Received: 25 May 2024 | Accepted: 16 November 2024

DOI: 10.1111/bjet.13548

ORIGINAL ARTICLE

British Journal of Educational Technology BERA

Providing tailored reflection instructions in collaborative learning using large language models

Atharva Naik | Jessica Ruhan Yin | Anusha Kamath | Qianou Ma | Sherry Tongshuang Wu | R. Charles Murray | Christopher Bogart | Majd Sakr | Carolyn P. Rose

Language Technologies Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA

Correspondence
Atharva Naik, Carnegie Mellon University, Pittsburgh, PA 15213, USA.
Email: arnaik@andrew.cmu.edu

Funding information
National Science Foundation, Grant/Award Number: DSES 2222762

Abstract

The relative effectiveness of reflection either through student generation of contrasting cases or through provided contrasting cases is not well-established for adult learners. This paper presents a classroom study to investigate this comparison in a college level Computer Science (CS) course where groups of students worked collaboratively to design database access strategies. Forty-four teams were randomly assigned to three reflection conditions ([GEN] directive to generate a contrasting case to the student solution and evaluate their trade-offs in light of the principle, [CONT] directive to compare the student solution with a provided contrasting case and evaluate their trade-offs in light of a principle, and [NSI] a control condition with a non-specific directive for reflection evaluating the student solution in light of a principle). In the CONT condition, as an illustration of the use of LLMs to exemplify knowledge transformation beyond knowledge construction in the generation of an automated contribution to a collaborative learning discussion, an LLM generated a contrasting case to a group's solution to exemplify application of an alternative problem solving strategy in a way that highlighted the contrast by keeping many concrete details the same as those the group had most recently collaboratively constructed. While there was no main effect of condition on learning based on a content test, low-pretest student learned more from CONT than GEN, with NSI not distinguishable from the other two, while high-pretest students learned marginally more from the GEN condition than the CONT condition, with NSI not distinguishable from the other two.

KEYWORDS

automated code generation, dynamic support for collaborative learning, large language models

Practitioner notes

What is already known about this topic

- Reflection during or even in place of computer programming is beneficial for learning of principles for advanced computer science when the principles are new to students.
- Generation of contrasting cases and comparing contrasting cases have both been demonstrated to be effective as opportunities to learn from reflection in some contexts, though questions remain about ideal applicability conditions for adult learners.
- Intelligent conversational agents can be used effectively to deliver stimuli for reflection during collaborative learning, though room for improvement remains, which provides an opportunity to demonstrate the potential positive contribution of large language models (LLMs).

What this paper adds

- The study contributes new knowledge related to the differences in applicability conditions between generation of contrasting cases and comparison across provided contrasting cases for adult learning.
- The paper presents an application of LLMs as a tool to provide contrasting cases tailored to the details of actual student solutions.
- The study provides evidence from a classroom intervention study for positive impact on student learning of an LLM-enabled intervention.

Implications for practice and/or policy

- Advanced computer science curricula should make substantial room for reflection alongside problem solving.
- Instructors should provide reflection opportunities for students tailored to their level of prior knowledge.
- Instructors would benefit from training to use LLMs as tools for providing effective contrasting cases, especially for low-prior-knowledge students.

INTRODUCTION

In computer science (CS) pedagogy, the prevailing practice emphasises computer programming, a form of problem solving, as the primary learning activity. This emphasis is carried out even to the exclusion of reflection activities, though best practices from the learning sciences suggest a different prioritisation might be more beneficial (Ryoo, 2019). This paper challenges current practices for CS education by contributing findings supporting the value of reflection in CS learning as well as contributing findings related to the manner in which that reflection should be elicited from students. In particular, this work seeks to apply the literature regarding reflection on contrasting cases as a paradigm for enhancing learning in CS through reflection (Durkin et al., 2017; Keech & Muldner, 2024; Ma et al., 2023). Though the effectiveness of reflection on contrasting cases for learning is well established, the relative effectiveness of reflection either through student generation of contrasting cases or through provided contrasting cases is not well established for adult learners (Baker et al., 2012; Bego et al., 2023; Griffin et al., 2024). This paper contributes insights to address this gap and embeds the derived insights into the design of an AI-enabled intervention to support collaborative learning in Computer Science for adult learners.

A distinguishing characteristic of collaborative learning encounters that are valuable for learning is that they engage learners in knowledge transformation beyond simple knowledge telling (Scardamalia & Bereiter, 1987). With recent advances in Generative AI (GenAI) enabled through large language models (LLMs) (Vaswani et al., 2017), researchers have begun to ask whether the technology is capable of entering into knowledge transformation or supporting students in engaging in this knowledge transformation process (Cress & Kimmerle, 2023; Kasneci et al., 2023). Past work in computer-supported collaborative learning (CSCL) has already yielded principles for the design of AI-enabled interactive collaborative scaffolding (Rosé & Ferschke, 2016). In particular, for nearly two decades intelligent conversational agents have been employed to increase reflection and learning in CSCL settings through a variety of interactive strategies (Gweon et al., 2007; Kumar et al., 2007; Sankaranarayanan, Ma, et al., 2022; Tegos et al., 2015).

Past work on LLM-enabled learning support lays a foundation for exploration of this space through development of support for individual programmers, such as programming assistance for individual novice learners (Jayagopal et al., 2022, Kazemitabaar, Chow, et al., 2023; Kazemitabaar, Hou, et al., 2023). What LLMs offer is more options for adapting the specific content of reflection instructions using specific details of student work and discussion in context. From a technical perspective, this article contributes to that past work by illustrating how recent work on LLMs is able to provoke reflection that is valuable for student learning during collaborative work. In particular, building on the literature regarding reflection with contrasting cases (Durkin et al., 2017; Schwartz et al., 2011; Schwartz & Bransford, 1998), a novel intervention is evaluated in which an LLM is used to analyse a student problem solution and generate a contrasting case suitable for collaborative reflection subsequent to problem solving.

THEORETICAL FRAMEWORK

In situating this study in adult learning of Computer Science, this study builds on the literature of contrasting example study and problem solving. It delves into a specific form of example study, namely reflection on contrasting cases, which requires new scientific knowledge on application of this approach to adult learning in order to be utilised. In particular, results contrasting student generation of contrasting cases versus students reflecting on provided contrasting cases has been inconsistent across studies. This study seeks to bring clarity by investigating the role of prior knowledge in determining how best to engage learners in reflection on contrasting cases. In doing so, it also builds on literature countering the prevailing pedagogy in college level CS that privileges learning through computer programming. The study counters that stance by demonstrating the value of reflection over problem solving.

The value of reflection on examples in advanced CS

One of the best studied forms of reflection-rich pedagogy in STEM domains is the literature on example-based learning (Paas & Van Merriënboer, 1994; Renkl, 2014; Sweller et al., 1998; Tuovinen & Sweller, 1999). Extensive problem-solving practice has been identified as inferior to the use of worked examples for positively impacting student learning before they have some rudimentary understanding of foundational concepts to build from (Chi et al., 1989; Renkl, 2014; Sweller & Cooper, 1985; Van Gog et al., 2011). Student reflection is realised in the form of explanation generation (Rittle-Johnson et al., 2017; Wylie & Chi, 2014), which has been studied in numerous domains, including computer programming (Fabic et al., 2019). Activities similar to example study may be involved in some typical programming-related practices, such as code tracing (Lee & Muldner, 2020), which have been noted to be challenging for novice learners. Generating code explanations has also been noted as challenging for novice learners (Lahtinen et al., 2005; Murphy et al., 2012; Simon and Snowdon, 2011). Nevertheless, the suggestion that reflection-rich practices might replace time typically spent on problem-solving practice in the form of computer programming has remained controversial (Kalyuga et al., 2001) especially in advanced CS. Acknowledging findings regarding reflection-rich learning at early stages of skill acquisition, the bulk of past work on CS education utilises reflection-rich learning no further than the early portions of undergraduate CS curricula. The prevailing belief is that these techniques do not extend to adult learning of advanced CS. This belief is able to persist because very little work focuses on advanced computer science courses, though this is beginning to change (Sankaranarayanan et al., 2020).

Investigating the application of learning sciences principles for advanced CS raises questions about the extent to which principles from past work generalise to substantially more advanced content and a substantially older learner population. For example, learning from worked examples through focused reflection is most beneficial as skills and concepts are at the early stages of acquisition (Paas & Van Merriënboer, 1994; Renkl, 2014). Active authentic problem solving is more beneficial once a foundation is laid through exposure and reflection. When it comes to advanced CS courses, where the students have already reached some level of expertise, the predominant pedagogy privileges problem-solving in the form of computer programming, though some past work highlights the advantage with regards to conceptual learning in taking some time away from problem solving to make time for reflection (Sankaranarayanan, Kandimalla, et al., 2022). Recent studies (Sankaranarayanan, Kandimalla, et al., 2022) focusing specifically on learning from collaborative problem solving in CS education suggest that collaboration support that shifts the focus of students more toward reflection and less toward the actual coding increases conceptual learning without harming the ability to write code in subsequent programming assignments. These past studies focused on manipulating the placement of the reflection instructions over the progression of activities within the session or the proportion of time dedicated to reflection versus programming. This study follows best practices learned from those past studies and pushes further to investigate how the form of reflection activities effects the value of the reflection in advanced CS instruction.

Contrasting cases as a paradigm for example study

Learning from worked examples hinges on being able to draw students' attention to the relevant problem states while helping them navigate away from superfluous ones (Kalyuga et al., 2001). In prior work, this has been achieved by using various means such as classification of examples by common schema (Tuovinen & Sweller, 1999), contrasting cases

(Durkin et al., 2017; Ma et al., 2023; Schwartz & Bransford, 1998), and prompt-directed self-explanation (Sidney et al., 2015). In CS where there are typically multiple viable solution paths, each with their own valuable trade-offs to consider and learn from, it is important not only to focus student attention on problem-solving paths that lead to a viable solution but also expose them to multiple such paths. Similarly, principles of software design can be applied in different ways, and in fact, lead to different choices depending upon the particulars of a scenario. For example, the most efficient indexing strategies for a database depend on the database schema and what types of queries are most frequent. To learn to apply these principles requires development of a generalisable framework for them that would enable making different choices about the application as is appropriate given the scenario. Investigations of example study have demonstrated that comparison of contrasting examples can be particularly valuable for fostering flexible application of principles (Durkin et al., 2017; Keech & Muldner, 2024; Ma et al., 2023).

In order to draw attention to the most important contrasts in cases being compared, example pairs need to be designed such that extraneous contrasts are not present as distractors (Roelle & Berthold, 2016). Thus, where the value of contrasting cases has been studied in connection with computer programming, the contrasts have been very local, such as the form of implementation for a loop in elementary programming instruction (Ma et al., 2023). In order to foster flexible application of principles that lead to different choices depending upon the particulars of a scenario, our work focuses instead on contrasts that are broader, pertaining to overarching design principles rather than minute details of local technical choices. Research is needed to evaluate the generality of past findings to this more challenging context. In this study, we explore the use of contrasting cases for fostering the ability to apply principles flexibly in more advanced CS as an evaluation of the generality of the more restrictive past work.

Advanced learners have different needs than novice learners and in particular may not need the scaffolding provided by examples that are beneficial for novice learners to reflect on (Paas et al., 2003). In that case, if scaffolding is unnecessary it might even hamper the productive solution space exploration advanced learners are capable of engaging in, which may apply also with reflection on contrasting cases. However, a more challenging engagement with contrasting cases may come in the form of requiring learners to generate their own contrasting cases. Recent work provides evidence from learning evaluations and self-report of the benefits of students generating examples and/or contrasting cases across a variety of STEM domains (Baker et al., 2012; Bego et al., 2023; Griffin et al., 2024). However, while computer programming itself could be conceptualised as an example generation task, questions remain about how to apply past findings about generation of contrasting cases in an advanced CS context. Thus, in our study, we include this more challenging condition in which students are required to generate the contrasting case for reflection themselves.

Hypotheses and study design

This study seeks empirical evidence regarding the most effective approach for learning from contrasting cases for adult learners. In particular, we experimentally compare two alternative forms of contrasting cases study, one of which requires students to generate their own contrasting cases versus one where the contrasting cases are constructed for them using an LLM. We further contrast both of those with a control condition featuring a generic instruction for students to reflect on their solution. More specifically, the three conditions thus include: [GEN] a directive to generate a contrasting case to the student solution and evaluate their trade-offs in light of the principle, [CONT] a directive to compare the student solution with a provided contrasting case and evaluate their trade-offs in light of a principle and

[NSI] a control condition with a non-specific directive for reflection evaluating the student solution in light of a principle. In the CONT condition, as an illustration of the use of LLMs to exemplify knowledge transformation beyond knowledge construction in the generation of an automated contribution to a collaborative learning discussion, an LLM generates a contrasting case to a group's solution to exemplify application of an alternative problem solving strategy in a way that highlights the contrast by keeping many concrete details the same as those the group had most recently collaboratively constructed. Just as worked examples provide concrete guidance for what technical choices to reflect on, we hypothesise that students will benefit from reflection on principles with the aid of scaffolding for reflection, and that the level of scaffolding required to achieve the positive effect will depend upon the level of prior knowledge within the same advanced technical content, with CONT providing more scaffolding than GEN, and both CONT and GEN being more demanding than NSI. We thus test four hypotheses:

- **H1 Reflection benefit:** As students actively respond to reflection opportunities across conditions, their reflection will be associated with increased learning.
- **H2 Contrasting case exploration:** Reflection opportunities presenting specific contrasting scenarios (ie, CONT in comparison with NSI and GEN) will broaden reflection on the application of principles.
- **H3 Contrasting case value:** Student learning benefits from reflection on contrasting scenarios (ie, CONT and GEN in comparison with NSI).
- **H4 Contrasting reasoning support:** Students with less prior knowledge regarding the space of possible applications of principles will require more support to actively contribute reflection on contrasting scenarios (ie, CONT vs. GEN).

TECHNICAL FOUNDATION FROM PAST WORK

Since its December 2022 launch, ChatGPT has emerged as a leading GenAI technology. Its accessibility has sparked innovations and debates on its role in education (Dai et al., 2023). For students, its ability to process cross-domain knowledge is appealing (Stokel-Walker & Van Noorden, 2023). Teachers, on the other hand, see both benefits—for example, support for content creation (Young & Shishido, 2023) and personalised tutoring (Stamper et al., 2024)—as well as risks—for example enabling plagiarism (Cotton et al., 2024) and dissemination of biased information (Sok & Heng, 2023). While the terms Generative AI (GenAI) and large language models (LLM) are sometimes used interchangeably, GenAI refers to a broader class of modelling techniques that includes large language models (more details about GenAI vs. LLMs can be found in the Appendix). In this paper, we use GPT-4, an LLM as a tool for enabling one form of intervention in our study by creating context-specific, personalised contrasting cases to support reflection. While the media argues that LLMs hold the potential for broad transformation of education, we argue for identification of specific opportunities where the capabilities of LLMs meet concrete needs. We also describe where our work falls in the realm of pair programming with AI in the Appendix.

In this work, we motivate the design of an intervention from learning sciences principles and questions that leads to an identified technical need: namely, construction of a problem solution that incorporates surface similarities with a student constructed solution but represents an application of a different strategy at a deeper level. Real time analysis of a collaborative problem solving process to identify a principle-based strategy, construction of an alternative strategy, and instantiating a generated generic solution with details extracted from the problem solving process are all technical problems that are easier to solve with LLMs than earlier forms of Artificial Intelligence.

LEARNING ACTIVITY DESIGN

Collaborative learning is most valuable for learning activities with multiple possible solution paths, and selecting a path is less about finding the right answer than evaluating complex sets of constraints and trade-offs (Cress et al., 2021; Koschmann, 2017). Amid these activities, students benefit from exposure to each other's alternative points of view. Advanced CS topics are ripe with opportunities for evaluation of design trade-offs. We select SQL database and query optimisation, a topic with important design trade-offs. We select Mob programming (Buchan & Pearl, 2018) as the paradigm for orchestrating the collaboration as it encourages sharing and challenging alternative perspectives.

SQL design activity

Database design offers a solution space with interesting trade-offs. The optimisation task involves using techniques like datatype modification, index creation and table joining (denormalisation) for a given scenario or query load, to minimise query cost while satisfying a few constraints. The rubric dimensions used to evaluate solutions are *data retrieval efficiency, write performance, disk storage* and *maintainability*. The students get primers related to the three optimisation techniques. More details about the rubric dimensions and primers can be found in the Supplementary Material. The learning activity requires students to apply the knowledge learned from the primers as well as integrate multiple optimisation techniques and reason about their impact on the rubric dimensions. We also describe the paradigm of Mob Programming in the Appendix.

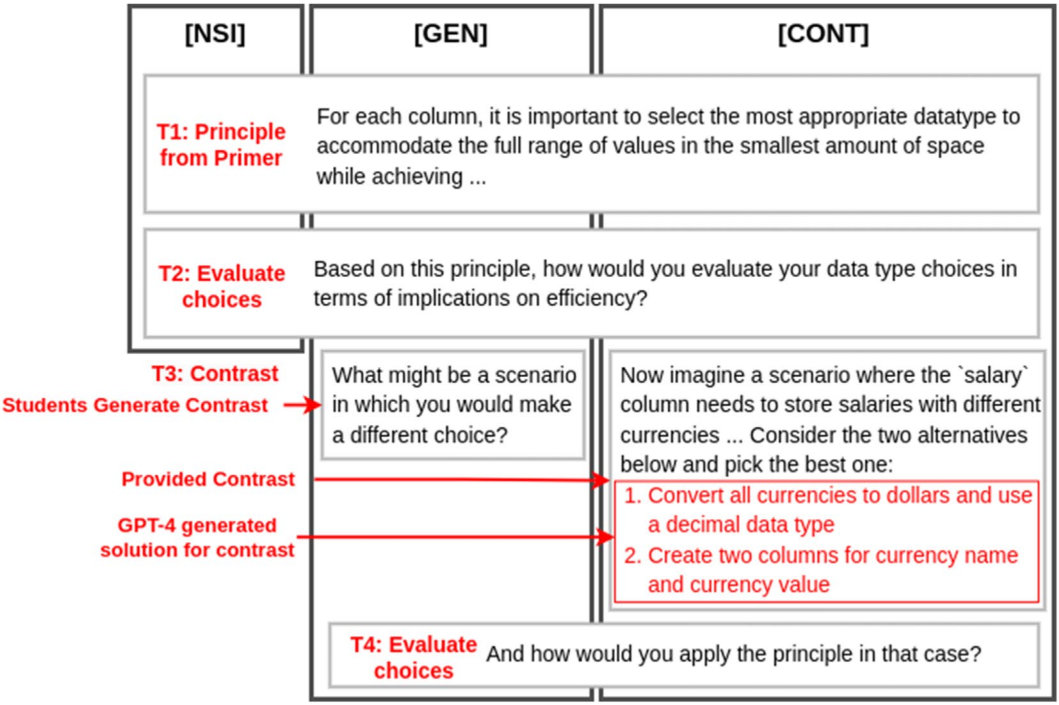
METHOD

Experimental procedure

The study was run in a collaborative database design assignment in an advanced CS course called Cloud Computing. To facilitate information sharing within groups, all students received reading material in the form of primers before the activity, with each student being responsible for reading at least one primer but having access to the others. At the beginning of the activity, each student individually took the pre-test, with 27 conceptual items organised into seven multi-part questions designed to test the individual learning objectives as well as their integration.

The groups then engaged in the design activity for 80 minutes. In all three conditions, the OPE_bot based on the Bazaar CSCL architecture (Kumar et al., 2009) played the role of a facilitator, providing the same task instructions, following the same role assignment strategy at the onset of each subtask, and providing the same task-relevant announcements at key times. In all conditions, there was a reflection phase after the completion of each of the three tasks to allow for reflection on design choices for the task before switching to the next one. *The 3-condition experimental manipulation* (*illustrated in* Figure 1) *took place during the reflection phase*.

Finally, the students individually took the posttest, which was identical to the pre-test. Students never received feedback on their performance on the test.


[NSI]
[GEN]
[CONT]
T1: Principle from Primer
T2: Evaluate choices
T3: Contrast
Students Generate Contrast
Provided Contrast
GPT-4 generated solution for contrast
T4: Evaluate choices

For each column, it is important to select the most appropriate datatype to accommodate the full range of values in the smallest amount of space while achieving ...

Based on this principle, how would you evaluate your data type choices in terms of implications on efficiency?

What might be a scenario in which you would make a different choice?

Now imagine a scenario where the `salary` column needs to store salaries with different currencies ... Consider the two alternatives below and pick the best one:

1. Convert all currencies to dollars and use a decimal data type
2. Create two columns for currency name and currency value

And how would you apply the principle in that case?

FIGURE 1 Format of the reflections for each of the conditions. [NSI] is constituted by T1 and T2, while [GEN] and [CONT] are made up of T1, T2, T3 and T4 with [GEN] having the generic alternative scenario turn on the right and [CONT] having the GPT-4 generated tailored alternatives shown on the left in T3.

Experimental design

In order to test the four hypotheses (see Section 'Hypotheses and study design'), namely, H1 Reflection benefit, H2 Contrasting case exploration, H3 Contrasting case value and H4 Contrasting reasoning support, we conducted an intervention study investigating the relative value of student reflection in response to different instructions. In particular, the intervention study featured three conditions that differed in terms of the instructions for reflection that were inserted between problem solving episodes, which were referred to as [GEN] (featuring instruction to generate a contrasting case to the student solution and evaluate their trade-offs in light of the principle), [CONT] (featuring instruction to compare the student solution with a provided contrasting case and evaluate their trade-offs in light of a principle) and [NSI] (a control condition with a non-specific directive for reflection evaluating the student solution in light of a principle). Thus, we manipulate a reflection engagement type [independent variable] in order to improve learning [dependent variable] by impacting the extent to which students engage in reflection that is valuable for learning [process variable]. To test H1, we test for a correlation between process and dependent variable. To test H2, we test for a significant difference in process associated with the independent variable. To test H3, we test for a significant difference in dependent variable associated with the independent variable. To test H4, we test for a significant interaction between a median split on prior knowledge and the independent variable on the dependent variable. Analysis details are found in Section 'Method'.

Intervention design

As illustrated in Figure 1, the [NSI] reflection involved 2 turns while both the [GEN] and [CONT] reflections involved 4 turns with all the reflections sharing the first 2 turns containing a principle from the primer to remind students of the relevant trade-offs involved so they are able to evaluate their design choices based on it (eg, for the data type primer, the principle tackles the issue of storage requirements, performance of the operations to be supported and capturing the seen range of values in the data). What the [GEN] condition adds in addition to what the [NSI] condition provides is two tutor turns that ask the student to generate a contrasting case and compare it to their original solution. What [CONT] adds instead is the presentation of a specific contrasting case and then a request for the students to compare it to their original solution.

We describe the technical details of implementing the intervention and how we prompt GPT-4 to construct the contrasting cases for the [CONT] condition as well as the model version and dates of the study in the Supplementary Material.

RESULTS

The evaluation of our four hypotheses required measures of learning (using a pre-posttest) and visible measures of reflection (using a textual analysis of chatlogs). We begin with the analysis of learning gains via pre- and posttest (to test H3 and H4) in Section ‘Learning gains’ and then proceed with the chatlog analysis (to test H1 and H2) in Section ‘Textual discussion analysis’.

One hundred and thirty students in the Cloud Computing class were arranged into 44 teams. At the request of the instructor, students were allowed to form their own teams, which worked together throughout the semester on a series of activities (42 of 3 and 2 teams of 2). The study took place during a specific unit related to database design. For the study, teams were randomly assigned to three conditions, namely NSI, GEN and CONT. Ten students did not submit the pretest, posttest or both and were thus dropped from the learning gains analysis. These students were roughly evenly distributed across conditions such that the final set of students 120 students used in the analysis included 37 from NSI (in 14 teams), 43 students in GEN (in 15 teams) and 40 students in CONT (in 15 teams).

Learning gains

The 27 item pre-posttest was divided into six different learning objectives, with three topic areas and two difficulty levels: namely, data types (simple and complex), indexing (simple and complex) and normalisation/denormalisation (simple and complex). In order to equally value each learning objective, we computed a normalised score by averaging test question scores within these learning objective sets such that students received a score between 0 and 100% for each set of questions. We then averaged across learning objectives to compute a combined score that equally valued each learning objective. Using a median split over the combined pretest scores, we also split students into a high-pretest group and a low-pretest group (Table 1). A Chi-square test confirmed that there was no significant difference in the distribution of students to high versus low pretest groups across conditions: $F(2, 120)=0.367$, $p=$N.S. To prepare for the analysis we checked the distribution of the dependent variable to test for normality using the Anderson-Darling test provided in the JMP Analyse Distribution tab. For all cells in a condition (NSI/Gen/Cont) by split (high pretest vs. low pretest), the p-value was greater than 0.05. Across these six cells we also tested for

TABLE 1 Summary statistics for combined pretest and posttest scores within condition and high versus low pretest split.

	Condition	Split	Mean	Median	Standard deviation	Standard error
Combined pretest	NSI	High-pretest	72.04	71.48	9.04	2.19
		Low-pretest	47.59	48.98	10.85	2.43
	Gen	High-pretest	79.96	73.66	5.04	1.08
		Low-pretest	39.74	36.57	13.82	3.02
	Cont	High-pretest	73.63	71.67	5.88	1.28
		Low-pretest	47.31	49.81	10.01	2.30
Combined posttest	NSI	High-pretest	76.49	78.98	11.03	2.67
		Low-pretest	69.65	69.91	14.05	3.14
	Gen	High-pretest	78.09	77.92	9.92	2.12
		Low-pretest	62.76	53.80	18.17	3.96
	Cont	High-pretest	72.65	72.6	11.62	2.54
		Low-pretest	75.23	79.17	14.66	3.36

unequal variance using an O'Brian, Brown-Forsythe and Bartlett tests on the JMP Fit X by Y tab, and in all of these cases, the *p*-value was higher than 0.05. We conclude that we can analyse student learning effects with ANOVAs and ANCOVAs at this level of aggregation of our data.

Hypotheses 3 and 4 both focus on pre to posttest learning gains.

- **H3 Contrasting case value:** Student learning benefits from reflection on contrasting scenarios (ie, CONT and GEN in comparison with NSI).
- **H4 Contrasting reasoning support:** Students with less prior knowledge regarding the space of possible applications of principles will require more support to actively contribute reflection on contrasting scenarios (ie, CONT vs. GEN).

We first verified that students learned during the activity prior to testing for differences across experimental groups. We computed a 2×2×3 ANOVA with the combined posttest score as the dependent variable and with phase (whether pre or post), split (high pretest vs. low pretest) and condition (NSI vs. Gen vs. Cont) as independent variables. In order to test whether students learned in all three conditions within both high and low pretest groups, we also included all pairwise and three-way interaction terms in the model.

The model estimates and other details are found in Table 2. The significant effect of phase and lack of significant interaction between phase and split and between phase, split, and condition indicates that students learned between the pretest phase and posttest phase in all conditions regardless of being in the high or low pretest groups. The effect size of 0.26 indicates a large effect. As a caveat, a student-*t* posthoc analysis on the significant effect of the phaseXsplit interaction term reveals that the pretest scores of high pretest students fell in between the posttest scores of high pretest students and the posttest scores of low pretest students, not being statistically distinguishable from either. The lack of significant effect of condition as well as the condition by phase interaction term suggests that there is no main effect of condition on learning, but we further test that with an ANCOVA below. The significant effect of split indicates that students in the high pretest group had higher scores than students in the low pretest group, which is expected. Overall, the evidence suggests that students learned across conditions and high versus low pretest groups.

TABLE 2 Model estimates for 2×3×3 ANOVA with the combined posttest score as the dependent variable and with phase (whether pre or post), split (high pretest vs. low pretest) and condition (NSI vs. GEN vs. CONT) as independent variables.

	Sum of squares	*df*	Mean square	*F*	*p*	Partial eta-squared
Overall model	3.92	11	0.36	25.96	0.0001	
Phase (2)	1.08	1	1.08	78.24	0.0001	0.26
Condition (3)	0.06	2	0.03	2.13	0.12	0.02
Split (2)	1.81	1	1.81	131.72	0.0001	0.32
Phase × Split (2×2)	0.71	1	0.71	51.55	0.0001	0.18
Phase × Condition (2×3)	0.00	2	0.00	0.00	0.99	0.00
Split × Condition (2×3)	0.18	2	0.09	6.66	0.0015	0.05
Phase × Split × Condition (2×2×3)	0.04	2	0.02	1.38	0.25	0.01
Residuals	3.13	228	0.01			

TABLE 3 Model estimates for 2×3 ANCOVA with the combined posttest score as the dependent variable, combined pre-test score as a covariate and split (high pretest vs. low pretest) and condition (NSI vs. GEN vs. CONT) as independent variables.

	Sum of squares	*df*	Mean square	*F*	*p*	Partial eta-squared
Overall model	0.49	6	0.08	4.81	0.0002	
Pretest	0.17	1	0.17	9.8	0.002	0.08
Condition (3)	0.01	2	0.005	0.26	0.77	0.01
Split (2)	0.02	1	0.02	1.23	1.23	0.01
Split × Condition (2×3)	0.11	2	0.06	3.35	0.05	0.05
Residuals	1.93	113	0.02			

Next, we formally tested for differences in learning across experimental groups in order to test H3 and H4. Specifically, we computed a 3×2 ANCOVA model with combined posttest score as the dependent variable, the condition and learning objective as independent variables, and the pre-test score as a covariate. We also included the pairwise interaction terms and the three-way interaction term. There was no significant main effect of condition. Thus, H3 in its simplest version was not supported. However, it is consistent with the literature on reflection on contrasting cases to expect that the benefit of the GEN and CONT conditions may vary depending upon preparation, which is what H4 addresses. To test H4, we pay attention to the analysis of the split variable as well as the two-way interaction term. We found a significant interaction between condition and split. Based on a Student-t post hoc analysis without adjustment for multiple comparisons, GEN was marginally better than CONT for high pretest students whereas CONT was significantly better than GEN for low pretest students. In both cases, NSI fell in between, not being significantly different from either. Thus, H4 is supported, and H3 is partially supported. The details of the ANCOVA are found in Table 3. The interaction is displayed in Figure 2, Its effect size is 0.05, which is a small to medium effect.

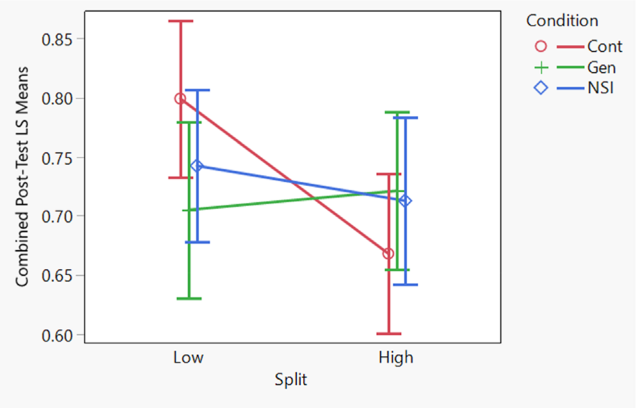


FIGURE 2 This figure displays the interaction between condition and split on the least squares mean of combined posttest scores. A simple *t*-test shows that the difference between GEN and CONT is significant both for low pretest students and marginal for high pretest students, though the ranking is opposite.

Textual discussion analysis

To understand the effect of the intervention better, we performed an analysis of the chat, thus enabling us to test H1 and H2. Beyond investigating how the chats across conditions proceeded, we were specifically interested in the extent to which any effect on discussion behaviour may have mediated or moderated the effect of condition on learning, especially for low pre-test students.

Discussion coding

To prepare for this analysis, for each of the three tasks, we extracted the portions of the chat log that occurred between the time when reflection directives were administered and the end of the task. We then coded each chat message as off-topic (OFF_TOPIC), on-topic and focusing on the current solution (CURR), or on-topic and focusing on a contrasting/alternative scenario (ALT) as described in Table 1 in the Supplementary Material. SUM is the total of CURR and ALT. For our analysis, we focus specifically on the reflective contributions to the discussion. Table 4 displays the descriptive statistics for CURR, ALT, and SUM for high pretest students and low pretest students in each condition.

In order to test H2, we measure a relationship between the independent variables (condition and split) and process measures. In particular, we tested for differences in discussion behaviour across conditions as well as specifically within low and high pretest groups. The distribution of contribution did not conform to a normal distribution, so we use a Wilcoxon test for pairwise differences between condition and split. Thus, for each process variable (CURR, ALT and SUM), we perform pairwise tests between conditions within the high and low pretest groups. Since there are six comparisons per process variable, using a Bonferroni correction for multiple comparisons, we use 0.008 as a threshold for significance and 0.02 for a marginal effect. The model estimates and effect sizes are found in Table 5. Here we discuss the significant pairwise contrasts. For low pretest students, students in the CONT condition contributed significantly more ALT reflections than students in the NSI condition. However, students in the NSI condition contributed significantly more CURR contributions than students in the CONT condition. For high pretest students, students in CONT contributed significantly more ALT contributions than students in the NSI condition, and students in

TABLE 4 Summary statistics for discussion contributions by type (CURR, ALT and SUM) within condition and high versus low pretest split.

	Condition	Split	Mean	Median	Standard deviation	Standard error
CURR	NSI	High-pretest	1.24	1	1.20	0.29
		Low-pretest	1.7	2	1.03	0.23
	GEN	High-pretest	1.18	1	1.10	0.23
		Low-pretest	1.29	0	1.68	0.37
	CONT	High-pretest	0.62	0	0.92	0.20
		Low-pretest	0.63	0	0.90	0.21
ALT	NSI	High-pretest	0	0	0	0
		Low-pretest	0	0	0	0
	GEN	High-pretest	0.45	0	0.91	0.19
		Low-pretest	0.38	0	1.16	0.25
	CONT	High-pretest	1.10	0	1.37	0.30
		Low-pretest	0.47	0	0.77	0.18
SUM	NSI	High-pretest	1.23	1	1.20	0.29
		Low-pretest	1.7	2	1.03	0.23
	GEN	High-pretest	1.64	1	1.71	0.36
		Low-pretest	1.67	1	2.33	0.51
	CONT	High-pretest	1.71	2	1.74	0.38
		Low-pretest	1.11	1	1.05	0.24

the GEN condition contributed significantly more ALT contributions than students in the NSI condition. The overall trend was for more total reflection (SUM) in the two contrasting cases conditions, despite Low Pretest students contributing more CURR reflections than those in GEN. There were no significant effects on SUM for any contrast. H2 was only partially and weakly supported in terms of reflection on the alternative contrasting case, whether it was provided to the students or they generated it themselves.

Moderation analysis

Testing H1 requires evaluating a relationship between the process measure and student learning. The aim of the intervention was to promote reflection on contrasting cases in order to enhance learning.

We begin with an analysis of the moderating effect of total reflection on learning by adding total reflection as a covariate in the learning gains 3×2 ANCOVA model. The model estimates and effect sizes for this new model are displayed in Table 6. Total reflection shows a significant relationship with the dependent variable. The presence of this additional variable does not substantially change any of the other effects in the model. Consistent with expectations, more reflection is associated with more learning. The main differentiator across conditions is the split between reflection on the current solution versus a contrasting solution, however a follow up analysis using ALT instead of SUM did not produce a significant effect. Thus, there is some evidence of the value of reflection in connection with learning (ie, H1 is supported). However, though intervention had an effect on the focus of the reflection, there was no demonstrated effect of that shift. Therefore, the effect of learning is not explained by a measured effect on reflection behaviour.

TABLE 5 Pairwise Wilcoxon nonparametric tests to compare average counts of CURR, ALT and SUM between conditions within high pretest and low pretest groups. Using a Bonferroni correction for multiple comparisons, we use 0.008 as a threshold for significance and 0.02 for a marginal effect since we do six comparisons per dependent variable.

Dependent variable	Condition level	Condition level	Split	Score mean difference	Standard error difference	*Z*	*p*	*N*	*r*
CURR	CONT	−GEN	High pretest	−6.65	3.58	−1.86	0.96	43	0.28
			Low pretest	−3.06	3.36	−0.91	0.36	40	0.14
	GEN	−NSI	High pretest	−0.15	3.54	−0.04	0.96	39	0.01
			Low pretest	−5.13	3.63	−1.41	0.16	41	0.22
	CONT	−NSI	High pretest	−5.38	3.32	−1.62	0.1	38	0.26
			Low pretest	−10.78	3.51	−3.07	0.002	39	0.49
ALT	CONT	−GEN	High pretest	5.26	3.31	1.59	0.11	43	0.24
			Low pretest	3.11	2.70	1.15	0.25	40	0.18
	GEN	−NSI	High pretest	5.27	2.31	2.28	0.02	39	0.37
			Low pretest	2.88	1.69	1.70	0.09	41	0.27
	CONT	−NSI	High pretest	8.99	2.81	3.21	0.001	38	0.52
			Low pretest	6.11	2.29	2.67	0.008	39	0.43
SUM	CONT	−GEN	High pretest	0.56	3.72	0.15	0.88	43	0.02
			Low pretest	−0.45	3.53	−0.13	0.90	40	0.02
	GEN	−NSI	High pretest	1.72	3.55	0.48	0.63	39	0.08
			Low pretest	−3.90	3.64	−1.07	0.28	41	0.17
	CONT	−NSI	High pretest	2.55	3.47	0.74	0.46	38	0.12
			Low pretest	−6.06	3.53	−1.72	0.08	39	0.28

TABLE 6 Model estimates for 2×3 ANCOVA with the combined posttest score as the dependent variable, combined pre-test score and SUM as covariates and split (high pretest vs. low pretest) and condition (NSI vs. GEN vs. CONT) as independent variables.

	Sum of squares	*df*	Mean square	*F*	*p*	Partial eta-squared
Overall model	0.57	7	0.08	4.94	0.0001	
Pretest	0.16	1	0.16	9.65	0.002	0.08
SUM	0.08	1	0.08	4.76	0.05	0.04
Condition (3)	0.01	2	0.01	0.40	0.67	0.01
Split (2)	0.02	1	0.02	1.17	0.28	0.01
Split×Condition (2×3)	0.13	2	0.07	4.08	0.02	0.07
Residuals	1.85	112	0.02			

DISCUSSION AND CONCLUSIONS

This paper offers insights into how to offer effective reflection opportunities related to studying examples for adult learners in the CS domain while also serving as a demonstration of capabilities of LLMs as a tool for enabling a tailored realisation of contrasting cases study in this context. In particular, it investigates the role of prior knowledge in determining the best approach for engaging learners in reflection on contrasting cases. At a more detailed level, the study was designed to test four hypotheses:

- **H1 Reflection benefit:** As students actively respond to reflection opportunities across conditions, their reflection will be associated with increased learning.
- **H2 Contrasting case exploration:** Reflection opportunities presenting specific contrasting scenarios (ie, CONT in comparison with NSI and GEN) will broaden reflection on the application of principles.
- **H3 Contrasting case value:** Student learning benefits from reflection on contrasting scenarios (ie, CONT and GEN in comparison with NSI).
- **H4 Contrasting reasoning support:** Students with less prior knowledge regarding the space of possible applications of principles will require more support to actively contribute reflection on contrasting scenarios (ie, CONT vs. GEN).

In the analysis, we first addressed H3 and H4, which focus on pre-to-posttest learning gains. While many results supporting the value of reflection on contrasting cases for learning exist, important questions have remained regarding whether students should be required to generate their own contrasting cases or whether they should be provided to them. Results have been inconsistent across studies, and the bulk of published studies have focused on children rather than adults, the more advanced intellectual skills of adult learners could be expected to enable them to engage more readily in generation of contrasting cases, which raises questions of the generalisability of findings from studies of primary and secondary school learners to adult learners. The results from evaluating H3 and H4 address these gaps. The results of the study support the value of contextually tailored reflection directives related to concrete contrasting cases, particularly for low pretest students. The ability to generate such contrasting cases in real time is a novel technical achievement. Without this, it would be challenging to apply the findings in practice. That supports the value of the technical contribution of this work, namely a prompt engineering approach to intervention development in which an LLM enabled the generation of tailored contrasting cases for reflection scaffolds that were able to draw from details of the ongoing discussion between students as they worked together on a database design task.

With the abundant literature supporting the value of reflection on contrasting cases, it might be considered somewhat surprising that H3 was not supported in terms of main effect of Condition. However, the results can be explained in light of the adult and relatively advanced student population by considering that contrasting cases are a form of worked example study, which is known to have particular applicability at early stages of skill acquisition. Generation of examples is more challenging. The finding is not only that high pretest students are able to generate their own contrasting cases, there is evidence that the added challenge is a desirable difficulty (Bjork & Bjork, 2020), and even that not requiring that of them might actually hamper their learning experience. Thus, we have evidence supporting a nuanced view from the significant interaction effect between condition and split, challenging the broad applicability of contrasting case study for adult learners in contrast to proposals from earlier studies (Durkin et al., 2017; Roelle & Berthold, 2016). Relatedly, the fourth hypothesis was supported (H4 Contrasting reasoning support). In this study, students never spontaneously offered reflection on a contrasting case when it was not specifically requested of them (eg, we see 0 counts for this form of reflection in the NSI condition). For low pretest students, we only see increasing in ALT reflection in the CONT condition where it was scaffolded. The conclusion is that the personalisation enabled by the novel LLM-enabled intervention is more valuable to low-pretest students, whereas high pretest students are able to benefit to some extent from generating contrasting cases themselves, although the effect is small. Thus, it is important to adjust the application of LLM capabilities in instructional interventions with consideration of the capability level of students so that the intervention acts as a scaffold and not a crutch.

H1 and H2 focus on analysis of process variables, examining the substance of the reflection elicited by the intervention rather than focusing purely on evidence of learning from tests. The hypothesis specifically focusing on the effect of the manipulation on the substance of reflection was H2 Contrasting Exploration: Reflection instructions soliciting contrasting scenarios will broaden reflection on the application of principles. The past literature on contrasting cases suggests that contrasting cases produce better reflection. We asked the complementary question of whether it is better because there is more of it. H2 was only partly and weakly supported in our data. There was a nonsignificant trend for the two contrasting cases conditions to elicit more reflection overall. There was evidence for a significant shift from reflection on the student solution to reflection on the contrasting case in the two contrasting cases conditions. Thus, the intervention successfully drew student attention away from their current solution and toward contrasting scenarios and how they would alter their solution as evidenced by an analysis of the text they generated during the reflection. This shows the success of a novel technical approach to catalysing a particular form of reflection demonstrated to have value for student learning, which is also consistent with expectations from past literature.

The reason for the interest in the effect of the manipulation of reflection behaviour was because of the hypothesised relationship between reflection and learning. The evaluation of pre-to-posttest gains showed an effect of condition. An analysis of process variables offers insights on how the learning took place. The final hypothesis was H1 Reflection Benefit: As students actively respond to reflection instructions, their reflection will be associated with increased learning. Given the vast literature on learning from reflection (Chi et al., 1989; Renkl, 2014; Rittle-Johnson et al., 2017; Sweller & Cooper, 1985; Van Gog et al., 2011; Wylie & Chi, 2014), it is not surprising from a theoretical perspective that this hypothesis was supported. However, the support for it specifically in the context of advanced Computer Science learning contributes toward building a case against the prevailing pedagogy in adult learning of Computer Science that privileges learning through problem solving (especially computer programming), almost to the exclusion of other forms of learning activities, including reflection. The results of this study build on some prior studies of learning through reflection in collaborative software development (Sankaranarayanan, Kandimalla, et al., 2022; Sankaranarayanan, Ma, et al., 2022). Reflection prompts in these past studies were similar to those offered in the NSI condition, in that they were generic reflection prompts that were not tailored to the details of specific student solutions. The current study probes deeper into the features of example study that affect the substance of student reflection. As we insert more contextual details into the prompts for reflection and then observe how students align the focus of their reflection with those details or not, as well as whether those details align with what students learn from engaging in reflection, we are able to learn more about how to guide student reflection to learn specific concepts.

LIMITATIONS

In this study, identical pre and posttests were used for the learning assessment, which does not allow separation of learning from the intervention from learning from the test. However, we carefully checked the framing of test questions to ensure that no answers were given away in the framing of any test question for the question itself or any other question on the test.

A final limitation is that students were not assigned randomly to teams. The instructor's practice is to allow students complete freedom to find team-mates and then declare their team early in the semester. Since teams were randomly assigned to conditions, there is no reason to suspect that there are systematic differences in team composition across conditions resulting from this. However, this lack of control may introduce noise in the analysis. This may be addressed in replication studies conducted in different classes where random assignment may be allowed.

FUTURE WORK

The results of this study show support for the value of an LLM-enabled intervention, however the effect is small and specific to a subpopulation of students. The analysis showed value for reflection in general, however neither GEN nor CONT increased the total amount of reflection over that of NSI. Going forward, future work should investigate how the intervention could be further adjusted in order to increase student engagement with it. Furthermore, neither the high pretest students nor the low pretest students benefited more from either form of example reflection than the completely generic NSI control condition. Thus, with respect to arguing the importance of the capabilities of LLMs to improve student learning substantially, the current findings do not rise to that level. Future work should continue to probe more precisely into how, when, and to what extent LLMs are able to positively impact student learning, in particular how the application of LLM capabilities aligns with best practices in scaffolding for learning as they vary with the abilities and dispositions of individual students.

FUNDING INFORMATION

This work was funded in part by NSF grant DSES 2222762.

CONFLICT OF INTEREST STATEMENT

The authors have no financial interest in the reported research. In terms of potential COIs with journal editorial members, one of the authors has recently published jointly with Sanna Järvelä, Andy Nguyen and Yannis Dimitriadis.

DATA AVAILABILITY STATEMENT

The Exempt determination does not allow making the student data from this research publically available. However, the data from the study will be preserved on secure servers at the host institution in perpetuity and thus usable by the research team specifically named on the approved protocol.

ETHICS STATEMENT

The Institutional Review Board from the host university where the study was run granted an Exempt determination for the Cloud Computing course such that anonymised data from course activities could be used for research purposes.

PATIENT CONSENT STATEMENT

All students participating in the research signed a consent form at the beginning of the academic term agreeing that anonymised data from their participation in course activities could be used for research purposes.

REFERENCES

Baker, R. S., Corbett, A. T., & Koedinger, K. R. (2012). Learning to distinguish between representations of data: A cognitive tutor that uses contrasting cases. In *Embracing diversity in the learning sciences* (pp. 58–65). Routledge.

Bego, C. R., Chastain, R. J., & DeCaro, M. S. (2023). Designing novel activities before instruction: Use of contrasting cases and a rich dataset. *British Journal of Educational Psychology*, *93*(1), 299–317.

Bjork, R. A., & Bjork, E. L. (2020). Desirable difficulties in theory and practice. *Journal of Applied Research in Memory and Cognition*, *9*(4), 475–479.

Buchan, J., & Pearl, M. (2018). Leveraging the mob mentality: An experience report on mob programming. In *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018, EASE 18*. Christchurch, New Zealand: ACM.

Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*(2), 145–182.

Cotton, D. R., Cotton, P. A., & Shipway, J. R. (2024). Chatting and cheating: Ensuring academic integrity in the era of ChatGPT. *Innovations in Education and Teaching International*, *61*(2), 228–239.

Cress, U., & Kimmerle, J. (2023). Co-constructing knowledge with generative AI tools: Reflections from a CSCL perspective. *International Journal of Computer-Supported Collaborative Learning*, *18*(4), 607–614.

Cress, U., Oshima, J., Rosé, C., & Wise, A. F. (2021). *International handbook of computer-supported collaborative learning* (Vol. *19*, p. 2021). Springer.

Dai, Y., Liu, A., & Lim, C. P. (2023). Reconceptualizing ChatGPT and generative ai as a student-driven innovation in higher education. *Procedia CIRP: The 33rd CIRP Design Conference*, *119*, 84–90.

Durkin, K., Star, J. R., & Rittle-Johnson, B. (2017). Using comparison of multiple strategies in the mathematics classroom: Lessons learned and next steps. *ZDM*, *49*, 585–597.

Fabic, G. V. F., Mitrovic, A., & Neshatian, K. (2019). Evaluation of parsons problems with menu-based self-explanation prompts in a mobile python tutor. *International Journal of Artificial Intelligence in Education*, *29*(4), 507–535.

Griffin, T. D., Jaeger, A. J., Britt, M. A., & Wiley, J. (2024). Improving multiple document comprehension with a lesson about multi-causal explanations in science. *Instructional Science*, *52*, 1–26.

Gweon, G., Rosé, C., Albright, E., & Cui, Y. (2007). Evaluating the effect of feedback from a CSCL problem solving environment on learning, interaction, and perceived interdependence. In *Proceedings of the International Conference of the Learning Sciences* (pp. 234–243). New Brunswick, NJ.

Jayagopal, D., Lubin, J., & Chasins, S. E. (2022). Exploring the learnability of program synthesizers by novice programmers. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22)*, New York, NY, USA. Association for Computing Machinery.

Kalyuga, S., Chandler, P., & Sweller, J. (2001). Learner experience and efficiency of instructional guidance. *Educational Psychology*, *21*(1), 5–23.

Kasneci, E., Sessler, K., Kuechemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, S., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Kazemitabaar, M., Chow, J., Ma, C. K. T., Ericson, B. J., Weintrop, D., & Grossman, T. (2023). Studying the effect of ai code generators on supporting novice learners in introductory programming. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, CHI '23*, New York, NY, USA. Association for Computing Machinery.

Kazemitabaar, M., Hou, X., Henley, A., Ericson, B. J., Weintrop, D., & Grossman, T. (2023, November). How novices use LLM-based code generators to solve CS1 coding tasks in a self-paced learning environment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research* (pp. 1–12). Koli, Finland.

Keech, A., & Muldner, K. (2024, July). Evaluating the effectiveness of comparison activities in a CTAT tutor for algorithmic thinking. In *International Conference on Artificial Intelligence in Education* (pp. 149–162). Springer Nature Switzerland.

Koschmann, T. (Ed.). (2017). *Computer supported collaborative learning 2005: The next 10 years!* Routledge.

Kumar, R., Rosé, C. P., Wang, Y.-C., Joshi, M., & Robinson, A. (2007). Tutorial dialogue as adaptive collaborative learning support. In *Proceedings of the 2007 Conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work* (pp. 383–390). Los Angeles, California.

Kumar, R., Rosé, C. P., & Witbrock, M. J. (2009). Build- ing conversational agents with basilica. In M. Johnston & F. Popowich (Eds.), *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session* (pp. 5–8, Boulder, CO.). Association for Computational Linguistics.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *ACM SIGCSE Bulletin*, *37*(3), 14–18.

Lee, B., & Muldner, K. (2020, April). Instructional video design: Investigating the impact of monologue-and dialogue-style presentations. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (pp. 1–12). Honolulu, HI.

Ma, N., Qian, J., Gong, K., & Lu, Y. (2023). Promoting programming education of novice programmers in elementary schools: A contrasting cases approach for learning programming. *Education and Information Technologies*, *28*(7), 9211–9234.

Murphy, L., Fitzgerald, S., Lister, R., & McCauley, R. (2012, September). Ability to 'explain in plain English' linked to proficiency in computer-based programming. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research* (pp. 111–118). Auckland, New Zealand.

Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. *Educational Psychologist*, *38*(1), 1–4.

Paas, F. G. W. C., & Van Merriënboer, J. J. G. (1994). Variability of worked examples and transfer of geometrical problem-solving skills: A cognitive-load approach. *Journal of Educational Psychology*, *86*(1), 122–133. https://doi.org/10.1037/0022-0663.86.1.122

Renkl, A. (2014). Learning from worked examples: How to prepare students for meaningful problem solving. In V. A. Benassi, C. E. Overson, & C. M. Hakala (Eds.), *Applying science of learning in education: Infusing psychological science into the curriculum* (pp. 118–130). Society for the Teaching of Psychology.

Rittle-Johnson, B., Loehr, A. M., & Durkin, K. (2017). Promoting self-explanation to improve mathematics learning: A meta-analysis and instructional design principles. *ZDM*, *49*, 599–611.

Roelle, J., & Berthold, K. (2016). Effects of comparing contrasting cases and inventing on learning from subsequent instructional explanations. *Instructional Science*, *44*, 147–176.

Rosé, C. P., & Ferschke, O. (2016). Technology support for discussion based learning: From computer supported collaborative learning to the future of massive open online courses. *International Journal of Artificial Intelligence in Education*, *26*, 660–678.

Ryoo, J. J. (2019). Pedagogy that supports computer science for all. *ACM Transactions on Computing Education (TOCE)*, *19*(4), 1–23.

Sankaranarayanan, S., Kandimalla, S. R., Bogart, C. A., Murray, R. C., Hilton, M., Sakr, M. F., & Rosé, C. P. (2022). Collaborative programming for work-relevant learning: Comparing programming practice with example-based reflection for student learning and transfer task performance. *IEEE Transactions on Learning Technologies*, *15*(5), 594–604.

Sankaranarayanan, S., Kandimalla, S. R., Cao, M., Maronna, I., An, H., Bogart, C., Murray, R. C., Hilton, M., Sakr, M., & Penstein Rosé, C. (2020). Designing for learning during collaborative projects online: Tools and takeaways. *Information and Learning Sciences*, *121*(7–8), 569–577. https://doi.org/10.1108/ILS-04-2020-0095

Sankaranarayanan, S., Ma, L., Kandimalla, S. R., Markevych, I., Nguyen, H., Murray, R. C., Bogart, C., Hilton, M., Sakr, M., & Rosé, C. P. (2022). Collaborative reflection in the flow of programming: Design- ing effective collaborative learning activities in advanced computer science contexts. In *Proceedings of the 15th International Conference on Computer-Supported Collaborative Learning-CSCL 2022* (pp. 67–74). Hiroshima, Japan: International Society of the Learning Sciences.

Scardamalia, M., & Bereiter, C. (1987). Knowledge telling and knowledge transforming in written composition. *Advances in Applied Psycholinguistics*, *2*, 142–175.

Schwartz, D. L., & Bransford, J. D. (1998). A time for telling. *Cognition and Instruction*, *16*(4), 475–522. https://doi.org/10.1207/s1532690xci1604_4

Schwartz, D. L., Chase, C. C., Oppezzo, M. A., & Chin, D. B. (2011). Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer. *Journal of Educational Psychology*, *103*(4), 759–776. https://doi.org/10.1037/a0025140

Sidney, P. G., Hattikudur, S., & Alibali, M. W. (2015). How do contrasting cases and self-explanation promote learning? Evidence from fraction division. *Learning and Instruction*, *40*, 29–38.

Snowdon, S. S. (2011, August). Explaining program code: Giving students the answer helps-but only just. In *Proceedings of the Seventh International Workshop on Computing Education Research, Providence* (pp. 93–100). Rhode Island, USA.

Sok, S., & Heng, K. (2023). ChatGPT for education and research: A review of benefits and risks. *SSRN Electronic Journal*. https://ssrn.com/abstract=4378735. https://doi.org/10.2139/ssrn.4378735

Stamper, J., Xiao, R., & Hou, X. (2024, July). Enhancing llm-based feedback: Insights from intelligent tutoring systems and the learning sciences. In *International Conference on Artificial Intelligence in Education* (pp. 32–43). Springer Nature Switzerland.

Stokel-Walker, C., & Van Noorden, R. (2023, February). What ChatGPT and generative AI mean for science. *Nature*, *614*, 214–216.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, *2*(1), 59–89.

Sweller, J., Van Merrienboer, J. J., & Paas, F. G. (1998). Cognitive architecture and instructional design. *Educational Psychology Review*, *10*, 251–296.

Tegos, S., Demetriadis, S., & Karakostas, A. (2015). Promoting academically productive talk with conversational agent interventions in collaborative learning settings. *Computers & Education*, *87*, 309–325.

Tuovinen, J. E., & Sweller, J. (1999). A comparison of cognitive load associated with discovery learning and worked examples. *Journal of Educational Psychology*, *91*(2), 334–341.

Van Gog, T., Kester, L., & Paas, F. (2011). Effects of worked examples, example-problem, and problem-example pairs on novices' learning. *Contemporary Educational Psychology*, *36*(3), 212–218.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30–40.

Wylie, R., & Chi, M. T. (2014). 17 the self-explanation principle in multimedia learning. In *The Cambridge handbook of multimedia learning*, edited by Mayer, Richard E. (pp. 413–432). Cambridge University Press.

Young, J. C., & Shishido, M. (2023). Evaluation of the potential usage of ChatGPT for providing easier reading materials for ESL students. In T. Bastiaens (Ed.), *Proceedings of EdMedia + Innovate Learning* (pp. 155–162, Vienna, Austria.). Association for the Advancement of Computing in Education (AACE).

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Naik, A., Yin, J. R., Kamath, A., Ma, Q., Wu, S. T., Murray, R. C., Bogart, C., Sakr, M., & Rose, C. P. (2025). Providing tailored reflection instructions in collaborative learning using large language models. *British Journal of Educational Technology*, *56*, 531–550. https://doi.org/10.1111/bjet.13548