2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50919

Towards a Metasynthesis of Recent K-12 Computer Science Education Research (Fundamental)

Dr. Julie M. Smith, Institute for Advancing Computing Education

Dr. Julie M. Smith is a senior education researcher at the Institute for Advancing Computing Education. She holds degrees in Software Development, Curriculum & Instruction, and Learning Technologies. Her research focus is computer science education and AI education, particularly the intersection of learning analytics, learning theory, and broadening participation. She was a research assistant at MIT's Teaching Systems Lab; she is also co-editor of the SIGCSE Bulletin.

Monica McGill, Institute for Advancing Computing Education

Monica McGill is President & CEO of the Institute for Advancing Computing Education (formerly known as CSEdResearch.org). Have previously worked in industry and academia, McGill is using her experiences as a computer scientist, professor, and researcher to enable others to build a strong foundation of CS education research focused on all children while also honoring their backgrounds and experiences.

Mr. Precious Chukwuweike Eze, Florida International University

Precious Eze is a first-year doctoral student in Engineering and Computing Education at SUCCEED, Florida International University, and advised by Dr. Bruk Berhane. Before Coming to FIU, he completed his undergraduate degree in Chemical Engineering (2008 - 2012) at Enugu State University, Nigeria, and his Master's degree in Chemical Engineering (Cum Laude) from 2017-2019 at the University of South Africa. He has also taught Physics & Engineering Sciences at High School, and College Levels in South Africa from 2014 - to 2017 and was a graduate research assistant at the University of South Africa from 2018 - to 2020. Eze is also a member of the Engineering Council of South Africa, IBM South Africa Data Science Group, and the Engineering Education African Fellows. His research interest is in Technology Enhanced Learning, specifically: "Examining the potential of AI to improve learning in Engineering & Computing Education in line with the sustainable development goals and UNESCO 2030 agenda."

Charity Oluwafunmbi Odetola

Charity Oluwafunmbi Odetola is a graduate student at the University of North Carolina Greensboro (UNCG), where she is pursuing a doctorate in Educational Research Methodology (ERM), with an emphasis on Program Evaluation. She is an emerging evaluator and dedicated researcher whose work focuses on decolonizing methodologies in both research and evaluation theory and practice. While her primary academic focus centers on the African diaspora and evaluation, her broader research spans STEM and social justice-oriented research.

Towards a Metasynthesis of Recent K-12 Computer Science Education Research (Fundamental)

Julie M. Smith[1], Monica M. McGill[2], Precious Chukwuweike Eze[3], and Charity Oluwafunmbi Odetola[4]

[1,2, 3, 4]Institute for Advancing Computing Education
[1]julie@csedresearch.org
[2]monica@csedresearch.org
[3]peze001@fiu.edu
[4]CharityOdetola@outlook.com

Abstract

Introduction. This *fundamental research* paper explores dimensions of a metasynthesis of K-12 computer science (CS) education research. In recent years, the study of K-12 CS education has grown in breadth and depth across various topics and contexts. Individual literature reviews can cover a topic or a context, but it is challenging to answer more comprehensive questions, like *What teaching strategies works best for various populations and classroom contexts?* Such a question requires a broader view of the corpus of literature.

Research Objective. This paper describes part of a pilot project designed to create a tool-supported, human-produced synthesis of recent K-12 CS education research. Our research question addressed in this paper was: *What does the set of human extracted, cleaned and organized findings reveal about the state of K-12 CS education research?*

Methodology and Processes. After establishing an inclusion/exclusion criteria of papers that focus on the student impacts of CS education, we then identified over 300 studies published from 2021 to mid-2024. We categorized study findings according to the extended CAPE framework, which is a taxonomy of factors related to capacity, access, participation, and experience in CS and extracted interventions and findings.

Results. Curricula designed to teach computational thinking (CT) are exceptionally common, while research on appropriate hardware for teaching computer science is quite uncommon. Similarly, affective outcomes (e.g., attitude toward CS) are often part of a study's outcomes, while social-familial influences are much less likely to be explored. Research that explores content knowledge frequently focuses on topics related to algorithms and programming and rarely focuses on social and cultural impacts of computing.

Implications. As a result of this analysis, we are able to provide computer science education researchers with both an overview of the landscape of existing research as well as indicators of underexplored topics that would benefit from additional research.

How AI Was Used. AI was used to support creating code for data analysis and visualization.

1 Introduction and Background

Computer science (CS) is a subfield of engineering education, having been born from the fields of engineering and mathematics [1, 2]. Departments of CS across postsecondary institutions can be found in Colleges and Schools of Engineering, still reflecting this history and embracing CS education as a critical part of engineering education. The nature of the fields reflects considerable overlap in certain skills required to successfully learn CS and engineering and, thus, how they should be taught. For example, computational thinking (CT) skills (e.g., pattern recognition, decomposition, abstraction) are increasingly recognized as important in the context of engineering education as well as CS [3–6]. Additionally, professional engineers are required to have CS skills [7] to be successful in their field, an increasing trend as AI (a subset of CS) becomes more integral to the practice of engineering [8].

Progress in engineering and CS education demands researchers and practitioners hold a common, current understanding of which practices are most promising to deliver positive outcomes to all students. Such practices must stand firmly on a foundation of research evidence. The collective work of researchers can coalesce into a clearer understanding of the curriculum, tools, and strategies that will help practitioners meet the educational goals of their students. For example, CT is widely regarded as a key competency in need of future exploration [9–11]. Yet, our knowledge of how best to support CT is still limited, including how to do so in a way that generates equitable outcomes.

The CS education research evidence continues to grow in amount and rigor. As this evidence propagates further, so too does the need to synthesize it. There are multiple ways to synthesize data around promising practices for teaching CS to students, including various types of literature reviews [12], meta-analysis, and novel empirical research. In addition to novel research, study replication and reproduction have been highlighted by the National Science Foundation and the Open Science initiatives as an important contributor to discovering evidence-based promising practices [13, 14], and replication is likewise important in computing education research [15].

Adjacent to this is the need for heterogeneous data collected across similar studies to enable meta-analyses [16]. Consolidating findings via careful meta-analysis can quantify a general effect across many studies and account for different student demographics, slight variations in design, sample size, measures used, and other differing contextual factors. Meta-analysis can therefore more accurately explain the effects of interventions on student and teacher mediating variables and outcome variables [17, 18]. Meta-analytic studies provide a formal quantitative method for synthesizing data and results from various, heterogeneous studies. Yet, in a recent analysis of 507 double-anonymous, peer reviewed articles in K-12 computing education, only 86 (35%) were identified as quantitative studies reporting data including number of participants, mean, and standard deviation – the data one needs to conduct a rigorous meta-analysis [19]. In other words, the field of K-12 computing education research is so new that quantifying promising practices via meta-analysis is challenged by the lack of quality study reports that report specific statistics needed for meta-analytic reviews [20]. Yet, there is a crucial need to understand promising practices so that teachers can fulfill their school's, district's, and state's rapidly changing mandates [21].

Given the above, the K-12 computing education research field writ large still needs to grow to support the data required in meta-analytic reviews. Therefore, we conducted a pilot project to produce a semi-automated, human-driven metasynthesis, in which we synthesize research on K-12 CS education using existing data from a wide review that includes venues focusing on the learning sciences as well as on elementary, middle, and high school education research. Unlike the purely quantitative meta-analysis, this metasynthesis includes studies with diverse methodologies (e.g., qualitative, single group pre/post, experimental/quasi-experimental, correlational) to examine K-12 CS education interventions and their impact. The overarching question for our project is *What is the feasibility of creating a purpose-built database tool that uses human extracted, cleaned, and organized findings from studies on impacts of student experiences in CS education?*. However, for this paper, our narrowed research question was:

> *What does the set of human extracted, cleaned and organized findings reveal about the state of K-12 CS education research?*

2 Methodology

This paper describes part of a larger project designed to create a tool-supported metasynthesis of recent research in K-12 CS education. To answer our research question, we first established inclusion and exclusion criteria [22]. We established that each paper identified for the study should:

- focus on PK-12 computer science education,
- describe their research methods,
- describe a direct effect on students, and
- investigate the student experience of learning computer science.

2.1 Process

2.1.1 Data Extraction

We identified over 300 studies published from 2021 to mid-2024 using our keyword search, finding a variety of CS related articles across several journals and conferences. The full list of studies is available upon request. One researcher first extracted information about the participants, type of study, geographical locale, and other setting variables, and then a second researcher reviewed data the first reviewer extracted. Similarly, one researcher assessed the ethical and technical soundness of each study (see the Appendix for details), and then a second reviewer reviewed this assessment.

During this process, we also categorized study findings according to the extended CAPE framework (see Figure 1).

CAPE is a taxonomy of factors related to four components: capacity, access, participation, and experience in CS. Through our previous work with CAPE, we extended CAPE by deriving subcomponents, categories, and subcategories for each component, resulting in over 300 subcategories. Table 1 shows an illustrative excerpt from the extended CAPE codebook. The codebook hierarchically captures factors that have or may have impacts on student learning.

For this study, since we wanted to synthesize findings that directly impacted student learning, we defined CAPE factors related to *Capacity*, *Access*, and *Participation* as influencing factors, and factors related to students' *Experience* as outcome factors. One researcher identified at least one influencing factor and at least one outcome factor for each study, and a second researcher

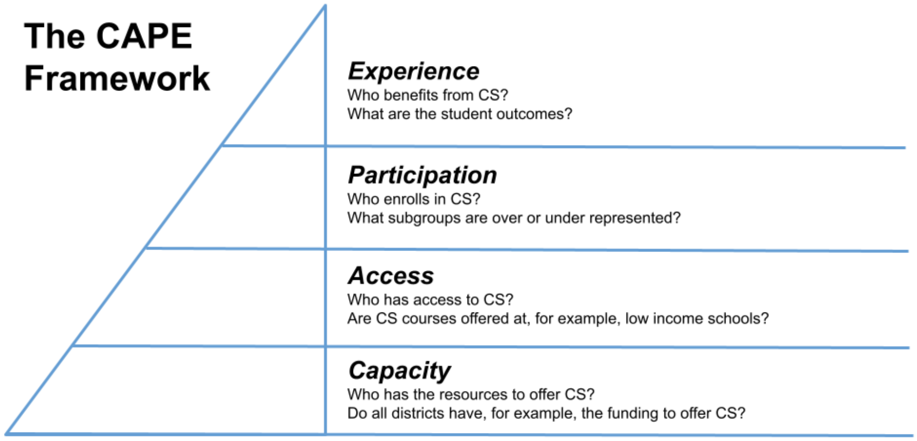

Figure 1: The CAPE Framework. Image adapted from [23].

reviewed the factors. For example, in the paper authored by Sun et al. [24], the authors analyzed the effect on a variety of outcomes (including CT skills, debugging behavior, confidence, and interest) for two groups: (1) students who experienced text-based programming and (2) students who began with block-based programming and then transitioned to text-based programming. CAPE codes applied to this study therefore included *Pedagogy → Student-centered → Text-based Programming*, *Pedagogy → Student-centered → Block-based Programming*, *Cognition → Confidence (in CS)*, and *Affect → Interest (Computing)*, among others.

One researcher then extracted interventions and student-related outcomes attributed to the intervention [25], and a second researcher reviewed the data.

Table 1: Excerpts from the extended CAPE codebook

Component	Subcomponent	Category	Subcategory
Capacity	Curriculum	Resource materials	Hardware
Capacity	Funding	Type	Salaries/Wages
Capacity	Pedagogy	Student-centered	Inquiry-based
Access	Curriculum offerings	Integrated CS	Mathematics
Access	Curriculum offerings	Types	AP CS A
Participation	Course enrollment	Integrated CS	Humanities
Experience	Content knowledge	Algorithms and programming	Debugging
Experience	Content knowledge	Impacts of computing	Safety, law, and ethics
Experience	Learning Strategies	Meta-cognitive	Critical thinking
Experience	Engagement	Affect	Identity in CS
Experience	Engagement	Cognition	Motivation in CS

2.1.2 Data Analysis of the Resulting Dataset

We identified which elements of the extended CAPE codebook are more and less common in our resultant dataset. We then examined the intersection of factors, such as which interventions (e.g., a particular pedagogical approach such as the use of unplugged activities) are more and less likely to be paired with which study outcomes (e.g., a change in students' sense of belonging in computing). We also mapped aspects of the extended CAPE codebook to the characteristics (e.g., grade level, research approach) of study participants to identify patterns (e.g., whether research on meta-cognitive outcomes occurs more often with younger or older students and/or more often in qualitative or quantitative research).

2.2 AI versus Human Review

Generative AI has made significant strides over the last few years, and tools like Claude and Gemini offer paper summaries and can aid the literature review process [26, 27]. As noted by Wagner et al., the potential for AI support is moderate for problem formulation, very high for searching papers, high for initial screening and moderate for secondary screening (which requires more expert judgment), and low-to-moderate for quality assessment [26]. When it comes to data extraction, they note that AI has moderate potential for reviews that require extraction of data, high potential for "objective and atomic data items" like sample sizes, and low potential for more complex data (e.g., main conclusions, theoretical arguments). Finally, they found the potential for using AI in literature reviews as very high for synthesis of descriptive studies, moderate for developing and testing theory, and low to non-existent for reviews that adopt traditional and/or interpretive methods.

Experienced human education researchers (particularly in K-12 CS) can identify nuances in the studies that may be more or less important [26]. Though it is not a small lift for one or two human researchers to review hundreds of papers and provide a consistent review of them, given that it is a time-intensive task, the results can be consistent, transparent, and easily verifiable. Humans also have the capacity to rate the technical and ethical soundness of the studies, which is needed to ensure that higher quality papers are given appropriately more weight when synthesizing the findings. Thus, for this study, we relied on humans for the data extraction process.

3 Results

3.1 Dataset Characteristics

There were a total of 337 papers in the dataset. Of those, 147 (or 44%) were quantitative research, 86 (26%) were qualitative, and 104 (31%) were mixed methods.

Table 2 shows the average (mean) scores for measures of ethical and technical soundness, where 4 = highly confident, 3 = moderately confident, 2 = somewhat confident, and 1 = no confidence.

3.2 Common Factors

By definition, all papers in the dataset had at least one *Experience* factor (i.e., an outcome). For the influencing factors, 99% (n = 334) had at least one *Capacity* factor, 1% (n = 4) had at least one *Access* factor, and 2% (n = 7) had at least one *Participation* factor.

Figure 2 shows the ten influencing factors that occurred most commonly in the dataset. All are part of the *Capacity* component, and all ten belong to either the *Curriculum* or the *Pedagogy*

Table 2: Mean Scores for Soundness Measures (4 = highly confident and 1 = no confidence).

Measure	**Mean**
Ethical Soundness	
Limitations acknowledged	3.0
Asset-based framing	2.6
Respect for participant knowledge	2.5
Adhered to ethics review boards	2.2
Author positionality acknowledged	1.4
Technical Soundness	
RQ(s) clearly framed	3.7
Results supported by appropriate statistics (quantitative)	3.6
Design and methodology clearly defined	3.6
Appropriate statistical analysis (quantitative)	3.5
Appropriate data analysis categories	3.4
Evidence of reliability and validity (quantitative)	3.2
Triangulation or integration (mixed methods)	3.0
Consistent data collection/analysis techniques (qualitative)	2.9
Described the coding approach used (qualitative)	2.9
Participants from all relevant groups	2.7
Evidence of reliability and validity (qualitative)	2.4

subcomponents of *Capacity*. Within Curriculum, computational thinking (CT) and CS curriculum were most common. The most common pedagogical practices were *Block-based Programming*, *Integrated* approaches, and *Robots/Physical Computing*.

By definition, all outcome factors belong to the *Experience* component of the CAPE framework. Figure 3 shows the most commonly occurring outcome factors. With the exception of *Social-Familial Influences*, all of the subcomponents of *Experience* (that is, *Content Knowledge*, *Student Engagement*, and *Learning Strategies*) are represented in the list of most commonly occurring factors, with *Content Knowledge* dominating the list.

We conducted a gap analysis to identify when pairs of factors were over- or under-represented in the dataset. To understand how this analysis was performed, it will be helpful to use two terms: the *main factor* will refer to each of the ten most common influencing factors and the ten most common outcome factors, and the *paired factor* will refer to any outcome factor that co-occurred with one of influencing main factors and to any influencing factor that co-occurred with one outcome main factors. We conduced the gap analysis for every main factor and for any paired factor that occurred at least 5 times in the dataset.

We determined for each main factor the ratio between (1) the percent of instances where the paired factor co-occurred with this factor and (2) the percent occurrence of the paired factor in the dataset overall:

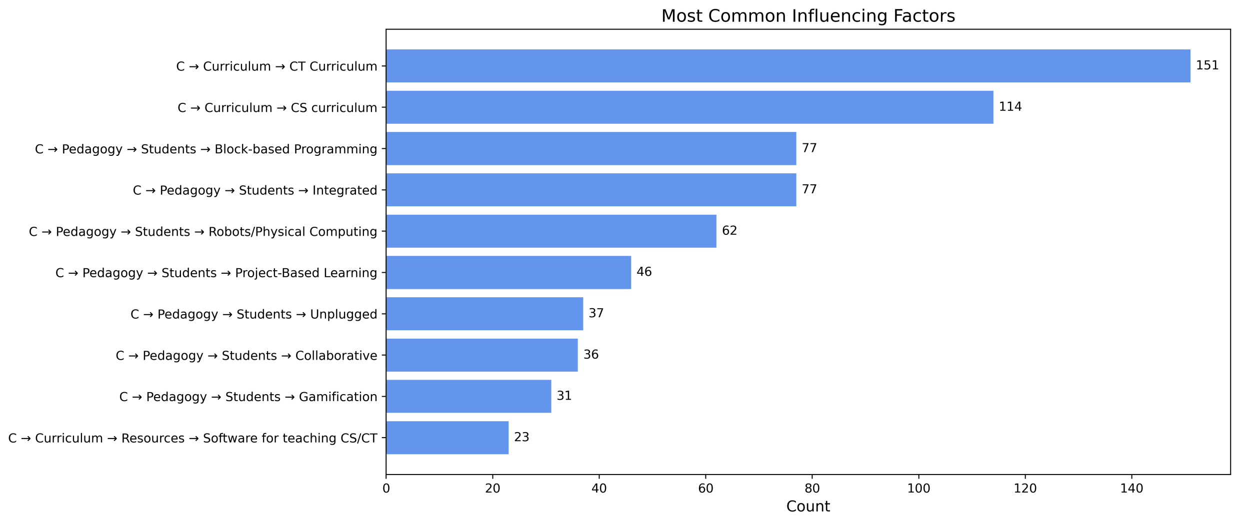

Figure 2: The ten most common influencing factors reported across the dataset.
Abbreviation: “C” is *Capacity*.

$$\text{gap score} = \frac{\text{\% of main factor papers with paired factor}}{\text{\% of all papers with paired factor}}$$

Thus, a gap score > 1 indicates that the paired factor co-occurs with the main factor more often than the dataset average for the paired factor.

Table 3 shows the gap scores for the ten most common influencing factors if the gap scores are >3 or <0.4. Gap scores ≥ 4 occurred for the following pairs: *Software for teaching CS/CT* and data *Visualization and Transformation* (4.9) and *Software for teaching CS/CT* and *Conceptions/Mental Models (in CS)* (4.0). The means that these items co-occurred at least four times more often than would be expected on the basis of the occurrences of these outcome factors in the dataset as a whole. The lowest gap score was for *Integrated* curriculum and *Confidence (in CS)* (0.2), which means that student CS confidence was only one-fifth as likely to be an outcome for studies of integrated curriculum as it was in the dataset as a whole.

Table 4 shows the gap scores for the ten most common outcome factors if the gap scores are >3 or <0.4. The largest gap score was for the pairing of *Interest (Computing)* and *Encouragement of their children* (4.8), which occurs almost five times as often as would be expected based on the overall occurrence of *Encouragement of their children* in the dataset. The lowest gap score was for *Modularity* and *Pair Programming* (0.2), meaning that measures student content knowledge for *Modularity* occurs about one-fifth as often in studies of *Pair Programming* than in the dataset overall.

3.3 Common Factors by Study Characteristics

We determined the most common influencing factors and outcome factors among the papers in our dataset by study characteristics.

Table 5 lists the most common influencing and outcome factors by grade band. There is a high

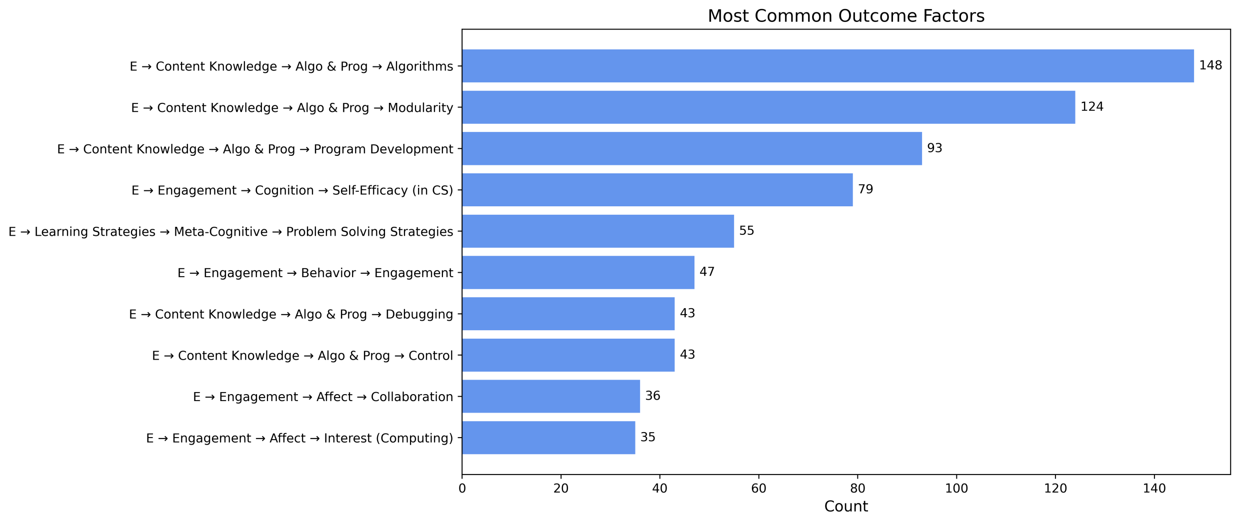

Figure 3: The ten most common outcome factors.
Abbreviations: "E" is *Experience*; "Algo & Prog" is *Algorithms & Programming*.

degree of uniformity across grade bands, evidenced by the extensive overlaps in entries in this table across the different grade bands. Exceptions to this trend are three influencing factors that appear for only one grade band: *Unplugged* activities (elementary school), *Block-based Programming* (middle school), and *Project-Based Learning* (high school). For the outcome factors, the only item that appears as a top factor for only one grade band is *Engagement* for the elementary grades.

Figure 4 shows the gap scores by grade band for all CAPE subcategories that occurred in the dataset. A score of 0.00 in this figure indicates that a subcategory does not occur for a given grade band. As the figure shows, the *Capacity* subcategories of *Funding* and *Standards* are about six times as likely to occur in studies with K-5 students as in the dataset as a whole.

We also determined the count of papers by research type for the influencing and for the outcome factors that occur at least 20 times. As Figure 5 shows, there is some variety in the distribution of research types. For example, the percent of a factor's quantitative research ranged from 21% for pedagogy that is *Integrated* with another subject to 65% for *Gamification*. The percent of qualitative research ranged from 17% for *Software for teaching CS/CT* and for *Interest Computing)*to 50% for *Attitudes (toward CS/CT)*.

4 Discussion

4.1 Ethical and Technical Soundness

As described above, studies were rated for several types of ethical and technical soundness. For ethical soundness, the highest level of confidence was earned (on average) for acknowledging limitations and the lowest confidence for acknowledging author positionality. For technical soundness, the highest confidence was for clearly framing research questions and the lowest confidence for providing evidence for the reliability and validity with qualitative methods.

Overall, the average score across all types of ethical soundness was 2.36 and across all types of technical soundness was 3.18. Thus, there was more confidence in the studies' technical

Table 3: Gap Scores (if >3 or <0.4) for the Ten Most Common Influencing Factors

Influencing Factor	**Outcome Factor**	**Gap Score**
C → Curriculum → CT Curriculum	E → Engagement → Cognition → Intentions (To Study Computing)	0.3
C → Curriculum → CS curriculum	E → Engagement → Affect → Perceptions	0.3
C → Pedagogy → Students → Integrated	E → Engagement → Cognition → Cognitive Load (in CS)	0.3
C → Pedagogy → Students → Integrated	E → Engagement → Cognition → Intentions (To Study Computing)	0.3
C → Pedagogy → Students → Integrated	E → Engagement → Cognition → Usefulness (of CS)	0.3
C → Pedagogy → Students → Integrated	E → Engagement → Cognition → Confidence (in CS)	0.2
C → Pedagogy → Students → Robots/Physical Computing	E → Content Knowledge → Computing Systems → Hardware/Software	3.3
C → Pedagogy → Students → Robots/Physical Computing	E → Content Knowledge → Computing Systems → Devices	3.0
C → Pedagogy → Students → Robots/Physical Computing	E → Engagement → Affect → Perceptions	0.3
C → Pedagogy → Students → Project-Based Learning	E → Content Knowledge → Data and Analysis → Vis. and Transform.	3.7
C → Pedagogy → Students → Collaborative	E → Engagement → Behavior → Participation	3.7
C → Pedagogy → Students → Collaborative	E → Social-Familial Influences → Peer Influences → Encouragement	3.7
C → Pedagogy → Students → Collaborative	E → Content Knowledge → Computing Systems → Devices	3.1
C → Pedagogy → Students → Gamification	E → Engagement → Affect → Collaboration	0.3
C → Pedagogy → Students → Gamification	E → Engagement → Cognition → Self-Efficacy (in CS)	0.3
C → Curriculum → Resources → Software for teaching CS/CT	E → Content Knowledge → Data and Analysis → Vis. and Transform.	4.9
C → Curriculum → Resources → Software for teaching CS/CT	E → Engagement → Cognition → Conceptions/Mental Models (in CS)	4.0
C → Curriculum → Resources → Software for teaching CS/CT	E → Engagement → Behavior → Agency	3.3
C → Curriculum → Resources → Software for teaching CS/CT	E → Content Knowledge → Computing Systems → Devices	3.3

Table 4: Gap Scores (if >3 or <0.4) for the Ten Most Common Outcome Factors
Abbreviation: CECI is "Community Environment, Culture, & Ideology Supporting CS Education Implementation"

Outcome Factor	**Influencing Factor**	**Gap Score**
E → Content Knowledge → Algo & Prog → Algorithms	C → Pedagogy → Students → Use-Modify-Create	0.3
E → Content Knowledge → Algo & Prog → Modularity	C → Pedagogy → Teacher-centered → Scaffolding	0.3
E → Content Knowledge → Algo & Prog → Modularity	C → Pedagogy → Students → Pair Programming	0.2
E → Content Knowledge → Algo & Prog → Control	C → Pedagogy → Students → Use-Modify-Create	3.4
E → Content Knowledge → Algo & Prog → Debugging	C → Pedagogy → Students → Differentiation	3.1
E → Engagement → Cognition → Self-Efficacy (in CS)	C → Pedagogy → Students → Gamification	0.3
E → Engagement → Affect → Collaboration	C → Pedagogy → Students → Pair Programming	3.9
E → Engagement → Affect → Collaboration	C → CECI → COVID-19 Pandemic	3.7
E → Engagement → Affect → Collaboration	C → Pedagogy → Teacher-centered → Feedback to students	3.7
E → Engagement → Affect → Collaboration	C → Pedagogy → Students → Gamification	0.3
E → Engagement → Affect → Interest (Computing)	C → CECI → Families → Encouragement of their children	4.8

soundness than in their ethical soundness. These findings may prove useful in determining, for example, what topics might be beneficial for further learning for new researchers, for further research (e.g., the development of best practices and frameworks for researchers to use), or for policies and norms for publication venues.

4.2 Frequent CAPE Factors

An analysis of which CAPE factors occur most frequently suggests what topics are well-covered in CS education research. As 2 shows, all of the most common influencing factors in the dataset are from either the *Curriculum* or the *Pedagogy* subcomponents of the *Capacity* component. No factors related to *Access* or *Participation* were included, suggesting that these factors (e.g., availability of school-based extracurricular activities, participation in AP CS courses) may be under-researched. This finding is supported by the fact that very few papers in the dataset had any *Access* or *Participation* factors, while nearly all papers had at least one *Capacity* factor. Additionally, some *Capacity*-related factors (e.g., policies related to dual credit, flipped classrooms) may also warrant further study.

Table 5: Most Common Factors by Grade Band

Band (count)	Influencing Factor	Outcome Factor
K-5 (56)	C → Curriculum → CT Curriculum	E → Content Knowledge → Algo & Prog → Algorithms
	C → Curriculum → CS curriculum	E → Content Knowledge → Algo & Prog → Modularity
	C → Pedagogy → Students → Robots/Physical Computing	E → Engagement → Cognition → Self-Efficacy (in CS)
	C → Pedagogy → Students → Unplugged	E → Content Knowledge → Algo & Prog → Program Development
	C → Pedagogy → Students → Integrated	E → Engagement → Behavior → Engagement
6-8 (166)	C → Curriculum → CT Curriculum	E → Content Knowledge → Algo & Prog → Algorithms
	C → Pedagogy → Students → Block-based Programming	E → Content Knowledge → Algo & Prog → Modularity
	C → Curriculum → CS curriculum	E → Engagement → Cognition → Self-Efficacy (in CS)
	C → Pedagogy → Students → Robots/Physical Computing	E → Content Knowledge → Algo & Prog → Program Development
	C → Pedagogy → Students → Integrated	E → Learning Strategies → Meta-Cognitive → Problem Solving Strategies
9-12 (159)	C → Curriculum → CS curriculum	E → Content Knowledge → Algo & Prog → Algorithms
	C → Curriculum → CT Curriculum	E → Content Knowledge → Algo & Prog → Modularity
	C → Pedagogy → Students → Integrated	E → Content Knowledge → Algo & Prog → Program Development
	C → Pedagogy → Students → Robots/Physical Computing	E → Engagement → Cognition → Self-Efficacy (in CS)
	C → Pedagogy → Students → Project-Based Learning	E → Learning Strategies → Meta-Cognitive → Problem Solving Strategies
Note: the sum of paper counts exceeds the total paper count since a paper may have participants from >1 grade band		

For the outcome factors (all of which belong to the *Experience* component of the CAPE framework), the most commonly occurring in the dataset were from the *Content Knowledge* subcomponent, with others from *Engagement* and *Learning Strategies*. Thus, one can expect to find many studies investigating *CS Curriculum* and *Integrated* approaches to instruction and/or assessing that curriculum via various measures of *Content Knowledge*. The subcategories *Algorithms* and *Modularity* claim the top two places on the list of common outcome factors, likely due to the fact that they are CT skills, a common topic of research in the dataset. Other *Experience*-related topics (e.g., knowledge of networks, social-familial influences) are less commonly studied and may require additional research.

The gap scores (see Tables 3 and 4) suggest which combinations of influencing and outcome factors are paired at rates much higher or lower than might be expected. We identified some pairs that occur ≥ 4 or ≤ 0.3 times what would be expected based on their average across the dataset. For some of these pairings, their high frequencies relate to the intrinsic relationship between some influencing and some outcome factors. For example, it makes sense that *Software for teaching CS/CT* might focus on data *Visualization and Transformation* and thus be highly correlated in the dataset.

Other often-paired factors may be more difficult to explain, such as *Software for teaching CS/CT* and *Conceptions/Mental Models (in CS)* or *Interest (Computing)* and families' *Encouragement of their children*. Regardless, these and other highly correlated pairs may be in less need of additional research relative to other pairs. Prime candidates for more research include the pairs with the lowest gap scores (which indicate that they are rarely paired); these factors include, for example, *Integrated* curriculum and *Intentions (to study computing)*.

We note that our analysis of gap scores was limited to (1) the influencing and outcome factors that independently occurred most frequently in the dataset and (2) the paired factors that occurred at least 5 times in the dataset. Future research might extend the gap analysis to additional factors in order to better capture which pairs may be under-researched.

Our analysis of the raw numbers of common CAPE factors by grade band (see Table 5) suggests a

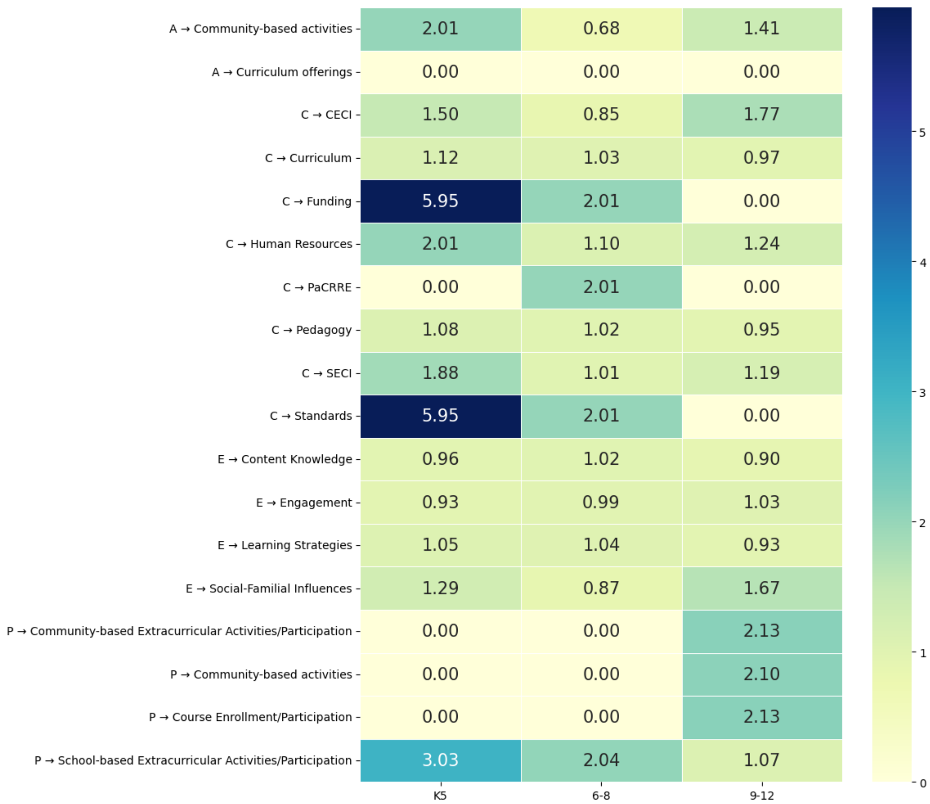

Figure 4: Gap Scores for CAPE Subcategories by Grade Band
Abbreviations: CECI is "Community Environment, Culture, & Ideology Supporting CS Education Implementation," PaCRRE is "Pedagogy and Curriculum are Rigorous, Relevant, and Encourage Sociopolitical Critiques," and SECI is "School Environment, Culture, & Ideology Supporting CS Education Implementation."

high level of uniformity across grade bands. Greater distinctions between grade bands are identified by the gap analysis (see Figure 4), which combines CAPE factors into their subcategories for a higher-level analysis. Some of these differences point to opportunities for additional research, such as the lack of research on *Standards* at the high school level or of *Course Enrollment/Participation* at the elementary level. Similarly, Figure 5 points to differences in research type (e.g., qualitative, quantitative) by CAPE factor and may suggest which types of future research could best expand our understanding of this factor, given the need for "big data" (i.e., quantitative research) combined with "rich data" (qualitative research) for a full picture of a phenomenon in computer science education [28].

Finally, we note the complete absence of some CAPE factors from our dataset (e.g., the influence of *Infrastructure* funding, the availability of *Guidance Counselors*, students' *Empathy* as an outcome). These factors were added to the extended CAPE framework based on a systematic mapping of research literature [25], suggesting their potential importance to computer science education. A key starting place for additional computer science education research is these factors.

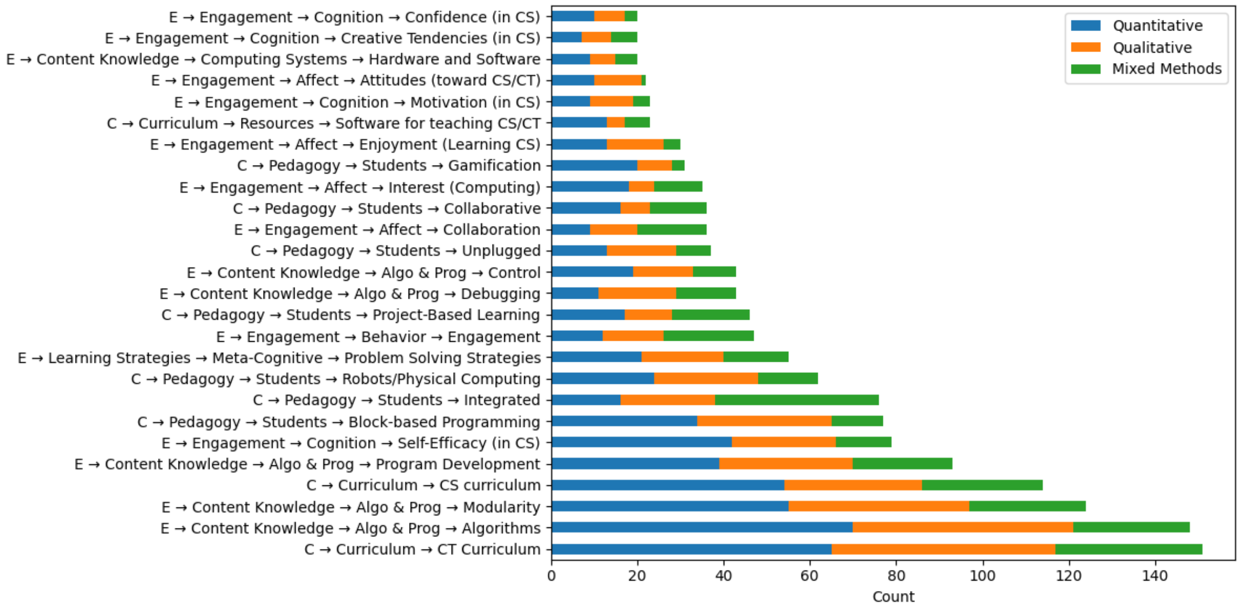

Figure 5: Research Type by Factor

5 Conclusions

Using AI to conduct metasynthesis can offer text related to the larger questions we seek answers to, such as ”For a rural, middle school, what are promising practices for engaging students?” However, while AI may lend itself to this process or appear to be answering prompts correctly and assuredly, there is little assurance that an AI response will be correct giving rise to how trustworthy and reliable these tools are [29]. Without significant lift from researchers to verify that the output it generates reflects an accurate metasynthesis without error, researchers cannot be certain that *AI slop* has not pervaded and corrupted the results.

Our pilot study aims to mitigate these recurring and systemic issues with AI output. For the part of the study that this paper covers, we identified several key findings about the state of K-12 CS education research:

- which ethical and technical factors are ranked relatively high or low when compared to other such factors, suggesting factors (e.g., providing evidence of reliability and validity in qualitative research)
- which CAPE factors are relatively over- or under-researched in the literature, either overall, by grade band, or in relation to another CAPE factor

For example, there is very little research related to how *Access* or *Participation* impact student learning. However, this may be okay, since students cannot learn CS if they don’t have access to or participate in CS education. The more entrenched *capacity* factors, like teacher training, can have a direct impact on student learning.

Overall, these findings provide a view of the landscape of CS education research. They may be useful for preparing future CS education researchers, for providing professional development to current researchers, and for planning future research projects.

Future work stemming from this project will include the development of publicly-accessible tools that will permit users to access the dataset. These tools should permit researchers from across engineering education fields to easily assess the state of CS education research and to determine where gaps in the research might relate to their own work. We hope that this tool will enable researchers to build on prior research findings as well as to identify gaps in the research which they might address. As described in the introduction, CS education is not a concern only for CS education researchers but also for other engineering education researchers, given the importance of CS (including AI) skills in the preparation of engineers in all fields.

These findings may be useful to education researchers who want a better understanding of the landscape of recent research. The findings will also assist in identifying the frequency with which the various influencing factors, outcome factors, and combinations of influencing and outcome factors are represented in the literature.

6 Acknowledgments

We acknowledge Google.org for their generous support of this project.

7 Appendix

To assess ethical and technical soundness, one researcher coded responses to each of the following questions for each study. This work was then reviewed by a second researcher. Response options for each item were *highly confident*, *moderately confident*, *somewhat confident*, and *no confidence*. Items specific to qualitative or to quantitative research included a "not applicable" option for papers that did not use that methodology.

Ethical Soundness

- **Limitations acknowledged**: Acknowledged limitations of study (including threats to validity).
- **Asset-based framing**: Framed problems and questions as asset-based rather than deficit-based, including the study's impacts on and harms to communities as well as ensuring that the choice of participants will enable the answering of the research question(s).
- **Respect for participant knowledge**: Respect for participant knowledge and diversity dimensions as appropriate for the research question(s) and context to maximize generalizability (if quantitative) or transferability (if qualitative) of findings.
- **Adhered to ethics review boards**: Adhere to ethics review boards, as well as the ethic of minimal burden when appropriate to the research methodology. This also includes consent from parents where needed and consent/assent from participants where needed.
- **Author positionality acknowledged**: Acknowledged author(s) positionality.

Technical Soundness

- **RQ(s) clearly framed**: Clearly formulated and framed the research problems and questions, considering scope and context. Whenever possible, a theoretical framework should be defined.

- **Results supported by appropriate statistics (quantitative)**: All results from statistical analyses are supported by appropriate statistical measures.
- **Design and methodology clearly defined**: Defined the research design, selecting appropriate methods of inquiry to answer the research question(s).
- **Appropriate statistical analysis (quantitative)**: Statistical analyses were appropriate for the research questions, and data cleaning for proper data analysis was performed and noted if applicable. This may include appropriate statistical coefficients (e.g., coefficients of determination, effect size, p-values) for inferential analysis.
- **Appropriate data analysis categories**: Discretized commonly used categories and metrics into smaller, more inclusive, and more descriptive categorizations and evaluations.
- **Evidence of reliability and validity (quantitative)**: Evidence of reliability and validity are presented for scales and/or instrumentation used.
- **Triangulation or integration (mixed methods)**: The authors used data triangulation or integration of the quantitive and qualitative results when considering findings.
- **Consistent data collection/analysis techniques (qualitative)**: Used consistent data collection and analysis techniques when multiple researchers are engaged in collecting and analyzing qualitative data. Described how data were transcribed (if applicable) and analyzed (e.g., which software was used).
- **Described the coding approach used (qualitative)**: Described the coding approach used (e.g., inductive vs. deductive, grounded theory, open-coding) and how codes led to findings. Clearly defined resultant themes, categories, and/or codes.
- **Participants from all relevant groups**: Recruited participants from all relevant groups, taking into account factors such as sample size.
- **Evidence of reliability and validity (qualitative)**: Appropriate reliability measures were used or supported. Inter-rater reliability was stated or author(s) established a norm. Described methods used to enhance validity (e.g., member checks, audit trails).

References

[1] Claude E. Shannon. A symbolic analysis of relay and switching circuits. *Electrical Engineering*, 57(12): 713–723, December 1938. ISSN 0095-9197, 2376-7804. doi: 10.1109/EE.1938.6431064. URL `https://ieeexplore.ieee.org/document/6431064/`.

[2] A. Newell, A. Perlis, and H.A. Simon. What is computer science? *Science*, (157):1373–4, 1967. URL `https://www.cs.cmu.edu/~choset/whatiscs.html`.

[3] Sebastian Gross, Mischa Kim, Joachim Schlosser, Coorous Mohtadi, Dan Lluch, and Dorian Schneider. Fostering computational thinking in engineering education: Challenges, examples, and best practices. In *2014 IEEE Global Engineering Education Conference (EDUCON)*, pages 450–459, April 2014. doi: 10.1109/EDUCON.2014.6826132. URL `https://ieeexplore.ieee.org/document/6826132`. ISSN: 2165-9567.

[4] Hendra Hidayat, Zulhendra Zulhendra, Efrizon Efrizon, Vera Irma Delianti, Fitrika Kumala Dewi, Mohd Rizal Mohd Isa, Dani Harmanto, and Jem Cloyd M. Tanucan. Computational Thinking Skills in Engineering Education: Enhancing Academic Achievement Through Innovations, Challenges, and Opportunities. *TEM Journal*, pages 3454–3467, November 2024. ISSN 2217-8333, 2217-8309. doi: 10.18421/TEM134-78. URL https://www.temjournal.com/content/134/TEMJournalNovember2024_3454_3467.html.

[5] Alejandra J. Magana and Genisson Silva Coutinho. Modeling and simulation practices for a computational thinking-enabled engineering workforce. *Computer Applications in Engineering Education*, 25(1):62–78, 2017. ISSN 1099-0542. doi: 10.1002/cae.21779. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cae.21779. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cae.21779.

[6] Noemi V. Mendoza Diaz, Russ Meier, Deborah A. Trytten, and So Yoon Yoon. Computational Thinking Growth During a First-Year Engineering Course. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–7, October 2020. doi: 10.1109/FIE44824.2020.9274250. URL https://ieeexplore.ieee.org/document/9274250. ISSN: 2377-634X.

[7] Osman Yasar and Rubin H. Landau. Elements of Computational Science and Engineering Education. *SIAM Review*, 45(4):787–805, January 2003. ISSN 0036-1445. doi: 10.1137/S0036144502408075. URL https://epubs.siam.org/doi/10.1137/S0036144502408075.

[8] Mohamed Khaleel, Abdussalam Ali Ahmed, and Abdulgader Alsharif. Artificial Intelligence in Engineering | Brilliance: Research of Artificial Intelligence. September 2023. URL https://jurnal.itscience.org/index.php/brilliance/article/view/2170.

[9] Michael Lodi and Simone Martini. Computational Thinking, Between Papert and Wing. *Science & Education*, 30(4):883–908, August 2021. ISSN 0926-7220, 1573-1901. doi: 10.1007/s11191-021-00202-5. URL https://link.springer.com/10.1007/s11191-021-00202-5.

[10] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, March 2006. ISSN 0001-0782, 1557-7317. doi: 10.1145/1118178.1118215. URL https://dl.acm.org/doi/10.1145/1118178.1118215.

[11] Sharif University of Technology and Yahya Tabesh. Computational Thinking: A 21st Century Skill. *OLYMPIADS IN INFORMATICS*, 11(2):65–70, July 2017. ISSN 18227732. doi: 10.15388/ioi.2017.special.10. URL http://www.ioinformatics.org/oi/shtm/v11si_2017_65_70.shtml.

[12] Maria J. Grant and Andrew Booth. A typology of reviews: an analysis of 14 review types and associated methodologies. *Health Information & Libraries Journal*, 26(2):91–108, June 2009. ISSN 1471-1834, 1471-1842. doi: 10.1111/j.1471-1842.2009.00848.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1471-1842.2009.00848.x.

[13] Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251): aac4716, August 2015. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aac4716. URL https://www.science.org/doi/10.1126/science.aac4716.

[14] National Science Foundation and Institute for Education Science. Companion guidelines on replication & reproducibility in education research: A supplement to the common guidelines for education research and development., 2018. URL https://ies.ed.gov/pdf/CompanionGuidelinesReplicationReproducibility.pdf.

[15] James Prather, Brett A. Becker, Michelle Craig, Paul Denny, Dastyni Loksa, and Lauren Margulieux. What Do We Think We Think We Are Doing?: Metacognition and Self-Regulation in Programming. In *Proceedings of the 2020 ACM Conference on International Computing Education Research*, pages 2–13, Virtual Event New Zealand, August 2020. ACM. ISBN 978-1-4503-7092-9. doi: 10.1145/3372782.3406263. URL https://dl.acm.org/doi/10.1145/3372782.3406263.

[16] Monica M. McGill. Discovering Empirically-Based Best Practices in Computing Education Through Replication, Reproducibility, and Meta-Analysis Studies. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research*, pages 1–5, Koli Finland, November 2019. ACM. ISBN 978-1-4503-7715-7. doi: 10.1145/3364510.3364528. URL `https://dl.acm.org/doi/10.1145/3364510.3364528`.

[17] John Hattie. *Visible Learning*. Routledge, 0 edition, November 2008. ISBN 978-1-134-02412-4. doi: 10.4324/9780203887332. URL `https://www.taylorfrancis.com/books/9781134024124`.

[18] John Hattie. *Visible learning for mathematics, grades K-12: what works best to optimize student learning*. Corwin Mathematics, Thousand Oaks, California, 2017. ISBN 978-1-5063-6294-6.

[19] Bishakha Upadhyaya, Monica M. McGill, and Adrienne Decker. A Longitudinal Analysis of K-12 Computing Education Research in the United States: Implications and Recommendations for Change. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 605–611, Portland OR USA, February 2020. ACM. ISBN 978-1-4503-6793-6. doi: 10.1145/3328778.3366809. URL `https://dl.acm.org/doi/10.1145/3328778.3366809`.

[20] Michael Borenstein, Larry V. Hedges, Julian P. T. Higgins, and Hannah R. Rothstein. *Introduction to Meta-Analysis*. Wiley, 1 edition, March 2009. ISBN 978-0-470-05724-7 978-0-470-74338-6. doi: 10.1002/9780470743386. URL `https://onlinelibrary.wiley.com/doi/book/10.1002/9780470743386`.

[21] Code.org, CSTA, and ECEP Alliance. 2023 State of Computer Science Education. Technical report, Code.org, CSTA & ECEP Alliance, 2023. URL `https://advocacy.code.org/2023_state_of_cs.pdf`.

[22] Andrew Booth, Anthea Sutton, Mark Clowes, and Marrissa Martyn-St James. Systematic approaches to a successful literature review. 2021. URL `https://books.google.com/books?hl=en&lr=&id=-n9JEAAAQBAJ&oi=fnd&pg=PP1&dq=Systematic+Approaches+to+a+Successful+Literature+Review&ots=Yb0bqNQxZm&sig=vhwHWaHkwspvKLTavt-2hXKKulo`.

[23] Carol L. Fletcher and Jayce R. Warner. CAPE: a framework for assessing equity throughout the computer science education ecosystem. *Communications of the ACM*, 64(2):23–25, January 2021. ISSN 0001-0782. doi: 10.1145/3442373. URL `https://doi.org/10.1145/3442373`.

[24] Dan Sun, Chengcong Zhu, Fan Xu, Yan Li, Fan Ouyang, and Miaoting Cheng. Transitioning From Introductory to Professional Programming in Secondary Education: Comparing Learners' Computational Thinking Skills, Behaviors, and Attitudes. *Journal of Educational Computing Research*, 62(3):427–454, June 2024. ISSN 0735-6331. doi: 10.1177/07356331231204653. URL `https://doi.org/10.1177/07356331231204653`.

[25] Isabella Gransbury, Monica M. McGill, Angelica Thompson, Sarah Heckman, Jennifer Rosato, and Leigh Ann Delyser. A Framework of Factors that Influence Academic Achievement in Computer Science within Capacity, Access, Participation and Experience. In *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2*, pages 28–29, Chicago IL USA, August 2023. ACM. ISBN 978-1-4503-9975-3. doi: 10.1145/3568812.3603481. URL `https://dl.acm.org/doi/10.1145/3568812.3603481`.

[26] Gerit Wagner, Roman Lukyanenko, and Guy Paré. Artificial intelligence and the conduct of literature reviews. *Journal of Information Technology*, 37(2):209–226, June 2022. ISSN 0268-3962, 1466-4437. doi: 10.1177/02683962211048201. URL `https://journals.sagepub.com/doi/10.1177/02683962211048201`.

[27] Kai R. Larsen, Dirk S. Hovorka, Alan R. Dennis, and Jevin D. West. Understanding the Elephant: The Discourse Approach to Boundary Identification and Corpus Construction for Theory Review Articles. *Journal of the Association for Information Systems*, pages 887–927, 2019. ISSN 15369323. doi: 10.17705/1jais.00556. URL `https://aisel.aisnet.org/jais/vol20/iss7/15/`.

[28] Neil C. C. Brown and Mark Guzdial. Confidence vs Insight: Big and Rich Data in Computing Education Research. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2024, pages 158–164, New York, NY, USA, March 2024. Association for Computing Machinery. ISBN 979-8-4007-0423-9. doi: 10.1145/3626252.3630813. URL https://dl.acm.org/doi/10.1145/3626252.3630813.

[29] David J. Hand and Shakeel Khan. Validating and Verifying AI Systems. *Patterns*, 1(3):100037, June 2020. ISSN 26663899. doi: 10.1016/j.patter.2020.100037. URL https://linkinghub.elsevier.com/retrieve/pii/S2666389920300428.