AvoidingHelpAvoidance manuscript No.
(will be inserted by the editor)

Avoiding Help Avoidance: Using Interface Design Changes to Promote Unsolicited Hint Usage in an Intelligent Tutor

Mehak Maniktala · Christa Cody · Tiffany Barnes · Min Chi

the date of receipt and acceptance should be inserted later

Abstract Within intelligent tutoring systems, considerable research has investigated hints, including how to generate data-driven hints, what hint content to present, and when to provide hints for optimal learning outcomes. However, less attention has been paid to *how* hints are presented. In this paper, we propose a new hint delivery mechanism called "Assertions" for providing unsolicited hints in a data-driven intelligent tutor. Assertions are *partially-worked* example steps designed to appear within a student workspace, and in the same format as student-derived steps, to show students a possible subgoal leading to the solution. We hypothesized that Assertions can help address the well-known *hint avoidance* problem. In systems that only provide hints upon request, hint avoidance results in students not receiving hints when they are needed. Our unsolicited Assertions do not seek to improve student help-seeking, but rather seek to ensure students receive the help they need. We contrast Assertions with Messages, text-based, unsolicited hints that appear after student inactivity. Our results show that Assertions significantly increase unsolicited hint usage compared to Messages. Further, they show a significant aptitude-treatment interaction between Assertions and prior proficiency, with Assertions leading students with low prior proficiency to generate shorter (more efficient) posttest solutions faster. We also present a clustering analysis that shows patterns of productive persistence among students with low prior knowledge when the tutor provides unsolicited help in the form of Assertions. Overall, this work provides encouraging evidence that hint presentation can significantly impact how students use them and using Assertions can be an effective way to address help avoidance.

M. Maniktala · C. Cody · T. Barnes · M. Chi
Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA
E-mail: mmanikt@ncsu.edu

Keywords intelligent tutoring system · help avoidance · user experience · unsolicited hints · aptitude-treatment interaction · logic proofs · productive persistence · clustering · problem solving

1 Introduction

Studies suggest that hints, when provided appropriately, can augment students' learning experience [15, 68] and improve their performance [11]. However, students may not use hints optimally [2, 31]; some abuse hints to expedite problem completion, and some avoid seeking help when they are in need [1, 65]. Our goal is to redesign the hint interface to solve this help avoidance problem. Considerable research has investigated hints from several perspectives, including hint generation [10, 63], adaptive hint content [22, 41, 83], student help-seeking behavior [2,65], and hint timing [69]. However, few studies have specifically investigated how hint interfaces could reduce help avoidance (e.g. [41, 50]).

Most intelligent tutoring systems (ITSs) provide solicited hints *on-demand*, i.e, upon student request [84]. Other tutors try to circumvent help avoidance by providing *unsolicited* hints when the system "determines" they are needed, for example, after a long period of inactivity [32]. However, students often ignore these unsolicited hints [22,55]. In this work, we designed a new interface for unsolicited hints, called *Assertions* to address this issue, and compared its impact on student learning outcomes with that of *Messages*, text-based unsolicited hints that appear after student inactivity. The ultimate goal of our research is to combine the new Assertions interface with a data-driven method to determine when providing an unsolicited hint would be most beneficial and least disruptive for students.

Our Assertions interface was designed based on user experience and multimedia design principles, including contiguity [50], attention [34], expectation [76], and persuasion [20, 29]. First and foremost, we hypothesized that placing Assertions *contiguously* within the area of student *attention* would make unsolicited hints more noticeable. Second, we believed students could more quickly interpret Assertions based on the *expectation* set by formatting them like other problem-solving steps. Finally, we used *persuasive* language asking students to use the Assertions as problem-solving subgoals. These features help Assertions act as partially-worked example steps, so they may garner the same benefits of worked examples, that have been shown to improve learning efficiency [49]. We hypothesized that Assertions would reduce help avoidance for all students, by increasing the percentage of times help was received when it was needed. Further, we hypothesize that Assertions would have an aptitude-treatment interaction effect, fostering productive persistence and improving posttest performance, among students with low prior proficiency. Persistence during training that leads to mastery of a subject or positive posttest outcomes is called productive persistence [39].

The main contribution of this work is a principled design for a hint interface, Assertions, and a study to show that Assertions can be used to significantly reduce help avoidance for all students through interface alone. Our new proposed Assertions appear as partially-worked example steps, reducing the barriers to help usage while leveraging benefits of worked examples. The second contribution of this work is a new cluster-based method that combines posttest performance, effort (to quantify persistence), and unsolicited hint usage to discover productive persistence. Based on these clusters, we were able to show that students with low prior proficiency who received Assertions exhibit productive persistence. Since Assertions are automatically provided to students, they can be thought of from two perspectives: either as unsolicited hints, or as partially worked example steps. Therefore, in our related work and design sections below, we discuss Assertions from both of these perspectives.

2 Related Work

2.1 Hints in ITSs

ITSs have the unique ability to offer individualized help and feedback. While research suggests that such individual adaptation can significantly improve learning, they can take a considerable time to construct [57]. Example-based authoring tools such as CTAT often employ examples to construct production rules, where teachers work problems, predict frequent incorrect approaches, and then develop rules and manually write appropriate hints [42]. Such systems have also been augmented by methods such as Bootstrapping Novice Data, where they use data to build initial models, but still require manual expert hint authoring [48]. However, considerable time must still be spent on identifying student approaches for hint generation and to write the messages that appear with hints.

Data-driven assistance saves time and resources by reducing the need for an expert – typically such systems include a hint template, authored once for all hints, and use algorithms to generate data to include in the template-based hint. Data-driven methods have increasingly been used to generate personalized help in intelligent tutors for open-ended multi-step problem-solving domains. Barnes and Stamper invented the Hint Factory, the first data-driven method to generate next-step hints, and demonstrated its effectiveness for propositional logic [9, 75]. The approach uses prior students' transaction log data to form an interaction network and then runs the Bellman backup for value iteration to score problem-solving states (snapshots of an on-going or completed problem solution attempt). To provide hints, the Hint Factory finds states matching students' current work and delivers hints using the next reachable state with the highest score. Barnes and Stamper and their colleagues did pioneering work to explore program representations that could be used for hint generation for novices [38] and demonstrated how they could be used to provide hints for 80% of the states in a historical dataset [37]. Similar to the Hint

Factory, Fossati et al. [33] devised the Procedural Knowledge Model (PKM), that uses students' global problem-solving behaviors to generate data-driven feedback for the iList tutor for programming with linked lists. Price, Barnes, and colleagues extended the Hint Factory approach to generate data-driven hints for novice programming [63, 64, 67]. Later, Paaßen et al. created the continuous hint factory to allow for hint generation for previously unobserved states [60], while Price et al. devised the SourceCheck algorithm that leveraged similar representations to generate hints based on a set of student solutions rather than the trace data that the original Hint Factory uses [62]. Rivers et al. developed a data-driven hint generator for ITAP (Intelligent Teaching Assistant for Programming) that uses a similar set of tools including state abstraction, path construction, and state reification to generate personalized hints [70]. This method extends the Hint Factory by enhancing the solution space and creating new edges for states that are disconnected. This allows the ITAP method to generate hints even for states that are not present in the prior data. For this work, we extended the Hint Factory to provide personalized hints for logic with 100% availability as described in Section 3.

Aleven et al. have shown that students often display poor help-seeking behaviors within intelligent tutors, including *help avoidance*, where students *could* benefit from seeking help but choose not to, and *help abuse*, where students use help excessively when they could solve a problem without assistance [2]. Studies by Price et al., Almeda et al., and Roll et al. have confirmed that help avoidance is pervasive across domains and systems with students ignoring hints [4, 66, 71]. In one study, Roll et al. showed that meta-cognitive feedback improved student's help-seeking skills but did not affect their domain learning [71]. Price et al.'s research study on help-seeking by novice programmers showed that students have several reasons for not requesting on-demand hints, including uncertainty about whether system help would be useful, or a desire to be independent [66]. Some tutoring systems prevent help avoidance by providing *unsolicited hints* rather than relying on student help-seeking through "on-demand" hint requests [5, 46, 56]. Arroyo et al. [5] and Murray et al. [56] showed that unsolicited hints promoted learning gains for a subset of students. However, a study by Muir and Conati showed that students often ignore unsolicited hints [55].

Several studies have tried to encourage students to use unsolicited help by changing its content or placement. For example, Cody et al. showed that unsolicited, data-driven hints were more likely to be used if their content focused on next-step hints rather than more abstract, high-level hints [21]. Conati et al. used eye-tracking to show that factors such as hint timing, and student's attitude and prior knowledge can affect students' attention towards unsolicited hints in a number factorization game [22]. Kardan and Conati showed that unsolicited hints with tailored hint content along with highlighting and proximal hint placement improved student learning in a controlled study with AI SPACE [41].

Despite their potential benefits, we argue that attempting to understand or use hints, and especially unsolicited ones, can increase students' cognitive load while learning new concepts within a tutoring system. This is because students have to mentally integrate several sources of information, including on-demand hints, unsolicited hints, and the student's own current solution attempt. Adding to this is the fact that, in many existing tutoring systems, the hints and the student solution workspace are physically located in different areas of the interface. As a result, we believe that by physically integrating those sources of information together, Assertions naturally reduce students' working memory load and thus would facilitate student learning by accelerating the changes in their long term memory associated with schema acquisition [77,78].

2.2 Worked Examples

Since we posit that Assertions can be seen not only as unsolicited hints, but from another perspective as partially-worked examples for single problem-solving steps, we discuss impacts of worked examples here. Extensive research has shown that worked examples, i.e. showing step-by-step problem solutions, can be as effective as problem solving to learn the same content yet the former generally need much less time [49, 54]. In our prior work, we have added whole-problem worked examples to our tutor to help students learn the problem interface and problem-solving skills. In [54], we found that the students who received data-driven worked examples were much more likely to complete the tutor, and did so in less time [54]. In another study [72], we found that when we use reinforcement learning (RL) to determine when to present whole-problem worked examples, the slow learners provided based on this RL policy had a significantly higher learning gains than their peers who received worked-examples at random. Further, our results from study on worked examples in Deep Thought [43] show that whole-problem worked examples benefit students early in the tutoring, but are comparable to hint-based scaffolding. We also observed that worked examples were less beneficial later in the tutoring sessions for lower proficiency students. Our work with Pyrenees, a probability tutor, suggests that step-level Worked Examples can also promote learning [92]. This work suggests that students do not resist following these step-level worked examples, that are essentially unsolicited hints provided in student workspace.

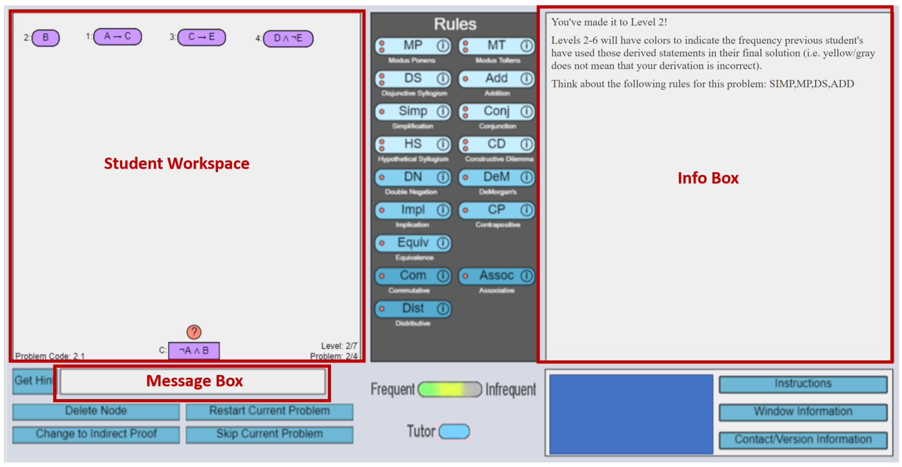
One mechanism proposed by Sweller et al. for the success of worked examples is through reduction in the cognitive load when students are learning new concepts [79]. Their work discusses the principles underlying cognitive load theory and how worked examples reduce the need for learners to engage in inference processes which might otherwise require heavy demands on students' working memory. On the other hand, much prior work found that asking students to *justify* their solution steps, referred to as self-explanations, can greatly improve their learning [3, 19, 24]. Furthermore, asking students to explain expert-designed worked examples can be more effective than problem solving alone [18,88]. For example, Weerasinghe and Mitrovic explored the impact of self-explanations in KERMIT-SE, a tutor for the open-ended domain of database design. They engaged students in tutorial dialogues upon errors in solutions and found that it improved student performance in both conceptual and procedural knowledge [88, 89]. In this work, we design our new Assertions hint interface to act as expert-designed partially-worked example steps with self-explanations. However, there are two key differences between our work and that by Weerasinghe and Mitrovic: Assertions are provided to guide students on the next step instead of the current step, and they are provided after correct steps instead of incorrect steps. As described in section 3, Assertions provide students with the content of a useful step, but students must provide an explanation before they can use the hint content in their solutions.

2.3 Aptitude-Treatment Interaction

Prior research in instructional strategies has shown the existence of aptitude-treatment interaction (ATI), where certain students are more sensitive to variations in the learning environment compared to less sensitive students who perform regardless of the treatment [25, 74]. Researchers have explored the complex relationship between student aptitude and their interaction with unsolicited help. While Razzaq et al. found that students learned more reliably with hints they requested than unsolicited hints [69], Arroyo et al. observed higher learning gains for low performing students when unsolicited hints were provided [5]. Further, Murray et al. found that unsolicited help avoided the negative effects of frustration and saved students time when they were struggling [56]. Muir and Conati showed that students with low prior knowledge are likely to need hints the most, but they do not look at the hints as often [55]. Kardan and Conati found that changes in unsolicited hint content and interface had a more pronounced effect on learning for students with lower initial knowledge [41]. Similar to these studies, we hypothesize that an improved interface for unsolicited hints can increase hint usage and outcomes, especially for students with low prior knowledge. In this work, we believe that students whose initial tutor performance is lower may need more assistance to develop strategies for solving logic proofs, and therefore, may benefit more from an improvement in the hint interface.

2.4 Productive Persistence

Recently, there is an increased interest in non-cognitive skills like persistence and self-control within education research [39]. Task persistence is defined as the continuation of a task despite difficulty. To quantify persistence, researchers used metrics of *effort* [28]. However, not all persistence is productive, Beck and Gong [12] define unproductive persistence or "wheel spinning" as when a student spends an excessively long time struggling to learn a topic without achieving mastery. They showed that if a student did not master a skill in ASSISTments (an online math learning platform) or the Cognitive Algebra Tutor in a reasonable amount of time, the student was likely to struggle and


You've made it to Level 2!

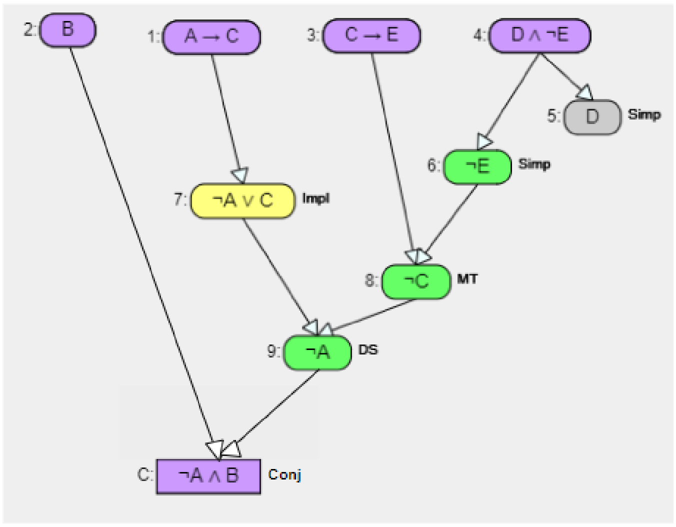
Levels 2-6 will have colors to indicate the frequency previous student's have used those derived statements in their final solution (i.e. yellow/gray does not mean that your derivation is incorrect).

Think about the following rules for this problem: SIMP,MP,DS,ADD

Fig. 1: Tutor's Interface: Student *workspace* (left), rules (middle), info box (right), and the *Hint button* and *message box* (bottom-left)

never master the skill. Their work presents connections between wheel-spinning and negative student behaviors such as disengagement and gaming, as well as recommendations to improve ITS design to address these issues. Research by Nelson et al. is well-known for their heuristic model of the help-seeking process where they suggest that unproductive persistence may be associated with help avoidance [58]. Studies suggest that the persistent effort that lead to mastery of a topic is *productive persistence* [39], and is often associated with short-term outcomes like improvement in performance [13, 61], and longer-term outcomes in higher education and future earning [27, 35]. Recent studies in educational data mining have attempted to predict when an intervention can help students by distinguishing between productive and unproductive behavior using decision trees [39] and Recurrent Neural Networks (RNN) [14]. The work by Kai et al. on ASSISTments used decision trees to identify when students are struggling and how to make students' persistence more productive. They found that interleaved practice of different skills is more advantageous than blocked practice, where the opportunities to learn a given skill are massed one after another. Another study on ASSISTments by Botelho et al. used RNNs to detect stopout (low persistence) and wheel-spinning (unproductive persistence) early to intervene and prevent unproductivity. They found that these models have high AUC and are also able to learn a set of features that generalize to predict each other. In this paper, we apply clustering to discover patterns of productivity, persistence, and unsolicited hint usage in our tutor.

Fig. 2: A sample solution of a training problem in Deep Thought



3 System Design

Deep Thought (DT, Figure 1) is an intelligent tutor for solving open-ended multi-step propositional logic problems that has data-driven features including next-step hints [8, 75], as well as adaptive problem selection [51, 53] and pedagogical policies for worked example presentation induced via reinforcement learning [6, 52, 72, 73]. Figure 1 shows the current tutor interface: the left window is the *workspace* where students construct solutions, the central window lists the domain rule buttons, and the right window provides instructions and information such as the rules that are meant to be practiced in the current problem. Each problem-solving statement is graphically represented as a *node*. Deep Thought shows several problem-provided statements (that are meant to be used as existing or known facts) at the top of the workspace, and a conclusion to derive at the bottom. Students iteratively carry out problem-solving steps by deriving new statements from old ones using domain rules. This is a typical procedure used across STEM domains to apply principles or rules to known information to derive new facts [59]. For example, in physics, if we know values for mass (m) and acceleration (a), we can apply the rule $F = ma$ with those values to find force (F). In this paper, a problem-solving step consists of a new *derived statement* and its *justification*, where the justification includes specifying the domain rule and the source statements used to show that the new derived statement is true. In logic, problem-solving continues until the conclusion is the derived statement in a step that is justified.

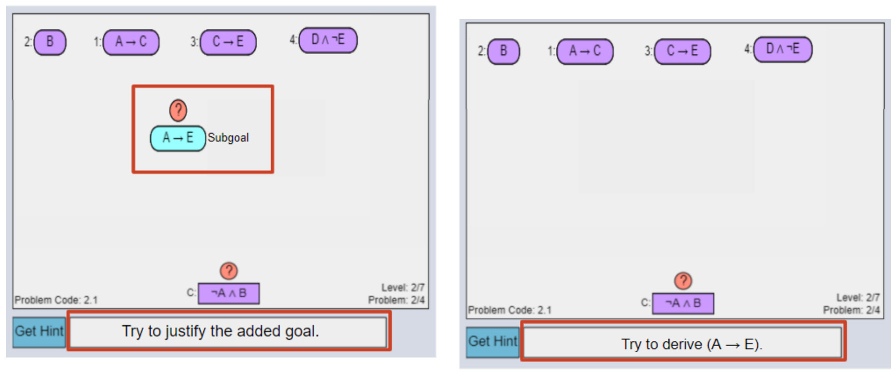
Figure 1 shows an example problem with three nodes 1-4 for the problem-provided statements (2: B, 1: $A \rightarrow C$, 3: $C \rightarrow E$), and 4: $D \wedge \neg E$ at the top of the workspace. The conclusion to be derived (C: $\neg A \wedge B$) is at the bottom, with a question mark indicating that it is not yet justified. Each problem solving step involves the same process: clicking on 1-2 source nodes and a rule button, and entering the new derived statement. The tutor verifies whether the source nodes and rule correctly justify the derived statement. Once a step is verified, a new node appears, colored based on how often the same node was necessary in previous student solutions to this problem, where green means frequent, yellow is infrequent, and gray is never. We call a node 'necessary' or 'needed' when its deletion would make a solution incomplete. These colorings give students an indication of whether they are on an optimal problem-solving path.

We now walk through the student experience of solving the problem shown in 1 to obtain the solution shown in Figure 2. First, the student clicks on node 4 and rule Simp, and is asked to type the new derived statement, D. The tutor verifies that Simp applied to node 4 is a correct justification, and draws node 5, labeled with Simp and an arrow from node 4 to 5. Node 5 is colored gray since it was never needed by previous students solving that same particular problem. Next, the student applies the same process to derive and justify node 6, which is green since it was frequently necessary in historical solutions. To derive node 7, the student clicks on node 1, and Impl rule, and types in the derived statement $\neg A \vee C$. After it is verified, node 7 appears, with the label Impl, and an arrow from node 1 to 7. The student then clicks "Get Hint" to request a hint, and "Try to derive $\neg C$" appears in the message box. Next, the student tries to follow the hint by selecting nodes 3 and 6 and the rule MP. The tutor detects this incorrect rule application, records the error in the data log, and provides an error-specific message, but since it was a mistake, no new node is created. Since nodes 3 and 6 are still selected, the student clicks on the correct rule – MT, and types in the derived statement $\neg C$. This process correctly *justified* the hint content statement $\neg C$, so node 8 appears with MT with arrows from nodes 3 and 6. The student similarly clicks on nodes 7 and 8, and rule DS to derive node 9. Finally, the student clicks on nodes 2 and 9, and rule Conj to derive the conclusion, and the tutor detects that the problem is complete.

3.1 Hints in Deep Thought

Deep Thought uses the Hint Factory [75] to generate hints, where the hint content depends only on the current problem solving state, a snapshot of a student problem-solving attempt. The Hint Factory [75] works by treating problem-solving data from prior students as a Markov Decision Process and using value iteration to assign values to each state based on its distance from a valid observed solution. Then, the *hint source* is set for a current student's state by selecting the subsequent reachable state with the highest value. If the current state is not found, we rollback current student solution states until a matching state and its hint source are found. Finally, the Hint Factory-derived **hint content** is the *newest derived statement* in the hint source state. Deep

Fig. 3: Differences between Assertions and Messages while delivering a logic hint statement $A \rightarrow E$



Try to justify the added goal.

Try to derive (A → E).

(a) *Assertion* is presented in the workspace, with the format of a student-derived step, and with a “Subgoal” label

(b) *Message* hint is provided textually below the student workspace

Thought inserts this derived statement, the hint content HC, into a template depending on the hint type, described below.

In this study, there are three types of hints, including on-demand hint requests, and two types of unsolicited hints: Messages and Assertions. The content of on-demand and unsolicited hints is identical and no additional justification/derivation help is given. Students request *on-demand* hints by clicking the “Get Hint” button, and the system shows “Try to derive HC,” in the message box. Both Messages and Assertions are unsolicited hints, meaning that they are not requested by students. *Messages* appear automatically after one minute of student inactivity, using the same Messages interface as on-demand hints. *Assertions* appear automatically after about 40% of steps. Since the mean student solution length in the training problems is 9 steps, this means that students are likely to encounter 3 - 4 hints per problem. The Assertions interface consists of 4 parts: (1) adding a new cyan-colored node containing the hint content HC in the workspace, (2) labeling the node as a “Subgoal,”, (3) including a question mark icon showing that the node is not yet justified, and (4) stating “Try to justify the added goal” in the message box. Figure 3 shows the Messages and Assertions interfaces suggesting the same logic statement $A \rightarrow E$ in different formats. Students must *explain* how the node is to be derived by *justifying* it before they can use the hint content in their solutions. While this is not a typical verbal self-explanation, we argue that, by justifying the step, the student is demonstrating that they know what domain principle (rule) and prior statements can be used to *explain* why the new derived statement is true. In the next section, we describe the design principles used to create the new Assertions interface.

3.2 Assertions Design

While there are many possible ways to encourage students to use hints more often [22], we hypothesize that our new Assertions interface changes would help all students notice and efficiently use unsolicited hints. As stated above, Assertions represent unsolicited, partially-worked example steps that appear in the workspace as shown in Figure 3a. In the remainder of this section, we discuss the design considerations that differentiate Assertions from Messages: contiguity, attention, expectation, and persuasion. All of these design changes were made to reduce students' cognitive load [80], and we hypothesized that these changes would reduce help avoidance as measured by increased hint usage. We further hypothesized that Assertions may have an aptitude-treatment interaction, with students with low prior knowledge benefiting more from the interface changes.

- **Contiguity and Attention**: Moreno, et al.'s *spatial contiguity* principle for multimedia learning materials states that a graphic should not be physically separated from its explanatory text [26, 50]. Hegarty et al. showed that contiguity supports student memory and understanding [36]. Butcher and Aleven showed that when interactive support was placed near a geometry diagram, student learning outcomes improved [16, 17]. Kardan and Conati in a controlled study on AI SPACE tailored the hint content, used hint highlighting and proximal hint placement to gain students attention towards unsolicited hints that improved learning for low prior knowledge students [41]. In this work we use similar proximal hint placement for Assertions but provide the same content in both Assertions and Messages. We strategically place Assertion hints where the student needs them. Although the message box is close to the workspace, it may still be subject to 'change blindness' [34], where students paying *attention* to nodes within the workspace may filter Messages out and simply not notice their appearance. Therefore, we provide Assertions in the workspace, where students have already focused their attention. Together, contiguity and attention are meant to help students notice the appearance of Assertion hints.
- **Expectation**: Research by Summerfield explains that the speed of visual interpretation is optimized by leveraging past experiences to form expectations [76]. Based on this principle, we design Assertions to leverage student expectations through an isomorphic visual format that may work together with reduced text to decrease cognitive load. First, the hint content HC of Assertions appears in the same visual node format as student-derived statements, enabling students to visually interpret an Assertion hint faster. Second, Messages require students to read the text "Try to derive HC" and determine that HC is a statement that should appear on a graphical node. This additional cognitive processing may pose a barrier that some students may not overcome [80], and this may be especially true for students with low prior knowledge [40]. Therefore, formatting the Assertions hint con-

tent HC as nodes may help students by leveraging visual expectation, or by reducing overall cognitive load [80].

- **Persuasion**: Dillard suggests that user experiences can be enhanced by using persuasion [29]. Cialdini has created six principles of influence, including reciprocity, commitment and consistency, liking, social proof, authority, and scarcity, that can be used to influence people's behaviors [20]. Assertions have two persuasive design aspects. First, we posit that adding Assertions directly to the workspace may make them seem required, leveraging the authority of the tutoring system itself. Assertion nodes are accompanied with a label "subgoal" (Figure 3a) and the message "Try to justify the added goal", persuasive and authoritative texts suggesting that justifying Assertions is just part of the tutor. The difference in the text accompanying Assertions and Messages is that an Assertion is called a "goal" but message hints do use that terminology while providing hints. Second, Assertion nodes are also formatted with a question mark like the conclusion. Formatting leverages both the visual expectation principle above, but also Cialdini's consistency notion that people prefer to be consistent. Once they get used to following tutor instructions and justifying nodes that have question marks, Assertions can rely on people's natural consistency that influences them to continue to make similar consistent choices. Previous studies on help-seeking and hint usage suggest that students have many different reasons for help avoidance, including their attitudes towards hints and their preference for autonomy [65]. Persuasive design elements may circumvent these preferences by simply influencing students to do what is suggested.

4 Method

Based on our foundational design principles and literature review, we propose the following three hypotheses: (H1) Assertions will increase the unsolicited hint usage for all students irrespective of their prior knowledge. (H2) Assertions will lead students with low prior knowledge to form shorter proofs faster in the posttest. (H3) Assertions will foster productive persistence among students with low prior knowledge.

4.1 Participants

The study was conducted with 122 participants at North Carolina State University, the top engineering university in the state, where Deep Thought was given as a homework assignment to a class of 312 undergraduate students in the College of Engineering majoring in Computer Science, Computer Engineering, or Electrical Engineering in a Fall 2018 discrete mathematics course. We do not have specific demographics of study participants, but the Fall 2018 College of Engineering demographics include 25.3% women, 67.2% white, 8.3% Asian, 6.5% Non-resident Alien, 0.3% American Indian/Native American, 3.3% Black/African American, 4.8% Hispanic/Latinx, 4.83% from two or more under-represented minorities, and 5.9% with unknown race/ethnicity [1].

4.2 Conditions

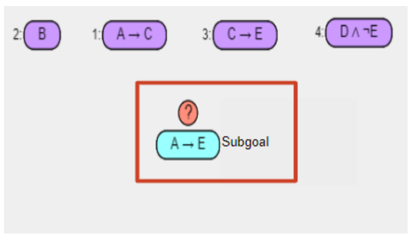
We used stratified sampling to split students based on their pretest performance, and then randomly assigned them to the conditions with *Assertions* as the treatment, and *Messages* as the control. The condition assignment resulted in $N = 73$ in Assertions, and $N = 49$ in Messages. The total number of participants who completed the study was 105 (61 in Assertions, 44 in Messages) but after removing logs with system errors, the dataset had 100 students with 57 in Assertions, and 43 in Messages. We performed a χ^2 test of independence to examine the impact of completion rate and system errors on the groups and found no significant differences among the two groups: $\chi^2(2, N = 122) = 1.88, p = 0.91$. This implies that the group sizes were not significantly impacted by the tutor completion rate or logging errors.

4.3 Procedure

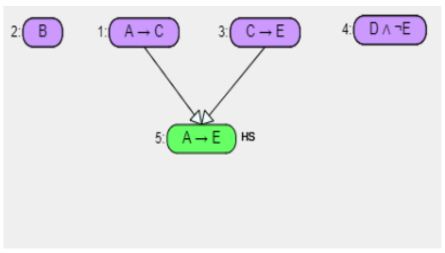
The student procedure is as follows: The tutor provides students with practice solving logic problems, divided into four sections: introduction, pretest, training, and posttest. The introduction presents two worked examples to familiarize students with the tutor interface. Next, students solve two problems in a *pretest*, which is used to determine students' incoming competence. Students are assigned a condition based on their pretest performance. The pretest problems are designed to be easy and short, using a few straightforward rules, and this is reflected in their short optimal solution lengths ($Mean = 3.5$, $SD = 0.71$). Next, the tutor guides students through the *training* section with five training levels with gradually increasing difficulty, and this is reflected in the average length of optimal solutions during training, with a mean optimal solution length of $Mean = 4.99$ steps, ($SD = 1.32$). For each level, each student must solve *four* training problems. Students may skip a maximum of three problems per level, with each skip taking students to easier problems. Students may also restart problems using the "Restart" button below the workspace. In both conditions, students in the training levels may request on-demand hints and always receive immediate feedback on rule application errors (see section 3). Students in the Messages (control) condition received unsolicited message hints upon one minute of inactivity. Students in the Assertions (treatment) condition were given Assertions after about 40% of their steps.The algorithm we use to provide Assertions uses two steps. In the first step, we decide at

[1] More details can be found on Fall 2018 student demographics at NCSU at https://www.engr.ncsu.edu/ir/fast-facts/fall-2018-fast-facts/ The CSC 226 course is typically composed of about 60% sophomores, 30% juniors, 9% seniors, and 1% freshmen

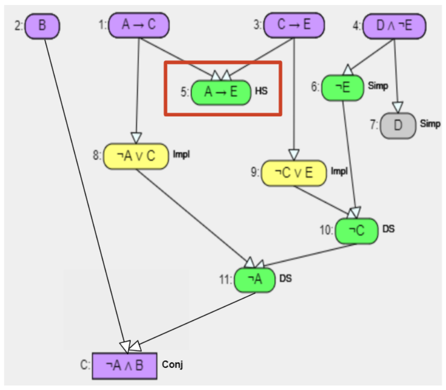
Fig. 4: Example scenarios of Assertion hint $A \to E$ usage usage



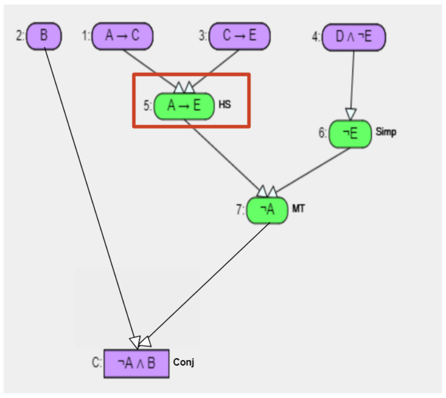
(a) The Assertion $A \to E$ node appears in the student workspace; if it is never justified, it remains as-is



(b) (The student has *justified* the hint by selecting nodes 1 and 3 and rule HS



(c) A student solution where hint $A \to E$ was justified but *not needed*



(d) Another student solution where the hint was *both justified and needed*

random whether the step should get a hint with 50% probability. In the second step, we check for the constraints that assertion should not be given in more than two consecutive steps. This resulted in an actual assertion provision rate of 40%. Note that both Messages and Assertions remain on the screen until a student justifies them[2]. Further, only one unsolicited hint, regardless of interface, may be present at a time, and the hint content is not updated based on new student work. Finally, students take a more difficult *posttest* with four problems, with longer optimal solution lengths compared to the other sections ($Mean = 7.25$, $SD = 1.89$). Students were given a week to complete the procedure. The average time students worked on the tutor was 3.4 hours, with a median of 2.6 hours and standard deviation of 2.8 hours.

[2] The tutor allows students to delete assertions but only two Assertions were deleted in the entire dataset, suggesting that students did not realize this was possible

4.4 Hint Usage

The motivation for using Assertions is to reduce students' reluctance towards using unsolicited help. We believe that increased attention will be paid to Assertions over Messages, and that this increased attention will lead to increased hint usage as prior studies have shown [22]. Figure 4 illustrates two types of hint usage we can observe through tutor logs: hints justified and hints needed. A statement or hint is *justified* when a student applies rules to existing statements to derive it. Figure 4a shows an Assertion suggesting $A \rightarrow E$. When a student selects nodes 1 and 3 and the rule *HS* to derive the hint $A \rightarrow E$, the Assertion hint $A \rightarrow E$ is said to be justified, and it becomes a numbered node 5 as in Figure 4b. The student may continue to solve the problem as in Figure 4c, without ever having used node 5 to justify any other node. As in this case, whenever an Assertion was justified but could be deleted without making the solution incomplete, we say the Assertion was *justified but not needed*. Another student may solve the problem as in Figure 4d where the same hint statement $A \rightarrow E$ is both justified and needed. If we remove node 5 from the solution, it becomes incomplete since nodes 7:$\neg A$ and C:$\neg a \wedge B$ could not be derived without it.

We assume that if students justify a hint, they have paid attention to it. The *Hint Justification Rate* (HJR) is defined as hints justified divided by the total given across the training problems. As in other multi-step open-ended problem domains, students may derive several statements that are not needed to solve a problem, making the solution longer than necessary. For a hint to be called *needed*, students must first justify it, but must also figure out how they can use it to derive the conclusion. *Hint Needed Rate (HNR)* is defined as hints needed divided by the total number of hints given across the training problems. We use unsolicited HJR to evaluate student attention towards unsolicited help, and unsolicited HNR to measure the influence of unsolicited hints on student problem solving.

4.5 Performance Measures

Our test performance measures include: solution length optimality, problem-solving time, and rule application accuracy. In open-ended domains, *solution length*, i.e., the number of derived statements in a *complete* solution, is a valuable performance metric as there is a vast diversity of possible student solution paths. Our aim with increasing unsolicited hint usage is to guide students to learn efficient problem-solving strategies from incorporating the partially worked example Assertion steps as necessary statements in their solutions. Since the posttest consists of four problems, we evaluate students based on their average solution length in the posttest, and shorter lengths are better[3].

[3] Note that solution length can only be calculated for complete solutions, and our data consists only of students who successfully completed the study by completing the mandatory pre- and post-test problems. $N = 5$ (10%) in Messages, and $N = 12$ (16%) in Assertions did

Problem solving time is also an important performance metric in open-ended domains. Similar to other studies [41, 81], we also assess students on the *total time* they spend solving problems. In order to account for outliers, while calculating problem solving time, we cap each click-based interaction time to five minutes, i.e., if a student took more than five minutes to perform an interaction, we cap it to five [4]. A shorter problem solving time suggests better performance. We hypothesized that an increased usage of unsolicited hints, will help students learn to solve problems more quickly and with shorter solution lengths, and that these effects will be more pronounced for students with low prior knowledge.

Finally, *Accuracy* is defined as the number of correct rule applications divided by the total number of applications. A higher accuracy value suggests better knowledge of how to apply domain rules. Since the tutor is designed to provide immediate feedback on incorrect rule applications without penalties, even within the pre- and post-tests (see section 3), we do not hypothesize differences in the accuracy between the two conditions. We report accuracy for both conditions, however, for completeness.

4.6 Prior Proficiency

We hypothesize that an increase in the unsolicited hint usage significantly impacts the performance of students with low prior knowledge. Our prior work [72] suggests that students with different incoming competencies can experience a treatment differently. To account for such aptitude-treatment interaction effects, we quantify prior knowledge by splitting the students into Low and High *Prior Proficiency* groups using a normalized pretest performance score that combines the number of problem-solving steps, the average time spent on each step, and accuracy. The three performance measures are normalized separately and equally weighted in a combined score that is again normalized. Students with pretest performance > 0.5 are classified as the High group, i.e., students with high prior proficiency, and students with lower pretest performance are classified as the Low group, i.e., students with low prior proficiency. We found an insignificant difference between the High and Low Prior Proficiency group sizes between the two Conditions ($\chi^2(1, N = 122) = 0.24, p = 0.62$). This allows us to compare the students in the two Conditions within each Prior Proficiency group.

4.7 Effort and Persistence

Hypothesis H3 states that Assertions will encourage students with low prior proficiency to engage in productive persistence. Our definition of *effort* is

not finish the tutor. A chi-square test shows no significant difference in the completion and non-completion group sizes between the two conditions ($\chi^2(1, N = 122) = 0.95, p = 0.33$)

[4] The 99th percentile of interaction action time in Fall 2018 was 99.03s; 811 out of 260,750 interaction logs for 100 students in the study, had an action time greater than 5min

highly motivated by prior research. More specifically, Venture et al. defined a metric for students' efforts as the amount of time spent on unsolved problems and they found that there was a significant correlation between the effort measured during the training and a self-report measure of persistence [86]. Later, in another study, they used this effort metric to measure persistence in an educational game that teaches Qualitative Physics [85]. In our tutor, students can skip up to three problems per training level and thus we also measure the time students spent in these unsolved skipped problems as a measure of effort. Moreover, Dumdumaya et al. defined their effort metric as the number of reattempts made on a problem after a failed attempt predicted task persistence [30]. In our tutor, this corresponds to the number of restarts on problems that students eventually solve. In the following, we separately track effort through two research-based measures: (1) time spent on unsolved (skipped) training problems, and (2) the number of restarts on solved training problems. For the purpose of this analysis, we define *productive persistence* as persistent (high) efforts that result in higher posttest performance.

5 Results

After cleaning the data as described in Section 4.2, an average of 2,483 interactions were logged and analyzed per student in our final sample of 100 students (with 57 in the Assertions condition, and 43 in Messages). We partitioned the students based on Prior Proficiency into Low (n = 41) and High (n = 59) groups. We then partitioned by Condition and Prior Proficiency resulting in 4 groups: Assertions-Low (n = 25), Assertions-High (n = 32), Messages-Low (n = 16), and Messages-High (n = 27).

Before investigating any of our hypotheses, we first compared the number of on-demand hints between the two conditions to ensure that any differences between groups could not be explained by differences in on-demand hint requests. Similar to other tutors [47, 65], students in this study rarely request on-demand help irrespective of Condition or Prior Proficiency. We found no significant differences in the number of on-demand hint requests between conditions or by prior proficiency, with all conditions requesting, on average, less than one on-demand hint per problem. Students in the Assertions condition requested few on-demand hints per problem, with Mean = 0.79 , SD = 3.92 (Assertions-Low group: Mean = 0.67, SD = 2.82, and Assertions-High group: Mean = 0.89, SD = 3.07). Students in the Messages condition similarly requested few on-demand hints per problem for the Messages group, with Mean = 0.55 , SD = 2.72 (Messages-Low: Mean = 0.46, SD = 2.36, and Messages-High: Mean = 0.59, SD = 2.43). The on-demand hint data was not normally distributed as tested by the Shapiro-Wilk's test (Assertion: W = 0.744, $p < 0.001$, Messages: W = 0.752, $p < 0.001$). So, a two-factor Aligned Ranks Transformation ANOVA [90] with the two factors as the Condition {Assertion, Messages} and Prior Proficiency {Low, High} on the number of on-demand hints shows no significant main effects (Condition: $F(1{,}100) = 0.132$, p =

Table 1: Comparison of unsolicited hint metrics between the two conditions, where a two-way Aligned Ranks Transformation ANOVA for each metric shows only a main effect of Condition ($p < 0.001$*)

Unsolicited Hint Metric	Assertions	Messages
Hints Given in Training	48.82 (9.85)*	32.74 (10.64)
Hint Justification Rate (HJR)	0.93 (0.07)*	0.63 (0.18)
Hint Needed Rate (HNR)	0.82 (0.09)*	0.62 (0.17)

0.718, Prior Proficiency: $F(1,100) = 1.075$, $p = 0.302$) or interaction ($F(1,100) = 0.006$, $p = 0.940$). Based on this analysis, the remaining analyses focus only on usage for unsolicited Assertion and Message hints.

5.1 **H1**: Assertions increase the unsolicited hint usage for all students irrespective of their prior knowledge

Table 1 shows the unsolicited hint metrics: #Given (number of unsolicited hints given during training), HJR (Hint Justification Rate - proxy for attention paid), and HNR (Hint Needed Rate - hints' influence on problem-solving) for the two Conditions[5]. Since we have hint data that is not normally distributed[6], we performed a two-way Aligned Ranks Transformation ANOVA [90] on each of the unsolicited hint metrics with the two factors as Condition {Assertions, Messages}, and Prior Proficiency {Low, High}.

We applied a two-way Aligned Ranks Transformation ANOVA on the unsolicited hint metrics of #Given, HJR and HNR as described above. For the #Given metric, we found a significant main effect of *Condition* ($F(1,100) = 40.26$, $p < 0.001$). While we expected this result because of the system design, we will discuss this further in the next section. For both hint usage metrics HJR and HNR, we observed a significant main effect of *Condition* (HJR: $F(1,100) = 191.10$, $p < 0.001$, and HNR: $F(1,100) = 62.30$, $p < 0.001$). The main effect of Prior Proficiency and the interaction effect were not significant for either HJR or HNR. As we hypothesized, the Assertions groups used a significantly higher proportion of unsolicited hints, both by justifying (HJR) and needing (HNR) more unsolicited hints than the Messages group, as shown in Table 1.

We did not observe a significant interaction between Condition and Prior Proficiency for any of the unsolicited hint metrics: #Given: $F(1,100) = 0.008$, $p = 0.929$, HJR: $F(1,100) = 0.221$, $p = 0.639$, and HNR: $F(1,100) = 0.009$, $p = 0.924$. The distribution parameters for each of the unsolicited hint metrics per Prior Proficiency group are provided in Appendix A. There was only a

[5] HJR and HNR are the proportion of hints justified and needed respectively

[6] Shapiro-Wilk's test on *Unsolicited Hints Given* for the Assertions group: $W = 0.904$, $p < 0.001$, and the Messages group: $W = 0.942$, $p = 0.030$; Shapiro-Wilk's test on *Unsolicited HJR* for the Assertions group: $W = 0.887$, $p < 0.001$, the Messages group: $W = 0.959$, $p < 0.001$; and Shapiro-Wilk's test on *Unsolicited HNR* for the Assertions group: $W = 0.904$, $p < 0.001$, and the Messages group: $W = 0.945$, $p < 0.001$

Table 2: Comparison of ***Posttest Performance metrics*** between the two conditions within each ***Prior Proficiency group*** - Average Solution Length ($p = 0.033$) and Total time ($p = 0.008$) are significantly different between the Assertions-Low and the Messages-Low groups

Prior	Avg. Sol. Length (#nodes)		Total Time (min)	
Profi-	Assertions	Messages	Assertions	Messages
ciency	Mean (SD)	Mean (SD)	Mean (SD)	Mean (SD)
Low	13.33 (1.10)*	15.09 (1.95)	36.03 (12.81)*	52.94 (18.19)
High	14.36 (1.57)	14.49 (1.38)	41.73 (17.12)	43.27 (19.09)
All	13.92 (1.44)	14.46 (1.69)	38.13 (14.95)	46.99 (17.65)

main effect of the Condition as shown above. This shows that the Assertions had a significant impact on unsolicited hint usage for all students, regardless of incoming proficiency, confirming hypothesis H1.

5.2 **H2**: Assertions will lead students with low prior knowledge to form shorter proofs faster in the posttest

Since all performance data were normal, we performed t-tests to compare conditions. A t-test on the average pretest solution length between the Assertions (Mean = 7.54 nodes, SD = 1.87 nodes) and the Messages (Mean = 7.64, SD = 2.21) conditions, showing no significant difference ($t(99) = 0.791$, $p = 0.215$). We also observed insignificant differences in the pretest problem-solving time ($t(99) = 0.683$, $p = 0.248$) using a t-test between the Assertions (Mean = 27.16 min, SD = 8.29 min) and the Messages (Mean = 25.89 min, SD = 10.23 min) conditions. While the H2 hypothesis does not predict differences in accuracy between conditions, students were assigned a condition based on their pretest performance, which includes rule application accuracy, so we compare it here. A t-test on the pretest rule accuracy between the Assertions (Mean = 0.52, SD = 0.16) and Messages (Mean = 0.52, SD = 0.14) conditions shows no significant difference ($t(99) = 0.111$, $p = 0.455$).

As mentioned earlier, hypothesis H2 is based on the reasoning that Assertions may guide students towards optimal strategies, which can lead students with low prior proficiency to form shorter solutions in less time. We examined the correlation between the dependent variables (average solution length and total time) to assess their overlap, both for the entire population and for the low prior proficiency group. We did not observe a significant correlation between the average posttest solution length and posttest time for the entire population: $Corr = 0.050$, $p = 0.615$ or for the Low Prior Proficiency group: $Corr = 0.015$, $p = 0.916$.

Table 2 shows the posttest performance of the two Conditions {Assertions, Messages} disaggregated for the *Low*, and *High* Prior Proficiency groups in the first two rows, and for *All* students as a summary in the bottom row. To

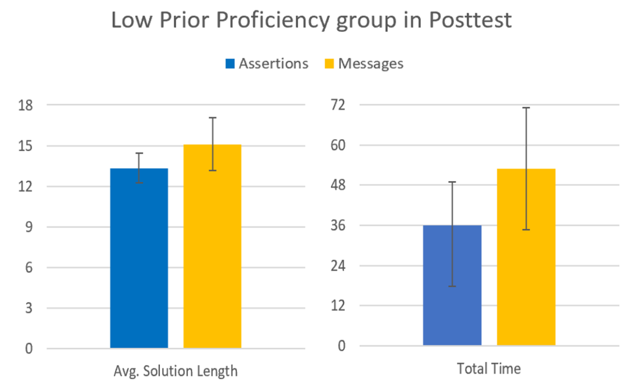


Fig. 5: Tukey's HSD shows that the Assertions-Low performed significantly better in posttest than the Message-Low group in average solution length ($p = 0.033$) and total time ($p = 0.008$)

investigate our H2 hypothesis, we performed a two-way ANCOVA on average solution length and total time, with the Condition {Assertions, Messages} and Prior Proficiency {Low, High} as the two factors, and the respective pretest performance metric as the covariate. For average solution length, we observed a significant interaction between the Condition and Prior Proficiency ($F(1,100) = 4.983$, $p = 0.027$). Neither main effect for Condition or Prior Proficiency were significant. We then performed the pairwise Tukey's Honest Significant (HSD) test for multiple comparisons and found a significant difference ($p = 0.033$) between the *Assertions-Low* and *Messages-Low* groups, showing that the Assertions-Low group formed significantly shorter proofs on the posttest than the Messages-Low group.

A two-factor ANCOVA on posttest total time as described above shows a significant interaction between the Condition and Prior Proficiency ($F(1,100) = 6.236$, $p = 0.014$), and a significant main effect of the Condition ($F(1,100) = 6.913$ $p = 0.010$). The main effect of Prior Proficiency was not significant. A pairwise Tukey's Honest Significant (HSD) test for multiple comparisons on the total posttest time shows a significant difference ($p = 0.008$) between *Assertions-Low* and *Messages-Low* groups. The Assertions-Low group spent significantly less time on the posttest than the Messages-Low group. Figure 5 summarizes the differences between the Assertions-Low and Messages-Low groups in their posttest performance. Together with the results above, the Assertions-Low group had significantly better posttest solution length and time than the Messages-Low group, confirming our H2 aptitude-treatment interaction hypothesis for posttest performance. While we did not hypothesize improvements in posttest accuracy, we provide these results in Appendix B for completeness.

Next, we investigated the correlation between average posttest solution length with the unsolicited hint metrics. First, the top row of Table 3 shows

Table 3: Correlation between ***average posttest solution length*** and unsolicited hint metrics for the entire population, and split by low and high prior proficiency groups

Posttest Solution Length with	Entire Population N = 100		Low Prior Proficiency N = 41		High Prior Proficiency N = 59	
	Corr	*p*	Corr	*p*	Corr	*p*
#Given (Frequency)	-0.13	0.18	-0.10	0.26	-0.03	0.83
HJR (Attention)	-0.25	0.01*	-0.37	< 0.001*	-0.07	0.65
HNR (Influence)	-0.02	0.87	-0.48	<0.001*	0.12	0.42

Table 4: Correlation between ***total posttest time*** and unsolicited hint metrics for the entire population, and split by low and high prior proficiency groups

Posttest Total Time with	Entire Population N = 100		Low Prior Proficiency N = 41		High Prior Proficiency N = 59	
	Corr	*p*	Corr	*p*	Corr	*p*
#Given (Frequency)	-0.02	0.88	-0.06	0.68	-0.17	0.23
HJR (Attention)	-0.28	<0.01*	-0.40	< 0.001*	-0.25	0.07
HNR (Influence)	-0.27	<0.01*	-0.36	<0.001*	-0.20	0.16

that the number of unsolicited hints given does not correlate to posttest solution length, suggesting that differences in posttest solution lengths between conditions cannot be attributed to the frequency of unsolicited hints. However, both HJR (second row) and HNR (third row) are significantly and negatively correlated to posttest solution length for students with Low Prior Proficiency (HJR: $p < 0.001$, HNR: $p < 0.001$), with a stronger correlation to posttest solution length for HNR than HJR[7]. This suggests that students with low prior knowledge learn more from the hints needed, rather than the ones they only justified (see Figure 4 differentiating hints justified and needed). A justified, but not needed, hint suggests that a student could determine how to derive the unsolicited hint content, but not *how to use it*. It is reasonable that lower prior proficiency students who were able to include the unsolicited hints as necessary components of their proof solutions were more likely to learn more optimal, shorter problem-solving strategies. We also observed an insignificant but positive correlation between average posttest solution length and HNR for the High prior knowledge group. While small and not significant, this inverted effect may indicate another aspect of aptitude treatment interaction, where high prior proficiency students may potentially learn less if they take too much advantage of unsolicited hints. This result suggests that it may be

[7] We did not test for the significance in the difference between the two correlation coefficients because the samples are not independent. Hints Needed are a subset of Hints Justified

preferable to build a more adaptive method to determine when to present unsolicited hints to students with high prior proficiency.

Table 4 shows the correlation between posttest time and the unsolicited hint metrics. First, Table 4 shows that the number of unsolicited hints given (top row) does not correlate to posttest time, suggesting that differences in posttest time between conditions cannot be attributed to the frequency of unsolicited hints. While HJR (second row) and HNR (third row) are significantly correlated to the posttest time for the entire population, the Pearson's Correlation Coefficient is less than 0.3, suggesting small coverage. However, students with Low Prior Proficiency have a significant correlation (that is also greater than 0.3) between posttest time and unsolicited hint usage metrics HJR and HNR.

Table 1 shows that, over the entire population, significantly more ($p < 0.001$) unsolicited hints were given on average (#Given) in the Assertions condition (39.78 % of total steps on average) than in Messages (26.93 % of total steps on average). We observed a significant main effect of *Condition* on the number of unsolicited hints given. Neither the main effect of Prior Proficiency nor the interaction effect were significant. It would be reasonable to expect that the frequency of hints might impact posttest performance. However, our correlation analysis shows that the significantly higher number of unsolicited hints given in the Assertions condition did not correlate with posttest performance for either solution length or time. Instead, the significant negative correlations between posttest length and time, and Hints Needed Rate for all students with low prior knowledge suggests that students in the Low group learned from using the unsolicited hints to achieve problem conclusions. These needed hints provided insight into efficient problem solving, by showing students optimal problem-solving steps. As shown in Table 1 above, students in the Assertions condition had higher HNR than students in the Messages condition. Therefore, our results confirm hypothesis H2 that there would be an aptitude-treatment interaction effect where Assertions helped students with low prior proficiency learn to construct more optimal (shorter) solutions more quickly on the posttest.

5.3 **H3**: Assertions foster productive persistence among students with low prior knowledge

We hypothesized that increased usage of unsolicited hints in the form of Assertions will lead students with low prior proficiency to exert persistent effort in training, and this persistence will be productive (i.e., improved posttest performance). We clustered students on five features including: two productivity measures (posttest solution length and time, where lower is better), two effort measures including time spent on unsolved (skipped) problems and the number of restarts, and unsolicited hint usage as measured by HJR. We used Hint Justification Rate (HJR) instead of Hint Needed Rate (HNR) since the hints needed cannot be determined for unsolved problems. The clustering analysis

Table 5: Selecting the number of clusters based on three cluster quality indices

#Clusters	Silhouette	Davies-Bouldin	Calinski-Harabasz
2	0.49	0.63	122.36
3	**0.55**	**0.51**	234.91
4	0.50	0.57	271.28
5	0.46	0.61	**285.74**

Table 6: Centroids (Mean) of the three clusters using Hierarchical Clustering with the Ward's method

Cluster No.	Cluster Label	Posttest		Training		
		Total Time (min)	Avg. Sol. Length (#nodes)	Unsolved Problem Time (min)	Re-starts	HJR
#1	Productive - High Effort - High HJR	30.11	13.08	19.62	4.58	0.88
#2	Productive - Low Effort - High HJR	38.51	13.98	3.79	1.63	0.83
#3	Unproductive - Low Effort - Low HJR	59.19	14.75	0.81	1.06	0.50

provides a deeper understanding of student behavior patterns involving productivity, effort, and proactive hint usage. An ANOVA on the effort metrics would not have helped us understand how student effort varies in tandem with both productivity and hint usage. Therefore, the cluster analysis is more geared towards answering H3 than an ANOVA.

We performed cluster analysis using Hierarchical clustering with Ward's method on standardized features. We selected the number of clusters using majority vote across three indices: Silhouette and Calinski-Harabasz, which both maximize inter-cluster similarity and minimize intra-cluster similarity (overall higher values are better), and the Davies-Bouldin Index, which prefers minimal intra-cluster similarity (overall lower values are better). Table 5 shows that using three clusters yields the best quality clusters.

Table 6 shows the centroids of the three clusters. We used the *class average (CA)*, i.e., average over the entire population to assess the clusters on each measure. The following order was observed for each feature used in the clustering analysis: (Note that lower posttest time and solution lengths are better)

- **Posttest Time (min)**: #1 $<$ #2 $<$ CA (42.81) $<$ #3
- **Posttest Sol. Length**: #1 $<$ #2 $<$ CA (14.01) $<$ #3
- **Unsolved Problem Time (min)**: #1 $>$ CA (5.16) $>$ #2 $>$ #3
- **Restarts**: #1 $>$ CA (2.44) $>$#2 $>$ #3
- **Hint Justification Rate (HJR)**: #1 $>$ #2 $>$ CA (0.80) $>$ #3

As shown in Table 6, Cluster # 1 shows the highest productivity in terms of posttest performance, with the highest effort averages, and HJR (with all

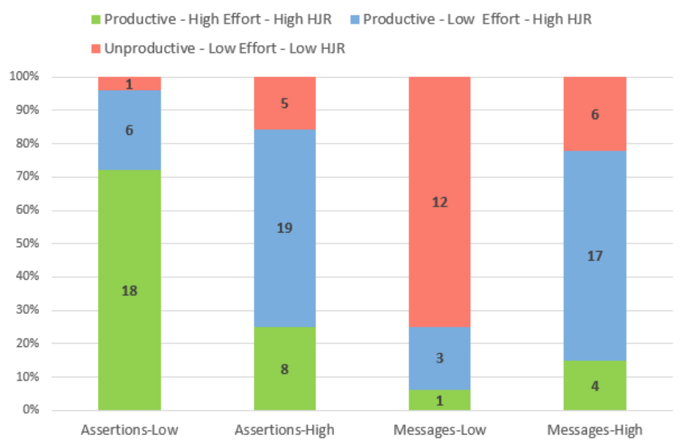


Fig. 6: Profile of the three Clusters based on the Condition and Prior Proficiency

five features better than the class average). In the following, we refer to this cluster as ***Productive - High Effort- High HJR***. Cluster #2's posttest performance and HJR measures are better than the class average but their average effort on both time on unsolved problems and number of restarts are lower than the class average. This cluster was productive without needing to exert a high effort on unsolved problems or restarting problems during training. So, we call this cluster ***Productive - Low Effort - High HJR***. Interestingly, a lot of the High Prior Proficiency students ended up in low effort but did no better than Assertions-Low group on the posttest. Lastly, cluster #3 shows the worst posttest time and solution length (both higher than the class average), the lowest effort on time on unsolved problems and number of restarts (both lower than the class averages), and the lowest HJR (lower than the class average), so we label this cluster ***Unproductive - Low Effort- Low HJR***.

We then profiled each cluster based on the pairs of the Condition and Prior Proficiency as shown in Figure 6. Interestingly, the majority of the Assertions-Low group students are in the *Productive - High Effort- High HJR* cluster, and the majority of the Messages-Low group students are in the *Unproductive - Low Effort- Low HJR* cluster. Most of the students in the Assertions-High and the Messages-High groups are in the *Productive - Low Effort- High HJR* cluster. Since we are interested in the Low Prior Proficiency group, we performed a chi-square test to compare the distribution of the Assertions-Low and Messages-Low students in the three clusters and found a significant difference ($\chi^2(1, N = 41) = 24.73, p < 0.001$). The majority of the Assertions-Low students show persistent effort as they are in the *Productive - High Effort- High HJR* cluster with the highest effort and unsolicited hint usage in training with productive posttest results, and this confirms our H3 hypothesis.

6 Discussion

6.1 **H1**: Assertions increase the unsolicited hint usage for all students irrespective of their prior knowledge

The hints in our tutor suggest the most optimal next-step statement to derive for any given student problem-solving state. Similar to other tutors [47, 65], in this study, we found that students rarely request on-demand help irrespective of condition. However, our results suggest that the difference in unsolicited hint usage between Messages and Assertions can be attributed to presentation alone. We found that Assertions, specifically designed using the principles of contiguity, attention, expectation, and persuasion, significantly increased both the attention students pay to unsolicited hints (HJR), and their influence on students' solutions (HNR) regardless of the students' prior knowledge. Conati and Manske suggested in [23] that students pay more *attention* to simpler hints. Assertions provide high immediacy (making the hint content immediately usable, [7]) since they leverage both spatial *contiguity* [50] by placing information right where it is needed and visual *expectation* [76] by formatting hints to make them more intuitive to follow. Studies have also found students' attitude towards unsolicited hints to be an important factor in help avoidance [22, 65]. Persuasive factors like increasing perceived authority [20] through formatting and language can make justifying Assertions seem to be required. Our results show that an unsolicited hint interface that combines persuasion, making hint usage seem required, with high immediacy, making it easy to see and do, can help overcome barriers to hint usage.

6.2 **H2**: Assertions will lead to students with low prior knowledge to form shorter proofs faster in the posttest

Several studies have found ATI effects surrounding hint usage where students with low prior knowledge or proficiency benefit more from interventions [5, 41, 56]. In particular, Kardan and Conati [41] found that attention to hints affected student performance in a tutor for teaching constraint satisfaction problems, and students with low prior knowledge experienced more pronounced effects from an adaptive hint design intervention. While their intervention dealt with both an unsolicited hint interface (highlights to direct attention) and scaffolding (incremental textual hints), our study focuses only on the interface of unsolicited hints. Our ANCOVA results showed a significant aptitude-treatment interaction between Prior Proficiency {High, Low} and Condition {Assertions, Messages}. Using Tukey's HSD tests, we inferred that the Assertions-Low group outperformed the Messages-Low group in posttest solution length and time. We also observed a significant correlation of the posttest solution length and time with hint needed rate (HNR) for the Low Prior Proficiency group, suggesting that using more unsolicited hints as necessary components of their proofs helped this group learn better strategies.

However, no such relations were observed for the High Prior Proficiency group. This suggests that adapting hint timing of Assertions based on proficiency may improve student performance as in other ITSs [82, 87, 91].

6.3 **H3**: Assertions foster productive persistence among students with low prior knowledge

Persistent effort is said to be productive when it is accompanied by an improvement in posttest performance [13]. Assertions are designed to encourage students to follow unsolicited hints that direct students toward optimal problem-solving strategies. Results from our empirical study support the notion that Assertions promote productive persistence. Our cluster analysis showed that the majority of the Assertions-Low group exerted more effort (high persistence) during training, justified a higher proportion of unsolicited hints, and performed better on the posttest than the class average. We also saw a higher proportion of the Messages-Low students in the cluster that exerted less effort (low persistence) in the training, justified a lower proportion of unsolicited hints, and performed worse on the posttest than the class average. Interestingly, while most of the Assertions-Low group spent more time on unsolved problems in training, they took a significantly shorter time on the posttest while creating shorter posttest solutions, suggesting that the Assertions promoted productive persistence (i.e. time well spent) among students with low prior proficiency.

6.4 Assertions - a new genre of hints

Overall, this study showcases the importance of effective delivery for unsolicited hints, and a new genre of hints that we call Assertions. We believe that providing unsolicited hints as partially worked steps reduced the cognitive load required for learning from them. Further, increasing spatial contiguity improved students' attention towards hints, and the isomorphic format may have made it easier for them to understand and use them in their solutions. We observed that Assertions led students with low prior knowledge to exert more productive persistence in training that resulted in better posttest performance, where they formed significantly shorter, more optimal, solutions in significantly less time than their peers in the control condition who only received Message hints. Assertions provide students with additional problem-solving resources that can enable them to learn through the process of self-explaining (justifying) expert steps. We believe that Assertions may be particularly helpful in multi-step domains, where providing students with partially-worked steps, right next to where they are needed, periodically, and in the same format as other problem-solving steps, could lead students to do more self-explanation (through justifying or completing the partially-worked steps) and by circumventing help avoidance.

A limitation of this work is the difference in the timing of Assertions and Messages, which could have impacted the results. While the hint frequency correlation analysis showed that students' posttest performance was not impacted by the *number* of unsolicited hints given, we recognize that the hint timing may have had an impact on students. This limitation arises from the fact that we are modifying a real adaptive system to achieve practical improvements. These two types of hints were designed for different purposes. Messages were intended to help someone who was struggling but forgot about the help feature. Assertions were intended to be proactive for students who wouldn't ask for help no matter what.Assertions were designed to address the problem that we observed, that Messages were not helping enough people improve their performance or learning.

7 Conclusion and Future Work

In this study, we investigated the impact of Assertions, a new genre of unsolicited hints, on the hint usage and posttest performance within a data-driven tutoring system. This work is novel in that it leveraged interface alone to address the help avoidance problem. However, this work did not seek to regulate students' help-seeking, rather we sought to make unsolicited hints more effective through changes in their delivery. The Assertions hint interface made the intelligent tutor more effective, significantly improving unsolicited hint usage for all students. We further demonstrated aptitude-treatment interaction effects where students with low prior proficiency receiving Assertions performed better in the posttest, in terms of both time and solution length. Our cluster analysis shows that the students with low prior knowledge who received Assertions demonstrate more productive persistence in that they exerted more persistent effort even when failing during training, and used a higher proportion of unsolicited hints, but performed better on the posttest than their low peers who received Messages.

There are three main limitations to this study. Assertions were provided significantly more frequently than Messages. Assertions did not seem to have a negative impact on learning, but rather leveled the playing field for students with low prior proficiency. However, our analyses demonstrated that it was not hint frequency but the Assertions interface alone that improved hint usage. The second limitation was that Assertions appeared randomly, and were not adapted to individual students. Our results confirm our hypothesis that the Assertions have a differential impact for students with different incoming proficiency, suggesting that there may be benefits to using individual factors to determine when to provide Assertions. A third limitation arises from splitting students into two prior proficiency groups. While some studies investigate finer-grained partitions, e.g. low, medium/average, and high groups [41], we refrained from doing so to maintain sufficiently high sample sizes within each group.

This study was a necessary first step to identify a hint interface that could solve the help avoidance problem. Future work could study the generalizability of this transformative new genre of unsolicited hints that use the design principles of contiguity, attention, and expectation to increase hint immediacy and persuasion to reduce help avoidance in other tutors. Within our tutor, we plan to apply reinforcement learning and other machine learning techniques to derive an adaptive policy to decide when and if Assertions should be provided to individual students. Since Assertions promote productive persistence among students with low prior knowledge, we also plan to develop a model that provides Assertions when the tutor detects or predicts unproductive behaviors [44, 45].

8 Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 1726550, "Integrated Data-driven Technologies for Individualized Instruction in STEM Learning Environments.", led by Min Chi and Tiffany Barnes. We would like to thank Nicholas Lytle (nalytle@ncsu.edu) for suggesting edits in the introduction section to enhance its clarity.

References

1. Aleven, V., Koedinger, K.R.: Limitations of student control: Do students know when they need help? In: International Conference on Intelligent Tutoring Systems, pp. 292–303. Springer (2000)
2. Aleven, V., Mclaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. International Journal of Artificial Intelligence in Education **16**(2), 101–128 (2006)
3. Aleven, V., Ogan, A., Popescu, O., Torrey, C., Koedinger, K.: Evaluating the effectiveness of a tutorial dialogue system for self-explanation. In: International conference on intelligent tutoring systems, pp. 443–454. Springer (2004)
4. Almeda, M.V.Q., Baker, R.S., Corbett, A.: Help avoidance: When students should seek help, and the consequences of failing to do so. In: Meeting of the Cognitive Science Society, vol. 2428, p. 2433 (2017)
5. Arroyo, I., Beck, J.E., Beal, C.R., Wing, R., Woolf, B.P.: Analyzing students' response to help provision in an elementary mathematics intelligent tutoring system. In: Papers of the AIED-2001 workshop on help provision and help seeking in interactive learning environments, pp. 34–46. Citeseer (2001)
6. Ausin, M.S., Azizsoltani, H., Barnes, T., Chi, M.: Leveraging deep reinforcement learning for pedagogical policy induction in an intelligent tutoring system. In: Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019), vol. 168, p. 177. ERIC
7. Bakke, S.: Immediacy in user interfaces: An activity theoretical approach. In: International Conference on Human-Computer Interaction, pp. 14–22. Springer (2014)
8. Barnes, T., Stamper, J.: Automatic hint generation for logic proof tutoring using historical data. Journal of Educational Technology & Society **13**(1), 3 (2010)
9. Barnes, T., Stamper, J., Croy, M.: Using markov decision processes for automatic hint generation. Handbook of Educational Data Mining **467** (2011)
10. Barnes, T., Stamper, J.C., Lehmann, L., Croy, M.J.: A pilot study on logic proof tutoring using hints generated from historical student data. In: EDM, pp. 197–201 (2008)

11. Bartholomé, T., Stahl, E., Pieschl, S., Bromme, R.: What matters in help-seeking? a study of help effectiveness and learner-related factors. Computers in Human Behavior **22**(1), 113–129 (2006)
12. Beck, J.E., Gong, Y.: Wheel-spinning: Students who fail to master a skill. In: International conference on artificial intelligence in education, pp. 431–440. Springer (2013)
13. Borghans, L., Duckworth, A.L., Heckman, J.J., Ter Weel, B.: The economics and psychology of personality traits. Journal of human Resources **43**(4), 972–1059 (2008)
14. Botelho, A., Varatharaj, A., Patikorn, T., Doherty, D., Adjei, S., Beck, J.: Developing early detectors of student attrition and wheel spinning using deep learning. IEEE Transactions on Learning Technologies (2019)
15. Bunt, A., Conati, C., Muldner, K.: Scaffolding self-explanation to improve learning in exploratory learning environments. In: International Conference on Intelligent Tutoring Systems, pp. 656–667. Springer (2004)
16. Butcher, K.R., Aleven, V.: Integrating visual and verbal knowledge during classroom learning with computer tutors. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 29 (2007)
17. Butcher, K.R., Aleven, V.: Using student interactions to foster rule–diagram mapping during problem solving in an intelligent tutoring system. Journal of Educational Psychology **105**(4), 988 (2013)
18. Chi, M.T., Bassok, M.: Learning from examples via self-explanations. Tech. rep., PITTSBURGH UNIV PA LEARNING RESEARCH AND DEVELOPMENT CENTER (1988)
19. Chi, M.T., De Leeuw, N., Chiu, M.H., LaVancher, C.: Eliciting self-explanations improves understanding. Cognitive science **18**(3), 439–477 (1994)
20. Cialdini, R.B.: Influence: Science and practice, vol. 4. Pearson education Boston, MA (2009)
21. Cody, C., Mostafavi, B.: Investigating the impact of unsolicited next-step and subgoal hints on dropout in a logic proof tutor. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education, pp. 705–705. ACM (2017)
22. Conati, C., Jaques, N., Muir, M.: Understanding attention to adaptive hints in educational games: an eye-tracking study. International Journal of Artificial Intelligence in Education **23**(1-4), 136–161 (2013)
23. Conati, C., Manske, M.: Evaluating adaptive feedback in an educational computer game. In: International workshop on intelligent virtual agents, pp. 146–158. Springer (2009)
24. Conati, C., Vanlehn, K.: Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation (2000)
25. Cronbach, L.J., Snow, R.E.: Aptitudes and instructional methods: A handbook for research on interactions. Irvington (1977)
26. Davies, W., Cormican, K.: An analysis of the use of multimedia technology in computer aided design training: Towards effective design goals. Procedia Technology **9**, 200–208 (2013)
27. Deke, J., Haimson, J.: Valuing student competencies: Which ones predict postsecondary educational attainment and earnings, and for whom? final report. Mathematica Policy Research, Inc. (2006)
28. DiCerbo, K.E.: Game-based assessment of persistence. Journal of Educational Technology & Society **17**(1), 17–28 (2014)
29. Dillard, J.P., Seo, K.: Affect and persuasion. James PRICE dILLARd y Lijian sHEn (coord.), The Sage handbook of persuasion pp. 150–166 (2013)
30. Dumdumaya, C., Rodrigo, M.M.: Predicting task persistence within a learning-by-teaching environment. In: Proceedings of the 26th International Conference on Computers in Education, pp. 1–10 (2018)
31. Duong, H., Zhu, L., Wang, Y., Heffernan, N.T.: A prediction model that uses the sequence of attempts and hints to better predict knowledge:" better to attempt the problem first, rather than ask for a hint". In: EDM, pp. 316–317 (2013)
32. Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, C., Chen, L.: Generating proactive feedback to help students stay on track. In: International Conference on Intelligent Tutoring Systems, pp. 315–317. Springer (2010)
33. Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, C., Chen, L.: Data driven automatic feedback generation in the ilist intelligent tutoring system. Technology, Instruction, Cognition and Learning **10**(1), 5–26 (2015)

34. Healey, C., Enns, J.: Attention and visual memory in visualization and computer graphics. IEEE transactions on visualization and computer graphics **18**(7), 1170–1188 (2012)
35. Heckman, J.J., Stixrud, J., Urzua, S.: The effects of cognitive and noncognitive abilities on labor market outcomes and social behavior. Journal of Labor economics **24**(3), 411–482 (2006)
36. Hegarty, M., Just, M.A.: Constructing mental models of machines from text and diagrams. Journal of memory and language **32**(6), 717–742 (1993)
37. Jin, W., Barnes, T., Stamper, J., Eagle, M.J., Johnson, M.W., Lehmann, L.: Program representation for automatic hint generation for a data-driven novice programming tutor. In: International Conference on Intelligent Tutoring Systems, pp. 304–309. Springer (2012)
38. Jin, W., Lehmann, L., Johnson, M., Eagle, M., Mostafavi, B., Barnes, T., Stamper, J.: Towards automatic hint generation for a data-driven novice programming tutor. In: Workshop on Knowledge Discovery in Educational Data, 17th ACM Conference on Knowledge Discovery and Data Mining. Citeseer (2011)
39. Kai, S., Almeda, M.V., Baker, R.S., Heffernan, C., Heffernan, N.: Decision tree modeling of wheel-spinning and productive persistence in skill builders. JEDM— Journal of Educational Data Mining **10**(1), 36–71 (2018)
40. Kanfer, R., Ackerman, P.L.: Motivation and cognitive abilities: An integrative/aptitude-treatment interaction approach to skill acquisition. Journal of applied psychology **74**(4), 657 (1989)
41. Kardan, S., Conati, C.: Providing adaptive support in an interactive simulation for learning: An experimental evaluation. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 3671–3680. ACM (2015)
42. Koedinger, K.R., Aleven, V., Heffernan, N., McLaren, B., Hockenberry, M.: Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. In: International Conference on Intelligent Tutoring Systems, pp. 162–174. Springer (2004)
43. Liu, Z., Mostafavi, B., Barnes, T.: Combining worked examples and problem solving in a data-driven logic tutor. In: International Conference on Intelligent Tutoring Systems, pp. 347–353. Springer (2016)
44. Maniktala, M., Barnes, T., Chi, M.: Extending the hint factory: Towards modelling productivity for open-ended problem-solving. In: Proceedings of the 13th International Conference on Educational Data Mining (2020)
45. Maniktala, M., Cody, C., Isvik, A., Lytle, N., Chi, M., Barnes, T.: Extending the hint factory for the assistance dilemma: A novel, data-driven helpneed predictor for proactive problem-solving help. In: JEDM— Journal of Educational Data Mining (2020)
46. Marwan, S., Lytle, N., Williams, J.J., Price, T.: The impact of adding textual explanations to next-step hints in a novice programming environment. In: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education, pp. 520–526. ACM (2019)
47. Mathews, M., Mitrovic, A.: How does students' help-seeking behaviour affect learning? In: International Conference on Intelligent Tutoring Systems, pp. 363–372. Springer (2008)
48. McLaren, B.M., Koedinger, K.R., Schneider, M., Harrer, A., Bollen, L.: Bootstrapping novice data: Semi-automated tutor authoring using student log files (2004)
49. McLaren, B.M., Lim, S.J., Koedinger, K.R.: When is assistance helpful to learning? results in combining worked examples and intelligent tutoring. In: International Conference on Intelligent Tutoring Systems, pp. 677–680. Springer (2008)
50. Moreno, R., Mayer, R.E.: Cognitive principles of multimedia learning: The role of modality and contiguity. Journal of educational psychology **91**(2), 358 (1999)
51. Mostafavi, B., Barnes, T.: Data-driven proficiency profiling: proof of concept. In: Proceedings of the Sixth International Conference on Learning Analytics & Knowledge, pp. 324–328 (2016)
52. Mostafavi, B., Barnes, T.: Evolution of an intelligent deductive logic tutor using data-driven elements. International Journal of Artificial Intelligence in Education **27**(1), 5–36 (2017)
53. Mostafavi, B., Eagle, M., Barnes, T.: Towards data-driven mastery learning. In: Proceedings of the Fifth International Conference on Learning Analytics And Knowledge, pp. 270–274 (2015)

54. Mostafavi, B., Zhou, G., Lynch, C., Chi, M., Barnes, T.: Data-driven worked examples improve retention and completion in a logic tutor. In: International Conference on Artificial Intelligence in Education, pp. 726–729. Springer (2015)
55. Muir, M., Conati, C.: An analysis of attention to student–adaptive hints in an educational game. In: International Conference on Intelligent Tutoring Systems, pp. 112–122. Springer (2012)
56. Murray, R.C., VanLehn, K.: A comparison of decision-theoretic, fixed-policy and random tutorial action selection. In: International Conference on Intelligent Tutoring Systems, pp. 114–123. Springer (2006)
57. Murray, T.: An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art. In: Authoring tools for advanced technology learning environments, pp. 491–544. Springer (2003)
58. Nelson-Le Gall, S.: Help-seeking: An understudied problem-solving skill in children. Developmental Review **1**(3), 224–246 (1981)
59. Newell, A., Simon, H.A., et al.: Human problem solving, vol. 104. Prentice-Hall Englewood Cliffs, NJ (1972)
60. Paaßen, B., Hammer, B., Price, T.W., Barnes, T., Gross, S., Pinkwart, N.: The continuous hint factory-providing hints in vast and sparsely populated edit distance spaces. arXiv preprint arXiv:1708.06564 (2017)
61. Paunonen, S.V., Ashton, M.C.: Big five predictors of academic achievement. Journal of Research in Personality **35**(1), 78–90 (2001)
62. Price, T., Zhi, R., Barnes, T.: Evaluation of a data-driven feedback algorithm for open-ended programming. International Educational Data Mining Society (2017)
63. Price, T.W., Dong, Y., Barnes, T.: Generating data-driven hints for open-ended programming. EDM **16**, 191–198 (2016)
64. Price, T.W., Dong, Y., Lipovac, D.: isnap: towards intelligent tutoring in novice programming environments. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on computer science education, pp. 483–488 (2017)
65. Price, T.W., Liu, Z., Catété, V., Barnes, T.: Factors influencing students' help-seeking behavior while programming with human and computer tutors. In: Proceedings of the 2017 ACM Conference on International Computing Education Research, pp. 127–135 (2017)
66. Price, T.W., Zhi, R., Barnes, T.: Hint generation under uncertainty: The effect of hint quality on help-seeking behavior. In: International Conference on Artificial Intelligence in Education, pp. 311–322. Springer (2017)
67. Price, T.W., Zhi, R., Dong, Y., Lytle, N., Barnes, T.: The impact of data quantity and source on the quality of data-driven hints for programming. In: International Conference on Artificial Intelligence in Education, pp. 476–490. Springer (2018)
68. Puustinen, M.: Help-seeking behavior in a problem-solving situation: Development of self-regulation. European Journal of Psychology of education **13**(2), 271 (1998)
69. Razzaq, L., Heffernan, N.T.: Hints: is it better to give or wait to be asked? In: International Conference on Intelligent Tutoring Systems, pp. 349–358. Springer (2010)
70. Rivers, K., Koedinger, K.R.: Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. International Journal of Artificial Intelligence in Education **27**(1), 37–64 (2017)
71. Roll, I., Aleven, V., McLaren, B.M., Ryu, E., d Baker, R.S., Koedinger, K.R.: The help tutor: Does metacognitive feedback improve students' help-seeking actions, skills and learning? In: International Conference on Intelligent Tutoring Systems, pp. 360–369. Springer (2006)
72. Shen, S., Chi, M.: Reinforcement learning: the sooner the better, or the later the better? In: Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, pp. 37–44. ACM (2016)
73. Shen, S., Mostafavi, B., Lynch, C., Barnes, T., Chi, M.: Empirically evaluating the effectiveness of pomdp vs. mdp towards the pedagogical strategies induction. In: International Conference on Artificial Intelligence in Education, pp. 327–331. Springer (2018)
74. Snow, R.E.: Aptitude-treatment interaction as a framework for research on individual differences in psychotherapy. Journal of consulting and clinical psychology **59**(2), 205 (1991)

75. Stamper, J., Barnes, T., Lehmann, L., Croy, M.: The hint factory: Automatic generation of contextualized help for existing computer aided instruction. In: Proceedings of the 9th International Conference on Intelligent Tutoring Systems Young Researchers Track, pp. 71–78 (2008)
76. Summerfield, C., Egner, T.: Expectation (and attention) in visual cognition. Trends in cognitive sciences **13**(9), 403–409 (2009)
77. Sweller, J.: Cognitive load during problem solving: Effects on learning. Cognitive science **12**(2), 257–285 (1988)
78. Sweller, J.: Instructional design in technical areas. camberwell. Victoria: ACER Press (1999)
79. Sweller, J.: The worked example effect and human cognition. Learning and instruction (2006)
80. Sweller, J.: Human cognitive architecture. Handbook of research on educational communications and technology pp. 369–381 (2008)
81. Tchétagni, J.M., Nkambou, R.: Hierarchical representation and evaluation of the student in an intelligent tutoring system. In: International Conference on Intelligent Tutoring Systems, pp. 708–717. Springer (2002)
82. Timms, M.J.: Using item response theory (irt) to select hints in an its. Frontiers in Artificial Intelligence and Applications **158**, 213 (2007)
83. Ueno, M., Miyazawa, Y.: Irt-based adaptive hints to scaffold learning in programming. IEEE Transactions on Learning Technologies **11**(4), 415–428 (2017)
84. Vanlehn, K.: The behavior of tutoring systems. International journal of artificial intelligence in education **16**(3), 227–265 (2006)
85. Ventura, M., Shute, V.: The validity of a game-based assessment of persistence. Computers in Human Behavior **29**(6), 2568–2572 (2013)
86. Ventura, M., Shute, V., Zhao, W.: The relationship between video game use and a performance-based measure of persistence. Computers & Education **60**(1), 52–58 (2013)
87. Villesseche, J., Le Bohec, O., Quaireau, C., Nogues, J., Besnard, A.L., Oriez, S., De La Haye, F., Noel, Y., Lavandier, K.: Enhancing reading skills through adaptive e-learning. Interactive Technology and Smart Education (2018)
88. Weerasinghe, A., Mitrovic, A.: Enhancing learning through self-explanation. In: International Conference on Computers in Education, 2002. Proceedings., pp. 244–248. IEEE (2002)
89. Weerasinghe, A., Mitrovic, A.: Supporting self-explanation in an open-ended domain. In: International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 306–313. Springer (2004)
90. Wobbrock, J.O., Findlater, L., Gergle, D., Higgins, J.J.: The aligned rank transform for nonparametric factorial analyses using only anova procedures. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp. 143–146. ACM (2011)
91. Wood, H., Wood, D.: Help seeking, learning and contingent tutoring. Computers & Education **33**(2-3), 153–169 (1999)
92. Zhou, G., Price, T.W., Lynch, C., Barnes, T., Chi, M.: The impact of granularity on worked examples and problem solving. In: CogSci (2015)

A : Unsolicited Hint Metrics for each prior proficiency group

Prior Proficiency	#Given		HJR		HNR	
	Assertions	Messages	Assertions	Messages	Assertions	Messages
	Mean (SD)	Mean (SD)	Mean (SD)	Mean (SD)	Mean (SD)	Mean (SD)
Low	48.92 (11.76)*	35.71 (10.52)	0.93 (0.09)*	0.63 (0.18)	0.83 (0.08)*	0.61 (0.17)
High	48.67 (7.80)*	30.93 (14.00)	0.92 (0.07)*	0.63 (0.15)	0.82 (0.10)*	0.62 (0.16)
All	48.82 (9.85)*	32.74 (10.64)	0.93 (0.07)*	0.63 (0.18)	0.82 (0.09)*	0.62 (0.17)

B : Comparison of Posttest Accuracy between the two conditions

Prior Proficiency	Assertions	Messages
	Mean (SD)	Mean (SD)
Low	0.74 (0.10)	0.72 (0.08)
High	0.75 (0.09)	0.73 (0.08)
All	0.74 (0.10)	0.73 (0.08)