CLST: Cold-Start Mitigation in Knowledge Tracing by Aligning a Generative Language Model as a Students' Knowledge Tracer

Heeseok Jung[1], Jaesang Yoo[1], Yohaan Yoon[1], and Yeonju Jang[2]([✉])

[1] Classting Edutech Research, Seoul, Republic of Korea
{poco2889,jsyoo,eddie}@classting.com
[2] Korea University, Seoul, Republic of Korea
spring0425@korea.ac.kr

Abstract. Knowledge tracing (KT), wherein students' problem-solving histories are used to estimate their current levels of knowledge, has attracted significant interest from researchers. However, most existing KT models were developed with an ID-based paradigm, which exhibits limitations in cold-start performance. These limitations can be mitigated by leveraging the vast quantities of external knowledge possessed by generative large language models (LLMs). In this study, we propose cold-start mitigation in knowledge tracing by aligning a generative language model as a students' knowledge tracer (CLST) as a framework that utilizes a generative LLM as a knowledge tracer. Upon collecting data from math, social studies, and science subjects, we framed the KT task as a natural language processing task, wherein problem-solving data are expressed in natural language, and fine-tuned the generative LLM using the formatted KT dataset. Subsequently, we evaluated the performance of the CLST in situations of data scarcity using various baseline models for comparison. The results indicate that the CLST significantly enhanced performance with a dataset of fewer than 100 students in terms of prediction, reliability, and cross-domain generalization.

Keywords: intelligent tutoring system · knowledge tracing (KT) · personalized learning.

1 Introduction

As we enter the era of digital transformation, online learning is becoming increasingly prevalent worldwide. In this context, intelligent tutoring systems (ITSs) have become increasingly significant in the field of education [3]. These systems can significantly enhance student learning by utilizing technologies that predict academic achievement and dropout rates [12,49,8].

Among the various technologies used in ITSs, knowledge tracing (KT) has emerged as a key component that enables personalized learning [40]. To predict students' responses to new exercises, KT models their knowledge states by analyzing their past performance [52]. Understanding the students' knowledge

states helps in recommending appropriate learning materials and offering relevant feedback, thereby enhancing learning outcomes.

A variety of KT models have been proposed to meet these expectations, including both traditional and deep learning (DL)-based models [3]. In particular, the performance of DL-based KT models in terms of predicting knowledge states has notably improved with successive developments.

The majority of existing KT models are dominated by the ID-based paradigm. Although such ID-based models often perform well in predicting students' knowledge states, they share one limitation: prediction performance is significantly poor in cold-start situations [60]. As an issue stemming from scarce data on users or items in recommendation systems [39], the cold-start problem is regarded as an important issue in the KT field, occurring in the presence of insufficient data pertaining to certain students or problem-solving interactions [53]. When an educational institution or EdTech company that operates an online learning platform attempts to develop an ITS using a KT model for the first time, it is likely to encounter cold-start problems in the early stages. Consequently, educational institutions may encounter challenges in utilizing most existing KT models during the development of new ITSs.

Many studies have made efforts to alleviate the cold-start issue inherent to KT, primarily by utilizing various types of side information such as the language proficiency of learners [19], difficulty level of questions [25,59], and students' question-answering patterns [54]. However, these methods require effort to obtain additional information beyond each learner's problem-solving history, which may be difficult in the early stages of a service.

On the other hand, generative large language models (LLMs) demonstrate outstanding performance in a wide range of fields and downstream applications [63,18,22]. Generative LLMs, such as GPT, are auto-regressive language models that can generate text similar to human writing. These models contain large quantities of external knowledge and are able to extract high-quality textual features.

The development of KT models using external knowledge inherently possessed by generative LLMs opens the possibility of improving performance in cold-start situations as well as enhancing the domain adaptability of KT models. In a cold-start scenario, the scarcity of information from the target domain can be supplemented by using information from other domains as training data [7]. However, domain adaptation is challenging because most ID-based KT models currently in use are domain-specific. If a generative LLM is used to train a KT model with diverse domains, high performance can be achieved even in a small number of target domains owned by educational institutions.

Despite the potential of this approach, little research has been conducted on the applicability of generative LLMs to KT tasks. One such study was conducted [31]; however, this attempt failed to fully utilize the strengths of generative LLMs, only expressing exercise description information through IDs. Furthermore, their model relied solely on the number of correct and incorrect answers submitted by each learner, whereas it is also important to account for

the order in which each learner solves problems. As such, research on modeling KT through language processing remains insufficient. Given their capabilities, generative LLMs can be deployed to overcome the limitations of the ID-based paradigm. As a result, more research is needed to determine the efficacy of expressing KT in natural language using generative LLMs.

In this study, we propose a framework called "cold-start mitigation in KT by aligning a generative language model as a students' knowledge tracer (CLST)." We conducted experiments to determine whether a generative LLM aligned with KT could perform well in cold-start scenarios. Initially, we collected data for a variety of subjects, including mathematics, social studies, and science. The KT task was then framed as a natural language processing (NLP) task by expressing problem-solving data in natural language, and the formatted KT dataset was used to fine-tune the generative LLM. Finally, multiple experiments were conducted to investigate CLST's performance in cold-start situations. This study was conducted to address the following research questions (RQs):

- RQ1: In cold-start scenarios, does the proposed method demonstrate successful prediction performance?
- RQ2: How can exercises be effectively represented in natural language?
- RQ3: Does the proposed model provide a convincing prediction of learners' knowledge states?
- RQ4: Is the proposed model effective at predicting student knowledge in cross-domain scenarios?

The remainder of this paper is organized as follows: A literature review on KT and generative LLMs in personalized learning are presented in Section 2. Section 3 describes the methodology developed in this study. Section 4 presents and discusses the experimental results. Finally, Section 5 concludes the study.

2 Literature Review

2.1 Knowledge tracing

KT utilizes students' problem-solving histories to approximate their current knowledge states and subsequently predict their future responses. A student's problem-solving history can be represented as $X = \{(e_1, y_1), (e_2, y_2), \cdots, (e_t, y_t)\}$, where e_t corresponds to an exercise that has been solved at a specific time t, and y_t represents the correctness of the student's response to e_t.

KT models can be distinctly classified as traditional or DL-based [3]. Traditional approaches of KT include bayesian knowledge tracing (BKT) and factor analysis models. BKT [9] is a Markov-process-based approach that represents each student's knowledge state as a collection of binary values, whereas factor analysis approaches analyze the factors that affect students' knowledge states, such as attempt counts and exercise difficulty. Performance factor analysis (PFA) [36] and knowledge tracing machines (KTM) [47] are representative works on factor analysis.

The growth of online education has led to a substantial accumulation of problem-solving data, allowing DL-based KT models to reach outstanding performance. Such methods include recurrent neural network (RNN)-based [38,56], memory-augmented neural network (MANN)-based [58,2,20], transformer-based [34,15,57], and graph neural network (GNN)-based [30,32,55] models. In this study, the BKT [9] and factor analysis models [36,23] were chosen from the traditional KT models, and the RNN-based model [38], MANN-based model [58], and transformer-based model [15] were chosen as baseline models and used in the experiment.

2.2 Generative LLMs in personalized learning

Generative LLMs are pre-trained auto-regressive language models, such as GPT-3, capable of generating human-like text [37]. These probabilistic models serve as the foundation for natural language processing (NLP) techniques, which enable the processing of natural language using algorithms. The term 'large' denotes the extensive number of parameters each of these models contains, while the term 'generative' refers to a subset of LLMs designed for text generation. Generative LLMs are now at the core of a variety of applications, including summarization and translation, generally delivered through dialogue-like communication with the user [62].

In light of their massive potential, attempts are being made to utilize generative LLMs in the field of education, with ongoing studies actively exploring methods to facilitate personalized learning through the use of these models. To date, LLMs have exhibited effective performance in generating various educational materials, including multiple-choice questions [33], stories for reading comprehension assessments [5], and quizzes [11]. Additionally, studies have been carried out on adaptive curriculum design [42], automatic grading [28], and automated feedback generation for assignments [1] utilizing generative LLMs.

Although several studies have been conducted on the use of generative LLMs in personalized learning, few attempts have been made to apply them to the KT field, which is a core technology to support personalized learning. The aforementioned study conducted by [31] was limited in that it did not fully leverage the capabilities of generative LLMs, as exercises were simply represented using IDs. Moreover, the sequence of problem-solving by learners is crucial in the KT task, whereas the study in question only considered the number of correct and incorrect answers by each learner.

Studies on KT have been conducted in an NLP context, leveraging textual data from questions. For instance, exercise-enhanced recurrent neural network (EERNN) [44] leverages the text of each exercise to predict student responses utilizing the Markov property or attention mechanism. Word2vec [29] is used to transform the words of each exercise into word embeddings, and exercise embeddings are subsequently learned by a bidirectional long short-term memory (LSTM) network. Exercise-aware knowledge tracing (EKT) [24] is an extended version of EERNN, where a memory network is deployed to quantify the contribution of each exercise to student mastery on multiple knowledge components

(KCs). Relation-aware self-attention model for knowledge tracing (RKT) [35] leverages interaction data and the textual content of each exercise to compute an exercise relation matrix for KT. Furthermore, [46] employed the BERT [10] embedding similarities between textual data from exercises to construct a graph, thereby predicting student responses using a hierarchical GNN and two attention mechanisms.

However, these methods did not rely on generative language models, instead solely utilizing NLP to obtain exercise representations. In other words, there is still plenty of potential for further research on the effectiveness of generative-LLM-based KT models. In this study, we examined the efficacy of aligning generative LLMs to KT by expressing the KT task in natural language.

3   Methodology

3.1 Problem formulation

Letting E be a set of exercises. A studnet's responses to exercises in E, and records of one's problem-solving can be represented as the set $X_{1:t} = \{x_1, x_2, \cdots, x_t\}$, comprising tuples $x_t = (e_t, y_t)$. Here, $e_t \in E$ represents the exercise solved at a specific time step t, and $y_t \in \{0, 1\}$ represents the correctness of the answer recorded by the student at t. With this notation, we can formulate the KT task in the following way: Given a student's historical information $X_{1:t}$, KT aims to predict the probability of correctly solving a new exercise at $t + 1$. In the following equation, f denotes the KT model.

$$\hat{y}_{t+1} = f(X_{1:t}, e_{t+1}) \tag{1}$$

3.2 CLST: Cold-start mitigation in KT by aligning a generative language model as a students' knowledge tracer

3.2.1 KTLP formatting: Utilizing natural language as KT information carrier

A student's problem-solving history X must be transformed into textual sequences X^{text} using prompt templates for generative LLMs. We start by organizing 'Task Input' to instruct the model to determine whether the student will solve the target exercise based on their historical interactions, and then output a binary response of "Yes" or "No". Here, each exercise is represented by a textual description (e.g., name of KC), with interactions represented as tuples of (exercise description, correctness), connected by the '->' symbol.

Additionally, the binary label $y \in 0, 1$ is converted into a binary key answer word $y^{text} \in \{yes, no\}$ to organize the 'Task Output'. The aforementioned process allows us to frame the KT task as a language processing task. We call this process KT as language processing (KTLP) formatting, which can be formulated as follows:

$$X_{1:t}{}^{text} = g(X_{1:t}, e_{t+1}) \ , \ y_{t+1}{}^{text} = g(y_{t+1}) \tag{2}$$

Here, g denotes the KTLP formatting function. Figure 1 depicts a sample KTLP-formatted interaction.

> **Task Input**
>
> Here is a student's problem-solving history (Format : (knowledge component, student's response) :
>
> ('Ordering Negative Numbers', 'correct') → ('Properties of Quadrilaterals', 'correct') →
>
> ('Mental Multiplication and Division', 'wrong') → ('Midpoint Between Two Coordinates', 'wrong') →
>
> ('Parts of a Circle', 'wrong') → ('Adding and Subtracting Negative Numbers', 'wrong') →
>
> ('Factorising into a Single Bracket', 'wrong')
>
> Will the student solve the next problem 'division' correctly? Answer "Yes" or "No".
>
> **Task Output**
>
> No

Fig. 1. Sample KTLP-formatted interaction.

3.2.2 Response prediction with CLST

Upon taking the discrete tokens of $X_{1:t}{}^{text}$ as input, an LLM is used to generate the next token $\hat{y}_{t+1}^{text}$ as the output. This process can be formulated as follows:

$$l_{t+1} = M(X_{1:t}{}^{text}) \in R^V \tag{3}$$

$$p_{t+1} = Softmax(l_{t+1}) \in R^V \tag{4}$$

$$\hat{y}_{t+1}^{text} \sim p_{t+1} \tag{5}$$

where M denotes the LLM, V denotes the vocabulary size, and $\hat{y}_{t+1}^{text}$ denotes the next predicted token drawn from the probability distribution p_{t+1}.

However, the KT model's output should be a floating-point number $\hat{y}_{t+1} \in [0,1]$ rather than a discrete token $\hat{y}_{t+1}^{text}$. Therefore, following prior studies on point-wise scoring tasks (e.g., recommendation), we take the logits $l_{t+1} \in R^V$ and apply a bi-dimensional softmax over the logits of binary answer words. The KT prediction of LLM can thus be written as:

$$\hat{y}_{t+1} = \frac{exp\left(l_{t+1,\ yes}\right)}{exp\left(l_{t+1,yes}\right) + exp\left(l_{t+1,no}\right)} \tag{6}$$

3.2.3 Instruction fine-tuning using KTLP-formatted data with LoRA architecture

Since we generated both the task input and the task output in natural language, we can optimize the LLM by following the common instruction fine-tuning and causal language modeling paradigm. This process can be formulated as follows:

$$\max_{\Phi} \sum_{(x,y)\in Z} \sum_{k=1}^{|y|} log(P_{\Phi}(y_k|x, y_{<k})) \tag{7}$$

Here, x and y denote the KTLP-formatted input and output, respectively, Z denotes the training dataset, y_k represents the k-th token of y, $y_{<k}$ denotes the token preceding y_k, and Φ represents the parameters of M.

However, it is highly resource-intensive to fine-tune every parameter of an LLM. To utilize training resources efficiently, we instead adopted low-rank adaptation (LoRA) [16]. As a parameter-efficient fine-tuning method, LoRA incorporates smaller trainable matrices for each layer of the model through low-rank decomposition and the freezing of pre-trained model parameters, thereby enabling lightweight tuning. As a result, the original parameters can be preserved in a frozen state while additional information in the fine-tuning dataset is efficiently incorporated by training smaller matrices. The final fine-tuning process can be formulated as follows:

$$\max_{\Theta} \sum_{(x,y)\in Z} \sum_{k=1}^{|y|} log(P_{\Phi+\Theta}(y_k|x, y_{<k})) \tag{8}$$

Here, Θ represents the LoRA matrices, which are only updated during the fine-tuning process.

3.3 Dataset

To investigate the effectiveness of CLST, we collected data pertaining to mathematics, social studies, and science subjects. Table 1 lists detailed statistics for each dataset.

For mathematics data, we selected the open benchmark datasets NIPS34, Algebra05, and Assistments09 (Assist09), which are used as standard benchmarks for KT methods [45,57]. The NIPS34 dataset was sourced from the third and fourth tasks of the NeurIPS 2020 Education Challenge. Gathered from the Eedi platform, the data encompass students' problem-solving histories for multiple-choice math exercises [50]. The Algebra05 dataset was sourced from the EDM challenge of the 2010 KDD Cup, comprising the responses of 13- to 14-year-old students to algebra exercises [43]. The Assistments09 dataset contains student response data on math exercises collected from the ASSISTments platform in 2009–2010 [13].

For social studies and science, we gathered data from Classting AI Learning, an online education platform that offers learning materials in a range of academic subjects for K–12 students.

Table 1. Detailed statistics of datasets used in this study.

	NIPS34	Algebra05	Assist09	Social studies	Science
# interaction	1,382,727	607,014	282,071	4,215,373	5,484,647
# learners	4,918	571	3,644	40,217	45,211
# exercises	948	173,113	17,727	14,168	29,795
# KCs	57	112	123	40	76
Median interactions per learner	239	580	25	40	35
Median KCs per learner	26	54	5	3	3

3.4 Baseline models

In our comparative experiments, traditional, DL-based, and NLP-enhanced DL-based KT models were selected as baselines.

As traditional KT models, we selected the BKT, item response theory (IRT), and PFA models. BKT is a hidden Markov model that encodes learners' knowledge states with binary variables [9]. IRT models KT via logistic regression, accounting for each student's ability as well as the difficulty of each exercise [23]. PFA is another logistic regression model that incorporates exercise difficulty alongside each student's prior successes and failures [36].

As DL-based KT models, we selected deep knowledge tracing (DKT), dynamic key-value memory networks for knowledge tracing (DKVMN), and attentive knowledge tracing (AKT). DKT [38] is a standard RNN-based KT model that predicts students' knowledge states with a single LSTM layer [41]. DKVMN [58] is a KT model that uses static memory to encode latent KCs and dynamic memory to track student proficiency for the latent KCs [41]. AKT [15] is an attention-mechanism-based KT model that improves predictive performance by incorporating context-aware attention with Rasch model-based encoding.

We implemented NLP-enhanced DL-based KT models. These models initialize exercise embedding vectors in DL-based KT models with encoded text features of the KC name that corresponds to each exercise. which covers each exercise Specifically, we used OpenAI[1]'s *text-embedding-3-large* to encode text features, with the resulting KT models denoted as DKT_{text}, $DKVMN_{text}$, and AKT_{text}.

3.5 Experimental settings

To guarantee the quality of the data, only interactions from students who responded to more than five exercises were used.

Because the objective of this study was to mitigate problems that arise from cold-start situations, we alternately limited the number of learners in the training set to 64, 32, 16, and 8. We executed each method five times with different random seeds and reported the average outcomes. For the mathematics data, 20% of the learners were held out for the test set. For the social studies and science data, we selected 1000 random learners as the test set for each subject.

The selection of a base model requires careful consideration. Among existing generative LLMs, many do not provide accessibility to their model weights or APIs. Furthermore, data security is a critical concern in the field of education, necessitating additional discussion about the use of third-party APIs (e.g., OpenAI). After careful consideration, we selected the instruction-tuned Mistral-7B[2] [17].

Because it was essential to evaluate predictive performance with limited data, we only considered the first 50 interactions per student in the experiment [48]. For other baseline hyperparameters, we adhered to the settings described by [48,21], as their evaluation protocols were similar to our own.

---

[1] https://platform.openai.com/docs/guides/embeddings
[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2

4   Results and discussion

To evaluate the effectiveness of CLST from multiple perspectives, as well as address the four RQs, we conducted four experiments with the following objectives: 1) compare predictive performance between CLST and the baseline models in cold-start scenarios; 2) investigate the effectiveness of each component of the CLST through an ablation study; 3) analyze learning trajectories using CLST; and 4) compare predictive performance between CLST and the baseline models in cross-domain tasks. The following subsections discuss our experimental results.

4.1 Predictive performance under cold-start scenarios

We evaluated the predictive performance of CLST in cold-start scenarios involving a limited number of students, with the training set sequentially reduced from 64 to 8 students. All baseline models listed in Section 3.4 were chosen for comparison in this experiment. We measured the area under the receiving operator characteristic (AUC) as an evaluation metric. AUC is the most widely used indicator for evaluating KT performance, with higher scores indicating better performance [26].

The experimental results are presented in Figure 2, where traditional, DL-based, and NLP-enhanced models are represented by yellow, red, and purple curves, respectively, whereas CLST is represented by a green curve.

As observed from the figure, CLST outperformed every baseline model, irrespective of the size of the training set. Furthermore, models utilizing NLP demonstrated superior overall performance in comparison to traditional and DL-based methods.

Table 2 presents detailed quantitative results of the cold-start experiments, where the best AUC score for each trial is bolded and underlined, while the second-highest AUC score is underlined. CLST outperformed the second-best model by up to 24.52%, 14.66%, 12.31%, and 8.71% for a training set containing 8, 16, 32, and 64 students, respectively. Furthermore, when comparing performance between CLST and the baseline models across different datasets, CLST exhibited improvements of up to 13.69% on the NIPS34 dataset, 5.62% on the Algebra05 dataset, 24.52% on the Assist09 dataset, 9.62% on the Classting Social Studies dataset, and 11.82% on the Classting Science dataset.

We can therefore provide the following response to RQ: In all datasets and all cold-start scenarios, the proposed method outperformed the baseline KT models; therefore, we can conclude that the cold-start problem in KT has been mitigated.

The cold-start problem has been a major concern in the field of KT [60,53]. Previous studies have suggested methods that utilize additional information from the exercise side (e.g., relations among KCs) or the learner side (e.g., language proficiency) to address this issue [19]. However, these methods require effort to acquire additional information that extends beyond each student's problem-solving history, which may be difficult to collect during the initial stages of service. In contrast, we suggest a novel approach to mitigating the cold-start issue
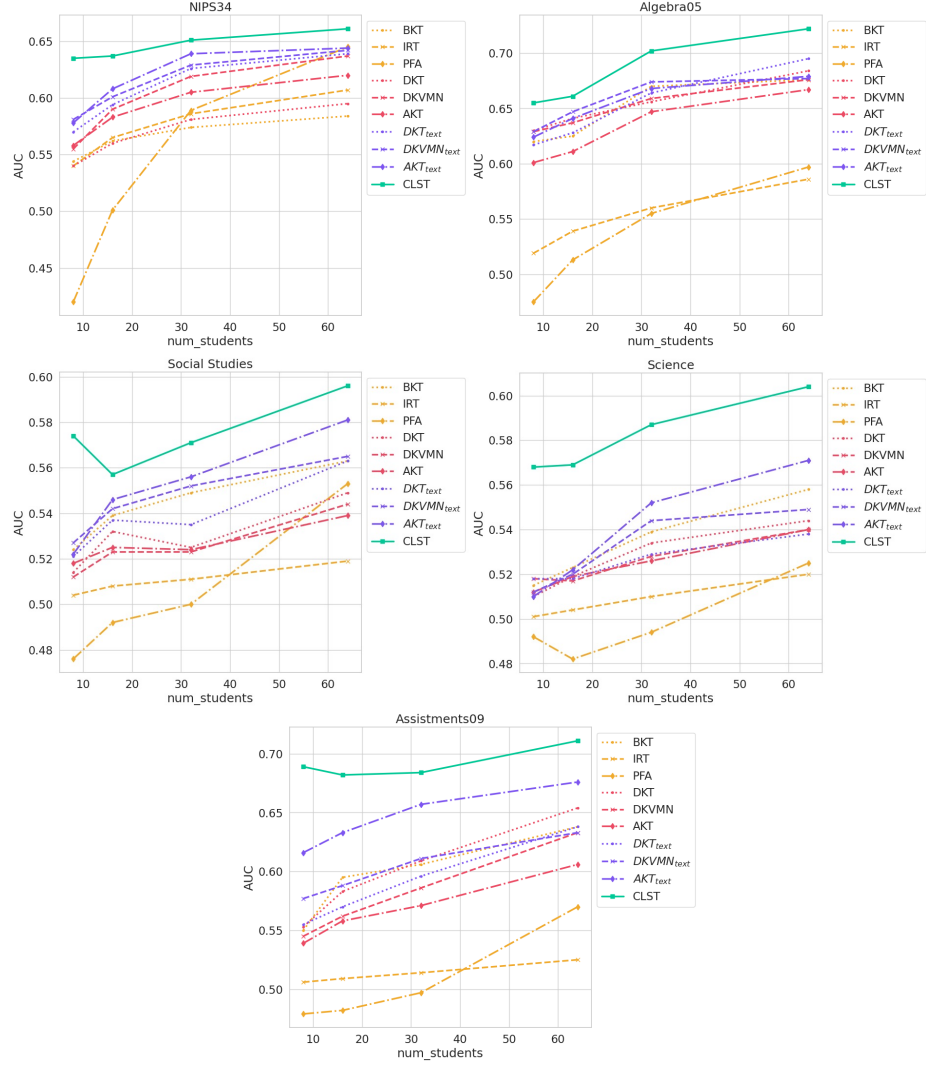
Fig. 2. Cold-start performance on each dataset.

Table 2. Detailed results of cold-start experiments.

Dataset	# Students	Traditional				DL-based		NLP-enhanced			CLST
		BKT	IRT	PFA	DKT	DKVMN	AKT	DKT_{text}	$DKVMN_{text}$	AKT_{text}	
NIPS34	8	0.544	0.540	0.420	0.540	0.555	0.558	0.570	0.581	0.578	**0.635**
	16	0.562	0.565	0.501	0.560	0.590	0.583	0.594	0.601	0.608	**0.637**
	32	0.574	0.586	0.589	0.581	0.619	0.605	0.626	0.629	0.639	**0.651**
	64	0.584	0.607	0.645	0.595	0.637	0.620	0.639	0.642	0.644	**0.661**
Algebra05	8	0.620	0.519	0.475	0.630	0.629	0.601	0.617	0.629	0.624	**0.655**
	16	0.625	0.539	0.513	0.641	0.637	0.611	0.628	0.647	0.641	**0.661**
	32	0.670	0.560	0.555	0.656	0.659	0.647	0.664	0.674	0.668	**0.702**
	64	0.676	0.586	0.597	0.684	0.676	0.667	0.695	0.677	0.679	**0.722**
Assist09	8	0.550	0.506	0.479	0.553	0.545	0.539	0.555	0.577	0.616	**0.689**
	16	0.595	0.509	0.482	0.583	0.562	0.558	0.570	0.588	0.633	**0.682**
	32	0.606	0.514	0.497	0.609	0.586	0.571	0.596	0.611	0.657	**0.684**
	64	0.638	0.525	0.570	0.654	0.633	0.606	0.638	0.633	0.676	**0.711**
Social Studies	8	0.524	0.504	0.476	0.514	0.512	0.518	0.521	0.527	0.522	**0.574**
	16	0.539	0.508	0.492	0.532	0.523	0.525	0.537	0.542	0.546	**0.557**
	32	0.549	0.511	0.500	0.525	0.523	0.524	0.535	0.552	0.556	**0.571**
	64	0.563	0.519	0.553	0.549	0.544	0.539	0.563	0.565	0.581	**0.596**
Science	8	0.515	0.501	0.492	0.510	0.518	0.512	0.518	0.512	0.510	**0.579**
	16	0.523	0.504	0.482	0.519	0.517	0.519	0.518	0.520	0.522	**0.580**
	32	0.539	0.510	0.494	0.534	0.528	0.526	0.529	0.544	0.552	**0.587**
	64	0.558	0.520	0.525	0.544	0.540	0.540	0.538	0.549	0.571	**0.604**

Average AUC

by conceptualizing KT as language processing and leveraging the capabilities of generative LLMs.

4.2 Ablation study

To investigate the effectiveness of each component of CLST in addressing the cold-start issue, we conducted two experiments with the following objectives: (1) comparison of performance with respect to exercise representations (2) to evaluate the effectiveness of fine-tuning on model's reliability.

4.2.1 Comparison of performance with respect to exercise representations

Most existing KT models represent exercises with their distinctive identity values, or IDs. In the experiments conducted in Section 4.1, all models representing exercises using textual descriptions outperformed those using ID-based representations. In particular, the proposed CLST, which expresses all information in natural language, exhibited the highest performance in each cold-start scenario.
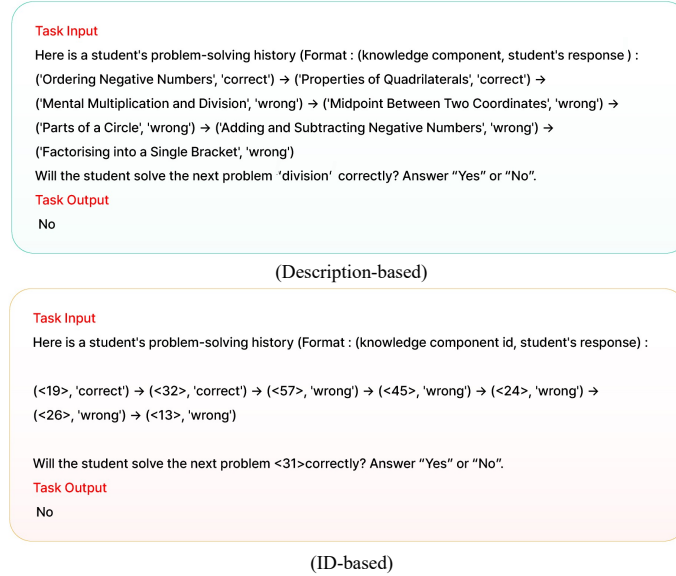
Task Input

Here is a student's problem-solving history (Format : (knowledge component, student's response) :

('Ordering Negative Numbers', 'correct') → ('Properties of Quadrilaterals', 'correct') →

('Mental Multiplication and Division', 'wrong') → ('Midpoint Between Two Coordinates', 'wrong') →

('Parts of a Circle', 'wrong') → ('Adding and Subtracting Negative Numbers', 'wrong') →

('Factorising into a Single Bracket', 'wrong')

Will the student solve the next problem 'division' correctly? Answer "Yes" or "No".

Task Output

No

(Description-based)

Task Input

Here is a student's problem-solving history (Format : (knowledge component id, student's response) :

(<19>, 'correct') → (<32>, 'correct') → (<57>, 'wrong') → (<45>, 'wrong') → (<24>, 'wrong') →

(<26>, 'wrong') → (<13>, 'wrong')

Will the student solve the next problem <31>correctly? Answer "Yes" or "No".

Task Output

No

(ID-based)

Fig. 3. Examples of the two methods for representing exercises.

We therefore conducted additional experiments to compare two ways of representing exercises in KTLP format: a description-based method that represents each exercise with the corresponding KC name, which is used in CLST, and the conventional ID-based approach, wherein the ID of the corresponding KC is used to represent each exercise. In this method, the ID of each KC is defined using

brackets in the format '<id>' and added as extra vocabulary. Figure 3 presents sample representations of an exercise using both approaches.

Table 3. Comparison of predictive performance (AUC) based on exercise representation method.

Dataset	Method	# Students			
		64	32	16	8
NIPS34	ID-based	0.658	0.646	**0.638**	0.629
	Description-based	**0.661**	**0.651**	0.637	**0.635**
Algebra05	ID-based	0.659	0.631	0.548	0.584
	Description-based	**0.722**	**0.702**	**0.661**	**0.655**
Assist09	ID-based	0.691	0.661	0.661	0.668
	Description-based	**0.711**	**0.684**	**0.682**	**0.689**
Social Studies	ID-based	**0.600**	0.541	0.537	0.567
	Description-based	0.596	**0.571**	**0.557**	**0.574**
Science	ID-based	0.599	0.584	0.577	0.570
	Description-based	**0.604**	**0.587**	**0.580**	**0.579**

Table 3 presents a comparison of predictive performance with respect to exercise representation. In most cases, the description-based method was observed to outperform the ID-based method.

Based on extensive external knowledge acquired during the pre-training process, the generative LLM is enabled to perform the KT task by considering the relationships between KCs. In general, KT models that account for relationships between KCs are more effective at predicting student performance [6,27]. As a result, when an exercise is represented based on its description, the external knowledge of generative LLMs can be fully utilized, resulting in improved prediction performance. Thus, the results of this experiment can be used to answer RQ2 as follows: When aligning generative LLM with KT, it is more effective to represent each exercise using a description-based method in cold-start situations.

4.2.2 Effectiveness of fine-tuning on model's reliability

To investigate the effects of fine-tuning with KTLP-formatted data, we compared the calibration plot outputs from an untuned model (*mistral-7b-instruct-v02*, the backbone model selected for this study) and its fine-tuned counterpart.

Figure 4 shows the changes in output calibration as an effect of fine-tuning for each dataset. Each calibration plot is an illustration that compares the average predicted value of a model's output for each bin (x-axis) with the fraction of positive classes (y-axis). A diagonal dotted line in the figure represents the ideal calibration that a model can achieve by predicting events based on the base rate of occurrences [14]. Therefore, the output values are more reliable when the output calibration is closer to the diagonal line. Overall, the untuned model exhibited poor output calibration; in contrast, output calibration significantly
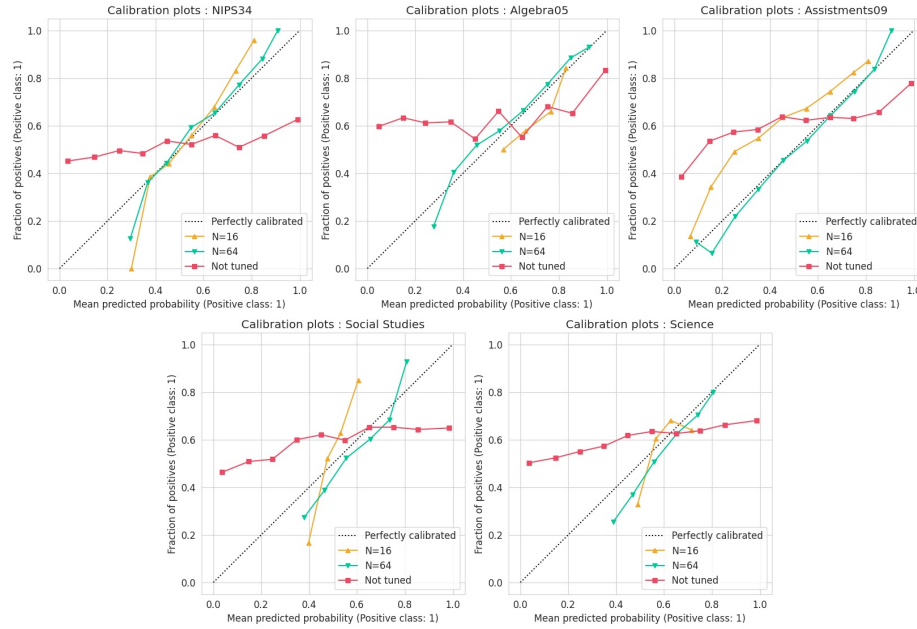
Fig. 4. Changes in output calibration as an effect of fine-tuning.

improved as the number of learners in the KTLP dataset used for fine-tuning increased to 16 and 64.

The output of a KT model is the probability of correctly answering an exercise that covers a specific KC. This probability is often regarded as the mastery level of that KC [38]. The output of a KT model can also be used for downstream tasks, such as personalized content recommendations [4], making output reliability an essential priority [21]. The results of this experiment demonstrate enhanced reliability when a model is fine-tuned using KTLP-formatted data.

4.3 Learning trajectory analysis

To determine whether the method proposed in this study successfully predicts students' knowledge states, we visually analyzed the students' understanding of each KC in the process of solving exercises. Random selections of students from the science, social studies, and mathematics datasets were made for this purpose. Among several datasets, students for the mathematics subject were selected at random from Assistments09.

Figures 5 to 8 depict the learning trajectories of the students selected for each subject. Heat maps and line graphs were used to visualize how the mastery level for each KC changed as each student progressed through solving problems for each subject.

The x-axis of each heat map represents the interaction (KC id, correctness) at each time step, while the y-axis represents the KC of the questions that the

student primarily answered. In other words, the value of each cell indicates the change in the mastery level for the corresponding KC after the interaction occurred. The x-axis of the line plot (lower part of Figures) likewise represents the interaction (KC id, correctness) at each time step, with the y-axis representing mastery level values; thus, each line represents the mastery level of the corresponding KC.
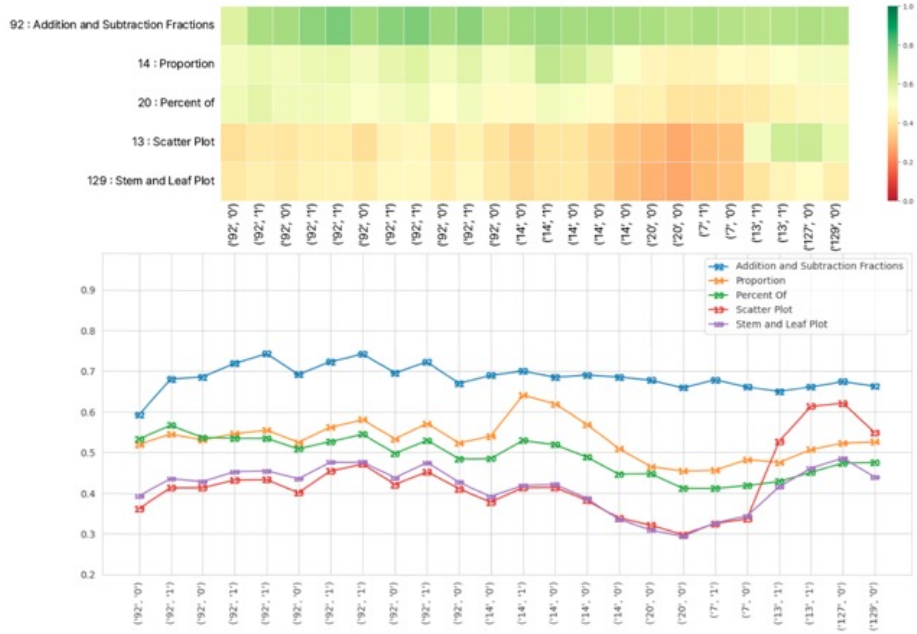

Fig. 5. Learning trajectories of student 'A' in mathematics

The learning trajectory analysis revealed two observed facts. Initially, each student's knowledge state predicted by CLST was subject to change in accordance with the correctness of the problem. In other words, when a student correctly or incorrectly answers a question about a specific KC, the mastery level of the corresponding KC increases or decreases, respectively. For example, in Figure 5, if the learner correctly answered the question about KC-92, the mastery level for the corresponding KC increased, whereas if the learner answered the question incorrectly, the mastery level decreased. Similarly, when questions about KC-14 were correctly and incorrectly answered, the mastery level of the corresponding KC (orange line in the figure) rose and fell accordingly.

Second, the CLST predicts each learner's knowledge state by considering relationships between KCs. For example, Figure 5 depicts a math student's learning trajectory, showing changes in mastery levels for KC-13 (scatter plot), KC-14 (proportion), KC-20 (percent of), KC-129 (stem and leaf plot), and KC-92 (frac-

Fig. 6. Learning trajectories of student 'B' in mathematics

tion addition and subtraction). Among these, KC-13 and KC-129 are both concepts related to plots, and KC-14 and KC-20 are similarly interconnected. Upon examining changes in the learners' mastery levels, the trends within these pairs of KCs exhibited mutually similar tendencies. In contrast, the mastery level for KC-92 exhibited a different pattern compared to that of the other KCs.

Furthermore, when a learner correctly responded to an item regarding KC-13, the corresponding mastery level increased along with that of the intuitively related KC-129. Consequently, it is possible to infer that CLST estimates the learner's knowledge state by examining relationships between KCs.

Similarly, in the learning trajectory of another math student shown in Figure 6, the mastery levels of related KCs, namely KC-25 (division of fractions) and KC-92 (addition and subtraction of fractions), as well as KC-20 (percent of) and KC-37 (conversion of fractions, decimals, and percents), displayed similar trends.

In the social studies learning trajectory shown in Figure 7, the mastery level of related concepts KC-2 (national economy) and KC-39 (economic life and choices) changed in a similar manner. For the science learner shown in Figure 8, the mastery levels of related KCs—KC-59 (composition of earth materials) and KC-60 (history of the earth)—changed in similar patterns.
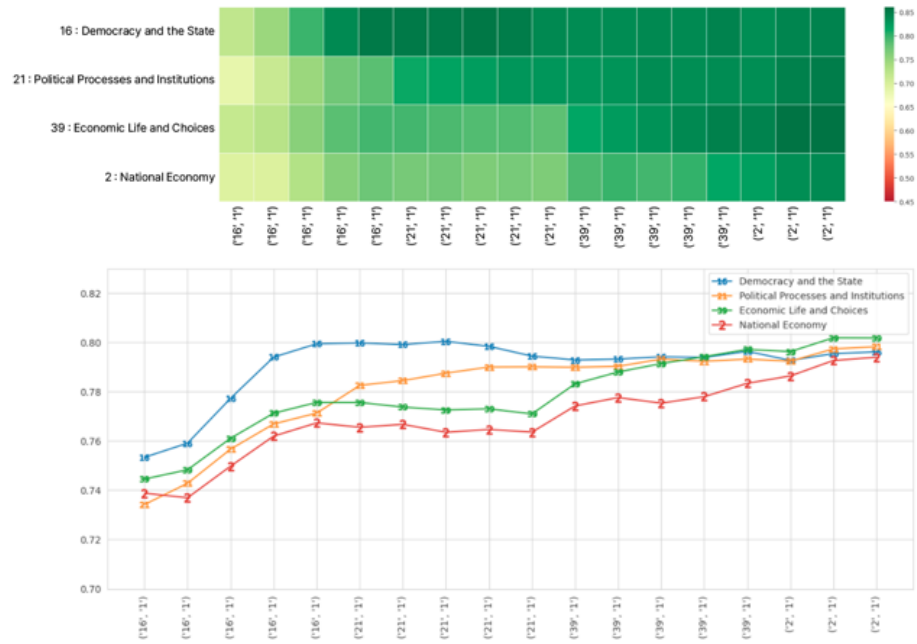
Fig. 7. Learning trajectories of student 'C' in social studies

Fig. 8. Learning trajectories of student 'D' in science

Ultimately, the analysis of learning trajectories yields the following answer to RQ3: The CLST plausibly predicts each student's mastery level and also understands the relationships between KCs.

4.4 Predictive performance under cross-domain scenarios

Finally, we examined the performance of CLST in a cross-domain scenario. In this experiment, the terms 'domain' and 'dataset' are used interchangeably [61]. To evaluate cross-domain performance, we selected two or more datasets for the same subject. Specifically, we conducted an experiment using the mathematics subject datasets NIPS34, Algebra05, and Assistments09.

We tuned CLST with samples from one source domain and subsequently evaluated its predictive performance on test samples from another target domain. For example, we evaluated the performance of the CLST tuned with samples from the NIPS34 dataset (referred to as CLST(nips) in the figure below) on the Algebra05 dataset. In addition, we evaluated the performance of DL-based KT models and the CLST trained with the target domain samples.
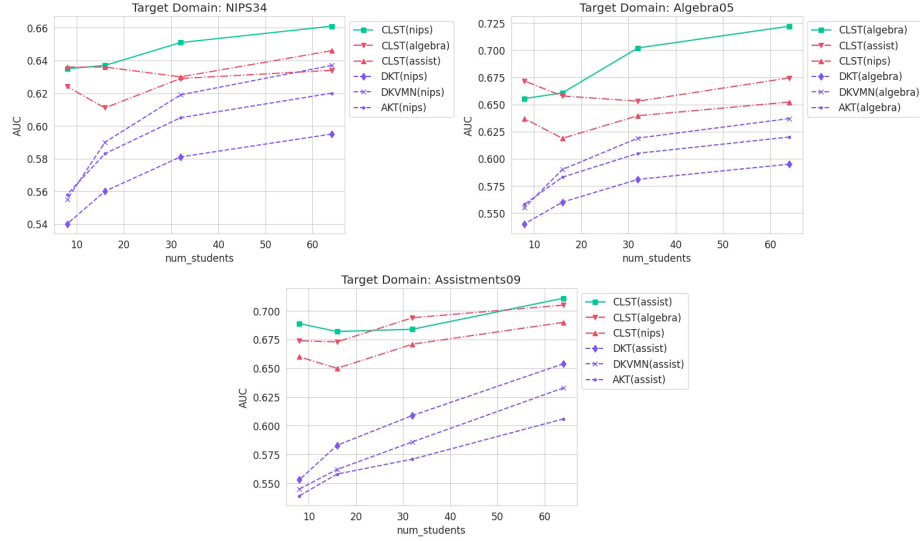

Fig. 9. Results of cross-domain experiments.

Figure 9 shows the predictive performance in the cross-domain scenario. According to the overall experimental results, CLST tuned with the target domain data, depicted in green, showed the highest performance. Nonetheless, the CLST trained on a different source domain dataset, denoted by pink dotted lines, outperformed the DL-based KT models trained on the target domain, denoted by purple dotted lines. In particular, in experiments with the Assistments09

dataset, the CLST tuned with different domains (such as algebra and NIPS) demonstrated performance comparable to that tuned with the target domain.

In fact, the majority of current research in the field of KT has primarily concentrated on domain-specific cases. However, real-world scenarios may present obstacles to this approach, such as the inability to obtain data on student–exercise interactions. In such cases, domain-specific models might not function effectively as a result of insufficient data [7], making the evaluation of cross-domain generalization capacity critical when deploying KT models in practice.

Ultimately, this experiment reveals the following answer to RQ4: Models trained with the CLST are more generalizable across domains; therefore, when building an ITS in situations where data is scarce and sufficient student–exercise interactions cannot be obtained, the CLST can be a promising choice.

5   Conclusion

In this study, we developed the CLST, which utilizes a generative LLM as a knowledge tracer. Most existing KT models are designed based on an ID-based approach, which exhibits poor predictive performance in cold-start cases with insufficient data. In contrast, the CLST demonstrates high performance even in cold-start scenarios and exhibits robust cross-domain generalizability. For the experiments, we collected data on the subjects of mathematics, social studies, and science. Subsequently, by expressing the problem-solving data in natural language, we framed the KT task as an NLP task and fine-tuned a generative LLM using the formatted KT dataset.

The experiment results showed that CLST outperformed the baseline models by up to 24.52% (mathematics), 9.62% (social studies), and 11.82% (science) in situations where the number of learners in the training data was insufficient. Furthermore, the ablation study showed that the description-based method applied in CLST, which represents each exercise with its description, achieved better performance than the conventional ID-based method. Additionally, the reliability experiment confirmed that fine-tuning with KTLP-formatted data enhances model reliability. Moreover, the learning trajectory analysis revealed that the CLST plausibly predicts students' mastery levels and understands the relationships between KCs. Finally, the cross-domain KT experiments verified that CLST is generalizable across multiple domains.

The results of this study have practical implications, serving as a guide for educational institutions and EdTech companies interested in fostering personalized learning through the development of ITS. Additionally, it offers theoretical implications by shedding light on strategies for leveraging generative LLMs in the KT domain.

Although the effectiveness of CLST has been demonstrated in numerous experiments, it has the potential to yield even higher predictive performance. For example, the inclusion of additional tasks may increase performance on the target task [51]. Therefore, it is worthwhile to contemplate future research endeavors to align generative LLMs as superior knowledge tracers by integrating a vari-

ety of educational tasks related to KT (e.g., difficulty estimation, KC relation prediction, etc.) into the fine-tuning process.

References

1. Abdelghani, R., Wang, Y., Yuan, X., Wang, T., Sauzéon, H., Oudeyer, P.: Gpt-3-driven pedagogical agents for training children's curious question-asking skills. arxiv. preprint arXiv **2211** (2022)
2. Abdelrahman, G., Wang, Q.: Knowledge tracing with sequential key-value memory networks. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp. 175–184 (2019)
3. Abdelrahman, G., Wang, Q., Nunes, B.: Knowledge tracing: A survey. ACM Computing Surveys **55**(11), 1–37 (2023)
4. Ai, F., Chen, Y., Guo, Y., Zhao, Y., Wang, Z., Fu, G., Wang, G.: Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. International Educational Data Mining Society (2019)
5. Bulut, O., Yildirim-Erbasli, S.N.: Automatic story and item generation for reading comprehension assessments with transformers. International Journal of Assessment Tools in Education **9**(Special Issue), 72–87 (2022)
6. Chen, P., Lu, Y., Zheng, V.W., Pian, Y.: Prerequisite-driven deep knowledge tracing. In: 2018 IEEE international conference on data mining (ICDM). pp. 39–48. IEEE (2018)
7. Cheng, S., Liu, Q., Chen, E., Zhang, K., Huang, Z., Yin, Y., Huang, X., Su, Y.: Adaptkt: A domain adaptable method for knowledge tracing. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining. pp. 123–131 (2022)
8. Colpo, M.P., Primo, T.T., de Aguiar, M.S.: Lessons learned from the student dropout patterns on covid-19 pandemic: An analysis supported by machine learning. British Journal of Educational Technology **55**(2), 560–585 (2024)
9. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction **4**, 253–278 (1994)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Dijkstra, R., Genç, Z., Kayal, S., Kamps, J., et al.: Reading comprehension quiz generation using generative pre-trained transformers. In: iTextbooks@ AIED. pp. 4–17 (2022)
12. Emerson, A., Min, W., Azevedo, R., Lester, J.: Early prediction of student knowledge in game-based learning with distributed representations of assessment questions. British Journal of Educational Technology **54**(1), 40–57 (2023)
13. Feng, M., Heffernan, N., Koedinger, K.: Addressing the assessment challenge with an online system that tutors as it assesses. User modeling and user-adapted interaction **19**, 243–266 (2009)
14. Gervet, T., Koedinger, K., Schneider, J., Mitchell, T., et al.: When is deep learning the best approach to knowledge tracing? Journal of Educational Data Mining **12**(3), 31–54 (2020)
15. Ghosh, A., Heffernan, N., Lan, A.S.: Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2330–2339 (2020)

16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
17. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)
18. Jin, M., Wen, Q., Liang, Y., Zhang, C., Xue, S., Wang, X., Zhang, J., Wang, Y., Chen, H., Li, X., et al.: Large models for time series and spatio-temporal data: A survey and outlook. arXiv preprint arXiv:2310.10196 (2023)
19. Jung, H., Yoo, J., Yoon, Y., Jang, Y.: Language proficiency enhanced knowledge tracing. In: International Conference on Intelligent Tutoring Systems. pp. 3–15. Springer (2023)
20. Kim, S., Kim, W., Jung, H., Kim, H.: Dikt: Dichotomous knowledge tracing. In: Intelligent Tutoring Systems: 17th International Conference, ITS 2021, Virtual Event, June 7–11, 2021, Proceedings 17. pp. 41–51. Springer (2021)
21. Lee, W., Chun, J., Lee, Y., Park, K., Park, S.: Contrastive learning for knowledge tracing. In: Proceedings of the ACM Web Conference 2022. pp. 2330–2338 (2022)
22. Li, L., Zhang, Y., Liu, D., Chen, L.: Large language models for generative recommendation: A survey and visionary discussions. arXiv preprint arXiv:2309.01157 (2023)
23. Van der Linden, W.J., Hambleton, R.: Handbook of item response theory. Taylor & Francis Group. Citado na pág $\mathbf{1}$(7), 8 (1997)
24. Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., Hu, G.: Ekt: Exercise-aware knowledge tracing for student performance prediction. IEEE Transactions on Knowledge and Data Engineering $\mathbf{33}$(1), 100–115 (2019)
25. Liu, Y., Yang, Y., Chen, X., Shen, J., Zhang, H., Yu, Y.: Improving knowledge tracing via pre-training question embeddings. arXiv preprint arXiv:2012.05031 (2020)
26. Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J., Luo, W.: pykt: a python library to benchmark deep learning based knowledge tracing models. Advances in Neural Information Processing Systems $\mathbf{35}$, 18542–18555 (2022)
27. Lu, Y., Chen, P., Pian, Y., Zheng, V.W.: Cmkt: Concept map driven knowledge tracing. IEEE Transactions on Learning Technologies $\mathbf{15}$(4), 467–480 (2022)
28. Malik, A., Wu, M., Vasavada, V., Song, J., Coots, M., Mitchell, J., Goodman, N., Piech, C.: Generative grading: near human-level accuracy for automated feedback on richly structured problems. arXiv preprint arXiv:1905.09916 (2019)
29. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems $\mathbf{26}$ (2013)
30. Nakagawa, H., Iwasawa, Y., Matsuo, Y.: Graph-based knowledge tracing: modeling student proficiency using graph neural network. In: IEEE/WIC/ACM International Conference on Web Intelligence. pp. 156–163 (2019)
31. Neshaei, S.P., Davis, R.L., Hazimeh, A., Lazarevski, B., Dillenbourg, P., Käser, T.: Towards modeling learner performance with large language models. arXiv preprint arXiv:2403.14661 (2024)
32. Ni, Q., Wei, T., Zhao, J., He, L., Zheng, C.: Hhskt: A learner–question interactions based heterogeneous graph neural network model for knowledge tracing. Expert Systems with Applications $\mathbf{215}$, 119334 (2023)
33. Olney, A.M.: Generating multiple choice questions from a textbook: Llms match human performance on most metrics. In: AIED Workshops (2023)
34. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)

35. Pandey, S., Srivastava, J.: Rkt: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 1205–1214 (2020)
36. Pavlik Jr, P.I., Cen, H., Koedinger, K.R.: Performance factors analysis–a new alternative to knowledge tracing. Online Submission (2009)
37. Peng, C., Yang, X., Chen, A., Smith, K.E., PourNejatian, N., Costa, A.B., Martin, C., Flores, M.G., Zhang, Y., Magoc, T., et al.: A study of generative large language model for medical research and healthcare. NPJ Digital Medicine **6**(1), 210 (2023)
38. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. Advances in neural information processing systems **28** (2015)
39. Sahebi, S., Cohen, W.W.: Community-based recommendations: a solution to the cold start problem. In: Workshop on recommender systems and the social web, RSWEB. vol. 60 (2011)
40. Shen, S., Liu, Q., Chen, E., Huang, Z., Huang, W., Yin, Y., Su, Y., Wang, S.: Learning process-consistent knowledge tracing. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. pp. 1452–1460 (2021)
41. Shen, S., Liu, Q., Huang, Z., Zheng, Y., Yin, M., Wang, M., Chen, E.: A survey of knowledge tracing: Models, variants, and applications. IEEE Transactions on Learning Technologies (2024)
42. Sridhar, P., Doyle, A., Agarwal, A., Bogart, C., Savelka, J., Sakr, M.: Harnessing llms in curricular design: Using gpt-4 to support authoring of learning objectives. arXiv preprint arXiv:2306.17459 (2023)
43. Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G.J., Koedinger, K.R.: Algebra I 2005-2006. development data set from kdd cup 2010 educational data mining challenge (2010), find it at `http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp`
44. Su, Y., Liu, Q., Liu, Q., Huang, Z., Yin, Y., Chen, E., Ding, C., Wei, S., Hu, G.: Exercise-enhanced sequential modeling for student performance prediction. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
45. Sun, J., Wei, M., Feng, J., Yu, F., Li, Q., Zou, R.: Progressive knowledge tracing: Modeling learning process from abstract to concrete. Expert Systems with Applications **238**, 122280 (2024)
46. Tong, H., Wang, Z., Zhou, Y., Tong, S., Han, W., Liu, Q.: Hgkt: Introducing hierarchical exercise graph for knowledge tracing. arXiv preprint arXiv:2006.16915 (2020)
47. Vie, J.J., Kashima, H.: Knowledge tracing machines: Factorization machines for knowledge tracing. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 750–757 (2019)
48. Wang, C., Ma, W., Zhang, M., Lv, C., Wan, F., Lin, H., Tang, T., Liu, Y., Ma, S.: Temporal cross-effects in knowledge tracing. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 517–525 (2021)
49. Wang, Q., Mousavi, A.: Which log variables significantly predict academic achievement? a systematic review and meta-analysis. British Journal of Educational Technology **54**(1), 142–191 (2023)
50. Wang, Z., Lamb, A., Saveliev, E., Cameron, P., Zaykov, Y., Hernández-Lobato, J.M., Turner, R.E., Baraniuk, R.G., Barton, C., Jones, S.P., et al.: Instructions and guide for diagnostic questions: The neurips 2020 education challenge. arXiv preprint arXiv:2007.12061 (2020)

51. Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M., Le, Q.V.: Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652 (2021)
52. Wu, T., Ling, Q.: Fusing hybrid attentive network with self-supervised dual-channel heterogeneous graph for knowledge tracing. Expert Systems with Applications **225**, 120212 (2023)
53. Wu, Z., Huang, L., Huang, Q., Huang, C., Tang, Y.: Sgkt: Session graph-based knowledge tracing for student performance prediction. Expert Systems with Applications **206**, 117681 (2022)
54. Xu, J., Huang, X., Xiao, T., Lv, P.: Improving knowledge tracing via a heterogeneous information network enhanced by student interactions. Expert Systems with Applications **232**, 120853 (2023)
55. Yang, H., Hu, S., Geng, J., Huang, T., Hu, J., Zhang, H., Zhu, Q.: Heterogeneous graph-based knowledge tracing with spatiotemporal evolution. Expert Systems with Applications **238**, 122249 (2024)
56. Yeung, C.K., Yeung, D.Y.: Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: Proceedings of the fifth annual ACM conference on learning at scale. pp. 1–10 (2018)
57. Yin, Y., Dai, L., Huang, Z., Shen, S., Wang, F., Liu, Q., Chen, E., Li, X.: Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In: Proceedings of the ACM Web Conference 2023. pp. 855–864 (2023)
58. Zhang, J., Shi, X., King, I., Yeung, D.Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th international conference on World Wide Web. pp. 765–774 (2017)
59. Zhang, M., Zhu, X., Zhang, C., Ji, Y., Pan, F., Yin, C.: Multi-factors aware dual-attentional knowledge tracing. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 2588–2597 (2021)
60. Zhao, J., Bhatt, S., Thille, C., Gattani, N., Zimmaro, D.: Cold start knowledge tracing with attentive neural turing machine. In: Proceedings of the Seventh ACM Conference on Learning@ Scale. pp. 333–336 (2020)
61. Zhao, S., Fu, H., Gong, M., Tao, D.: Geometry-aware symmetric domain adaptation for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9788–9798 (2019)
62. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
63. Zhou, H., Gu, B., Zou, X., Li, Y., Chen, S.S., Zhou, P., Liu, J., Hua, Y., Mao, C., Wu, X., et al.: A survey of large language models in medicine: Progress, application, and challenge. arXiv preprint arXiv:2311.05112 (2023)