Use of Large Language Models in the Teaching of Computational Thinking: A Systematic Mapping Study

João Antônio Misson Milhorim[1], **Diego Gomes de Santana**[1], **Diego Fernandes Lemos**[1], **Lina Garcés**[1]

[1] Instituto de Ciências Matemáticas e de Computação (ICMC)
Universidade de São Paulo (USP)
São Carlos - SP, Brasil

{joao.misson, diegogsantana94,diego.lemos, linagarces}@usp.br

Abstract. *This study presents the results of a systematic mapping of the use of Large Language Models (LLMs) in Computational Thinking (CT) education. The study was designed using the Goal–Question–Metric approach and well-known secondary study guidelines. A final set of 13 primary studies was selected from 10,022 records retrieved from Scopus, IEEE Xplore, and the ACM Digital Library. This study provides an overview of technological configurations, pedagogical practices, learning outcomes, evaluation strategies, and educational contexts in which LLMs have been used in CT education. Results show that LLMs are most commonly utilized as evaluators, feedback generators, and intelligent tutors, with emerging applications in pair programming and scaffolding. Pedagogical approaches favor structured support and project- or design-based learning, with teachers acting primarily as facilitators and intervention designers and students as the main protagonists of their education. LLMs' adoption is growing in higher education, while adoption at primary and secondary levels remains underexplored. Most interventions take place in synchronous, face-to-face formats and display substantial variation in evaluation rigor. Key gaps include limited use of validated knowledge assessment tools and insufficient attention to teacher preparation for hybrid human–AI instruction. These findings inform future research, design, and policy for AI-enhanced computing education.*

1. Introduction

Over recent decades, Computational Thinking (CT) has become established as a core competence in Computer Science (CS) education and related fields. More recently, the popularization of Large Language Models (LLMs) and generative AI systems has introduced new possibilities for teaching programming and its foundational concepts [Raihan et al. 2024]. As these models increasingly enter educational contexts, understanding their pedagogical roles and implications has become critical for institutions, instructors, and policymakers [Miao and Holmes 2023].

Advances in LLMs have expanded instructional possibilities for programming and related competencies, raising questions about how such systems can be meaningfully integrated into CT education. Although recent studies have explored these issues, the evidence remains dispersed and heterogeneous, varying widely in methodological rigor, application contexts, and learning objectives [Elnaffar et al. 2025, Raihan et al. 2024].

Despite the rapid growth in research on LLMs in education, several gaps remain [Zhang et al. 2024]. Much of the existing work consists of local or small-scale interventions, often employing heterogeneous designs and ad hoc assessment instruments, which complicates comparison across studies and limits cumulative evidence building [Zhang et al. 2024, Elnaffar et al. 2025, Raihan et al. 2024]. Moreover, much of the literature focuses on programming or computing education broadly, without explicitly addressing CT as a construct with distinct dimensions such as abstraction, decomposition, algorithmic thinking, and debugging [Grover and Pea 2013, Liao et al. 2024]. To date, there is no consolidated understanding of how LLMs have been used to support CT teaching and learning that integrates technological configurations, pedagogical practices, educational contexts, and reported learning outcomes. This fragmentation limits the ability to inform evidence-based instructional design and educational policy in AI-enhanced CT education.

In this context, this article presents a systematic mapping of scientific studies on the use of LLMs in CT education, synthesizing evidence across four principal axes: (i) technologies and solution approaches used to integrate LLMs into CT education, including the functional roles assumed by LLMs (RQ1); (ii) pedagogical characteristics of LLM-based CT approaches, including educational levels, disciplines, formats, and contexts (RQ2); (iii) pedagogical practices adopted to integrate LLMs into CT instruction (RQ3); and (iv) learning outcomes analyzed, along with the evaluation strategies and indicators employed (RQ4). The study follows established guidelines for systematic mapping studies [Petersen et al. 2008] and adopts the Goal–Question–Metric (GQM) approach [Basili and Weiss 1984] to align research objectives, questions, and analysis metrics.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 describes the research design, including the search strategy, inclusion and exclusion criteria, and data extraction procedures. Section 4 synthesizes the findings of the selected primary studies according to the four research questions. Section 5 discusses the main results and research gaps. Section 6 outlines threats to validity, and Section 7 concludes with directions for future research.

2. Related Work

Several systematic reviews have investigated the use of AI in education. Zhan et al. [Zhan et al. 2022] examined gamification in programming, and Benavides-Varela et al. [Benavides-Varela et al. 2020] investigated digital interventions for mathematics learning. These studies established important methodological precedents but do not specifically address Large Language Models in Computational Thinking education.

Specifically regarding LLMs, Heung and Chiu [Heung and Chiu 2025] revealed medium to large effect sizes on student engagement in 17 empirical studies. Li et al. [Li et al. 2025a] identified significant effects of ChatGPT highlighting the need for teacher support. Al Husaeni et al. [Husaeni et al. 2025] conducted bibliometric analysis (2007–2024) identifying sharp growth since 2019 (CAGR 22.8%) and revealing that *machine learning*, although frequent, remains a basic theme without deep pedagogical development. These reviews do not offer systematic analysis of how LLMs are specifically integrated into CT teaching.

Regarding Computational Thinking as a construct with its own dimensions, abstraction, decomposition, pattern recognition, algorithmic thinking, and debugging [Wing 2006, Brennan and Resnick 2012], the literature remains scattered. Al Husaeni et al. [Husaeni et al. 2025] identified that the connection between AI and CT lacks consolidated pedagogical *frameworks*. Although *frameworks* exist to assess CT, such as the *Computational Thinking Scale* [Korkmaz et al. 2017], little is known about adaptations in contexts with LLMs. Emerging themes such as *language models* remain underexplored [Husaeni et al. 2025], indicating a critical methodological *gap*.

Unlike reviews that examine general impacts of LLMs in programming or computing education [Heung and Chiu 2025, Husaeni et al. 2025], often emphasizing performance or engagement outcomes, this systematic mapping focuses explicitly on CT as a multidimensional construct and examines how LLMs are pedagogically integrated, including their roles and teacher mediation. This multidimensional approach, aligned with the Goal–Question–Metric (GQM) *framework*, enables identification of gaps and future directions for research, instructional design, and educational policies in AI-enhanced CT.

3. Methods

This study aims to offer an overview of the use of LLMs in CT education and characterize those studies regarding the following research questions (RQs):

- **RQ1** - What technologies and solution approaches are used to integrate LLMs into CT education, and what functional roles do LLMs assume in these implementations?
- **RQ2** - What are the pedagogical characteristics of LLM-based approaches (educational levels, disciplines, formats, and contexts) for teaching CT?
- **RQ3** - Which pedagogical practices are adopted to integrate LLMs into CT instruction?
- **RQ4** - Which learning outcomes are analyzed and which evaluation strategies and indicators are employed when LLMs are used for CT education?

To ensure a systematic alignment between the research objectives and the data analysis procedures, the study design employed the GQM methodology [Basili and Weiss 1984]. The conceptual structure of this approach, as depicted in Figure 1, illustrates the hierarchical relationships among goals, their associated analytical questions, and the corresponding evaluation metrics.

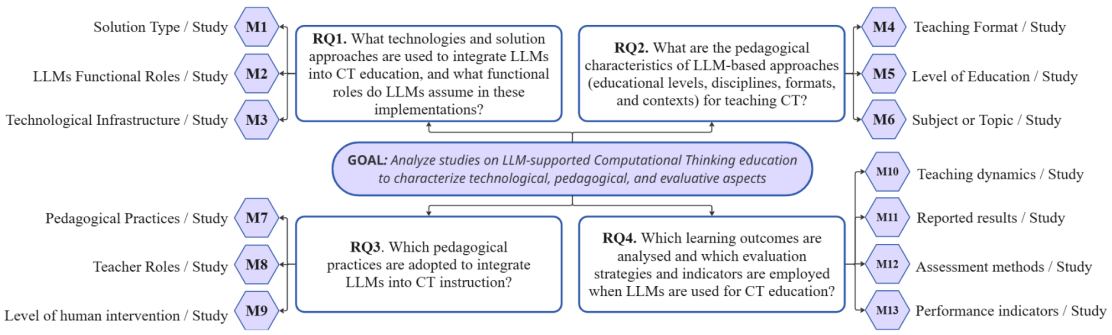

Figure 1. Study Design following GQM.

For conducting this secondary study, the research protocol followed the guidelines proposed by [Petersen et al. 2008]. The mapping process was organized into five distinct stages, as depicted in Figure 2. Each stage and its results are explained as follows.

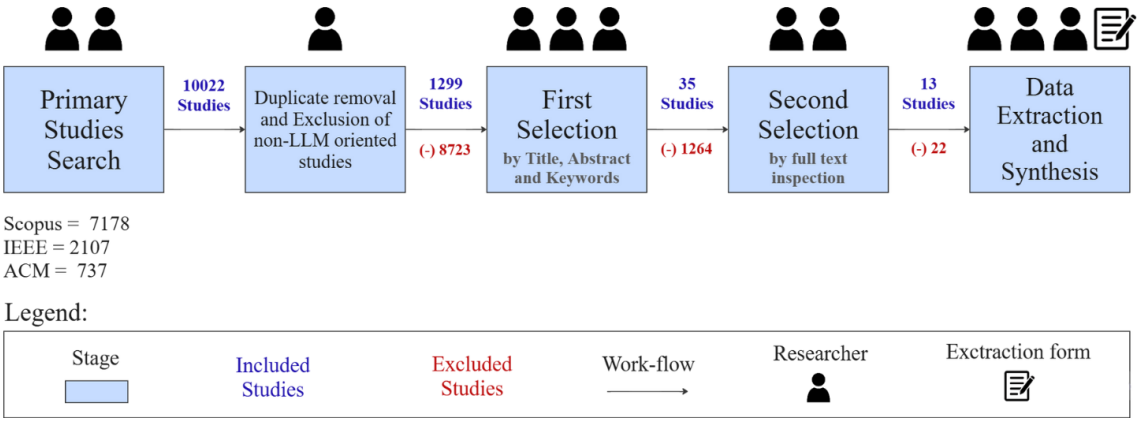

Figure 2. Literature review process

Stage 1 - Primary studies search. The search process was executed on October 7, 2025. Three digital libraries were selected due to their relevance in the fields of computer science and education, as suggested by [Kitchenham et al. 2015], namely, Scopus, IEEE Xplore, and ACM Digital Library. The search string is presented in Figure 3. The string was adapted to the syntax of each digital library engine. No temporal restrictions were applied during the search process; therefore, all studies indexed in the selected databases at the time of the search, regardless of publication year, were retrieved. As a result of this stage, the initial search yielded 10,022 studies.

```
TITLE-ABS-KEY(("artificial intelligence" OR "AI" OR "ML"
OR "Machine Learning" OR "LLM" OR "Language Model" OR
"Generative AI" OR "Gen AI" OR "GenAI" OR "Natural Language
Processing" OR "Deep Learning" OR "NLP") AND (teaching OR
education OR instruction) AND ("computational thinking" OR
"computing thinking" OR "programming"))
```

Figure 3. Search string.

Stage 2 - Duplicate Removal and Exclusion of non LLM-oriented Studies. Duplicate studies were removed, reducing the set to 8,512 unique papers. An additional filter was applied to refine the scope to Generative AI and LLMs exclusively; a keyword-based inclusion filter was applied to the *titles* of the remaining papers. Studies were maintained only if their title contained at least one of the specific LLM-related terms as listed in Figure 4. This step allowed us to select 1,299 studies, which were then considered for the next phase of detailed screening.

```
"LLM", "ChatGPT", "Gemini", "Copilot", "Codex", "DeepSeek",
"Grok", "Gen AI", "GenAI", "Generative AI", "Language
Model", "Large Language Model"
```

Figure 4. List of keywords applied to filter LLM-related studies.

Stage 3 - First Selection. This selection was performed by three researchers who read the titles, abstracts, and keywords of the 1,299 studies. To be selected for the next step, studies had to meet the Inclusion Criteria (IC) and not meet any Exclusion Criteria (EC), as detailed as follows:

- **IC1:** Studies that propose or use LLMs to teach Computational Thinking.
- **EC1:** Studies in progress or published as short papers.
- **EC2:** Studies not available for full access.
- **EC3:** Studies not peer-reviewed.
- **EC4:** Secondary or tertiary studies.
- **EC5:** Studies not related to Computational Thinking education with LLMs.
- **EC6:** Studies using LLMs for education in domains other than Computational Thinking.

Disagreements regarding the inclusion/exclusion of studies were discussed among the researchers until a consensus was reached, ensuring selection reliability. The result of this screening was 35 studies included for the subsequent stage.

Stage 4 - Second Selection. The full texts of the 35 studies were reviewed, and the Inclusion and Exclusion criteria were reapplied. Two researchers conducted this activity. A third researcher intervened when disagreements persisted between the two primary reviewers. After this detailed assessment of the full content, 22 studies were excluded because they failed to meet the quality or scope criteria upon detailed review, leaving 13 included as the final set of primary studies for this mapping.

Stage 5 - Data Extraction and Synthesis. For data extraction, a structured extraction form was proposed using *Google Forms*[1], and the data collected was managed via *Google Sheets*[2]. The extraction was performed by two reviewers. A third researcher, more experienced, participated in resolving disagreements or questions regarding data interpretation. The extraction schema was primarily aligned with the metrics defined in the GQM planning, as shown in Figure 1. However, complementary metrics were incorporated to answer the RQs comprehensively. Data was collected across the four main dimensions detailed in Table 1.

Data analysis adopted a mixed-methods approach to address the four research questions [Felizardo et al. 2017]. Quantitative descriptive analysis was used to examine technologies, solution approaches, and functional roles of LLMs (RQ1), as well as pedagogical characteristics of LLM-based CT approaches (RQ2). Thematic and content analyses were applied to identify pedagogical practices (RQ3) and the learning outcomes, evaluation strategies, and indicators reported in the studies (RQ4).

4. Results

This section presents the synthesis of the data extracted from these studies, organized according to the defined RQs. The selected primary studies are referenced in this section using identifiers S01 through S13, as listed in Table 2.

[1]https://docs.google.com/document/d/e/2PACX-1vRsZ3nlE19EWf-eb72wDI7M7N2fnTrhCm-xBE2
pub

[2]https://docs.google.com/spreadsheets/d/e/2PACX-1vTLeLklbtAHMJhK98if98B4WswxNRpaPvt
pubhtml?gid=1192766039&single=true

Table 1. Data extraction dimensions, extracted information, and related RQs

Dimension	Description & Extracted Data	Related RQs
Study Identification	General study metadata, including ID, title, authors, publication year, venue, country, and research design. (Collected for reporting; not tied to an RQ.)	—
Educational Context	Educational level, discipline, delivery modality, learning environment, and contextual characteristics of the CT instruction setting.	RQ2
Pedagogical Aspects	Teaching methodologies, instructional strategies, teacher roles, level of human intervention, and integration approaches used in CT instruction.	RQ2, RQ3
Technology	LLM infrastructures, models/tools employed, system functionalities (e.g., tutor, evaluator, collaborator), prompting strategies, and technological configurations.	RQ1
Metrics and Impacts	Data sources, sample characteristics, instruments, evaluation methods, analyzed learning outcomes, indicators, and impacts on performance, psychological factors, and teaching dynamics.	RQ4

4.1. RQ1: LLM-based technologies, solution approaches, and functional roles in CT education

This research question investigates the technological landscape and the modalities through which LLMs are integrated into CT education. We analyzed the types of solutions proposed and the underlying technological infrastructure employed.

Solutions Types. The analysis identifies four approaches to integrating LLMs into CT education. (i) *Frameworks and Pedagogical Models* (30.8%, or 4/13, namely S04, S07, S08, and S10) propose theoretical structures, conceptual models, or taxonomies that guide instructional design for AI integration, emphasizing how teaching and learning processes should be conceptualized rather than prescribing specific tools or interventions; (ii) *Systems and Technological Solutions* (23.1%, or 3/13, namely S02, S09, and S12) focus on the development of software artifacts or intelligent tutoring systems that operationalize LLM support in CT-related tasks; (iii) *Interventions and Educational Practices* (23.1%, or 3/13, namely S03, S11, and S13) implement LLMs through courses, learning scenarios, or classroom activities, translating AI capabilities into concrete instructional strategies; and (iv) *Exploratory and Analytical Studies* (23.1%, or 3/13, namely S01, S05, and S06) examine learner perceptions, interaction data, or behavioral patterns to understand the effects of LLM use rather than proposing new instructional solutions.

Across these solution types, the studies reveal a balanced distribution between conceptual, technical, and empirical contributions. This pattern suggests that the field is simultaneously concerned with defining pedagogical rationales, building technological artifacts, and empirically examining their educational effects, indicating an exploratory but methodologically diverse research landscape.

Technological Infrastructure. The technological landscape spans from widely accessible commercial platforms to specialized, custom-built systems. The most frequently used infrastructure corresponds to *Standard Chatbot Ecosystems* (30.8%, or 4/13, namely S03, S07, S11, and S13), reflecting low adoption barriers and immediate usability in classroom settings. *Integrated Development Assistants* (23.1%, or 3/13, namely S01, S04, and S08), such as GitHub Copilot, embed LLM capabilities directly within programming workflows. *Custom Systems and API-based Wrappers* (23.1%, or 3/13, namely S01, S02, and S10) enable greater control over AI behavior, interaction design, and pedagogical constraints. Finally, *Open-Source and Regional Alternatives* (23.1%, or 3/13, namely

Table 2. General characterization of the primary studies

ID	Ref.	Year	Study Purpose	Contribution Type	Research Design	Educational Level	Technology Category	Key Models/Tools	LLM Functional Roles	Teacher Roles	Instructional Format	Pedagogical Practice Category
S01	[Zönnchen et al. 2025]	2025	Investigate generative AI as a virtual tutor in programming education.	Exploratory and analytical study	Exploratory case study and mixed methods.	Higher Education (Undergraduate)	Custom Systems / Wrappers	CS50 Duck, ChatGPT, Github Copilot	Tutor / ITS; Evaluation / Feedback	Technology integrator	Blended (Hybrid)	LLM as Primary Tutor and Structured Problem-Solving Agent
S02	[Liao et al. 2024]	2024	Propose and evaluate IPSSC for CT problem solving using LLMs.	Technological solution	Quasi-experiment with pretest–posttest and group comparison.	Higher Education (Undergraduate)	Custom Systems / Wrappers	GPT-3, GPT-3.5	Scaffolding; Evaluation / Feedback; Q&A Support	Not explicitly specified	Synchronous face-to-face	Scaffolding-Centered Use of LLMs
S03	[Hong et al. 2024]	2024	Develop CT skills using AI chatbots integrated with Design Thinking.	Educational intervention	Quasi-experiment comparing experimental vs. control groups.	K-12 (High School)	Standard Chatbot Ecosystem	ChatGPT	Tutor / ITS; Evaluation / Feedback	Not explicitly specified	Synchronous face-to-face	Design Thinking and Creative Problem Solving
S04	[Nathaniel et al. 2025]	2025	Examine AI integration in programming courses.	Pedagogical framework	Sequential mixed methods with one-group pretest–posttest.	Higher Education (Undergraduate)	Integrated Assistants	ChatGPT, GitHub Copilot	Code Generation / Pair Programming; Scaffolding; Evaluation / Feedback	Facilitator and designer of the pedagogical intervention	Blended (Hybrid)	Scaffolding-Centered Use of LLMs; Project- and Experience-Based Learning
S05	[Dawson et al. 2025]	2025	Explore cognitive dissonance when using LLMs in programming learning.	Exploratory and analytical study	Qualitative observational study.	Higher Education (Undergraduate)	Open Source and Alternatives	Local or Self-built LLMs, BetterGPT	Tutor / ITS; Code Generation / Pair Programming; Evaluation / Feedback	Not explicitly specified	Not specified	Exploratory and Recommendation-Oriented Studies
S06	[Li et al. 2025b]	2025	Analyze CT process differences across two AI-supported conditions.	Exploratory and analytical study	Experimental comparison between two AI intervention types.	Higher Education (Undergraduate)	Custom Systems / Wrappers	No Code GPT, GPT-4o	Tutor / ITS; Code Generation / Pair Programming; Evaluation / Feedback	Technical support / minimal pedagogical presence	Synchronous face-to-face	Human–AI Collaborative Programming
S07	[Sanchez et al. 2025]	2025	Integrate ChatGPT into CT classes through Design Thinking.	Pedagogical framework	Action-research with iterative refinement cycles.	K-12 (High School)	Standard Chatbot Ecosystem	ChatGPT	Tutor / ITS; Scaffolding; Evaluation / Feedback	Facilitator and designer of the pedagogical intervention	Synchronous face-to-face	Design Thinking and Creative Problem Solving; Project- and Experience-Based Learning
S08	[Lei et al. 2025]	2025	Assess AIGC-CDIO model effects on motivation and CDIO competences.	Pedagogical framework	Quasi-experiment with one experimental and two control groups.	Higher Education (Undergraduate)	Integrated Assistants	ChatGPT, GitHub Copilot	Code Generation / Pair Programming; Scaffolding; Evaluation / Feedback	Facilitator and technology integrator	Synchronous face-to-face	Scaffolding-Centered Use of LLMs; Project- and Experience-Based Learning
S09	[Zhao et al. 2025]	2025	Develop CER-based LLM-supported system for scientific reasoning and CT.	Technological solution	Quasi-experiment with three comparison groups.	K-12 (Elementary)	Open Source and Alternatives	ERNIE Bot (Baidu)	Tutor / ITS; Q&A Support	Designer of the intervention; teacher role minimized during instruction	Synchronous face-to-face	Scaffolding-Centered Use of LLMs; LLM as Primary Tutor and Structured Problem-Solving Agent
S10	[Yan et al. 2025]	2025	Enhance CT and self-efficacy using speech-based AI programming support.	Pedagogical framework	Quasi-experimental pretest–posttest with control group.	K-12 (Elementary)	Custom Systems / Wrappers	SpeechGPT (GPT-4-0314)	Tutor / ITS; Code Generation / Pair Programming; Scaffolding	Facilitator and technology integrator	Synchronous face-to-face	Scaffolding-Centered Use of LLMs; Human–AI Collaborative Programming
S11	[Ouaazki et al. 2023]	2023	Investigate effectiveness of ChatGPT in CT learning.	Educational intervention	Exploratory case study with mixed data analysis.	Higher Education (Graduate)	Standard Chatbot Ecosystem	ChatGPT	Code Generation / Pair Programming; Evaluation / Feedback; Q&A Support	Technical support / minimal pedagogical presence	Blended (Hybrid)	Learner Autonomy with Computational Infrastructure
S12	[Li et al. 2024]	2024	Use LLM-generated diagrams to improve code understanding and CT.	Technological solution	Technical experimental evaluation.	Higher Education (Undergraduate)	Open Source and Alternatives	CodeLlama, GPT-4	Scaffolding; Evaluation / Feedback; Data Augmentation	Teacher role strongly reduced / AI-centered learning	Not specified	LLM as Primary Tutor and Structured Problem-Solving Agent
S13	[Yunianto et al. 2024]	2024	Examine ChatGPT support in solving CT tasks involving math explanations.	Educational intervention	Educational design research with qualitative triangulation.	Higher Education (Graduate)	Standard Chatbot Ecosystem	ChatGPT, ChatGPT Pro	Tutor / ITS; Code Generation / Pair Programming; Evaluation / Feedback	Designer of the pedagogical intervention	Asynchronous remote	Project- and Experience-Based Learning

S05, S09, and S12) address localization needs and linguistic or cultural alignment.

Although standard chatbot ecosystems dominate as entry points due to their accessibility, the presence of custom and open-source systems indicates a growing research interest in adaptable infrastructures. This trend suggests an early shift from general-purpose tools toward more research-driven and pedagogically configurable LLM environments.

Functional Roles Performed by LLMs. The analysis identifies six functional roles assumed by LLMs in CT education. The most prevalent role is *Automatic Evaluation and Feedback Generation* (84.6%, or 11/13, namely S01, S02, S03, S04, S05, S06, S07, S08, S11, S12, and S13), in which LLMs assess learner outputs and provide formative or corrective feedback. This is followed by the *Intelligent Tutoring System (ITS)* role (61.5%, or 8/13, namely S01, S03, S05, S06, S07, S09, S10, and S13), where LLMs deliver personalized guidance through adaptive dialogue. *Code Generation and Pair Programming* appears in 53.8% of the studies (7/13, namely S04, S05, S06, S08, S10, S11, and S13), supporting learners through example generation, code completion, or collaborative problem solving. Additional roles include *Scaffolding Tools* (46.2%, or 6/13, namely S02, S04, S07, S08, S10, and S12), *Conversational Question-and-Answer (Q&A) Agents* (23.1%, or 3/13, namely S02, S09, and S11), and *Data Augmentation* (7.7%, or 1/13, namely S12).

The distribution of functional roles reveals a strong emphasis on evaluative and tutoring capacities, reflecting a priority on scalable and individualized support in CT education. While code-generation and scaffolding roles indicate alignment with professional and instructional practices, the limited presence of generative or design-oriented roles suggests an underexplored research space for more autonomous and adaptive CT learning environments.

4.2. RQ2: Pedagogical characteristics of LLM-based approaches

This research question examines the pedagogical characteristics of the approaches described in the primary studies, with the aim of providing an overview of the educational contexts in which LLMs have been adopted to support the teaching and learning of CT.

Education Level. The analysis identifies the following educational contexts in which LLMs are applied to support CT development. (i) *Undergraduate Education* represents the predominant setting, accounting for 53.8% of studies (7/13, namely S01, S02, S04, S05, S06, S08, and S12), where LLM integration is explored within undergraduate courses, largely situated in STEM-related programs; (ii) *Graduate Education* constitutes a smaller proportion of the literature (15.4%, or 2/13, namely S11 and S13), focusing on graduate-level learning environments, also primarily within higher education contexts. Overall, studies in Higher Education dominate the field, with limited representation in areas outside STEM, such as business or the humanities.

In contrast, earlier stages of schooling remain relatively underexplored. (iii) *Secondary Education* is addressed by only a small subset of studies (namely S03 and S07), while (iv) *Primary Education* is similarly sparsely represented (namely S09 and S10), highlighting a notable gap in the literature regarding the use of LLMs to support CT development in K–12 contexts.

Instructional Formats. Regarding modes of instruction, the analysis identifies

that (i) *Synchronous Face-to-Face Instruction* is the most prevalent format (53.8%, or 7/13, namely S02, S03, S06, S07, S08, S09, and S10), indicating a preference for controlled classroom environments when examining the integration of LLMs into CT education; (ii) *Blended Learning* appears less frequently (23.1%, or 3/13, namely S01, S04, and S11), reflecting more flexible instructional arrangements that combine in-person and online components; (iii) *Asynchronous Remote Instruction* is rarely adopted (7.7%, or 1/13, namely S13), suggesting limited exploration of fully online, self-paced instructional contexts; and (iv) *Unreported Instructional Modality* accounts for a small subset of studies (15.4%, or 2/13, namely S05 and S12), in which the mode of instruction is not explicitly specified.

4.3. RQ3: Pedagogical practices and teacher roles in LLM-supported CT education

This research question examines the pedagogical practices used to integrate LLMs into CT instruction and the instructional roles teachers assume in these AI-supported learning environments.

Pedagogical Practices. Seven recurring pedagogical approaches for integrating LLMs into CT education were identified. (i) *Scaffolding-Centered Use of LLMs* (38.5%, or 5/13, namely S02, S04, S08, S09, and S10) employs structured, teacher- or system-guided workflows supporting decomposition, reasoning, and stepwise problem solving; (ii) *Project- and Experience-Based Learning* (23.1%, or 3/13, namely S04, S08, and S13, with partial inclusion of S07) integrates LLMs into authentic project-based contexts; (iii) *Design Thinking and Creative Problem Solving* (15.4%, or 2/13, namely S03 and S07) embeds LLMs within Design Thinking cycles; (iv) *Human–AI Collaborative Programming* (15.4%, or 2/13, namely S06 and S10) positions LLMs as co-programming partners; (v) *LLM as Primary Tutor and Structured Problem-Solving Agent* (23.1%, or 3/13, namely S01, S09, and S12) assigns LLMs a central instructional role; (vi) *Learner Autonomy with Computational Infrastructure* (7.7%, or 1/13, namely S11) emphasizes self-directed learning with reduced teacher involvement; and (vii) *Exploratory and Recommendation-Oriented Studies* (7.7%, or 1/13, namely S05) provides qualitative insights without consolidated instructional designs.

Overall, the distribution highlights a strong preference for structured and scaffolded implementations, suggesting a cautious pedagogical stance toward LLM integration. Across studies, LLMs primarily support reflection, sense-making, and metacognitive regulation rather than open-ended exploration, indicating an emphasis on epistemic control and alignment with CT learning goals, while more autonomous configurations remain underexplored.

Teacher Roles. Five dominant instructional roles adopted by teachers in LLM-supported CT education were identified. (i) *Facilitator, Scaffolder, or Mediator* (30.8%, or 4/13, namely S04, S07, S08, and S10) positions teachers as guides who support interpretation of AI feedback and higher-order reasoning; (ii) *Designer of the Pedagogical Intervention* (30.8%, or 4/13, namely S04, S07, S09, and S13) emphasizes intentional task design and prompting strategies; (iii) *Technology Integrator* (23.1%, or 3/13, namely S01, S08, and S10) focuses on operationalizing responsible LLM use; (iv) *Technical Support or Minimal Pedagogical Presence* (15.4%, or 2/13, namely S06 and S11) limits teachers' roles to logistical assistance; and (v) *Absent or Strongly Reduced Teacher*

Role (AI-Centered Learning) (15.4%, or 2/13, namely S09 and S12) delegates substantial instructional responsibility to LLMs, with some residual teacher involvement during design.

These roles indicate a shift from direct instruction toward the orchestration of hybrid human-AI learning environments. The limited prevalence of AI-centered models reflects unresolved concerns regarding the reliability and pedagogical legitimacy of LLM feedback, highlighting a research gap related to teacher preparation, professional development, and AI literacy in CT education.

4.4. RQ4: Learning Outcomes, Evaluation Strategies and Indicators

This research question examined the instruments and indicators used to evaluate learning outcomes in LLM-supported CT education.

Measurement Instruments. The literature is dominated by *researcher-developed, curated, or locally adapted instruments*, with limited use of standardized CT measures. Only 15.4% (2/13; S03, S09) employed a validated CT instrument, namely the Computational Thinking Scale (CTS; adapted from [Korkmaz et al. 2017]). Similarly, 15.4% (2/13; S03, S08) used validated questionnaires for affective, workload, or UX dimensions (e.g., MSLQ, LES, cognitive load questionnaires in S03; SUPR-Q in S08). In contrast, most studies (84.6%, 11/13; S01, S02, S04, S05, S06, S07, S08, S10, S11, S12, S13) relied on researcher-developed or adapted instruments, including HOTS/CT/Bloom-based tests and rubrics (S02), curriculum-aligned achievement tests and CT core-skills questionnaires (S04), discourse coding schemes and ENA pipelines (S05), perception surveys and chat-history classifications (S06, S11), log-based prompt categorizations and usage metrics (S07, S08), localized CTS-based CT tests with expert review (S10), action-research rubrics for prompt efficiency and artifacts (S12), and Bebras-based CT tests combined with self-efficacy and cognitive load questionnaires (S13). Validated and bespoke instruments were frequently combined within the same study. Overall, limitations related to construct validity and cross-study comparability were rarely addressed.

Indicators and Metrics by Domain.

(i) Cognitive Domain (92.3%, 12/13; S01–S05, S07–S13) was assessed using heterogeneous operationalizations. Indicators included *CT tests and subfactor assessments* (CTS in S03, S09; localized CTS-based tests in S10; Bebras-based tests in S13; CT core-skills questionnaires in S04; HOTS/CT/Bloom hybrids in S02), *performance proxies* (grades, exams, or curriculum-aligned tests in S03, S04, S08, S11), and *interaction-grounded signals*, such as CT discourse coding and network structures (S05), solution correctness or quality (S07, S11), and refinement of reasoning in student artifacts (S12). Consequently, CT learning was often inferred indirectly rather than measured through shared instruments.

(ii) Affective–Motivational Domain (84.6%, 11/13; S01, S02, S03, S04, S06, S07, S08, S09, S11, S12, S13) relied mainly on *questionnaires and surveys* (e.g., MSLQ and LES in S03; SUPR-Q and custom usefulness items in S08; learning experience questionnaires in S09; structured perception surveys in S06, S11), complemented by *interviews and qualitative reflections* (S01, S04, S09, S12, S13). Indicators addressed motivation and engagement (S03, S06, S08, S09), self-efficacy and confidence (S13; qualitatively in S01, S02), perceived usefulness and satisfaction (S07, S08), and interest or

creative expression in student work (S11, S12). Evidence was largely self-reported, with limited alignment to validated motivational frameworks or explicit consideration of LLM-specific risks.

(iii) Behavioral and Process Domain (92.3%, 12/13; S01–S09, S11–S13) was examined primarily through *trace- and log-based analytics* and *interaction coding* (with S10 as the main exception). Indicators included prompt quality and evolution (S02, S07, S08, S12), iteration and attempt patterns (S07), frequency and type of help-seeking or tool usage (S01, S06, S08, S11), and structured modeling techniques such as ENA (S05), lag sequential analysis and clustering (S09), and coded screen-record analyses (S04). Despite their prevalence, definitions and metrics (e.g., "prompt quality") varied substantially across studies, limiting cross-study comparability.

5. Discussions

The literature on using LLMs to teach CT reveals both emerging practices and persistent gaps. This mapping analyzed 13 primary studies, mostly published between 2024 and 2025, suggesting a field still in early consolidation.

Across studies, LLMs are primarily deployed for evaluation/feedback and tutoring, reflecting a focus on scalable, individualized support for CT instruction. This prevalence of tutoring and scaffolding roles is consistent with foundational CT frameworks that emphasize the importance of structured, mediated support for the development of skills such as abstraction and algorithmic decomposition [Brennan and Resnick 2012], suggesting that LLMs are being adopted in roles that mirror established principles of supported CT learning. The frequent use of code generation and pair programming also indicates increasing alignment with contemporary programming workflows, positioning LLMs as both cognitive partners and productivity enhancers. Overall, implementations emphasize responsive support (evaluation, tutoring, assistance) rather than generative or design-oriented roles, which remain a promising but underdeveloped direction.

In terms of pedagogy, teachers are central to integrating LLMs into CT learning. Only S12 and S09 substantially minimize the teacher role through LLM-centered approaches; in most studies, teachers mediate tool use and guide interpretation of AI feedback. The predominance of structured, scaffolded designs indicates a cautious stance toward LLM integration, while more autonomous learner-driven configurations remain insufficiently understood, particularly regarding students' regulation of AI-supported learning.

Methodologically, evaluation strategies and indicators vary widely, and discussions of construct validity and cross-study comparability are rare. Only two studies (S03, S09) employed a validated CT instrument (CTS), and two (S03, S08) used validated questionnaires for affective/UX dimensions; most (11/13) relied on researcher-developed or locally adapted measures. This pattern underscores exploratory study designs and the absence of consolidated assessment protocols, which limits generalizability.

Finally, the evidence base is concentrated in higher education, with only four studies in K-12 settings, indicating a critical gap in primary and secondary contexts. Instruction is predominantly synchronous face-to-face (7/13; 53.8%), with fewer hybrid (3/13; 23.1%) and asynchronous implementations (1/13; 7.7%). Longitudinal evidence is virtually absent, limiting insight into sustained effects. In addition, teacher preparation and

AI literacy are underexplored despite the centrality of pedagogical design and mediation, and contextual reporting (e.g., participant diversity and setting characteristics) is often limited.

6. Threats to Validity

Construct validity was addressed through the adoption of a GQM-based protocol, which ensured alignment among research objectives, questions, and data extraction items. The study was conducted following established guidelines for systematic mapping studies [Petersen et al. 2008]. Additionally, a predefined search string was applied across multiple digital libraries as recommended by [Kitchenham et al. 2015]. However, the application of LLM-related filtering terms at the title level may have reduced recall by excluding studies that discuss generative AI primarily in abstracts or full texts. This strategy was adopted to prioritize precision and ensure that included studies explicitly centered LLM use. The internal validity threats were mitigated by conducting study selection and data extraction independently by multiple researchers, with disagreements resolved through consensus. Finally, the conclusion validity was mitigated by relying on descriptive synthesis methods appropriate for systematic mappings [Felizardo et al. 2017] and by cross-checking results among the researchers, avoiding biased interpretations beyond the scope of the data.

7. Conclusion

Research on the use of LLMs in CT education has grown in recent years, but the existing literature remains largely exploratory and uneven in methodological rigor. Across studies, LLMs are most commonly examined as tools for evaluation, feedback provision, or tutoring, often within structured, teacher-mediated instructional designs. In these implementations, teachers typically assume the role of designers and facilitators of human–AI interaction. At the same time, the evidence base is dominated by short-term studies conducted in higher education, with comparatively limited representation of K–12 settings and other educational contexts.

Future research can address these limitations in several ways. Additional studies are needed in primary and secondary education and across a wider range of learning contexts. There is also a need for standardized and validated assessment tools that take into account students' use of LLMs when evaluating CT, including formative and continuous assessment of student CT development when LLMs are used with a tutor. Further research should examine how LLM supported CT instruction functions across in person, hybrid, and online settings. In addition, studies on teacher preparation and AI literacy should consider both the potential benefits and limitations of LLM use, as well as the role of teacher mediation in shaping the impact of LLM based approaches on student CT development.

Use of Artificial Intelligence

ChatGPT (OpenAI) and Gemini (Google) were used to support the preparation of this manuscript. This tools assisted with proofreading, improving clarity and academic wording, rephrasing sentences, and providing technical help with LaTeX formatting (e.g., organizing tables according to the conference template). They were not used to generate or analyze data, produce results, interpret findings, or create references. Full responsibility for the content and accuracy remains with the authors.

Acknowledgments

The authors thank the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for institutional support for the graduate program in which this research was conducted and the REBECA project (CNPq Process: 440425/2024-7).

References

Basili, V. R. and Weiss, D. M. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering*, SE-10(6):728–738.

Benavides-Varela, S., Callegher, C. Z., Fagiolini, B., Leo, I., Altoè, G., and Lucangeli, D. (2020). Effectiveness of digital-based interventions for children with mathematical learning difficulties: A meta-analysis. *Computers & Education*, 157:103953.

Brennan, K. and 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, Vancouver, Canada*, volume 1, page 25.

Dawson, M. G., Deer, R., and Boguslawski, S. (2025). Cognitive dissonance in programming education: A qualitative exploration of the impact of generative ai on application-directed learning. *Computers in Human Behavior Reports*, page 100724.

Elnaffar, S., Rashidi, F., and Abualkishik, A. Z. (2025). Teaching with ai: A systematic review of chatbots, generative tools, and tutoring systems in programming education.

Felizardo, K. R., Nakagawa, E. Y., Scannavino, Fabbri, S. C. P. F., and Ferrari, F. C. (2017). *Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática*. Elsevier Brasil, Rio de Janeiro, 1 edition.

Grover, S. and Pea, R. (2013). Computational thinking in k–12 a review of the state of the field. *Educational Researcher*, 42:38–43.

Heung, Y. M. E. and Chiu, T. K. (2025). How chatgpt impacts student engagement from a systematic review and meta-analysis study. *Computers and Education: Artificial Intelligence*, 8:100361.

Hong, S.-J., Lee, Y., and Kim, S.-W. (2024). Effect of artificial intelligence convergence education using chatgpt on computational thinking of high school students in korea. *International Journal on Advanced Science, Engineering & Information Technology*, 14(6).

Husaeni, A., Fitria, D., Abdullah, A. G., Septem Riza, L., Suherman, A., Husaeni, A., Novia, D., et al. (2025). Trends and impacts of artificial intelligence application in the development of computational thinking skills. *Informatics in Education*, 24(2):261–298.

Kitchenham, B. A., Budgen, D., and Brereton, P. (2015). *Evidence-based software engineering and systematic reviews*. CRC press.

Korkmaz, Ö., Çakir, R., and Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (cts). *Computers in human behavior*, 72:558–569.

Lei, Y., Liu, J., Fu, X., Zhao, J., and Yi, B. (2025). The effects of a generative ai-enabled cdio teaching model on undergraduates' computational thinking and individual psychological constructs. *Computer Applications in Engineering Education*, 33(5):e70075.

Li, Y., Yang, R., Gui, S., Shi, P., Huang, X., Yang, D., Zhang, X., and Gai, Y. (2024). Visualizing program behavior: A study of enhanced program diagrams using llm. In *2024 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE.

Li, Y., Zhou, X., and Chiu, T. K. (2025a). Systematics review on artificial intelligence chatbots and chatgpt for language learning and research from self-determination theory (sdt): what are the roles of teachers? *Interactive Learning Environments*, 33(3):1850–1864.

Li, Z., Zheng, X., and Fu, Q. (2025b). Exploring the computational thinking process of college students: Collaborative programming with llms. In *2025 7th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 6–10. IEEE.

Liao, J., Zhong, L., Zhe, L., Xu, H., Liu, M., and Xie, T. (2024). Scaffolding computational thinking with chatgpt. *IEEE Transactions on Learning Technologies*, 17:1628–1642.

Miao, F. and Holmes, W. (2023). Guidance for generative ai in education and research. Technical report, UNESCO.

Nathaniel, J., Oyelere, S. S., Suhonen, J., and Tedre, M. (2025). Investigating the impact of generative ai integration on the sustenance of higher-order thinking skills and understanding of programming logic in programming education. *Computers and Education: Artificial Intelligence*, page 100460.

Ouaazki, A., Bergram, K., and Holzer, A. (2023). Leveraging chatgpt to enhance computational thinking learning experiences. In *2023 IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)*, pages 1–7. IEEE.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *12th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pages 1–10. BCS, BCS Learning & Events Ltd.

Raihan, N., Siddiq, M. L., Santos, J. C. S., and Zampieri, M. (2024). Large language models in computer science education: A systematic literature review.

Sanchez, J. A., Flores-Eraña, G., Silva-Campos, J. M., Chavira-Quintero, R., and Olais-Govea, J. M. (2025). Generative ai as a cognitive mediator: A critical-constructivist inquiry into computational thinking in secondary education. In *Frontiers in Education*, volume 10, page 1597249. Frontiers.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.

Yan, Y.-M., Chen, C.-Q., Hu, Y.-B., and Ye, X.-D. (2025). Llm-based collaborative programming: impact on students' computational thinking and self-efficacy. *Humanities and Social Sciences Communications*, 12(1):1–12.

Yunianto, W., Lavicza, Z., Kastner-Hauler, O., and Houghton, T. (2024). Investigating the use of chatgpt to solve a geogebra based mathematics+ computational thinking task in a geometry topic. *Journal on Mathematics Education*.

Zhan, Z., He, L., Tong, Y., Liang, X., Guo, S., and Lan, X. (2022). The effectiveness of gamification in programming education: Evidence from a meta-analysis. *Computers and Education: Artificial Intelligence*, 3:100096.

Zhang, X., Zhang, P., Shen, Y., Liu, M., Qiong, W., Gasevic, D., and Fan, Y. (2024). A systematic literature review of empirical research on applying generative artificial intelligence in education. *Frontiers of Digital Education*, 1:223–245.

Zhao, J.-H., Shangguan, S.-T., and Wang, Y. (2025). Exploring the effects of the cer model-based genai learning system to cultivate elementary school students' computational thinking core skills in science courses. *Journal of Computer Assisted Learning*, 41(5):e70110.

Zönnchen, B., Hobelsberger, M., Socher, G., Thurner, V., and Ottinger, S. (2025). Exploring the role of large language models as artificial tutors. In *2025 IEEE Global Engineering Education Conference (EDUCON)*, pages 1–10. IEEE.