When Wrong is Right: The Instructional Power of Multiple Conceptions

Lauren Margulieux
Georgia State University
Atlanta, Georgia, USA
lmargulieux@gsu.edu

Paul Denny
The University of Auckland
Auckland, New Zealand
paul@cs.auckland.ac.nz

Kathryn Cunningham
Northwestern University
Evanston, Illinois, USA
kcunningham@northwestern.edu

Michael Deutsch
McGill University
Montreal, Canada
michael.deutsch@mail.mcgill.ca

Benjamin R. Shapiro
Georgia State University
Atlanta, Georgia, USA
bshapiro@gsu.edu

ABSTRACT

For many decades, educational communities, including computing education, have debated the value of telling students what they need to know (i.e., direct instruction) compared to guiding them to construct knowledge themselves (i.e., constructivism). Comparisons of these two instructional approaches have inconsistent results. Direct instruction can be more efficient for short-term performance but worse for retention and transfer. Constructivism can produce better retention and transfer, but this outcome is unreliable. To contribute to this debate, we propose a new theory to better explain these research results. Our theory, multiple conceptions theory, states that learners develop better conceptual knowledge when they are guided to compare multiple conceptions of a concept during instruction. To examine the validity of this theory, we used this lens to evaluate the literature for eight instructional techniques that guide learners to compare multiple conceptions, four from direct instruction (i.e., test-enhanced learning, erroneous examples, analogical reasoning, and refutation texts) and four from constructivism (i.e., productive failure, ambitious pedagogy, problem-based learning, and inquiry learning). We specifically searched for variations in the techniques that made them more or less successful, the mechanisms responsible, and how those mechanisms promote conceptual knowledge, which is critical for retention and transfer. To make the paper directly applicable to education, we propose instructional design principles based on the mechanisms that we identified. Moreover, we illustrate the theory by examining instructional techniques commonly used in computing education that compare multiple conceptions. Finally, we propose ways in which this theory can advance our instruction in computing and how computing education researchers can advance this general education theory.

ICER 2021, August 16–19, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8326-4/21/08…$15.00
https://doi.org/10.1145/3446871.3469750

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

multiple conceptions, conceptual knowledge, instructional design, instructional approaches

ACM Reference Format:
Lauren Margulieux, Paul Denny, Kathryn Cunningham, Michael Deutsch, and Benjamin R. Shapiro. 2021. When Wrong is Right: The Instructional Power of Multiple Conceptions. In *Proceedings of the 14th ACM Conference on International Computing Education Research (ICER 2021), August 16–19, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3446871.3469750

1 INTRODUCTION

The primary goal of this paper is to provide new information for researchers and practitioners who use, blend, and theorize about direct and constructivist instructional approaches. For decades, the education community has debated the merits of each approach, with interest growing broader during the past 20 years [5, 17, 42, 49, 60, 77, 114]. Direct instruction explicitly tells learners content that they are expected to know and, typically, only canonically correct information. Critics of direct instruction argue that it tries to transpose experts' knowledge into a learner's brain in ways that do not draw from principles of how people learn (e.g., [24, 49]). In contrast, constructivism guides learners to construct knowledge for themselves, sometimes by providing space for learners to explore canonically incorrect knowledge to better understand the boundaries of correct knowledge. Critics of constructivism provide evidence that it is less efficient than direct instruction with equivalent learning outcomes (e.g., [60]).

To contribute to this debate, we have analyzed four instructional techniques from each approach, eight in total. Our goal was to find shared underlying mechanisms across techniques that make them successful regardless of whether they used a direct instruction or constructivist approach. Using evidence from this analysis, we propose a new theory to explain the success of instructional techniques from both approaches. Our theory posits that learners develop better conceptual knowledge when they are guided to compare multiple conceptions of a concept, which we will explain in detail in the next section. Using this theory and evidence from the literature, we propose instructional design principles that empower

instructors to successfully implement whichever technique they choose based on their preferred teaching style.

This paper begins by introducing concepts related to our proposed theory. Next, we describe and summarize literature about the eight focal instructional techniques. Then, we share our analysis of mechanistic similarities among the techniques. We use these similarities to propose our theory and related instructional design principles. Finally, we provide examples of these principles applied to computing education contexts and present necessary next steps to further this work.

2 MULTIPLE CONCEPTIONS THEORY

We propose a new theory called multiple conceptions theory – learners develop better **conceptual knowledge** when they are **guided** to compare **multiple conceptions** of a concept. To help explain this statement, this section will define conceptual knowledge, guided, and multiple conceptions. Throughout this section, we ground our definitions in the context of a simple example relevant to computing education: the difference between integers and floats.

Multiple conceptions is a new term that we use when describing the comparison of different conceptions of a concept. In most of the techniques we will showcase, canonically correct conceptions are compared to non-correct conceptions for the purpose of learning. In the case of integers and floats, the correct conception is that integers are a data type that stores whole numbers without decimal places and floats store numbers with decimal places.

To compare against correct conceptions, we considered three types of non-correct conceptions. (1) Incomplete conceptions are not incorrect, but they are missing knowledge. If a learner did not know that integers and floats are different data types, they would have an incomplete conception. Alternatively, if they did not recognize that both fall under the category of data types, their conception is also incomplete. (2) Incorrect conceptions include erroneous information. If a learner misunderstood and thought integers had decimals and floats did not, this would be an incorrect conception. (3) Misconceptions are a type of incorrect conception that often results from the misapplication of prior knowledge to a new concept. A common misconception is that integer division that has a remainder will create a float, which is a misapplication of arithmetic to data types [34, 90]. Comparing multiple conceptions might involve comparing any of these types of conceptions among each other or to the correct conception.

Our theory posits that learners should be guided to compare multiple conceptions. The amount and type of guidance largely relies on the instructional approach: direct instruction or constructivism. Sharing correct conceptions with students is the cornerstone of direct instruction approaches, but as shown in the next section, some techniques explicitly call attention to incorrect conceptions, especially misconceptions. In contrast, constructivist approaches tend to start with learners' incomplete conceptions and build knowledge to develop complete conceptions. Another major difference between approaches is the source of the conceptions, whether they are identified by the teacher or shared by the learner and their peers. Despite differences in approach, all techniques discussed in this paper guide learners to compare multiple conceptions.

Our theory specifically targets conceptual knowledge. From the Revised Bloom's Taxonomy [62], conceptual knowledge is the understanding of how or why a process works. In our example, understanding why different data types are used and how to use them appropriately to solve a problem would demonstrate conceptual knowledge. In contrast, procedural knowledge is knowing how to apply a process to solve a problem, such as knowing how to make an integer or a float. A learner can have conceptual knowledge without procedural knowledge or vice versa. In addition, a precursor to conceptual or procedural knowledge is declarative/factual knowledge, which refers to the ability to recall facts. For example, knowing that an integer does not have decimals and a float does. Conceptual knowledge, while not guaranteeing procedural success, is often considered deep learning and leads to better retention and transfer [62]. From our analysis, comparing multiple conceptions often improves only conceptual knowledge. Given the importance of conceptual knowledge, we do not consider this a limitation of the theory.

We have chosen to submit this new theory to ICER for five reasons. (1) The computing education community, like other education communities, currently debates the trade-offs of direct instruction and constructivist approaches [51, 72]. (2) Like in many other procedural domains, computing students struggle to achieve conceptual knowledge in addition to procedural knowledge [35, 107]. (3) As a domain that is similar to but distinct from others, computing education offers a unique context to examine and advance this area of research. (4) The computing education community is actively calling for theoretical work, especially "to explain how computing teachers design and implement their pedagogical actions," [71, p. 193]. (5) As computing education research continues to grow, it consistently welcomes adaptations of research from other education fields. Thus, when selecting techniques to analyze (see Section 4), we included some that are familiar in computing education research and some that are not to expand the repertoire of techniques used to teach computing. When selecting examples to illustrate multiple conceptions theory in computing education (see Section 7), we have chosen activities that are widely studied in education (Peer Instruction) and unique to computing education (Parsons problems).

3 METHOD

To identify instructional techniques to include in our analysis, we created a list of direct instruction and constructivist approaches that included multiple conceptions. This list was likely not comprehensive, but with the varied expertise of the authors, it likely included most common techniques. Besides selecting techniques based on familiarity in computing education research, as described above, we used several other criteria to create a cohesive list. Given our goal of contributing to the debate between direct and constructivist approaches, we decided to have half of the techniques use direct instruction and the other half constructivism. Because there were more constructivist approaches than direct instruction approaches, we eliminated minimal guidance and sense-making from the list. We chose to eliminate these techniques because they are less defined than the others. We also eliminated peer instruction from the list because it is a variation of test-enhanced learning, which is included,

and we wanted to discuss it as an exemplar of multiple conceptions applied to computing education. We included techniques that range from activities that take a few minutes to module-long projects because our goal is to provide various options to compare multiple conceptions based on the needs and preferences of the instructor.

Just as we are not attempting to provide a comprehensive list of techniques, we are also not attempting to conduct a systematic literature review about the selected techniques. In most cases, each technique has a systematic review or meta-analysis elsewhere, which we cite. Our goal was to analyze the literature with a new lens to determine whether comparing multiple conceptions is consistently a feature of effective techniques. Instead of a systematic review, we used the snowball method for identifying key papers about the techniques and their variations. From commonly cited papers about a technique, we used the reference list to identify new papers to consider. Once new papers did not yield new citations to explore, we considered the core literature about a technique to be complete. Then we mined this core literature for variations of the techniques and differences in efficacy to conduct our analysis about comparing multiple conceptions.

4 TECHNIQUES FOR COMPARING MULTIPLE CONCEPTIONS

For each technique, we have included four key pieces of information, 1) a description of the technique, 2) how it guides comparisons of multiple conceptions, 3) the theory or mechanism that it is based upon, and 4) variations of the technique related to multiple conceptions that are more or less successful.

4.1 Test-Enhanced Learning

Test-enhanced learning refers to the use of testing or quizzing to increase students' retention and understanding of material [93]. The basis for this strategy is the robust phenomenon known as the testing effect: retrieving information from memory enhances the long-term retention of that information [92]. The benefits of test-enhanced learning are well established and generalize to realistic classroom environments across a variety of types of learners, learning materials, and performance measures [75, 83, 112]. The extent to which learners compare multiple conceptions during testing depends on the format and design of the questions used. For instance, multiple-choice questions present learners with the correct answer alongside deliberately incorrect distractor options, which can be designed to target known misconceptions.

Test-enhanced learning is often cited as an example of a desirable difficulty. Desirable difficulties refer to a wide range of learning conditions that are challenging, and hence create difficulty for the learner while acquiring new knowledge and skills [12]. Such conditions tend to slow the rate of apparent learning, often impairing short-term performance relative to easier learning conditions, but lead to better retention over time and more effective transfer [12, 100].

Beyond desirable difficulties, Roediger et al. described both direct and indirect benefits of using test-enhanced learning in the classroom [94]. The direct benefits – improved long-term retention and retrieval accuracy – are hypothesized to connect memory pathways to cues used to retrieve information. Indirect benefits include improved study habits as students prepare for testing and metacognitive benefits relating to the feedback generated from a test, such as helping students identify their knowledge gaps thus improving the efficiency of subsequent study sessions. Corrective feedback is particularly important for preventing students who hold an incorrect conception from reinforcing that misunderstanding through the act of being tested [19].

The benefits of testing can be enhanced when learners are encouraged to make deliberate comparisons between the correct and incorrect conceptions presented in the answers. In reviewing whether multiple-choice testing can promote the type of retrieval processes known to improve learning, Bjork et al. concluded that students should be trained in self-explaining both why incorrect options are wrong and their selected option is correct [11]. To incentivize these comparisons, confidence-weighted scoring can be used to reward students for having high confidence in correct answers [18]. This type of scoring can encourage students to reflect more deeply on incorrect alternatives and their corresponding incorrect conceptions. Sparck et al. found that the use of confidence-weighted scoring on an initial test significantly improved subsequent performance on related questions [108]. The effectiveness of testing can also be enhanced through repetition, particularly if the practice is spread out over time rather than concentrated in a single session [92, 121].

4.2 Erroneous Examples

Worked examples are a form of direct instruction in which learners are presented with a step-by-step solution to a problem [44]. A controversial adaptation of worked examples is erroneous examples, which deliberately replace some of the correct steps in the solution with incorrect steps. Though some intuitively believe that intentionally introducing students to incorrect steps hinders learning, researchers argue that it introduces a desirable difficulty in which students must self-explain why steps are incorrect and reconcile incorrect and correct conceptions [2, 3, 44, 102].

This reconciliation process can uniquely develop conceptual knowledge and promote retention better than correct worked examples while achieving the same procedural knowledge [31]. Whether the erroneous examples are compared to correct examples does not seem to matter, but students must compare multiple examples to be effective [14]. Comparing examples that include erroneous examples is also linked to improved metacognition, specifically error detection [115] and more flexibility in the problem-solving process [102]. Erroneous examples that target common misconceptions can be particularly useful in guiding learners to detect and repair incorrect conceptions [76].

Consistent with other types of desirable difficulties, using erroneous examples can negatively impact students' learning experience and performance early in the process. McLaren et al. [76, 77] found that students who studied erroneous examples exhibited better long-term learning, but not immediate performance, compared to correct examples. Perhaps as a consequence of lower immediate performance, Tsovaltzi et al. [115] found that studying erroneous examples can cause lower self-efficacy than those who practiced solving problems. Similarly, studying erroneous examples causes more confusion and frustration than correct examples [2, 91]. While

examining learner differences, Große & Renkl [44] found that erroneous examples were more effective for students with higher prior knowledge while correct examples were more effective for students with lower prior knowledge. As a result, erroneous examples might be best suited for more advanced learners who can treat finding and recovering from errors as a puzzle rather than a reflection of their aptitude.

4.3 Analogical Reasoning

Analogical reasoning is a powerful teaching tool because analogies can facilitate far transfer, in which students are able to understand new situations that differ substantially from those with which they are already familiar [40]. During analogical reasoning, students learn by comparing examples that share a similar underlying structure, even if they do not have many other commonalities. The goal is to help learners better identify conceptual principles that can aid them when approaching new problems.

There are two main ways to teach with analogical reasoning [50]. Projective analogies suggest how a new concept is similar to something learners already understand, as in the common instruction that "electricity flows like running water." In contrast, mutual analogies do not invoke learners' prior knowledge. Instead, learners compare two examples of a new concept to understand their shared structure, such as the solutions to two algebra problems.

To apply knowledge learned during analogical comparison, learners must generate a mental schema abstract enough to fit both objects in the analogy [40]. The key to building this schema is to recognize that both objects have aspects that relate in the same way to an underlying concept [37, 40]. For example, a learner might realize that both flowing water and electric circuits have small units (i.e., water molecules or charges) that move in a uniform direction through a medium (i.e., pipes or wires). To generate a schema that is effective for analogical reasoning, learners must align aspects of the objects that they are comparing, mapping them such that it becomes clear how different parts of each object play similar roles [37]. When learners describe their schema in terms of relationships rather than common surface details, they are more successful applying the knowledge they have learned [40].

When designing exercises that require analogical reasoning, instructors should consider how easily learners will be able to align the objects they are comparing. Objects that share many concrete similarities (such as two hydrocarbon combustion reactions) are easier to align [39]. However, learners might overfit their schema, and include irrelevant concrete aspects as well as conceptual principles (e.g. believing that combustion occurs only with hydrocarbons). If learners are able to align objects that contain few concrete similarities, the underlying structure is more evident, and analogical reasoning is more likely to facilitate far transfer [45]. Research suggests a balance in practice: comparing objects with some concrete similarity, or using a highly similar analogy at first and a more distant analogy later [21].

Learners are most successful at analogical reasoning when they have the opportunity to build a schema based on multiple examples and when they are explicitly guided to perform alignment. Even when support for abstraction is provided, such as a diagram or description of the conceptual principle at play, it is more challenging for learners to apply analogous knowledge after they have learned from one example than when they have learned from two analogous examples [40]. Even for the same set of two analogous examples, comparing them is more effective than studying them sequentially [38]. However, learners do not always spontaneously identify the key elements of analogous examples. Analogical comparison leads to more successful learning when students are explicitly directed to identify corresponding elements or when the underlying structure is explicitly provided [6].

4.4 Refutation Texts

Refutation texts are an approach to addressing misconceptions, which has been primarily studied in science education [113]. They are effective in science education because, as people interact with the physical world, they can develop misconceptions about how it works [28]. To fully address a misconception, learners must reorganize existing knowledge structures to fit new information (explained further in Section 5) [89]. This requirement makes direct-instruction approaches, which are rarely responsive to individual students' existing knowledge, often ineffective for addressing misconceptions [91]. Refutation texts, however, are consistently more effective than other direct-instruction approaches at addressing misconceptions in science education and improving conceptual knowledge [28, 29, 113].

Compared to expository text, which gives only an explanation, refutation texts have two components: a commonly held misconception and an explanation of the correct concept [113]. These pieces are typically joined by a third component, called a cue, that indicates the misconception is incorrect. Here is a truncated version of an example from Tippett [113].

- Common Misconception: Many people believe that ostriches bury their head in the sand when faced with danger.
- Cue: If this were true, they wouldn't be able to breathe.
- Correct Concept: Ostriches will sometimes listen for sounds with their heads near the ground.

Tippett's [113] review of refutation text literature found that their benefits are robust to most differences in learner characteristics and environments, but they require a couple of features to be successful. They are more effective when students are asked to explain or predict phenomena before reading the text. This activates prior knowledge on the subject and makes the learner recognize whether it is wrong, making them more likely to address existing knowledge if needed. In addition, Tippett found that refutation texts are most effective for students in grades 3-10 (i.e., 8-16 years old). They can be effective outside of this range, but less reliably so. For younger students, it is possible that they have not developed misconceptions yet, so refutation texts are equally effective as expository text because no conceptual change is necessary. For older students, many knowledge structures are more interconnected and crystallized, making them more difficult to change. In this situation, a constructivist approach to conceptual change that more deeply addresses existing knowledge might be more effective than a direct-instruction approach like refutation texts.

4.4.1 Impasse triggers. Related to refutation texts, there is a small body of work on impasse triggers, which at a theoretical level work

the same way – by making learners aware of common misconceptions or gaps in knowledge before explaining a concept. Refutation texts state these misconceptions or gaps explicitly while impasse triggers give students a problem that highlights a misconception, which they are expected to fail to solve [98]. Another variation of an impasse trigger is warning students of common errors made while solving a problem [1]. These impasse triggers are then followed by a correct explanation of how to solve the problem. Like with refutation texts, impasse triggers followed by correct explanations are more effective than correct explanations alone for developing and retaining conceptual knowledge [1, 98].

4.5 Productive Failure

Productive failure asks students to use their prior knowledge and logical reasoning to attempt to solve a problem in small groups before the instructor compares their solutions to the canonical solution [56]. The comparison between multiple conceptions is two-fold: comparisons among learners' incomplete solutions during small group work, and the learners' incomplete solutions compared to the canonical solution. This technique provides a unique combination of constructive, small-group work, called the generation or invention phase, followed by direct instruction that explicitly compares learners' solutions to the canonical solution, called the consolidation phase [52, 53]. When productive failure is successful, students develop better conceptual knowledge than in direct instruction with equivalent procedural knowledge [54, 56].

Proponents of productive failure argue that it is effective because it primes students' prior knowledge as they attempt to find a solution to a problem [55, 70]. When students fail, they recognize the gaps in their knowledge and are better prepared to learn the canonical solution than in pure direct instruction [57]. Furthermore, the student-responsive instruction addresses the prior knowledge that they have applied and whether it has been applied appropriately to the new concept [70].

Comparing solutions is a critical feature of productive failure. Loibl and Rummel [70] found that learners benefited from productive failure only when students' solutions were explicitly addressed in the following instruction – not when instruction focused on only the canonical solution. Moreover, Kapur and Bielaczyc [57] found that students who generated and compared more incorrect solutions performed better than those who found the correct solution early in the exploration process and compared fewer solutions. Similarly, in his review of productive failure literature, Kapur [56] argues that the quality of students' solutions does not matter for later learning, but the number of comparisons being made does.

Comparing typical student solutions to the correct solution rather than students' actual solutions can also be effective. Kapur evaluated a spinoff of productive failure called vicarious failure. In vicarious failure, students do not generate their own solutions, but they still evaluate the typical solutions of students before comparing these incomplete solutions to the canonical correct solution during instruction [55]. He found that vicarious failure was not as potent as productive failure, but it was more effective than instruction that did not compare incomplete solutions to the correct solution. In addition, Loibl and Rummel [70] found that the consolidation phase could be based on typical student solutions rather than solutions generated by the current group of students without affecting performance. Using typical student solutions might capture the benefits of comparing incomplete to correct solutions while providing support for instructors during the consolidation phase.

4.5.1 Invention approach. The invention approach is a predecessor to productive failure. Both approaches ask students to generate procedures to solve problems before giving them direct instruction on the canonical solution. The research conducted on the invention approach focused more on the role of prior knowledge during the generation phase and scaffolding the generation phase [97, 101]. Schwartz and Martin [101] argued that generating solutions allows learners to understand how prior knowledge does or does not apply to a new concept, prompting them to abandon inappropriate knowledge structures and build new ones. They found that following the generation phase with direct instruction was critical to filling new knowledge structures with accurate information [101]. Furthermore, the generation phase appears to improve conceptual knowledge only, and direct instruction is necessary to achieve procedural knowledge [95]. Regarding scaffolding during the generation phase, Roll et al. [97] found that scaffolds must be domain-independent to be effective, such as prompts to engage in metacognition or prompts to discuss a topic with peers. They argued that domain-specific scaffolds were ineffective because they hindered connections to prior knowledge [97]. Scaffolds during the generation phase, however, are unnecessary when students already attempt to make abstract procedures for solving problems, even if those procedures are incorrect [96].

4.6 Ambitious Pedagogy

Ambitious pedagogy combines a cognitively complex instructional task or problem with a reflective class discussion. It shares elements with productive failure, including carefully designed problems with multiple solution paths [52, 118] and distinct generation and consolidation phases [15, 52]. Ambitious pedagogy differs in the aim and orchestration of the consolidation phase [70, 80]. In an ambitious lesson the teacher wields the task and the discussion as complementary instructional tools [15, 79, 118]. The task is designed to be approachable from multiple angles, requiring students to compare multiple conceptions, and is introduced with little scaffolding. Students apply their full range of conceptions to it [22, 79], and reach a result in any way they can. The lesson concludes with a whole-class discussion in which the students share their work, explain their reasoning, and make connections between their ideas.

Ambitious pedagogy was developed in primary level mathematics education in the 1990s out of efforts to promote equitable, rigorous, authentic learning in mixed student populations [22, 80]. In the decades since, educators have applied the ambitious approach at the secondary level and in other subjects, including science [80, 118], history [36], and language arts [36, 80].

Ambitious instruction has been shown to support conceptual learning in many contexts: with high-achieving students [13], students with mixed academic achievement [13, 111], students of different cultural and socioeconomic backgrounds [80]; students with individualized education plans [80]; and language-learning students [80]. In addition to conceptual learning, ambitious instruction has

been shown to support process knowledge and strategic flexibility [13, 111]. It has a behavioral-cultural aspect as well, and a classroom that consistently uses ambitious instruction can begin to feel like a community of practice. Students have reported high levels of engagement and intention to pursue the subject [13] and may begin to identify as practitioners of the field [79].

During ambitious pedagogy, a class co-creates new knowledge out of students' collective thinking [15, 79, 117], or incomplete conceptions. Building upon constructivist and sociocultural learning theories [22] learners are positioned as sense-makers, constructors of their own meaning within the social classroom environment [36]. Ambitious teaching practice relies heavily on instructional structure [64, 118], and has a well established repertoire of discourse "moves" to elicit and develop student responses [58, 64, 80]. These include revoicing or rephrasing student ideas [58], pressing for deeper reasoning [15, 80], extending lines of thinking [13, 58], linking or orienting toward other ideas [13, 15], and revising earlier statements [58]. The teacher takes care to ensure that the students are the ones doing the cognitive and social work of explaining, justifying, and stating and improving ideas [15, 80, 111, 118].

A teacher using ambitious instruction ensures that a wide variety of conceptions are aired in a lesson, offering students a range of ideas they might be able to grasp, evaluate, and either integrate or discard [15, 79]. Teachers make deliberate use of these conceptions to scaffold the class as a whole [118]. Through revoicing, pressing, linking, and other talk moves, the teacher guides students to compare and further develop the conceptions they used to approach the complex task [80]. Ambitious instruction does not seek to limit students' exposure to incorrect, incomplete, or misconceptions; nor does it privilege optimal correct conceptions over less optimal (but still correct) ones. Students holding these other conceptions contribute alternate—if not sophisticated or optimal—ways of thinking about the task at hand and are valued for their legitimate participation in the discourse [13, 79, 80, 111].

4.7 Problem-Based Learning

Problem-based learning (PBL) provides student groups opportunities to consider a variety of ways to conceive and approach an ill-structured problem and, subsequently, construct a solution to that problem. The solution is built over a period of time called a cycle, ranging from a single class or an entire semester. As a result, during a PBL cycle students iteratively grapple with an authentic problem, reflect on their progress and challenges, develop strategies, and construct integrative knowledge [8, 9, 47, 48]. During this iterative process, students are comparing their own incomplete conceptions of the solution to those of their team members, identifying gaps in their knowledge, and asking the instructor for help in identifying a conceptually sound solution.

While many in-depth reviews have been published on PBL [4, 48, 116], we highlight a few key themes here that are relevant to the intended outcomes and the effectiveness of this instructional technique. First, PBL is designed to foster self-directed learning, metacognition or the ability to monitor one's own learning, problem solving skills, teamwork and collaboration, and motivation [61, 78, 99]. Second, a primary goal of PBL is to foster the abilities for students to consider different approaches to problem solving in professional settings. For example, biomedical engineering departments use PBL because students are increasingly asked by employers to demonstrate collaboration skills, integrative thinking, a respect for the value of diverse teams, and the ability to quickly consider multiple approaches to solving a problem [81, 82, 119]. Third, the role of the teacher as a scaffold during a PBL cycle is central. Notably, PBL draws from the model of cognitive apprenticeship [23]. In PBL the teacher acts as a facilitator who provides scaffolding to group problem-solving, reflection, and knowledge construction at the beginning of a PBL cycle but slowly removes this scaffolding over the course of the cycle or in later cycles.

Advocates of PBL argue that it supports learning by developing the types of integrative thinking outlined above (e.g., critical thinking, reflection, diverse perspectives). Reviews and meta-analyses of PBL have found that it is effective for long-term knowledge retention and transferring knowledge outside the classroom and into new contexts [109, 122]. As with many of the techniques that we have discussed, PBL often appears less effective for short-term knowledge acquisition. Many studies call for research to better describe the mechanisms involved in this trade-off. Multiple conceptions theory and our analysis in this paper stand to contribute to describing such mechanisms.

4.8 Inquiry/Discovery Learning

Inquiry learning, interchangeable with discovery learning for our purposes, is a similar technique to PBL. Both techniques ask learners to construct knowledge around a target topic by identifying missing knowledge and using resources to fill that knowledge, with the support of an instructor [5, 65]. The primary difference is that the guiding structure for PBL is to solve an authentic problem while inquiry learning is more open-ended. It is the most unstructured technique that we considered. It does not necessarily include student teams, but like most constructivist techniques, students typically collaborate and learn socially [5]. Thus, inquiry learning involves comparisons of multiple conceptions much like PBL does, by comparing the incomplete conceptions among learners and by asking the instructor to fill in particular knowledge gaps, once they are identified, with canonical conceptions.

Though there is much debate over the efficacy of inquiry learning (e.g., [60, 74]), the underlying mechanisms that support the efficacy of inquiry learning are similar to those for other constructivist techniques. These mechanisms include activation and utilization of prior knowledge, development of metacognitive skills, overcoming desirable difficulties, and the generation effect [17]. The generation effect is a robust phenomenon that refers to the long term benefits of knowledge generation activities, like inquiry learning, due to the cognitive conditions for encoding knowledge being similar to those when retrieving knowledge [10, 105]. Meta-analyses of this technique have found it to be more effective than direct instruction techniques under certain conditions [5, 65].

Both major meta-analyses of discovery learning emphasize the role of guidance in making the technique successful [5, 65]. Alfieri et al. [5] identified four types of discovery learning ranging from least to most guidance: unguided, generation, elicited explanation, and guided discovery. Only elicited explanation and guided discovery (in which students were given scaffolding, unexplained worked

examples, and formative feedback) consistently performed better than direct instruction. The generation types of discovery learning did not perform better or worse than direct instruction.

Multiple conceptions theory can help explain why the generation type of discovery learning is less successful than another technique that uses generation, productive failure. In productive failure, generation is paired with direct instruction, and students are guided to compare their incorrect and incomplete conceptions with the canonical conception. This consolidation phase is a key feature of productive failure. Loibl and Leuders [69] found that in the productive failure paradigm, including students' generated solutions during the consolidation phase was beneficial only when students were prompted to compare the incorrect to correct solutions. This comparison of conceptions might be part of the reason that Alfieri et al. found elicited explanation and guided discovery to be effective as well. Similarly, Lazonder and Harmsen [65] found that more guidance during inquiry learning was more effective than less guidance, in which higher levels of guidance typically involved prompting explanations and comparisons. The importance of comparisons among conceptions is why multiple conceptions theory states that learners must be guided to compare multiple conceptions.

5 SYNTHESIS OF SIMILARITIES AND MECHANISMS

5.1 Themes among direct and constructivist approaches

Among the eight instructional techniques we have profiled, there are similarities in the types of conceptions that are compared and the mechanisms that make them effective. To highlight these similarities, we created Table 1. The table lists each technique as a row and has two sets of columns. The first set of columns indicates the types of conceptions that are compared within the technique. The second set of columns indicates the mechanism that helps learners to build conceptual knowledge.

To be conservative, Table 1 indicates mechanisms only if they are inherent in a technique. For example, all of the techniques could cause conceptual change, but only some of them inherently target it. The mechanisms are ordered from most common to least common.

In this table, an interesting but unsurprising theme emerged regarding the type of conceptions that were compared. Direct instruction approaches focused on comparing incorrect and misconceptions to correct conceptions. Incomplete conceptions were not a central tenet of any direct instruction technique that we examined because direct instruction approaches introduce the canonically correct solution early in the learning process. In contrast, every constructivist approach included building knowledge from incomplete conceptions because they ask learners to identify existing knowledge and construct on top of it. We cannot speculate whether one approach is more effective than the other without further research, but within these approaches, comparing multiple conceptions was more effective than learning only one correct conception. The constructivist approaches also invoked more of the mechanisms on average than the direct instruction approaches, perhaps by virtue of being longer-duration techniques. All of the constructivist techniques took at least a full class period to complete, whereas some of the direct instruction techniques could be completed in as little as a minute. The literature suggests that comparing more conceptions is more effective than less, but the trade-off in time-on-task is always an important consideration in instructional design.

5.2 Mechanisms for effective learning from multiple conceptions

Through our analysis of techniques, we identified five mechanisms that lead to development of conceptual knowledge by comparing multiple conceptions. Each of the mechanisms come from our literature review, but we have extended some mechanisms to techniques whose literature does not explicitly discuss it. For example, vicarious failure is explicitly discussed as a mechanism in the productive failure literature but not in the refutation text literature, even though it applies. Part of our contribution, therefore, is to aggregate mechanisms from various literatures to better explain how these techniques work. This aggregation provides the computing education community common terminology with which to discuss mechanisms underlying instructional techniques.

5.2.1 Vicarious failure. Vicarious failure, or learning from other's errors, is a mechanism for almost every technique except analogical reasoning, which compares only correct conceptions. In direct instruction techniques, the errors are selected by the teacher, and in constructivist techniques, the errors typically come from learners' peers. In either case, learners are guided to compare more conceptions than they would otherwise be exposed to without the instructional scaffolding.

5.2.2 Self-explanation. The second most common mechanism is prompting self-explanation. In direct instruction approaches, the type of self-explanation is typically explaining the differences between incorrect or correct conceptions to determine why they are correct or not. In constructivist approaches, the type of self-explanation is typically building upon incomplete conceptions by applying prior knowledge or processing new information. The constructivist approaches also ask students to work in small groups, so self-explanation can also be an interactive process called peer explanation [20].

5.2.3 Inductive reasoning. Some of the techniques employ inductive reasoning to develop conceptual knowledge. In these techniques, learners receive many examples or cases and compare them to extrapolate canonical concepts. This mechanism can be particularly effective for learners with little prior knowledge because concrete examples are easier to grasp than abstract concepts [32]. We found that many of the techniques that employ this mechanism use scaffolding to support learners in this inductive process. In the constructivist approaches, this process is typically conducted in small groups and includes a social element.

5.2.4 Conceptual growth. The last two mechanisms come from Posner et al.'s model of conceptual change [89]. In this model, there are four ways to process new information. The first way is conceptual growth, which happens when learners add new information to existing, correct knowledge structures. A critical part of conceptual growth is identifying gaps in knowledge that need to be filled [89]. Many of the constructivist approaches primarily aim for conceptual

Table 1: The type of concept comparisons and concept-building mechanisms encouraged by each instructional technique. All techniques make comparisons with correct conceptions. Comparisons that are used in some variations but not others are annotated with '~'. Subcategories of mechanisms are annotated with '∗'.

	Type of Concept Comparison				Concept-building Mechanism				
Instructional Technique	Correct	Incomplete	Incorrect	Misconception	Vicarious failure (∗Student-generated)	Prompts self-explanation (∗Interactive)	Inductive / case-based reasoning (∗Social)	Conceptual growth (∗Multi-step)	Conceptual change
Direct Instruction Approaches									
Test-enhanced learning	x		x	~x	x			x	x
Erroneous examples	x		x	~x	x	x	x		
Analogical reasoning	x					x	x	x	
Refutation texts	x			x	x				x
Constructivist Approaches									
Productive failure	x	x		~x	∗x	∗x		x	x
Ambitious pedagogy	x	x	x	x	∗x	∗x	∗x	∗x	x
Problem-based learning	x	x			∗x	∗x	∗x	∗x	
Inquiry learning	x	x			∗x	∗x	∗x	∗x	

growth on learners' incomplete conceptions. Most of them do this iteratively, going through multiple rounds of identifying knowledge gaps and filling them.

5.2.5 Conceptual change. The other three ways of processing information from Posner et al.'s model of conceptual change deal with existing, *incorrect* knowledge structures [89]. According to Posner et al., learners can take one of three paths when faced with facts that contradict their prior knowledge:

- Ignore the new information because it doesn't fit in existing knowledge structures, and thus, doesn't make sense
- Develop a separate knowledge structure disconnected from the existing knowledge structure for the new information (and perhaps not realize that they are in conflict)
- Reorganize existing knowledge structures to incorporate new information (i.e., conceptual change)

Achieving conceptual change is the most demanding path, which is why misconceptions are notoriously difficult to remedy. All of the techniques that we claim invoke conceptual change explicitly test learners' knowledge and then call attention to incorrect knowledge as a way of forcing learners to recognize when they have faulty knowledge structures.

6 MECHANISM-BASED INSTRUCTIONAL DESIGN PRINCIPLES

From the five mechanisms identified in Table 1, we have generated instructional design principles for creating activities that guide learners to compare multiple conceptions and develop conceptual knowledge. Conceptual knowledge emerges from a web-like cognitive architecture that connect various nodes of information [7]. Each of the mechanisms provides a method for growing, pruning, or reinforcing these connections, captured by the proposed design principles. For each principle, we provide an example of an empirical study from the computing education literature that applies the principle to an instructional design and reports measurable learning benefits.

6.1 Vicarious failure: Introduce learners to non-correct conceptions

Although it may seem counter-intuitive, exposing students to incorrect conceptions, incomplete conceptions, or misconceptions can accelerate their learning through the mechanism of *vicarious failure*. Introducing learners to common non-correct conceptions can help them identify the boundaries of correct conceptions and prune their conceptual network appropriately. Our review of literature showed that the number of conceptions that a student has compared is more predictive of conceptual knowledge than whether those conceptions are correct [56, 69]. We have also seen that non-correct conceptions can come from a variety of sources, including a students' own work or typical student solutions.

This principle was successfully applied by Ginat and Shmallo in an OOP-based CS1 course in which students completed a set of activities that highlighted erroneous conceptions that the authors had commonly observed in previous student cohorts [41]. They found the approach to be effective, as it triggered cognitive conflict in students with misconceptions, and recommend the use of authentic student errors as the basis for these kinds of activities.

6.2 Self-explanation: Prompt learners to compare conceptions

Viewing multiple examples is particularly beneficial when learners are guided to compare key aspects of examples and reconcile differences between them. This process encourages *self-explanation* in which learners reason about the relationships among pieces of information. Comparing conceptions clarifies how a novice should build mental connections between information nodes to develop a network of conceptual knowledge.

In recent work by Tamang et al., this design principle was applied by eliciting self-explanations from students as they were shown a sequence of short code examples targeting fundamental programming concepts [110]. In a controlled experiment, students who self-explained the code examples performed significantly better than those who only made output predictions, and these students frequently drew comparisons between different parts of the code examples while generating their explanations.

6.3 Inductive reasoning: Provide concrete examples, correct or not

To promote *inductive reasoning*, help learners develop new conceptual knowledge by providing multiple correct or non-correct examples. Novices need concrete examples to form nodes that combine into the cognitive architecture that is conceptual understanding. If they have prior knowledge, activate these nodes by asking them to think of examples.

Patitsas et al. apply this principle through the aesthetic design of their instructional materials, by placing examples side by side to guide students in the direct comparison of different solution approaches [85]. They report significant benefits to students, for both code reading and code writing performance, and suggest that the approach helps students appreciate that not all problems have a single correct answer.

6.4 Conceptual growth: Identify gaps in students' knowledge

To facilitate *conceptual growth*, explicitly identify gaps in students' knowledge. Our review of techniques revealed that multiple approaches can be effective in producing conceptual growth. Whether through testing, examples students cannot yet explain, or problems students cannot yet solve, use activities that show students the limitations of their knowledge and what they need to learn.

Xie et al. demonstrate the value of identifying gaps in students' knowledge through their use of formative assessments to measure introductory programming students' performance on discrete programming skills [120]. They found that identifying missing knowledge through formative assessments and targeting that knowledge with explicit instruction improved completion rates on exercises and decreased error rates. Furthermore, their approach improved performance on a delayed test, suggesting better conceptual knowledge [120].

6.5 Conceptual change: Show how misconceptions are incorrect

To enact *conceptual change*, explicitly identify misconceptions from prior knowledge and show how they are incorrect. Whether misconceptions are identified by researchers, instructors, peers, or learners themselves, test whether relevant prior knowledge is applied correctly or incorrectly to new concepts. If misconceptions are present, recognize that conceptual change is an effortful and time-consuming process.

Kennedy et al. apply this idea to a novel approach they call misconception-based feedback, in which students work in pairs to discuss their programs using structured prompts that target common misconceptions [59]. Evaluation of this approach showed that it was effective at enacting conceptual change for students who initially held misconceptions about the difference between pass by reference and pass by value.

7 MULTIPLE CONCEPTIONS IN COMPUTING EDUCATION INSTRUCTION AND RESEARCH

In this section we describe two instructional approaches that have both received much attention in the computing education literature, Peer Instruction and Parsons Problems, and illustrate how their benefits can be explained by multiple conceptions theory. We also propose ideas for future work, including how existing instructional approaches could be modified to better guide students to compare conceptions and how computing education research could help to validate the theory, using notional machines as an example.

7.1 Peer Instruction

Peer instruction is a popular active learning technique, originally developed and studied in physics classrooms [25] and since used with considerable success in many discipline areas. Multiple conceptions theory provides an elegant explanation of the mechanisms that make this technique effective.

A typical peer instruction session takes place during an in-person lecture and involves students responding to multiple-choice questions that target conceptual knowledge. Usually several conceptual questions are presented to students in the course of one session, interspersed with more traditional direct instruction. To support the instructor in collecting and aggregating responses to the questions, students submit votes using a small electronic device, commonly a smartphone in recent years. A typical protocol has students vote individually on a question, then engage in discussion of their responses in small groups with peers sitting nearby, then finally vote a second time on the same question after discussion. Following the second vote, the instructor leads a class-wide discussion of the related concepts, shares the aggregated vote tallies, and explains the correct conception as well as addresses any misconceptions that were targeted by distractor options. Bouvier et al. provide a detailed summary of the structure of a peer instruction session in their review of the factors that influence its adoption in computing classrooms [16].

Over the last decade, peer instruction has received a great deal of attention in the computing education research literature. It has

been shown to scale well as class sizes grow [67, 87], to be an effective method for identifying at-risk students early in a semester [68], and to provide a more enjoyable learning environment compared to classes that do not use peer instruction [104]. Conceptual understanding developed through peer instruction appears to successfully translate to practical computing tasks. Zingaro and Porter found that students who demonstrated learning during a peer instruction session, consisting of conceptually-focused questions, exhibited better performance on later code-writing questions [123]. In addition to direct learning benefits, there is also evidence that peer instruction offers broader positive outcomes for computing students, such as reducing failure rates [86] and improving the retention of majors [88]. While most research on peer instruction in computing courses has been focused at the introductory programming level, there is an increasing interest on its application in upper-level courses such as cybersecurity [27], software testing [43], computer architecture, and courses on the theory of computation [66], which all require substantial development of conceptual knowledge.

Peer instruction guides students to compare multiple conceptions in several ways. This guidance occurs during the initial vote, the peer discussion phase, and the review phase. When selecting a response for the initial vote, students are prompted to compare non-correct to correct conceptions as they evaluate each answer choice [11]. Incorrect conceptions are embedded in the distractor options, and often target common misconceptions [103]. During the peer discussion phase, students are challenged to argue for and defend the answer options they selected, thus incorrect and correct conceptions are directly compared within small peer groups. In fact, it is common for peer groups to eventually arrive at a correct conceptual understanding even when no individual in the group selects a correct answer on the initial vote [106]. Finally, in the review phase, the instructor explains the correct conception and explicitly calls attention to common misconceptions that are highlighted in the vote tally to ensure students have reconciled these incorrect conceptions.

7.2 Parsons Problems

Parsons problems are an alternative activity to writing code in which learners are provided with pre-formed lines of code that they must place in the correct order to solve a problem [26, 84]. Compared to writing the same code, a Parsons problem greatly narrows the problem-solving space and is thus more efficient for novices while maintaining performance on later tasks [33]. Variations of Parsons problems, such as two-dimensional Parsons problems that require the learner to add appropriate indentation, can create a wider problem-solving space.

One variation that is particularly relevant to multiple conceptions theory is Parsons problems with distractors. This variation intentionally includes two lines of code, one correct and one incorrect, that serve the same purpose, and the learner has to pick which to use [33]. Related to multiple conceptions theory, Parsons problems with distractors present a correct and incorrect conception that the learner must compare to successfully complete the problem.

The results of research about Parsons problems with distractors are similar to those about other direct instruction activities that require students to compare multiple conceptions. First, they are more efficient than more open-ended problem solving tasks, resulting in less time on instruction to achieve the same level of performance, even on delayed tests of knowledge and performance [33]. Second, they enable misconceptions to be targeted directly because misconceptions are often used as the distractors [26]. As with the other direct instruction approaches, the misconceptions are only those identified and implemented by the instructor or designer who writes the problem. Most interesting for multiple conceptions theory, though, is the comparison between Parsons problems with and without distractors. Problems with distractors are less efficient during instruction than those without distractors, but they result in no difference in performance [46].

This pattern of results is common when comparing activities that promote procedural knowledge to those that promote conceptual knowledge. For example, worked examples that can be directly mapped onto near transfer problems are more efficient than erroneous examples that require comparison of multiple conceptions, but they result in worse retention and transfer [3, 115]. Similarly, if students are prompted to compare worked examples to inductively extrapolate concepts, such as in analogical reasoning or by using subgoal labels, they typically gain better retention and transfer than if they receive the same examples without a prompt to compare them [63, 73]. Activities that promote conceptual knowledge can be less efficient during instruction than those that promote procedural knowledge for achieving the same procedural proficiency. In farther transfer and retention tests, though, students who have learned with concept-promoting activities typically perform better [49, 114]. This trade-off is a key argument on both sides of the debate between direct instruction and constructivist approaches [49, 60, 114] because each side of the trade-off has value in certain situations. **We argue that by employing multiple conceptions theory and its associated instructional design principles, instructors and designers can help students to develop conceptual knowledge while employing whichever approach best serves their needs.**

7.3 Future work

In this section, we discuss examples of future areas of work in computing education research in which multiple conceptions theory can be validated and expanded. Computing education is similar to some domains, especially other technical fields that require robust procedural and conceptual knowledge, yet unique, especially in the number of canonical, correct conceptions that can co-exist. These features make it a particularly good area of application for research on comparing multiple conceptions.

One potential area to validate the theory and create more specific instructional design principles for computing education is to use multiple conceptions theory to inform new research designs. For example, one relevant research design would be to compare learners writing code with learners solving Parsons problems with and without distractors. In this paradigm, each condition would make different use of multiple conceptions. Parsons problems without distractors do not include multiple conceptions; distractors add direct comparisons between deliberately chosen correct and incorrect conceptions; and writing code implicitly entails multiple

conceptions as students decide what to write and how to fix errors. If these three conditions were compared on immediate near transfer, immediate far transfer, delayed near transfer, and delayed far transfer tasks, we predict that we would find differences among the groups of novice programmers based on multiple conceptions theory. We predict that the Parsons problems without distractors group would be most efficient in successfully completing the immediate near transfer assessment, but they would perform worse on the other assessments that require deeper conceptual knowledge. The Parsons problem with distractors condition is predicted to be more efficient than the writing code condition with equal and more consistent performance across all assessments. In the writing code condition, we expect some students would perform well, but others who do not spontaneously develop conceptual knowledge might perform poorly. These predicted results are mostly aligned with the research described in the Parsons problems section, but they are from separate studies that do not compare all conditions and assessments together. A unified study that tested these a priori hypotheses would lend credence to the theory.

To expand knowledge about computing education, multiple conceptions theory could be applied to support important areas of work, such as the use of notional machines. Notional machines are a pedagogical device designed by instructors to conceptually explain some aspect of computation, such as how code execution works [30, 107]. They are intended to be a bridge between conceptual models (i.e., scientifically-informed, complete, correct models accepted by experts in the field) and a student's mental model (i.e., an often incomplete and imprecise model that increases in accuracy through learning) [35]. As Fincher et al. explain in a recent ITiCSE working group report on notional machines, a teacher creates a notional machine by interpreting a conceptual model into a simplified pedagogical device that is designed to help students form or build upon their personal mental models [35]. The goal of notional machines is to increase the accuracy and sophistication of students' mental models without overwhelming them.

Using notional machines, therefore, is an instructional technique that is well-suited to comparing multiple conceptions. The conceptual model is the same as canonical, correct conceptions that we have discussed throughout the paper. The notional machine is a simplified correct conception used in direct instruction. The student mental models are the same as the incomplete conceptions often compared during techniques that follow a constructivist approach. If teachers preferred a direct instruction approach, they might present multiple concrete examples and ask students to map them onto the notional machine, or they might compare notional machines that include mis- or incorrect conceptions to a correct notional machine to help students develop mental models. If teachers wanted to add a constructivist approach, they might provide a basic notional machine as a starting point followed by a constructivist activity that prompts learners to compare their mental models to build towards the conceptual model. Sorva makes a compelling argument that notional machines should be explicitly acknowledged and used by computing instructors [107]. Multiple conceptions theory provides guidance for *how* this should be done, regardless of the instructional paradigm being used, to produce conceptual knowledge.

In addition to these examples, multiple conceptions theory may provide new contributions to two common challenges in computing education. First, many undergraduate computer science departments face overwhelming enrollment in introductory programming courses. As a result these courses can have large class sizes, making individual interactions between instructors and students difficult and encouraging teachers to use direct instruction techniques. Multiple conceptions theory proposes a strategy, guiding comparison of multiple conceptions, for building conceptual knowledge within direct instruction techniques that can function in small or large class sizes. A second challenge is that computer science courses are increasingly taught online, in which learning environments need to be more thoroughly designed to overcome difficulties in communication. Our proposed instructional design principles may provide guidance for or be expanded by instructors in the future to help their students develop conceptual knowledge specifically in different types of learning environments.

8 CONCLUSION

Computing students face the challenge of developing accurate conceptual knowledge for machines that are both complex and opaque [107]. Thus, instructional approaches that target conceptual understanding are useful to computing educators. Indeed, the computing education research community specifically calls for theoretical work that guides computing teachers to effectively design and implement their pedagogy to complement more plentiful theories about students' understandings of content [71]. In this paper, we proposed a new theory, multiple conceptions theory, that states learners develop better conceptual knowledge when they are guided to compare multiple conceptions of a concept during instruction.

While this theory is applicable to any field, we have focused on the contributions that it can make to computing education. We started developing this theory by analyzing instructional techniques across the spectrum of direct instruction and constructivist approaches from education research in a wide range of STEM fields. All of the techniques that we included can be used in computing education. Through our analysis, we identified five mechanisms that help learners build conceptual knowledge by comparing multiple conceptions. We then translated these mechanisms into five instructional design principles to provide guidance for teachers. Next, we proposed how these mechanisms explain the success of two instructional approaches that are commonly used in computing classrooms, Peer Instruction and Parsons Problems. Finally, we recommended ways in which computing instruction can be improved through application of this theory and how computing education researchers can advance the theory thus making contributions to STEM and teacher education more broadly. We invite others to further explore and define this emerging area of research, and we hope it can also inspire conversations between computing education and other education sub-fields.

ACKNOWLEDGMENTS

This work is funded in part by the National Science Foundation under grants #1941642 and #2030859.

REFERENCES

[1] Santiago R. Acuña, Héctor García Rodicio, and Emilio Sánchez. 2011. Fostering active processing of instructional explanations of learners with high and low prior knowledge. *European Journal of Psychology of Education* 26, 4 (2011), 435–452.

[2] Deanne M. Adams, Bruce M. McLaren, Kelley Durkin, Richard E Mayer, Bethany Rittle-Johnson, Seiji Isotani, and Martin Van Velsen. 2014. Using erroneous examples to improve mathematics learning with a web-based tutoring system. *Computers in Human Behavior* 36 (2014), 401–411.

[3] Deanne M. Adams, Bruce M. McLaren, Richard E. Mayer, George Goguadze, and Seiji Isotani. 2013. Erroneous examples as desirable difficulty. In *International Conference on Artificial Intelligence in Education*. Springer, 803–806.

[4] Mark A. Albanese and Susan Mitchell. 1993. Problem-based learning: A review of literature on its outcomes and implementation issues. *ACADEMIC MEDICINE* 68, 1 (1993), 52–81. https://doi.org/10.1097/00001888-199301000-00012

[5] Louis Alfieri, Patricia J. Brooks, Naomi J. Aldrich, and Harriet R. Tenenbaum. 2011. Does discovery-based instruction enhance learning? *Journal of Educational Psychology* 103, 1 (2011), 1.

[6] Louis Alfieri, Timothy J. Nokes-Malach, and Christian D. Schunn. 2013. Learning Through Case Comparisons: A Meta-Analytic Review. *Educational Psychologist* 48, 2 (2013), 87–113. https://doi.org/10.1080/00461520.2013.775712

[7] John R Anderson. 1996. ACT: A simple theory of complex cognition. *American psychologist* 51, 4 (1996), 355.

[8] The Cognition & Technology Group at Vanderbilt. 1997. *The Jasper Project: Lessons in curriculum, instruction, assessment, and professional development.* Psychology Press.

[9] Howard S. Barrows and Robyn M. Tamblyn. 1980. *Problem-based learning: An approach to medical education.* Springer Publishing Company.

[10] Sharon Bertsch, Bryan J. Pesta, Richard Wiscott, and Michael A. McDaniel. 2007. The generation effect: A meta-analytic review. *Memory & Cognition* 35, 2 (2007), 201–210. https://doi.org/10.3758/BF03193441

[11] Elizabeth Ligon Bjork, Nicholas C. Soderstrom, and Jeri L. Little. 2015. Can Multiple-Choice Testing Induce Desirable Difficulties? Evidence from the Laboratory and the Classroom. *The American Journal of Psychology* 128, 2 (2015), 229–239.

[12] Robert A. Bjork. 1994. Memory and Meta-memory Considerations in the Training of Human Beings. In *Metacognition: Knowing about knowing*, J. Metcalfe & A. Shimamura (Ed.). MIT Press, Cambridge, MA, 185–205.

[13] Jo Boaler and Megan Staples. 2008. Creating Mathematical Futures Through an Equitable Teaching Approach: The Case of Railside School. *Teachers College Record* 110, 3 (2008), 608–645.

[14] Julie L. Booth, Karin E. Lange, Kenneth R. Koedinger, and Kristie J. Newton. 2013. Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction* 25 (2013), 24–34.

[15] Melissa D. Boston and Anne Garrison Wilhelm. 2017. Middle School Mathematics Instruction in Instructionally Focused Urban Districts. *Urban Education* 52, 7 (2017), 829–861.

[16] Dennis Bouvier, Ellie Lovellette, John Matta, Jing Bai, Jacqueline Chetty, Stan Kurkovsky, and Jia Wan. 2019. Factors Affecting the Adoption of Peer Instruction in Computing Courses. In *Proceedings of the Working Group Reports on Global Computing Education* (Chengdu, Sichuan, China) *(CompEd-WGR '19)*. Association for Computing Machinery, New York, NY, USA, 1–25. https://doi.org/10.1145/3372262.3375396

[17] Jerome S. Bruner. 1961. The act of discovery. *Harvard Educational Review* (1961).

[18] James E. Bruno. 1989. Using MCW–APM Test Scoring to Evaluate Economics Curricula. *The Journal of Economic Education* 20, 1 (1989), 5–22.

[19] Andrew C. Butler and Henry L. Roediger. 2008. Feedback enhances the positive effects and reduces the negative effects of multiple-choice testing. *Memory & Cognition* 36, 3 (2008), 604–616. https://doi.org/10.3758/MC.36.3.604

[20] Michelene T.H. Chi and Ruth Wylie. 2014. The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational psychologist* 49, 4 (2014), 219–243.

[21] John Clement. 1993. Using bridging analogies and anchoring intuitions to deal with students' preconceptions in physics. *Journal of Research in Science Teaching* 30, 10 (1993), 1241–1257. https://doi.org/10.1002/tea.3660301007

[22] Paul Cobb, Ada Boufi, Kay McClain, and Joy Whitenack. 1997. Reflective Discourse and Collective Reflection. *Journal for Research in Mathematics Education* 28, 3 (1997), 258–277. https://doi.org/10.2307/749781

[23] Allan Collins, John S. Brown, and Susan E. Newman. 1989. Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, Lauren B. Resnick (Ed.). Erlbaum, Hillsdale, NJ, 453–494.

[24] National Research Council. 2000. *How people learn: Brain, mind, experience, and school: Expanded edition.* National Academies Press.

[25] Catherine H. Crouch and Eric Mazur. 2001. Peer Instruction: Ten years of experience and results. *American Journal of Physics* 69, 9 (2001), 970–977. https://doi.org/10.1119/1.1374249

[26] Paul Denny, Andrew Luxton-Reilly, and Beth Simon. 2008. Evaluating a New Exam Question: Parsons Problems. In *Proceedings of the Fourth International Workshop on Computing Education Research* (Sydney, Australia) *(ICER '08)*. Association for Computing Machinery, New York, NY, USA, 113–124. https://doi.org/10.1145/1404520.1404532

[27] Pranita Deshpande, Cynthia B. Lee, and Irfan Ahmed. 2019. Evaluation of Peer Instruction for Cybersecurity Education. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 720–725. https://doi.org/10.1145/3287324.3287403

[28] Irene-Anna N. Diakidoy and Panayiota Kendeou. 2001. Facilitating conceptual change in astronomy: A comparison of the effectiveness of two instructional approaches. *Learning and Instruction* 11, 1 (2001), 1–20.

[29] Irene-Anna N Diakidoy, Panayiota Kendeou, and Christos Ioannides. 2003. Reading about energy: The effects of text structure in science learning and conceptual change. *Contemporary Educational Psychology* 28, 3 (2003), 335–356.

[30] Benedict Du Boulay. 1986. Some Difficulties of Learning to Program. *Journal of Educational Computing Research* 2, 1 (1986), 57–73. https://doi.org/10.2190/3LFX-9RRF-67T8-UVK9

[31] Kelley Durkin and Bethany Rittle-Johnson. 2012. The effectiveness of using incorrect examples to support learning about decimal magnitude. *Learning and Instruction* 22, 3 (2012), 206–214.

[32] Elsa Eiriksdottir and Richard Catrambone. 2011. Procedural instructions, principles, and examples: How to structure instructions for procedural tasks to enhance performance, learning, and transfer. *Human factors* 53, 6 (2011), 749–770.

[33] Barbara J. Ericson, Lauren E. Margulieux, and Jochen Rick. 2017. Solving Parsons Problems versus Fixing and Writing Code. In *Proceedings of the 17th Koli Calling International Conference on Computing Education Research* (Koli, Finland) *(Koli Calling '17)*. Association for Computing Machinery, New York, NY, USA, 20–29. https://doi.org/10.1145/3141880.3141895

[34] Andrew Ettles, Andrew Luxton-Reilly, and Paul Denny. 2018. Common Logic Errors Made by Novice Programmers. In *Proceedings of the 20th Australasian Computing Education Conference* (Brisbane, Queensland, Australia) *(ACE '18)*. Association for Computing Machinery, New York, NY, USA, 83–89. https://doi.org/10.1145/3160489.3160493

[35] Sally Fincher, Johan Jeuring, Craig S. Miller, Peter Donaldson, Benedict du Boulay, Matthias Hauswirth, Arto Hellas, Felienne Hermans, Colleen Lewis, Andreas Mühling, Janice L. Pearce, and Andrew Petersen. 2020. *Notional Machines in Computing Education: The Education of Attention.* Association for Computing Machinery, New York, NY, USA, 21–50. https://doi.org/10.1145/3437800.3439202

[36] Miranda Suzanne Fitzgerald and Annemarie Sullivan Palincsar. 2019. Teaching Practices That Support Student Sensemaking Across Grades and Disciplines: A Conceptual Review. *Review of Research in Education* 43, 1 (March 2019), 227–248. https://doi.org/10.3102/0091732X18821115

[37] Dedre Gentner. 1983. Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science* 7, 2 (1983), 155–170. https://doi.org/10.1207/s15516709cog0702_3

[38] Dedre Gentner, Jeffrey Loewenstein, and Leigh Thompson. 2003. Learning and transfer: A general role for analogical encoding. *Journal of educational psychology* 95, 2 (2003), 393. https://doi.org/10.1037/0022-0663.95.2.393

[39] Dedre Gentner and Cecile Toupin. 1986. Systematicity and surface similarity in the development of analogy. *Cognitive science* 10, 3 (1986), 277–300.

[40] Mary L. Gick and Keith J. Holyoak. 1983. Schema induction and analogical transfer. *Cognitive Psychology* 15, 1 (1983), 1–38. https://doi.org/10.1016/0010-0285(83)90002-6

[41] David Ginat and Ronit Shmalo. 2013. Constructive Use of Errors in Teaching CS1. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 353–358. https://doi.org/10.1145/2445196.2445300

[42] Inga Glogger-Frey, Katharina Gaus, and Alexander Renkl. 2017. Learning from direct instruction: Best prepared by several self-regulated or guided invention activities? *Learning and Instruction* 51 (2017), 26–35. https://doi.org/10.1016/j.learninstruc.2016.11.002

[43] Bhuvana Gopal and Stephen Cooper. 2021. Peer Instruction in Software Testing and Continuous Integration. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 548–554. https://doi.org/10.1145/3408877.3432404

[44] Cornelia S. Große and Alexander Renkl. 2007. Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and instruction* 17, 6 (2007), 612–634.

[45] Diane F. Halpern, Carol Hansen, and David Riefer. 1990. Analogies as an aid to understanding and memory. *Journal of Educational Psychology* 82, 2 (1990), 298.

[46] Kyle James Harms, Jason Chen, and Caitlin L. Kelleher. 2016. Distractors in Parsons Problems Decrease Learning Efficiency for Young Novice Programmers. In *Proceedings of the 2016 ACM Conference on International Computing Education*

Research (Melbourne, VIC, Australia) *(ICER '16)*. Association for Computing Machinery, New York, NY, USA, 241–250. https://doi.org/10.1145/2960310.2960314

[47] Cindy E. Hmelo. 1998. Problem-based learning: Effects on the early acquisition of cognitive skill in medicine. *The Journal of the Learning Sciences* 7, 2 (1998), 173–208. https://doi.org/10.1207/s15327809jls0702_2

[48] Cindy E. Hmelo-Silver. 2004. Problem-based learning: What and how do students learn? *Educational Psychology Review* 16, 3 (2004), 235–266. https://doi.org/10.1023/B:EDPR.0000034022.16470.f3

[49] Cindy E. Hmelo-Silver, Ravit Golan Duncan, and Clark A. Chinn. 2007. Scaffolding and achievement in problem-based and inquiry learning: a response to Kirschner, Sweller, and. *Educational psychologist* 42, 2 (2007), 99–107.

[50] Benjamin D. Jee, David H. Uttal, Dedre Gentner, Cathy Manduca, Thomas F. Shipley, Basil Tikoff, Carol J. Ormand, and Bradley Sageman. 2010. Commentary: Analogical Thinking in Geoscience Education. *Journal of Geoscience Education* 58, 1 (2010), 2–13. https://doi.org/10.5408/1.3544291

[51] Yasmin Kafai, Chris Proctor, and Debora Lui. 2019. From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 101–109. https://doi.org/10.1145/3291279.3339400

[52] Manu Kapur. 2010. Productive failure in mathematical problem solving. *Instructional Science* 38, 6 (2010), 523–550.

[53] Manu Kapur. 2011. A further study of productive failure in mathematical problem solving: Unpacking the design components. *Instructional Science* 39, 4 (2011), 561–579.

[54] Manu Kapur. 2012. Productive failure in learning the concept of variance. *Instructional Science* 40, 4 (2012), 651–672.

[55] Manu Kapur. 2014. Comparing learning from productive failure and vicarious failure. *Journal of the Learning Sciences* 23, 4 (2014), 651–677.

[56] Manu Kapur. 2016. Examining productive failure, productive success, unproductive failure, and unproductive success in learning. *Educational Psychologist* 51, 2 (2016), 289–299.

[57] Manu Kapur and Katerine Bielaczyc. 2012. Designing for productive failure. *Journal of the Learning Sciences* 21, 1 (2012), 45–83.

[58] Elham Kazemi and Allison Hintz. 2014. *Intentional talk: How to structure and lead productive mathematical discussions*. Stenhouse.

[59] Cazembe Kennedy, Aubrey Lawson, Yvon Feaster, and Eileen Kraemer. 2020. Misconception-Based Peer Feedback: A Pedagogical Technique for Reducing Misconceptions. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education* (Trondheim, Norway) *(ITiCSE '20)*. Association for Computing Machinery, New York, NY, USA, 166–172. https://doi.org/10.1145/3341525.3387392

[60] Paul Kirschner, John Sweller, and Richard E. Clark. 2006. Why unguided learning does not work: An analysis of the failure of discovery learning, problem-based learning, experiential learning and inquiry-based learning. *Educational Psychologist* 41, 2 (2006), 75–86.

[61] Janet L. Kolodner, Paul J. Camp, David Crismond, Barbara Fasse, Jackie Gray, Jennifer Holbrook, Sadhana Puntambekar, and Mike Ryan. 2003. Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting learning by design into practice. *Journal of the Learning Sciences* 12, 4 (2003), 495–547.

[62] David R. Krathwohl. 2002. A revision of Bloom's taxonomy: An overview. *Theory Into Practice* 41, 4 (2002), 212–218.

[63] Kenneth J. Kurtz, Chun-Hui Miao, and Dedre Gentner. 2001. Learning by Analogical Bootstrapping. *Journal of the Learning Sciences* 10, 4 (2001), 417–446. https://doi.org/10.1207/S15327809JLS1004new_2

[64] Magdalene Lampert, Megan Loef Franke, Elham Kazemi, Hala Ghousseini, Angela Chan Turrou, Heather Beasley, Adrian Cunard, and Kathleen Crowe. 2013. Keeping It Complex: Using Rehearsals to Support Novice Teacher Learning of Ambitious Teaching. *Journal of Teacher Education* 64, 3 (2013), 226–243. Publisher: SAGE Publications Sage CA: Los Angeles, CA.

[65] Ard W. Lazonder and Ruth Harmsen. 2016. Meta-analysis of inquiry-based learning: Effects of guidance. *Review of Educational Research* 86, 3 (2016), 681–718.

[66] Cynthia Bailey Lee, Saturnino Garcia, and Leo Porter. 2013. Can Peer Instruction Be Effective in Upper-Division Computer Science Courses? *ACM Trans. Comput. Educ.* 13, 3, Article 12 (Aug. 2013), 22 pages. https://doi.org/10.1145/2499947.2499949

[67] Soohyun Nam Liao, William G. Griswold, and Leo Porter. 2017. Impact of Class Size on Student Evaluations for Traditional and Peer Instruction Classrooms. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education* (Seattle, Washington, USA) *(SIGCSE '17)*. Association for Computing Machinery, New York, NY, USA, 375–380. https://doi.org/10.1145/3017680.3017764

[68] Soohyun Nam Liao, Daniel Zingaro, Michael A. Laurenzano, William G. Griswold, and Leo Porter. 2016. Lightweight, Early Identification of At-Risk CS1 Students. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Melbourne, VIC, Australia) *(ICER '16)*. Association for Computing Machinery, New York, NY, USA, 123–131. https://doi.org/10.1145/2960310.2960315

[69] Katharina Loibl and Timo Leuders. 2019. How to make failure productive: Fostering learning from errors through elaboration prompts. *Learning and Instruction* 62 (2019), 1–10.

[70] Katharina Loibl, Ido Roll, and Nikol Rummel. 2017. Towards a theory of when and how problem solving followed by instruction supports learning. *Educational Psychology Review* 29, 4 (2017), 693–715.

[71] Lauri Malmi, Judy Sheard, Päivi Kinnunen, Simon, and Jane Sinclair. 2019. Computing Education Theories: What Are They and How Are They Used?. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 187–197. https://doi.org/10.1145/3291279.3339409

[72] Lauren Margulieux, Brian Dorn, and Kristin Searle. 2019. Learning Sciences for Computing Education. In *The Cambridge Handbook of Computing Education Research*, S. A. Fincher and A. V. Robins (Eds.). Cambridge University Press, Cambridge, Chapter 8, 208–230.

[73] Lauren E. Margulieux, Mark Guzdial, and Richard Catrambone. 2012. Subgoal-Labeled Instructional Material Improves Performance and Transfer in Learning to Develop Mobile Applications. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research* (Auckland, New Zealand) *(ICER '12)*. Association for Computing Machinery, New York, NY, USA, 71–78. https://doi.org/10.1145/2361276.2361291

[74] Richard E. Mayer. 2004. Should there be a three-strikes rule against pure discovery learning? *American Psychologist* 59, 1 (2004), 14.

[75] Mark A. McDaniel, Henry L. Roediger, and Kathleen B. McDermott. 2007. Generalizing test-enhanced learning from the laboratory to the classroom. *Psychonomic Bulletin & Review* 14, 2 (2007), 200–206. https://doi.org/10.3758/BF03194052

[76] Bruce M. McLaren, Deanne Adams, Kelley Durkin, George Goguadze, Richard E Mayer, Bethany Rittle-Johnson, Sergey Sosnovsky, Seiji Isotani, and Martin Van Velsen. 2012. To err is human, to explain and correct is divine: A study of interactive erroneous examples with middle school math students. In *European Conference on Technology Enhanced Learning*. Springer, 222–235.

[77] Bruce M. McLaren, Tamara van Gog, Craig Ganoe, Michael Karabinos, and David Yaron. 2016. The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. *Computers in Human Behavior* 55 (2016), 87–99.

[78] John R. Mergendoller, Nan L. Maxwell, and Yolanda Bellisimo. 2006. The Effectiveness of Problem-Based Instruction: A Comparative Study of Instructional Methods and Student Characteristics. *Interdisciplinary Journal of Problem-based Learning* 1, 2 (2006), 46–69. https://doi.org/10.7771/1541-5015.1026

[79] Sarah Michaels, M. C. O'Connor, Richard Sohmer, and Lauren Resnick. 2009. Guided Construction of Knowledge in the Classroom: Teacher, Talk, Task, and Tools. In *Transformation of Knowledge Through Classroom Interaction*, Baruch Schwarz, Tommy Dreyfus, and Rina Hershkowitz (Eds.). Routledge, 105–129.

[80] Sarah Michaels and Catherine O'Connor. 2015. Conceptualizing Talk Moves as Tools: Professional Development Approaches for Academically Productive Discussions. In *Socializing Intelligence Through Academic Talk and Dialogue*, Lauren Resnick, Christa Asterhan, and Sherice Clarke (Eds.). American Educational Research Association, 347–361.

[81] Wendy C. Newstetter. 2005. Designing Cognitive Apprenticeships for Biomedical Engineering. *Journal of Engineering Education* 94, 2 (2005), 207–213. https://doi.org/10.1002/j.2168-9830.2005.tb00841.x arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.2168-9830.2005.tb00841.x

[82] Wendy C. Newstetter. 2006. Fostering integrative problem solving in biomedical engineering: the PBL approach. *Annals of biomedical engineering* 34, 2 (2006), 217–225. https://doi.org/10.1007/s10439-005-9034-z

[83] Steven C. Pan and Timothy C. Rickard. 2018. Transfer of test-enhanced learning: Meta-analytic review and synthesis. *Psychological Bulletin* 144, 7 (2018), 710–756. https://doi.org/10.1037/bul0000151

[84] Dale Parsons and Patricia Haden. 2006. Parson's Programming Puzzles: A Fun and Effective Learning Tool for First Programming Courses. In *Proceedings of the 8th Australasian Conference on Computing Education - Volume 52* (Hobart, Australia) *(ACE '06)*. Australian Computer Society, Inc., AUS, 157–163.

[85] Elizabeth Patitsas, Michelle Craig, and Steve Easterbrook. 2013. Comparing and Contrasting Different Algorithms Leads to Increased Student Learning. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (San Diego, San California, USA) *(ICER '13)*. Association for Computing Machinery, New York, NY, USA, 145–152. https://doi.org/10.1145/2493394.2493409

[86] Leo Porter, Cynthia Bailey Lee, and Beth Simon. 2013. Halving Fail Rates Using Peer Instruction: A Study of Four Computer Science Courses. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. Association for Computing Machinery, New York,

NY, USA, 177–182. https://doi.org/10.1145/2445196.2445250
[87] Leo Porter, Saturnino Garcia, John Glick, Andrew Matusiewicz, and Cynthia Taylor. 2013. Peer Instruction in Computer Science at Small Liberal Arts Colleges. In *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education* (Canterbury, England, UK) *(ITiCSE '13)*. Association for Computing Machinery, New York, NY, USA, 129–134. https://doi.org/10.1145/2462476.2465587
[88] Leo Porter and Beth Simon. 2013. Retaining Nearly One-Third More Majors with a Trio of Instructional Best Practices in CS1. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education* (Denver, Colorado, USA) *(SIGCSE '13)*. Association for Computing Machinery, New York, NY, USA, 165–170. https://doi.org/10.1145/2445196.2445248
[89] George J. Posner, Kenneth A. Strike, Peter W. Hewson, William A. Gertzog, et al. 1982. Accommodation of a scientific conception: Toward a theory of conceptual change. *Science education* 66, 2 (1982), 211–227.
[90] Yizhou Qian and James D. Lehman. 2019. Using Targeted Feedback to Address Common Student Misconceptions in Introductory Programming: A Data-Driven Approach. *SAGE Open* 9, 4 (2019), 2158244019885136. https://doi.org/10.1177/2158244019885136
[91] J. Elizabeth Richey, Juan Miguel L. Andres-Bray, Michael Mogessie, Richard Scruggs, Juliana MAL Andres, Jon R. Star, Ryan S. Baker, and Bruce M. McLaren. 2019. More confusion and frustration, better learning: The impact of erroneous examples. *Computers & Education* 139 (2019), 173–190.
[92] Henry L. Roediger and Andrew C. Butler. 2013. *Retrieval Practice (Testing) Effect*. Vol. 1. SAGE Publications, Thousand Oaks, California. 661–662 pages. https://doi.org/10.4135/9781452257044.n246
[93] Henry L. Roediger and Jeffrey D. Karpicke. 2006. Test-Enhanced Learning: Taking Memory Tests Improves Long-Term Retention. *Psychological Science* 17, 3 (2006), 249–255. https://doi.org/10.1111/j.1467-9280.2006.01693.x
[94] Henry L. Roediger, Adam L. Putnam, and Megan A. Smith. 2011. *Ten Benefits of Testing and Their Applications to Educational Practice*. Psychology of Learning and Motivation, Vol. 55. Academic Press, New York. 1–36 pages.
[95] Ido Roll, Vincent Aleven, and Ken Koedinger. 2011. Outcomes and mechanisms of transfer in invention activities. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 33.
[96] Ido Roll, Vincent Aleven, and Kenneth R. Koedinger. 2009. Helping students know'further'-increasing the flexibility of students' knowledge using symbolic invention tasks. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*. Cognitive Science Society Austin, TX, 1169–1174.
[97] Ido Roll, Natasha G. Holmes, James Day, and Doug Bonn. 2012. Evaluating metacognitive scaffolding in guided invention activities. *Instructional Science* 40, 4 (2012), 691–710.
[98] Emilio Sánchez, Héctor García-Rodicio, and Santiago R. Acuna. 2009. Are instructional explanations more effective in the context of an impasse? *Instructional Science* 37, 6 (2009), 537–563.
[99] Henk G. Schmidt and Jos H. C. Moust. 2000. Factors affecting small-group tutorial learning: A review of research. In *Problem-Based Learning: A Research Perspective on Learning Interactions*, Dorothy Evensen and Cindy E. Hmelo (Eds.). Erlbaum, Mahwah, NJ, 19–52.
[100] Richard A. Schmidt and Robert A. Bjork. 1992. New Conceptualizations of Practice: Common Principles in Three Paradigms Suggest New Concepts for Training. *Psychological Science* 3, 4 (1992), 207–217. https://doi.org/10.1111/j.1467-9280.1992.tb00029.x
[101] Daniel L. Schwartz and Taylor Martin. 2004. Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and instruction* 22, 2 (2004), 129–184.
[102] Robert S. Siegler et al. 2002. Microgenetic studies of self-explanation. *Microdevelopment: Transition processes in development and learning* (2002), 31–58.
[103] Beth Simon and Quintin Cutts. 2012. Peer Instruction: A Teaching Method to Foster Deep Understanding. *Commun. ACM* 55, 2 (Feb. 2012), 27–29. https://doi.org/10.1145/2076450.2076459
[104] Beth Simon, Sarah Esper, Leo Porter, and Quintin Cutts. 2013. Student Experience in a Student-Centered Peer Instruction Classroom. In *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research* (San Diego, San California, USA) *(ICER '13)*. Association for Computing Machinery, New York, NY, USA, 129–136. https://doi.org/10.1145/2493394.2493407
[105] Norman J. Slamecka and Peter Graf. 1978. The generation effect: Delineation of a phenomenon. *Journal of Experimental Psychology: Human Learning and Memory* 4, 6 (1978), 592–604. https://doi.org/10.1037/0278-7393.4.6.592
[106] M. K. Smith, W. B. Wood, W. K. Adams, C. Wieman, J. K. Knight, N. Guild, and T. T. Su. 2009. Why Peer Discussion Improves Student Performance on In-Class Concept Questions. *Science* 323, 5910 (2009), 122–124. https://doi.org/10.1126/science.1165919
[107] Juha Sorva. 2013. Notional Machines and Introductory Programming Education. *ACM Trans. Comput. Educ.* 13, 2, Article 8 (July 2013), 31 pages. https://doi.org/10.1145/2483710.2483713
[108] Erin M. Sparck, Elizabeth L. Bjork, and Robert A. Bjork. 2016. On the learning benefits of confidence-weighted testing. *Cognitive Research: Principles and Implications* 1, 3 (2016), 1–10.
[109] Johannes Strobel and Angela Van Barneveld. 2009. When is PBL more effective? A meta-synthesis of meta-analyses comparing PBL to conventional classrooms. *Interdisciplinary Journal of Problem-based Learning* 3, 1 (2009), 4. https://doi.org/10.7771/1541-5015.1046
[110] Lasang Jimba Tamang, Zeyad Alshaikh, Nisrine Ait Khayi, Priti Oli, and Vasile Rus. 2021. A Comparative Study of Free Self-Explanations and Socratic Tutoring Explanations for Source Code Comprehension. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education* (Virtual Event, USA) *(SIGCSE '21)*. Association for Computing Machinery, New York, NY, USA, 219–225. https://doi.org/10.1145/3408877.3432423
[111] Howard Tanner and Sonia Jones. 2000. Scaffolding for Success: Reflective Discourse and the Effective Teaching of Mathematical Thinking Skills. *Research in Mathematics Education* 2, 1 (2000), 19–32.
[112] Ruthann C. Thomas, Christina R. Weywadt, Janis L. Anderson, Brenda Martinez-Papponi, and Mark A. McDaniel. 2018. Testing Encourages Transfer Between Factual and Application Questions in an Online Learning Environment. *Journal of Applied Research in Memory and Cognition* 7, 2 (2018), 252–260. https://doi.org/10.1016/j.jarmac.2018.03.007
[113] Christine D. Tippett. 2010. Refutation text in science education: A review of two decades of research. *International Journal of Science and Mathematics education* 8, 6 (2010), 951–970.
[114] Sigmund Tobias and Thomas M. Duffy. 2009. *Constructivist instruction: Success or failure?* Routledge.
[115] Dimitra Tsovaltzi, Bruce M. McLaren, Erica Melis, and Ann-Kristin Meyer. 2012. Erroneous examples: effects on learning fractions in a web-based setting. *International Journal of Technology Enhanced Learning* 4, 3-4 (2012), 191–230.
[116] David T. Vernon and Robert L. Blake. 1993. Does problem-based learning work? A meta-analysis of evaluative research. *Academic Medicine* 68, 7 (1993), 550–563. https://doi.org/10.1097/00001888-199307000-00015
[117] Grant Williams and John Clement. 2017. Co-Constructing Models in High School Physics: Comparing Degrees of Teacher and Student Participation in Whole Class Discussions. In *Annual Meeting of the National Association for Research in Science Teaching*.
[118] Mark Windschitl, Jessica Thompson, and Melissa Braaten. 2011. Ambitious Pedagogy by Novice Teachers: Who Benefits from Tool-Supported Collaborative Inquiry into Practice and Why?. *Teachers College Record* 113, 7 (2011), 1311–1360.
[119] Donald R. Woods. 1996. Problem-based learning for large classes in chemical engineering. In *New directions for teaching and learning. Bringing Problem-Based Learning to Higher Education: Theory and Practice*, LuAnn Wilkerson and Wim H. Gijselaers (Eds.). Number 68. Wiley, 91–99.
[120] Benjamin Xie, Dastyni Loksa, Greg L. Nelson, Matthew J. Davidson, Dongsheng Dong, Harrison Kwik, Alex Hui Tan, Leanne Hwa, Min Li, and Amy J. Ko. 2019. A theory of instruction for introductory programming skills. *Computer Science Education* 29, 2-3 (2019), 205–253.
[121] Iman YeckehZaare, Paul Resnick, and Barbara Ericson. 2019. A Spaced, Interleaved Retrieval Practice Tool That is Motivating and Effective. In *Proceedings of the 2019 ACM Conference on International Computing Education Research* (Toronto ON, Canada) *(ICER '19)*. Association for Computing Machinery, New York, NY, USA, 71–79. https://doi.org/10.1145/3291279.3339411
[122] Elaine H.J. Yew and Karen Goh. 2016. Problem-Based Learning: An Overview of its Process and Impact on Learning. *Health Professions Education* 2, 2 (2016), 75–79. https://doi.org/10.1016/j.hpe.2016.01.004
[123] Daniel Zingaro. 2014. Peer Instruction Contributes to Self-Efficacy in CS1. In *Proceedings of the 45th ACM Technical Symposium on Computer Science Education* (Atlanta, Georgia, USA) *(SIGCSE '14)*. Association for Computing Machinery, New York, NY, USA, 373–378. https://doi.org/10.1145/2538862.2538878