

RPKT: Learning What You Don't Know - Recursive Prerequisite Knowledge Tracing in Conversational AI Tutors for Personalized Learning

Jinwen Tang*

EECS Department
University of Missouri
Columbia, MO, USA
jt4cc@umsystem.edu

Qiming Guo*

Department of Computer Science
Texas A&M University–Corpus Christi
Corpus Christi, TX, USA
qguo2@islander.tamucc.edu

Zhicheng Tang

Independent Researcher
University of Missouri
Columbia, MO, USA
robert.tang.robot@gmail.com

Yi Shang

EECS Department
University of Missouri
Columbia, MO, USA
shangy@umsystem.edu

Abstract—Educational systems often assume learners can identify their knowledge gaps, yet research consistently shows that students struggle to recognize what they don't know they need to learn—the "unknown unknowns" problem. This paper presents a novel Recursive Prerequisite Knowledge Tracing (RPKT) system that addresses this challenge through dynamic prerequisite discovery using large language models. Unlike existing adaptive learning systems that rely on pre-defined knowledge graphs, our approach recursively traces prerequisite concepts in real-time until reaching a learner's actual knowledge boundary. The system employs LLMs for intelligent prerequisite extraction, implements binary assessment interfaces for cognitive load reduction, and provides personalized learning paths based on identified knowledge gaps. Demonstration across computer science domains shows the system can discover multiple nested levels of prerequisite dependencies, identify cross-domain mathematical foundations, and generate hierarchical learning sequences without requiring pre-built curricula. Our approach shows great potential for advancing personalized education technology by enabling truly adaptive learning across any academic domain.

Index Terms—Recursive Knowledge Tracing; Prerequisite Discovery; Large Language Models; Unknown Unknowns; Intelligent Tutoring Systems; Knowledge Gap Identification; Adaptive Learning; Educational AI; Dynamic Knowledge Graphs

I. INTRODUCTION

Effective education requires alignment between what instructors teach and what students need to learn. However, this alignment faces some fundamental challenges that create persistent barriers to learning success.

First, the expert blind spot phenomenon causes instructors who have mastered a domain to unconsciously skip over foundational concepts that novices need [1]. Experts develop automated knowledge structures that make it difficult to remember the incremental steps required for initial understanding [2]. This cognitive gap between expert instructors and novice learners results in curricula that assume prerequisite knowledge students may not possess.

Second, learners face dual cognitive challenges. They suffer from "unknown unknowns", the inability to recognize knowledge gaps they need to address [3]. Research consistently

shows that students overestimate their understanding and cannot identify missing prerequisites [4]. Additionally, previous research reveals that students rate passive learning experiences more favorably despite achieving lower learning outcomes, with this preference driven by the increased cognitive effort required for active engagement [5]. The cognitive load of formulating questions about concepts they don't understand creates a barrier that prevents students from seeking necessary clarification.

Current educational technologies fail to bridge this expert-novice gap effectively. Traditional approaches rely on static prerequisite structures that cannot adapt to individual knowledge profiles [6]. While intelligent tutoring systems have shown promise, they require extensive domain modeling and pre-built knowledge graphs [7]. Recent LLM-based educational tools can generate explanations but still assume students can articulate what they need to learn, which is an assumption invalidated by the unknown unknowns problem.

This paper presents **RPKT - Recursive Prerequisite Knowledge Tracing**, a novel system that addresses these challenges through three key innovations:

- 1) It eliminates the cognitive burden of question formulation through binary knowledge assessment (know/don't know), enabling learners to focus on self-evaluation rather than articulation.
- 2) It dynamically discovers prerequisite dependencies through recursive LLM-powered tracing, systematically revealing the "unknown unknowns" that learners cannot identify independently.
- 3) It generates personalized learning paths from identified knowledge boundaries to target concepts, bridging the expert-novice gap without requiring pre-built curricula or domain-specific knowledge graphs.

Our approach transforms the learning problem from "what questions should I ask?" to simple binary decisions, aligning with learners' cognitive preferences while systematically uncovering unknown unknowns.

* The first two authors contributed equally to this work.

II. RELATED WORK

a) Knowledge Tracing and Adaptive Learning: Traditional knowledge tracing models focus on estimating what students know rather than discovering what they need to learn. Deep Knowledge Tracing [8] uses recurrent neural networks to model student knowledge states, while subsequent work has incorporated attention mechanisms [9]. However, these approaches require predefined skill sets and cannot discover unknown prerequisites dynamically. Adaptive learning systems like those reviewed by [7] personalize content delivery but remain constrained by expert-authored knowledge structures that may not reflect individual learning needs.

b) Question Generation and Student Engagement: Research on educational question generation has primarily focused on creating assessment items rather than discovering knowledge gaps [10]. Studies consistently show that students struggle with question formulation, with many being reluctant to reveal knowledge gaps [11]. The cognitive load theory explains why students prefer passive reception—question generation requires significant mental effort that many learners find overwhelming [12]. Our binary assessment approach directly addresses this limitation by eliminating the question generation burden.

c) Recent Trends in LLM Disciplinary Applications: Since 2023, AI has evolved from computer vision and machine learning to NLP and toward AGI through LLMs. GPT-based models have integrated into daily life while excelling in interdisciplinary domains including knowledge question-answering, medical diagnosis, mental health support, and autonomous driving [13]–[17]. LLMs have also demonstrated significant impact in clinical medicine with models like ChatDoctor providing medical consultation [18] and Med-PaLM achieving expert-level performance on medical exams [19], legal document analysis [20], scientific research acceleration [21], code generation and software engineering [22], financial market analysis [23], and multimodal biomedical applications [24]. These technologies increasingly automate complex cognitive tasks across diverse professional domains.

d) LLMs in Education: Recent work has explored using large language models specifically for educational applications. ChatGPT and similar models have been used for tutoring, feedback generation, and content creation [25]. However, these applications require students to formulate effective prompts to get useful responses, which is a significant barrier based on the difficulty in question generation. Additionally, existing LLM tools typically provide direct answers or explanations, assuming students can articulate their learning needs. [26] provides a comprehensive survey of knowledge tracing approaches but notes the limitation that current systems cannot identify what students don’t know they don’t know. Our work differs by using LLMs for dynamic prerequisite discovery rather than direct instruction, eliminating the need for students to formulate questions.

e) Prerequisite Learning and Curriculum Design: Traditional curriculum design relies on expert-defined prerequisite chains that may not match individual learning trajectories [27]. Research in computer science education has shown

that students often lack unexpected prerequisites, particularly mathematical foundations [28]. Recent work on prerequisite relation extraction from educational materials [29] still requires existing content and cannot generate prerequisites dynamically. Our recursive approach discovers these hidden dependencies in real-time without predefined structures.

The key limitation across all related work is the assumption that either experts can fully specify prerequisite structures or students can identify their own knowledge gaps. RPKT addresses both limitations through dynamic discovery and binary assessment, creating a novel approach to personalized learning.

Algorithm 1 Recursive Prerequisite Knowledge Tracing

Input: User question Q , education level E , max depth d_{\max}

Output: Knowledge tree T , explanation \mathcal{E}

```

1: analysis  $\leftarrow$  ANALYZEQUESTION( $Q, E$ )
2: status  $\leftarrow$  {},  $T \leftarrow$  INITTREE( $Q$ )
3: for  $c \in$  analysis.key_concepts do
4:   RECURSIVETRACE( $c, 1, d_{\max}, T, status$ )
5:  $\mathcal{E} \leftarrow$  GENERATEEXPLANATION( $Q, status$ )
6: return  $T, \mathcal{E}$ 
7: function RECURSIVETRACE( $c, depth, d_{\max}, T, status$ )
8:   if  $depth > d_{\max} \vee$  ISFUNDAMENTAL( $c$ ) then
9:     return
10:  if  $c \notin status$  then
11:    status[ $c$ ]  $\leftarrow$  USERASSESS( $c$ )
12:  if status[ $c$ ] = False then
13:    ADDTOTREE( $T, c, depth$ )
14:    prereqs  $\leftarrow$  EXTRACTPREREQS( $c$ )
15:    for  $p \in$  prereqs do
16:      RECURSIVETRACE( $p, depth + 1, d_{\max}, T, status$ )

```

III. METHOD

a) System Architecture: Our Recursive Prerequisite Knowledge Tracing (RPKT) system employs a three-component architecture that discovers knowledge dependencies dynamically without requiring pre-built curricula. The Knowledge Tracer Engine leverages GPT-4o to extract prerequisite relationships in real-time based on the learner’s query and educational context. This engine interfaces with an Interactive Assessment Interface presenting binary evaluations through a three-tab design. The Session Management System maintains state consistency across recursive expansion levels, tracking assessed concepts and managing prerequisite exploration depth.

Unlike traditional adaptive learning systems dependent on expert-authored knowledge graphs, RPKT discovers dependencies dynamically, enabling application across any academic domain. The system transforms complex self-assessment into simple binary decisions, addressing the cognitive burden that prevents learners from identifying their own knowledge gaps.

How it works

1. Ask a question about any topic
2. Understand the topic - we'll explain what you want to learn
3. Check prerequisites - mark what you know (✓) or don't know (✗)
4. Trace deeper - unknown concepts expand automatically
5. Learn efficiently - get a personalized learning path

Recursive Prerequisite Knowledge Tracer

Learn efficiently by understanding what you don't know

What would you like to learn about?

How does backpropagation work in neural networks?

Analyze Prerequisites

Understanding Your Question

What you want to learn:

Backpropagation is a fundamental algorithm used in training artificial neural networks. It involves calculating the gradient of the loss function with respect to each weight by the chain rule, allowing the model to update its weights in the direction that minimizes the error. This process is essential for optimizing the performance of neural networks by iteratively adjusting the weights to improve accuracy. Backpropagation is typically used in conjunction with optimization techniques like gradient descent to efficiently train deep learning models.

Why it's important:

Understanding backpropagation is crucial for anyone studying neural networks because it is the backbone of how these models learn from data. It enables the adjustment of model parameters to minimize errors, which is essential for achieving high performance in tasks such as image recognition, natural language processing, and other AI applications. Mastery of this concept allows students to design and train effective neural network models, a key skill in the field of machine learning.

Key concepts to check:

These will be checked for your understanding below

• Forward Propagation

• Gradient Descent

• Chain Rule

• Loss Functions

Fig. 1: RPKT system interface showing initial assessment for a backpropagation query.

Knowledge Check

Knowledge Map

Learning Path

Check Your Prerequisites

Click ✓ if you know it, ✗ if you don't. Unknown concepts will expand automatically.

Forward Propagation [L1]

What: Forward propagation is the process of passing input data through a neural network to obtain an output prediction. It involves calculating the weighted sum of inputs and applying activation functions at each layer to generate the final output.

Why needed: Forward propagation is the first step in the learning process, providing the initial predictions that are compared against the actual targets to calculate the error.

In a neural network predicting house prices, forward propagation would involve inputting features like size and location to predict the price.

☒
☐

Gradient Descent [L1]

What: Gradient descent is an optimization algorithm used to minimize the loss function by iteratively adjusting the model's parameters. It calculates the gradient of the loss function with respect to the parameters and updates them in the opposite direction of the gradient.

Why needed: Gradient descent is used in backpropagation to update the weights of the neural network, ensuring that the error is minimized over time.

Adjusting the weights of a neural network to reduce the error in predicting stock prices.

☐
☒ Already confirmed

Prerequisites for Gradient Descent:

Derivative [L2]

What: A derivative represents the rate at which a function is changing at any given point and is a fundamental concept in calculus.

Why needed: In gradient descent, derivatives are used to determine the direction and rate of change of the cost function, guiding the optimization process.

For a function $f(x) = x^2$, the derivative is $f'(x) = 2x$, which indicates how steeply the function is increasing or decreasing.

☒
☐

Cost Function [L2]

What: A cost function, also known as a loss function, measures how well a model's predictions match the actual data, quantifying the error in the predictions.

Why needed: Gradient descent aims to minimize the cost function by iteratively adjusting the model parameters to reduce prediction error.

☒
☐

Fig. 2: Example of RPKT's recursive expansion in action. After marking "Gradient Descent" as unknown at Level 1, the system expands inline to reveal Level 2 prerequisites (Derivative, Cost Function), demonstrating the recursive prerequisite discovery process.

b) *Recursive Prerequisite Extraction Algorithm:* Algorithm 1 presents our recursive extraction process that systematically uncovers knowledge dependencies until reaching the learner's actual knowledge boundary. Given a target concept C_0 , the system extracts immediate prerequisites $P_1 = \{p_1, p_2, \dots, p_n\}$ through structured GPT-4o prompting that emphasizes directly relevant technical dependencies. Each prerequisite undergoes binary assessment where learners indicate their knowledge state as $K(p_i) \in \{0, 1\}$, eliminating the cognitive burden of granular self-rating scales.

When learners mark prerequisites as unknown ($K(p_i) = 0$), the algorithm recursively extracts sub-prerequisites, creating an expanding dependency tree. Recursion continues until reaching either the maximum depth d_{\max} or fundamental concepts requiring no further prerequisites. This bounded approach ensures computational efficiency while discovering multi-level

knowledge gaps. The recursive nature addresses the "unknown unknowns" problem by revealing dependencies learners would not anticipate.

c) *Language Model Integration:* GPT-4o integration employs carefully engineered prompts with structured JSON output format for consistent prerequisite extraction. The prompting strategy extracts 2-4 critical prerequisites per concept, a range selected to balance comprehensive coverage with manageable cognitive load. Prompts explicitly instruct the model to identify immediate technical dependencies rather than broad foundational skills, contextualizing extraction based on the specified education level.

For complex concepts, the system implements prompt chaining where initial extraction results inform subsequent queries. This approach enables cross-domain dependency discovery across mathematics, programming, machine learning, and

Your Knowledge Graph

Node size decreases with depth, labels show level

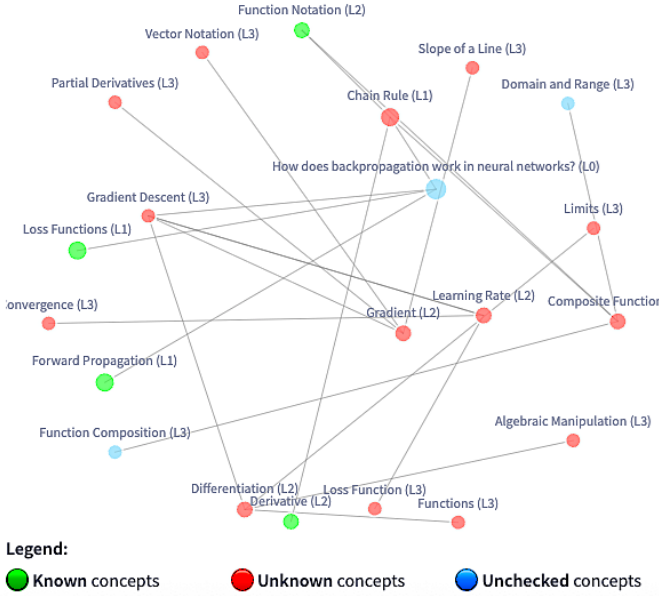


Fig. 3: Knowledge dependency graph revealing cross-domain prerequisites across mathematics, programming, and ML/DL domains.

deep learning domains. The structured JSON format ensures consistent parsing while maintaining flexibility across diverse academic topics.

d) Interactive Interface Design: The interface design minimizes cognitive load through binary assessment mechanisms replacing traditional multi-point scales. The three-tab structure separates knowledge assessment, visual exploration, and learning path review, allowing focused interaction while maintaining overall progress awareness. Real-time visual feedback immediately reflects assessment decisions—known concepts appear dimmed while unknown concepts expand inline to reveal prerequisites. Level indicators provide hierarchical orientation, and the system handles duplicate concepts intelligently, displaying "Already confirmed" status when prerequisites appear across multiple branches.

IV. DEMONSTRATION

We demonstrate RPKT's capabilities through a case query "How does backpropagation work in neural networks?" to validate our recursive prerequisite discovery approach.

a) Interface in Practice: Figure 1 shows the RPKT system interface presenting the initial analysis for a backpropagation query. The system displays the question understanding, explains why backpropagation is important, and identifies four key Level 1 concepts to check: Forward Propagation, Gradient Descent, Loss Functions, and Chain Rule. This clean interface design with clear sections for question analysis and prerequisite identification demonstrates how the system prepares learners for the assessment phase before any binary decisions are made.

Your Personalized Learning Path

Nested structure showing dependencies:

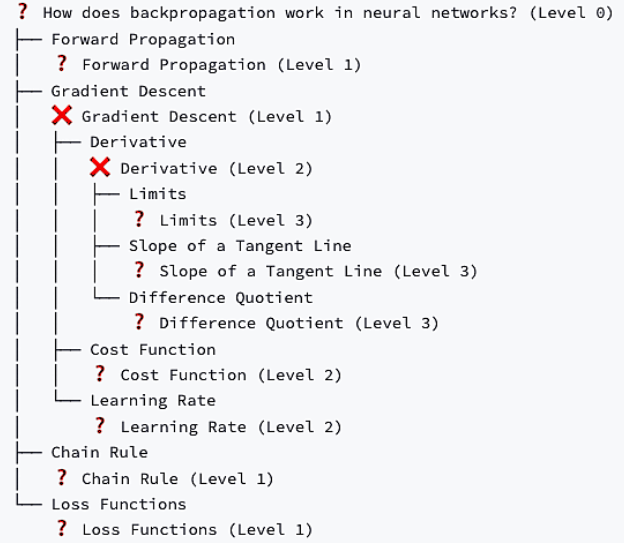


Fig. 4: Generated learning path with hierarchical prerequisite structure from foundations to target concept.

b) Depth of Discovery: Figure 2 demonstrates the recursive expansion mechanism when a user marks "Gradient Descent" as unknown. The system shows "Gradient Descent" displays "Already confirmed" status and expands to reveal its Level 2 prerequisites: Derivative and Cost Function awaiting users' further confirmation. This inline expansion maintains context while revealing the prerequisite hierarchy, validating our approach to discovering knowledge gaps learners cannot identify independently.

c) Cross-Domain Dependencies: The knowledge graph in Figure 3 visualizes the prerequisite network for the backpropagation query. Node sizes decrease with depth (L0 to L3) while colors indicate assessment status: green for known concepts, red for unknown, and blue for unchecked. The graph reveals multi-level dependencies from immediate prerequisites like Gradient Descent (L1) through mathematical foundations like Differentiation (L2) to advanced concepts like Limits (L3), demonstrating the complex prerequisite web that our system systematically uncovers.

d) Personalized Learning Sequence: Figure 4 presents the generated learning path showing nested prerequisites for backpropagation. The hierarchical structure traces dependencies from L0 through L3, with unknown concepts marked with X and unassessed concepts marked with ?. For example, Gradient Descent (X) expands to reveal Derivative (X), which further requires Limits (?). This automatically generated sequence guides learners from mathematical foundations through progressive concepts to the target knowledge.

e) Personalized Explanation Generation: Figure 5 demonstrate the system's ability to generate tailored explanations based on the identified knowledge gaps. After recursive

TABLE I: Comparative Analysis of RPKT vs. Standard LLM (GPT-4o) Educational Explanations

Criterion	RPKT System	Standard LLM (GPT-4o)
Knowledge Gap Discovery	Actively discovers "unknown unknowns" through recursive prerequisite tracing without requiring question formulation	Assumes users can identify and articulate what they need to learn
User Interaction	Simple binary assessment (know/don't know) eliminates need for question generation	Requires users to formulate effective questions and prompts to identify gaps
Cognitive Burden	Minimal: users only make binary decisions, no question formulation needed	High: users must generate questions about concepts they don't understand
Pedagogical Strategy	Bottom-up approach; builds from identified knowledge boundaries to target concept	Top-down approach; explains target concept directly with occasional context
Content Structure	Dynamic hierarchical tree based on individual assessment results	Static linear or sectioned format regardless of user knowledge state
Prerequisite Handling	Systematically identifies and addresses missing prerequisites before main content	Mentions prerequisites briefly or assumes prior knowledge
Personalization Level	Personalized based on recursive knowledge state assessment	Generic explanations with same content for all users
Learning Path	Generates personalized prerequisite sequence tailored to individual gaps	Provides standard explanation without customized learning trajectory
Coverage Completeness	Systematically traces prerequisite dependencies to foundational concepts	May have gaps in foundational concepts depending on response scope
Best Use Case	Self-directed learning, knowledge gap identification, comprehensive understanding	Quick references, specific questions, users with clear learning objectives
Scalability	Domain-agnostic recursive algorithm designed for cross-subject application	Depends on training data coverage and prompt engineering quality

prerequisite discovery and assessment, RPKT creates a personalized tutorial that explicitly acknowledges the learner's existing knowledge ("Since you already understand forward propagation, derivatives, function notation, and loss functions") and thoroughly explains unknown concepts before connecting them to the target topic. The system first addresses each unknown prerequisite individually, then synthesizes these concepts to explain how they relate to backpropagation. This approach ensures learners build understanding progressively rather than encountering unexplained prerequisites within the main explanation.

f) Comparative Analysis: Table I contrasts RPKT with standard LLM tutoring approaches. The key distinction lies in knowledge gap discovery: RPKT actively uncovers unknown prerequisites without requiring question formulation, while standard LLMs assume users can articulate their learning needs. RPKT's bottom-up pedagogical strategy builds from identified knowledge boundaries, whereas standard LLMs provide top-down explanations that may assume prerequisite knowledge. This systematic approach addresses the fundamental challenge that learners cannot identify what they don't know they need to learn.

The demonstration confirms several key design choices. The recursive discovery successfully identifies multi-level prerequisite chains that span unexpected domains. The binary assessment interface enables efficient progression without cognitive overload. The visual representations effectively communicate both dependency depth and breadth. These findings validate that dynamic prerequisite discovery through recursive tracing provides a practical solution to the "unknown unknowns" problem in education.

V. DISCUSSION

The demonstration of RPKT validates our approach to addressing the "unknown unknowns" problem through recur-

sive prerequisite discovery. The system successfully identifies multi-level prerequisite chains spanning unexpected domains—revealing that understanding backpropagation requires foundational mathematics like calculus and linear algebra that learners would not independently identify.

Our approach is particularly valuable for learners beginning new disciplines or exploring interdisciplinary fields. Traditional learning often overwhelms beginners with complex explanations that assume prerequisite knowledge. RPKT systematically traces back to foundational concepts, preventing the frustration of encountering incomprehensible content. This makes the system especially suited for emerging interdisciplinary domains where prerequisites span multiple fields and traditional curricula may not exist.

The binary assessment interface, visual dependency graphs, and personalized explanations work together to transform learning from "what should I ask?" to simple decisions about current knowledge. By eliminating question formulation and revealing hidden prerequisites, the system enables efficient, comprehensive learning paths.

Currently, we present a demonstration system validating our design approach. The next critical step is experimental testing with real learners to measure learning outcomes and efficiency gains compared to traditional methods. Such empirical evaluation will provide insights into the system's educational effectiveness and guide further refinements.

ACKNOWLEDGMENT

Claude refined the writing to improve the overall readability.

REFERENCES

- [1] M. J. Nathan, K. R. Koedinger, M. W. Alibali *et al.*, "Expert blind spot: When content knowledge eclipses pedagogical content knowledge," in *Proceedings of the third international conference on cognitive science*, vol. 644648, 2001, pp. 644–648.

Personalized Explanation

To understand how backpropagation works in neural networks, we need to first build a foundation of several key concepts that you might not be familiar with. Since you already understand forward propagation, derivatives, function notation, and loss functions, we'll build from there, carefully explaining each of the other concepts before tying them together to describe backpropagation.

Gradient Descent

Gradient Descent is an optimization algorithm used to minimize the loss function in machine learning models. The goal is to find the set of parameters (weights and biases) that minimize the loss function, which measures how far off a model's predictions are from the actual outcomes.

Imagine you're walking down a hill and want to get to the bottom where the slope is zero. Each step you take is determined by the steepness or "gradient" of the hill at your current position. You take steps in the direction of the steepest descent.

- **Example:** If you're at a point on a curve and the slope is steep, you take a larger step downwards. If the slope is shallow, you take a smaller step.

Connecting Concepts to Backpropagation

Now that we have a foundation, let's connect these concepts to explain backpropagation:

1. **Forward Propagation:** In neural networks, you pass inputs through layers of nodes, applying weights, biases, and activation functions to produce an output.
2. **Calculate Loss:** The loss function computes the error between the predicted output and the true output.
3. **Backpropagation:** This process involves calculating the gradient of the loss function with respect to each weight by applying the chain rule. We work backwards from the output layer to the input layer to update the weights.
4. **Update Weights:** Using gradient descent, we adjust the weights in the opposite direction of the gradient to minimize the loss. The learning rate determines the size of each step.
5. **Iterate:** This process is repeated for many epochs (iterations) until the model converges to a minimum loss.

Fig. 5: Personalized explanation generation based on identified knowledge gaps. Top: System addresses unknown prerequisite (Gradient Descent) while acknowledging existing knowledge. Bottom: Synthesis connecting learned concepts back to the original backpropagation query.

- [2] P. J. Hinds, "The curse of expertise: The effects of expertise and debiasing methods on prediction of novice performance." *Journal of experimental psychology: applied*, vol. 5, no. 2, p. 205, 1999.
- [3] J. Kruger and D. Dunning, "Unskilled and unaware of it: how difficulties in recognizing one's own incompetence lead to inflated self-assessments." *Journal of personality and social psychology*, vol. 77, no. 6, p. 1121, 1999.
- [4] D. Dunning, "The dunning-kruger effect: On being ignorant of one's own ignorance," in *Advances in experimental social psychology*. Elsevier, 2011, vol. 44, pp. 247–296.
- [5] L. Deslauriers, L. S. McCarty, K. Miller, K. Callaghan, and G. Kestin, "Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom," *Proceedings of the National Academy of Sciences*, vol. 116, no. 39, pp. 19 251–19 257, 2019.
- [6] B. S. Bloom, "The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring," *Educational researcher*, vol. 13, no. 6, pp. 4–16, 1984.
- [7] P. Brusilovsky, "Adaptive hypermedia," *User modeling and user-adapted interaction*, vol. 11, no. 1, pp. 87–110, 2001.
- [8] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," *Advances in neural information processing systems*, vol. 28, 2015.
- [9] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 2330–2339.
- [10] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and S. Al-Emari, "A systematic review of automatic question generation for educational purposes," *International journal of artificial intelligence in education*, vol. 30, no. 1, pp. 121–204, 2020.
- [11] S. A. Karabenick, "Seeking help in large college classes: A person-centered approach," *Contemporary educational psychology*, vol. 28, no. 1, pp. 37–58, 2003.
- [12] J. Sweller, "Cognitive load theory," in *Psychology of learning and motivation*. Elsevier, 2011, vol. 55, pp. 37–76.
- [13] Q. Guo, J. Tang, W. Sun, H. Tang, Y. Shang, and W. Wang, "Soullmate: An application enhancing diverse mental health support with adaptive llms, prompt engineering, and rag techniques," *arXiv preprint arXiv:2410.16322*, 2024.
- [14] —, "Soullmate: An adaptive llm-driven system for advanced mental health support and assessment, based on a systematic application survey," *arXiv preprint arXiv:2410.11859*, 2024.
- [15] J. Tang, Q. Guo, W. Sun, and Y. Shang, "A layered multi-expert framework for long-context mental health assessments," *arXiv preprint arXiv:2501.13951*, 2025.
- [16] J. Tang and Y. Shang, "Advancing mental health pre-screening: A new custom gpt for psychological distress assessment," in *2024 IEEE 6th International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 2024, pp. 162–171.
- [17] Y. Yan, D. Qin, and E. E. Kuruoglu, "Llm online spatial-temporal signal reconstruction under noise," *arXiv preprint arXiv:2411.15764*, 2024.
- [18] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, and Y. Zhang, "Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge," *Cureus*, vol. 15, no. 6, p. e40895, 2023.
- [19] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl *et al.*, "Large language models encode clinical knowledge," *Nature*, vol. 620, no. 7972, pp. 172–180, 2023.
- [20] J. Cui, M. Ning, Z. Li, B. Chen, Y. Yan, H. Li, B. Ling, Y. Tian, and L. Yuan, "Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model," *arXiv preprint arXiv:2306.16092*, 2023.
- [21] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic, "Galactica: A large language model for science," *arXiv preprint arXiv:2211.09085*, 2022.
- [22] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.
- [23] S. Wu, O. Irsoy, S. Lu, V. Dabravolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, and G. Mann, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.
- [24] M. Moor, Q. Huang, S. Wu, M. Yasunaga, C. Zakka, Y. Dalmia, E. P. Reis, P. Rajpurkar, and J. Leskovec, "Med-palm m: A multimodal generalist medical ai system," *arXiv preprint arXiv:2307.14334*, 2023.
- [25] E. Kasneci, K. Seßler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günemann, E. Hüllermeier *et al.*, "Chatgpt for good? on opportunities and challenges of large language models for education," *Learning and individual differences*, vol. 103, p. 102274, 2023.
- [26] G. Abdelrahman, Q. Wang, and B. Nunes, "Knowledge tracing: A survey," *ACM Computing Surveys*, vol. 55, no. 11, pp. 1–37, 2023.
- [27] A. Tucker, *A model curriculum for k–12 computer science: Final report of the acm k–12 task force curriculum committee*. ACM, 2003.
- [28] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer science education*, vol. 13, no. 2, pp. 137–172, 2003.
- [29] C. Liang, J. Ye, S. Wang, B. Pursel, and C. L. Giles, "Investigating active learning for concept prerequisite learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.