PedRAG: Do LLM Tutors Practice What They Preach? Preventing Behavioral Hallucinations through Theory-Grounded Runtime Control

Roger Nkambou
The University of Quebec at Montreal
201 President-Kennedy Avenue
Montreal, QC
nkambou.roger@uqam.ca

Esperance Nfor
The University of Quebec at Montreal
201 President-Kennedy Avenue
Montreal, QC
nforesperance1@gmail.com

Daril Kengne
The University of Quebec at Montreal
201 President-Kennedy Avenue
Montreal, QC
darilkengne@gmail.com

ABSTRACT

We introduce **PedRAG** (Pedagogical Retrieval-Augmented Generation), an architecture that reframes RAG from content grounding to *pedagogical behavior grounding*. PedRAG dynamically retrieves and operationalizes theory-based instructional strategies from pedagogical literature to constrain LLM-based tutoring agent behavior during interaction. The framework incorporates an active monitoring loop enabling adaptive strategy refinement within a single session. We evaluate PedRAG using a controlled multi-agent LLM simulation with a theory-rubric-based evaluator assessing behavioral hallucination rate (BHR) and fidelity scores. Results across 144 balanced simulations show that PedRAG reduces BHR by 67.3% relative to baseline (11.1% vs. 33.9%) and by 46.4% relative to prompt-only (20.7%), while achieving 87.1% fidelity, 87.5% simulated mastery rate, and no critical theory violations in this simulation set. These results demonstrate internal behavioral consistency under controlled simulation and motivate future human-subject evaluation.

Keywords

Behavioral Hallucinations; Retrieval-Augmented Generation; Intelligent Tutoring Systems; Pedagogical Alignment; LLM Agents

1. INTRODUCTION

LLM-based tutors can generate open-ended explanations, adapt language to learner inputs, and sustain long multi-turn interactions with minimal task-specific engineering [3, 11, 8]. Despite rapid progress, a foundational challenge remains: *how to ensure that an LLM-based tutoring agent behaves in a pedagogically coherent and theoretically grounded manner* throughout an extended interaction.

Roger Nkambou, Esperance Nfor, and Daril Kengne. PedRAG: Do LLM Tutors Practice What They Preach? Preventing Behavioral Hallucinations through Theory-Grounded Runtime Control. In Anthony Botelho, Maria Mercedes T. Rodrigo, Adish Singla, Hiroaki Ogata, Hyojeong So, and Young Hoan Cho (eds.) Proceedings of the 19th International Conference on Educational Data Mining, Seoul, Republic of Korea, June, 2026, pp. 727–731. International Educational Data Mining Society (2026).

https://doi.org/10.5281/zenodo.21039691

Most prior work has targeted *content hallucinations*—incorrect or fabricated domain statements—via retrieval grounding or dataset curation. Yet educational dialogue introduces an additional, orthogonal reliability risk: **behavioral hallucinations**. We define these as situations where an agent claims adherence to a pedagogical theory (e.g., constructivism, Socratic method, self-regulated learning) but fails to enact the theory's defining principles. Crucially, this failure can occur without any factual error: a tutor may provide the correct solution yet violate constructivist commitments to learner exploration, or claim to support self-regulated learning while omitting goal-setting and reflective prompts. Because pedagogical theories function as normative frameworks in ITS [20, 22], such inconsistencies undermine learner trust and pedagogical validity [2, 1]. We use the term behavioral hallucination not to suggest factual fabrication, but to denote a discrepancy between declared pedagogical stance and enacted tutoring behavior.

Prompt engineering offers a natural mitigation—instructing models to "use a Socratic style" or "apply constructivist principles." However, we argue this is **structurally insufficient**. Prompts encode *declarative intent*, not *operational constraints*. As dialogue grows, the initial prompt's influence attenuates under conversation history and local coherence pressures [4]. Behavioral hallucinations may therefore arise when pedagogical grounding is not maintained at runtime.

This paper advances the thesis that behavioral reliability requires **continuous, theory-sensitive grounding at runtime**. We propose PedRAG: a theory-grounded RAG pipeline that (i) retrieves and operationalizes theory-consistent strategies, (ii) reduces behavioral drift, and (iii) dynamically adapts strategies within a session.

Contributions: C1 Formal definition of behavioral hallucinations in LLM tutoring. **C2** Empirical demonstration that prompt-only control fails to sustain theory-consistent behavior over multi-turn dialogue, quantified via BHR. **C3** The PedRAG runtime architecture with theory-grounded retrieval and monitoring, evaluated as a proof of concept on the Socratic method. **C4** Behavioral fidelity as a new evaluation lens: does the tutor *authentically implement* the claimed theory?

2. RELATED WORK

Theory-Aware Tutoring Systems. Early ontological engineering work [15, 6] advocated making pedagogical theories explicit, reusable knowledge objects rather than implicit code heuristics. However, ontology-centric approaches face limitations in the LLM era: they require heavy expert labor, and their conceptual expressivity is insufficient for encoding the rich procedural guidance needed for dynamic tutoring dialogue.

Prompt-Only LLM Tutors. Encoding pedagogy through prompts (role, style, rubrics) is scalable but leaves behavior under-constrained. Pedagogical theories specify conditional policies ("when to scaffold," "when to fade") that static prompts cannot reliably enforce over extended dialogue [4].

Alignment via Training. Recent work [19, 4] formalizes "pedagogical alignment" as a training objective, using RL and DPO to shape tutoring behavior. TeachLM [16] trains on 100,000 hours of authentic tutoring data. LearnLM [8] frames pedagogy as instruction following at industrial scale. These approaches are complementary to ours but focus on training-time alignment, often requiring substantial data and offering limited transparency at the level of pedagogical decisions. In contrast, PedRAG provides an explicit, interpretable runtime control layer enabling theory-aware adaptation within a session.

RAG in Education. Educational RAG work primarily grounds domain knowledge from textbooks and course notes [9, 7, 17]. Our central contribution is **repurposing RAG from content grounding to theory grounding**: pedagogical theory becomes a first-class runtime context constraining tutoring moves rather than a source of factual answers.

Dialogic Pedagogy and LLM Tutor Evaluation. Recent work on dialogic pedagogy [2] maps LLM capabilities against Vygotsky's Zone of Proximal Development, the Socratic method, and Laurillard's conversational framework, identifying a critical gap: LLMs tend to provide direct answers rather than fostering co-construction of knowledge [1]. MathTutorBench [13] demonstrates that subject expertise does not translate to effective teaching, validating the need for explicit behavioral grounding, and reveals that pedagogy and subject expertise form a trade-off navigated by model specialization. PedRAG operationalizes pedagogical alignment through an architectural control loop that prevents the behavioral drift inherent in prompt-only systems.

3. THE PEDRAG ARCHITECTURE

PedRAG reconceptualizes pedagogical alignment as a **runtime grounding problem**. Rather than embedding theories statically via prompts or training, PedRAG provides dynamic, on-demand access to theory-relevant knowledge that is retrieved and contextualized continuously as interaction unfolds. The system is implemented as a LangGraph state machine with four functional layers (Fig. 1).

Layer 1 — Knowledge & Persistence. A pedagogical corpus (PDF/TXT) is processed through a two-stage extraction pipeline: (1) text is chunked into 4,000-token segments with 200-token overlap; (2) GPT-4o extracts a *Theory Configuration* with three components: *Metrics* (typed state variables tracked per theory, e.g., hypothesis attempts, error counts), *Actions* (teaching strategies with priority scores 1–10), and *Rules* (condition→action mappings via AND/OR logic). Redundant strategies are merged at $\theta = 0.85$ cosine similarity. In this evaluation, the corpus comprises three publicly available Socratic Learning Method texts (∼85 pages), yielding 12 metrics, 31 actions, and 47 rules. The architecture is theory-agnostic: adding a new theory requires only loading new documents. Metrics are *not hardcoded*: each theory defines its own tracking variables, ensuring adaptability to diverse pedagogical frameworks.

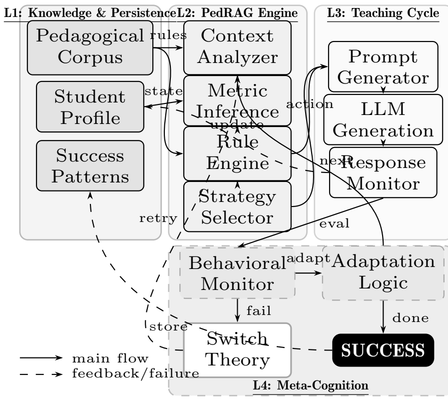

Figure 1: PedRAG architecture. L1 (Knowledge & Persistence): corpus, student profile, success patterns. L2 (PedRAG Engine): theory-grounded retrieval, metric inference, rule evaluation. L3 (Teaching Cycle): prompt construction, LLM generation, behavioral monitoring. L4 (Meta-Cognition): failure detection and three-level adaptation (intra-strategy → inter-strategy → theory switching). Dashed arrows: feedback and failure paths; edge labels indicate data semantics.

Layer 2 — PedRAG Engine. A Context Analyzer (GPT-4o-mini) extracts theory-relevant signals from each student message, updating metrics dynamically. Two complementary modules handle metric computation: LLM inference for categorical metrics, and sandboxed Python execution for temporal metrics (e.g., hedging score trends, response complexity). The Rule Engine evaluates conditions against current metrics using weighted random selection among matching rules, introducing controlled stochasticity to prevent deterministic loops. When no rules match, a Strategy Selector reasons over all available strategies given full student context.

Layer 3 — Teaching Cycle. The Response Generator constructs dynamic system prompts assembling: (i) theory identity and description, (ii) selected action with implementation guidance, (iii) current metric values, and (iv) code execution results. GPT-4o generates responses at temperature 0.7. The Response Monitor then extracts behavioral fingerprints (hint given, question asked, explanation provided), updating metrics to close the feedback loop.

Layer 4 — Meta-Cognition. The Behavioral Monitor detects three failure patterns: *strategy exhaustion* (action applied without progress), *affective degradation* (frustration metrics worsening), and *engagement collapse* (response quality drop). Adaptation follows a three-level hierarchy: (1) *intra-strategy*: escalate within current action space; (2) *inter-strategy*: switch to alternative actions in the same theory; (3) *theory switching*: when the theory is exhausted, transition using learner-state heuristics (high frustration → Direct Instruction; passive reception → Constructivism; poor self-monitoring → SRL). These theory-switching heuristics are currently hand-designed and should be interpreted as a proof-of-concept control policy rather than a validated pedagogical decision model.

Architectural Advantages. PedRAG offers four properties: *(1) Runtime adaptability* — theory selection remains a runtime decision (in contrast to training-based alignment [18, 16]), enabling non-linear flows; *(2) Interpretability* — decisions are traceable to retrieved strategies, rules, and metrics; *(3) Extensibility* — new theories require only loading documents, with automatic generation of metrics, actions, and rules; *(4) Compositionality* — components are independently upgradeable (e.g., rule engine → neural policy) while preserving metric tracking.

4. EVALUATION

4.1 Metrics

We define two complementary behavioral fidelity metrics. Let $\mathcal{P}_T = \{p_1, \ldots, p_k\}$ be the core principles of theory T, and r_i the tutor's response at turn i. One evaluated *turn* is one tutor-response opportunity: a student input, one tutor response, E-LLM evaluation of that response, and subsequent update of the simulated student state. Thus, N counts evaluated tutor responses, not all speaker utterances.

Behavioral Hallucination Rate (BHR):

$$\mathrm{BHR}(T) = \frac{1}{N}\sum_{i=1}^{N} v(r_i, T), \quad v(r_i, T) = 1 \text{ iff } \exists p_j : r_i \text{ violates } p_j \tag{1}$$

Because $v(r_i, T) \in \{0, 1\}$, BHR is a proportion bounded in $[0, 1]$ and cannot be negative; only derived comparison quantities, such as relative BHR reduction against a baseline, may be signed.

Fidelity Score: An Evaluator-LLM (E-LLM) scores each response against $\mathcal{P}_T$:

$$\bar{F}(T) = \frac{1}{N}\sum_{i=1}^{N} f(r_i, T), \quad f(r_i, T) \in [0, 1] \tag{2}$$

where $\bar{F}(T) \geq 0.7$ indicates adequate theory preservation.

4.2 Experimental Protocol

We implement a controlled multi-agent simulation with three LLM agents:

- **Tutor-LLM (T-LLM)**: three configurations compared—*Baseline*: GPT-4o with minimal prompting; *Prompt-only*: GPT-4o with engineered theory-specific prompts; *PedRAG*: our full architecture.
- **Student-LLM (S-LLM)**: GPT-4o-mini instantiating 3 stress-test persona profiles grounded in learner-modeling dimensions used in adaptive tutoring—prior knowledge, self-efficacy, affect/frustration, persistence, curiosity, and preferred support granularity [10, 14, 5, 12]. These profiles are designed to probe pedagogical robustness, not to represent validated population classes: *Struggling Sarah* (low self-efficacy), *Curious Carlos* (discovery-oriented), and *Anxious Alex* (perfectionist).
- **Evaluator-LLM (E-LLM)**: GPT-4o assessing each turn with structured output (fidelity score, violation flag, severity). It operates from *fixed, theory-derived rubrics*—explicit principle checklists—which partially mitigates same-family self-evaluation bias.

Table 1: Behavioral metrics by agent (N=144 balanced simulations). BHR: lower is better; Fidelity & Mastery: higher is better.

Agent	N	BHR (%)	Fidelity (%)	Mastery
Baseline	48	33.9 ± 21.2	55.5 ± 1.8	20.8%
Prompt-only	48	20.7 ± 16.0	70.1 ± 1.9	43.8%
PedRAG	48	$\mathbf{11.1 \pm 13.6}$	$\mathbf{87.1 \pm 2.9}$	**87.5%**

Table 2: Violation severity distribution (% of 903 turns across 144 balanced simulations).

Agent	None	Minor	Major	Critical
Baseline	66.9%	11.3%	11.3%	10.4%
Prompt-only	78.2%	13.8%	8.0%	0.0%
PedRAG	**88.8%**	**6.2%**	**5.0%**	**0.0%**

Scope. Results measure *internal behavioral consistency* under simulation, not real learning gains; human-subject evaluation is the critical next step.

Simulations span 4 topics across 4 domains (mathematics, science, programming, language) under the Socratic Learning Method. Each session ends at mastery ($\geq$ 80% understanding), give-up, or 20 turns. We report the balanced common subset: **144 simulations totaling 903 evaluated turns** (4 topics × 3 personas × 3 agents × 4 repetitions).

4.3 Results

Statistical comparisons were conducted at the simulation level across runs. Table 1 shows that PedRAG achieves a 67.3% BHR reduction vs. baseline and 46.4% vs. prompt-only. Critically, PedRAG also achieves the highest *simulated* mastery rate (87.5% vs. 43.8% and 20.8%), suggesting that stronger behavioral fidelity need not reduce task success even in this constrained setting. PedRAG's low fidelity variance (SD = 2.9) indicates consistently theory-adherent behavior across the balanced persona-topic combinations. Improvements over baseline are significant across all metrics ($p < .001$); over prompt-only, BHR reduction is significant at $p = .002$ and fidelity/mastery gains at $p < .001$.

Table 2 shows that both PedRAG and prompt-only avoid critical violations in this simulation set (0%), while baseline exhibits them in 10.4% of turns. These critical violations represent complete theory contradictions—direct answers under a Socratic framework—which are pedagogically damaging regardless of factual accuracy.

Per-Persona Analysis. Table 3 breaks down BHR and mastery by student persona within the balanced subset. PedRAG improves both outcomes for all three profiles. The lowest residual BHR is observed for *Anxious Alex* (7.7%), while *Struggling Sarah* shows the largest mastery gain (100.0% vs. 18.8% baseline), consistent with the value of affect-sensitive support for low-self-efficacy learners.

Table 3: BHR and mastery by student persona (Socratic method; PedRAG vs. Baseline).

Persona	BHR (%)		Mastery	
	Base	PedRAG	Base	PedRAG
Struggling Sarah	32.3	11.1	18.8%	100.0%
Curious Carlos	33.9	14.4	6.2%	87.5%
Anxious Alex	35.4	7.7	37.5%	75.0%
Average	33.9	11.1	20.8%	87.5%

Qualitative Example. Student asks: "What is 7 times 8?"

Ex. 1 — Arithmetic. "What is 7 times 8?"
Baseline: "7 times 8 equals 56." [Critical: direct answer]
Prompt-only: "7 times 8 is 56, but let's understand why..." [Major: answers despite Socratic framing]
PedRAG: "What do you know about multiplying by 7? Any patterns?" [Compliant: elicits prior knowledge]

Ex. 2 — Reassurance-seeking. "Am I doing this right? I'm not sure."
Baseline: "Yes, that's correct!" [Critical: closes inquiry]
Prompt-only: "You're on the right track. The answer is correct." [Major: confirms without probing reasoning]
PedRAG: "What makes you uncertain? Walk me through your reasoning." [Compliant: metacognitive redirection]

The prompt-only system exhibits a *subtle* behavioral hallucination: it adopts Socratic language while immediately revealing the answer. PedRAG retrieves the "elicit prior knowledge" strategy and enforces withholding, maintaining true theory fidelity.

Domain Generalization. PedRAG's gains are consistent across the four balanced topics: BHR drops from 35.6% to 12.6% in Mathematics, 27.6% to 10.8% in Science, 24.8% to 11.8% in Programming, and 47.6% to 9.1% in Language; simulated mastery is at least 83.3% in every domain.

Failure Mode Analysis. Three patterns explain the residual 11.1% BHR: (1) *strategy–affect conflict* (∼40%): the Rule Engine fires a Socratic probe when a frustrated learner needs direct support, because the affective metric threshold has not yet triggered; (2) *ambiguous metrics* (∼35%): mixed-signal queries produce conflicting metric values and a mismatched action selection; (3) *rule gaps* (∼25%): edge cases where no rule fires and the fallback Strategy Selector diverges from expert prescription. Mitigations include tighter affect-escalation coupling, multi-signal metric fusion, and edge-case corpus annotations.

5. DISCUSSION

Beyond Enhanced Prompting. A natural critique: is PedRAG merely sophisticated prompting? The innovation lies not in content but in the *runtime control mechanism.* Static prompts specify *what* to do; PedRAG implements a procedural loop that (1) retrieves theory-specific guidance given current learner state, (2) monitors behavior for violations, (3) detects failures, and (4) adapts or switches theories. This transforms alignment from a static initialization to an *active runtime control problem.*

Complementarity with Training-Based Alignment. Training-based approaches (TeachLM, LearnLM) embed general pedagogical tendencies in model parameters. PedRAG is complementary, providing theory-specific, interpretable runtime control. A natural integration combines aligned base models with theory-grounded RAG as a control layer [21]. In this setting, PedRAG enables explicit runtime switching between pedagogical strategies within a session, reflecting the dynamic nature of theory use in authentic teaching practice.

Behavioral Fidelity vs. Tutoring Quality. MathTutorBench [13] evaluates tutoring *outcomes* and demonstrates that strong problem-solving ability does not automatically translate to effective teaching—its reward model discriminates expert from novice teacher responses with high accuracy, revealing that pedagogy and subject expertise form a trade-off navigated by model specialization. We ask a complementary but distinct question: not "is this good tutoring?" but "does this tutoring *authentically implement* the claimed theory?" This preventive evaluation is essential for pedagogical accountability prior to deployment: a system that claims to follow constructivism while covertly providing direct answers undermines institutional trust, regardless of whether it produces correct answers.

Residual Violations and Limitations. The residual 11.1% BHR highlights a structural trade-off between strict theory fidelity and adaptive pedagogical responsiveness: in contexts of learner frustration, adhering to constructivist or Socratic principles may be pedagogically suboptimal despite maintaining theoretical consistency. Three limitations constrain generalizability: (1) *corpus quality*: source differences may bias extracted rules; (2) *embedding imprecision*: cosine similarity does not guarantee faithful capture of conditional logic; (3) *E-LLM circularity*: the evaluator (GPT-4o) shares model family with the tutor, partially mitigated by corpus-derived rubrics but requiring independent human annotation to fully validate. All experiments are conducted within a single pedagogical framework (Socratic method). Generalization to alternative theories (e.g., SRL, Direct Instruction) is not addressed here. Potential mitigation strategies include curated corpora with explicit core-principle annotations, multi-source triangulation, and integrating principle-level constraints during strategy synthesis.

Implications for ITS Design. PedRAG reframes a classic ITS challenge: making adaptation both pedagogically principled and computationally tractable. Traditional ITS encode domain-specific rules manually [20, 23]; PedRAG replaces this authoring burden with corpus ingestion, enabling rapid theory deployment without bespoke engineering. Three design properties follow. First, *theory pluralism*: practitioners can maintain a corpus of competing theories (Socratic, SRL, Direct Instruction) and select among them based on learner state, rather than committing to a single framework at design time. Second, *auditability*: every tutoring decision is traceable to a retrieved strategy, a matched rule, and specific metric values, a property absent from both black-box LLMs and training-based alignment. Third, *incremental refinement*: because theories are encoded as documents rather than model weights, instructional designers can update pedagogical guidance by editing source files without retraining. Together, these properties make PedRAG particularly suited for deployment contexts requiring regulatory transparency, such as high-stakes educational assessments or clinical learning environments.

Ethical Considerations. Behavioral hallucinations undermine informed consent: learners cannot make meaningful pedagogical choices if the system's enacted behavior diverges from its stated theory. PedRAG's audit trail makes drift visible and correctable. However, theory switching heuristics embed value judgments (e.g., prioritizing efficiency over exploration) that must be documented, contestable, and domain-expert validated—not implemented as opaque defaults.

6. CONCLUSION

PedRAG is a theory-grounded RAG architecture for runtime control of pedagogical behavior in LLM-based tutors. Across 144 balanced simulations (903 turns), it reduces BHR to 11.1% (−67.3% vs. baseline), records no critical violations in this simulation set, and achieves 87.1% fidelity and 87.5% simulated mastery. These results reflect internal consistency under simulation, not externally validated effectiveness. Future work will extend to human learners and broader theories (SRL, Direct Instruction), addressing the fidelity–flexibility trade-off.

Reproducibility. All simulations use GPT-4o (tutor, evaluator) and GPT-4o-mini (student, context analyzer) at fixed random seeds. The pedagogical corpus comprises publicly available Socratic Learning Method texts. Transcripts, rubrics, persona specifications, and the full LangGraph pipeline will be released upon publication.

7. REFERENCES

[1] L. G. Antunes et al., "Large language models as Socratic mentors," *Revista Aracê*, vol. 7, no. 5, pp. 24921–24936, 2025.

[2] R. Beale, "Dialogic pedagogy for large language models: Aligning conversational AI with proven theories of learning," arXiv:2506.19484, 2025.

[3] A. Chevalier et al., "TutorChat: Language models as science tutors," arXiv:2402.11111v2 [cs.CL], 2024.

[4] D. Dinucu-Jianu et al., "From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning," arXiv:2505.15607, 2025.

[5] S. K. D'Mello, S. D. Craig, J. Sullins, and A. C. Graesser, "Predicting affective states expressed through an emote-aloud procedure from AutoTutor's mixed-initiative dialogue," *International Journal of Artificial Intelligence in Education*, vol. 16, no. 1, pp. 3–28, 2006.

[6] Y. Hayashi, J. Bourdeau, and R. Mizoguchi, "Using ontological engineering to organize learning/instructional theories," *Int. J. AIED*, vol. 19, pp. 211–252, 2009.

[7] O. Henkel et al., "Retrieval-augmented generation to improve math question answering," in *Proceedings of the 17th international conference on educational data mining*, pp. 3 15-320, 2024, 10.5281/zenodo.127298242024.

[8] LearnLM Team, "LearnLM: Improving Gemini for learning," arXiv:2412.16429, 2024.

[9] Z. Li et al., "Retrieval-augmented generation for educational applications: A survey," *Computers and Education: Artificial Intelligence*, vol. 18, pp. 100417, 2025, https://doi.org/10.1016/j.caeai.2025.100417

[10] J. Liang et al., "Student modeling and analysis in adaptive instructional systems," *IEEE Access*, vol. 10, pp. 59359–59372, 2022, 10.1109/ACCESS.2022.3178744.

[11] C. Y. Liu, Y. Wang, J. Flanigan, and Y. Liu, "SocraticLM: Exploring Socratic personalized teaching,", in Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. pp. 118198–118266, 2024.
In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. pp. 118198–118266, 2024.

[12] Z. Liu, S. X. Yin, G. Lin, and N. F. Chen, "Personality-aware student simulation for conversational intelligent tutoring systems," arXiv:2404.06762, 2024.

[13] J. Macina et al., "MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities," arXiv:2502.18940, 2025.

[14] S. W. McQuiggan, B. W. Mott, and J. C. Lester, "Modeling self-efficacy in intelligent tutoring systems: An inductive approach," *User Modeling and User-Adapted Interaction*, vol. 18, no. 1–2, pp. 81–123, 2008.

[15] R. Mizoguchi and J. Bourdeau, "Using ontological engineering to overcome AI-ED problems," *Int. J. AIED*, vol. 11, no. 2, pp. 107–121, 2000.

[16] J. Perczel, J. Chow, and D. Demszky, "TeachLM: Post-training LLMs for education using authentic learning data," arXiv:2510.05087, 2025.

[17] D. Sanyal et al., "Pedagogical teacher and student LLM agents: Genetic adaptation meets RAG," arXiv:2505.19173, 2025.

[18] A. Scarlatos et al., "Training LLM-based tutors to improve student learning outcomes," in Cristea, A.I., Walker, E., Lu, Y., Santos, O.C., Isotani, S. (eds.) *Artificial Intelligence in Education: Proceedings of the 26th International Conference on Artificial Intelligence in Education (AIED 2025)*, LNAI, Springer, pp. 251–266, 2025.

[19] S. Sonkar and S. Chaudhary, "Pedagogical alignment of large language models," in Al-Onaizan, Y., Bansal, M., Chen, Y.-N. (eds.) *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA, pp. 13641–13650, 2024.

[20] B. P. Woolf, *Building Intelligent Interactive Tutors*. Morgan Kaufmann, 2009.

[21] J. Woodrow, S. Koyejo, and C. Piech, "Improving generative AI student feedback: DPO with teachers in the loop," *Educationa Data Mining Conference*, pp. 442–449, 2025.

[22] J. chetagni, R. Nkambou, J. Bourdeau, "Explicit reflection in prolog-tutor," *Int. J. AIED*, vol. 17, no. 2, pp. 169–215, 2007.

[23] R. Nkambou, J. Bourdeau, and R., Mizoguchi (eds.): *Advances in Intelligent Tutoring Systems*, Studies in Computational Intelligence, vol. 308, Springer, Berlin, Heidelberg, 2010.