CoLearn: An Agentic Tutor that Learns its Learner in a Human–AI Co-Learning Loop

Kailai He, Zhihao Wu, Linhai Zhang, Runcong Zhao, Yulan He, Jiazheng Li
King's College London
hekailai@gmail.com, {zhihao.wu, linhai.zhang, runcong.zhao, yulan.he, jiazheng.li}@kcl.ac.uk

Abstract

Good tutoring adapts to the individual: it tracks what a learner knows, notices *why* they go wrong, and asks the next question that will help most. Most deployed tutoring tools instead serve fixed item banks and treat a wrong answer as a single bit of signal. We present **CoLearn**, an interactive, agentic tutor that supports an iterative tutoring loop: the learner practises, and the system builds an evidence-grounded memory of the learner's mastery and misconceptions. This memory is updated as evidence accumulates and is used to generate the next personalised question. **CoLearn** has three components: (i) a persistent learner-state memory that updates per-topic mastery with a soft-evidence variant of Bayesian Knowledge Tracing, where a large language model acts as a continuous observation function; (ii) adaptive question generation that targets the learner's weakest topic and recurring misconceptions; and (iii) an evidence view that makes personalisation visible and testable through live progress visualisation and blind A/B comparison. In blind A/B evaluation, questions conditioned on this memory are preferred over non-personalised ones 68–69% of the time, and in persona simulations with hidden ground-truth mastery the agent's belief converges toward the learner's true mastery.

1 Introduction

Good teaching is not the scoring of answers as right or wrong. A long tradition in learning science treats genuine learning as *conceptual change*: the learner gradually replaces a faulty understanding with a more accurate one (Piaget, 1952; Smith III et al., 1994), so a wrong answer is informative. It exposes the specific misconception the next round of teaching should address (Tatsuoka, 1983), where earlier behaviourist accounts recorded only whether a response was correct (Thorndike, 1913). Good instruction also meets each learner where they are, pitching the next task into the gap between what a student can do alone and what they can do with help (Vygotsky, 1978; Bloom, 1984; VanLEHN, 2011; Li et al., 2023, 2024b, 2025; Wong et al., 2026), which requires continuous diagnosis of the learner's present understanding (Black and Wiliam, 1998; Li et al., 2026). The ideal tutor, in short, maintains a progressively refined account of each learner's understanding and misconceptions and acts on it. However, a human teacher who could continuously diagnose every student is unrealistic and expensive.

Intelligent tutoring systems promise to scale this ideal, but existing deployed tools approximate it badly in three ways. **(i)** *Knowledge-tracing systems.* Classic adaptive tutors built on knowledge tracing (Corbett and Anderson, 1994; Anderson et al., 1995; Piech et al., 2015) model knowledge *coarsely*: every response is reduced to a correct/incorrect bit per skill, discarding the diagnostic content of a free-text answer. **(ii)** *LLM-based tutors.* Recent large language model tutors converse fluently and give high-quality feedback (LearnLM Team et al., 2025; Kwon et al., 2024; Macina et al., 2023; Sonkar et al., 2023; Li et al., 2024a; Zhao et al., 2025), but are largely *memoryless*. **CoLearn** focuses on connecting answer-level diagnosis to persistent learner-state memory and subsequent adaptive practice. **(iii)** *Opaque adaptation.* In both families the adaptation is *invisible*: the learner cannot inspect what the system believes about them (Bull and Kay, 2007; Bodily and Verbert, 2017).

To close this gap and let more students benefit from personalised tutoring, we present **CoLearn**, an interactive tutor that operationalises the ideal tutor as a *human–AI co-learning loop* and, crucially, makes its mastery and misconception estimates *visible* and its personalisation *testable*. The learner answers a question; the agent grades it, mines the misconception it reveals, updates a persistent per-topic mastery estimate, and uses that updated belief to generate the next, better-targeted question.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 56–65
October 24-29, 2026 ©2026 Association for Computational Linguistics

Figure 1: **CoLearn** demo workflow across learner and administrator views. The learner first configures a practice session, answers an adaptive question, and receives diagnostic feedback with answer-linked annotations and live mastery estimates. The progress and administrator views expose the same personalization evidence through learner progress, blind A/B analytics, and agent convergence diagnostics.

The interface renders the agent's evolving belief in real time, so personalisation is something the user can see rather than take on trust, and a built-in blind A/B test supports blinded comparison of adaptive and random-topic questions. **CoLearn** is multi-subject (evaluated on Biology and Chemistry curricula), supports both multiple-choice and short-answer style questions, and deploys as a single-node container stack with a provider-selectable LLM backend. We use *co-learning* in a deliberately modest sense: the learner practises the subject while the system refines its memory of the learner's mastery and misconceptions. **CoLearn** surfaces how these estimates change and stabilise during use, while the question strategy stays fixed (§2.3). Our demo video is available at `https://youtu.be/I1Lxg4ud3-Y`.

2 Overview of CoLearn

CoLearn is organised as *two views of one loop* (Figure 1). The **learner's view** (§2.1) is where practice happens: the learner answers a question, receives feedback, and watches the agent's current mastery estimates fill in and update in real time. The **evidence view** (§2.4) is where personalisation can be inspected through a progress dashboard and a blind A/B comparison of personalised and non-personalised questions. Between the two sits the engine of the loop: a **persistent learner-state memory** that stores mastery estimates and mined misconception summaries (§2.2), and an **adaptive question generator** that acts on this memory (§2.3). Figure 2 sketches the full loop.

2.1 Learner Interface

A session begins with a short setup: the learner picks a *subject*, the *topics* to work on, and a *mode* (§3 explains the modes; the default is the adaptive system).

CoLearn then drives a streamed loop. It selects or generates the next item and streams its reasoning stages ("generating your question…") so the agent's work is visible. The learner answers (free text for short-answer items, or an option for multiple-choice), and the answer is graded with streamed stages ("marking your answer…"). The feedback card shows a plain-language verdict, a "watch out" line surfacing the most salient misconception as friendly guidance, and, for free-text answers, the learner's own words re-rendered with span-level highlights linked to a mark-scheme breakdown.

After each answer, the targeted topic's mastery bar animates from its previous value to its new one, and a "why this question" chip names what the item was chosen to probe, making the personalisation explicit rather than implicit.

2.2 Persistent Learner-State Memory

To personalise tutoring across sessions, **CoLearn** maintains a persistent learner-state memory con-

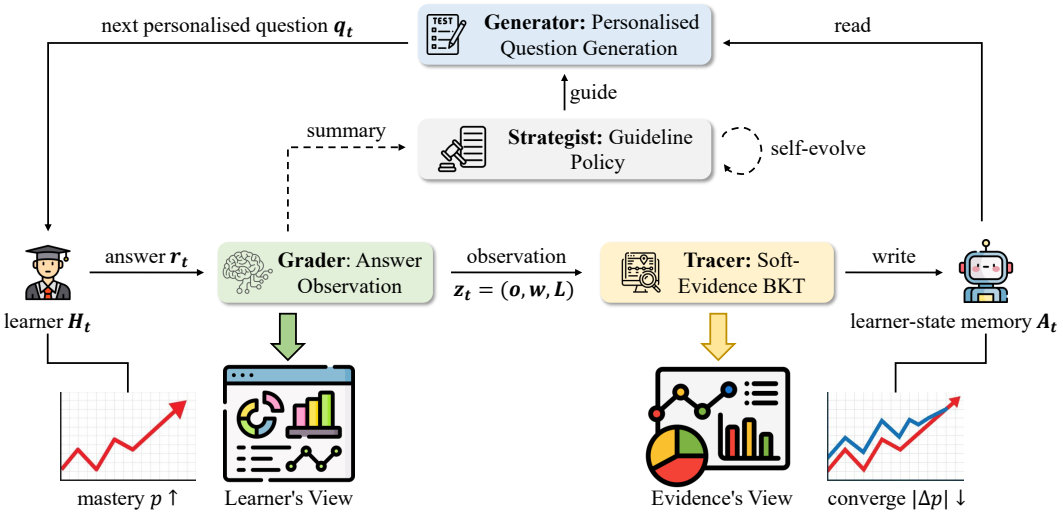

Figure 2: The **CoLearn** co-learning loop. The learner answers; an LLM grades the answer into a continuous observation; soft-evidence BKT (Eq. 1) updates the agent's cross-session memory A_t; the personalised generator reads that memory and a fixed strategy guideline to write the next, better-targeted question; and the loop repeats (the dashed *self-evolve* arrow is planned, not yet implemented). Two surfaces expose it: the *learner view* (live feedback and mastery) and the *evidence view* (progress dashboard + blind A/B).

taining topic-level mastery, recurring misconception, and the mastery trajectory. Its contents are stored in per-*(learner, subject)* records rather than per-session logs, allowing the agent to accumulate evidence over time and visualise how its belief evolves across sessions.

Formally, we model the interaction as a sequence of rounds: at round t the agent poses a question q_t, the learner responds with r_t, and the agent returns feedback f_t. Each answered round drives an update $A_t \to A_{t+1}$ of the agent's belief about the learner, while the learner's true knowledge state H_t remains latent. Concretely, A_t holds a mastery estimate $p = P(\text{known})$ per topic, a de-duplicated list of mined misconceptions, and the mastery trajectory. **CoLearn** measures and visualises A_t; a rising mastery estimate is therefore evidence that the agent's belief has changed, not proof that the learner has mastered the skill (see the Limitations section).

Soft-Evidence Knowledge Tracing. Mastery is updated with a **soft-evidence** variant of Bayesian Knowledge Tracing (Corbett and Anderson, 1994); extensions to standard BKT typically individualise its parameters or enrich its inputs (Yudelson et al., 2013; Piech et al., 2015; Abdelrahman et al., 2023; Ghosh et al., 2020). Standard BKT consumes a binary correct/incorrect observation. Instead, our LLM grader acts as a *continuous observation function*: from an answer it emits graded evidence $o = \textit{mastery_evidence} \in [0, 1]$ together with a confidence $w = \textit{evidence_strength} \in [0, 1]$, and we feed these in without binarising at 0.5. Given prior p and BKT parameters $(p_{\text{slip}}, p_{\text{guess}}, p_{\text{transit}})$, we form the Bayesian posteriors for a correct and an incorrect observation,

$$p^+ = \frac{p\,(1 - p_{\text{slip}})}{p\,(1 - p_{\text{slip}}) + (1 - p)\,p_{\text{guess}}},$$
$$p^- = \frac{p\,p_{\text{slip}}}{p\,p_{\text{slip}} + (1 - p)\,(1 - p_{\text{guess}})},$$

blend them by o, and shrink toward the prior by the confidence w, giving the confidence-shrunk posterior $p_{\text{cond}} = p + w\,(o\,p^+ + (1-o)\,p^- - p)$. We then apply a *competence-gated* learning transition:

$$p_{\text{next}} = p_{\text{cond}} + o\,(1 - p_{\text{cond}})\,p_{\text{transit}}. \quad (1)$$

Gating the learning term by o ensures a clearly wrong answer cannot raise the mastery estimate; since standard BKT applies the transition after every attempt, the update is a *variant* of BKT rather than a strict reduction (recovering standard BKT when o=1, w=1). The scheme is pure, unit-tested arithmetic with shared default parameters.

Misconception Memory. For seed items, candidate misconceptions are derived once from curriculum-aligned mark-scheme negative phrasing (e.g. "do not accept…") and attached to the item, so the diagnostic targets come from the assessment

material itself rather than a separately maintained catalogue. For *short-answer questions*, the grader returns short misconception summaries with verbatim supporting quotes. It distinguishes a *positive* label (the learner stated something wrong) from an *absence* label (an omission or a too-vague answer), and reuses a candidate's canonical wording when one matches, so labels stay consistent across sessions; the summaries accumulate (de-duplicated) in memory. For *multiple-choice items*, each distractor is tagged with the specific misconception it embodies, so *choosing a distractor is itself the diagnostic label*, graded deterministically with no LLM call (Gierl et al., 2017; Wang et al., 2021).

2.3 Adaptive Question Generation

Weakness-Targeted Generation. To act on this memory, **CoLearn** generates the next item to target their specific weaknesses (Kurdi et al., 2020). It ranks the learner's in-scope topics, targets the weakest, and selects up to two of its stored misconceptions to probe, optionally asking the LLM to link related misconceptions above a threshold. It then generates either a short-answer item or a multiple-choice item whose distractors are each bound to a known misconception. Generated items carry metadata (the targeted topic and misconceptions) that drives the visible "why this question" chip. When the learner has no in-scope history yet, curated seed items give a sensible cold start.

Strategy Layer. Beyond adapting its *belief* about the learner, **CoLearn** is designed to adapt its *strategy* that targets each individual learner's diagnosis based on the previous outcome: the question generator consults a natural-language guideline appended to its prompt (e.g. "when targeting a stored misconception, prefer MCQ with one distractor per misconception"), so revising this text immediately changes how the next item is written, with no code change or redeployment. This is the hook for the agent-side half of co-learning, stated plainly: today the guideline is a *fixed* text consulted on every generation; automatic revision from accumulated outcomes is planned but not yet implemented (see the Limitations section). The co-learning the live system demonstrates is therefore memory-side: the learner receives targeted practice while the agent updates its stored estimates from accumulated evidence.

2.4 Evidence View

To make the agent's adaptation transparent, **CoLearn** provides a progress dashboard with a branch-graph timeline, concept-mastery radar, and mastery-over-time trajectory, allowing users to inspect how the mastery estimates and misconception summaries change across sessions. **CoLearn** also includes a blind A/B interface to evaluate personalisation capability, presenting personalised and non-personalised questions in randomised, unlabelled positions and asking users which is more relevant. Results are aggregated into an analytics dashboard reporting the preference rate, a 95% confidence interval, and a two-sided sign test against the 0.5 chance rate.

2.5 System Architecture

CoLearn is a container-deployed web application. The frontend is a Vite + React + TypeScript single-page app. The backend is FastAPI with an async MongoDB store, JWT/bcrypt authentication, and SSE streaming for live generation and grading. The LLM backend is provider-selectable (AWS Bedrock with Claude Sonnet 4.5 as the live default; an OpenAI-compatible path is also supported). The whole stack starts with one command via Docker Compose (MongoDB, backend, a one-shot idempotent seeder, and an Nginx proxy serving the SPA and proxying the API). The backend ships with a test suite covering BKT, grading, generation, streaming, RBAC, and the A/B analytics.

3 Evaluation

We evaluate **CoLearn** along three dimensions: perceived relevance of personalised versus non-personalised questions (§3.2), learner-state estimation accuracy and convergence under persona simulation (§3.3), and deployment responsiveness and cost (§3.4). §3.1 first defines the session conditions and metrics shared across these studies.

3.1 Conditions and Metrics

CoLearn supports three session conditions that isolate *personalisation* as the variable of interest:

- **Adaptive**: the full system. Items are generated to target the learner's weakest topics, and memory is updated after every answer.
- **Random-topic**: the non-personalised control. Items are drawn from topics chosen at random *within the learner's selected scope* (mastery-blind), but memory is still updated. This iso-

Breakdown	n	Pref. rate	95% CI	p
Biology (overall)	18	0.68	[0.60, 0.77]	0.008
mixed learners	6	0.74	[0.71, 0.77]	
strong learners	6	0.73	[0.59, 0.88]	
Chemistry (overall)	18	0.69	[0.60, 0.77]	0.008
mixed learners	6	0.77	[0.61, 0.93]	
strong learners	6	0.66	[0.52, 0.80]	

Table 1: **Blind A/B preference for personalised over non-personalised questions.** *Pref. rate* is the mean per-rater preference, *95% CI* is across raters, and p is a two-sided sign test vs. 0.5. The mixed/strong rows are learner ability tiers (not subjects); weak-tier learners are excluded because, being weak on every skill, they cannot discriminate targeting, so the tier rows sum to fewer than the *all* row.

lates the effect of *targeting* specifically, holding adaptivity constant.

- **Frozen**: a memory-convergence control. Fixed items are served in order, and memory is read but not written, giving a flat agent-state baseline.

3.2 Does Personalisation Produce More Relevant Questions?

Using the built-in A/B tool, each comparison presents a personalised (Adaptive) question and a non-personalised (Random-topic) question in two blinded slots; slot order is randomised and provenance is hidden server-side. The rater answers the question "which question feels more relevant to this learner right now?". Picks are stored with their (hidden) provenance for later analysis; A/B items are not written back into the learner-state memory.

We recruited 18 raters with a relevant background (e.g., they have studied and passed the curriculum). Each rater completed 12 comparisons, yielding $M = 18 \times 12$ judgements. We report the *preference rate* (fraction of comparisons in which the personalised item was chosen), its 95% confidence interval, and a two-sided sign test against the chance rate of 0.5.

The personalised item is preferred 0.68 of the time across Biology raters (0.74 and 0.73 for mixed- and strong-ability learners) and 0.69 across Chemistry raters ($p = 0.008$, sign test over 18 raters). Preference is lower in the strong tier (0.73 Biology, 0.66 Chemistry) than the mixed tier, consistent with targeting mattering less when there is little weakness to target; we do not test the six-rater tiers individually.

Metric	Adaptive	Random	Frozen
Weak-skill hit (warm)	**0.72**	0.55	0.57
Mixed learners	**0.89**	0.57	–
Strong learners	**0.28**	0.08	–
Mastery MAE↓	**0.12**	0.16	0.20

Table 2: **User-simulation** (hidden ground-truth mastery). Weak-skill hit rows are from an 18-persona warm-start sweep; the Mastery MAE row is from the 6-persona Claude Sonnet 4.5-graded run. Weak-skill hit = fraction of served items on a truly-weak skill (higher = better targeting); Mastery MAE = final agent belief vs. true mastery (lower = better). Adaptive beats the Random-topic / Frozen controls on both. (Weak personas are weak on every skill, so they cannot discriminate targeting and are excluded from the per-tier rows.)

3.3 Do Learner-State Estimates Track Ground Truth?

Mastery prediction error and misconception P/R/F1 require ground truth, so we drive the loop with synthetic personas whose per-skill mastery and misconceptions are hidden from the system (Markel et al., 2023; Tack and Piech, 2022): a database-free harness exercises the *real* grading and BKT services, so answers flow through the live system's observation function. On a run graded by the live framework model (Claude Sonnet 4.5; 6 personas, 132 rounds; the LLM grade serves as the observation, cf. LLM-as-judge reliability (Zheng et al., 2023)), per-answer *mastery_evidence* correlates with true mastery at Pearson $r = 0.68$ pooled across tiers. This pooled figure is driven mainly by between-tier separation; however, mean evidence rises cleanly by ability tier (0.25 weak / 0.67 mixed / 0.77 strong), but the within-tier correlation is much weaker ($r \approx 0.15$ / 0.48 / 0.41), and weakest for low-ability personas. The observation function is therefore least reliable for exactly the learners who most need accurate diagnosis (see Limitations). Mined misconceptions match the personas' assigned ones at F1 ≈ 0.56.[1] This F1 score evaluates the grader's recovery of persona misconception labels from learner answers. It does not test whether a generated question elicits the misconception named in its generation metadata.

Accuracy and convergence. A shrinking update magnitude shows only that the belief has *stabilised*, not that it is correct, so accuracy against the hidden ground truth is the primary metric: final-

[1] Computed post hoc by token-overlap matching of mined labels against assigned misconceptions over all 132 rounds.

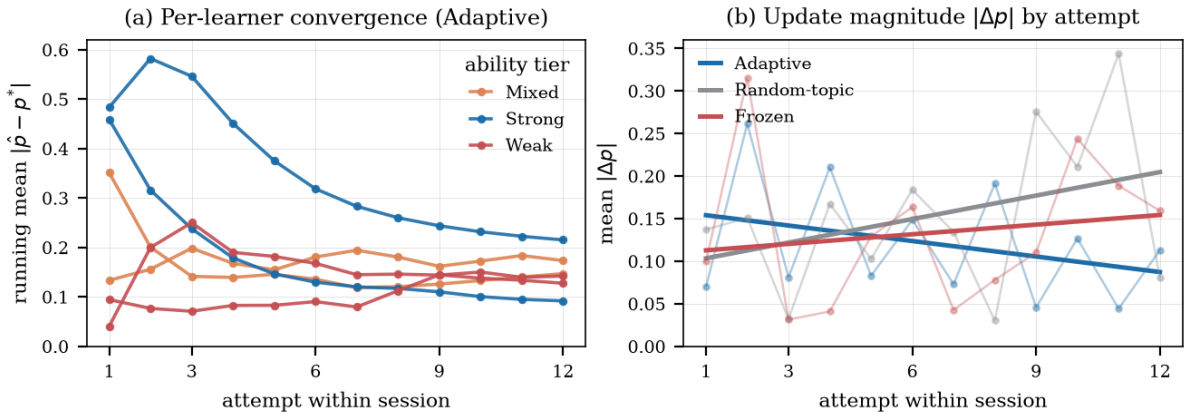

Figure 3: Memory convergence from the logged attempts (6 personas, 132 rounds, live framework grader Claude Sonnet 4.5). **(a)** Per-learner running mean belief error $|\hat{p} - p^*|$ under *Adaptive*: every learner's estimate settles toward its hidden true mastery, most visibly the two strong learners that start far off. **(b)** Mean update magnitude $|\Delta p|$ by attempt (thin = per-attempt mean, bold = linear trend): under *Adaptive* it shrinks ($0.14 \rightarrow 0.10$) as the belief locks on, while the *Random-topic* and *Frozen* controls stay flat or drift upward.

belief mastery MAE is lowest under *Adaptive* (0.12 vs. 0.16/0.20; Table 2), and the per-learner belief error $|\hat{p} - p^*|$ falls in step (Figure 3a). As a secondary aggregate signal, mean $|\Delta p|$ shrinks across the six *Adaptive* persona sessions ($0.14 \rightarrow 0.10$) but stays flat or drifts upward under both controls (Figure 3b). Appendix C walks one live adaptive round end-to-end.

3.4 Usability and Efficiency

To evaluate the learner-facing experience, we asked 16 human raters to answer a 1–5 Likert questionnaire of 13 items across six dimensions after a session (full instrument in Appendix A). This questionnaire are designed to capture perceived relevance and usefulness. It does not evaluate whether generated questions reliably elicit their intended misconceptions. Figure 4 shows the per-item distribution; the overall mean is $4.0/5$, no dimension below $\approx$3.7, and 75% ($12/16$) would reuse. The lowest-rated dimension is *Responsiveness* (3.7), consistent with the measured short-answer latency (§3.4); researchers were the most critical (mean 3.8) and students the most positive (4.4).

Efficiency. Deployability hinges on cost and latency. A short-answer round uses two LLM calls (grading and feedback) plus generation of the next item; a multiple-choice round is graded and explained *deterministically* with no LLM call. Table 3 reports mean latency measured on the live single-node deployment (AWS Bedrock, Claude Sonnet 4.5). The grader/feedback pass dominates short-answer latency; multiple-choice rounds are $\approx 4\times$ faster because only the next-item generation calls the LLM. A faster model (e.g. Claude Haiku) is the obvious lever for a booth setting.

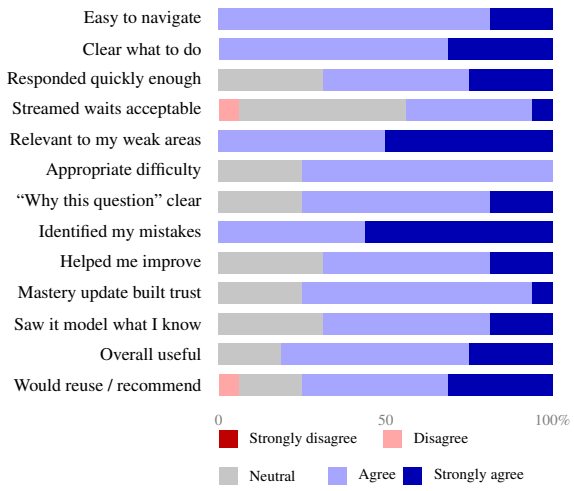

Figure 4: Per-item response distribution of the 1–5 Likert questionnaire over 16 raters. Blue = agree, grey = neutral, red = disagree. No item drew a strongly-disagree response; the weakest is the streamed-wait item (Q4), echoing the measured latency (§3.4).

Round type	Next/gen (s)	Answer (s)	Cost ($/rnd)
Short-answer	12.1	20.2	$\approx$0.015
Multiple-choice	7.6	0.01	$\approx$0.005

Table 3: Efficiency on the live backend (Bedrock Claude Sonnet 4.5). Short-answer $\approx$32 s/round (grade+feedback dominate); MCQ $\approx$7.6 s (deterministic answer). Cost is estimated from token usage.

4 Conclusion

CoLearn demonstrates how an LLM tutor can maintain a persistent learner-state memory, use that memory to generate personalised questions, and expose the resulting adaptation through live visualisation and built-in A/B evaluation. Its soft-evidence knowledge-tracing memory converges on the learner across sessions, its generator targets the learner's weakest topics and misconceptions, and its evidence view makes the personalisation testable by anyone. By connecting learner-state estimation,, question generation, and transparent evidence surfaces in one deployable, container-deployed research prototype, **CoLearn** provides a platform for the harder question ahead: whether visible, converging personalisation produces durable, transferable learning.

Acknowledgments

This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2) and the Prosperity Partnership scheme (grant no. UKRI566).

Limitations

The agent's mastery estimate is its belief A_t, not a direct measurement of human knowledge H_t; we therefore do not claim learning gains, and our A/B study measures *perceived* relevance rather than learning outcomes. BKT parameters are shared priors, not fit per skill, so absolute mastery values are indicative rather than calibrated. The LLM observation function is also least reliable for low-performing learners: within-tier, per-answer evidence correlates with true mastery at only $r \approx 0.15$ for the weak tier, versus $0.48/0.41$ for mixed/strong (the higher pooled $r = 0.68$ reflects between-tier separation, not within-tier accuracy). These are precisely the learners who most need accurate diagnosis; improving grading fidelity on weak answers is a priority for future work. Our evaluation does not establish that generated questions elicit their intended misconceptions, nor does it independently assess factual correctness, curriculum fit, answerability, or distractor quality. These properties require expert annotation and learner-response studies. Automatic strategy self-evolution and the co-evolution diagnostics (convergence/transfer estimation) are designed but not yet implemented: during use the strategy guideline is fixed, so only the agent's memory adapts. Far-transfer learning, the strongest test of genuine competence, is out of scope here and left to future work. Due to differences in experimental setup and evaluation protocols, we do not make direct comparisons with other LLM-based tutoring systems.

Ethics Statement

CoLearn involves human participants (the A/B and survey raters) and learner data. Accounts are admin-provisioned with no public signup, learner data is stored locally in the single-node deployment, and the system can run fully offline with a deterministic LLM for privacy-sensitive settings. Misconception summaries are surfaced to learners as supportive guidance rather than verbatim labels.

References

Ghodai Abdelrahman, Qing Wang, and Bernardo Nunes. 2023. Knowledge tracing: A survey. *ACM Comput. Surv.*, 55(11).

John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray. Pelletier. 1995. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207.

Paul Black and Dylan Wiliam. 1998. Assessment and classroom learning. *Assessment in Education*, 5:7–74.

Benjamin S. Bloom. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16.

Robert Bodily and Katrien Verbert. 2017. Review of research on student-facing learning analytics dashboards and educational recommender systems. *IEEE Transactions on Learning Technologies*, 10(4):405–418.

Susan Bull and Judy Kay. 2007. Student models that invite the learner in: The smili:() open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120.

Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. 2020. Context-aware attentive knowledge tracing. *Preprint*, arXiv:2007.12324.

Mark J. Gierl, Okan Bulut, Qi Guo, and Xinxin Zhang. 2017. Developing, analyzing, and using distractors for multiple-choice tests in education: A comprehensive review. *Review of Educational Research*, 87(6):1082–1116.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.

Soonwoo Kwon, Sojung Kim, Minju Park, Seunghyun Lee, and Kyuseok Kim. 2024. Biped: Pedagogically informed tutoring system for esl education. *Preprint*, arXiv:2406.03486.

LearnLM Team, Abhinit Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, Brett Wiltshire, Brian Veprek, Daniel Gillick, Daniel Kasenberg, Derek Ahmed, Irina Jurenka, James Cohan, Jennifer She, Julia Wilkowski, Kaiz Alarakyia, Kevin R. McKee, Lisa Wang, Markus Kunesch, Mike Schaekermann, and 27 others. 2025. Learnlm: Improving gemini for learning. *Preprint*, arXiv:2412.16429.

Jiazheng Li, Artem Bobrov, David West, Cesare Aloisi, and Yulan He. 2024a. An automated explainable educational assessment system built on llms. *Preprint*, arXiv:2412.13381.

Jiazheng Li, Lin Gui, Yuxiang Zhou, David West, Cesare Aloisi, and Yulan He. 2023. Distilling ChatGPT for explainable automated student answer assessment. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6007–6026, Singapore. Association for Computational Linguistics.

Jiazheng Li, Hainiu Xu, Zhaoyue Sun, Yuxiang Zhou, David West, Cesare Aloisi, and Yulan He. 2024b. Calibrating LLMs with preference optimization on thought trees for generating rationale in science question scoring. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5452–5479, Miami, Florida, USA. Association for Computational Linguistics.

Jiazheng Li, Emine Yilmaz, Bei Chen, and Thu Le. 2026. Towards self-improving error diagnosis in multi-agent systems. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 2063–2077, San Diego, California, United States. Association for Computational Linguistics.

Jiazheng Li, Yuxiang Zhou, Junru Lu, Gladys Tyen, Lin Gui, Cesare Aloisi, and Yulan He. 2025. Two heads are better than one: Dual-model verbal reflection at inference-time. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3119–3140, Suzhou, China. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Julia M. Markel, Steven G. Opferman, James A. Landay, and Chris Piech. 2023. Gpteach: Interactive ta training with gpt-based students. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, L@S '23, page 226–236, New York, NY, USA. Association for Computing Machinery.

J. Piaget. 1952. *The Origins of Intelligence in Children*. International Universities Press paperback library. International Universities Press.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

John P. Smith III, Andrea A. diSessa, and Jeremy Roschelle. 1994. Misconceptions reconceived: A constructivist analysis of knowledge in transition. *Journal of the Learning Sciences*, 3(2):115–163.

Shashank Sonkar, Naiming Liu, Debshila Basu Mallick, and Richard G. Baraniuk. 2023. Class: A design framework for building intelligent tutoring systems based on learning science principles. *Preprint*, arXiv:2305.13272.

Anaïs Tack and Chris Piech. 2022. The ai teacher test: Measuring the pedagogical ability of blender and gpt-3 in educational dialogues. *Preprint*, arXiv:2205.07540.

KIKUMI K. Tatsuoka. 1983. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354.

E.L. Thorndike. 1913. *The psychology of learning*. Educational Psychology. Teachers College, Columbia University.

Kurt VanLEHN. 2011. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221.

L. S. Vygotsky. 1978. *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press.

Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E. Turner, Richard G. Baraniuk, Craig Barton, Simon Peyton Jones, Simon Woodhead, and Cheng Zhang. 2021. Instructions and guide for diagnostic questions: The neurips 2020 education challenge. *Preprint*, arXiv:2007.12061.

Gee-Lyle Wong, Runcong Zhao, Yulan He, and Jiazheng Li. 2026. From questions to assessment tuples: A multi-agent framework with bloom-specialized agents and automated verification. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, pages 292–335, San Diego, California, USA. Association for Computational Linguistics.

Michael V. Yudelson, Kenneth R. Koedinger, and Geoffrey J. Gordon. 2013. Individualized bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180, Berlin, Heidelberg. Springer Berlin Heidelberg.

Runcong Zhao, Artem Bobrov, Jiazheng Li, Cesare Aloisi, and Yulan He. 2025. LearnLens: LLM-enabled personalised, curriculum-grounded feedback with educators in the loop. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 625–633, Suzhou, China. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A User-study questionnaire

The instrument used in the user study (§3.4). Each statement is rated 1–5 (1 = strongly disagree, 5 = strongly agree); Q13 is the adoption item. Raters span a range of backgrounds (teacher, researcher, student, parent) and complete the items after a session.

Usability.
Q1 The interface was easy to navigate.
Q2 At each step it was clear what I was meant to do.

Responsiveness.
Q3 The system responded quickly enough to keep me engaged.
Q4 The streamed "thinking" stages made any wait feel acceptable.

Question quality.
Q5 The questions were relevant to my weak areas.
Q6 The questions were pitched at an appropriate difficulty.
Q7 The "why this question" note made the targeting understandable.

Feedback usefulness.
Q8 The feedback clearly identified my mistakes/misconceptions.
Q9 The feedback helped me understand how to improve.

Transparency (co-learning).
Q10 Seeing the mastery estimate update after each answer built my trust.
Q11 I could see the tutor building a model of what I know.

Overall.
Q12 Overall, CoLearn was useful for learning.

Adoption.
Q13 I would use CoLearn again / recommend it.

B Generation & grading prompts

The boxes below give the system prompts used by the live system, lightly reformatted for the single-column width (code identifiers are shown in `{}`; the `{subject_label}` slot is filled with the active subject, e.g. "AQA GCSE Biology").

Grading: the LLM observation function

Role. An expert `{subject_label}` answer analyst that reads one student answer at a time and extracts evidence for a topic-level memory profile.

Core rules.

- Output only valid JSON matching the schema below.
- `{mastery_evidence}` and `{evidence_strength}` are floats in $[0, 1]$.
- `{label_summaries}` and `{label_evidence_quotes}` have equal length and correspond index-for-index.
- The predicted label list starts EMPTY and grows only from clear evidence in the answer.

Two label types.

- **Positive** — the student stated something factually wrong; the evidence quote is a verbatim phrase copied from `student_answer`.
- **Absence** — an omission or a too-vague answer; the evidence quote is either the full short answer or the literal token `<missing: "the specific omitted idea">`. Never invent a quote not in the answer.

Canonical preference. When a `candidate_label_summaries` entry matches the detected misconception in meaning, copy that candidate string verbatim; invent new wording only when no candidate fits. *No two labels may share an evidence quote* (or have one be a substring of another).

Schema: `{ "topic": ..., "mastery_evidence": 0.0, "evidence_strength": 0.0, "label_summaries": [...], "label_evidence_quotes": [...], "rationale": ... }`

Personalised question generation (short-answer)

Role. Writes a new `{subject_label}`-style diagnostic question targeting a weak topic for a specific student and, when supplied, one or two target diagnostic error labels.

Rules.

- Output only valid JSON.
- Write a concise exam-style question; include a mark scheme or expected answer.
- Make it diagnostic: a weak answer should expose the target error or vague understanding.

- Use GFM Markdown for tables/lists and LaTeX for equations.
- Do not mention AI generation; do not include hidden mastery scores.

Schema: `{ "question": ..., "mark_scheme": ..., "expected_diagnostic_signal": "what a weak/incorrect answer should reveal" }`

Misconception-tagged MCQ generation

Role. Writes a new `{subject_label}`-style multiple-choice question from a topic, a student profile, and a list of target misconceptions.

Rules.

- Output only valid JSON; exactly one correct option.
- Provide one distractor for EACH supplied misconception; a student holding that misconception should be tempted to pick it.
- Map each distractor to the exact misconception text it tests.
- Keep options parallel in length/style; do not reveal the correct option or mention AI.

Schema: `{ "question": ..., "correct_option": ..., "distractors": [{"text": ..., "misconception": ...}], "expected_diagnostic_signal": "what choosing a distractor reveals" }`

A fourth prompt plans whether two misconceptions can be tested together (*same mechanism*, *cause–effect*, *commonly-confused contrast*, etc.) before generation; it is omitted here for space.

Diagnostic-quality judge (0–3)

Role. An expert `{subject_label}` question-quality judge that scores whether a generated question matches its intended weakness target.

Rules.

- Output only valid JSON.
- Choose the topic the question actually assesses, and only the supplied label summaries it could diagnose.
- Score diagnostic quality: **0** not diagnostic; **1** loosely related but weak; **2** clearly tests the topic; **3** clearly exposes the target diagnostic error label.

Schema: `{ "judged_topic": ..., "judged_label_summaries": ["exact supplied label summary"], "diagnostic_quality_score": 0, "rationale": ... }`

C Worked co-learning trace

One real round captured from the live deployment (AWS Bedrock, Claude Sonnet 4.5), GCSE Biology, skill *Levels of organisation within an ecosystem*.

Question. "Water is an abiotic factor. Name one biotic factor which may affect the number of buttercups growing on the field."

Learner answer. "Water is a biotic factor, and biotic and abiotic factors basically mean the same thing."

Grader (observation function) → `mastery_evidence=0.0`; mined misconceptions: (i) "incorrectly states water is a biotic factor"; (ii) "incorrectly states biotic and abiotic factors mean the same thing"; (iii) "fails to name any biotic factor".

Feedback shown. "Not quite yet — let's work on this together."

Memory update. mastery $0.56 \rightarrow 0.23$ (the competence-gated BKT lowers the estimate because the answer is clearly wrong; the misconceptions are stored on the skill).

Next turn. this skill is now the learner's weakest, so the adaptive engine re-targets it; on the follow-up item the learner answered correctly and the estimate recovered ($0.23 \rightarrow 0.56$).

This is the loop in miniature: a graded answer becomes structured evidence, the agent's belief moves, and the next question is chosen from that updated belief.