PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

LLM-Based Intelligent Tutoring Systems: A Survey

Li Kong[1,4] , **Jianwen Sun**[2] , **Junsheng Zhou**[1,4*] , **Vincent Ng**[3]
[1]School of Computer and Electronic Information, Nanjing Normal University, Nanjing, China
[2]Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan, China
[3]Human Language Technology Research Institute, University of Texas at Dallas, Richardson, Texas, USA
[4]Jiangsu Provincial Engineering Research Center for Intelligent Information Technology and Software, Nanjing, China
kongli@nnu.edu.cn, sunjw@ccnu.edu.cn, zhoujs@njnu.edu.cn, vince@hlt.utdallas.edu

Abstract

Large Language Models (LLMs) are reshaping the design and capabilities of intelligent tutoring systems (ITS) by providing powerful generative, reasoning and interaction abilities, which surpass traditional rule-based approaches. This survey presents a structured overview of LLM-based ITS and analyzes how these models transform classical system components and architectures. We first review the foundational concepts of traditional ITS and introduce the functional roles of the main components, followed by LLM-based techniques and related datasets for realizing each of these components. Furthermore, we examine the key application domains and concludes the survey by outlining future research directions.

1 Introduction

In the field of education, intelligent tutoring systems (ITS) powered by artificial intelligence (AI) have emerged as a prominent area of research. They usually mimic human tutors by modeling learners' knowledge, skills and progress to diagnose misconceptions and deliver timely, targeted feedback [Nye *et al.*, 2014]. Building upon early research on computer-assisted instruction (CAI), ITS have continuously evolved through advances in cognitive science and AI. ITS is an area that draws upon a broad spectrum of fields, such as computer science, information science, cognitive science, educational psychology, and linguistics. Nowadays, ITS are widely deployed in the teaching and learning of disciplines such as mathematics, physics, medicine and computer science, to provide personalized and adaptive instruction for learners [Sleeman and Brown, 1982]. Over several decades, ITS research, exemplified by systems like AutoTutor and MetaTutor [Rus *et al.*, 2010], has shown strong effects on learning outcomes [Conati, 2009].

Despite steady progress, traditional ITS have long been constrained by their reliance on rule-based architectures and hand-crafted, domain-specific knowledge representations [Conati, 2009]. These systems require extensive expert engineering to encode curricular content, design dialogue flows and model learner states, resulting in high development costs and limited scalability [Singh and Singh, 2022]. Although these systems have achieved notable success in well-defined domains, they often exhibit several core limitations, including (1) inflexible knowledge representation that struggles with open-ended or evolving content; (2) constrained natural language interaction due to predefined scripts and shallow parsing; and (3) insufficient personalization beyond coarse-grained learner profiles. Their limited adaptability leads to brittle interactions and high costs in both system building and maintenance, which in turn restrict their deployment across diverse learning contexts.

The rapid progress of Large Language Models (LLMs) since 2022 has fundamentally changed the research landscape in ITS. Models such as GPT, Gemini and their contemporaries have demonstrated emergent abilities in reasoning, natural dialogue and pedagogical adaptation, enabling ITS to provide context-aware and personalized guidance at scale [Tabarsi, 2025]. Critically, LLMs can unify previously isolated components of ITS into a *single* cognitive core. This integration not only reduces the heavy authoring and expert-engineering burden that earlier ITS have suffered, but also enhances responsiveness and generalization across domains. Their emergent abilities in reasoning, explanation and self-reflection have enabled a unified framework that integrates domain knowledge, pedagogical strategies and learner modeling within a single adaptive engine. Nevertheless, integrating LLMs into ITS is not without challenges. Issues such as factual hallucination, lack of alignment with established learning theories, and limited interpretability necessitate careful design and evaluation.

Our goal in this paper is to provide the general AI audience with a systematic yet accessible overview of recent research on LLM-based ITS. As noted above, given how rapidly LLMs have transformed the landscape in this area of research, we believe that time is ripe to summarize the recent developments. While this survey would be of primary interest to researchers in the AI in Education (AIED) community, we believe that it would also be of interest to researchers in different AI subcommunities, such as NLP, human-computer interaction, applied machine learning, automated reasoning, and Generative AI, given the broader impacts associated with ITS research. We believe that this survey can provide a good starting point for researchers in different subareas of AI to tap into

*Corresponding author

IJCAI-ECAI 2026 PREPRINT

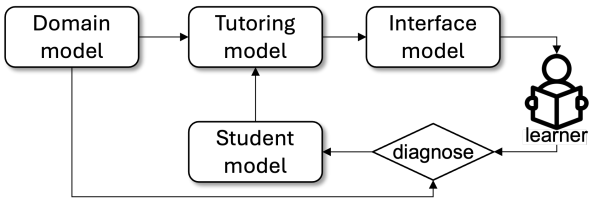

Figure 1: Modules in Classical ITS.

this exciting, rapidly expanding area of research.

While there have been surveys focusing on AI for education [Wang *et al.*, 2024c; Chu *et al.*, 2025], to the best of our knowledge, this survey is the first to provide a focused and integrative architectural perspective on tutoring intelligence in the LLM era, specifically through a discussion of how LLMs reconfigure classical modules into a unified cognitive framework. Unlike the broader educational overviews of previous work, we systematically analyze how LLMs reshape tutoring system design and offer a synthesis of advances in system architecture, core capabilities and application scenarios.

The remainder of this paper is organized as follows. Section 2 provides background knowledge of classical ITS research. Section 3 presents different architectures of LLM-based ITS and the roles of LLMs within them. We discuss the key capabilities of LLM-based ITS in Section 4, including the tasks, datasets and methods, and application scenarios in Section 5. Section 6 concludes with future directions.

2 Preliminaries

The concept of ITS can be traced back to the 1980s [Sleeman and Brown, 1982], which is commonly referred to as a teaching system that provides personalized guidance. The architecture of classical ITS is typically organized into the following four interdependent modules, as illustrated in Figure 1:

- *Domain Model* encodes expert knowledge and problem-solving paths within a specific subject area.
- *Student Model* tracks the learner's knowledge, skills and misconceptions, which is a different concept from "Student" in knowledge distillation.
- *Tutoring Model* selects instructional strategies and intervention rules based on the learner's current state.
- *Interface Model* manages the interactions between the learner and the system via natural communication.

Together, these modules form a closed-loop architecture, in which the *Interface Model* captures learner behavior and delivers responses, the *Student Model* infers cognitive states, and the *Tutoring Model* decides on pedagogical actions grounded in the *Domain Model*. An intelligent tutoring system can continuously monitor and dynamically adjust the learning process, thereby providing a personalized teaching experience that closely resembles human tutoring.

While the classical framework in Figure 1 is virtually adopted by all traditional ITS, the clear boundaries among the four modules in this framework have become increasingly blurred, both functionally and architecturally, in LLM-based ITS. For instance, a LLM may simultaneously model learner knowledge and provide target feedback within a single conversational context *without* the need for having separate modules (i.e., the *Student Model* and the *Tutoring Model*) for handling these two tasks. Moreover, LLM-based architectures often integrate additional components such as knowledge retrievers, learning monitors and multi-agent coordination mechanisms to enhance adaptability and interpretability. Therefore, rather than replacing the classical architecture, LLMs reconfigure and expand it.

Nevertheless, we note that some LLM-based ITS have chosen to stick with this classical framework. In these ITS, some have chosen to implement each module using LLMs, while others have adopted a *hybrid* approach where some modules are implemented using LLMs while others are implemented using traditional methods (i.e., via rule-based [Conati, 2009] or expert-system [Singh and Singh, 2022] mechanisms).

3 Taxonomy of LLM-Based ITS

As mentioned at the end of the last section, LLM-based ITS can be broadly divided into two categories: (1) ITS that adopt a *modular* architecture, in which the system is designed based on the modules in classical ITS, where one or more modules are implemented using LLMs; and (2) ITS that adopt an *integrated* architecture, in which the system is designed without a clear division of labor into different modules. Below we discuss the roles played by LLMs in these two architectures.

3.1 The Modular Architecture

In this subsection, we discuss the *functional* roles of LLMs when they are used to implement each of the modules in the classical ITS framework. Note that the *Interface Model* manages the I/O interface and is therefore not closely related to LLMs. Consequently, the discussion below focuses on the remaining three modules.

LLM for Domain Model. In classical ITS, the *Domain Model* encodes expert knowledge through rules, ontologies, or traditional problem-solving models. LLMs enhance the *Domain Model* in three key ways: (1) knowledge base (KB) / knowledge graph (KG) construction and evolution, where LLMs instantiate and augment a KB or KG with the vast amount of knowledge encoded implicitly within their parameters (**D1**) [Agrawal *et al.*, 2024]; (2) knowledge retrieval and grounding, where LLMs retrieve relevant knowledge from the KBs/KGs, allowing the system to apply the retrieved knowledge effectively for solving problems [Cheng *et al.*, 2025] (**D2**); (3) instructional path planning, where LLMs assemble the relevant knowledge retrieved into instructional sequences [Chen *et al.*, 2024] (**D3**). These new functions preserve structured curriculum control while leveraging LLMs' generation and Chain-of-Thought (CoT) reasoning to produce contextual knowledge and solution path, which is valuable for adaptive tutoring.

LLM for Student Model. LLMs enhance learner profiling by addressing four key tasks: (1) knowledge tracing, where LLMs track and predict a learner's knowledge state over time based on their interactions and responses (**S1**) [Wang *et al.*, 2025]; (2) student profiling, where LLMs enable

IJCAI-ECAI 2026 PREPRINT

Module	Classical ITS	LLM-based ITS
Domain	Explicit, Static	Implicit, Flexible
Student	Coarse, Fixed	Fine-grained, Dynamic
Tutoring	Prescriptive, Rigid	Adaptive, Generative

Table 1: Comparison of modules in classical and LLM-based ITS.

dynamic, multifaceted representations of cognitive and affective states, such as engagement, confusion, or personality, inferred from natural language interactions (**S2**) [Zhu *et al.*, 2025]; (3) student simulation, where LLMs simulate the learner's thought process, generating potential responses or strategies that reflect a learner's evolving understanding (**S3**) [Wu *et al.*, 2025]; and (4) intent recognition, where LLMs identify underlying goals from student queries and responses (**S4**) [Puech *et al.*, 2025]. Through the integration of LLMs, the *Student Model* offers a responsive and pedagogically interpretable portrait of the learner.

LLM for Tutoring Model. The *Tutoring Model* governs the instructional strategy, determining how to provide appropriate feedback and guidance to learners. LLMs can significantly enhance this component through six key tasks: (1) correction, where LLMs evaluate, identify errors and provide corrections (**T1**); (2) hinting, offering incremental clues without revealing the answer directly (**T2**); (3) questioning, posing open-ended or Socratic questions to stimulate further thinking (**T3**) [Ma *et al.*, 2025; Ang *et al.*, 2023]; (4) encouragement, delivering motivational feedback to maintain learner engagement and motivation (**T4**) [Yang *et al.*, 2025]; (5) explanation, providing explanations of concepts or solutions (**T5**); and (6) progression, generating task instructions based on the learner's progress (**T6**). These features enable the tutoring process to be more responsive, adaptive, and personalized.

Table 1 presents a comparison of these modules in classical ITS and LLM-based ITS. Traditional *Domain Model* uses explicit knowledge with predefined rules, *Student* is coarse and updates slowly, *Tutoring* relies on templates, while LLM can use internal knowledge, support fine-grained and dynamic modeling, produce context-aware and adaptive instruction.

3.2 The Integrated Architecture

ITS adopting the integrated architecture use a LLM as the sole tutoring engine. Student queries and interactions are passed via a *prompt* (i.e., a set of natural-language instructions needed to accomplish a target task) to the LLM and get a response [Wang *et al.*, 2024a; Chang and Ginter, 2024]. This approach typically relies on *prompt engineering* (e.g., determining how the instructions in the prompt should be worded), *few-shot* prompting (i.e., the inclusion of a few labeled training examples in the prompt to help an LLM better understand how to accomplish the given task), and *chain-of-thought* (CoT) (i.e., asking the LLM to output not only the answer for the target task but also the reasoning steps used by the LLM to arrive at the answer in the form of natural-language sentences), and uses the context window to capture recent interaction history. This simple architecture is suited for broad tutoring scenarios, where adaptability and general reasoning capabilities of LLMs can effectively support personalized and open-ended learning experiences.

ID	Definition
	Instructional Content Synthesis
I1	**Question Generation**: Generate questions from materials.
I2	**Answer Generation**: Generate answer for a question.
I3	**Support Material Generation:** Generate support materials.
I4	**Explanation Generation**: Explain concepts/questions.
I5	**Structured Knowledge Generation**: Generate KBs/KGs.
	Learner Response Evaluation
R1	**Scoring**: Assign a score to a response.
R2	**Narrative Evaluation**: Narrate the strengths/weaknesses.
R3	**Constructive Feedback**: Generate feedback to improve responses.
	Learner Modeling
M1	**Knowledge Tracing**: Predict student's knowledge level.
M2	**Misconception Diagnosis**: Identify error types from responses.
	Pedagogical Dialogue Management
P1	**Socratic Questioning**: Generate Socratic questions from responses.
P2	**Scaffolded Hinting**: Generate progressive hints on questions.
P3	**Affect-Aware Feedback**: Generate affect-aware feedback.

Table 2: Common tasks addressed in LLM-based ITS research.

4 Key Capabilities of LLM-Based ITS

In this section, we explore the key capabilities of LLM-based ITS, which can be broadly categorized into four components, including *Instructional Content Synthesis*, *Learner Response Evaluation*, *Learner Modeling*, and *Pedagogical Dialogue Management*. Each of these capabilities plays a pivotal role in addressing different aspects of the learning process.

4.1 Instructional Content Synthesis

LLMs have endowed ITS with powerful capabilities for instructional content synthesis, including the generation of disciplinary questions (I1), answers to questions (I2), supporting materials (I3), explanations to questions/answers (I4), and structured KBs/KGs (I5) [see Table 2 for their definitions]. The generative capability of LLMs provides a distinct advantage over static template-based or retrieval-based systems in terms of flexibility, diversity and contextual appropriateness.

Datasets

To support the training and evaluation of models for generating subject-specific problems, answers to questions and explanations of questions/answers, numerous datasets provide critical benchmarks, such as ScienceQA [Lu *et al.*, 2022] for multi-modal scientific question answering (QA), EduAdapt [Naeem *et al.*, 2025] for QA in K-12 education, EduMath [Cheng *et al.*, 2025] for mathematical question generation, and GSM8K [Cobbe *et al.*, 2021] for arithmetic reasoning. Although these datasets are primarily framed around QA, they often include rich auxiliary materials such as solution steps and illustrative examples. These elements make them valuable for generating learning support materials and serve as potential resources for building KBs and KGs.

Methods

To generate rich educational materials aligned with curricular objectives, researchers have employed strategies that include

IJCAI-ECAI 2026 PREPRINT

(1) zero-shot/few-shot prompting, which is often supplemented with CoT reasoning, and (2) supervised fine-tuning (SFT) on task-specific data.[1]

For the *prompting-based* methods, Wang et al. [2024a] employ GPT-4 in combination with few-shot prompting to create explanations towards math problems. Analysis of the problems, related to knowledge concepts, is performed within the reasoning process of the LLM. CyberGen [Agrawal *et al.*, 2024] enhances this strategy with external knowledge, extracting triples from educational resources to generate QA pairs, which is stored in a structured KB (D1) and can be used for interactive learning. This kind of knowledge-grounded generation addresses factual accuracy and reduces hallucinations by grounding the content in authoritative sources such as textbooks and scientific articles. Following educational objectives of the learner, EQPR [Cheng *et al.*, 2025] retrieves related content from a KB with a separate component (D2) and performs planning via Monte Carlo Tree Search (MCTS), which orchestrates iterative cycles of critique and reflection to generate math questions (T6).

As for SFT methods, Jia et al. [2025] generate course materials aligning with the framework of Gagné's Nine Events of Instruction[2] for math using SFT on Llama via LoRA[3], outperforming GPT models. For question generation under specific user needs (T6), Wan et al. [2025] use SFT and direct preference optimization (DPO) to guide LLMs, which proves effective on Qwen2 and DeepSeekMath. They define a math task space with question types, concepts and difficulty levels, which offers control for new question generation (D1).

Hierarchical content planning facilitates the development of a strategic blueprint for complex content creation, encompassing both broad strokes and detailed components for items such as essays [He *et al.*, 2024] and solutions [Lyu *et al.*, 2024], thereby enhancing organizational coherence. For example, to generate student essays, He et al. [2024] employ a two-step *planning* strategy, involving sketch planning and dialectical planning, followed by final essay generation using Llama2 via zero-shot prompting. Knowledge about writing strategies and formats are both embedded within the prompt template. Given a question posed by a student, RPG [Lyu *et al.*, 2024] uses Llama2 to implement a tutor via a two-stage framework, where the *plan* stage (1) generates the topics on which the answer should be centered, and (2) the *answer* stage generates answers (T5) using the topics.

4.2 Learner Response Evaluation

This functionality enables the system to perform a real-time evaluation of student responses expressed in the form of natural or programming language, and provide guidance, which is directly related to function T1 in the *Tutoring Model*. LLMs are capable of evaluating student-submitted content, such as essays [Wang and Liu, 2025], short answers [Chang and Ginter, 2024], code/programs [Havare *et al.*, 2025], by assigning scores (R1)[4] and providing detailed narrative evaluation, which may contain identified weaknesses as well as strengths (R2). Rather than merely indicate whether an answer is right, constructive feedback also guides the student towards a better response (R3). See Table 2 for the definition of these tasks.

Datasets

The development and evaluation of these assessment capabilities rely on specialized annotated datasets. For example, ICLE++ [Li and Ng, 2024] and ASAP++ [Mathias and Bhattacharyya, 2018] provide large-scale essay responses with both holistic and dimension-specific scores (R1). By comparison, the Short Answer Feedback dataset (SAF) [Filighera *et al.*, 2022] contains elaborated feedback explaining why points were gained or lost (R2) and providing guidance on how to improve the response based on the errors (R3). For assessing student answers to subject-related questions, datasets originally developed for QA systems can also been utilized. For example, EduAdapt [Naeem *et al.*, 2025], GSM8K [Cobbe *et al.*, 2021] and ScienceQA [Lu *et al.*, 2022] provide correct answers, so that we can evaluate whether a student response is correct or wrong (R1) and what is the difference from the correct answer of open-end questions (R2).

Methods

For scoring student responses, both prompting and SFT-based methods have been developed. For the *prompting-based* methods, Zhu et al. [2025] apply few-shot prompting in combination with GPT-4 and LLaMA-Vision to score student responses from K–12 science education containing handwritten text and drawings. Similarly, Chang et al. [2024] use GPT via zero/one-shot prompting to grade short student responses across disciplines like biology and psychology, predicting a numeric or categorical score. For SFT methods, T-MES [Wang and Liu, 2025] uses fine-tuning with a Mixture-of-Experts framework with specialized expert networks to predict the overall score and the dimension-specific scores of student essays. Havare et al. [2025] optimize Qwen-2.5-Coder with SFT and DPO to grade introductory C++ programming assignments by generating multi-criteria scores.

As for feedback provision and guidance, both prompting and SFT methods have been used. For instance, ITS-LLM [Gaeta *et al.*, 2025] produces detailed comments in response to students' answers in computer science courses through zero-shot prompt engineering. Specifically, it evaluates answers using the disciplinary knowledge retrieved from KB (D2), including facts, rules and concepts, and provides personalized feedback according to pre-collected student profiles (S2) including learning preferences and progress. Sonkar et al. [2024] train a LLM via SFT and Learning from Human Preferences (LHP) to generate structured feedback based on student responses of biological questions, including

[1] To facilitate understanding, below we link the description of each method to the module-specific functions discussed in Section 3.1 whenever applicable.

[2] Gagné's Nine Events of Instruction is a widely used instructional design framework developed in the 1960s to ensure effective learning by aligning instructional events with internal cognitive processes. See Hricko [2008] for details.

[3] LoRA, or Low-Rank Adaptation, is a method for efficiently fine-tuning LLMs [Hu *et al.*, 2021].

[4] For essays, the scores may include the holistic score, which summarizes overall quality, and the trait scores, which evaluate specific dimensions of quality, such as COHERENCE; for code/programs and short answers, the score reflect the overall quality.

IJCAI-ECAI 2026 PREPRINT

evaluation categories (correct/incorrect) and corrective guidance. They build a KB based on biology textbooks (D1) and retrieve knowledge during response evaluation (D2). By limiting model output to the scope covered by the knowledge in the KB, it can reduce the generation of irrelevant or incorrect content. Such a domain-specific *knowledge-enhanced* strategy has been proven effective in relieving hallucination [Lewis *et al.*, 2020], which places strict requirements on data quality.

4.3 Learner Modeling

LLM-based ITS increasingly emphasize learner modeling that can adapt to individual learners' knowledge states, cognitive profiles, and interaction histories. The core tasks are (1) knowledge tracing (M1), which predicts the knowledge level of a student on concepts, and (2) misconception diagnosis, which identifies the error types or cognitive slips (M2) based on student interaction data [see Table 2 for details]. The focus of this capability is students rather than their responses. Note that this capability is reminiscent of the responsibilities of the *Student Model* in classical ITS. In fact, the two core tasks in Learning Modeling are similar to some for the roles played by LLMs for the *Student Model* (see Section 3.1). Specifically, M1 is the same as S1, and M2 is related to S3.

Datasets

Datasets developed to support knowledge tracing and error diagnosis typically consist of large-scale student–problem interaction logs annotated with correctness, skill or concept tags, temporal information, error labels, and solution steps. Standard benchmark datasets for evaluating knowledge tracing models include ASSISTments, Junyi [Wang *et al.*, 2025], and EdNet [Choi *et al.*, 2020], which provide long interaction sequences across diverse learners and knowledge components. In contrast, datasets such as Eedi[5] and Math Misconceptions and Errors Dataset (MAE)[6] explicitly annotate incorrect responses with misconception or error types, enabling diagnostic modeling beyond binary correctness.

Methods

Knowledge tracing (M1) methods have employed both prompting and SFT. For the prompting methods, Wu et al. [2025] use GPT-4o and Claude to model student cognitive prototypes in programming (S3), leveraging learning records to construct knowledge graphs that represent a student's concept mastery (D1). KCD [Dong *et al.*, 2025] employs GPT-3.5 to generate textual diagnoses of students' cognitive states and attributes of exercise using zero-shot prompting, which enhances knowledge tracking. For SFT, Liu et al. [2025b], by using interaction histories, fine-tune Llama-3 to assess children's programming skills (S1) and track their knowledge gaps in programming (S3). To help determine if a student masters a set of knowledge concepts, they retrieve relevant knowledge from a KB that they constructed from PDF files and slides (D1) and employ the retrieved knowledge to improve learner modeling. LLM-KT [Wang *et al.*, 2025] performs parameter-efficient fine-tuning on LLaMA2 via LoRA to model student knowledge tracing (S1). LLMKT [Scarlatos *et al.*, 2024] fine-tunes Llama-3 to perform knowledge tracing, modeling student mastery across multiple knowledge components in tutor-student math dialogues. Similarly, Yin et al. [2023] fine-tune a LLM via contrastive learning to enhance knowledge tracing and diagnose students' evolving knowledge proficiency and error patterns from sequential learning activities in math and engineering.

To diagnose misconceptions (M2), FineCD [Li *et al.*, 2025] uses GPT-4 with zero-shot prompting to perform fine-grained cognitive diagnosis by analyzing students' potential error reasons and modeling their specific knowledge misconceptions.

Learner modeling typically involves monitoring a learner's progress over an extended period of time. As a LLM acquires new information about the learner (e.g., interaction patterns) it tries to model, the new information may overwrite the old information that the LLM has about the learner due to *catastrophic forgetting*. To address this problem, LLM-KT [Wang *et al.*, 2025] integrates an additional sequence encoder that compresses and retains students' historical interaction patterns as external memory, which could also be retrieved for occurrence patterns and correlations of stored questions or concepts related to the current problem (D2).

4.4 Pedagogical Dialogue Management

This capability focuses on engaging students in multi-turn, open-ended dialogue via interaction to stimulate deep reasoning. There are three core tasks associated with this capability: employing Socratic questioning to help students explore complex ideas and uncover underlying assumptions themselves (P1), providing scaffolded hinting to provide implicit guidance (P2), and generating affect-aware feedback and encouragement to students (P3) [see Table 2 for their definitions]. This capability is reminiscent of the functions that LLMs are being used to perform for the *Tutoring Model* (see Section 3.1). Specifically, P1 is related to function T3 in the *Tutoring Model*, P2 is related to T2, and P3 is related to T4.

Datasets

Diverse dialogue-centric datasets have been developed to support pedagogical dialogue management. For Socratic questioning (P1), representative datasets include (1) SocraticQ, which comprises the first large-scale collection of Socratic QA pairs derived from real-world opinion-driven discussions [Ang *et al.*, 2023]; (2) SocraTeach, which contains well-designed Socratic-style multi-round teaching dialogues grounded in fundamental mathematical problems [Liu *et al.*, 2024]; and (3) SocraticMATH, which also provides Socratic-style conversations of mathematical problems with extra knowledge [Ding *et al.*, 2024].

For scaffolded hint generation (P2), datasets derived from large-scale tutoring platforms, such as ASSISTments [Wang *et al.*, 2025] and EdNet [Choi *et al.*, 2020] provide step-level problem-solving traces with explicitly ordered hints and observable student–hint interaction behaviors, enabling modeling of adaptive support with varying levels of specificity.

To address affect-aware feedback generation (P3), datasets for the Emotional Support Conversation task such as ES-

[5]https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics

[6]https://huggingface.co/datasets/nanote/algebra_misconceptions

IJCAI-ECAI 2026 PREPRINT

Conv [Liu *et al.*, 2021], ESD and ESD-CoT [Zhang *et al.*, 2024], which normally include problem type, emotion type and situation description, are highly valuable.

Methods

Research on interactive tutoring dialogue primarily focuses on Socratic dialogue generation, where guided and structured generation methods, including prompting, fine-tuning, and reinforcement learning of LLMs [Shridhar *et al.*, 2022; Kumar and Lan, 2024] are employed. For instance, a *multi-agent* method is used for role-based dialogue orchestration in SocraticLM [Liu *et al.*, 2024], which is fine-tuned from ChatGLM3 constructed through a "Dean–Teacher–Student" pipeline. In this pipeline, different LLM agents collaboratively produce multi-turn teaching dialogues: the Teacher agent generates Socratic guidance, the Student agent simulates responses, and the Dean agent revises the Teacher's instructions to ensure they adhere to the Socratic pedagogical style. Such mechanism enables complex tasks like pedagogical dialogue management.

Some ITS focus on a particular aspect of Socratic dialogue generation. For Socratic question generation (P1), Ang et al. [2023] fine-tune GPT-2 to generate type-specific Socratic questions (T3), such as probing assumptions or implications, for reflective and dialogue-driven tutoring. The prompt contains opinion information extracted from the Reddit database (D2). Qi et al. [2023] implement Socratic questioning through zero-shot prompting on ChatGPT, but they recursively decompose complex problems into sub-questions and synthesize answers through a backtracking process. Kumar et al. [2024] fine-tune Llama-2 using SFT and DPO to generate Socratic questions that guide students through debugging code incrementally without revealing solutions. SocraticLLM [Ding *et al.*, 2024] implements a pipeline, where Qwen is trained via SFT at the first stage and further optimized via DPO on their synthetic preference pairs, to generate Socratic-style questions.

Methods for providing scaffolded hinting (P2) have employed prompting methods. For instance, CoachGPT [Chen *et al.*, 2025] uses zero-shot prompting with structured constraints to generate step-by-step scaffolding-style guidance through the writing process. Similarly, DBox [Ma *et al.*, 2025] employs CoT-prompting on GPT-4o to deliver scaffolded hints for algorithmic programming at an appropriate level during coding.

Some ITS address both Socratic question generation and scaffolded hinting. StratL [Puech *et al.*, 2025] uses GPT-4o in combination with prompt engineering to generate Socratic questions (T3) and scaffolding hints (T2) to guide student exploration, while integrating affect factor based on student states predicted via a classifier (S2). PACE [Liu *et al.*, 2025a] builds upon LLaMA2 using a *tuning strategy* on a GPT-4-synthesized personalized dialogue dataset, and incorporates personality-aware prompts (S2) to guide Socratic-style questioning (T3) and hierarchical hint generation (T2) aligned with student learning styles.

Affective-aware interaction (P3) is integral to maintaining engagement. KEMI [Deng *et al.*, 2023] is fine-tuned on a LLM that jointly predicts emotion-related support strategies and generates responses in order to produce emotion-aware feedback (T5). It relies on an external mental health KG for embedding-based case retrieval (D2) and emotion-aware prompting. Apart from performing learner modeling (see Section 4.3), Liu et al. [2025b] fine-tune another LLM to deliver emotion-aware output (T5) to the students.

Dataset	Task	Annotation	Dataset Size
Math-related Datasets			
EduMath	I1,I2,I4	Q&A; optional target, solution step, explanation	∼10K Qs
GSM8K	I1-4 R1&2	Q, reasoning, final answer	8.5k Qs
Eedi	M2	Multiple-choice Q, error type	∼1.6M resps
MAE	M2	Misconception description, topic, diagnostic example	55 descriptions 220 examples
SocraTeach	P1	Dialogs, guiding questions, Q, analysis, teacher responses	∼57k Ds
SocraticMATH	P1	Socratic dialogs, difficulty, solutions, knowledge tags	∼6,846 Ds
Science-related Datasets			
ScienceQA	I1-I4 R1&2	Multiple-choice Q&A, rationale; optional image	∼21k Qs
Writing-related Datasets			
ASAP++	R1	Essays w/ holistic & trait scores	∼13k essays
ICLE++	R1	Essays w/ holistic & trait scores	∼6k essays
Datasets with Multiple Subjects			
EduAdapt	I1-I4 R1&2	Q&A, C, difficulty, optional knowledge tag	∼1–2M Ds
SAF	R2&3	Short A, comment, optional C or quality labels	∼2k instances
ASSISTments	M1,P2	Q, C, skill tag, timestamp	∼.4–.7M Ds
Junyi	M1	Q, C, timestamp, skill tag	∼72M Ds
EdNet	M1,P2	C, concept label, temporal log	∼100M Ds
SocraticQ	P1	Context, Socratic Q, Q type	∼110K Ds
ESConv	P3	Dialogs with support strategy, emotion context	∼1k–1.6K Ds
ESD-CoT	P3	Dialogs with emotion, stimuli, appraisal, strategy reason	∼1.7K+ Ds

Table 3: Commonly-used datasets used in ITS research. Descriptions of the task(s) covered by each dataset can be found in Table 2 (Q: Question; A: Answer; C: Correctness; D: Dialogue).

4.5 Summary of Discussion

We conclude this section by summarizing the datasets and methods we have discussed so far. Specifically, Table 3 categorizes the datasets commonly-used in ITS research based on its subject (i.e., whether a dataset is math- or science-related, for instance). For each dataset, we show (1) which of the task(s) in Table 2 it was created for, (2) the annotations it contains, and (3) its size. Table 4 presents a categorization of the methods used to address the tasks in Table 2. For each method, we show (1) which task(s) it addresses, (2) whether it adopts a modular or integrated architecture, and if a modular architecture is adopted, (3) which module(s) are implemented and which functions in Section 3.1 they focus on. However, no systematic comparison exists between the two architectures on the same task, remaining an open question.

5 Application Scenarios

While the above sections delineate the capabilities enabled by LLMs within ITS, these capabilities manifest differently across educational contexts. Next, we will discuss how such functionalities are adapted in diverse application scenarios.

IJCAI-ECAI 2026 PREPRINT

Work	Task	Modular			Integrated
		Domain	Student	Tutoring	
Instructional Content Synthesis					
CyberGen [2024]	I5	D1			
Wan et al. [2025]	I1	D1		T6	
RPG [2024]	I2			T5	
EQPR [2025]	I1	D2		T6	
Jia et al. [2025]	I3				✓
Wang et al. [2024b]	I4				✓
He et al. [2024]	I2				✓
Learner Response Evaluation					
ITS-LLM [2025]	R2	D2	S2	T1	
Sonkar et al. [2024]	R2	D1,D2		T1	
Zhu et al. [2025]	R1				✓
Chang et al. [2024]	R1				✓
T-MES [2025]	R1				✓
Havare et al. [2025]	R1				✓
NarGINA [2025]	R1				✓
Learner Modeling					
Wu et al. [2025]	M2	D1	S2		
Liu et al. [2025b]	M1,M2	D1	S1,S2	T5	
LLM-KT [2025]	M1	D2	S1		
KCD [2025]	M1,M2				✓
FineCD [2025]	M2				✓
LLMKT [2024]	M1				✓
Yin et al. [2023]	M1,M2				✓
Pedagogical Dialogue Management					
StratL [2025]	P1,P2		S2	T2,T3	
PACE [2025a]	P1,P2		S2	T2,T3	
Ang et al. [2023]	P1	D2		T3	
KEMI [2023]	P3	D2		T5	
Liu et al. [2025b]	P3	D1	S1,S2	T5	
CoachGPT [2025]	P2				✓
DBox [2025]	P2				✓
Qi et al. [2023]	P1				✓
SocraticLM [2024]	P1,P2				✓
Kumar et al. [2024]	P1				✓
SocraticLLM [2024]	P1,P2				✓

Table 4: Categorization of existing LLM-based ITS based on (1) which task(s) in Table 2 they address and (2) whether they adopt a modular or integrated system architecture. If a modular architecture is adopted, details on which module(s) are implemented and which functions described in Section 3.1 they focus on.

STEM Education. LLM-based ITS have shown significant promise in supporting science, technology, engineering and mathematics (STEM) education. The systems can interpret diverse student inputs, including equations, diagrams and natural language reasoning, and deliver hints or problem reformulations. Recent work demonstrates that LLMs can automatically grade in science-related subjects [Zhu *et al.*, 2025], generate step-by-step explanations for physics [Pang *et al.*, 2025], scaffold math teaching through Socratic questioning [Liu *et al.*, 2025a], and provide real-time feedback on student-written code [Havare *et al.*, 2025].

Language Learning. Language learning is a prominent application domain for LLM-based ITS. Recent studies [Gao *et al.*, 2025] and systems, including Duolingo[7] and Grammarly[8], highlight their capacity to simulate authentic conversational scenarios, enabling learners to refine pronunciation, fluency and pragmatic competence. Moreover, LLM-powered tutors can generate learning materials, such as reading passages or listening exercises [Huang *et al.*, 2025], and provide formative assessment by analyzing texts generated by learners for syntactic accuracy, lexical diversity and discourse coherence, offering actionable insights that mirror human instructor feedback [Rashkin *et al.*, 2025].

Professional Training. LLM-based ITS have significant potential in professional education due to their ability to simulate real-world tasks and provide personalized guidance, enhancing social skills by generating authentic scenarios and providing interactions, which is valuable for communication-intensive professions [Guevarra *et al.*, 2025]. This can extend to fields requiring high-level expertise, such as law, medicine and engineering. Systems can imitate complex situations, support case-based reasoning, and train professionals in critical analysis, diagnostic accuracy, and compliance with domain-specific protocols [Wang *et al.*, 2024b].

6 Concluding Remarks

We conclude our discussion with several research directions.

Pedagogical Alignment. While current LLM-based ITS have made progress in generating educational content, their pedagogical alignment remains narrow, usually limited to Socratic [Ang *et al.*, 2023] or scaffolding theory [Chen *et al.*, 2025]. A critical issue lies in achieving deeper, more principled alignment with a broader spectrum of instructional strategies, which could even dynamically switch based on reality. Particularly, in scenarios where human teachers remain central, LLM tutors must operate as coherent extensions of the instructor's intent. Thus, research should investigate mechanisms for *human-in-the-loop* alignment. Additionally, future work should establish multi-faceted evaluation protocols that assess not only response quality, but also the strategic appropriateness, theoretical consistency, and educational impact of the system's response.

Transparent Educational Reasoning. Instructional decisions of current systems are often generated in a black-box manner, lacking explicit justification grounded in observable data or educational theory. Future systems should produce transparent, traceable reasoning chains that explain not only what action is taken, but also when and why it is appropriate [Yang *et al.*, 2025]. This requires the reasoning process to explicitly refer to verifiable educational artifacts. For example, a LLM should justify its answer to a question by citing the dependencies among the related concepts in a curriculum-aligned KG. By anchoring LLM-generated teaching actions to auditable components such as curriculum standards and historical interaction logs, the system can output human-interpretable rationales, so that the decisions can be validated.

Multi-modal Interaction Support. Existing ITS are predominantly text-centric, creating a gap in processing and responding in multi-modal scenarios. While proof-of-concept systems highlight the promise of multi-modal interaction support [Lu *et al.*, 2022], they usually focus on multi-modal interactions a in certain specific stage, instead of a closed-loop educational system. Future ITS should seek deeper integration, such as real-time interpretation of affective states from facial expressions or speech prosody for the *Student Model*, as well as pedagogy-driven generation of coordinated verbal and visual guidance by the *Tutoring Model*.

[7]https://www.duolingo.com
[8]https://www.grammarly.com

IJCAI-ECAI 2026 PREPRINT

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.62306145), Frontier Technologies R&D Program of Jiangsu (No.BF2024076); the Adolescent Education and Intelligence Support Laboratory of Nanjing Normal University; the Laboratory of Philosophy and Social Sciences at Universities in Jiangsu Province.

References

[Agrawal *et al.*, 2024] Garima Agrawal, Kuntal Pal, Yuli Deng, et al. CyberQ: Generating questions and answers for cybersecurity education using knowledge graph-augmented llms. In *AAAI*, 2024.

[Ang *et al.*, 2023] Beng Ang, Sujatha Gollapalli, and See-Kiong Ng. Socratic question generation: A novel dataset, models, and evaluation. In *EACL*, 2023.

[Chang and Ginter, 2024] Li-Hsin Chang and Filip Ginter. Automatic short answer grading for finnish with chatgpt. In *AAAI*, 2024.

[Chen *et al.*, 2024] Yulin Chen, Ning Ding, Hai-Tao Zheng, et al. Empowering private tutoring by chaining large language models. In *CIKM*, 2024.

[Chen *et al.*, 2025] Fumian Chen, Sotheara Veng, Joshua Wilson, et al. CoachGPT: A scaffolding-based academic writing assistant. In *SIGIR*, 2025.

[Cheng *et al.*, 2025] Cheng Cheng, Zhenya Huang, Guan-Hao Zhao, et al. From objectives to questions: A planning-based framework for educational mathematical question generation. In *ACL*, 2025.

[Choi *et al.*, 2020] Youngduck Choi, Youngnam Lee, Dongmin Shin, et al. EdNet: A large-scale hierarchical dataset in education. In *AIED*, 2020.

[Chu *et al.*, 2025] Zhendong Chu, Shen Wang, Jian Xie, et al. LLM agents for education: Advances and applications. *CoRR*, 2025.

[Cobbe *et al.*, 2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *CoRR*, 2021.

[Conati, 2009] Cristina Conati. Intelligent tutoring systems: New challenges and directions. In *IJCAI*, 2009.

[Deng *et al.*, 2023] Yang Deng, Wenxuan Zhang, Yifei Yuan, et al. Knowledge-enhanced mixed-initiative dialogue system for emotional support conversations. In *ACL*, 2023.

[Ding *et al.*, 2024] Yuyang Ding, Hanglei Hu, Jie Zhou, et al. Boosting large language models with socratic method for conversational mathematics teaching. In *CIKM*, 2024.

[Dong *et al.*, 2025] Zhiang Dong, Jingyuan Chen, and Fei Wu. Knowledge is power: Harnessing large language models for enhanced cognitive diagnosis. In *AAAI*, 2025.

[Filighera *et al.*, 2022] Anna Filighera, Siddharth Singh Parihar, Tim Steuer, et al. Your answer is incorrect... would you like to know why? introducing a bilingual short answer feedback dataset. In *ACL*, 2022.

[Gaeta *et al.*, 2025] Angelo Gaeta, Francesco Orciuoli, Antonella Pascuzzo, et al. Enhancing traditional its architectures with large language models for generating motivational feedback. *Computers and Education: Artificial Intelligence*, 9, 2025.

[Gao *et al.*, 2025] Rena Gao, Jingxuan Wu, Xuetong Wu, et al. An interpretable and crosslingual method for evaluating second-language dialogues. In *NAACL*, 2025.

[Guevarra *et al.*, 2025] Michael Guevarra, Indronil Bhattacharjee, Srijita Das, et al. An LLM-guided tutoring system for social skills training. In *AAAI*, 2025.

[Havare *et al.*, 2025] Jayant Havare, Varsha Apte, Kaushikraj Maharajan, et al. Ai-based automated grading of source code of introductory programming assignments. In *ICPC*, 2025.

[He *et al.*, 2024] Yuhang He, Jianzhu Bao, Yang Sun, et al. Decomposing argumentative essay generation via dialectical planning of complex reasoning. In *Findings of ACL*, 2024.

[Hricko, 2008] Mary Hricko. Gagne's nine events of instruction. In *Encyclopedia of Information Technology Curriculum Integration*. IGI Global, 2008.

[Hu *et al.*, 2021] Edward Hu, Yelong Shen, Phillip Wallis, et al. Lora: Low-rank adaptation of large language models, 2021.

[Huang *et al.*, 2025] Minjiang Huang, Jipeng Qiang, Yi Zhu, et al. AI4Reading: Chinese audiobook interpretation system based on multi-agent collaboration. In *ACL Demo*, 2025.

[Jia *et al.*, 2025] Linzhao Jia, Changyong Qi, Yuang Wei, et al. Fine-tuning large language models for educational support: Leveraging gagné's nine events of instruction for lesson planning. *CoRR*, 2025.

[Kumar and Lan, 2024] Nischal Kumar and Andrew Lan. Improving socratic question generation using data augmentation and preference optimization. In *BEA*, 2024.

[Lewis *et al.*, 2020] Patrick Lewis, Ethan Perez, Aleksandra Piktus, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.

[Li and Ng, 2024] Shengjie Li and Vincent Ng. ICLE++: Modeling fine-grained traits for holistic essay scoring. In *NAACL*, 2024.

[Li *et al.*, 2025] Mingjia Li, Hong Qian, Jinglan Lv, et al. Foundation model enhanced derivative-free cognitive diagnosis. *Front. Comput. Sci.*, 19(1), 2025.

[Liu *et al.*, 2021] Siyang Liu, Chujie Zheng, Orianna Demasi, et al. Towards emotional support dialog systems. In *ACL*, 2021.

[Liu *et al.*, 2024] Jiayu Liu, Zhenya Huang, Tong Xiao, et al. SocraticLM: Exploring socratic personalized teaching with large language models. In *NeurIPS*, 2024.

[Liu *et al.*, 2025a] Ben Liu, Jihai Zhang, Fangquan Lin, et al. *One Size doesn't Fit All*: A personalized conversational tutoring agent for mathematics instruction. In *WWW (Companion Volume)*, 2025.

IJCAI-ECAI 2026 PREPRINT

[Liu *et al.*, 2025b] Qi Liu, Yunhao Sha, Kai Zhang, et al. Advancements in digital humans for recorded courses: Enhancing learning experiences via personalized interaction. *Front. Digit. Educ.*, 2(4), 2025.

[Lu *et al.*, 2022] Pan Lu, Swaroop Mishra, Tony Xia, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *NeurIPS*, 2022.

[Lyu *et al.*, 2024] Yuanjie Lyu, Zihan Niu, Zheyong Xie, et al. Retrieve-plan-generation: An iterative planning and answering framework for knowledge-intensive llm generation. *CoRR*, 2024.

[Ma *et al.*, 2025] Shuai Ma, Junling Wang, Yuanhao Zhang, et al. Dbox: Scaffolding algorithmic programming learning through learner-llm co-decomposition. In *CHI*, 2025.

[Mathias and Bhattacharyya, 2018] Sandeep Mathias and Pushpak Bhattacharyya. ASAP++: enriching the ASAP automated essay grading dataset with essay attribute scores. In *LREC*, 2018.

[Naeem *et al.*, 2025] Numaan Naeem, Abdellah El Mekki, and Muhammad Abdul-Mageed. EduAdapt: A question answer benchmark dataset for evaluating grade-level adaptability in llms. In *EMNLP*, 2025.

[Nye *et al.*, 2014] Benjamin Nye, Arthur Graesser, and Xiangen Hu. Autotutor and family: A review of 17 years of natural language tutoring. *INT J ARTIF INTELL E*, 24, 2014.

[Pang *et al.*, 2025] Xinyu Pang, Ruixin Hong, Zhanke Zhou, et al. Physics reasoner: Knowledge-augmented reasoning for solving physics problems with large language models. In *COLING*, 2025.

[Puech *et al.*, 2025] Romain Puech, Jakub Macina, Julia Chatain, et al. Towards the pedagogical steering of large language models for tutoring: A case study with modeling productive failure. In *Findings of ACL*, 2025.

[Qi *et al.*, 2023] Jingyuan Qi, Zhiyang Xu, Ying Shen, et al. The art of SOCRATIC QUESTIONING: recursive thinking with large language models. In *EMNLP*, 2023.

[Rashkin *et al.*, 2025] Hannah Rashkin, Elizabeth Clark, Fantine Huot, et al. Help me write a story: Evaluating llms' ability to generate writing feedback. In *ACL*, 2025.

[Rus *et al.*, 2010] Vasile Rus, Mihai Lintean, and Roger Azevedo. Computational aspects of the intelligent tutoring system metatutor. In *FLAIRS*, 2010.

[Scarlatos *et al.*, 2024] Alexander Scarlatos, Ryan Baker, Andrew Lan, et al. Exploring knowledge tracing in tutor-student dialogues using llms. In *LAK*, 2024.

[Shridhar *et al.*, 2022] Kumar Shridhar, Jakub Macina, Mennatallah El-Assady, et al. Automatic generation of socratic subquestions for teaching math word problems. In *EMNLP*, 2022.

[Singh and Singh, 2022] Sanjay Singh and Vikram Singh. An architecture of domain independent and extensible intelligent tutoring system based on concept dependencies and subject paths. *Int J Adv Comput Sci Appl*, 13(5), 2022.

[Sleeman and Brown, 1982] Derek Sleeman and John Brown. *Intelligent Tutoring Systems: An Overview*. Academic Press, 1982.

[Sonkar *et al.*, 2024] Shashank Sonkar, Kangqi Ni, Sapana Chaudhary, et al. Pedagogical alignment of large language models. In *Findings of EMNLP*, 2024.

[Tabarsi, 2025] Benyamin Tabarsi. Developing llm-powered trustworthy agents for personalized learning support. In *AAAI*, 2025.

[Wan *et al.*, 2025] Qian Wan, Wangzi Shi, Jintian Feng, et al. Empowering math problem generation and reasoning for large language model via synthetic data based continual learning framework. In *EMNLP*, 2025.

[Wang and Liu, 2025] Jiong Wang and Jie Liu. T-MES: trait-aware mix-of-experts representation learning for multi-trait essay scoring. In *COLING*, 2025.

[Wang *et al.*, 2024a] Allison Wang, Ethan Prihar, Aaron Haim, et al. Can large language models generate middle school mathematics explanations better than human teachers? In *AIED*, 2024.

[Wang *et al.*, 2024b] Ruiyi Wang, Stephanie Milani, Jamie Chiu, et al. Patient-ψ: Using large language models to simulate patients for training mental health professionals. In *EMNLP*, 2024.

[Wang *et al.*, 2024c] Shen Wang, Tianlong Xu, Hang Li, et al. Large language models for education: A survey and outlook. *CoRR*, 2024.

[Wang *et al.*, 2025] Ziwei Wang, Jie Zhou, Qin Chen, et al. LLM-KT: aligning large language models with knowledge tracing using a plug-and-play instruction. *CoRR*, 2025.

[Wu *et al.*, 2025] Tao Wu, Jingyuan Chen, Wang Lin, et al. Embracing imperfection: Simulating students with diverse cognitive levels using llm-based agents. In *ACL*, 2025.

[Yang *et al.*, 2025] Qihao Yang, Yu Yang, Sixu An, et al. Llm-based collaborative agents with pedagogy-guided interaction modeling for timely instructive feedback generation in task-oriented group discussions. In *IJCAI*, 2025.

[Yin *et al.*, 2023] Yu Yin, Le Dai, Zhenya Huang, et al. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *WWW*, 2023.

[Zhang *et al.*, 2024] Tenggan Zhang, Xinjie Zhang, Jinming Zhao, et al. ESCoT: Towards interpretable emotional support dialogue systems. In *ACL*, 2024.

[Zhong *et al.*, 2025] Jun Zhong, Longwei Xu, Li Kong, et al. Nargina: Towards accurate and interpretable children's narrative ability assessment via narrative graphs. In *Fingdings of ACL*, 2025.

[Zhu *et al.*, 2025] Haohao Zhu, Tingting Li, Peng He, et al. Enhancing automated grading in science education through llm-driven causal reasoning and multimodal analysis. In *IJCAI*, 2025.

IJCAI-ECAI 2026 PREPRINT