

# Turning 500+ Students into Teachers: A Semester-Long Study of an AI Teachable Agent in an Undergraduate Algorithms Course

Chenyang Wang  
CHILI Lab, EPFL  
Lausanne, Switzerland  
chenyang.wang@epfl.ch

Christopher Petrie  
CHILI Lab, EPFL  
Lausanne, Switzerland  
christopher.petrie@epfl.ch

Miltiadis Stouras  
Theory of Computation Laboratory 2,  
EPFL  
Lausanne, Switzerland  
miltiadis.stouras@epfl.ch

Nicolas Ettlín  
Theory of Computation Laboratory 2,  
EPFL  
Lausanne, Switzerland  
nicolas.ettlin@epfl.ch

Amaury George  
Theory of Computation Laboratory 2,  
EPFL  
Lausanne, Switzerland  
amaury.george@epfl.ch

Paola Mejia-Domenzain  
Machine Learning for Education  
Laboratory, EPFL  
Lausanne, Switzerland  
paola.mejia@epfl.ch

Vinitra Swamy  
Machine Learning for Education  
Laboratory, EPFL  
Lausanne, Switzerland  
vinitra.swamy@epfl.ch

Tanja Käser  
Machine Learning for Education  
Laboratory, EPFL  
Lausanne, Switzerland  
tanja.kaeser@epfl.ch

Ola Svensson  
Theory of Computation Laboratory 2,  
EPFL  
Lausanne, Switzerland  
ola.svensson@epfl.ch

## Abstract

Large language model (LLM) tools can provide students with rapid solutions but may reduce opportunities for productive struggle and explanation generation that support conceptual learning. Learning-by-teaching (LBT) offers an alternative solution by positioning students as tutors; however, evidence for LLM-based teachable agents remains limited, particularly for longitudinal deployments and large-scale evaluations that connect LBT interactions to conceptual understanding in authentic courses. We present *Explicite*, a platform that integrates an AI teachable agent, *Algorithm Apprentice*, into an undergraduate algorithms course to operationalise LBT at scale. We report an 11-week field deployment in a real course with 546 students, analysing 3,809 student-agent LBT dialogues alongside quiz and survey data. Students engaged consistently in multi-turn teaching interactions over the semester, although the depth and authenticity of these interactions varied, including instances of direct reuse of externally sourced content. Using generalised linear mixed-effects models, we find that explanation-oriented dialogue behaviours (e.g., elaboration and showing reasoning) are associated with fewer quiz attempts (i.e., fewer incorrect submissions), whereas external-content reuse is associated with slightly more repeated attempts. Compared to a baseline reading activity, the LBT condition corresponds to a modest reduction in expected quiz attempts, although this comparison is confounded by substantial differences in time-on-task. Overall, these results provide longitudinal, large-scale evidence on LLM-based teachable agents in an

authentic computer science course and inform the design and practice of systems that aim to support sustained, effortful and scalable LBT interactions.

## CCS Concepts

• **Applied computing** → **Interactive learning environments**; • **Social and professional topics** → *Computing education*; • **Human-centered computing** → *Human computer interaction (HCI)*.

## Keywords

Teachable Agent, Learning by Teaching, Large Language Model, Productive Struggle, Computer Science Education

## ACM Reference Format:

Chenyang Wang, Christopher Petrie, Miltiadis Stouras, Nicolas Ettlín, Amaury George, Paola Mejia-Domenzain, Vinitra Swamy, Tanja Käser, and Ola Svensson. 2026. Turning 500+ Students into Teachers: A Semester-Long Study of an AI Teachable Agent in an Undergraduate Algorithms Course. In *Proceedings of the Thirteenth ACM Conference on Learning @ Scale (L@S '26)*, June 29–July 03, 2026, Seoul, Republic of Korea. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3774398.3811623>

## 1 Introduction

Large language model (LLM) tools like ChatGPT now provide students with near-instant answers and polished explanations for many course tasks. While these tools can reduce search and drafting effort, they can also take away opportunities for productive struggle—trying to solve before seeing a solution and using errors and feedback to refine understanding—that supports robust learning [6, 21, 22, 25]. Moreover, when an LLM provides completed reasoning, it can displace the learner’s own explanation generation (the self-explanation effect) [9], which is also important for learning.



This work is licensed under a Creative Commons Attribution 4.0 International License. *L@S '26, Seoul, Republic of Korea*

© 2026 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-2293-6/2026/06  
<https://doi.org/10.1145/3774398.3811623>

Recent research in educational technology has begun to address these concerns of LLMs by shifting from “answer engines” toward interactive learning supports that draw on learning sciences, including Socratic questioning, stepwise scaffolding and learning through argumentation [3, 16, 36, 38]. For example, OpenAI’s Study Mode guides learners through reasoning with questions and structured prompts rather than providing full solutions [3, 37]. Similarly, Anthropic’s Claude for Education introduces a Learning Mode that emphasises Socratic dialogue [37]. Despite this progress, such systems often preserve the conventional roles of student-as-asker and AI-as-expert. An alternative is flipping the role using a learning-by-teaching (LBT) approach, where students learn by teaching and explaining to a novice, monitoring understanding, and revising instruction. Prior work on LBT pedagogy shows that the act of teaching can benefit people by prompting knowledge organisation and metacognitive monitoring [14, 30, 33]. However, sustaining LBT opportunities at scale is difficult: in a lecture hall of 500+ students, it is rarely feasible for each learner to regularly teach a human partner.

LLM-based teachable agents offer a potential pathway to operationalise LBT in large classes by positioning the AI as an apprentice that must be taught rather than consulted. Recent work has begun to investigate LLM-based LBT systems that position students to instruct an AI agent about algorithmic and procedural concepts, with the goal of supporting learning and metacognitive skills [8, 11, 18, 32]. Yet the empirical evidence base for such systems remains limited in two ways. First, existing evaluations are mainly short-term or lab-based, leaving open how student interactions change over time as novelty fades and course demands fluctuate. Second, there is limited large-scale evidence in authentic, high-enrollment courses that connects LBT interactions with measures of conceptual understanding, which is necessary for informing both learning design and responsible deployment.

To address these gaps, we present *Explicue*, an application that integrates an AI teachable agent, *Algorithm Apprentice*, into an undergraduate algorithms course. *Explicue* is designed to consistently invert the typical interaction: students explain core concepts and procedures to *Algorithm Apprentice*, and the agent responds as a novice by asking clarification questions, attempting to restate ideas, and signalling uncertainty when appropriate. We conducted a semester-long (11-week) field deployment with 546 students in a real course setting and investigated the following research questions:

**RQ1:** How do students’ interactions with an AI teachable agent evolve over the course of a semester-long deployment in a real undergraduate algorithms course?

**RQ2:** How do students’ LBT activities with the AI teachable agent enhance their conceptual understanding of course content?

The contributions of this work are threefold. First, we present an 11-week longitudinal field study of an AI teachable agent deployed in an undergraduate algorithms course with 546 students, providing large-scale empirical evidence on learning-by-teaching with AI in authentic educational settings. Second, we offer quantitative insights into how engaging in learning-by-teaching activities with an AI teachable agent relates to students’ conceptual understanding and knowledge development. Third, we derive practice and design

implications for the scalable integration of AI teachable agents into large computer science courses, informing future AI-supported learning at scale initiatives.

## 2 Related Work

### 2.1 LLMs in Computer Science Courses

LLMs have been widely explored in computer science education because they can generate, document, and explain code, creating new opportunities alongside new risks in programming and algorithms courses [31]. A prominent research direction seeks to move beyond “answer engines” by shaping LLM behaviour to emphasise guidance over solutions, leveraging tutoring principles such as Socratic dialogue and instructional scaffolding [3, 16, 36]. This approach is reflected in deployed systems including Khanmigo, OpenAI’s Study Mode, and Anthropic’s Learning Mode, which encourage learners to reason through questions, hints, and incremental steps rather than receiving complete answers [1, 2, 37]. However, students may bypass these constraints by turning to unmodified models. Classroom evidence on the use of unmodified models like ChatGPT remains mixed [20]: heavy use can undermine productive struggle, lead to overreliance, and hinder deeper engagement with algorithmic concepts [4, 12, 21]. Reported outcomes range from decreased self-efficacy or weaker early-course performance [20]. The adoption of LLM tools in computer science classes is also uneven and appears sensitive to peer norms and perceived career value [29].

### 2.2 Learning-by-Teaching

One way to ensure active cognitive engagement is to flip tutor–student roles by letting learners act as teachers. Learning-by-teaching (LBT) is an active learning approach in which students deepen their understanding by teaching others. Compared to passive reception, LBT requires learners to externalise and organise their knowledge—e.g., by explaining concepts, generating examples, and responding to questions—which can increase engagement and support higher-order reasoning [10]. Studies of peer tutoring show that when students explain concepts to others, the tutor often benefits as much as the tutee [33]. Such reciprocal teaching involved students asking questions and summarising, which significantly improved comprehension [13]. On the other side, research work also showed that tutors gain a deeper understanding, especially when responding to probing questions [33].

### 2.3 Teachable Agents

To facilitate learning-by-teaching, researchers have developed teachable agents—computer systems designed to enable students to teach. Students who teach artificial agents have demonstrated greater mastery than those who studied normally [5, 23]. This phenomenon, the *protégé effect*, suggests that students put forth more effort and engage in deeper reflection when they feel responsible for teaching [14]. In teaching an agent, learners are compelled to clarify misconceptions and fill gaps, thereby effectively learning through explanation and metacognition [30, 33].

Various agents have been extensively studied and applied in different educational contexts [7, 27, 28], such as learning mathematics [24], science [5] and pronunciation skill [40]. For instance, Biswas *et al.* [5, 23] developed Betty’s Brain, a teachable computer

agent in the domain of river ecosystems, allowing students to construct causal concept models by teaching the agent. Physically embodied teachable agents have also been explored. Tanaka *et al.* utilised a humanoid robot as a pupil, asking children to teach it verbs [35]. Gargot *et al.* [15] proposed a teachable robot to treat severe dysgraphia, enabling children to practice handwriting by teaching it. Similarly, Wang *et al.* [39] investigated the use of a teachable robot to support the development of children’s AI literacy.

More recently, multiple works have explored LLM-based teachable agents, since LLMs can not only create more realistic dialogues between teachable agents and students but also simplify the construction and deployment of these agents compared to previous AI approaches. Jin *et al.* developed a prompting pipeline for LLMs in the *TeachYou* environment to restrain model competence and solicit “why/how” questions, enhancing learner explanation in algorithm learning [18]. Chen *et al.* studied teaching ChatGPT as a teachable agent in programming education, finding improvements in knowledge gains and self-regulated learning, though limited effects on error-correction practice [8]. Ma *et al.* introduced *HypoCompass*, where students help an agent debug code generated by an LLM, supporting hypothesis construction and deliberate practice [26]. Extensions of LLM-powered teachable agents to other domains (e.g., music theory) likewise suggest benefits for conceptual learning and cognitive load reduction [19].

Overall, these works highlight the emerging potential of LLMs as teachable agents that scaffold reflection, dialogue, and learner articulation across domains. However, important gaps remain. Most evaluations are short-term or controlled, leaving open how teaching-oriented interactions with LLM-based agents evolve across a long period. In addition, large-scale, in-situ evidence from high-enrollment computer science courses that links LBT interactions to conceptual understanding is still limited. Therefore, this work aims to bridge these gaps, which is necessary to guide future learning design and responsible deployment at scale.

### 3 Methods

#### 3.1 The AI Algorithm Apprentice Implementation

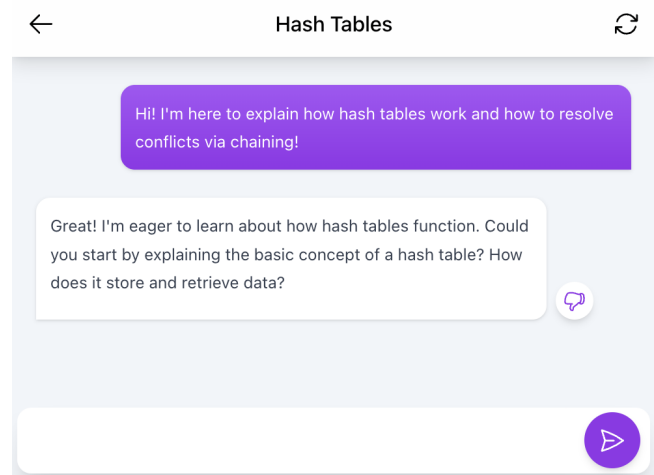
We implemented the Explique application<sup>1</sup>, an AI conversational teachable agent, in the role of an *Algorithm Apprentice* rather than a tutor, based on OpenAI’s GPT-4o large language model (LLM). The Explique application is integrated into the digital learning platform at EPFL and is available to all registered students. Prompts were carefully engineered to establish three constraints: (a) the agent should always maintain the stance of a novice student, (b) it should persistently request relevant questions or clarifications until the student gives a satisfactory and accurate explanation, and (c) it must never reveal, hint at, or generate the target explanation. For each LBT exercise targeting a specific algorithm, the prompt is tailored to the corresponding essential concept and learning objective. For example, in the exercise for the Insertion Sort algorithm, the system prompt required the AI to ask probing questions about the steps of the algorithm while resisting attempts by students to reverse the

teaching roles. For Strassen’s Algorithm, the prompt was adapted to emphasise high-level divide-and-conquer concepts rather than technical details of matrix multiplication. These variations illustrate how Explique can be tuned to different pedagogical focal points by aligning the agent’s “learning role” with course objectives.

As shown in Figure 1, students accessed Explique through a chat interface in their weekly after-class coursework. Each LBT exercise began with a predefined student message instructing them to teach the Algorithm Apprentice the target concept. The AI apprentice responded with clarifying questions, reflections, or requests for elaboration. The interaction concluded only when the model determined that a sufficient explanation had been provided; this completion criterion was operationalised as an LLM tool function<sup>2</sup> using a course-objective-aligned, algorithm-specific prompt. For example, in the exercise for the Insertion Sort algorithm, we designed a tool function prompt

*“Call this function when the student has sufficiently explained the Insertion Sort algorithm, including: (1) how each new element is compared with elements in the sorted portion and inserted into the correct position; (2) ...”*

to determine the LBT exercise completion criterion. Importantly, Explique enforced productive struggle. Students who attempted to bypass the activity (e.g., by asking the apprentice to explain the algorithm back) were met with resistance, since the apprentice was explicitly instructed to decline answer-giving.



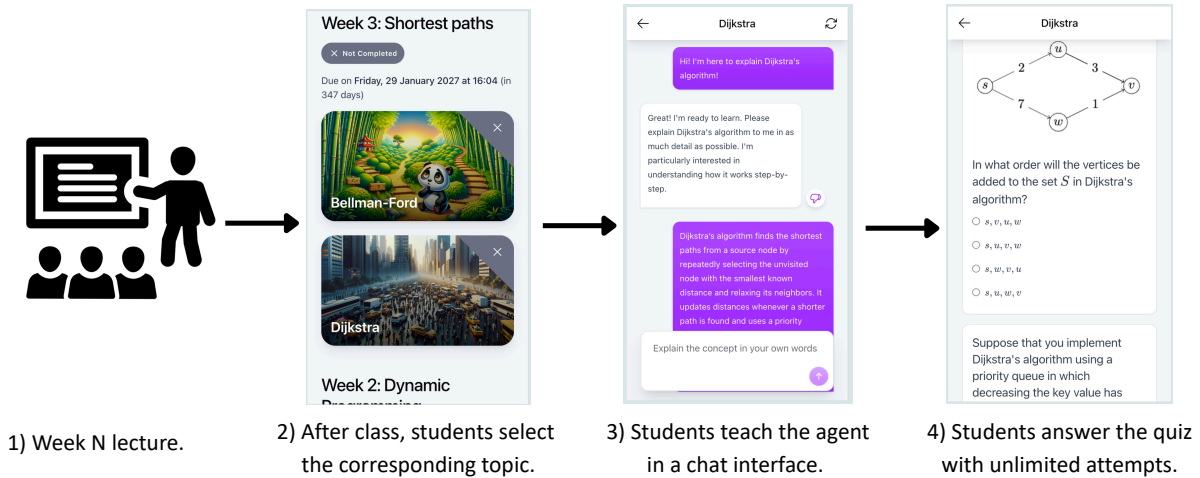
**Figure 1: The Learning-by-Teaching (LBT) exercise interface in Explique.** The student (purple) begins with an automated message stating that they are here to explain a concept (e.g., hash tables). The AI agent (white) then takes on the role of an apprentice by asking questions about the target concept, thereby establishing the LBT interaction design.

#### 3.2 Semester-long Study in Undergraduate Algorithms Course

**Integration in the Course.** We integrated the Explique application into the Algorithm I course of 546 students at EPFL as a weekly

<sup>1</sup><https://explique.epfl.ch>

<sup>2</sup><https://developers.openai.com/api/docs/guides/tools/>



**Figure 2: The AI teachable agent, named *Algorithm Apprentice*, was integrated into the *Algorithm I* course as a weekly after-class exercise. Steps 2–4 are provided in the *Explique* application. Students first complete an exercise by teaching the *Algorithm Apprentice* through a chat interface. After this step, the corresponding multiple-choice quiz is unlocked. Students may attempt the quiz an unlimited number of times until all items are answered correctly. Students submit all answers at once, and no feedback is provided after an unsuccessful submission.**

after-class exercise, co-designed with the instructor<sup>3</sup>. This course is offered as the first algorithm course for computer science undergraduate students. As shown in Figure 2, the system was integrated directly into the rhythm of weekly coursework and paired with the existing multiple-choice question (MCQ) quiz structure of the class. Each week, students first attended a lecture covering algorithmic theory. After class, they were offered corresponding LBT exercises in the *Explique* application, followed by a quiz of 3–5 MCQs (integrated into the application) designed by the instructor to assess their conceptual understanding of the algorithms. Specifically, students could attempt the quiz an unlimited number of times, with a one-minute lockout after incorrect submissions. No feedback was provided following unsuccessful attempts. To access each quiz, students first had to complete the LBT exercise. Instructors retained oversight through dashboards that displayed completion rates and qualitative patterns in students’ interactions during the LBT exercise. While the LBT exercise itself provided no grades, integrating *Algorithm Apprentice* as a gatekeeper for quizzes ensured that its Learning-by-Teaching pedagogy was a consistent part of the course. Quizzes were autograded and counted for an optional 5% of the final grade, providing extrinsic motivation for students to engage with the system. Overall, this way of integrating LBT exercises into the existing course curriculum creates repeated, scalable opportunities for students to teach core algorithms to an AI apprentice while holding constant the stakes and timing of the activity.

**Study Design.** To gain insights into the benefits of LBT activities on students’ conceptual understanding of the algorithms, we additionally set up a baseline condition, in which students read an instructor-provided explanation of the algorithm concept instead of teaching the *Algorithm Apprentice*. We adopted a crossover study design by randomly assigning students into two groups (A and B).

<sup>3</sup>This research has been reviewed and approved by EPFL Human Research Ethics Committee.

**Table 1: Algorithm topics covered over 11 weeks**

Week	Group A LBT exercise	Group B LBT exercise
1	Merge Sort	Insertion Sort
2	Maximum Sum Subarray	Strassen’s Algorithm
3	Priority Queues	Heap Sort
4	Rod Cutting	Matrix-Chain Multiplication
5	Longest Common Subsequence	Optimal Binary Search Trees
6	Breadth-First Search	Depth-First Search
7	Topological Sorting	Ford–Fulkerson Algorithm
8	Bipartite Matching via Max Flow	Edge-disjoint paths via max flow
9	Prim’s Algorithm	Kruskal’s Algorithm
10	Bellman–Ford	Dijkstra
11	Quicksort	Hash Tables

As shown in Table 1, for instance, in Week 1, Group A completed the LBT activity on Merge Sort before Quiz 1 and the reading activity on Insertion Sort before the corresponding Quiz 2; Group B did the reverse, starting with the reading activity on Merge Sort. Thus, the order of the two conditions was counterbalanced across weeks. This ensured that every student experienced both conditions each week, but in different sequences. Across 11 weeks of the semester, this setup yielded 22 exercises and 22 quiz results per student, with half in the LBT condition and half in the reading condition. To gather feedback on the experiment, we conducted an optional survey with open-ended questions such as “What are the limitations?” and “What could be improved?” the following semester.

### 3.3 Data Collection and Analysis

Out of 546 students taking the course, 456 agreed to have their data collected and used in this study. Due to a technical issue in the exercises for Priority Queues and Depth-First Search, the corresponding data were excluded from the analysis. In the end, we collected the 3809 student-agent LBT dialogues with completed quizzes from 453 students, including a total of 35,702 student messages. The number of attempts until a successful quiz submission in both conditions was logged to measure learning performance. Interaction logs, including the content and timestamp of each student message, were stored for analysis.

**Interaction Pattern Analysis.** To holistically examine how students interacted with the AI apprentice, we categorised each LBT conversation into three interaction patterns: *Detailed*, *Minimal*, and *External Content Use*. *Detailed* interactions refer to conversations in which students provided extended explanations and articulated stepwise reasoning of the target concept beyond the immediate prompt. *Minimal* interactions capture exchanges characterised by brief, prompt-bound responses with limited elaboration. *External Content Use* denotes conversations exhibiting behavioural or stylistic signals consistent with externally generated or copy-pasted material.

Specifically, to categorise each complete student-agent conversation, we adopted an established three-tier hybrid classification procedure of [41]. To balance scalability and interpretability, the procedure combined minimal heuristics, Word-per-Minute (WPM) based detection, and LLM-based semantic classification through a hierarchical decision procedure: 1) Minimal-response heuristic. We first applied a conservative length-based heuristic to identify potentially minimal interactions. Conversations in which the longest student message did not exceed a predefined word-count threshold (longest message < 50 words) were flagged as candidates for the *Minimal* category. 2) WPM-based detection. To detect potential *External Content Use*, we computed WPM for the longest student message based on timestamp differences. Conversations whose longest message exceeded a conservative WPM threshold were classified as *External Content Use*, as such production speeds are unlikely under typical typing behaviour. 3) LLM-based semantic classification. Finally, we employed an LLM to classify conversations according to their overall discourse characteristics. The model was prompted with structured descriptions of the three predefined categories and instructed to assign a single label per conversation. LLM outputs were used to label cases not captured by earlier steps.

**Learning-by-Teaching Dialogue Analysis.** To gain detailed qualitative insights into how students teach the AI agent, we used a deductive-inductive coding approach to analyse each student’s message. Specifically, informed by the coding scheme of the educational dialogue analysis framework [17] and the prior work incorporated LLM-facilitated LBT dialogue qualitative analysis [32], we iteratively annotated and developed the codebook on 20% randomly sampled dialogues per week, including a total of 6800 students’ messages. The final derived codebook is summarised in Table 2 with three categories: topic relevance, knowledge construction, and prompting type. To better understand how various dialogue acts can enhance student learning, we consider the *Show Reasoning*, *Provide Elaboration*, and *Give Examples* as dialogue acts conducive

to deep learning based on previous literature [17]. The dialogue acts of *Produce Basic Information*, *Provide Acknowledgement*, and *External Content Use* are considered as promoting shallow learning.

With the manually annotated 20% data as human reference, we then implemented an LLM-based coding pipeline using GPT-5.2, in which the system prompt was iteratively refined using a separate 5% development subset until the model’s agreement with the human codes reached a Krippendorff’s  $\alpha$  of 0.92 on the development data. Using this fine-tuned prompt, we evaluated the LLM on the remaining 15% annotated subset and observed substantial agreement with the human labels (Krippendorff’s  $\alpha = 0.82$ ). Given this level of reliability, we applied the LLM coder to code the remaining, unlabelled dialogues to enable analysing large-scale LBT dialogues. In addition, the aforementioned WPM-based detection approach was integrated into the post-processing phase per message to enhance the *External Content Use* label detection.

**Table 2: Codebook for LBT dialogue analysis. The symbol “+” denotes deep learning acts, while “-” denotes shallow learning acts.**

Code	Definition
	(a) <i>Topic relevance</i>
On Topic	Student message relates to the exercise.
	(b) <i>Knowledge construction</i>
Show Reasoning <sup>+</sup>	Student provides reasoning/justification/causal tracing.
Provide Elaboration <sup>+</sup>	Student elaborates/explains in more detail (e.g., procedural description, expanded explanation).
Give Examples <sup>+</sup>	Student provides a concrete example/case to illustrate.
Produce Basic Info. <sup>-</sup>	Student provides minimal basic factual/algorithmic information relevant to the task.
Provide Ack. <sup>-</sup>	Student acknowledges understanding, confirms, or paraphrases comprehension.
External Content Use <sup>-</sup>	Student copies information/answers from external sources (e.g., Wikipedia).
	(c) <i>Prompting type</i>
Seek Elaboration	Student requests more explanation/clarification from the chatbot tutee.
Seek Example	Student requests an example from the chatbot tutee.
Seek Answer	Student requests the direct answer from the chatbot tutee.

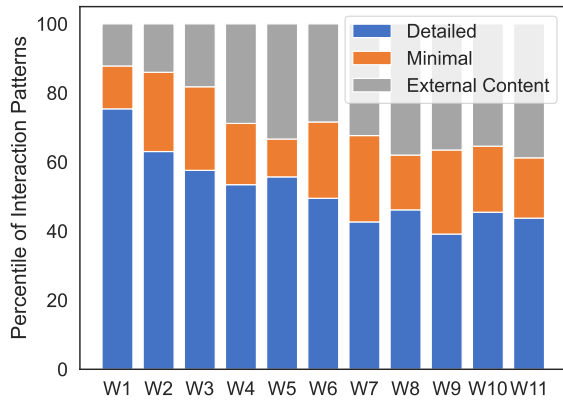


Figure 3: Evolution of LBT interaction patterns across 3,809 dialogues over 11 weeks.

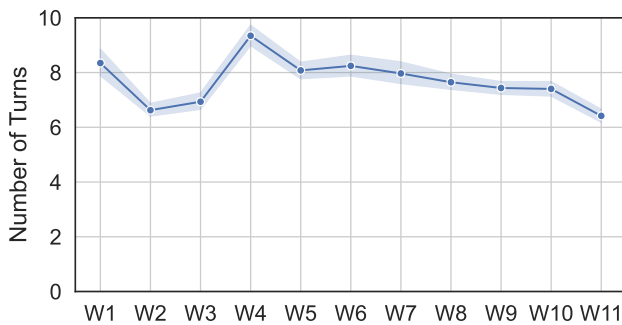


Figure 4: Average number of LBT dialogue turns per student by week. Students consistently made around 8 turns per LBT exercise, indicating relatively stable multi-step dialogue.

## 4 Results

### 4.1 RQ1: How do students’ interactions with an AI teachable agent evolve over the course of a semester-long deployment in a real undergraduate algorithms course?

To approach RQ1, we analysed the proportion of three interaction patterns and the number of turns per LBT dialogue over 11 weeks.

**Evolution of LBT Interaction Patterns.** Overall, among 3,809 student–agent LBT dialogues, 53.3% of conversations were classified as *Detailed*, 19.5% as *Minimal*, and 27.2% as *External Content Use*.

Figure 3 further reveals a clear temporal shift in interaction patterns. The proportion of *Detailed* dialogues was highest at the beginning of the semester (approximately three-quarters in Week 1) and gradually declined over time, stabilising around 45–55% in the later weeks. These interactions were characterised by extended explanations, stepwise reasoning, and connections across algorithmic concepts. For example, in Figure 5, a student articulated both iterative and recursive implementations of Depth-First Search (DFS) before being prompted further, reflecting proactive elaboration. In

contrast, *Minimal* interactions remained relatively stable throughout the semester, fluctuating around 15–25% across weeks without a clear monotonic trend. These dialogues were typically concise and prompt-driven, with students providing short, factual responses sufficient for task completion but offering limited elaboration. The proportion of *External Content Use* interactions increased over time. While initially below 15% in Week 1, this pattern rose steadily from mid-semester onward and reached approximately 30–35% by Weeks 10–11. These cases were often identified through unusually high WPM rates, consistent with direct reuse of content from external sources.

**User:** Hi! I’m here to explain the DFS algorithm to you!

**System:** Great! I’m eager to learn about the Depth-First Search (DFS) algorithm. Could you start by explaining what the main purpose of this algorithm is?

**User:** DFS can be coded iteratively or recursively. It consists of starting at the top node of the graph (the root) and adding it to an imaginary stack. Then the node is popped from the stack and marked visited, adding its children to the stack, the right one at the bottom. It then pops the element of the stack one by one, and adds the node’s children when it’s popped out.

**System:** I see, so DFS uses a stack to keep track of nodes to visit next, and it always visits the child nodes before the siblings. Could you elaborate on how DFS decides which node to visit next? And what happens if a node doesn’t have any children?

**User:** If a node doesn’t have children, the algorithm keeps popping the next element out of the stack. If no elements are left and the last element has no children, the algorithm ends.

**System:** I see, so when a childless node is encountered, the algorithm moves on.

...

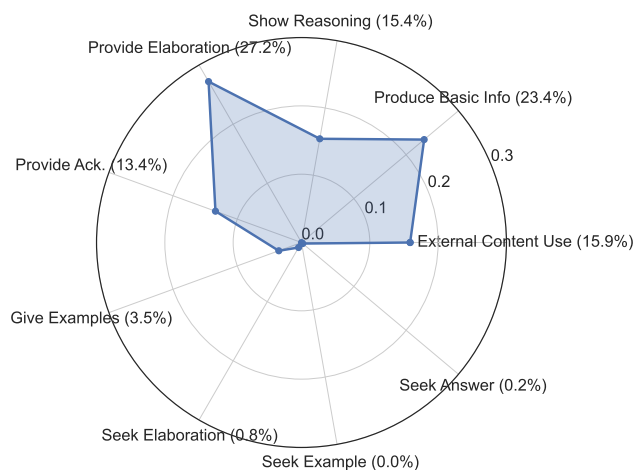
Figure 5: Example of a *Detailed* LBT Interaction.

**Trajectory of LBT Conversation Turns.** Although students exhibited various interaction patterns during the learning-by-teaching activity, their engagement was operationalised by the number of conversation turns with Algorithm Apprentice and remained relatively stable throughout the semester. As shown in Figure 4, students averaged about eight conversational turns per activity each week (ranging from 6.8 to 9.7), indicating that students consistently engaged with Algorithm Apprentice to teach and explain the given topic with multi-step dialogue. Additionally, we analysed the average word count per student message, which rose from roughly 31 words in the early weeks to over 70 words by Week 11. However, this may not be strong evidence that students gradually elaborated

their reasoning more fully, because this measure is confounded by the increased use of external content.

**Key takeaway:** Our approach effectively enabled students to act as teachers, with stable engagement of about eight multi-turn interactions per activity. Student interactions with the AI teachable agent clustered into three broad patterns—*Detailed*, *Minimal*, and *External Content Use*—that changed over time: *Detailed* explanations were most common early in the semester but declined, *Minimal* interactions remained relatively stable, and *External Content Use* increased toward the end of the semester. This indicates that while our system successfully prompted students' LBT dialogue at scale, sustaining cognitive effort and preventing circumvention are central design challenges.

#### 4.2 RQ2: How do students' LBT activities with the AI teachable agent enhance their conceptual understanding of course content?

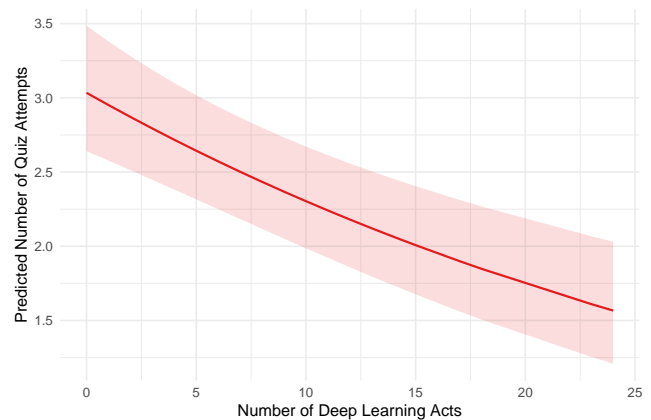


**Figure 6: Proportion of student Learning-by-Teaching dialogue acts.**

To address RQ2, we quantitatively analysed the student LBT dialogue acts derived from our qualitative coding, investigated their impact on student's follow up quiz performance via Mixed Effect Modelling, and compared it with the baseline reading condition.

**The Impact of Student LBT Dialogue Acts.** Overall, among 35,702 student messages, 32,585 were on-topic and relevant messages, as identified by the *On Topic* label, and were retained for subsequent analyses. The proportion of each LBT dialogue act is summarised in Figure 6, providing a fine-grained characterisation of how students enacted the teaching role when interacting with the Algorithm Apprentice. The most prevalent dialogue act was *Provide Elaboration* (27.2%), indicating that over one quarter of student acts involved extending explanations beyond minimal responses, such as clarifying intermediate steps or unpacking algorithmic logic. This

was followed by *Produce Basic Information* (23.4%), where students supplied concise factual statements (e.g., definitions, properties, or single-step operations). Notably, *External Content Use* accounted for 15.9% of messages, reflecting instances in which students directly use externally sourced material in their explanations. Acts explicitly oriented toward reasoning processes—*Show Reasoning*—comprised 15.4% of the messages, suggesting that a substantial portion of interactions involved articulating procedural or causal logic. In contrast, more understanding confirmation-oriented contributions, such as *Provide Acknowledgment*, represented 13.4%. Dialogue acts associated with generative pedagogical moves were comparatively rare: *Give Examples* constituted 3.5%, while *Seek Elaboration* (0.8%), *Seek Answer* (0.2%), and *Seek Example* (0.01%) occurred only sparsely.



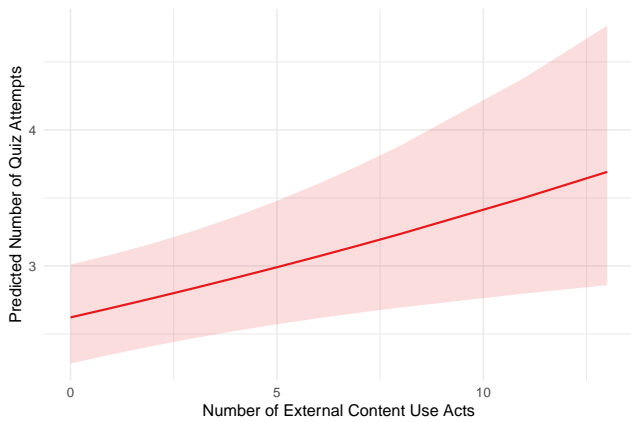
**Figure 7: The predicted number of quiz attempts in relation to the number of deep learning acts during the LBT dialogue. Note that the students can attempt the quiz an unlimited number of times in order to pass the quiz.**

**Generalised Linear Mixed Effect Model.** To investigate the impact of those acts on students' conceptual understanding, we employed a generalised linear mixed effect model (GLMM) with a negative binomial distribution to examine the relationship between students' deep learning acts (i.e., *Show Reasoning*, *Provide Elaboration*, and *Give Examples*) during LBT interactions and the number of quiz attempts. Specifically, we modelled the fixed effect of deep learning acts, with random intercepts for student and week number. This approach is appropriate for count data and accounts for overdispersion and the nested structure of observations within students ( $n = 453$ ) and weeks ( $n = 11$ ).

Descriptive statistics showed that students made an average of 2.9 quiz attempts ( $SD = 3.33$ ) and engaged in an average of 4.06 deep learning acts ( $SD = 3.03$ ) during the student-agent LBT exercises. The negative binomial GLMM revealed a significant relationship between deep learning acts and quiz attempts ( $\beta = -0.028$ ,  $SE = 0.006$ ,  $z = -4.85$ ,  $p < .001$ ). The incidence rate ratio (IRR) was 0.973 (95% CI [0.962, 0.984]), indicating that each additional deep learning act was associated with a 2.7% decrease in the expected number of quiz attempts, as shown in Figure 7. To gain lower-level insights, the same model was also fitted on the two major

*Provide Elaboration* and *Show Reasoning* acts, with both significant relationships  $\beta = -0.049$ ,  $SE = 0.019$ ,  $z = -4.90$ ,  $p < .001$  and  $\beta = -0.059$ ,  $SE = 0.013$ ,  $z = -4.63$ ,  $p < .001$ , respectively.

To address the possibility that the observed relationship reflects general engagement rather than deep learning dialogue acts, we estimated an additional negative binomial GLMM including the number of conversational turns as a control variable. The model retained random intercepts for student and week. The effect of deep learning acts remained significant after controlling for the engagement ( $\beta = -0.027$ ,  $SE = 0.006$ ,  $z = -4.59$ ,  $p < .001$ ). In contrast, the total number of turns was not significantly associated with quiz attempts ( $\beta = -0.002$ ,  $SE = 0.006$ ,  $z = -0.26$ ,  $p = .794$ ), suggesting the effect is not explained by the overall engagement.



**Figure 8: The predicted number of quiz attempts in relation to the number of *External Content Use* acts during the LBT dialogue.**

In addition, considering that around a quarter of conversations were identified with *External Content Use*, we fitted a separate negative binomial GLMM predicting the number of quiz attempts from the number of message-level *External Content Use* acts, with random intercepts for student and week to account for repeated measures. The model revealed a significant positive association between *External Content Use* and the number of attempts ( $\beta = 0.026$ ,  $SE = 0.010$ ,  $z = 2.75$ ,  $p = .006$ ), indicating that each additional *External Content Use* act was associated with a 2.7% increase in the expected number of quiz attempts, as shown in Figure 8. This pattern suggests that greater reliance on externally sourced content was associated with more repeated attempts, which may reflect less stable conceptual understanding compared to interactions characterised by students' own elaborated explanations.

**Comparison with Baseline Condition.** To investigate whether the LBT exercise with Algorithm Apprentice can enhance students' conceptual understanding towards the algorithms in comparison with the baseline reading exercise, we examine the difference in the number of quiz attempts between the two conditions.

**Generalised Linear Mixed Effect Model.** We fitted a negative binomial GLMM model predicting the number of attempts from two conditions, with random intercepts for student and week to account

for repeated measures. The model indicated a significant effect of condition ( $\beta = -0.050$ ,  $SE = 0.020$ ,  $z = -2.53$ ,  $p = .012$ ). Specifically, compared to the baseline reading condition, the LBT exercise condition was associated with a lower expected number of attempts. Exponentiating the coefficient yields an incidence rate ratio (IRR) of 0.95, corresponding to an approximate 4.9% decrease in attempts.

To contextualise the comparison between conditions, we also examined students' time-on-task. The mean time spent on the LBT exercise was  $M = 14.38$  minutes ( $SD = 10.66$ ), whereas the baseline reading condition showed a substantially lower mean time-on-task ( $M = 3.42$  minutes,  $SD = 7.56$ ). This large discrepancy suggests that students spent considerably more time engaging with the LBT activity than with the reading exercise. However, this comparison should be interpreted with caution. The reading condition allowed students to skim or rush through the material with a continue button, potentially underestimating actual cognitive engagement. By contrast, the LBT condition required sustained interaction with the Algorithm Apprentice until it determined that a sufficient explanation had been given. Therefore, differences in quiz attempts between conditions may partially reflect differences in time-on-task rather than purely the instructional format.

**Key takeaway:** Students' learning-by-teaching (LBT) behaviours during interactions with Algorithm Apprentice were associated with improved quiz efficiency. In the LBT condition, each additional deep learning act predicted fewer quiz attempts (IRR = 0.973; 95% CI [0.962, 0.984]), with similar negative associations for *Provide Elaboration* and *Show Reasoning*. Moreover, compared to a baseline reading activity, the LBT exercise corresponded to a modest reduction in expected attempts (IRR = 0.95), although this difference should be interpreted cautiously because students spent substantially more time on the LBT activity than on the reading activity. In contrast, a higher frequency of *External Content Use* dialogue acts was positively associated with the number of quiz attempts (IRR = 1.03), indicating that greater reliance on externally sourced content corresponded to slightly more quiz attempts.

## 5 Discussion

Recent work has started to explore LLM-based teachable agents in computer science education, suggesting that role-flipped interactions may enhance student learning [8, 11, 18, 32]. However, much of the existing evidence comes from small-scale or short-term deployments, leaving open how students' learning-by-teaching (LBT) interactions with an LLM-based teachable agent evolve over time and how such interactions relate to conceptual understanding in authentic, high-enrollment courses. Addressing these gaps, our semester-long deployment of Explique in a 546-student algorithms course provides longitudinal, at-scale evidence aligned with our research questions. The study results revealed that 1) students consistently engaged in multi-turn teaching dialogues throughout the semester, while the interaction depth and pattern varied over time (RQ1); 2) students' specific LBT dialogic behaviours (e.g., more deep learning acts, including *Provide Elaboration* and *Show Reasoning*) were associated with improved quiz efficiency, and the LBT condition showed statistically significant but modest advantages relative

to a baseline reading activity, though interpretation is tempered by differences in time-on-task (RQ2).

### 5.1 At-Scale Interaction with a Teachable Agent in an Algorithms Course

Our study demonstrates how LBT exercises with teachable agents can be scaled to facilitate algorithm learning in settings of 500+ students. In smaller classes, instructors can occasionally ask students to teach peers or present solutions; in a cohort of over 500 students, such practices are logistically challenging. By contrast, our deployment of the Algorithm Apprentice teachable agent demonstrates that this approach can operationalise LBT at scale beyond prior small-scale trials [5, 23]: over 11 weeks, more than 500 students engaged relatively consistently, averaging about eight multi-turn interactions per activity and completing one LBT activity per week.

Overall, more than half of student–agent LBT dialogues were identified as *Detailed*, indicating that Algorithm Apprentice was able to engage students in in-depth teaching dialogues. This detailed teaching behaviour declined from 75.4% in Week 1 but stabilised around 45–55% in later weeks, often reflecting deeper reasoning and reflection, consistent with findings on tutor learning [33]. Notably, *External Content Use* increased toward the end of the semester. Although these interactions were less common overall (27.2%), they highlight the need for system designs that discourage cognitive short-cutting. As a result, future versions of Explique might consider disabling the pasting function.

### 5.2 Student Learning-by-Teaching Acts

To characterise how students enacted LBT with Algorithm Apprentice, we examined the distribution of dialogue acts summarised in the Figure 6. The dominant pattern reflects an “explain-to-teach” strategy: students primarily advanced the interaction by stating relevant definitions and procedures, elaborating key steps, and making reasoning explicit, consistent with prior work arguing that explanation is a central mechanism through which teaching activities support learning [9, 14, 33].

In contrast, more agent-directed teaching moves—such as posing Socratic questions to diagnose the agent’s understanding or requesting examples to probe its reasoning—were comparatively rare. This imbalance may reflect pragmatic constraints of a required, text-based activity (e.g., an emphasis on completion to unblock the quiz) that make exploratory pedagogical moves less likely, but it may also reflect the side impact of system’s prompt design: because Algorithm Apprentice was instructed to persistently request clarification (Methods Section 3.1; constraint (b)), students could reasonably interpret the task as being completed primarily through explanation rather than through broader instructional strategies. At the same time, whether these less-frequent strategies would reliably produce additional learning benefits in LLM-mediated settings remains uncertain. While questioning and diagnosing understanding can be productive in human tutoring, an LLM agent may answer in ways that reduce students’ need to monitor understanding. These findings motivate future work that experimentally tests designs that more strongly cue diagnostic questioning and example-based probing, and evaluates when the additional effort yields measurable learning gains.

### 5.3 Enhancing Conceptual Understanding Through Student–AI Learning-by-Teaching

Concerning the pedagogical benefits of student–AI LBT dialogues for enhancing conceptual understanding of algorithms, our results suggest that explanation-focused teaching behaviours are meaningfully related to students’ performance on course quizzes. Specifically, the negative binomial GLMM showed that a higher frequency of deep learning acts during LBT dialogues—including mainly *Provide Elaboration* and *Show Reasoning*—was associated with fewer quiz attempts, a pattern consistent with the interpretation that sustained explanation and justification can support more efficient problem solving. Students’ qualitative feedback also suggests a mechanism: explaining to Algorithm Apprentice prompted students to restate concepts and notice gaps. As one student noted, “Having to reformulate the contents in my own words not only reviews the content but also reveals flaws in my understanding.” Such reflections echo previous research showing that explaining and grappling with difficulty strengthens conceptual mastery [9, 34]. In this sense, our study provides longitudinal, at-scale evidence that explanation-centred LBT activities can be integrated into a large algorithms course and are associated with indicators of improved conceptual understanding.

Beyond these correlational insights, the between-condition comparison provides suggestive (but not definitive) evidence of a causal effect. We observed a statistically significant difference between instructional formats: relative to a baseline reading activity, the LBT condition was associated with a modest reduction in expected quiz attempts. However, this comparison should be interpreted cautiously because the reading activity did not equate time-on-task or interaction demands; students spent substantially more time in the LBT condition, in part because the activity required continued dialogue until the agent judged the explanation sufficient. Future work should therefore employ stronger control conditions—for example, time-matched activities, or designs that isolate LBT from exposure duration—to more precisely identify which components of student–AI LBT drive learning gains.

In addition, our LBT dialogue act analysis found that *External Content Use* was positively associated with the number of quiz attempts (Figure 8). This pattern suggests that greater reliance on externally sourced content was associated with more repeated attempts, which may reflect less stable conceptual understanding compared to interactions characterised by students’ own elaborated explanations. Indeed, such copy-paste behaviour might directly reduce the cognitive load. By contrast, a few students who acknowledged this tactic argued that using external content can be an acceptable teaching approach, in that the effort of searching for and processing information may also have learning benefits (“I think it’s okay that I can fool it with ChatGPT, because, regardless, it forces me to learn about the concept.”).

### 5.4 Design and Practice Implications for Scaling AI Teachable Agents

Our findings suggest two main design and practice implications for deploying AI teachable agents in large computer science courses. First, *design for effortful explanation while managing circumvention*: a role-flipped agent that resists answer-giving can reliably elicit

explanation, but scaling also introduces incentives for low-effort completion (e.g., external content reuse); deployments should combine interaction design (e.g., discouraging copy-paste, prompting for personalised reasoning) with course policies and transparency to support authentic engagement. Second, *make desired teaching moves legible and learnable*: in our implementation, prompt constraints and completion criteria might implicitly cue an “explain-only” strategy. If other teaching moves are desired in the future design, interfaces and prompts should explicitly model and reward moves that go beyond explanation (e.g., using concrete examples, contrasting cases, or checking understanding) when appropriate, rather than assuming students will adopt them spontaneously.

## 5.5 Limitations and Future Work

Several limitations must be acknowledged. First, the study took place in a single undergraduate algorithms course at a technical university, with second-year computer science students. The findings may not generalise to other institutions, class sizes, or domains—particularly those that emphasise open-ended reasoning, creative synthesis, or qualitative argumentation. Second, participation was self-directed and occurred asynchronously outside controlled settings. This introduces substantial variability in learners’ context and tool use, limiting causal claims; while our modelling included random effects to account for between-student and between-week differences, unobserved factors may still influence both engagement and outcomes. Third, the uncontrolled nature of the deployment means that we could not directly measure or manipulate students’ prior knowledge of each weekly topic, yet prior knowledge is likely to shape both the quality and frequency of explanations and subsequent quiz performance. Also, the number of quiz attempts may not be a precise indicator of conceptual understanding. Future work could incorporate topic-level pretests and confidence checks to better characterise how initial understanding moderates the benefits of student–AI LBT.

## 6 Conclusion

This work presented Explique, an application that integrates an AI teachable agent (Algorithm Apprentice) into a large undergraduate algorithms course to operationalise learning-by-teaching (LBT) at scale. In an 11-week field deployment with 546 students and 3,809 student–agent dialogues, we found that students consistently engaged in multi-turn LBT interactions, but that the depth and authenticity of explanations varied over time, including instances of externally sourced content. We further found that explanation-oriented LBT behaviours (e.g., providing elaboration and showing reasoning) were associated with more efficient quiz performance, whereas reuse of external content was associated with more repeated attempts—suggesting trade-offs between scalable participation and authentic cognitive effort. Relative to a baseline reading activity, the LBT condition was associated with a modest reduction in expected quiz attempts, though this difference should be interpreted cautiously given substantial differences in time-on-task. Overall, these results provide longitudinal, large-scale evidence on LLM-based teachable agents in an authentic course context and highlight design and practice directions for future systems that better support sustained, effortful, and scalable LBT interactions.

## Acknowledgments

This work was partially funded by the Swiss State Secretariat for Education, Research, and Innovation (SERI). Miltiadis Stouras and Ola Svensson are supported by the SERI under contract number MB22.00054. Paola Mejia-Domenzain and Vinitra Swamy are supported by Innosuisse.

## References

- [1] 2024. Powering virtual education for the classroom. <https://openai.com/index/khan-academy/>
- [2] 2025. AI-powered tutor tested as a way to help educators and students | 60 Minutes - CBS News. <https://www.cbsnews.com/news/khanmigo-ai-powered-tutor-teaching-assistant-tested-at-schools-60-minutes-transcript/>
- [3] Ambroise Baillifard, Maxime Gabella, Pamela Banta Lavenex, and Corinna S. Martarelli. 2023. Implementing Learning Principles with a Personal AI Tutor: A Case Study. doi:10.48550/arXiv.2309.13060 arXiv:2309.13060 [cs].
- [4] Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakçı, and Rei Mariman. 2024. Generative AI Can Harm Learning. doi:10.2139/ssrn.4895486
- [5] Gautam Biswas, Krittaya Leelawong, Daniel Schwartz, Nancy Vye, and The Teachable Agents Group at Vanderbilt. 2005. Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence* 19, 3-4 (2005), 363–392.
- [6] Robert A. Bjork. 1994. Memory and Metamemory Considerations in the Training of Human Beings. In *Metacognition*, Janet Metcalfe and Arthur P. Shimamura (Eds.). The MIT Press, 185–206. doi:10.7551/mitpress/4561.003.0011
- [7] Catherine C Chase, Doris B Chin, Marily A Oppezzo, and Daniel L Schwartz. 2009. Teachable agents and the protégé effect: Increasing the effort towards learning. *Journal of science education and technology* 18 (2009), 334–352.
- [8] Angxuan Chen, Yuang Wei, Huixiao Le, and Yan Zhang. 2024. Learning by teaching with ChatGPT: The effect of teachable ChatGPT agent on programming education. *British Journal of Educational Technology* (2024).
- [9] Michelene T.H. Chi, Miriam Bassok, Matthew W. Lewis, Peter Reimann, and Robert Glaser. 1989. Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science* 13, 2 (April 1989), 145–182. doi:10.1207/s15516709cog1302\_1
- [10] Michelene TH Chi, Stephanie A Siler, Heisawn Jeong, Takashi Yamauchi, and Robert G Hausmann. 2001. Learning from human tutoring. *Cognitive science* 25, 4 (2001), 471–533.
- [11] Dan Davies, Divya Jindal-Snape, Chris Collier, Rebecca Digby, Penny Hay, and Alan Howe. 2013. Creative learning environments in education—A systematic literature review. *Thinking Skills and Creativity* 8 (April 2013), 80–91. doi:10.1016/j.tsc.2012.07.004
- [12] Ruiqi Deng, Yi Yang, and Suqin Shen. 2025. Impact of question presence and interactivity in instructional videos on student learning. *Education and Information Technologies* 30, 2 (Feb. 2025), 1635–1663. doi:10.1007/s10639-024-12862-1
- [13] David Duran. 2017. Learning-by-teaching. Evidence and implications as a pedagogical mechanism. *Innovations in education and teaching international* 54, 5 (2017), 476–484.
- [14] Logan Fiorella and Richard E. Mayer. 2013. The relative benefits of learning by teaching and teaching expectancy. *Contemporary Educational Psychology* 38, 4 (Oct. 2013), 281–288. doi:10.1016/j.cedpsych.2013.06.001
- [15] Thomas Gargot, Thibault Asselborn, Ingrid Zammouri, Julie Brunelle, Wafa Johal, Pierre Dillenbourg, Dominique Archambault, Mohamed Chetouani, David Cohen, and Salvatore M Anzalone. 2021. “It Is Not the Robot Who Learns, It Is Me.” Treating Severe Dysgraphia Using Child–Robot Interaction. *Frontiers in Psychiatry* 12 (2021), 596055.
- [16] A.C. Graesser, P. Chipman, B.C. Haynes, and A. Olney. 2005. AutoTutor: An Intelligent Tutoring System With Mixed-Initiative Dialogue. *IEEE Transactions on Education* 48, 4 (Nov. 2005), 612–618. doi:10.1109/TE.2005.856149
- [17] Sara Hennessy, Sylvia Rojas-Drummond, Rupert Higham, Ana María Márquez, Fiona Maine, Rosa María Ríos, Rocío García-Carrión, Omar Torreblanca, and María José Barrera. 2016. Developing a coding scheme for analysing classroom dialogue across educational contexts. *Learning, culture and social interaction* 9 (2016), 16–44.
- [18] Hyoungwook Jin, Seonghee Lee, Hyungyu Shin, and Juho Kim. 2024. Teach ai how to code: Using large language models as teachable agents for programming education. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–28.
- [19] Lingxi Jin, Baicheng Lin, Mengze Hong, Kun Zhang, and Hyo-Jeong So. 2025. Exploring the impact of an LLM-Powered teachable agent on learning gains and cognitive load in music education. *arXiv preprint arXiv:2504.00636* (2025).
- [20] Gregor Jošt, Viktor Taneski, and Sašo Karakatič. 2024. The impact of large language models on programming education and student learning outcomes. *Applied Sciences* 14, 10 (2024), 4115.

- [21] Manu Kapur. 2016. Examining Productive Failure, Productive Success, Unproductive Failure, and Unproductive Success in Learning. *Educational Psychologist* 51, 2 (April 2016), 289–299. doi:10.1080/00461520.2016.1155457
- [22] Manu Kapur and Katerine Bielaczyc. 2012. Designing for Productive Failure. *Journal of the Learning Sciences* 21, 1 (Jan. 2012), 45–83. doi:10.1080/10508406.2011.591717
- [23] Krittaya Leelawong and Gautam Biswas. 2008. Designing learning by teaching agents: The Betty's Brain system. *International Journal of Artificial Intelligence in Education* 18, 3 (2008), 181–208.
- [24] Douglas B Lenat and Paula J Durlach. 2014. Reinforcing math knowledge by immersing students in a simulated learning-by-teaching experience. *International Journal of Artificial Intelligence in Education* 24, 3 (2014), 216–250.
- [25] 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 (Dec. 2017), 693–715. doi:10.1007/s10648-016-9379-x
- [26] Qianou Ma, Hua Shen, Kenneth Koedinger, and Sherry Tongshuang Wu. 2024. How to teach programming in the ai era? using llms as a teachable agent for debugging. In *International Conference on Artificial Intelligence in Education*. Springer, 265–279.
- [27] Fumiaki Obayashi, Hiroshi Shimoda, and Hidekazu Yoshikawa. 2000. Construction and evaluation of CAI system based on learning by teaching to virtual student. In *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Vol. 3. 94–99.
- [28] Amy Ogan, Samantha Finkelstein, Elijah Mayfield, Claudia D'adamo, Noboru Matsuda, and Justine Cassell. 2012. "Oh dear stacy!" social interaction, elaboration, and learning with teachable agents. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 39–48.
- [29] Aadarsh Padiyath, Xinying Hou, Amy Pang, Diego Viramontes Vargas, Xingjian Gu, Tamara Nelson-Fromm, Zihan Wu, Mark Guzdial, and Barbara Ericson. 2024. Insights from social shaping theory: The appropriation of large language models in an undergraduate programming course. In *Proceedings of the 2024 ACM Conference on International Computing Education Research-Volume 1*. 114–130.
- [30] R. Ploetzner, P. Dillenbourg, M. Praier, and D. Traum. 1999. Learning by explaining to oneself and to others. In *P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches*. Elsevier, Oxford, 103–121. <https://infoscience.epfl.ch/handle/20.500.14299/212708>
- [31] James Prather, Paul Denny, Juho Leinonen, Brett A Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, et al. 2023. The robots are here: Navigating the generative ai revolution in computing education. In *Proceedings of the 2023 working group reports on innovation and technology in computer science education*. 108–159.
- [32] Kantwon Rogers, Michael Davis, Mallesh Maharana, Pete Etheredge, and Sonia Chernova. 2025. Playing Dumb to Get Smart: Creating and Evaluating an LLM-based Teachable Agent within University Computer Science Classes. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–22.
- [33] Rod D. Roscoe and Michelene T. H. Chi. 2007. Understanding Tutor Learning: Knowledge-Building and Knowledge-Telling in Peer Tutors' Explanations and Questions. *Review of Educational Research* 77, 4 (Dec. 2007), 534–574. doi:10.3102/0034654307309920
- [34] Daniel L. Schwartz and John D. Bransford. 1998. A Time For Telling. *Cognition and Instruction* 16, 4 (Dec. 1998), 475–5223. doi:10.1207/s1532690xcii1604\_4
- [35] Fumihide Tanaka and Shizuko Matsuzoe. 2012. Children teach a care-receiving robot to promote their learning: field experiments in a classroom for vocabulary learning. *J. Hum.-Robot Interact.* 1, 1 (July 2012), 78–95. doi:10.5898/JHRI.1.1.Tanaka
- [36] Kurt VanLEHN. 2011. The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist* 46, 4 (Oct. 2011), 197–221. doi:10.1080/00461520.2011.611369
- [37] Alessandro Vanzo, Sankalan Pal Chowdhury, and Mrinmaya Sachan. 2024. GPT-4 as a Homework Tutor can Improve Student Engagement and Learning Outcomes. doi:10.48550/arXiv.2409.15981 arXiv:2409.15981 [cs].
- [38] Chenyang Wang, Filippo Forte, Léane Wettstein, Pierre Dillenbourg, and Thimo Wambsganss. 2026. ArgueMate: Designing an Arguing Agent with Maximised Disagreement to Support Student Peer-Argumentation Exercise. In *Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26)*. Association for Computing Machinery, New York, NY, USA, Article 95, 9 pages. doi:10.1145/3772363.3799300
- [39] Chenyang Wang, Julien Jordan, Alice Reymond, and Pierre Dillenbourg. 2026. Social Robot Teaching as a Pathway to Children's AI Literacy. In *Companion Proceedings of the 21st ACM/IEEE International Conference on Human-Robot Interaction (Edinburgh, Scotland, UK) (HRI Companion '26)*. Association for Computing Machinery, New York, NY, USA, 599–603. doi:10.1145/3776734.3794465
- [40] Chenyang Wang, Yujing Zhang, Lucas Burget, and Pierre Dillenbourg. 2025. Learning by Teaching: Designing Teachable Agents to Support Children's Pronunciation Skill Learning. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '25)*. Association for Computing Machinery, New York, NY, USA, Article 374, 8 pages. doi:10.1145/3706599.3719863
- [41] Ryan C. Yeung and Myra A. Fernandes. 2022. Machine learning to detect invalid text responses: Validation and comparison to existing detection methods. *Behavior Research Methods* 54, 6 (Feb. 2022), 3055–3070. doi:10.3758/s13428-022-01801-y