






Enhancing LLM-Based Feedback: Insights from Intelligent Tutoring Systems and the Learning Sciences

John Stamper¹(✉) , Ruiwei Xiao¹(✉) , and Xinying Hou²(✉) 

¹ Carnegie Mellon University, Pittsburgh, PA 15213, USA
{jstamper, ruiweix}@cs.cmu.edu

² University of Michigan, Ann Arbor, MI 48109, USA
xyhou@umich.edu

Abstract. The field of Artificial Intelligence in Education (AIED) focuses on the intersection of technology, education, and psychology, placing a strong emphasis on supporting learners' needs with compassion and understanding. The growing prominence of Large Language Models (LLMs) has led to the development of scalable solutions within educational settings, including generating different types of feedback in Intelligent Tutoring Systems. However, the approach to utilizing these models often involves directly formulating prompts to solicit specific information, lacking a solid theoretical foundation for prompt construction and empirical assessments of their impact on learning. This work advocates careful and caring AIED research by going through previous research on feedback generation in ITS, with emphasis on the theoretical frameworks they utilized and the efficacy of the corresponding design in empirical evaluations, and then suggesting opportunities to apply these evidence-based principles to the design, experiment, and evaluation phases of LLM-based feedback generation. The main contributions of this paper include: an avocation of applying more cautious, theoretically grounded methods in feedback generation in the era of generative AI; and practical suggestions on theory and evidence-based feedback design for LLM-powered ITS.

Keywords: Intelligent Tutoring System (ITS) · Large Language Models (LLMs) · Generative AI (GenAI) · Hint · Formative Feedback

1 Introduction

While the term Generative AI (GenAI) has become synonymous with LLMs and exploded with the release of ChatGPT in 2022 [6], it is important to note that GenAI has been a part of the AIED community for many years. A long line of work has been seen in areas where LLMs have shown particular usefulness such as content generation [12], including generating formative feedback [51]. The rush to apply LLMs to these areas to improve AIED certainly represents a

good opportunity to help in education, but the community should not discount the decades of work that has been done in these areas before the current interest in the latest LLMs. In particular, the AIED community has decades of research on the proper way to implement hints and feedback, and while LLMs seem to be helpful in different phases of feedback generation, the approaches used should be carefully designed and evaluated through theories and empirical evidence.

This work sets forth a balanced discourse on the integration of LLMs with learning science work, focusing on leveraging both previous insights on feedback design in intelligent tutoring systems (ITSS) and contemporary advancements in generative AI. It outlines a strategic blueprint for infusing LLM-based feedback by building on previous contributions around ITSS and the learning sciences, aiming to refine feedback generation processes and achieve more effective learning results. This work underscores the significance of adhering to established educational frameworks and validates the potential of LLMs to improve feedback components in educational systems, thus offering a pathway toward more effective and responsible future educational technologies.

2 The Development of ITS Feedback Generation

In this section, we synthesized existing research on ITS feedback generation based on how they were generated. Three primary methods were identified for generating feedback: the expert-created learner model, the data-driven learner model, and the use of large language models. The first method involves experts manually inputting models of learners' potential behaviors or constraints of the problem. The second one compiles data from learners' interactions with similar problems to automatically build a model of learner behavior. The third method leverages LLMs, focusing on supplying the appropriate context and requirements, striving to generate more adaptive feedback with less human labor.

2.1 Feedback Generated from Expert-Generated Learner Model

To generate feedback for students, traditional ITSS heavily rely on experts' input on learner modeling, with an emphasis on student problem-solving states. There are two main lines of learner modeling methods in these intelligent tutors: the production rules model, which originated from Anderson's ACT-R theory [5], and the constraint-based model (CBM), which is based on Ohlsson's theory of learning from performance errors [35]. Specifically, for a given problem, a production rules model uses a set of if-then rules or example solutions [3] generated by experts to model the knowledge and possible decision-making process to solve this problem. A CBM is composed of a set of expert-written constraints that should not be broken by any potential correct solutions. When generating feedback, the learner's solution will be compared to the rule-based model or constraint-based model, and the point the student gets astray from the right track will be included in the delivered feedback.

The effectiveness of expert-generated models for both approaches has been repeatedly proven by evidence from theories and classroom studies. For example, feedback in CTAT tutors [2] is generated from production rules. Until 2016, there were 18 CTAT tutors distributed in real educational settings, used by approximately 44,000 students. One of the CTAT tutors, Andes Physics Tutor [55] with its immediate feedback, helping students achieve significantly better learning in five years' repeated measurements. CBM has also proven to be extremely effective and efficient on highly structured procedural tasks and open-ended tasks such as programming. For instance, the feedback generated by CBM in SQL-Tutor led to significantly higher learning outcomes in 4 studies during 1998–2000 [32]. The learning curve analysis further grounded the experiment result with sound psychological foundations aligning with the smooth learning curve criterion [29].

2.2 Feedback Generated from Data-Driven Learner Model

Regardless of their effectiveness, building expert models can be extremely taxing, making them hard to scale up. Furthermore, these methods have inherent limitations: experts may overlook common mistakes made by students, and both strategies often yield less-optimal results in ill-defined domains [14]. Therefore, researchers have started to seek scalable solutions by applying data-driven methods to aggregate previous students' solutions and construct the learner's model. The automated feedback generation nature of the learner model positions the data-driven feedback approach as an initial application of generative AI in creating feedback.

The earliest work of generative AI on feedback generation can be traced back to the DIAG system [12]. The system utilized the NLP model to aggregate system messages in various structures and found the feedback aggregated by functions led to higher learning gain in the classroom study. Another stream of data-driven works built on the production rule approach automated the construction of the learner's cognitive model during problem-solving. The foundation work of the data-driven cognitive model, regardless of its non-generative nature, can be found in 1997 [11], where researchers initially applied Bayesian Networks to students' problem-solving data for plan recognition and action prediction. This approach was further developed by the Hint Factory [52], which employed Markov decision processes to analyze Logic Proof Intelligent Tutor submission data [50], generating production rules and comparing student submissions to these rules to tailor hints. This method of production-rule-based feedback generation garnered attention across ITS research in more disciplines [13, 14, 41]. The subsequent classroom studies and learning factor analyses [7] confirmed this approach's efficacy and versatility across various domains, demonstrating its potential with minimal data requirements [46].

2.3 Feedback Generated from LLMs

Some major shortcomings of the data-driven approach are: 1) it relies on the quantity and quality of the training data, and 2) the feedback in these systems is in fixed templates with limited adaptation. The recent prevalence of LLMs provides opportunities to advance the field of feedback generation by generating adaptive, human-like feedback without training data. For the feedback generation pipeline, most LLM-based work takes the student’s current state together with certain prompts asking for feedback as input, treats the LLMs as a black box to process the prompt, and directly uses the output as the personalized feedback.

Indeed, LLMs enhance the scalability of adaptive feedback across various domains [20,25,31,34,48], and help in bypassing the expert model or cognitive model building process from scratch. However, there remains a significant amount of work overlooked by many researchers in the pedagogical design of feedback and the evaluation of its impact on learning. Few works are backed by learning sciences principles such as learning-by-teaching [48] and self-reflection [24], while others did not elaborate learning design considerations in their design rationale. Moreover, most of the evaluations on the LLM-based feedback systems only reported classroom usage data, and there is a lack of theoretical support or evidence on learning for these emerging systems. To guide better LLM-based feedback design and evaluation, in the next section, we highlight theoretical and empirical evidence from previous ITS and learning sciences work, hence suggesting implications for LLM-based feedback accordingly.

3 Implications for LLM-Based Feedback

Prior research in ITSs has built strong groundwork for LLM-based feedback generation. For instance, the left figure in Fig. 1 illustrates the key functions of a traditional intelligent tutoring system, showing how feedback is triggered, generated, and delivered in an ITS [21]. However, methods proven effective in traditional ITS have not yet been fully leveraged in LLM-based feedback research. Additionally, technological advancements in GenAI now enable scaling approaches previously limited by technological constraints.

This section discusses recommendations grounded in prior ITS research and new scalable opportunities from GenAI advancements in the feedback design, generation, delivery, and evaluation phases. The right figure in Fig. 1 shows how new opportunities in LLM and GenAI could fit into and extend the traditional key functions in ITS (the left one in Fig. 1), particularly around the feedback.

3.1 Trigger to Deliver the Feedback

Feedback in ITSs is often designed to be triggered either by the system or initiated by the students [3,33]. One special type of feedback is suggestions or supportive materials to help students move forward. It is common to provide

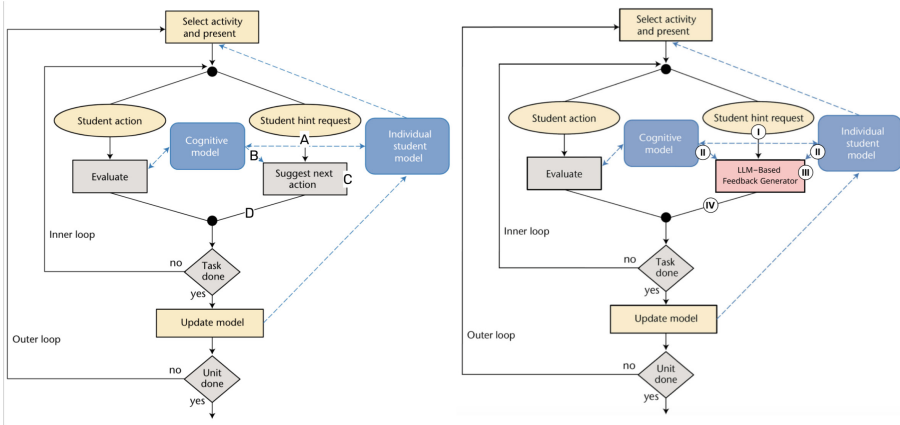


Fig. 1. Key functions of Intelligent Tutoring Systems from Koedinger et al. [21] (left) and Opportunities to Optimize LLM-based Feedback in ITS (right, adapted from [21]): I - Trigger to deliver the feedback; II - Information needed to communicate with LLMs as input; III - Content requested in the generated LLM feedback output; IV - Modality for delivering feedback to students.

such feedback based on students' request actions [17, 18, 30, 46, 57], as shown in A of Fig. 1-left. However, in some other work, concerning that novices might lack the metacognitive skills to identify when and what to ask for help, researchers believed that the model-guided feedback delivery could scaffold learners better. Therefore, Help Tutor was designed to provide metacognitive feedback automatically after the system identifies learners' misbehaviors [1]. While researchers found that having this did not lead to higher learning gains, the concerns about abusing feedback can be more serious when LLM is widely applied in educational settings. On the one hand, students nowadays can access commercial LLM products easily for timely formative feedback on their progress or even direct answers, so they are more likely to abuse them by asking for feedback too frequently. On the other hand, to avoid over-reliance on these LLM tools, some self-regulated learners might avoid asking for formative feedback, even when they are struggling or unsure about their answers. With those two new phenomena in mind, future research needs to find a balanced timing and optimized trigger to deliver feedback and establish a more effective learning experience (I in Fig. 1-right).

3.2 Information Needed to Communicate with LLM as Input

Prompt engineering is an important step when using LLMs to generate feedback in the existing learning systems. Prompt engineering refers to the practice of creating and optimizing prompts to communicate with large language models effectively. There are guidelines about how to generate effective prompts for LLMs. For example, the CLEAR framework comprises five key principles to achieve prompts for more effective AI-generated content creation [26]. Besides

these general guidelines, the required information in prompts can vary based on the specific roles AI is expected to take. When AI takes on the role of providing feedback to students' answer inputs [27,34], information resources like the grading rubric, the student answers, and the task description, are commonly included in the prompt. While current implications have already integrated such basic related information to generate prompts, one future direction could involve incorporating various types of information into prompt engineering. For example, many traditional ITSs represent students' current understanding of the subject as a model and adapt instructions to students' needs based on it [54] (B in Fig. 1-left). Such student models play an important role in understanding and identifying students' needs before offering appropriate adaptation [9]. Therefore, to enhance the quality of LLM outputs in meeting individual student's needs, different layers of information from a student model, such as the estimations of the student's knowledge level, cognitive state, strengths or weaknesses, affective state, and meta-cognitive skills, can be integrated when conducting prompt engineering in the context of education [9] (II in Fig. 1-right).

3.3 Content Requested in the Generated LLM Feedback Output

One type of ITS feedback output is a suggestion for the next action (C in Fig. 1-left). When integrating LLM into learning systems, such output can also be optimized using learning science theories (III in Fig. 1-right). Particularly, the prevalent form of feedback in recent LLM-based learning systems is a high-level natural language explanation without revealing the answer. In certain systems, persistent student inquiries for the answer lead the feedback to progressively reveal more comprehensive hints, culminating in responses that closely resemble complete, bottom-out solutions. Although such feedback can assist students in solving their problems, its effectiveness in promoting long-term learning is uncertain. Should this feedback fail to facilitate learning, the LLM-based Intelligent Tutoring System (ITS) might merely become an ineffective variant or, in the best-case scenario, a duplicate of ChatGPT. To better trigger learning, this section aims to encourage future feedback implementation guided by learning sciences frameworks such as Bloom's Taxonomy [23] and Knowledge-Learning-Instruction (KLI) framework [22], and we employ the KLI framework as a guiding example to illustrate how to select feedback content based on the knowledge type to enhance learner outcomes. The KLI framework allocates learning processes into three types: memory and fluency, induction and refinement, understanding, and sense-making, and lists seven instructional principles that are effectively robust within each category, all of which are backed by substantial experimental validation.

For enhancing memory and fluency, instructional principles such as spacing, testing, and optimized scheduling are pivotal. They focus on the timing and frequency of practice repetitions. Feedback mechanisms like flashcards, visual cues [25], or multiple-choice questions, inspired by the testing principle, offer repeated practice opportunities at short intervals, reinforcing learning. While spacing and optimization extend beyond feedback's scope, their implementation

in ITS is beneficial to effective instructional strategies in memory and fluency processes.

For tasks involving induction and refinement processes, such as solving programming or math problems, faded worked examples are effective in demonstrating the desired process with the appropriate level of assistance needed by fading the demonstration to ensure the scaffolding is within Vygotsky’s Zone of Proximal Development [28], tailored to the learner’s existing knowledge and areas for growth. The implementation of faded worked examples into intelligent tutors also resulted in higher learning efficiency and a deeper conceptual understanding of the problems in existing ITS works [49].

For understanding and sense-making processes, self-explanation questions are one of the most effective forms of feedback that elicit active learning and deeper thinking [15]. Previous ITS works also support its effects on consolidating learners’ understanding of fractions [44].

3.4 Modality for Delivering Feedback to Students

Due to technology limitations, traditional ITS systems mainly deliver feedback to students as text (D in Fig. 1-left). Recent advancements in GenAI techniques provide opportunities to expand the spectrum of feedback modalities (IV in Fig. 1-right) from text to images, audios, videos, and combinations of these modalities. As the main focus of this paper is LLM-based feedback, we look into how GenAI techniques could deliver such feedback to students in diverse modalities. We first referred to classic multimedia learning principles [10] and chose related principles that provide recommendations on how feedback might be delivered. Then, we provided examples of recent GenAI techniques and corresponding commercial applications that could be used to achieve such feedback modalities. Finally, for each principle-based feedback modality, we provided one example use scenario. We summarized these in Table 1. As new GenAI technologies are continuously growing, we believe it is important for future work to look at both the principle and technology sides to decide what feedback modalities should and could be provided in different learning contexts.

3.5 Evaluate the Quality of the Generated Feedback

Existing studies has tested the scope of LLMs’ feedback generation capabilities with data-driven system evaluation (whether the system can perform well on a comprehensive testing dataset) [16] and expert evaluation (given an evaluation matrix, the expert(s) would rate the performance of each feedback) [34, 47, 57] on mainly precision and coverage [39, 40]. Moreover, it is important to recognize the contributions of GenAI-generated feedback as it expands the array of metrics for assessing feedback quality beyond the traditional ITS evaluation scope, with new metrics like appropriateness, conciseness, and comprehensiveness. Compared to traditional feedback generation approaches, LLMs’ advantages in natural language tasks enable the generation of personalized feedback with an encouraging tone [47, 57]. Taking a step further from examining the tone, studies have

Table 1. Example feedback modalities supported by Generative AI techniques

Principle (Selected from [10])	Feedback modality	Example GenAI Techniques Involved (Example GenAI Applications)	Example Feedback Scenario
Multimedia	Images	Text-to-image (OpenAI DALL·E [36]; Stable Diffusion ^a ; Runway ^b)	Content-related images as visual cue for vocabulary memorization
Personalization	Text	Text-to-text (OpenAI GPT ^c ; Anthropic Claude ^d ; Meta Llama ^e)	Text feedback in more informal and conversational styles
Embodiment	Human-like agent	Text-to-image; Text-to-video (Runway ^f ; OpenAI Sora [37]) Text-to-speech (Deepgram ^g ; OpenAI TTS models ^h ; WellSaid Labs ⁱ)	Virtual teaching assistant avatar talking about the next-step hints
Modality	Audio	Text-to-speech	Add audio feedback as a new option to existing systems
Segmentation	Segmented Text	Text-to-text	Apply LLMs to segment one feedback into multi-levels of feedback.
Redundancy	Audio, Image, Video	Text-to-speech; Text-to-image; Text-to-video	When both audio and visual feedback is generated, allow users to turn off the text on screen
Temporal Contiguity	Audio, Image, Video	Text-to-speech; Text-to-image; Text-to-video	When both audio and visual feedback are generated, present them simultaneously
Spatial Contiguity	Text, Video	Text-to-text; Text-to-video	Place feedback close to the part that it elaborates on
Coherence	Text, Image	Text-to-text; Text-to-image	Evaluate whether existing content are on-topic or not
Signaling	Text	Text-to-text	Highlight the errors on the screen

^a<https://stability.ai/stable-image>

^b<https://runwayml.com/ai-tools/text-to-image/>

^c<https://openai.com/gpt-4>

^d<https://www.anthropic.com/news/claude-3-family>

^e<https://llama.meta.com/>

^f<https://runwayml.com/ai-tools/gen-2-text-to-video/>

^g<https://deepgram.com/>

^h<https://platform.openai.com/docs/models/tts>

ⁱ<https://wellsaidlabs.com/features/api/>

also assessed the capability of GenAI to play specific roles (e.g., instructor [53], student[48]) in the context of feedback generation. These initial assessments illuminate the potential to scale both the implementation and evaluation of specific agents and their behaviors that are crafted according to learning sciences principles, such as facilitating a growth mindset as an instructor or enhancing the dynamics of a collaborative learning environment as a helpful peer.

In addition to system and expert evaluation, classroom deployment allows researchers to gather feedback from real learners in authentic educational settings. Under this context, student usage log data [19, 20, 24, 43] and student self-reported survey data [20, 24, 38] are two commonly gathered data types for LLM-generated material evaluation. With few existing LLM-generated feedback works applying these approaches, the next steps for evaluating this feedback could involve: 1) designing controlled experiments with pre-post tests to assess learning outcomes [4, 42, 55], and 2) employing learning analytics on log data to explore feedback’s effectiveness on help-seeking [57] and learning [38] with greater detail

and nuances. For instance, the impacts of feedback on learning outcomes can be elucidated by performing a learning curve analysis to compare the slopes, which represent varying learning rates under different feedback conditions [45]. Finally, ethical concerns such as biases [56] and hallucination [8] associated with LLMs could jeopardize an effective and equitable learning environment. Therefore, it is crucial to incorporate relevant metrics into the evaluation of feedback quality and mitigate these issues with automated approaches or human intervention.

4 Conclusion

To fully utilize the potential of generative AI in the educational context, it is essential to approach its integration with pedagogical design. This paper synthesized the progression of AIED focusing on feedback generation, emphasized the importance of grounding the current LLM-based feedback generation in theoretical frameworks and evidence-based approaches to prompt its learning effectiveness, and suggested corresponding evidence-grounded implications through four feedback generation stages. We aim to evoke the awareness of AIED researchers and practitioners on the legacy of pre-LLM feedback generation efforts, and offer a toolkit as a foundational resource for designing pedagogical LLM-based feedback generation systems that foster the advancement of the AIED field in the era of generative AI.

References

1. Aleven, V., McLaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: a model of help seeking with a cognitive tutor. *Int. J. Artif. Intell. Educ.* **16**(2), 101–128 (2006)
2. Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: The cognitive tutor authoring tools (CTAT): preliminary evaluation of efficiency gains. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 61–70. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_7
3. Aleven, V., McLaren, B.M., Sewall, J., Koedinger, K.R.: A new paradigm for intelligent tutoring systems: example-tracing tutors. *Int. J. Artif. Intell. Educ.* **19**(2), 105–154 (2009)
4. Alhazmi, S., Thevathayan, C., Hamilton, M.: Interactive pedagogical agents for learning sequence diagrams. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) *AIED 2020. LNCS (LNAI)*, vol. 12164, pp. 10–14. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_2
5. Anderson, J.R.: Act: a simple theory of complex cognition. *Am. Psychol.* **51**(4), 355 (1996)
6. Baidoo-Anu, D., Ansah, L.O.: Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. *J. AI* **7**(1), 52–62 (2023)
7. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis – a general method for cognitive model evaluation and improvement. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) *ITS 2006. LNCS*, vol. 4053, pp. 164–175. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_17

8. Chen, Y., et al.: Hallucination detection: robustly discerning reliable answers in large language models. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 245–255 (2023)
9. Chrysafiadi, K., Virvou, M.: Student modeling approaches: a literature review for the last decade. *Expert Syst. Appl.* **40**(11), 4715–4729 (2013)
10. Clark, R.C., Mayer, R.E.: *E-learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. Wiley, Hoboken (2023)
11. Conati, C., Gertner, A.S., VanLehn, K., Druzdzel, M.J.: On-line student modeling for coached problem solving using Bayesian networks. In: Jameson, A., Paris, C., Tasso, C. (eds.) *User Modeling. ICMS*, vol. 383, pp. 231–242. Springer, Vienna (1997). https://doi.org/10.1007/978-3-7091-2670-7_24
12. Di Eugenio, B., Fossati, D., Yu, D., Haller, S.M., Glass, M.: Natural language generation for intelligent tutoring systems: a case study. In: *AIED*, pp. 217–224 (2005)
13. Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, C., Chen, L.: Data driven automatic feedback generation in the ilist intelligent tutoring system. *Technol. Instr. Cogn. Learn.* **10**(1), 5–26 (2015)
14. Fournier-Viger, P., Nkambou, R., Nguifo, E.M.: Learning procedural knowledge from user solutions to ill-defined tasks in a simulated robotic manipulator. In: Romero, et al. (eds.) *Handbook of Educational Data Mining*, pp. 451–465 (2010)
15. Hausmann, R.G., VanLehn, K.: Explaining self-explaining: a contrast between content and generation. *Front. Artif. Intell. Appl.* **158**, 417 (2007)
16. Hellas, A., Leinonen, J., Sarsa, S., Koutchme, C., Kujanpää, L., Sorva, J.: Exploring the responses of large language models to beginner programmers’ help requests. In: Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1, pp. 93–105 (2023)
17. Hou, X., Ericson, B.J., Wang, X.: Using adaptive parsons problems to scaffold write-code problems. In: Proceedings of the 2022 ACM Conference on International Computing Education Research-Volume 1, pp. 15–26 (2022)
18. Hou, X., Ericson, B.J., Wang, X.: Understanding the effects of using parsons problems to scaffold code writing for students with varying CS self-efficacy levels. In: Proceedings of the 23rd Koli Calling International Conference on Computing Education Research, pp. 1–12 (2023)
19. Kazemitabaar, M., Hou, X., Henley, A., Ericson, B.J., Weintrop, D., Grossman, T.: How novices use LLM-based code generators to solve cs1 coding tasks in a self-paced learning environment. In: Proceedings of the 23rd Koli Calling International Conference on Computing Education Research, pp. 1–12 (2023)
20. Kazemitabaar, M., et al.: Codeaid: evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. arXiv preprint [arXiv:2401.11314](https://arxiv.org/abs/2401.11314) (2024)
21. Koedinger, K., Brunskill, E., Baker, R., Mclaughlin, E., Stamper, J.C.: New potentials for data-driven intelligent tutoring system development and optimization. *AI Mag.* **34**, 27–41 (2013). <https://api.semanticscholar.org/CorpusID:13189100>
22. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning. *Cogn. Sci.* **36**(5), 757–798 (2012)
23. Krathwohl, D.R.: A revision of bloom’s taxonomy: an overview. *Theory Pract.* **41**(4), 212–218 (2002)

24. Kumar, H., Musabirov, I., Williams, J.J., Liut, M.: Quickta: exploring the design space of using large language models to provide support to students. In: Learning Analytics and Knowledge Conference. Learning Analytics and Knowledge Conference 2023 (LAK'23). ACM, Arlington, Texas (2023)
25. Lee, J., Lan, A.: Smartphone: exploring keyword mnemonic with auto-generated verbal and visual cues. In: Wang, N., Rebolledo-Mendez, G., Matsuda, N., Santos, O.C., Dimitrova, V. (eds.) AIED 2023. LNCS, vol. 13916, pp. 16–27. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-36272-9_2
26. Lo, L.S.: The clear path: a framework for enhancing information literacy through prompt engineering. *J. Acad. Librariansh.* **49**(4), 102720 (2023)
27. Lu, C., Cutumisu, M.: Integrating deep learning into an automated feedback generation system for automated essay scoring. International Educational Data Mining Society (2021)
28. Malik, S.A.: Revisiting and re-representing scaffolding: the two gradient model. *Cogent Educ.* **4**(1), 1331533 (2017)
29. Martin, B., Koedinger, K.R., Mitrovic, A., Mathan, S.: On using learning curves to evaluate its. In: AIED, pp. 419–426 (2005)
30. McLaren, B.M., Richey, J.E., Nguyen, H., Hou, X.: How instructional context can impact learning with educational technology: lessons from a study with a digital learning game. *Comput. Educ.* **178**, 104366 (2022)
31. McNichols, H., Zhang, M., Lan, A.: Algebra error classification with large language models. In: Wang, N., Rebolledo-Mendez, G., Matsuda, N., Santos, O.C., Dimitrova, V. (eds.) AIED 2023. LNCS, vol. 13916, pp. 365–376. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-36272-9_30
32. Mitrovic, A., Mayo, M., Suraweera, P., Martin, B.: Constraint-based tutors: a success story. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 931–940. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45517-5_103
33. Mitrovic, A., Ohlsson, S., Barrow, D.K.: The effect of positive feedback in a constraint-based intelligent tutoring system. *Comput. Educ.* **60**(1), 264–272 (2013)
34. Nguyen, H.A., Stec, H., Hou, X., Di, S., McLaren, B.M.: Evaluating ChatGPT's decimal skills and feedback generation in a digital learning game. In: Viberg, O., Jivet, I., Muñoz-Merino, P., Perifanou, M., Papathoma, T. (eds.) EC-TEL 2023. LNCS, vol. 14200, pp. 278–293. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-42682-7_19
35. Ohlsson, S.: Learning from performance errors. *Psychol. Rev.* **103**(2), 241 (1996)
36. OpenAI: Dall.e 2 (2023). <https://openai.com/dall-e-2/>. Accessed 12 Mar 2024
37. OpenAI: Sora (2024). <https://openai.com>. Accessed 12 Mar 2024
38. Pankiewicz, M., Baker, R.S.: Navigating compiler errors with AI assistance—a study of GPT hints in an introductory programming course. arXiv preprint [arXiv:2403.12737](https://arxiv.org/abs/2403.12737) (2024)
39. Phung, T., et al.: Generating high-precision feedback for programming syntax errors using large language models. arXiv preprint [arXiv:2302.04662](https://arxiv.org/abs/2302.04662) (2023)
40. Phung, T., et al.: Automating human tutor-style programming feedback: leveraging GPT-4 tutor model for hint generation and GPT-3.5 student model for hint validation. In: Proceedings of the 14th Learning Analytics and Knowledge Conference, pp. 12–23 (2024)
41. Price, T.W., Dong, Y., Barnes, T.: Generating data-driven hints for open-ended programming. International Educational Data Mining Society (2016)

42. Price, T.W., Zhi, R., Barnes, T.: Hint generation under uncertainty: the effect of hint quality on help-seeking behavior. In: André, E., Baker, R., Hu, X., Rodrigo, M.M.T., du Boulay, B. (eds.) AIED 2017. LNCS (LNAI), vol. 10331, pp. 311–322. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61425-0_26
43. Qi, J.Z.P.L., Hartmann, B., Norouzi, J.D.N.: Conversational programming with LLM-powered interactive support in an introductory computer science course. In: NeurIPS'23 Workshop on Generative AI for Education (GAIED) (2023)
44. Rau, M.A., Alevan, V., Rummel, N.: Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. In: AIED, pp. 441–448 (2009)
45. Rivers, K., Harpstead, E., Koedinger, K.R.: Learning curve analysis for programming: which concepts do students struggle with? In: ICER, vol. 16, pp. 143–151. ACM (2016)
46. Rivers, K., Koedinger, K.R.: Data-driven hint generation in vast solution spaces: a self-improving python programming tutor. *Int. J. Artif. Intell. Educ.* **27**, 37–64 (2017)
47. Roest, L., Keuning, H., Jeuring, J.: Next-step hint generation for introductory programming using large language models. In: Proceedings of the 26th Australasian Computing Education Conference, pp. 144–153 (2024)
48. Schmucker, R., Xia, M., Azaria, A., Mitchell, T.: Ruffle&Riley: towards the automated induction of conversational tutoring systems. arXiv preprint [arXiv:2310.01420](https://arxiv.org/abs/2310.01420) (2023)
49. Schwonke, R., Wittwer, J., Alevan, V., Salden, R., Krieg, C., Renkl, A.: Can tutored problem solving benefit from faded worked-out examples. In: European Cognitive Science Conference, pp. 23–27 (2007)
50. Stamper, J.: Automating the generation of production rules for intelligent tutoring systems. In: Proceedings of 9th International Conference on Interactive Computer Aided Learning (2006)
51. Stamper, J., Barnes, T., Croy, M.: Enhancing the automatic generation of hints with expert seeding. In: Alevan, V., Kay, J., Mostow, J. (eds.) ITS 2010. LNCS, vol. 6095, pp. 31–40. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13437-1_4
52. 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)
53. Tack, A., Piech, C.: The AI teacher test: measuring the pedagogical ability of blender and GPT-3 in educational dialogues. arXiv preprint [arXiv:2205.07540](https://arxiv.org/abs/2205.07540) (2022)
54. VanLehn, K.: Student modeling. In: Foundations of Intelligent Tutoring Systems, pp. 55–78 (2013)
55. VanLehn, K., et al.: The Andes physics tutoring system: five years of evaluations. In: AIED, pp. 678–685 (2005)
56. Wei, Y., Carvalho, P.F., Stamper, J.: Uncovering name-based biases in large language models through simulated trust game. arXiv preprint [arXiv:2404.14682](https://arxiv.org/abs/2404.14682) (2024)
57. Xiao, R., Hou, X., Stamper, J.: Exploring how multiple levels of GPT-generated programming hints support or disappoint novices. arXiv preprint [arXiv:2404.02213](https://arxiv.org/abs/2404.02213) (2024)