>
- Kapur, M. (2016). Examining productive failure, productive success, unproductive failure, and unproductive success in learning. *Educational Psychologist*, 51(2), 289–299. <https://doi.org/10.1080/00461520.2016.1155457>
- Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., Weller, J., Kuhn, J., & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103, 102274. <https://doi.org/10.1016/j.lindif.2023.102274>
- Kohen-Vacs, D., Usher, M., & Jansen, M. (2025). Integrating generative AI into programming education: Student perceptions and the challenge of correcting AI errors. *International Journal of Artificial Intelligence in Education*, 35, 3166–3184. <https://doi.org/10.1007/s40593-025-00496-4>
- Liu, D., Fan, G., & Pan, L. (2026). Tool, tutor, or crutch?: A grounded theory of cognitive scaffolding and offloading in AI-assisted programming education. *International Journal of STEM Education*, 13, 10. <https://doi.org/10.1186/s40594-025-00592-w>
- Liu, Z., Xing, W., Jiao, X., Li, C., & Zhu, W. (2025). What are the differences between student and ChatGPT-generated pseudocode? Detecting AI-generated pseudocode in high school programming using explainable machine learning. *Education and Information Technologies*, 30, 14853–14892. <https://doi.org/10.1007/s10639-025-13385-z>

- Long, D., & Magerko, B. (2020). What is AI literacy? Competencies and design considerations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1–16). ACM. <https://doi.org/10.1145/3313831.3376727>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61. <https://doi.org/10.1016/j.chb.2014.09.012>
- López-Pernas, S., Misiejuk, K., Oliveira, E., & Saqr, M. (2025). The dynamics of the self-regulation process in student-AI interactions: The case of problem-solving in programming education. In *Proceedings of the 25th Koli Calling International Conference on Computing Education Research* (pp. 1–12). ACM. <https://doi.org/10.1145/3769994.3770043>
- Mannila, L., Dagienė, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., & Settle, A. (2014). Computational thinking in K-9 education. In *Proceedings of the Working Group Reports of the 2014 Innovation & Technology in Computer Science Education Conference* (pp. 1–29). ACM. <https://doi.org/10.1145/2713609.2713610>
- Markauskaite, L., & Goodyear, P. (2017). *Epistemic fluency and professional education: Innovation, knowledgeable action and actionable knowledge*. Springer.
- Narciss, S. (2008). Feedback strategies for interactive learning tasks. In J. M. Spector, M. D. Merrill, J. van Merriënboer, & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology* (3rd ed., pp. 125–144). Lawrence Erlbaum.
- Nathaniel, J., Oyelere, S. S., Suhonen, J., & Tedre, M. (2026). An experimental study of structured generative AI integration to mitigate pedagogical, cognitive, and ethical barriers in programming education. *Frontiers in Computer Science*, 8, 1789829. <https://doi.org/10.3389/fcomp.2026.1789829>
- Ng, D. T. K., Leung, J. K. L., Chu, S. K. W., & Qiao, M. S. (2021). Conceptualising AI literacy: An exploratory review. *Computers and Education: Artificial Intelligence*, 2, 100041. <https://doi.org/10.1016/j.caeai.2021.100041>
- Parasuraman, R., & Manzey, D. H. (2010). Complacency and bias in human use of automation: An attentional integration. *Human Factors*, 52(3), 381–410. <https://doi.org/10.1177/0018720810376055>
- Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39(2), 230–253. <https://doi.org/10.1518/001872097778543886>
- Pea, R. D. (2004). The social and technological dimensions of scaffolding and related theoretical concepts for learning, education, and human activity. *Journal of the Learning Sciences*, 13(3), 423–451. https://doi.org/10.1207/s15327809jls1303_6
- Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *ACM SIGCSE Bulletin*, 39(4), 204–223. <https://doi.org/10.1145/1345375.1345441>
- Redecker, C. (2017). *European framework for the digital competence of educators: DigCompEdu*. Publications Office of the European Union. <https://doi.org/10.2760/159770>
- Reiser, B. J. (2004). Scaffolding complex learning: The mechanisms of structuring and problematizing student work. *Journal of the Learning Sciences*, 13(3), 273–304. https://doi.org/10.1207/s15327809jls1303_2
- Risko, E. F., & Gilbert, S. J. (2016). Cognitive offloading. *Trends in Cognitive Sciences*, 20(9), 676–688. <https://doi.org/10.1016/j.tics.2016.07.002>
- Sánchez Muñoz, J. A., Flores-Eraña, G., Silva-Campos, J. M., Chavira-Quintero, R., & Olais-Govea, J. M. (2025). GenAI as a cognitive mediator: A critical-constructivist inquiry into computational thinking in pre-university education. *Frontiers in Education*, 10, 1597249. <https://doi.org/10.3389/educ.2025.1597249>

- Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22, 469–495. <https://doi.org/10.1007/s10639-016-9482-0>
- Sentance, S., Barendsen, E., & Schulte, C. (Eds.). (2018). *Computer science education: Perspectives on teaching and learning in school*. Bloomsbury Academic.
- Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, 78(1), 153–189. <https://doi.org/10.3102/0034654307313795>
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142–158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Soderstrom, N. C., & Bjork, R. A. (2015). Learning versus performance: An integrative review. *Perspectives on Psychological Science*, 10(2), 176–199. <https://doi.org/10.1177/1745691615569000>
- UNESCO. (2024a). *AI competency framework for students*. UNESCO.
- UNESCO. (2024b). *AI competency framework for teachers*. UNESCO.
- Vuorikari, R., Kluzer, S., & Punie, Y. (2022). *DigComp 2.2: The digital competence framework for citizens - with new examples of knowledge, skills and attitudes*. Publications Office of the European Union. <https://doi.org/10.2760/115376>
- Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology*, 25, 127–147. <https://doi.org/10.1007/s10956-015-9581-5>
- Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2), 89–100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>
- Zhao, G., Yang, L., Hu, B., & Wang, J. (2025). A generative artificial intelligence (AI)-based human-computer collaborative programming learning method to improve computational thinking, learning attitudes, and learning achievement. *Journal of Educational Computing Research*, 63(5), 1059–1087. <https://doi.org/10.1177/07356331251336154>
- Žáček, M. (2026). Search data for review on generative AI in upper-secondary programming education (Version 1) [Data set]. Mendeley Data. <https://doi.org/10.17632/d246s355tz.1>
- Žáček, M., & Konečná, P. (2024). Algorithmization with AI. In R. Maršálek (Ed.), *DIDSCI+ 2024: Proceedings of extended abstracts* (pp. 84–85). University of Ostrava. <https://doi.org/10.15452/DIDSCI.2024>