Exploring Knowledge Tracing in Tutor-Student Dialogues using LLMs

Alexander Scarlatos
Manning College of Information and
Computer Sciences
University of Massachusetts Amherst
Amherst, Massachusetts, USA
ajscarlatos@umass.edu

Ryan S. Baker
Graduate School of Education
University of Pennsylvania
Philadelphia, Pennsylvania, USA
rybaker@upenn.edu

Andrew Lan
Manning College of Information and
Computer Sciences
University of Massachusetts Amherst
Amherst, Massachusetts, USA
andrewlan@cs.umass.edu

Abstract

Recent advances in large language models (LLMs) have led to the development of artificial intelligence (AI)-powered tutoring chatbots, showing promise in providing broad access to high-quality personalized education. Existing works have studied how to make LLMs follow tutoring principles, but have not studied broader uses of LLMs for supporting tutoring. Up until now, tracing student knowledge and analyzing misconceptions has been difficult and time-consuming to implement for open-ended dialogue tutoring. In this work, we investigate whether LLMs can be supportive of this task: we first use LLM prompting methods to identify the knowledge components/skills involved in each dialogue turn, i.e., a tutor utterance posing a task or a student utterance that responds to it. We also evaluate whether the student responds correctly to the tutor and verify the LLM's accuracy using human expert annotations. We then apply a range of knowledge tracing (KT) methods on the resulting labeled data to track student knowledge levels over an entire dialogue. We conduct experiments on two tutoring dialogue datasets, and show that a novel yet simple LLM-based method, LLMKT, significantly outperforms existing KT methods in predicting student response correctness in dialogues. We perform extensive qualitative analyses to highlight the challenges in dialogueKT and outline multiple avenues for future work.

CCS Concepts

• **Applied computing** → **Education**; • **Computing methodologies** → *Machine learning*; *Natural language processing*.

Keywords

Knowledge Components, Knowledge Tracing, Large Language Models, Tutoring dialogues

1 Introduction

Tutoring, often in the form of natural-language dialogues between tutors and students, has proven to effectively improve student learning outcomes, in both human tutoring [40] and dialogue-based intelligent tutoring systems [41]. Recent advances in generative artificial intelligence (AI), especially large language models (LLMs), have led to the development of generative AI-powered tutoring chatbots, such as Khan Academy's Khanmigo [26] and Carnegie Learning's LiveHint [5]. By wrapping LLMs around learning content such as textbooks and practice questions, these automated chatbots can interact with students in real time, potentially reaching a larger number of students than human tutors, while greatly reducing authoring efforts relative to previous-generation dialogue tutors [23].

To use LLMs as tutors, it is important to i) ensure they use effective pedagogical strategies and ii) control the safety of their output. Most existing LLM-powered tutoring chatbots are developed by either fine-tuning on actual tutor utterances in tutoring dialogue transcripts or prompt engineering. Along these lines, plenty of works over the last two years have studied how to make LLMs follow effective tutoring principles. The work in [59] studies the latent decision-making process behind human tutor utterances and shows that adopting decisions by human experts can benefit LLMs in tutoring. The work in [36] identifies tutor moves when they interact with an LLM-powered simulated student agent and shows that move annotation data can make LLMs become better tutors. The work in [19] generates large amounts of synthetic tutoring data via AI roleplay and shows that fine-tuning on this data, in addition to data collected from human tutors and students, makes LLMs more effective in pedagogy. The work in [15] proposes a way to measure tutor uptake, i.e., acknowledging what students have said, which is another beneficial strategy for LLMs to adopt. The work in [47] attempted to summarize tutor utterances into actionable feedback for students using LLMs, but found that the resulting messages were unable to outperform those written by real human teachers.

One important observation is that existing works involving LLMs have almost exclusively studied the *tutor turns* in dialogues rather than the *student turns*. A possible reason is that student turns can be noisy in nature, partially due to students not fully mastering the required knowledge, resulting in utterances that contain errors, which are harder to analyze than tutor utterances. Take Google's LearnLM [19] for example: student dialogue turns in their AI roleplay-based synthetic training data are generated by prompting an LLM to make an error that a real student would make. This error information is then fed to the tutor LLM to help the tutor model identify the

error and provide corresponding feedback. However, there is a key limitation to this approach: assuming LLMs can behave like real students may not be reliable. As recent work [16] shows, LLMs are ineffective at anticipating, parsing, and following flawed reasoning. More importantly, real open-ended student discourse is highly valuable since it may indicate flaws in student knowledge and even reveal specific misconceptions. Therefore, analyzing student dialogue turns can be a form of formative assessment to help tutors and AI-powered chatbots understand student progress and provide more targeted feedback. It can also potentially improve simulated student agents [25] to make them better at reflecting real student behavior, which can, in turn, improve the training of tutor chatbots through reinforcement learning [21, 48].

1.1 Contributions

In this work, we propose dialogue knowledge tracing (dialogueKT), a novel task, which analyzes student discourse within the knowledge tracing (KT) framework [45]. We enable KT on dialogues via a two-step process: we i) use proprietary LLMs, specifically GPT-4o [42], to annotate dialogues with student response correctness and knowledge component (KC) tags using Common Core standards, and ii) train KT models on the annotated data to estimate student knowledge at the dialogue-turn level. We introduce a novel LLM-based KT method, LLMKT, that leverages the textual content in dialogues, by fine-tuning the open-source Llama 3 LLM [20] on the KT objective. Using two existing tutor-student math dialogue datasets, we show that i) our new methods significantly outperform existing KT methods, especially when little training data is available, and ii) all KT methods can accurately predict student correctness given enough training data. We also use qualitative analysis and learning curve visualizations to show that LLMKT learns meaningful knowledge state estimates. Finally, we show that GPT-4o's dialogue annotations are accurate, according to human experts. Overall, by assessing student performance in more detail, we can make it possible to apply KT to LLM-human tutoring dialogues, enabling mastery learning and various other pedagogies and forms of detection afforded by KT [45].

We acknowledge many limitations of our work upfront since it is, to the best of our knowledge, the first at attempting to perform KT in open-ended tutoring dialogues. First, a "time step" in our KT setup corresponds to a single student turn in the dialogue, which is highly granular; we cannot explore setting an entire dialogue as a time step since no existing open-source large-scale tutoring dialogue dataset links students across dialogues. Second, our work is restricted to math since analyzing the KCs in math problems is challenging yet well-studied; it remains to be seen whether our approach of Common Core standard tagging generalizes to other subjects. Third, due to the smaller scale of tutoring dialogue datasets compared to those typically used in standard KT works, our methods need to be tested on larger-scale data. One dataset we use in our experiments contains partial dialogues that are cut short for LLM evaluation purposes [39], which further adds to this issue. Therefore, we publicly release our code[1] so that researchers who own a large amount of dialogue data can experiment on their proprietary data.

---

[1]https://github.com/umass-ml4ed/dialogue-kt

2 Related Work

2.1 Knowledge Tracing and Knowledge Component Labeling

Knowledge tracing (KT) [45] is a well-studied task in the student modeling literature. It breaks down student learning into a series of time steps practicing certain KCs and uses the correctness of each response step to track student knowledge. Classic Bayesian KT methods use a latent binary-valued variable to represent whether a student masters a KC or not. Factor analysis-based methods [8, 44] use features in addition to latent ability parameters in a logistic regression setting to predict student response correctness. Since neural networks became popular, plenty of deep learning-based KT methods were developed. These models have limited interpretability since student knowledge is represented as hidden states in complex, deep neural networks. Most of these methods use long short-term memory networks [46] or variants [53], coupled with with memory augmentation [64], graph neural networks [62], attention networks [17, 43], or pre-trained word embeddings in question text [31]. Many recent works have attempted to use LLMs in some ways for KT. Some use the neural network architecture of LLMs and train on student response data [63]. Others use LLMs to represent questions and responses in certain domains such as math [24], language learning [13, 55], and computer science education [27, 30]. More recently, [29] attempted prompting-based KT that directly asks LLMs to represent questions and student knowledge states as text. Despite some success in question response modeling, it remains to be seen whether these models can be useful in text-heavy KT situations such as dialogues.

There exist many prior works on automated KC tagging and discovery, including the Q-matrix method [3], which assigns KCs to questions using student response data. More recently, methods such as [52] also learn latent KCs by training deep learning models on KT data and enforcing priors from pedagogical theory. While these methods effectively learn new KCs, they do not produce textual descriptions of KCs, making them difficult to interpret. Several recent works have used LLMs to assign expert-written KCs to math problems via prompting [34, 61], fine-tuning [51], and reinforcement learning [28]. However, none have attempted assigning KCs in open-ended dialogues. A prior work has attempted to tag KCs in dialogue turns [12], although using a manual approach that cannot be scaled.

2.2 Dialogue Analysis

Existing works on dialogue analysis using LLMs have primarily focused on analyzing tutor behavior. The works in [14, 56] automatically classify the "talk moves" that tutors and students make via fine-tuned LLMs, and the work in [1] uses talk moves in dialogues to predict student performance in post-tests. In pre-LLM dialogue tutors, AutoTutor used latent semantic analysis to infer whether a student response was correct, and when incorrect, whether misconceptions were involved [4]. Why2-Atlas used deep syntactic analysis to convert a student response to a proof, analyzed the proof for errors represented as lack of knowledge [58], and later used Bayesian networks [38]. The Watson Tutor used ensembles of
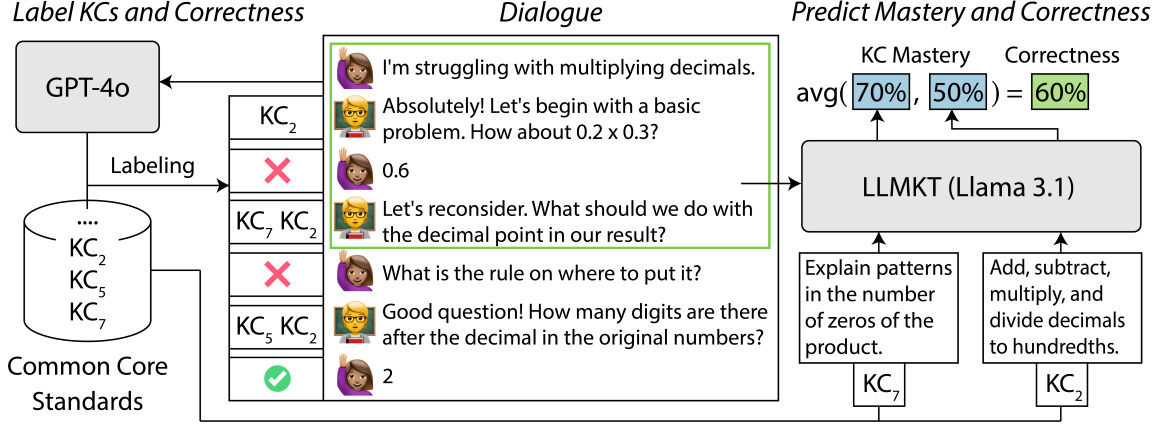
Figure 1: Overview of the dialogueKT framework and LLMKT. For annotation, we provide a full dialogue to GPT-4o, which generates correctness labels and retrieves Common Core standards as KCs for each turn. For KT, LLMKT takes a dialogue up to the target turn and its KCs to predict mastery on each KC and response correctness.

different classifiers on a range of textual features for correctness evaluation [7] and Glicko-2 [18] for KT.

3 Knowledge Tracing in Dialogues

We now formulate the dialogueKT framework. Our goal is to develop methods that can i) estimate student knowledge on KCs involved in each dialogue turn and ii) use these estimates to predict the correctness of student dialogue turns. Intuitively, we can view tutor-student dialogues as sequences of (formative) assessments. Visualized in Figure 1, some tutor turns can be viewed as posing a task that is related to some learning objectives and one or more KCs. They can reflect different tutor moves: asking a clarification question, identifying a student error, providing actionable feedback, prompting the student to think in a Socratic way, or offering encouragement [15, 57]. Student turns, on the other hand, contain correct or incorrect responses to tasks posed by the tutor, which reflect their thinking process and offer insights into their knowledge state. A pair of tutor-student dialogue turns corresponds to a time step in KT. Therefore, we split the dialogueKT framework into three stages: i) identify the KCs involved in each student dialogue turn, ii) classify whether the student response is correct or incorrect, and iii) apply new and existing KT methods on the resulting data.

Formally, we define each dialogue d to be a sequence of alternating tutor turns, t, and student turns, s, i.e., $d = (s_0, t_1, s_1, \ldots, t_M, s_M)$, where M is the total number of turn pairs for that dialogue and s_0 is present if the student initiates the dialogue. Each relevant turn pair is assigned a correctness label $y_j \in \{0, 1\}$, where $j \in \{1, \ldots, M\}$ is the turn pair index; $y_j = 1$ means that s_j is a correct response to t_j, and $y_j = 0$ means that it is incorrect. In practice, we also use $y_j = $ na in cases where correctness is not well-defined for t_j, e.g., the turn is off-topic or the tutor is offering emotional support. Each tutor turn is also associated with a set of KCs, $C_j = \{c_{j1}, \ldots, c_{jK}\}$, where K is the number of KCs for that turn. These KCs are related to learning objectives outlined by the tutor, which lists knowledge elements that the student must posses to respond correctly. In this work, we use the Common Core math standards [10] as the KCs

since they are i) standard and widely adopted and ii) textual in nature and are interpretable to both humans and LLMs. For irrelevant dialogue turns that we do not consider, $C_j = \emptyset$.

In a dialogue turn, we use $z_{jk} \in \{0, 1\}$ to denote the student's unobserved, binary-valued latent mastery status of the KC, c_{jk}, that is involved in the current step; $z_{jk} = 1$ means that the student masters KC k in dialogue turn j. In our framework, the student's mastery of all KCs involved in a turn determines whether they will respond correctly to the task posed by the tutor. Formally, the probability of a correct student response at turn j can be written as $P(y_j = 1 | z_{j1}, \ldots, z_{jK})$. The latent KC mastery depends on the entire dialogue history, i.e., $P(z_{jk} = 1) = \phi(c_{jk}, t_{\leq j}, s_{<j}, y_{<j}, C_{<j})$, where ϕ returns the probability of the student mastering KC c_{jk}, and subscripts $< j$ and $\leq j$ correspond to all past student and (including the current) tutor dialogue turns, respectively. In the next section, we detail how we parameterize and train this function in practice.

4 Methodology

We now detail our two primary technical contributions. First, we detail how to prompt pre-trained LLMs to automatically annotate dialogue turns to collect the KC and response correctness labels needed for dialogue knowledge tracing. Second, we propose new KT methods for the dialogueKT task that incorporate the rich textual information in tutor-student dialogues. We visualize the dialogueKT framework and the LLMKT method in Figure 1.

4.1 Automated Dialogue Annotation

In order to perform knowledge tracing on dialogues, each student conversation turn must first be annotated with two labels: i) the correctness of the student response, and ii) a list of KCs involved in the turn. While it is possible for human experts to provide these annotations, doing so would be incredibly time consuming and difficult to scale. Following recent works that have had success using LLMs to annotate data in educational tasks [50, 59], we use a state-of-the-art LLM, GPT-4o [42], to perform the annotation via

simple, zero-shot chain-of-thought (CoT) prompting[2]. We conduct the two annotation tasks separately since we find that doing so leads to improved accuracy compared to combining them. In Section 7, we conduct a human evaluation with math teachers to show that the GPT-provided annotations are mostly accurate and aligned with human judgment.

For correctness annotation, we provide the full dialogue as input to the LLM and instruct it to first summarize each turn in the dialogue in a CoT way, before labeling the correctness of each turn. We specifically instruct the LLM to "summarize" to leverage the fact it is pre-trained extensively on summarization tasks; this approach is inspired by recent work [35] that has shown that such an approach is effective for tasks such as code editing.

For KC annotation, we adapt the recursive tagging algorithm from [34], which tags math word problems with Common Core math standards. The algorithm uses hierarchically organized standards from the Achieve the Core[3] (ATC) coherence map, where standards are grouped into mid-level "clusters", which are further grouped into high-level "domains". To assign standards to a dialogue, we first provide the LLM with the full dialogue and the list of all 11 domains, and ask it to summarize the dialogue before selecting all domains that are relevant to the learning objectives. We then take the union of all clusters that are children of the selected domains and prompt the LLM again, this time asking it to summarize at the turn level but still select relevant clusters at the dialogue level. Finally, we take the union of all standards that are children of the selected clusters and prompt the LLM a final time, having it summarize each turn and assign standards at the turn level. We treat each Common Core math standard as a KC.

We note that by annotating a full dialogue with a single prompt, KT in real-time becomes impossible since information from future turns is available when annotating any given turn. This limitation could be alleviated in practice by annotating dialogues one turn at a time; however, such an approach is costly and we leave that for future work.

4.2 LLMKT

Another feature that makes dialogueKT different than the regular KT task is that most dialogue turns (at least 58%) are associated with more than one KC; see Table 1 for details. In contrast, most existing KT methods are developed for questions on a single KC; solutions that can deal with the occasional multi-KC question include i) converting a single question-response pair with multiple KCs to multiple pairs for different "questions", each covering a single KC, and ii) treating a unique combination of multiple KCs as a new KC. However, due to the prevalence of dialogue turns tagged with multiple KCs, we found that these solutions are not applicable to our setting. Therefore, we consider the student's likelihood of responding correctly to a tutor-posed task to be their average mastery of all KCs involved in the turn. Empirically, we found that averaging over KC masteries performed better than taking a product over them, consistent with findings in prior work [37]. Formally, we model the

probability of a correct student response at a turn as

$$P(y_j = 1) = \frac{1}{K} \sum_{k=1}^{K} P(z_{jk} = 1). \quad (1)$$

In other words, we use a compensatory model rather than a conjunctive model [37]. We now detail our proposed LLM-based KT method, which we refer to as LLMKT. One can, in theory, estimate a student knowledge state, z_{jk}, by simply providing the text of the dialogue up to t_j to a pre-trained LLM and prompting it to predict whether the student masters the KC, c_{jk}, for each $k \in [1, K]$. However, since LLMs are not pre-trained on the specific KT task, we cannot expect them to do well in this "zero-shot" setting. Therefore, we adopt a fine-tuning approach using Llama-3.1-8B-Instruct, an open-source, decoder-only LLM, for the KT task. We model the estimated probability of a student mastery $\hat{z}_{jk}$ as

$$\hat{z}_{jk} = P_\theta(z_{jk} = 1) = \frac{\exp(v_\theta^{\mathcal{T}}(t_{\leq j}, s_{<j}, c_{jk}))}{\sum_{\text{tok} \in \{\mathcal{T}, \mathcal{F}\}} \exp(v_\theta^{\text{tok}}(t_{\leq j}, s_{<j}, c_{jk}))}, \quad (2)$$

where θ represents the model parameters, and $v_\theta^{\mathcal{T}}(\cdot)$ and $v_\theta^{\mathcal{F}}(\cdot)$ return the output logits of the LLM for the "True" and "False" vocabulary tokens, respectively. In other words, with i) the entire conversation history and ii) a textual prompt instructing the LLM to estimate the student's current KC mastery level as input, we use an LLM to parameterize this estimate through two token generation probabilities. In practice, to improve model efficiency, we pack all KCs in C_j into a single prompt; we use customized attention masks and position embeddings to ensure that KCs do not attend to each other and their mastery levels are estimated independently. We do not explicitly provide KCs or correctness labels for previous turns due to memory constraints, but observe that LLMs can infer them from the context.

We fine-tune the LLM by maximizing the likelihood of all observed student response correctness labels, or equivalently, minimizing the binary cross entropy loss between the ground truth and predicted correctness values, y_j and $\hat{y}_j$, as

$$\hat{y}_j = \frac{1}{K} \sum_{k=1}^{K} \hat{z}_{jk} \quad (3)$$

$$\mathcal{L}(d; \theta) = -\sum_{j=1}^{M} y_j \log \hat{y}_j + (1 - y_j) \log(1 - \hat{y}_j). \quad (4)$$

While this method is relatively simple, it has many advantages. First, because we start with a powerful pre-trained LLM, LLMKT is able to leverage existing textual analysis capabilities that are not available in other KT architectures. Second, we do not add any new parameters to the pre-trained model, likely reducing the need for large amounts of training data. Third, since LLMKT operates on textual content rather than question/KC embeddings that need to be learned from scratch, it should be more generalizable to new KCs that are not seen during training. We note that LLMKT is similar to CLST [24], which trains an LLM to predict student correctness using a softmax over tokens (Eq. 2). However, unlike LLMKT, their method does not handle multiple KCs per item, which is critical to the dialogueKT task.

4.3 DKT-Sem

In addition to LLMKT, we experiment with an alternative, simpler KT method for dialogues, which can be viewed as a strong baseline. We slightly modify the deep KT (DKT) model [46] to use semantic embeddings of the textual content in dialogues, resulting in a

---

[2]We release our prompts in an online supplementary material: https://osf.io/873ms?view_only=f32472a6560649e692a4e2ba603ef52c
[3]https://achievethecore.org/

Dataset	Num. Dia.	Num. Labels	Correct	Num. KCs	Avg. KCs/Dia.	Avg. KCs/Turn	Turns w/ >1 KC
CoMTA	153	623	57.78%	164	3.20	1.94	58.75%
MathDial	2,823	13,200	49.86%	145	4.16	2.32	82.89%

Table 1: Statistics for the CoMTA and MathDial datasets, from left to right: total number of dialogues, total number of turn pairs with correctness labels, percentage of turn pairs where the student response is correct, total number of unique KCs in the dataset, average number of unique KCs per dialogue, average number of KCs per turn pair, and percentage of turn pairs with more than one KC.

method we term DKT-Sem. We map the text embeddings of every dialogue turn to the latent knowledge space in DKT as input, and map the text embeddings of KCs to the latent knowledge space to predict KC masteries. We believe that this simple method can result in better generalization across KCs and dialogues compared to DKT, which learns question and KC representations from scratch. Formally, we model KC mastery probability as

$$\mathbf{t}_j = \text{S-BERT}(t_j), \mathbf{s}_j = \text{S-BERT}(s_j), \mathbf{c}_j = \frac{1}{K}\sum_{k=1}^{K}\text{S-BERT}(c_{jk}) \quad (5)$$

$$\mathbf{y}_j = \text{emb}(y_j), \quad \mathbf{x}_j = \tanh([\mathbf{t}_j; \mathbf{s}_j; \mathbf{c}_j]\mathbf{W}) \oplus \mathbf{y}_j \quad (6)$$

$$\mathbf{h}_j = \text{LSTM}(\mathbf{x}_{\leq j}), \quad \hat{\mathbf{z}}_{j+1} = \sigma(\mathbf{h}_j \mathbf{B}\mathbf{C}^T). \quad (7)$$

The input at each time step is created by first computing the sentence BERT (S-BERT) [49] embeddings of the teacher turn, $\mathbf{t}_j$, student turn, $\mathbf{s}_j$, and element-wise average embeddings of the KCs, $\mathbf{c}_j$, respectively. Following DKT, response correctness is mapped to a learnable embedding, $\mathbf{y}_j$, which is then added element-wise to a transformed version (through $\mathbf{W}$ and tanh) of the concatenated tutor turn/student turn/KC embeddings, resulting in the input vector $\mathbf{x}_j$. Finally, the vector of estimated KC masteries at the next turn, $\hat{\mathbf{z}}_{j+1}$, are computed via a bilinear projection, $\mathbf{B}$, between the LSTM hidden state, $\mathbf{h}_j$, and the matrix of all KC S-BERT embeddings, $\mathbf{C}$, followed by the sigmoid function. We train DKT-Sem using the same objective as LLMKT (Eq. 4). Due to the smaller model size, DKT-Sem is easier to train than LLMKT, especially when access to computational resources to support LLM fine-tuning is limited.

5 Experiments

5.1 Datasets

We experiment with two datasets, CoMTA and MathDial, to study the dialogueKT task and validate the proposed KT methods. We select CoMTA since it is recent and highly reflective of AI-student dialogues that are becoming prevalent, and we select MathDial since it is relatively large-scale. We acknowledge that there are many other datasets available, such as the ones in [14, 57, 59] and ones not focusing on math education; our overall framework should be applicable to these as well. We show statistics for CoMTA and MathDial in Table 1.

The CoMTA dataset [39] contains 188 dialogues between human students and Khanmigo, Khan Academy's GPT-4-powered tutor. In most dialogues, students ask for help with a specific topic or problem(s) that they do not know how to solve, and work through one (or more) problems with the tutor throughout the dialogue. The tutor is primed with a series of customized prompts developed by Khan Academy to behave like real human tutors: to use the Socratic method, frequently asking questions at almost every dialogue turn, and to not directly reveal correct answers to students.

Each dialogue centers around a math subject among Elementary, Algebra, Trigonometry, Geometry, and Calculus. However, since Common Core does not contain Calculus standards, we remove Calculus dialogues, resulting in 153 remaining dialogues. Since no train/test split is defined, we perform a 5-fold cross-validation in experiments.

The MathDial dataset [36] contains 2,848 dialogues between students simulated by GPT-3.5 and crowd workers role-playing as human tutors. Each dialogue is in the context of an incorrect "student" solution to a math problem, and the tutor's goal is to help the student arrive at the correct solution by helping them identify and correct their errors. We use the original train/test split, containing 2,235/588 dialogues, respectively, after removing a total of 25 dialogues where GPT-4o's annotation of student response correctness and turn KCs failed.

5.2 KT Methods

In addition to LLMKT and DKT-Sem, we also evaluate the performance of a suite of widely-used KT methods on the dialogueKT task. Our goal is to test whether existing KT methods, developed for a slightly different use case, are applicable to our setting and how they compare to methods designed for dialogueKT.

We evaluate multiple popular deep learning-based KT methods: **DKT** [46], **DKVMN** [64], **AKT** [17], **SAINT** [9], and **simpleKT** [32], all sourced from the pyKT code repository [33]. There are two key differences between these existing KT methods and our proposed KT methods for dialogues: i) they learn new embeddings for questions, KCs, and response correctness without leveraging any textual information, and ii) they are designed to analyze questions that cover a single KC. Since most of the dialogue turns contain at least two KCs in the CoMTA and MathDial datasets, we convert each turn into K "pseudo-turns" that are each associated with one of the KCs, c_{jk}; all such pseudo-turns are associated with the same correctness label, y_j, of that turn. This modification follows common practice in the KT literature [17, 32]. We predict a single $\hat{\mathbf{z}}_j$ vector for each turn pair, using the output of the most recent time step that does not take y_j as input, to avoid leaking the correctness label across pseudo-turns. We note that predicting a vector instead of a single probability requires altering the size of the output projection for all methods except DKT. We train all methods using the same objective as LLMKT and DKT-Sem (Eq. 4). We also compare against Bayesian KT (**BKT**) [11], a classical KT method that uses a hidden Markov model to learn interpretable, per-KC learning acquisition parameters and use them to predict changing mastery levels for a student. Since BKT is designed to model data where each question is only assigned a single KC, we similarly expand

Method	CoMTA			MathDial		
	Acc.	AUC	F1	Acc.	AUC	F1
BKT	$51.02_{\pm 11.16}$	$52.50_{\pm 10.61}$	$53.41_{\pm 13.53}$	60.71	64.19	56.71
DKT	$52.59_{\pm 7.37}$	$53.20_{\pm 10.50}$	$49.56_{\pm 4.73}$	59.90	63.22	55.18
DKVMN	$47.22_{\pm 9.12}$	$46.81_{\pm 8.04}$	$48.80_{\pm 12.37}$	57.53	60.42	52.45
AKT	$49.95_{\pm 4.74}$	$51.40_{\pm 7.34}$	$50.78_{\pm 11.04}$	62.07	63.31	57.14
SAINT	$49.17_{\pm 4.88}$	$47.81_{\pm 3.89}$	$50.38_{\pm 6.68}$	55.97	60.10	54.24
simpleKT	$51.25_{\pm 4.05}$	$51.25_{\pm 5.60}$	$49.82_{\pm 10.45}$	59.24	63.83	53.85
DKT-Sem	$\underline{56.83}_{\pm 9.75}$	$\underline{61.82}_{\pm 6.56}$	$\mathbf{63.27}_{\pm 9.57}$	$\underline{62.17}$	$\underline{66.18}$	$\underline{59.30}$
LLMKT	$\mathbf{58.01}_{\pm 5.40}$	$\mathbf{65.79}_{\pm 4.44}$	$\underline{60.72}_{\pm 10.77}$	$\mathbf{68.41}$	$\mathbf{76.71}$	$\mathbf{62.21}$

Table 2: Student dialogue turn correctness prediction performance (averaged with standard deviation shown when applicable) for all KT methods, with best numbers in bold and second-best underlined. LLMKT outperforms all other methods on both datasets.

turns into "pseudo-turns" as described above. Due to the rigidity of BKT, we use a slightly different setup compared to other methods. At train-time, we train on all pseudo-turn labels, rather than aggregating at the true turn level. At test-time, we take the average pseudo-turn prediction to be the true turn prediction. In contrast to the other methods, this pseudo-turn level prediction setup does not leak correctness labels because information does not interact across KCs in BKT. We use pyBKT [2] to train and generate predictions.

5.3 Metrics and Experimental Setup

To quantitatively evaluate the effectiveness of KT methods in the dialogueKT task, we employ 3 widely used metrics: i) accuracy (**Acc.**), the portion of predicted labels that match the ground truth after rounding $\hat{y}_j$ to 0 or 1, ii) the area under the receiver operating characteristic curve (**AUC**), the most widely used metric in the KT literature as it accounts for class imbalance and all prediction thresholds, and iii) **F1**, the harmonic mean between binary precision and recall. We compute these metrics on all correctness labels except for the first label in each dialogue, since existing KT methods are unable to make educated estimates without prior labels in context.

We implement LLMKT using the Huggingface Transformers library [60] and perform fine-tuning with LoRA [22] on NVIDIA RTX A6000 GPUs. We implement DKT-Sem using PyTorch and use the `all-mpnet-base-v2` S-BERT model to generate text embeddings. During training, we randomly take 20% of the dialogues in the train set to use as a validation set and perform early stopping. We then evaluate all metrics on the test set, and repeat for all 5 folds on CoMTA.

Before computing final results, we perform a grid search to find optimal hyperparameters for each method, keeping the checkpoint with the highest AUC on the validation set. For LLMKT, we perform the search on the CoMTA dataset and use the same parameters on MathDial to reduce training time, and for other methods we use the best hyperparameter values found for each dataset. For LLMKT, we search over learning rate $\in \{5 \cdot 10^{-5}, 1 \cdot 10^{-4}, 2 \cdot 10^{-4}, 3 \cdot 10^{-4}\}$ and LoRA's rank $r \in \{4, 8, 16, 32\}$, and fix LoRA's α to 16, batch size to 64 using gradient accumulation, gradient norm clipping to 1.0, and train for 5 epochs. For other KT methods, we search over learning rate $\in \{1 \cdot 10^{-4}, 2 \cdot 10^{-4}, 5 \cdot 10^{-4}, 1 \cdot 10^{-3}, 2 \cdot 10^{-3}, 5 \cdot 10^{-3}\}$ and embedding size $\in \{8, 16, 32, 64, 128, 256, 512\}$, and fix batch size to 64, train for 100 epochs, and set all other model-specific

hyperparameters to the defaults in pyKT. For all methods, we train with the AdamW optimizer with a weight decay of $1 \cdot 10^{-2}$. We found the optimal learning rate and r of LLMKT to be $2 \cdot 10^{-4}$ and 16, respectively; the optimal hyperparameters for other KT methods varied across methods and datasets, but tended to have smaller embedding sizes on CoMTA and larger ones on MathDial since the former is much smaller in scale compared to the latter.

6 Experimental Results

6.1 Quantitative Results

Table 2 shows the quantitative performance of all KT methods on both datasets. We see that LLMKT significantly outperforms all existing KT methods and generally outperforms DKT-Sem on both datasets, confirming its effectiveness in the dialogue setting. This advantage over other KT methods is likely due to the use of a pretrained LLM to capture the rich textual content in dialogues. As a result, the LLM can estimate student knowledge after fine-tuning on student turn response correctness labels. We also see that DKT-Sem performs better than all existing KT methods, even though it largely uses the same model architecture as DKT. This result further emphasizes the benefit of utilizing the textual dialogue context in some way, even if the original model architecture of existing KT methods remains largely unchanged.

On CoMTA, we see that all existing KT methods almost completely fail, unable to outperform the simple baseline of predicting the majority class label (which corresponds to an Acc. of $57.83_{\pm 5.08}$, AUC of $50.0_{\pm 0.00}$, and F1 of $58.87_{\pm 29.70}$). This result is expected since the size of the CoMTA dataset is small and neural network-based KT methods are data-hungry. Another important factor is that there are 164 unique KCs across only 623 labeled dialogue turn pairs, which is not enough for these methods to properly learn the KC embeddings or for BKT to learn the per-KC parameters. In contrast, LLMKT needs only minimal training data to reach acceptable accuracy, due to the underlying LLM being highly effective at understanding previous student dialogue turns. It can also leverage the textual content of the KC definitions to generalize across KCs, which is especially beneficial for ones that do not occur frequently.

On MathDial, which has 21x more data than CoMTA, we see that existing KT methods perform significantly better (for reference, the majority baseline only gets an Acc. of 52.64 since class labels

Speaker	Turn Text	KCs and Predicted Masteries
Tutor	Point X is at (-4, 6) and point Z is at (6, -2). Determine the coordinates of point Y on line segment XZ such that XY is 3 times as long as YZ. What do you think is the important information here?	1) Use a pair of perpendicular number lines to define a coordinate system. `0.4688` 2) Apply the Pythagorean Theorem to find the distance between two points. `0.4688` 3) Recognize and represent proportional relationships between quantities. `0.5622` 4) Find the point on a line segment between two points that partitions the segment in a given ratio. `0.6225`
Student	Firstly, I need to determine the distance between point X and Z. That would be the square root of 164, which is roughly 12.8 ✓	
Tutor	Excellent beginning! You've calculated the total length of line segment XZ. Now, how can we use this information to determine the coordinates of point Y? Keep in mind, XY is 3 times as long as YZ.	1) Recognize and represent proportional relationships between quantities. `0.3208` 2) Find the point on a line segment between two points that partitions the segment in a given ratio. `0.3486`
Student	I would divide 12.8 by 3 total parts. ✗	

Table 3: An example dialogue with student turn correctness labels, KC labels, and LLMKT's predicted mastery level for each KC. LLMKT accurately predicts student correctness at both turns by leveraging the difficulty of the tutor-posed questions.

are balanced). This result is expected and shows that they can perform the dialogueKT task when given enough training data. While DKT-Sem still outperforms existing KT methods, it does so by a smaller margin than on CoMTA, indicating DKT-Sem has its greatest advantage when little training data is given. However, LLMKT still maintains a large advantage over all other KT methods. This result suggests that even on large-scale datasets, LLM inference over prior dialogue history and textual content of KCs can still significantly improve KT performance, even when labeled student turn correctness data is abundant.

Finally, we see that performance of all KT methods is relatively low on the dialogueKT task, with a maximum of around 76% AUC for student dialogue turn correctness prediction. This result is in stark contrast to results on the standard KT task, where state-of-the-art KT methods can surpass 80% AUC for question response correctness prediction on some datasets. This observation suggests that dialogueKT is a challenging task, due to various reasons: First, dialogues can be relatively short, giving very little context for KT methods to leverage. For example, dialogues in CoMTA are cut short on purpose, with only 4-5 labeled turn pairs on average, to serve the goal in [39] on evaluating how accurate LLMs are at math. Therefore, we call upon researchers who have access to dialogues with more turns to study these methods on their (possibly proprietary) data using our publicly available implementation. Second, student behavior in dialogues is more unpredictable compared to traditional KT, since there can be many diverse types of student responses to tutor turns. We believe that these challenges make dialogueKT a task worth studying in future work; it is likely that new KT methods need to be developed to improve performance on this task.

6.2 Qualitative Analysis: Case Study

We now detail findings from our qualitative analysis, where we seek to i) identify challenges in the dialogueKT task, ii) understand why LLMKT has an advantage over existing KT methods, and most importantly, iii) show how LLMKT can be useful to tutors in practice.

Therefore, we perform a case study using a dialogue in the test set of CoMTA.

We identify two primary challenges in the data that are unique to the dialogueKT task and are difficult for prior KT methods. First, the open-ended nature of dialogues means that student behavior is much less structured than in traditional assessment or practice settings. For example, when the tutor poses a question to the student, the student often i) does not answer directly but instead asks a clarifying question, ii) skips ahead to the final solution in the case of step-by-step problem solving, or iii) asks the tutor to give them a different problem. These student actions are inherently difficult to anticipate ahead of time since they define behavior beyond what can be captured by the KCs. As a result, there are many turns in these dialogues that are "irrelevant" to the KT task.

Second, since dialogues can be short (around 4 turn pairs on average in CoMTA), there can be little historical context to estimate student knowledge from. In many dialogues, students are proficient in one aspect of problem solving, responding to several questions correctly at first, but then struggle when the tutor moves on to a new KC. Without sufficiently long student history, it is difficult to estimate KC mastery levels for ones that are seen for the first time. This problem is partially due to existing datasets not linking students over multiple dialogues.

We find that, unsurprisingly, LLMKT's advantage over existing KT methods comes from leveraging nuanced details in the dialogue, rather than solely relying on previous correctness and KC labels. For the most part, changes in predicted mastery levels correlate with previous turn correctness; as a result, dialogues with many correct or incorrect turns in a row are easy to predict for all KT methods. LLMKT excels in cases when it relies on textual information, rather than correctness labels, to adjust its KC mastery estimates. Table 3 shows one such example: At the first turn, the tutor asks the student for relevant information regarding the problem, and the student responds correctly by applying the Pythagorean Theorem, which the model accurately predicts. However, after the tutor asks the student to find the coordinates of Y, the student incorrectly wants
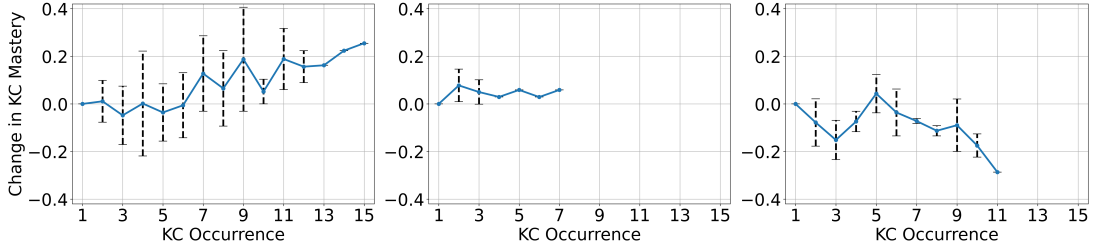
Figure 2: Knowledge change curves, i.e., estimated KC mastery level changes across dialogue turns, averaged over students for selected KCs in the CoMTA dataset. More common KCs in longer dialogues tend to show increasing mastery levels as the dialogue progresses, while less common KCs tend to show the opposite. KCs shown: "Solve linear equations and inequalities in one variable," "Understand that polynomials form a system analogous to the integers," and "Apply the properties of operations to generate equivalent expressions."

to divide the segment by 3 rather than 4, which the model also accurately predicts. The trained model seems to have estimated that the second tutor question is more difficult than the first, using the exact textual content of the tutor question, even though they have overlapping KC labels. This correct prediction is not possible without the question parsing ability enabled by using an LLM as the underlying KT model.

We also found that LLMKT tends to predict low mastery when a KC is seen for the first time in a dialogue, possibly through observing that students do not react well to the tutor shifting to new KCs mid-way through the dialogue. This observation also suggests that LLMKT can effectively use dialogue context to detect when new KCs are introduced. Overall, these observations suggest that there are numerous advantages in exploiting the textual content in dialogues for the dialogueKT task, rather than just relying the correctness and KC labels; LLMKT seems to be an encouraging step in this direction for future KT methods.

6.3 "Learning Curve" Analysis

A common method for assessing the quality of KCs and KT methods is examining how well they match pedagogical or cognitive theory, and in particular, the power law of practice [54]: as students practice a particular skill, their mastery will increase rapidly at first and then more slowly over time. Following prior work [52], we visualize changes in predicted mastery levels at the KC-level, examine trends in the curves, and investigate whether there is evidence of the power law of practice. We use LLMKT's predicted masteries on CoMTA for all dialogues when they are in the test split of the cross-fold experiment. We use CoMTA instead of MathDial for this analysis since the latter uses LLMs to simulate students; it is unlikely for simulated students to behave like real ones, which we found in a preliminary analysis.

For a KC c, we first find each time that KC occurs in the data, use $c_j^{(i)}$ to denote its jth occurrence in the ith dialogue, and use $\hat{z}_j^{(i)}$ to denote the corresponding predicted KC mastery level. We then subtract $\hat{z}_1^{(i)}$ from each $\hat{z}_j^{(i)}$ to track the *change* in predicted mastery from the first prediction, since different students may have different initial knowledge levels. We plot the average KC mastery level, i.e., $\frac{1}{N}\sum_{i=1}^{N}\hat{z}_j^{(i)} - \hat{z}_1^{(i)}$, where N is the number of dialogues that have

at least j turns that have c, and show the standard deviation using error bars.

We examine plots of the 15 most frequent KCs since less common KCs do not have enough data to show clear trends. Out of these, the results are mixed: 5 out of 15 KCs trend towards increasing mastery as the dialogues go on, 5 KCs trend towards decreasing mastery, and 5 KCs do not change significantly in mastery. Figure 2 shows a representative example of each case. KCs with an upward trend typically have few changes in their masteries for the first few turns, as well as relatively high variance. Additionally, these KCs are more prevalent in longer dialogues and correspond to the more commonly occurring KCs in the data. These patterns can be explained by two concurrent trends in the data: First, there are several long dialogues where a student struggles with a math topic at the beginning, getting some questions correct and others incorrect, but after some time becomes more proficient and consistently responds correctly; these dialogues contribute to the steady increase in KC mastery. Second, there are several short dialogues where the student either simply knows the topic and consistently responds correctly, or struggles with the topic and responds incorrectly; these dialogues contribute to the high variance and the minimal change in early turns.

On the contrary, KCs with a downward trend or no trend at all tend to be associated with shorter dialogues and occur in fewer dialogues. These dialogues tend to feature a student repeatedly trying to solve a problem without any success, with the dialogue often ending immediately after a correct solution is reached. KC mastery can also drop significantly at the beginning of the dialogue once the student starts to respond incorrectly. Since there are few dialogues to average over, these downward trends can dominate the knowledge change curves; the most common KC with a downward trend only appears in 8 dialogues, which is less than most KCs with upward trends.

Overall, when we take into account the small amount of data and how dialogues in the CoMTA dataset are truncated, the knowledge change curve trends seem to mostly resemble the power law of practice; when there are sufficient turns in a dialogue, the student's predicted mastery increases before leveling out eventually. The main reason these trends are not more prevalent is likely due to these dialogues typically ending after only a few turns. This setting is different than the problem-solving setup mostly used in past

Label	Score	Overlap	α	GPT-4o Final Turn Acc.	Human Final Turn Acc.
Correctness	0.9317 / 1	0.8434	0.1809	75.82 (83.33)	85.56
KCs	3.2831 / 4	0.3494	0.4383	–	–

Table 4: Scores and IRR metrics for the human evaluation. Additionally, final turn correctness accuracy for GPT-4o (on the full dataset followed by only on the human evaluation dataset) and the human annotators.

KT research, where students practice a KC repeatedly by solving different problems, even after mostly mastering it. Again, we call upon researchers who have access to data with longer dialogues to conduct more extensive analyses.

7 Annotation Quality Experiments

7.1 Human Evaluation

To quantify GPT-4o's accuracy on the correctness and KC labeling tasks, we conduct an IRB-approved human evaluation using expert human annotators. We hired 3 former math teachers at $40 US/hour, each rating the same set of 30 randomly selected dialogues from CoMTA with GPT-4o's labels, totaling 166 turn pairs. For each turn in each dialogue, we ask annotators to score GPT-4o's label i) on correctness, as 1 if they think the label is accurate and 0 otherwise, and ii) on the standards selected by GPT-4o on a 1-to-4 scale with 4 as the highest. We provide the exact instructions given to annotators in the online Supplementary Material[4]. For both correctness and KCs, we report the average human-assigned **Score** and two inter-rater reliability (IRR) metrics: i) the portion of times that the labels of all 3 annotators are exactly the same (**Overlap**) and ii) the Krippendorff's α coefficient [6]. We also compute the accuracy of correctness predictions at the final turn using CoMTA's ground truth labels for both GPT-4o and our annotators.

Table 4 shows the human evaluation results. We see that our annotators give GPT-4o very high scores for correctness and moderate-to-high scores for KCs, indicating that GPT-4o is able to competently perform these tasks. IRR is volatile, with high overlap and low α on correctness, and low overlap but moderate α on KCs. However, the low α on correctness is expected given the high class imbalance of the data (more than 90% of labels are 1), and the low overlap on KCs is expected since there are 4 classes and exact ratings can be subjective. While GPT-4o's accuracy on final turn correctness is lower than the correctness score might suggest, this is expected since the final turns are much harder to compute correctness on without subsequent tutor turns to help indicate the label. As reference, the CoMTA tech report [39] reports final turn correctness evaluation with GPT-4o at similar accuracy (78.3). Finally, given that the average human annotator's final turn correctness accuracy is only slightly higher than GPT-4o's, we can conclude that the correctness labeling task is a challenging one where GPT-4o achieves close to human-level performance.

7.2 Error Analysis

To understand when GPT-4o makes errors in labeling correctness and KCs, we qualitatively analyze its annotations. When labeling correctness, most of GPT-4o's errors are on the final turn, especially

---

[4]https://osf.io/wrkej?view_only=f32472a6560649e692a4e2ba603ef52c

in questions that require some numerical calculations to verify correctness, such as when the student provides the numerical result of a sum or product calculation. This is a known limitation in GPT-4/4o [50] and can possibly be improved through external tools such as calculators. The other common error tends to be when GPT-4o should label a turn as "na" but does not, e.g., when the tutor asks the student whether they know how to use a function on their calculator. GPT-4o also sometimes labels a turn as correct when the tutor asks a student if they know a concept and the student indicates they do not, showing that GPT-4o can be confused by the tutor's intentions.

When labeling KCs, GPT-4o's primary error is not assigning sufficient standards to a turn to describe all the required KCs. This error is mostly due to not selecting enough relevant domains and clusters early in the annotation process, especially for ones with broad or abstract descriptions. As a result, there are not enough related standards to select from to use as KCs. On the other hand, GPT-4o occasionally assigns too many KCs to a turn, selecting all standards that are related to the topic but not necessarily only ones relevant to the current question. These errors can likely be resolved with more sophisticated prompting, potentially running the annotation process multiple times through sampling and taking a majority vote. We leave the exploration of such methods that require more API calls for future work. Finally, there are several dialogues not covered by Common Core standards; there is little that GPT-4o can do other than select a somewhat related standard. This observation suggests that it may be necessary to augment the Common Core standards with other sources to have a more complete set of candidate KCs, or defer to humans for labeling in these cases.

8 Discussion, Conclusions, and Future Work

In this paper, we introduced the task of dialogueKT with the goal of estimating student knowledge from open-ended tutor-student dialogues. We used GPT-4o to automatically annotate student response correctness and relevant KCs in the dialogue turns, and showed via an expert human evaluation that these annotations are largely accurate. We introduced a new LLM-based KT method that leverages the textual content in dialogues, and showed that LLMKT i) significantly outperforms existing methods on dialogueKT and ii) works with limited training data. We also showed that our models use the textual content in dialogues to capture student knowledge in meaningful ways. There are numerous avenues for future work. First, we can jointly model tutor moves and student knowledge states to analyze what types of tutor moves are effective at improving student knowledge. Second, in addition to student knowledge, we also need to account for student affect dynamics in dialogues, possibly with the help of other data modalities such as audio and

video. Third, the scope of dialogueKT can be expanded upon: in this work, we only focus on math dialogues between a single student and tutor, but future work can extend to other subjects and dialogues involving multiple students. Fourth, there are many potential applications of dialogueKT that can be explored. Future work should investigate whether estimated knowledge states are useful for personalization, providing teacher support, or can even be used to train AI-based tutors by powering simulated student agents. Overall, we hope that this initial investigation can broaden the scope of student modeling to conversational settings, which have become of significant interest lately.

Acknowledgments

References

[1] Mark Abdelshiheed, Jennifer K. Jacobs, and Sidney K. D'Mello. 2024. Aligning Tutor Discourse Supporting Rigorous Thinking with Tutee Content Mastery for Predicting Math Achievement. In *Artificial Intelligence in Education*. Springer Nature Switzerland, Cham, 150–164.

[2] Anirudhan Badrinath, Frederic Wang, and Zachary Pardos. 2021. pyBKT: An Accessible Python Library of Bayesian Knowledge Tracing Models. In *Proceedings of the 14th International Conference on Educational Data Mining*. 468–474.

[3] Tiffany Barnes. 2005. The Q-matrix method: Mining student response data for knowledge. In *American association for artificial intelligence 2005 educational data mining workshop*. AAAI Press, Pittsburgh, PA, USA, 1–8.

[4] Zhiqiang Cai, Arthur C Graesser, Leah Windsor, Qinyu Cheng, David W Shaffer, and Xiangen Hu. 2018. Impact of corpus size and dimensionality of LSA spaces from Wikipedia articles on AutoTutor answer evaluation. *Journal of educational data mining* (2018).

[5] Carnegie Learning. 2024. LiveHint Overview. Online: https://support.carnegielearning.com/help-center/math/livehint/article/livehint-overview/.

[6] Santiago Castro. 2017. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure. https://github.com/pln-fing-udelar/fast-krippendorff.

[7] Maria Chang, Matthew Ventura, Jae-wook Ahn, Peter Foltz, Tengfei Ma, Tejas I Dhamecha, Smit Marvaniya, Patrick Watson, Cassius D'helon, Amy Wetzel, et al. 2018. Dialogue-based tutoring at scale: Design and Challenges. In *Practitioner and Industrial Track Proceedings of the 13th International Conference of the Learning Sciences*.

[8] Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. 2019. DAS3H: Modeling student learning and forgetting for pptimally scheduling distributed practice of skills. In *Proc. Int. Conf. Educ. Data Mining*. 29–38.

[9] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. 2020. Towards an Appropriate Query, Key, and Value Computation for Knowledge Tracing. In *Proceedings of the Seventh ACM Conference on Learning @ Scale* (Virtual Event, USA) *(L@S '20)*. Association for Computing Machinery, New York, NY, USA, 341–344.

[10] Common Core State Standards Initiative. 2024. Mathematics Standards. Online: https://www.thecorestandards.org/Math/.

[11] Albert Corbett and John Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-adapted Interact.* 4, 4 (Dec. 1994), 253–278.

[12] Ethan A Croteau, Neil T Heffernan, and Kenneth R Koedinger. 2004. Why are algebra word problems difficult? Using tutorial log files and the power law of learning to select the best fitting cognitive model. In *Intelligent Tutoring Systems: 7th International Conference, ITS 2004, Maceió, Alagoas, Brazil, August 30-September 3, 2004. Proceedings 7*. Springer, 240–250.

[13] Peng Cui and Mrinmaya Sachan. 2023. Adaptive and Personalized Exercise Generation for Online Language Learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 10184–10198.

[14] Dorottya Demszky and Heather Hill. 2023. The NCTE Transcripts: A Dataset of Elementary Math Classroom Transcripts. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*. 528–538.

[15] Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori B Hashimoto. 2021. Measuring Conversational Uptake: A Case Study on Student-Teacher Interactions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1638–1653.

[16] Nigel Fernandez, Alexander Scarlatos, Wanyong Feng, Simon Woodhead, and Andrew Lan. 2024. DiVERT: Distractor Generation with Variational Errors Represented as Text for Math Multiple-choice Questions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Miami, Florida, USA, 9063–9081.

[17] Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-Aware Attentive Knowledge Tracing. In *Proc. ACM SIGKDD*. 2330–2339.

[18] Mark E Glickman. 2012. Example of the Glicko-2 system. *Boston University* 28 (2012).

[19] Google. 2024. How generative AI expands curiosity and understanding with LearnLM. Online: https://blog.google/outreach-initiatives/education/google-learnlm-gemini-generative-ai/.

[20] Aaron Grattafiori and Others. 2024. The Llama 3 Herd of Models. arXiv:2407.21783

[21] Joy He-Yueya and Adish Singla. 2021. Quizzing Policy Using Reinforcement Learning for Inferring the Student Knowledge State. *International Educational Data Mining Society* (2021).

[22] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[23] Pamela Jordan, Carolyn Penstein Rosé, and Kurt VanLehn. 2001. Tools for authoring tutorial dialogue knowledge. In *Proceedings of AI in Education 2001 Conference*. IOS Press Amsterdam.

[24] Heeseok Jung, Jaesang Yoo, Yohaan Yoon, and Yeonju Jang. 2024. CLST: Cold-Start Mitigation in Knowledge Tracing by Aligning a Generative Language Model as a Students' Knowledge Tracer. arXiv:2406.10296

[25] Tanja Käser and Giora Alexandron. 2024. Simulated learners in educational technology: A systematic literature review and a turing-like test. *International Journal of Artificial Intelligence in Education* 34, 2 (2024), 545–585.

[26] Khan Academy. 2023. Supercharge your teaching experience with Khanmigo. Online: https://www.khanmigo.ai/.

[27] Unggi Lee, Jiyeong Bae, Dohee Kim, Sookbun Lee, Jaekwon Park, Taekyung Ahn, Gunho Lee, Damji Stratton, and Hyeoncheol Kim. 2024. Language Model Can Do Knowledge Tracing: Simple but Effective Method to Integrate Language Model and Knowledge Tracing Task. arXiv:2406.02893 [cs.CL]

[28] Hang Li, Tianlong Xu, Jiliang Tang, and Qingsong Wen. 2024. Knowledge Tagging System on Math Questions via LLMs with Flexible Demonstration Retriever. arXiv:2406.13885 [cs.CL]

[29] Haoxuan Li, Jifan Yu, Yuanxin Ouyang, Zhuang Liu, Wenge Rong, Juanzi Li, and Zhang Xiong. 2024. Explainable few-shot knowledge tracing. *arXiv preprint arXiv:2405.14391* (2024).

[30] Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. 2022. Open-ended knowledge tracing for computer science education. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

[31] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* 33, 1 (2019), 100–115.

[32] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. 2023. simpleKT: A Simple But Tough-to-Beat Baseline for Knowledge Tracing. In *The Eleventh International Conference on Learning Representations*.

[33] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Jiliang Tang, and Weiqi Luo. 2022. pyKT: A Python Library to Benchmark Deep Learning based Knowledge Tracing Models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[34] Li Lucy, Tal August, Rose E Wang, Luca Soldaini, Courtney Allison, and Kyle Lo. 2024. MathFish: Evaluating Language Model Math Reasoning via Grounding in Educational Curricula. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, Miami, Florida, USA, 5644–5673.

[35] Qianou Ma, Hua Shen, Kenneth Koedinger, and Sherry Tongshuang Wu. 2024. How to Teach Programming in the AI Era? Using LLMs as a Teachable Agent for Debugging. In *Artificial Intelligence in Education*. Springer Nature Switzerland, Cham, 265–279.

[36] 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*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 5602–5621.

[37] Cristina Maier, Ryan Baker, and Steve Stalzer. 2021. Challenges to applying performance factor analysis to existing learning systems. In *International Conference on Computers in Education*.

[38] Maxim Makatchev and Kurt VanLehn. 2007. Combining Bayesian Networks and Formal Reasoning for Semantic Classification of Student Utterances. In *Proceedings of the 2007 conference on Artificial Intelligence in Education: Building Technology Rich Learning Contexts That Work*. 307–314.

[39] Pepper Miller and Kristen DiCerbo. 2024. LLM Based Math Tutoring: Challenges and Dataset. https://doi.org/10.35542/osf.io/5zwv3

[40] Andre Nickow, Philip Oreopoulos, and Vincent Quan. 2020. *The Impressive Effects of Tutoring on PreK-12 Learning: A Systematic Review and Meta-Analysis of the Experimental Evidence*. Working Paper 27476. National Bureau of Economic Research.

[41] Benjamin D Nye, Arthur C Graesser, and Xiangen Hu. 2014. AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education* 24 (2014), 427–469.

[42] OpenAI. 2024. Hello GPT-4o. Online: https://openai.com/index/hello-gpt-4o/.

[43] Shalini Pandey and George Karypis. 2019. A self attentive model for knowledge tracing. In *Proc. Int. Conf. Educ. Data Mining*. 384–389.

[44] Philip Pavlik Jr, Hao Cen, and Kenneth Koedinger. 2009. Performance factors analysis–A new alternative to knowledge tracing. In *Proc. Int. Conf. Artif. Intell. Educ.*

[45] Radek Pelánek. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User modeling and user-adapted interaction* 27 (2017), 313–350.

[46] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Proc. NeurIPS*. 505–513.

[47] Prihar, Ethan and Others. 2023. Comparing different approaches to generating mathematics explanations using large language models. In *International Conference on Artificial Intelligence in Education*. 290–295.

[48] Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. 2016. Faster teaching via pomdp planning. *Cognitive science* 40, 6 (2016), 1290–1332.

[49] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[50] Alexander Scarlatos, Digory Smith, Simon Woodhead, and Andrew Lan. 2024. Improving the Validity of Automatically Generated Feedback via Reinforcement Learning. In *Artificial Intelligence in Education*. Springer Nature Switzerland, Cham, 280–294.

[51] Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Sean McGrew, and Dongwon Lee. 2021. Classifying math knowledge components via task-adaptive pre-trained BERT. In *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part I 22*. Springer, 408–419.

[52] Yang Shi, Robin Schmucker, Min Chi, Tiffany Barnes, and Thomas Price. 2023. KC-Finder: Automated Knowledge Component Discovery for Programming Problems. *International Educational Data Mining Society* (2023).

[53] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. 2021. Saint+: Integrating temporal features for ednet correctness prediction. In *11th Int. Learn. Analytics Knowl. Conf.* 490–496.

[54] George S Snoddy. 1926. Learning and stability: a psychophysiological analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology* 10, 1 (1926), 1.

[55] Megha Srivastava and Noah Goodman. 2021. Question Generation for Adaptive Education. In *Proc. Annual Meeting of the Association for Computational Linguistics and the International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 692–701.

[56] Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022. The TalkMoves Dataset: K-12 Mathematics Lesson Transcripts Annotated for Teacher and Student Discursive Moves. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*. 4654–4662.

[57] Abhijit Suresh, Jennifer Jacobs, Margaret Perkoff, James H Martin, and Tamara Sumner. 2022. Fine-tuning transformers with additional context to classify discursive moves in mathematics classrooms. In *Proc. Workshop on Innovative Use of NLP for Building Educational Applications*.

[58] Kurt VanLehn, Pamela W Jordan, Carolyn P Rosé, Dumisizwe Bhembe, Michael Böttner, Andy Gaydos, Maxim Makatchev, Umarani Pappuswamy, Michael Ringenberg, Antonio Roque, et al. 2002. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Intelligent Tutoring Systems: 6th International Conference, ITS 2002 Biarritz, France and San Sebastian, Spain, June 2–7, 2002 Proceedings 6*. Springer, 158–167.

[59] Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. 2174–2199.

[60] Wolf, Thomas and Others. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38–45.

[61] Kaiqi Yang, Yucheng Chu, Taylor Darwin, Ahreum Han, Hang Li, Hongzhi Wen, Yasemin Copur-Gencturk, Jiliang Tang, and Hui Liu. 2024. Content Knowledge Identification with Multi-agent Large Language Models (LLMs). In *Artificial Intelligence in Education*. 284–292.

[62] Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. 2020. GIKT: A Graph-based Interaction Model for Knowledge Tracing. In *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*.

[63] Bojun Zhan, Teng Guo, Xueyi Li, Mingliang Hou, Qianru Liang, Boyu Gao, Weiqi Luo, and Zitao Liu. 2024. Knowledge Tracing as Language Processing: A Large-Scale Autoregressive Paradigm. In *Artificial Intelligence in Education*. Springer Nature Switzerland, Cham, 177–191.

[64] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proc. Int. Conf. World Wide Web*. 765–774.