Expert Systems With Applications 266 (2025) 125988

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

csKT: Addressing cold-start problem in knowledge tracing via kernel bias and cone attention

Youheng Bai [a], Xueyi Li [a], Zitao Liu [a,*], Yaying Huang [a], Teng Guo [a], Mingliang Hou [b], Feng Xia [c], Weiqi Luo [a]

[a] Guangdong Institute of Smart Education, Jinan University, Guangzhou 510632, China
[b] TAL Education Group, Beijing 100080, China
[c] School of Computing Technologies, RMIT University, Melbourne, VIC 3000, Australia

ARTICLE INFO

Keywords:
Knowledge tracing
Cold-start problem
Cone attention
AI for education

ABSTRACT

Knowledge tracing (KT) is the task of predicting students' future performances based on their past interactions in online learning systems. When new students enter the system with short interaction sequences, the cold-start problem commonly arises in KT. Although existing deep learning based KT models exhibit impressive performance, it remains challenging for these models to be trained on short student interaction sequences and maintain stable prediction accuracy as the number of student interactions increases. In this paper, we propose cold-start KT (**csKT**) to address this problem. Specifically, csKT employs kernel bias to guide learning from short sequences and ensure accurate predictions for longer sequences, and it also introduces cone attention to better capture complex hierarchical relationships between knowledge components in cold-start scenarios. We evaluate csKT on four public real-world educational datasets, where it demonstrated superior performance over a broad range of deep learning based KT models using common evaluation metrics in cold-start scenarios. Additionally, we conduct ablation studies and produce visualizations to verify the effectiveness of our csKT model. In support of reproducible research, we have made all datasets and the corresponding code publicly accessible at https://pykt.org/.

1. Introduction

Knowledge tracing (KT) is a crucial component in modern online learning systems. It uses students' past interactions, especially their answers to previous questions, to model their mastery of knowledge over time. This modeling enables predictions of their performance in future learning. Fig. 1 gives an illustrative example of the KT task. KT's prediction capabilities and versatility extend across various online learning systems, prominently in massive open online courses (MOOCs) and intelligent tutoring systems, thereby markedly enhancing the adaptability of online education to meet diverse learner needs. As a key technology in personalized and adaptive learning, KT has attracted widespread attention from education researchers.

Research related to KT has been underway since the 1990s. The seminal work by (Corbett & Anderson, 1994) is widely recognized as the first significant attempt to assess students' current understanding in relation to individual knowledge components (KCs).[1] A KC is a description of a mental structure or process that a learner uses, either alone or combined with other KCs, to perform specific actions within a task or to solve a problem.

Motivated by the remarkable progress in deep learning, particularly in the domain of time series prediction, various deep learning based knowledge tracing (DLKT) models have rapidly emerged, such as auto-regressive based deep sequential KT models (Guo et al., 2021; Käser et al., 2017; Piech et al., 2015), memory-augmented KT models (Abdelrahman & Wang, 2019; Zhang et al., 2017), attention based KT models (Choi et al., 2020; Ghosh et al., 2020; Pandey & Karypis, 2019; Yin et al., 2023), and graph based KT models (Cui et al., 2024; Nakagawa et al., 2021; Yang et al., 2020). In addition to diverse neural network architectures, these DLKT models incorporate a broad range of educational data, including question content (Liu et al., 2021), similarity measures, difficulty levels (Ghosh et al., 2020; Shen et al., 2022), question pre-training (Sun et al., 2023; Wang et al., 2024) and the relationships between questions and KCs (Pandey & Srivastava, 2020;

* Corresponding author.
E-mail addresses: yhbai@stu2024.jnu.edu.cn (Y. Bai), lixueyi@stu2021.jnu.edu.cn (X. Li), liuzitao@jnu.edu.cn (Z. Liu), huangyaying@jnu.edu.cn (Y. Huang), tengguo@jnu.edu.cn (T. Guo), houmingliang@tal.com (M. Hou), feng.xia@rmit.edu.au (F. Xia), lwq@jnu.edu.cn (W. Luo).

[1] A knowledge component (KC) is a generalization of everyday terms like concept, principle, fact, or skill.

https://doi.org/10.1016/j.eswa.2024.125988
Received 7 May 2024; Received in revised form 11 November 2024; Accepted 28 November 2024
Available online 6 December 2024

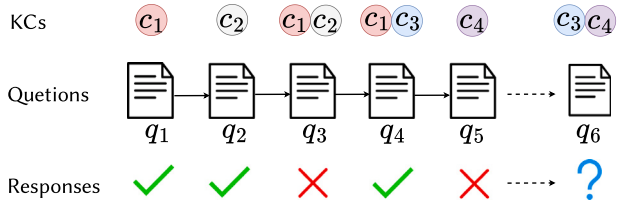

Fig. 1. Graphical illustration of knowledge tracing task.

Tong et al., 2022; Yang et al., 2020). This integration has significantly enhanced their prediction accuracy of KT tasks.

The effectiveness of deep learning models in predicting time series largely depends on the length of the time series data. This is also true for DLKT models in KT tasks. Sufficient student interaction data is crucial to the effectiveness of DLKT models. However, collecting sufficiently long interaction sequences is not always possible in all scenarios, such as when a student first starts using a smart education system. Therefore, effectively modeling students' knowledge mastery with short sequences (i.e., the cold-start scenario) remains a crucial challenge in KT research.

In cold-start scenarios, two key challenges remain unresolved. First, to meet the demands for real-time and effective applications in smart education, we aim for KT models trained on short sequences to be directly applicable to longer interaction sequences during the inference stage (Press et al., 2022). However, in cold-start scenarios, the sequence lengths of all available data are inherently short, resulting in limited information. Consequently, as the sequence length increases during extrapolation, the performance of KT models tends to deteriorate significantly. As shown in Section 4.4 (Tables 3 and 4), the extrapolation performance of current attention based KT models declines markedly when the prediction sequence length extends from 50 to 500.

Secondly, a consensus among existing solutions addressing the cold-start issue is that introducing more dimensions of auxiliary features can effectively mitigate the challenges associated with early cold-start situations for users (Patel & Thakkar, 2022; Wang, Peng, et al., 2020). In KT, the relationships between KCs represent one of the crucial pieces of information embedded in student interaction data, as shown in Fig. 2. However, current research on KC relationships often relies on feature representations designed for large-scale, long-sequence data, which are inherently coarse-grained and fail to capture more fine-grained relationships between knowledge points. When sequence lengths are sufficiently long, models can learn the relationships between KCs from ample interaction data. However, in the case of short interaction sequences, the existing feature representation approaches, which are tailored for long-sequence data, fail to capture the critical KC relationships beyond the current interactions, leading to degraded model performance. Therefore, understanding how to enable models to capture the relationships between KCs in short-sequence scenarios remains a primary challenge in the field.

In this study, we tackle the previously mentioned challenges in KT by introducing a novel model, called csKT. Our key innovation lies in the unique *train short, test long* training-evaluation approach in the KT task, specifically designed to tackle the problem caused by short interaction sequences in cold-start scenarios. This approach enables the proposed model to efficiently train on shorter sequences and test it on longer sequences, incorporating a kernel bias for enhanced adaptability. Furthermore, we introduce the cone attention module to capture the rich implicit hierarchical relationships between KCs in cold-start scenarios. Additionally, the effectiveness of csKT is rigorously evaluated across four KT datasets, comparing its performance with existing KT methods under a comprehensive evaluation protocol (Liu, Liu, Chen, et al., 2022). The main contributions of our work are summarized as follows:

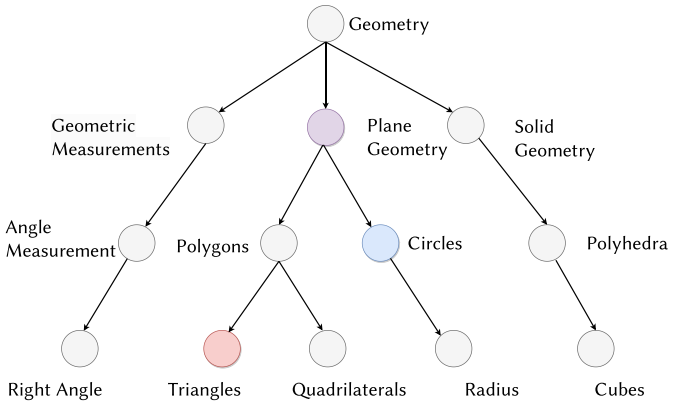

Fig. 2. A graph for hierarchical relationships between KCs in geometry subjects. For instance, *Circles* (blue) and *Triangles* (red) represent fundamental elements of *Plane Geometry* (purple), exhibiting a hierarchical relationship.

- We introduce kernel bias to address the limitation of fixed-length training and testing in KT tasks, thereby achieving the prediction for ever-growing student interaction sequences.
- We utilize cone attention in hyperbolic space to model the similarity between interaction sequences, which captures hierarchical relationships within limited interaction sequences in cold-start scenarios.
- We perform detailed experiments to confirm the effectiveness of the csKT model across four public datasets, comparing it against a diverse set of baseline models. The well-designed experiments illustrate the superiority of our approach in prediction performance.

The remainder of this paper is structured as follows: Section 2 provides a review of literature on KT and its applications in cold-start scenarios. Section 3 introduces the csKT framework proposed in this paper, including KT's problem definition, csKT overview, and components of csKT. Section 4 describes the experimental validation process of the csKT framework, including quantitative and qualitative analysis of the experimental results. Finally, Section 5 concludes the paper.

2. Related work

2.1. Deep learning based KT models

KT has been studied for decades (Corbett & Anderson, 1994). In the early stages, KT research primarily followed two traditional paths: Bayesian Knowledge Tracing (BKT) methods and Item Response Theory (IRT) approaches. BKT based models often use probabilistic graphical models, such as Hidden Markov Models and Bayesian Belief Network (Villano, 1992) to trace changes in students' knowledge states through skill practice (Corbett & Anderson, 1994; Käser et al., 2017; Pardos & Heffernan, 2011). These models explained the student learning process considering the initial knowledge state, what the student has mastered, and the likelihood of guessing or slipping (Liu, Kong, Chen, et al., 2023; Liu, Kong, Peng, et al., 2023). In contrast, IRT based methods estimate student performance by deriving a function — typically a logistic function as suggested by (Rasch, 1993) — that considers factors such as student interactions, abilities, and problem difficulty levels.

In recent years, integrating deep learning methodologies into KT has significantly enhanced the importance of this research domain. Deep knowledge tracing (DKT) was the first model to incorporate deep learning into KT tasks (Piech et al., 2015). Piech et al. used recurrent neural networks (RNN) and long short-term memory (LSTM) to estimate the likelihood of accurately answering a question at each stage of the assessment. Inspired by DKT, several approaches have

emerged. For example, Yeung et al. applied regularization constraints to improve DKT's stability (Yeung & Yeung, 2018) . Guo et al. employed adversarial training methods to train the KT model (Guo et al., 2021), reducing the overfitting problem. Shen et al. integrated learning and time interval between problem responses into the hidden state transition (Shen et al., 2021). These sequence based models combine knowledge and problems into interaction inputs to continually update the state of knowledge but face challenges in effectively capturing the complex relationship between KCs and problems. Inspired by the Transformer architecture, some researchers have introduced attention mechanisms to KT tasks (Vaswani et al., 2017). The attention based KT models emphasize the significance of each question in forecasting the correct response to subsequent questions by adapting the attention weights assigned to the questions. Attention based KT models, such as the scaled dot-product (Pandey & Karypis, 2019; Vaswani et al., 2017; Yang et al., 2023) and time-aware attention based KT models (Ghosh et al., 2020; Yin et al., 2023), have made remarkable progress in KT tasks. However, they often fall short in handling short sequence data in cold-start scenarios. These attention based DLKT models mainly rely on extensive interaction data, often ignoring the importance of KC hierarchical relationships in students' historical interaction sequences (Hou et al., 2006; Huang, Hu, et al., 2024; Huang, Wei, et al., 2024; Nguyen et al., 2019; Sun et al., 2022). In contrast, our study introduces a new perspective for modeling similarities between student interaction sequences in short sequences by proposing a cone attention mechanism for hyperbolic space. Cone attention is particularly adept at capturing implicit hierarchical relationships within short interaction sequences, an aspect largely overlooked in the current literature.

2.2. Cold-start problem in knowledge tracing

The cold-start problem refers to the challenge of making reliable predictions with new users or limited user data (Lika et al., 2014), which is a common issue in recommender systems. To deal with this issue, content based recommendation methods learn the respective representations by using information about users and items, such as locations, apps, images and categories (Chen et al., 2012; Narducci et al., 2016; Wei & Chow, 2023; Wu et al., 2019). Collaborative filtering based recommendation methods make recommendations for new users by calculating similarity between users according to the user ratings for items (Alharbe et al., 2023; Liu et al., 2020; Xu et al., 2023; Xue et al., 2019). Hybrid recommendation methods integrate different recommendation approaches to extract complex features between users and items, to achieve personalized recommendations (Cai et al., 2023; Darban & Valipour, 2022; Feng et al., 2021; Li et al., 2020). KT tasks also face the cold-start problem. When new students enroll in online intelligent tutoring systems and have short interaction sequences (Jeevamol & Renumol, 2021; Pliakos et al., 2019; van der Velde et al., 2024), training a KT model on such short sequences and making accurate predictions on longer sequences as the number of student interactions increases become a challenge (Liu et al., 2021; Wu et al., 2022; Zhao et al., 2020). To the best of our knowledge, there is a few works tackling the cold-start problem in KT with DLKT methods. Liu et al. proposed EERNN, an RNN based model, to deal with the cold-start problem by learning question representation based on its original concept (Liu et al., 2021). Zhao et al. proposed ANTM to model the relation between sequential learning activities with an extra memory bank (Zhao et al., 2020). Wu et al. proposed SGKT to model students' answering process by session graph and the relationship between questions and KCs (Wu et al., 2022). Jung et al. introduced a KT method that leverages language proficiency as auxiliary information to address the cold-start issue (Jung et al., 2023). Similarly, Mao et al. developed the FGKT approach, which utilizes an attention mechanism to discern the connections between assessment activities and past interactions, thereby identifying individual prior knowledge (Mao et al., 2023). However, for these DLKT models trained in cold-start scenarios, the increasing length of students' interaction sequences makes it difficult to accurately evaluate students' future answering behavior.

Different from the aforementioned works tackling the cold-start problem in KT, our proposed csKT effectively learns from short sequences and accurately predicts from growing student interactions by penalizing attention scores with kernel bias. Furthermore, in contrast to other DLKT models, we utilize cone attention, instead of dot-product attention, to better capture hierarchical relationships between questions and their associated KCs in cold-start scenarios.

Table 1
Key mathematical notations.

Notations	Description
S	Student
q	Question
c	KC
r	Binary response
t	Time step
d	Dimension of latent representations
$\mathbf{x}_t$	KC representation (with question) at timestep t
$\mathbf{y}_t$	Interaction representation at timestep t
$\mathbf{z}_{\mathbf{c}_t}$	KC representation at timestep t
$\mathbf{a}_{\mathbf{r}_t}$	Response representation at timestep t
$\mathbf{q}, \mathbf{k}, \mathbf{v}$	Query, key and value vectors in cone attention
$\mathbf{W}_c, \mathbf{W}_r$	Trainable matrix in input encoding
$\mathbb{R}^d, \mathbb{H}^d$	d-dimensional Euclidean and hyperbolic space

3. The csKT framework

3.1. Problem formulation

The objective of KT is to leverage a sequence of educational exercises over time to forecast the likelihood that a student will correctly respond to an upcoming question. More specifically, for each student S, we track a time-ordered sequence of T past interactions, denoted as $\mathrm{S} = \{\mathrm{s}_j\}_{j=1}^{T}$. Each interaction is captured as a 4-tuple $\mathrm{s} = \langle q, \{c\}, r, t\rangle$, where $q, \{c\}, r, t$ correspond to the specific question, the associated set of KCs, the student's binary response (1 for correct, 0 for incorrect), and the time step of the response, respectively. KT utilizes a series of past interactions, S_t, along with the next question q_{t+1}, to estimate the probability $\hat{r}_{t+1}$ that a student will correctly answer the new question q_{t+1} at the subsequent time step $t+1$. This process effectively traces the progression of the student's knowledge state. In this paper, the csKT aims to develop a predictive function as follows:

$$\hat{r}_{t+1} = f(\mathrm{S}_t, q_{t+1} \mid \Theta) \tag{1}$$

where Θ represents the trainable parameters of the model. Our goal is to enable csKT to train f in cold-start scenarios with short sequences (e.g., l_t), and to equip csKT with the ability to extrapolate and accurately predict student responses $\hat{r}_{t+1}^{(l_p)}$, where $l_p \gg l_t$.

To elucidate the architecture of our model and the methods used for its evaluation, we have compiled all critical mathematical symbols in Table 1. These symbols are uniformly applied throughout this paper. It is our convention to represent matrices with bold uppercase letters and vectors with bold lowercase letters.

3.2. The framework overview

In this section, we provide an overview of our csKT framework, as shown in Fig. 3. The framework comprises four components: (1) Input encoding, which expands KT problem-response data to include KC-response data and uses learnable scalars to implicitly represent potential factors influencing problem difficulty (see Section 3.3); (2) Cone attention, employing hyperbolic mapping functions and a hyperbolic cone to capture the hierarchical relationships among interactions (see Section 3.4); (3) Kernel bias, designed to enhance the KT model's

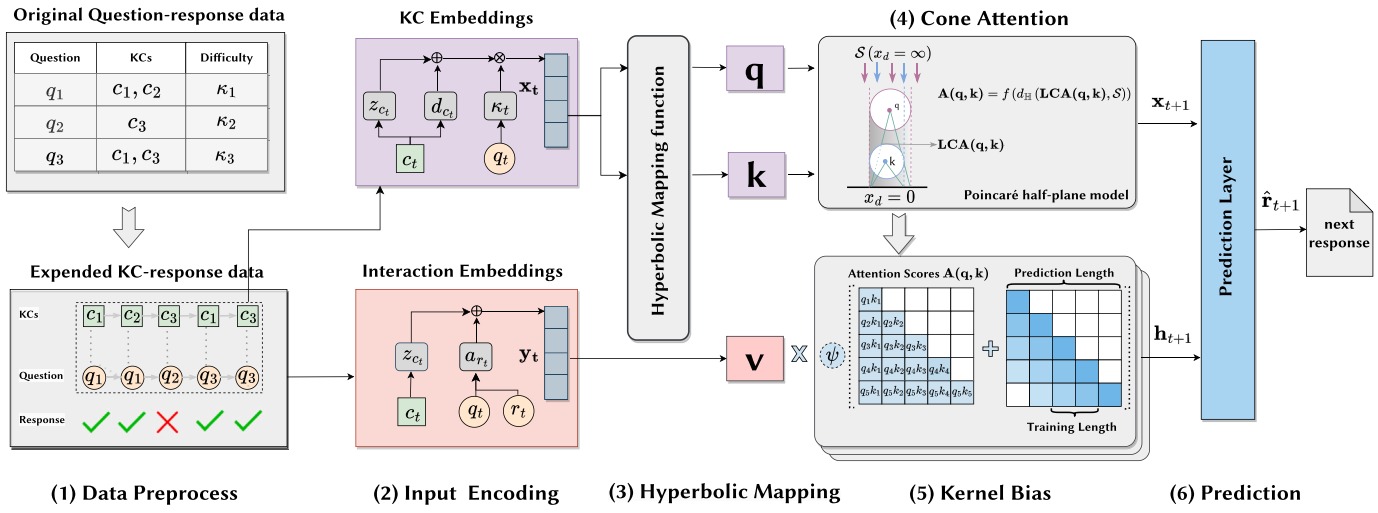

Fig. 3. The overview architecture of csKT includes: (1) transforming original question-response data into expanded KC-response data; (2) input encoding for KC embeddings and Interaction embeddings; (3) hyperbolic mapping to generate inputs **q** and **k** for cone attention; (4) calculating the attention score A(**q**,**k**) using cone attention; (5) adding kernel bias and attention scores to alleviate the limitation of fixed-length training and testing in KT tasks; (6) using the attention score and next question to predict the response.

prediction performance for longer sequences in cold-start limited data scenarios (see Section 3.5); (4) Prediction layer, using a two-layer fully connected network for KT prediction (see Section 3.6).

3.3. Input encoding

In online learning, a common challenge in real-world datasets is the imbalance between a large pool of questions and a smaller set of KCs. To effectively learn and evaluate from such highly sparse data, we follow previous work and transform the initial question-response data into KC-response data (Liu, Liu, Chen, et al., 2022). This process includes splitting each problem-level interaction into multiple KC-focused interactions, particularly for questions containing multiple KCs, as illustrated in Fig. 3(1). The input for csKT model comprises two parts: KC embedding $\mathbf{x}$ and interaction embedding $\mathbf{y}$.

3.3.1. KC embedding

To capture personalized differences in questions covering the same set of KCs, we use the classic and interpretable Rasch model from psychometrics (Rasch, 1993). This model represents the latent factor of question difficulty by a learnable scalar. It integrates a question-specific difficulty embedding to differentiate personalized variations in questions covering the same KCs. Specifically, the KC embedding, denoted by $\mathbf{x}_t$, is constructed at time step t as follows:

$$\mathbf{x}_t = \mathbf{z}_{c_t} \oplus \boldsymbol{\kappa}_{q_t} \odot \mathbf{d}_{c_t} \tag{2}$$

$$\mathbf{z}_{c_t} = \mathbf{W}_c \cdot \mathbf{e}_{c_t} \tag{3}$$

where $\mathbf{z}_{c_t}$ represents the latent representation of the KC c_t. The $\mathbf{d}_{c_t}$ refers to a variation embedding that summarizes the variation in questions covering c_t, while $\boldsymbol{\kappa}_{q_t}$ is a scalar learnable difficulty parameter for question q_t. $\mathbf{e}_{c_t}$ is an n-dimensional one-hot embedding representing the corresponding KC. Here, n refers to the total number of KCs. $\odot$ and $\oplus$ are the element-wise product and addition operators, respectively.

3.3.2. Interaction embedding

The interaction embedding has a structure similar to that of the question embedding, but it excludes any difficulty parameters. Questions and responses are combined to create the interaction embedding. Its construction proceeds as follows:

$$\mathbf{y}_t = \mathbf{z}_{c_t} \oplus \mathbf{a}_{r_t} \tag{4}$$

$$\mathbf{a}_{r_t} = \mathbf{W}_r \cdot \mathbf{e}_{r_t} \tag{5}$$

where $\mathbf{e}_{r_t}$ is the 2-dimensional one-hot vector indicating the correctness of the response to a question. $\mathbf{a}_{r_t}$ represents the student's response to question q_t. $\mathbf{z}_{c_t}, \boldsymbol{\kappa}_{q_t}, \mathbf{d}_{c_t}$ and $\mathbf{e}_{r_t}$ are d-dimensional and consist of learnable real-valued parameters. The matrices $\mathbf{W}_c \in \mathbb{R}^{d\times n}$ and $\mathbf{W}_r \in \mathbb{R}^{d\times 2}$ function as learnable linear transformations, facilitating the mapping of inputs to the desired output dimensions.

3.4. Cone attention

In KT tasks, attention based models significantly improve prediction performance (Ghosh et al., 2020; Liu, Liu, Chen, Huang, & Luo, 2023; Pandey & Karypis, 2019). This improvement is attributed to the dot-product mechanism. However, the dot-product approach in Euclidean space has limitations in representing complex structural features of real-world datasets (Tay et al., 2023; Tseng et al., 2023), such as the tree-like hierarchical relationships between questions and KCs. Similar to the exponential increase in the number of leaves with the depth in a tree, the volume of a hyperbolic ball expands exponentially in relation to its radius. This characteristic renders hyperbolic spaces highly suitable for embedding data with tree-like hierarchical structures.

3.4.1. Hyperbolic mapping

In order to represent the hierarchical structural relationships within KC embeddings accurately and efficiently, it is necessary to project the KC embeddings into a d-dimensional hyperbolic space, denoted as $\mathbb{H}^d$. The hyperbolic space is ideal for hierarchical data due to their exponential expansion, which allows for more accurate embeddings with less distortion compared to the polynomial growth of Euclidean spaces. However, as hyperbolic space cannot be isometrically embedded into Euclidean space, we use a mapping function to bridge this gap and transform the KC embeddings $\mathbf{x}$ into hyperbolic embedding representations $\phi(\mathbf{x})$ (Tseng et al., 2023). $\mathbf{x}$ is a d-dimensional vector represented as $\mathbf{x} = (x_1, x_2, \ldots, x_d)$, where each x_i is a component in Euclidean space $\mathbb{R}^d$. A mapping function ϕ maps $\mathbf{x}$ from $\mathbb{R}^d$ to hyperbolic space $\mathbb{H}^d$, which can be expressed as follows:

$$\phi(\mathbf{x})_{:d-1} = \mathbf{x}_{:d-1} \cdot \exp\left(\mathbf{x}_d\right) \quad \phi(\mathbf{x})_d = \exp\left(\mathbf{x}_d\right) \tag{6}$$

where $\phi(\mathbf{x})_{:d-1}$ represents the first $d-1$ elements of $\mathbf{x}$, and $\phi(\mathbf{x})_d$ denotes the d-th element of $\mathbf{x}$. The $\exp(\cdot)$ represents the exponential function.

In order to obtain the input of attention computation in hyperbolic space, similar to dot-product attention, we map KC embeddings $\mathbf{x}$ into query vector $\mathbf{q}$ and key vector $\mathbf{k}$ in hyperbolic space through the hyperbolic mapping function ϕ. This mapping ensures that the

calculation of attention scores reflects the hierarchical structure of the KC embedding, utilizing the intrinsic geometric properties of hyperbolic space to enhance the model's performance in hierarchical interaction data. $\mathbf{q}$ and $\mathbf{k}$ are expressed as:

$$\mathbf{q} = [\phi(\mathbf{q})_{:d-1}, \phi(\mathbf{q})_d], \tag{7}$$

$$\mathbf{k} = [\phi(\mathbf{k})_{:d-1}, \phi(\mathbf{k})_d] \tag{8}$$

3.4.2. Pairwise similarity

In the hyperbolic space, the similarity between two data points reflects their hierarchical relationships. After obtaining the hyperbolic representations $\mathbf{q}, \mathbf{k}$, we calculate the similarities using the implicit hierarchy defined by hyperbolic cone. Inspired by the work of (Tseng et al., 2023), we use this structure to facilitate the encoding of hierarchical relationships using the Poincaré half-space model. As shown in Fig. 3(4), a hyperbolic cone is generated by the shadows cast by points and a single light source S. It represents a point at infinity, with the points being the centers of spheres with a fixed radius r. When S is at infinity, the shadows are regions bounded by Euclidean lines perpendicular to the x-axis.

To build cone attention, we associate two points by the depth of their lowest common ancestor (LCA) in the cone partial ordering in hyperbolic cone, which is analogous to finding their LCA in a latent tree and captures how divergent two points are. We can calculate the LCA of two points by finding physically the hyperbolic cone that contains both. The LCA is expressed as:

$$LCA(\mathbf{q}, \mathbf{k}) = r\left(\arg\max_{\mathbf{q},\mathbf{k} \in C} d_{\mathbb{H}}(r(C), S)\right) \tag{9}$$

where $r(C)$ is the root of cone C, $d_{\mathbb{H}}(S, r(C))$ is hyperbolic distance between the root of all hierarchies S and the root of cone C.

The similarity scores of two points in a hyperbolic cone, calculated by S and LCA, reflect how closely related or similar the two points are in the hierarchy, with higher scores indicating closer proximity to the root of the cone. The similarity can be calculated by S and LCA:

$$A(\mathbf{q}, \mathbf{k}) = f\left(d_{\mathbb{H}}\left(LCA(\mathbf{q}, \mathbf{k}), S\right)\right) \tag{10}$$

where f is a monotonically increasing function, $d_{\mathbb{H}}(\cdot, \cdot)$ is the hyperbolic distance between the points.

In hyperbolic cone, $A(\mathbf{q}, \mathbf{k})$ represents how much attention or importance one data point (or node in the hierarchy) should give to another. The higher $A(\mathbf{q}, \mathbf{k})$ indicates that $\mathbf{q}$ and $\mathbf{k}$ are closely related in the hierarchy, similar to sharing an LCA in the tree. This means that in hyperbolic space, their proximity is not only spatial, but also reflects a close hierarchical relationship. The attention score for a hyperbolic space is calculated as follows:

$$A(\mathbf{q}, \mathbf{k}) = \exp\left(-\gamma \max\left(\phi(\mathbf{q})_d, \phi(\mathbf{k})_d, \frac{\left\|\phi(\mathbf{q})_{:d-1} - \phi(\mathbf{k})_{:d-1}\right\|}{2\sinh(r)} + \frac{\phi(\mathbf{q})_d + \phi(\mathbf{k})_d}{2}\right)\right) \tag{11}$$

where $\|\cdot\|$ is the L2-norm distance. $\phi(\mathbf{q})_{:d-1}$ and $\phi(\mathbf{k})_{:d-1}$ denote the first $d-1$ dimensions of $\mathbf{q}$ and $\mathbf{k}$ respectively. $\gamma \in \mathbb{R}^+$ is scaling coefficient in attention. $\mathrm{r} \in \mathbb{R}^+$ is the radius of the center of the sphere at each point in the hyperbolic cone.

Here, we point out that the entire design of Cone Attention is fundamentally similar to kernel methods (Fang et al., 2021; Gretton et al., 2005); both approaches project information that is difficult to model and distinguish in the original space into another space. Below, we provide an analysis of the relationship between Cone Attention and kernel methods.

First, feature space transformation is a common element between Cone Attention and kernel methods. Kernel methods, such as those used in Support Vector Machines (SVMs) (Mavroforakis & Theodoridis, 2006), implicitly map data to a higher-dimensional space through a kernel function, allowing linear relationships to emerge in what was previously a non-linear problem. Similarly, Cone Attention transforms the representation of KCs into a hyperbolic space, where the exponential properties of this space make it well-suited for modeling hierarchical structures. By mapping data to a hyperbolic space, Cone Attention effectively captures the layered relationships between KCs, which would be challenging to model in Euclidean space.

Second, both approaches emphasize relationship modeling through similarity measures. Kernel methods compute similarity between data points via inner products in the transformed space, thereby helping to capture non-linear patterns. Cone Attention, on the other hand, uses a "cone" structure to calculate similarities among KCs in hyperbolic space, leveraging the hierarchical distance between tokens. This is conceptually similar to kernel functions that aim to capture hidden relationships, as Cone Attention focuses on capturing the implicit, hierarchical connections that exist in educational data.

Moreover, both techniques address complex feature interactions that are not easily captured by traditional linear methods. Kernel methods achieve this through the use of non-linear kernel functions that enhance the expressive power of traditional algorithms. In a similar vein, Cone Attention enhances attention mechanisms by incorporating a structure-aware representation. This allows the model to disentangle and learn intricate, hierarchical relations between short sequences, which is particularly beneficial in cold-start scenarios where data are sparse.

3.5. Kernel bias

To address challenges associated with extrapolating from limited-length interaction data in cold-start scenarios, we have enhanced the csKT model with a kernel bias, inspired by (Chi et al., 2022). This enhancement enables effective training on short sequences while maintaining robust performance with increasing sequence lengths. Specifically, the kernel bias strengthens the attention mechanism by incorporating position-dependent information through a logarithmic decay function. In online learning systems, as students progress, the topics or skills they initially practice may be closely related. However, over time, the strength of the connection between earlier and later exercises naturally diminishes. The kernel bias effectively simulates this natural forgetting behavior of students, mirroring how the relevance of their exercise interactions decreases over time. Consequently, the kernel bias ensures that the model focuses on more recent questions and skills, maintaining relevance and accuracy despite increasing volumes of interaction data and extended time intervals. We formulate the kernel bias as follows:

$$\mathbf{A}_{bias} = \psi\left(A(\mathbf{q}, \mathbf{k}) + \tilde{B}\right) \tag{12}$$

where $\tilde{B}$ denotes a matrix of kernel bias. $\mathbf{q}$ and $\mathbf{k}$ represent query and key respectively. The $\psi(\cdot)$ is the Softmax function.

As shown in Fig. 3(5), we penalize attention scores with a kernel bias calculated by the query-key position (m, n), which implicitly contains position information of student interactions and forgetting information. According to this principle, a larger $|m-n|$ corresponds to a lower attention weight, whereas a smaller $|m-n|$ corresponds to a higher attention weight. This adjustment effectively expands the attention window to facilitate the extrapolation in the KT cold-start scenarios. Following previous work (Chi et al., 2022), we calculate the kernel bias with a logarithmic kernel:

$$\tilde{b}_{m,n} = \lambda_1 \cdot \log(1 - \lambda_2 |m-n|) + c$$

$$\tilde{B} = \begin{bmatrix} \tilde{b}_{11} & \tilde{b}_{12} & \cdots & \tilde{b}_{1n} \\ \tilde{b}_{21} & \tilde{b}_{22} & \cdots & \tilde{b}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{b}_{m1} & \tilde{b}_{m2} & \cdots & \tilde{b}_{mn} \end{bmatrix} \tag{13}$$

where $\lambda_1, \lambda_2 < 0$ and $c \in \mathbb{R}$. The $\tilde{b}_{m,n}$ denotes the element at the mth row and nth column of kernel bias $\tilde{B}$.

Subsequently, the output attention matrix is computed by multiplying $\mathbf{A}_{bias}$ with the interaction embedding $\mathbf{v}$. To prevent future information leakage in predicting the next question, we employ a causal mask matrix. This matrix ensures that current questions have access only to previous questions and themselves. Then we compute the output matrix $\mathbf{h}$ as:

$$\mathbf{h} = (\mathbf{A}_{bias} + \mathbf{M}) \cdot \mathbf{v} \tag{14}$$

where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is an upper triangular matrix, its lower triangular elements are zero, while the non-diagonal upper triangular elements are assigned a value of $-\infty$.

Here, we provide a theoretical discussion from the perspective of entropy to explain why the design of kernel bias enhances the extrapolation capability of attention based KT architectures. Overall, we posit that the kernel bias fundamentally reduces the entropy of the attention map. To ensure that model results generalize effectively to unseen sequence lengths, the design of the attention mechanism should ideally maintain the entropy invariance of the attention weights $a_{i,j}$. In cold-start scenarios, extrapolating to unknown lengths refers to achieving satisfactory performance when there is a mismatch between the training and inference sequence lengths, such as training with $n = 20$ and extrapolating to test lengths of $n = 50$ or $n = 500$. From another perspective, we can consider uncertainty as the 'focus degree' of attention: if the entropy is zero, the attention is concentrated on a single token; if the entropy is $\log n$, the attention is uniformly distributed across all tokens. We aim for entropy invariance so that when new tokens are introduced, the existing tokens still focus on the original ones in the same manner. This prevents the newly introduced tokens from excessively 'diluting' the original attention, which would result in significant changes to the summation outcome. The detailed analysis is as follows:

The scaled dot-product attention can be rewritten as follows:

$$\mathbf{o_i} = \sum_{j=1}^{n} a_{i,j}\mathbf{v_j}, \quad a_{i,j} = \frac{e^{\lambda \mathbf{q_i} \cdot \mathbf{k_j}}}{\sum_{j=1}^{n} e^{\lambda \mathbf{q_i} \cdot \mathbf{k_j}}}, \tag{15}$$

where $\lambda = \frac{1}{\sqrt{d}}$. One perspective on the extrapolation capability of attention based models presented in this paper is that, to *enhance the generalization of the results to unknown lengths, the design of the attention mechanism should ensure that $a_{i,j}$ possesses entropy invariance*.

Specifically, $a_{i,j}$ can be regarded as a conditional distribution where i is the condition and j is the random variable, with its entropy being:

$$\mathcal{H} = -\sum_{j=1}^{n} a_{i,j} \log a_{i,j} \tag{16}$$

Entropy invariance means that $\mathcal{H}$ should be insensitive to the length n. More specifically, if additional tokens are appended to the existing tokens, the newly calculated $a_{i,j}$ will naturally change, but we hope that $\mathcal{H}$ does not change significantly. We aim for entropy invariance to ensure that after introducing new tokens, the existing tokens can still focus on the original tokens in the same manner. We do not want the introduction of new tokens to excessively 'dilute' the original attention, leading to a significant change in the summation result.

Next, we will demonstrate that introducing attention penalties through a bias approach can better ensure entropy invariance of the attention matrix during extrapolation. Here, we define an $n \times n$ attention score matrix and the indices i and j correspond to the row and column positions within the matrix, respectively, where $0 \le i \le n$ and $0 \le j \le n$.

First, assume $\mathbf{q}_i \mathbf{k}_j = \mathbf{s}_i$, we have:

$$p_i = \frac{e^{\lambda \mathbf{s}_i}}{\sum_{i=1}^{n} e^{\lambda \mathbf{s}_i}} \tag{17}$$

The entropy is:

$$\begin{aligned} \mathcal{H} &= -\sum_{i=1}^{n} p_i \log p_i \\ &= \log \sum_{i=1}^{n} e^{\lambda \mathbf{s}_i} - \lambda \sum_{i=1}^{n} p_i \mathbf{s}_i \\ &= \log n + \log \frac{1}{n} \sum_{i=1}^{n} e^{\lambda \mathbf{s}_i} - \lambda \sum_{i=1}^{n} p_i \mathbf{s}_i \end{aligned} \tag{18}$$

Based on mean field theory (Furusawa, 2024), we have:

$$\log \frac{1}{n} \sum_{i=1}^{n} e^{\lambda \mathbf{s}_i} \approx \log \exp\left(\frac{1}{n} \sum_{i=1}^{n} \lambda \mathbf{s}_i \right) = \lambda \bar{s} \tag{19}$$

Moreover, the softmax operation tends to the max value of $a_{i,j}$ (Qin et al., 2022), we have:

$$\lambda \sum_{i=1}^{n} p_i \mathbf{s}_i \approx \lambda s_{\max} \tag{20}$$

Therefore, the entropy in the attention mechanism can ultimately be approximated as follows:

$$\mathcal{H} \approx \log n - \lambda(s_{\max} - \bar{s}) = \log n - \frac{1}{\sqrt{d}}(s_{\max} - \bar{s}) \tag{21}$$

Assume that the form of the bias for penalizing attention is $f(|i-j|)$, where $f(|i-j|) > 0$, we denote the entropy after adding the bias as $\mathcal{H}'$, we have:

$$\begin{aligned} &\mathcal{H}' \approx \log n - \frac{1}{\sqrt{d}}((s_{\max} - a) - (\bar{s} - b)), \\ &a = f(|i_{s_{max}} - j_{s_{max}}|) > 0 \\ &b = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} f(|i-j|)}{nn} > 0. \end{aligned} \tag{22}$$

Based on Eq. (21), we have:

$$\mathcal{H} - \mathcal{H}' = \frac{1}{\sqrt{d}}(b - a) \tag{23}$$

The final result in Eq. (23) depends on the disparity corresponding to the coordinates of the maximum attention value s_{max} and the monotonicity of the bias function. Generally, the maximum attention value is likely to be concentrated near the diagonal (Chi et al., 2022; Press et al., 2022), hence the disparity $|i-j|$ is expected to be smaller than the average $|i-j|$ within the matrix, i.e., $\frac{2}{3}n - \frac{2}{3}$. The kernelized bias function utilized in this paper is monotonically increasing. Therefore, the final result of Eq. (23) is highly likely to be greater than zero, i.e., $\mathbb{P}(\mathcal{H} - \mathcal{H}' > 0) \approx 1$.

Furthermore, we believe that the entropy-invariance property also contributes to modeling stable cognitive behaviors in students. In knowledge tracing, students' cognitive behaviors can generally be categorized into two types: stable cognitive behaviors, which reflect trends in a student's learning process (e.g., accuracy gradually improving with increased practice); and slipping/guessing behaviors (Liu, Kong, Peng, et al., 2023), which are often more random, including fluctuations due to factors like fatigue, forgetting, or guessing. Slipping/guessing behaviors tend to interfere with stable cognitive behaviors, thereby affecting the KT model's ability to capture overall learning trends (Liu, Liu, Liu, et al., 2022). The emergence of slipping/guessing features typically increases the entropy of the attention map, adversely impacting the model's extrapolation capability. Therefore, another reason Kernel Bias enhances the performance of attention based KT models is its effectiveness in recognizing trend-specific feature patterns, which helps the attention mechanism become more 'focused'.

3.6. Prediction and optimization

We use a two-hidden-layer fully connected neural network for prediction. To optimize the model, we use a cross-entropy loss function. This function compares the predicted probability of a correct answer, denoted as $\hat{\mathbf{r}}_{t+1}$, with the actual correct answer, represented by $\mathbf{r}_{t+1}$. The following objective function is defined over training data:

$$\hat{\mathbf{r}}_{t+1} = \sigma(\mathrm{ReLU}(\boldsymbol{W}_2 \cdot \mathrm{ReLU}(\boldsymbol{W}_1 \cdot [\mathbf{h_{t+1}}; \mathbf{x_{t+1}}]) + \boldsymbol{b}_1) + \boldsymbol{b}_2) \tag{24}$$

$$\mathcal{L} = -\sum_{t} (\mathbf{r_{t+1}} \cdot \log \hat{\mathbf{r}}_{\mathbf{t+1}} + (1 - \mathbf{r_{t+1}}) \cdot \log(1 - \hat{\mathbf{r}}_{\mathbf{t+1}})) \tag{25}$$

where σ denotes Sigmoid function. $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ are trainable parameters.

Table 2
Statistics of 4 real-world KT datasets in this paper.

Datasets	# Interactions	# Students	# Questions	# KCs
Statics2011	194,947	333	1,224	–
AL2005	809,694	574	210,710	112
BD2006	3,679,199	1,146	207,856	493
NIPS34	1,382,727	4,918	948	57

4. Experiments

4.1. Datasets

We use four KT datasets in real educational scenarios to evaluate csKT models across varying test lengths. Table 2 presents the statistics of all the datasets. Each dataset is introduced and compared as follows:

- **Statics2011**: This dataset was gathered from an Engineering Statics course conducted at Carnegie Mellon University in Fall 2011 (Steif & Bier, 2014). Due to a significant number of interactions missing details about the correctness of responses, the preprocessing required combining the *problem name* and *step name* attributes and focusing solely on interactions that included a valid initial attempt.
- **Algebra2005 (AL2005)**: This dataset, originating from the KDD Cup 2010 EDM Challenge, captures the interactions of 13–14-year-old students with an Algebra tutoring system (Stamper & Pardos, 2016). It is notable for its detailed tracing of student responses, uniquely identifying questions by combining problem and step names.
- **Bridge2006 (BD2006)**: This dataset, also sourced from the KDD Cup 2010 EDM Challenge, documents student interactions with mathematical problems in an intelligent tutoring system. It uniquely identifies questions in a manner similar to the AL05 dataset and offers an extensive compilation of questions and KCs for analysis.
- **NIPS34**: This dataset is provided by the NeurIPS 2020 Education Challenge, which comprises extensive student interactions with mathematics questions (Wang, Lamb, et al., 2020). It offers a rich array of data for Tasks 3 and 4, including a vast number of interactions, sequences, questions, and KCs, representing a diverse student demographic across various educational levels.

4.2. Baselines

In this study, we evaluate the csKT framework against 14 DLKT baselines. The specifics of each comparative baseline are detailed below:

- **DKT** introduced by (Piech et al., 2015), employs recurrent neural networks to simulate students' learning processes via sequences of questions and answers. In our implementation, we have utilized LSTM networks.
- **DKT+** (Yeung & Yeung, 2018) is an advanced version of DKT that focuses on solving the reconstruction of observed inputs and improving the consistency of performance predictions across different time steps.
- **DKVMN** (Zhang et al., 2017) uses memory networks with a static key matrix for latent KCs and a dynamic value matrix for updating learners' proficiency on the corresponding concepts.
- **EERNN** (Su et al., 2018) proposed a text-aware model that utilizes bi-LSTM to extract question representations. In this paper, we adopt EERNNA variant, which employs attention mechanism over all historical states.
- **ATKT** (Guo et al., 2021) enhances attention-LSTM based KT models by introducing adversarial perturbations to student interaction sequences, improving model generalization and mitigating overfitting.
- **SAKT** (Pandey & Karypis, 2019) uses self-attention to identify correlations in learning interactions and capture long-term dependencies for KT performance.
- **LPKT** (Shen et al., 2021) predicts student response by combining time based learning progress and forgetting factors with exercise sequence embeddings.
- **AKT** (Ghosh et al., 2020) models forgetting behaviors and predicts student performance by linking past responses to future outcomes through adaptive, distance-aware attention weights.
- **AT-DKT** (Liu, et al., 2023) enhances the DKT model by introducing two auxiliary learning tasks: question tagging and individualized prior knowledge prediction, improving prediction performance on student outcomes.
- **sparseKT** (Huang et al., 2023) improves the robustness of attention based DLKT models by soft thresholds and top-k sparse attention.
- **FoLiBiKT** (Im et al., 2023) introduces a module to the DLKT model, integrating a forgetting-aware bias that reflects forgetting behavior.
- **simpleKT** (Liu, Liu, Chen, Huang, & Luo, 2023) uses scaled dot-product attention in KT to distinguish question-specific characteristics and contextualize student interactions, simplifying attention based DLKT models while maintaining high performance.
- **Dtransformer** (Yin et al., 2023) employs a Transformer based model with a two-level framework to realize knowledge state estimation and maintains stability by contrastive learning.
- **HD-KT** (Ma et al., 2024) proposes knowledge state-guided and student profile guided anomaly detectors to identify and filter out anomalous interactions, improving KT model robustness and performance.

4.3. Experimental setting

In our experimental setup, we implement a comprehensive evaluation of various KT models, following the standardized evaluation process established in (Liu, Liu, Chen, et al., 2022). For the training and evaluation phase, we configured the maximum sequence length for student interaction sequences at 20 and employed 5-fold cross-validation, with 80% of the sequences allocated to training and validation sets. The remaining 20% were reserved for the testing phase, where we evaluated model performance across multiple sequence lengths (50, 100, and 500). We train models using the Adam optimizer with a maximum of 200 iterations, while incorporating an early stopping mechanism that terminated training if the AUC score on the validation set showed no improvement after 10 consecutive iterations.

The hyper-parameters r, γ, learning rate, and embedding size d are selected from the respective ranges: [0.1, 0.2, 0.4, 0.6, 0.8, 1.0], [0.05, 0.1, 1, 5, 10], [1e–3, 1e–4, 1e–5], and [32, 64, 128], the number of blocks and attention heads are set to [1, 2, 4, 8] and [4, 8], the seeds are selected from [42, 3407] for reproducing the experimental results. All models are implemented in PyTorch and trained on a cluster of Linux servers equipped with NVIDIA RTX 3090 GPUs. Consistent with existing DLKT research, we use the AUC as our primary evaluation metric. Additionally, we adopt accuracy as the secondary evaluation metric. All baseline models are optimally tuned to ensure a fair comparison.

4.4. Quantitative analysis

4.4.1. Overall performance

Experimental results are shown in Tables 3 and 4. **Bold** font represents the best results, and underlined font denotes the second-best results. From Tables 3 and 4, we have the following findings:

Table 3
Comparison of AUC for different models on four datasets at three prediction lengths.

Model	Statics2011			AL2005			BD2006			NIPS34		
	50	100	500	50	100	500	50	100	500	50	100	500
DKT	0.8172 ± 0.0007	0.8172 ± 0.0007	0.8172 ± 0.0007	0.8069 ± 0.0004	0.8070 ± 0.0004	0.8070 ± 0.0004	0.7901 ± 0.0004	0.7900 ± 0.0005	0.7900 ± 0.0005	0.7627 ± 0.0004	0.7628 ± 0.0004	0.7628 ± 0.0004
DKT+	0.8178 ± 0.0044	0.8140 ± 0.0074	0.8064 ± 0.0152	0.8058 ± 0.0006	0.8058 ± 0.0005	0.8059 ± 0.0005	0.7899 ± 0.0003	0.7897 ± 0.0004	0.7898 ± 0.0006	0.7618 ± 0.0004	0.7619 ± 0.0005	0.7619 ± 0.0005
DKVMN	0.7669 ± 0.0054	0.7552 ± 0.0087	0.7493 ± 0.0141	0.7482 ± 0.0029	0.7446 ± 0.0047	0.7346 ± 0.0083	0.7315 ± 0.0059	0.7245 ± 0.0071	0.6981 ± 0.0108	0.7255 ± 0.0015	0.7281 ± 0.0023	0.7244 ± 0.0029
EERNN	–	–	–	0.7747 ± 0.0011	0.7727 ± 0.0020	0.7701 ± 0.0011	0.7562 ± 0.0039	0.7511 ± 0.0031	0.7502 ± 0.0029	0.7566 ± 0.0007	0.7556 ± 0.0005	0.7536 ± 0.0003
ATKT	0.7554 ± 0.0116	0.7212 ± 0.0152	0.6690 ± 0.0193	0.7714 ± 0.0019	0.7268 ± 0.0033	0.6312 ± 0.0052	0.7553 ± 0.0017	0.7030 ± 0.0029	0.5651 ± 0.0077	0.7487 ± 0.0006	0.7281 ± 0.0023	0.6788 ± 0.0088
LPKT	–	–	–	0.8189 ± 0.0013	0.8055 ± 0.0051	0.6615 ± 0.0137	0.7919 ± 0.0014	0.7742 ± 0.0069	0.6170 ± 0.0578	0.7869 ± 0.0028	0.7641 ± 0.0054	0.7092 ± 0.0153
SAKT	0.7855 ± 0.0015	0.7833 ± 0.0014	0.7780 ± 0.0017	0.7087 ± 0.0058	0.6925 ± 0.0135	0.6689 ± 0.0195	0.7354 ± 0.0044	0.7234 ± 0.0024	0.6807 ± 0.0037	0.6987 ± 0.0027	0.6872 ± 0.0022	0.6780 ± 0.0015
AKT	0.8206 ± 0.0035	0.8178 ± 0.0053	0.8099 ± 0.0073	0.8205 ± 0.0026	0.8158 ± 0.0044	0.7992 ± 0.0083	0.8061 ± 0.0017	0.7978 ± 0.0036	0.7668 ± 0.0089	0.7953 ± 0.0006	0.7928 ± 0.0013	0.7864 ± 0.0032
AT-DKT	–	–	–	0.8166 ± 0.0021	0.8166 ± 0.0022	0.8147 ± 0.0023	0.8006 ± 0.0011	0.8003 ± 0.0021	0.7984 ± 0.0030	0.7756 ± 0.0006	0.7750 ± 0.0004	0.7748 ± 0.0008
sparseKT	0.8098 ± 0.0039	0.8074 ± 0.0034	0.8014 ± 0.0050	0.7682 ± 0.0048	0.7597 ± 0.0143	0.7513 ± 0.0087	0.7865 ± 0.0038	0.7793 ± 0.0061	0.7542 ± 0.0074	0.7857 ± 0.0026	0.7788 ± 0.0024	0.7735 ± 0.0038
FoLiBiKT	0.8205 ± 0.0025	0.8163 ± 0.0031	0.8047 ± 0.0034	0.8203 ± 0.0025	0.8162 ± 0.0032	0.7990 ± 0.0042	0.8046 ± 0.0011	0.7929 ± 0.0015	0.7493 ± 0.0033	0.7958 ± 0.0005	0.7937 ± 0.0015	0.7870 ± 0.0042
simpleKT	0.8033 ± 0.0068	0.7994 ± 0.0087	0.7918 ± 0.0100	0.7749 ± 0.0076	0.7725 ± 0.0118	0.7624 ± 0.0068	0.7883 ± 0.0020	0.7815 ± 0.0037	0.7641 ± 0.0097	0.7844 ± 0.0018	0.7800 ± 0.0035	0.7744 ± 0.0039
Dtransformer	0.8107 ± 0.0039	0.8037 ± 0.0044	0.7903 ± 0.0054	0.8127 ± 0.0044	0.8085 ± 0.0036	0.7939 ± 0.0058	0.7969 ± 0.0009	0.7882 ± 0.0033	0.7638 ± 0.0113	0.7915 ± 0.0007	0.7882 ± 0.0014	0.7806 ± 0.0022
HD-KT	–	–	–	0.8003 ± 0.0007	0.7964 ± 0.0019	0.7941 ± 0.00038	0.7812 ± 0.0040	0.7735 ± 0.0020	0.7720 ± 0.0060	0.7942 ± 0.0136	0.7861 ± 0.0084	0.7813 ± 0.0045
csKT	**0.8265 ± 0.0016**[a]	**0.8262 ± 0.0020**[a]	**0.8248 ± 0.0025**[a]	**0.8213 ± 0.0025**	**0.8199 ± 0.0023**[a]	**0.8156 ± 0.0020**[a]	**0.8094 ± 0.0008**	**0.8072 ± 0.0007**[a]	**0.7996 ± 0.0021**[a]	**0.7985 ± 0.0003**	**0.7985 ± 0.0004**[a]	**0.7966 ± 0.0005**[a]

[a] Indicate a statistically significant difference ($p < 0.01$) compared to the second-best result.

Table 4
Comparison of Accuracy for different models on four datasets at three prediction lengths.

Model	Statics2011			AL2005			BD2006			NIPS34		
	50	100	500	50	100	500	50	100	500	50	100	500
DKT	0.7950 ± 0.0007	0.7951 ± 0.0007	0.7951 ± 0.0007	**0.8075 ± 0.0007**	**0.8076 ± 0.0006**	**0.8077 ± 0.0006**	0.8528 ± 0.0002	0.8528 ± 0.0003	0.8527 ± 0.0003	0.6979 ± 0.0005	0.6981 ± 0.0006	0.6981 ± 0.0006
DKT+	0.7946 ± 0.0012	0.7934 ± 0.0024	0.7921 ± 0.0031	0.8062 ± 0.0004	0.8061 ± 0.0003	0.8062 ± 0.0003	0.8529 ± 0.0003	0.8529 ± 0.0003	0.8527 ± 0.0003	0.6972 ± 0.0007	0.6973 ± 0.0008	0.6973 ± 0.0008
DKVMN	0.7691 ± 0.0030	0.7672 ± 0.0039	0.7674 ± 0.0042	0.7826 ± 0.0016	0.7803 ± 0.0021	0.7758 ± 0.0085	0.8390 ± 0.0029	0.8417 ± 0.0016	0.8355 ± 0.0039	0.6667 ± 0.0011	0.6717 ± 0.0032	0.6670 ± 0.0022
EERNN	–	–	–	0.7892 ± 0.0012	0.7886 ± 0.0010	0.7880 ± 0.0015	0.8359 ± 0.0007	0.8330 ± 0.0006	0.8330 ± 0.0008	0.6878 ± 0.0004	0.6869 ± 0.0007	0.6860 ± 0.0009
ATKT	0.7539 ± 0.0123	0.7162 ± 0.0189	0.6833 ± 0.0209	0.7869 ± 0.0021	0.7500 ± 0.0071	0.6679 ± 0.0148	0.8316 ± 0.0020	0.7853 ± 0.0064	0.6522 ± 0.0281	0.6869 ± 0.0009	0.6717 ± 0.0032	0.6480 ± 0.0059
LPKT	–	–	–	0.8068 ± 0.0036	0.7736 ± 0.0279	0.6180 ± 0.1259	0.8133 ± 0.0215	0.7268 ± 0.0155	0.6253 ± 0.1100	0.7131 ± 0.0041	0.6638 ± 0.0153	0.6032 ± 0.0102
SAKT	0.7700 ± 0.0057	0.7657 ± 0.0112	0.7719 ± 0.0077	0.7647 ± 0.0012	0.7604 ± 0.0024	0.7520 ± 0.0012	0.8423 ± 0.0012	0.8398 ± 0.0013	0.8350 ± 0.0028	0.6350 ± 0.0026	0.6174 ± 0.0024	0.6194 ± 0.0028
AKT	0.7975 ± 0.0012	0.7965 ± 0.0019	0.7937 ± 0.0036	0.8064 ± 0.0015	0.8037 ± 0.0020	0.7977 ± 0.0032	0.8546 ± 0.0007	0.8521 ± 0.0014	0.8462 ± 0.0022	0.7247 ± 0.0019	0.7172 ± 0.0033	0.7153 ± 0.0038
AT-DKT	–	–	–	0.8102 ± 0.0009	0.8101 ± 0.0010	0.8091 ± 0.0007	0.8530 ± 0.0004	0.8530 ± 0.0003	0.8526 ± 0.0005	0.7107 ± 0.0009	0.7099 ± 0.0012	0.7097 ± 0.0008
sparseKT	0.7927 ± 0.0019	0.7924 ± 0.0014	0.7891 ± 0.0021	0.7868 ± 0.0047	0.7867 ± 0.0026	0.7863 ± 0.0019	0.8495 ± 0.0027	0.8489 ± 0.0009	0.8453 ± 0.0011	0.7114 ± 0.0057	0.6925 ± 0.0166	0.6966 ± 0.0078
FoLiBiKT	0.7972 ± 0.0014	0.7959 ± 0.0018	0.7934 ± 0.0021	0.8065 ± 0.0010	0.8046 ± 0.0010	0.7986 ± 0.0011	0.8536 ± 0.0003	0.8503 ± 0.0003	0.8436 ± 0.0007	0.7253 ± 0.0018	0.7196 ± 0.0034	0.7179 ± 0.0040
simpleKT	0.7875 ± 0.0038	0.7878 ± 0.0037	0.7831 ± 0.0046	0.7887 ± 0.0042	0.7887 ± 0.0028	0.7901 ± 0.0025	0.8491 ± 0.0019	0.8480 ± 0.0023	0.8468 ± 0.0013	0.7099 ± 0.0043	0.6935 ± 0.0074	0.6997 ± 0.0080
Dtransformer	0.7930 ± 0.0020	0.7908 ± 0.0016	0.7859 ± 0.0032	0.8037 ± 0.0026	0.8027 ± 0.0030	0.7985 ± 0.0016	0.8516 ± 0.0010	0.8497 ± 0.0015	0.8453 ± 0.0022	0.7226 ± 0.0012	0.7165 ± 0.0023	0.7145 ± 0.0022
HD-KT	–	–	–	0.8037 ± 0.0026	0.7832 ± 0.0005	0.7711 ± 0.0009	0.8487 ± 0.0008	0.8345 ± 0.0015	0.8301 ± 0.0010	0.7210 ± 0.0140	0.7156 ± 0.0115	0.7078 ± 0.0088
csKT	**0.7990 ± 0.0016**[a]	**0.7988 ± 0.0020**[a]	**0.7984 ± 0.0024**[a]	0.8045 ± 0.0005	0.8037 ± 0.0006	0.8024 ± 0.0008	**0.8555 ± 0.0003**	**0.8548 ± 0.0004**[a]	**0.8528 ± 0.0003**[a]	**0.7284 ± 0.0006**	**0.7278 ± 0.0007**[a]	**0.7265 ± 0.0009**[a]

[a] Indicate a statistically significant difference ($p < 0.01$) compared to the second-best result.

1. In the cold-start KT scenario, our csKT model consistently demonstrates stable performance in predicting student interaction sequences of various lengths 50,100 and 500 across four datasets. As the evaluated student interaction sequences increase, most baseline models, such as DKVMN, ATKT, LPKT, SAKT, AKT, simpleKT, sparseKT, FoLiBiKT, and DTransformer on BD2006, exhibit a significant decrease in AUC performance with longer sequences. Notably, the LPKT model shows strong performance in short sequence predictions; however, its performance declines the most when the sequence length extends to 500. This suggests that complex sequential KT models are not well-suited for cold-start KT tasks. Moreover, attention based models exhibit varying degrees of decline, with FoLiBiKT and AKT dropping to 5.53% and 3.91% on the BD2006 dataset respectively.
2. For student interaction sequences of varying evaluation lengths, our csKT model surpassed all baseline models across all four datasets. Specifically, on AL2005, csKT achieves a 5.32% and 6.43% improvement in AUC compared to simpleKT and sparseKT, respectively, for sequences of length 500.
3. DKT and DKT+ did not show significant performance degradation on the four datasets, but their AUC was much lower than csKT model. This is because sequence-based KT models, which do not include attention mechanisms, assign equal dependency weights to previous student interaction sequence and predict for longer sequences. However, compared to csKT model, these models find it's difficult to effectively simulate the complex knowledge state of students.

4.4.2. Computation efficiency

We analyze the computational efficiency between the baselines and csKT. 'Params(M)' and 'MFLOPs' quantify the number of model parameters (in millions) and the number of floating-point operations required to perform an inference (in millions), respectively. These are standard metrics for evaluating model size and operational efficiency. Following the standard comparison method,[2] all experiments are conducted under the same settings to ensure a fair comparison. We measure the parameters on the NIPS34 dataset. As observed in Table 5, our csKT is notably lightweight and generates slightly less computational overhead than simpleKT, which employs simple dot-product attention. This is due to two main reasons: First, the computational complexity of cone attention and dot-product attention is almost identical. Second, adding a kernel bias to the attention matrix before applying the softmax operation does not significantly increase the number of additional parameters. Details on the number of parameters for csKT in other datasets are shown in Fig. 4.

Table 5
Comparisons of computation efficiency.

Model	AKT	simpleKT	sparseKT	Dtransformer	csKT
Params(M)	4.41	1.62	1.73	1.12	1.57
MFLOPs	189.07	72.68	75.63	120.2	71.87

4.4.3. Sensitivity analysis

This section analyzes the impact of different hyperparameters of csKT on the final performance. The results on the NIPS34 dataset are shown in Fig. 5. We observe a positive correlation between the shadow sphere radius r and AUC performance, suggesting that a larger shadow sphere radius better models complex hierarchical relationships in the data. The scaling factor γ indicates that as its value increases, the initial improvement in performance reached a specific point before declining. This pattern implies that there is an optimal γ range in which the probability distribution of the output is neither too flat nor too concentrated, and balances certainty and variability in the forecast.

[2] https://github.com/Lyken17/pytorch-OpCounter.

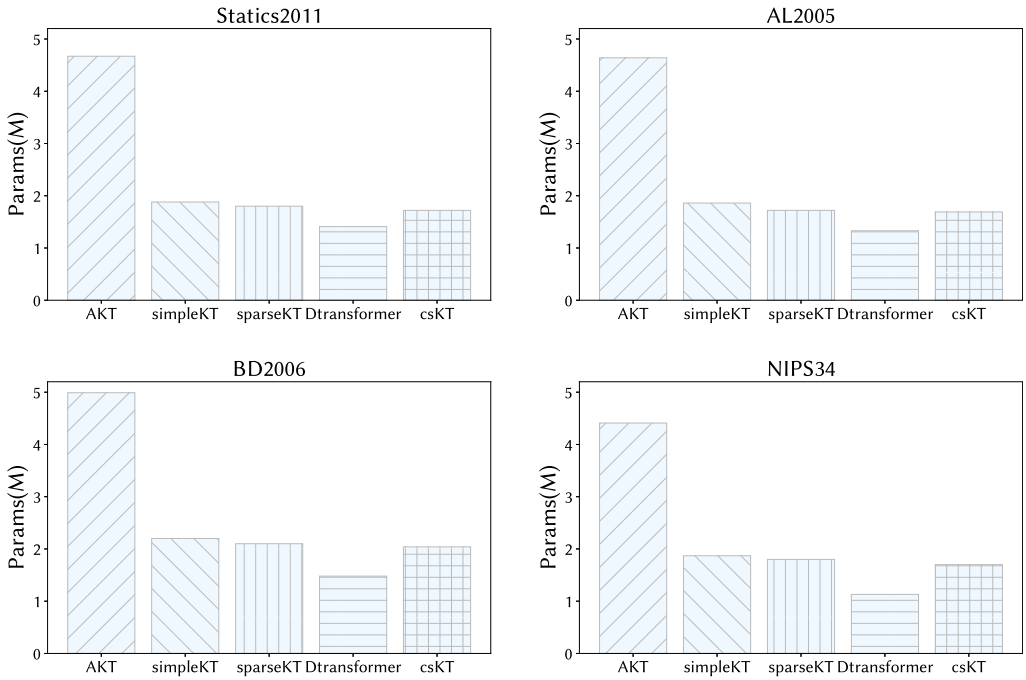

Fig. 4. Parameter comparison on four KT datasets.

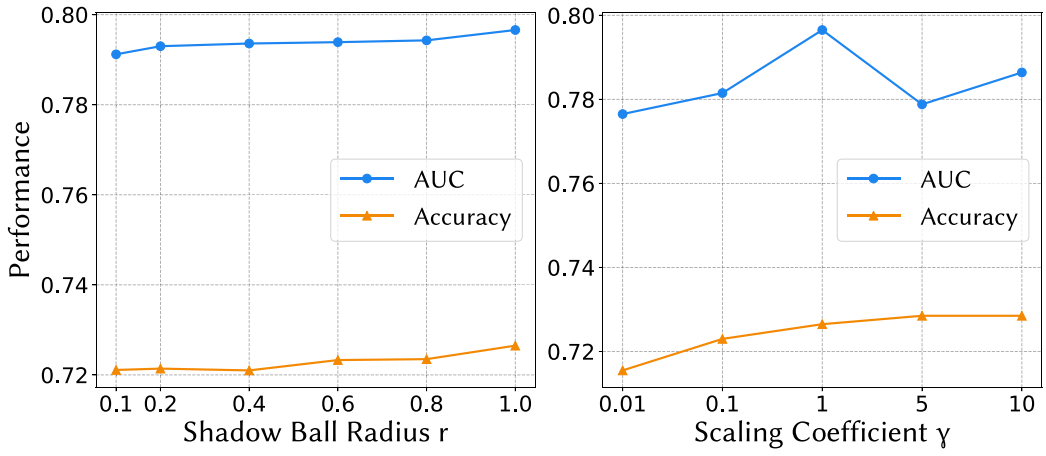

Fig. 5. AUC performance of csKT with different hyperparameters on NIPS34 dataset.

4.4.4. Kernel function analysis

In the cold-start setting, where effective knowledge decay modeling is crucial, we analyze how well different kernel functions perform in modeling student forgetting behavior, specifically comparing logarithmic (default in csKT), exponential (exp), and polynomial (poly) kernels (Wang et al., 2022). These decay functions have been extensively studied in cognitive science and memory research (Darwish & Zarras, 2023; Sikström, 2002).

As shown in Figs. 6 and 7, the log kernel consistently outperforms the exponential and polynomial kernels on all datasets and metrics. In Statics2011, the AUC of the log kernel is 0.8265 when the window size is 50, with exp and poly achieving 0.8230 and 0.8214 respectively, and this advantage expands when the context window is expanded to 500. csKT maintains stable prediction performance with the increasing interaction sequence. This stability is attributed to the log kernel's ability to better simulate the gradual nature of forgetting, avoiding the excessive decay of the exponential function or the strict bounds of the polynomial function. In the knowledge tracing scenario, the student's forgetting pattern follows a more natural logarithmic decay; and the log kernel function effectively balances the recent and historical interactions in knowledge tracing.

4.4.5. Ablation study

We investigated the effects of key components in csKT by constructing two model variants on four datasets. The term 'w/o' indicates the exclusion of a module from csKT; specifically, we removed cone attention (csKT w/o attn) and kernel bias (csKT w/o bias). According to Fig. 8, it is evident that (1) csKT consistently achieves the highest AUC scores in all scenarios compared to its variants. This result illustrates the significant drop in prediction performance when cone attention and kernel bias integral for capturing layers are omitted. Hence, it is crucial to incorporate the hierarchical relationship and location information of KCs for effective cold-start KT prediction tasks. (2) With an increased evaluation length, a comparison between csKT, csKT w/o attn, and csKT w/o bias reveals that kernel bias plays a significant role in prediction stability. Notably, when kernel bias is absent in csKT w/o bias, especially with a prediction length of 500, the performance of csKT on Statics2011, AL2005, BD2006, and NIPS34 datasets declined by 1.99%, 1.41%, 2.70%, and 0.97%, respectively.

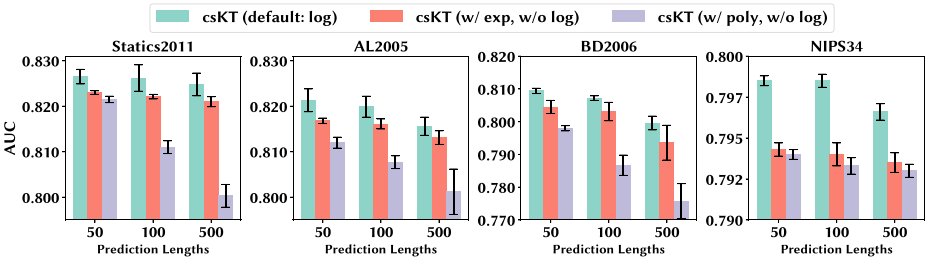

Fig. 6. AUC comparison for different kernel decay function.

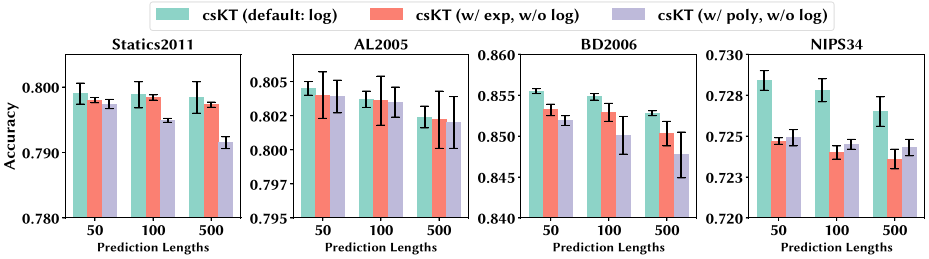

Fig. 7. Accuracy comparison for different kernel decay function.

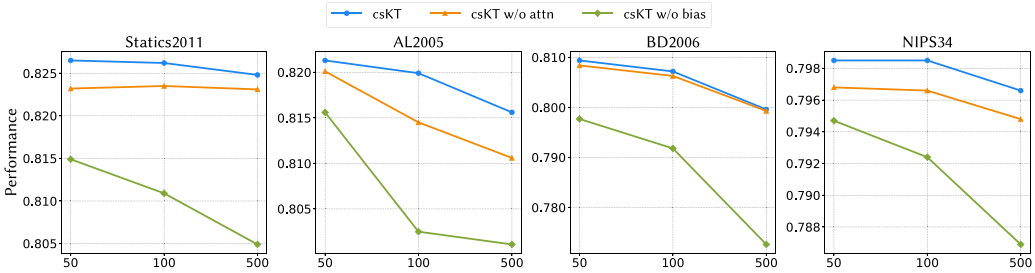

Fig. 8. AUC performance comparison of different variants. The **x**-axis represents the different evaluation lengths for 20 training length.

The ablation experiments demonstrate that both Cone Attention and Kernel Bias are essential for stable extrapolation across different learning stages, with Kernel Bias being particularly critical. While Cone Attention improves the model's initial performance, its effect becomes less prominent compared to Kernel Bias as sequence length increases. This aligns with our previous analysis showing that maintaining entropy invariance in the attention map is crucial for model extrapolation in cold-start scenarios.

4.5. Qualitative analysis

4.5.1. Attention visualization

We analyze and compare the interpretability of cone attention and dot-product attention in the NIPS34 dataset. Note that row 0 shows no attention scores as it represents the initial time step with no historical interactions available for modeling. Fig. 9(a) shows the attention weight distribution for the first 20 questions of a specific learner in the cold-start scenario, with the index representing questions at different time steps. Initially, in the first eight problems, cone and dot-product attention exhibited similar patterns, effectively capturing different levels of attention allocated to various interactions. However, when predicting question 9, the pattern changed. Dot-product attention primarily focused on questions 8 and 9, but it struggled to capture relevant questions before or after 8 due to the short sequence. Specifically, the similarity computation in dot-product attention makes it difficult to capture relevant information before question 8. In contrast, cone attention maintains focus on recent issues and captures important interactions before the question 8 in a short sequence. This suggests that both adjacent and distant information in short sequence can provide prediction insights into current problems for learners. Due to its hierarchical modeling ability, cone attention can effectively identify and utilize this information. More visualization insights can be found in Fig. 9(b)–(d).

4.5.2. Hyperbolic embedding visualization

We demonstrate the capability of hyperbolic space in capturing the latent hierarchical structures within data. Following the methodology in (Nickel & Kiela, 2017), we used hyperbolic embedding to process KCs from the NIPS34 dataset and projected them onto a two-dimensional Poincaré disk model for visual interpretation. As shown in Fig. 10, abstract concepts that cover a broad range of ideas in the original data, such as *Mathematics*, are positioned near the center of the circle. This central position signifies their foundational and pervasive importance. Conversely, more specific concepts, such as *Cumulative Frequency Diagram* and *Multiplying Fractions*, tend to be located around the circumference. These observations suggest that hyperbolic space is effective at capturing the hierarchical relationships among KCs, even in low-dimensional representations.

5. Conclusion

In this paper, we presented csKT, a novel knowledge tracing model designed to address cold-start challenges in educational systems. The

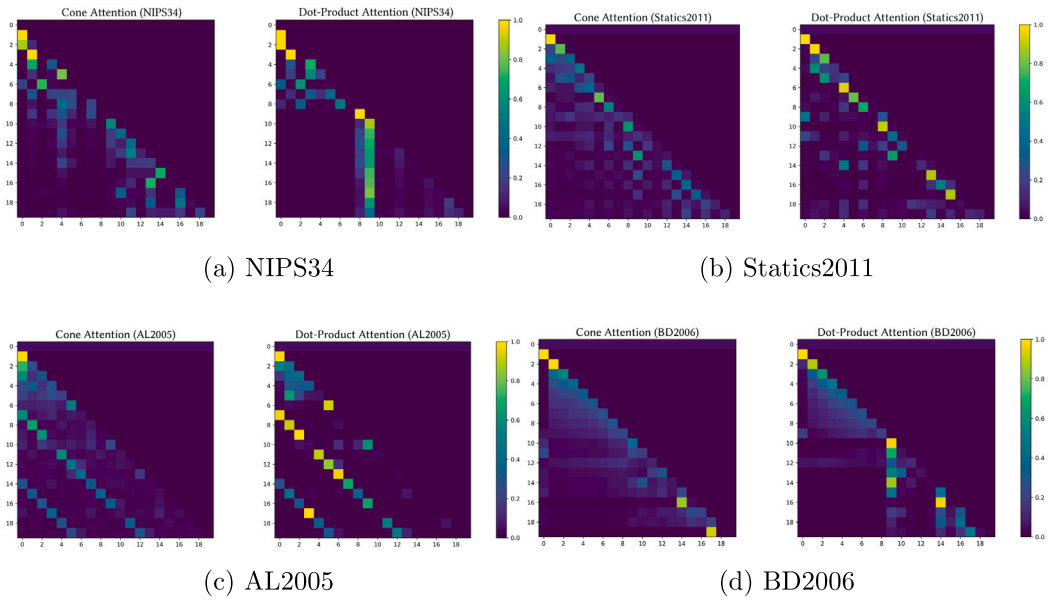

Fig. 9. Attention visualization for four KT datasets. The **x**-axis and **y**-axis index represent questions in the student interaction sequence.

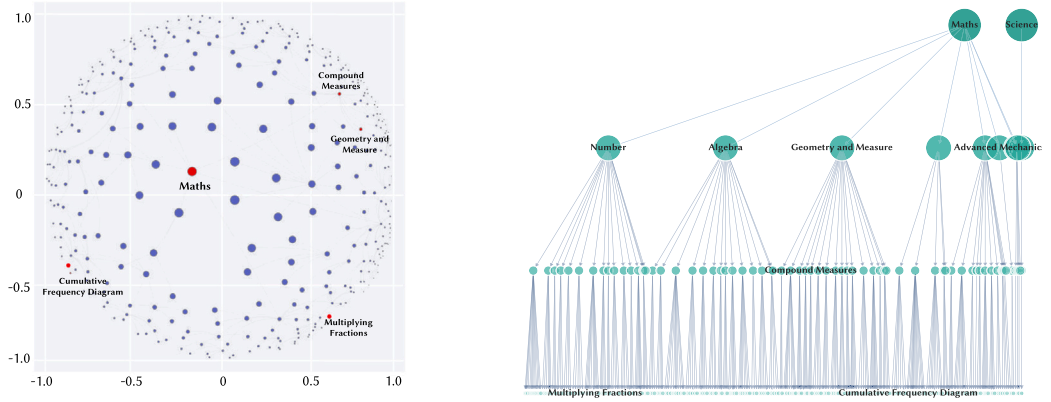

Fig. 10. Hyperbolic embedding visualizations and hierarchical structure for NIPS34 dataset.

key innovation of csKT lies in its 'train short, test long' approach, enabling effective training on short learner interaction sequences while maintaining performance on longer sequences. We introduced kernel bias to enhance csKT's adaptability, allowing prediction for growing interaction sequences. Additionally, we proposed a cone attention mechanism in hyperbolic space to model hierarchical relationships between KCs, capturing fine-grained features even in short-sequence scenarios.

The main contributions of csKT are: (1) introducing kernel bias for efficient prediction of extended sequences, (2) using cone attention to capture hierarchical relationships in limited interaction sequences, and (3) demonstrating csKT's effectiveness across multiple datasets, outperforming existing models.

Despite promising results, csKT has limitations. It currently focuses on learners' performance based on historical responses, without incorporating cognitive or emotional engagement. Future work could explore integrating these dimensions to enhance predictive accuracy. Another direction includes capturing context-dependent learning behaviors by incorporating more sophisticated contextual features.

In summary, csKT contributes significantly to knowledge tracing by improving cold-start performance and holding practical value in intelligent tutoring systems and personalized learning environments. Its capability to handle short sequences while maintaining robustness for longer ones makes csKT a promising tool for adaptive learning solutions.

CRediT authorship contribution statement

Youheng Bai: Conceptualization, Methodology, Software, Data curation, Writing – original draft, Writing – review & editing. **Xueyi Li:** Methodology, Software, Visualization. **Zitao Liu:** Data curation, Validation, Writing – original draft. **Yaying Huang:** Methodology, Investigation, Visualization. **Teng Guo:** Writing – review & editing,Validation. **Mingliang Hou:** Conceptualization, Writing – review & editing. **Feng Xia:** Conceptualization, Writing – review & editing. **Weiqi Luo:** Writing – review & editing, Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by National Key R&D Program of China, under Grant No. 2022YFC3303600 and in part by Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University, China (2022LSYS003).

Data availability

The data used in this study is publicly available and accessible.

References

Abdelrahman, G., & Wang, Q. (2019). Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019* (pp. 175–184).

Alharbe, N., Rakrouki, M. A., & Aljohani, A. (2023). A collaborative filtering recommendation algorithm based on embedding representation. *Expert Systems with Applications*, *215*, Article 119380.

Cai, D., Qian, S., Fang, Q., Hu, J., & Xu, C. (2023). User cold-start recommendation via inductive heterogeneous graph neural network. *ACM Transactions on Information Systems*, *41*(3), 1–27.

Chen, T., Zhang, W., Lu, Q., Chen, K., Zheng, Z., & Yu, Y. (2012). SVDFeature: a toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research*, *13*, 3619–3622.

Chi, T., Fan, T., Ramadge, P. J., & Rudnicky, A. (2022). KERPLE: Kernelized relative positional embedding for length extrapolation. In *Advances in Neural Information Processing Systems 2022, New Orleans, La, USA, November 28 - December 9, 2022*.

Choi, Y., Lee, Y., Cho, J., Baek, J., Kim, B., Cha, Y., Shin, D., Bae, C., & Heo, J. (2020). Towards an appropriate query, key, and value computation for knowledge tracing. In *L@s'20: Seventh ACM Conference on Learning @ Scale, Virtual Event, USA, August 12-14, 2020* (pp. 341–344).

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

Cui, C., Yao, Y., Zhang, C., Ma, H., Ma, Y., Ren, Z., Zhang, C., & Ko, J. (2024). DGEKT: a dual graph ensemble learning method for knowledge tracing. *ACM Transactions on Information Systems*, *42*(3), 1–24.

Darban, Z. Z., & Valipour, M. H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. *Expert Systems with Applications*, *200*, Article 116850.

Darwish, M. A., & Zarras, A. (2023). Digital forgetting using key decay. In *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing* (pp. 34–41).

Fang, P., Harandi, M., & Petersson, L. (2021). Kernel methods in hyperbolic spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 10665–10674).

Feng, J., Xia, Z., Feng, X., & Peng, J. (2021). RBPR: A hybrid model for the new user cold start problem in recommender systems. *Knowledge-Based Systems*, *214*, Article 106732.

Furusawa, T. (2024). Mean field theory in deep metric learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Ghosh, A., Heffernan, N. T., & Lan, A. S. (2020). Context-aware attentive knowledge tracing. In *KDD '20: the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020* (pp. 2330–2339).

Gretton, A., Herbrich, R., Smola, A., Bousquet, O., Schölkopf, B., & Hyvärinen, A. (2005). Kernel methods for measuring independence. *Journal of Machine Learning Research*, *6*(12), 2075–2129.

Guo, X., Huang, Z., Gao, J., Shang, M., Shu, M., & Sun, J. (2021). Enhancing knowledge tracing via adversarial training. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021* (pp. 367–375).

Hou, J.-L., Yu, F.-J., & Lin, R.-K. (2006). A knowledge component analysis model based on term frequency and correlation analysis. *Journal of Computer Information Systems*, *46*(4), 64–77.

Huang, T., Hu, S., Yang, H., Geng, J., Li, Z., Xu, Z., & Ou, X. (2024). Response speed enhanced fine-grained knowledge tracing: A multi-task learning perspective. *Expert Systems with Applications*, *238*, Article 122107.

Huang, S., Liu, Z., Zhao, X., Luo, W., & Weng, J. (2023). Towards robust knowledge tracing models via k-sparse attention. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023* (pp. 2441–2445).

Huang, C., Wei, H., Huang, Q., Jiang, F., Han, Z., & Huang, X. (2024). Learning consistent representations with temporal and causal enhancement for knowledge tracing. *Expert Systems with Applications*, *245*, Article 123128.

Im, Y., Choi, E., Kook, H., & Lee, J. (2023). Forgetting-aware linear bias for attentive knowledge tracing. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023* (pp. 3958–3962).

Jeevamol, J., & Renumol, V. (2021). An ontology-based hybrid e-learning content recommender system for alleviating the cold-start problem. *Education and Information Technologies*, *26*, 4993–5022.

Jung, H., Yoo, J., Yoon, Y., & Jang, Y. (2023). Language proficiency enhanced knowledge tracing. *vol. 13891*, In *Augmented Intelligence and Intelligent Tutoring Systems - 19th International Conference, ITS 2023, Corfu, Greece, June 2-5, 2023, proceedings* (pp. 3–15).

Käser, T., Klingler, S., Schwing, A. G., & Gross, M. H. (2017). Dynamic Bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, *10*(4), 450–462.

Li, H., Ding, W., & Liu, Z. (2020). Identifying at-risk K-12 students in multimodal online environments: A machine learning approach. In *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully Virtual Conference, July 10-13, 2020*. International Educational Data Mining Society.

Lika, B., Kolomvatsos, K., & Hadjiefthymiades, S. (2014). Facing the cold start problem in recommender systems. *Expert Systems with Applications*, *41*(4), 2065–2073.

Liu, Q., Huang, Z., Yin, Y., Chen, E., Xiong, H., Su, Y., & Hu, G. (2021). EKT: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, *33*(1), 100–115.

Liu, Z., Kong, X., Chen, H., Liu, S., & Yang, Z. (2023). MOOC-BERT: Automatically identifying learner cognitive presence from MOOC discussion data. *IEEE Transactions on Learning Technologies*, *16*(4), 528–542.

Liu, Z., Kong, W., Peng, X., Yang, Z., Liu, S., Liu, S., & Wen, C. (2023). Dual-feature-embeddings-based semi-supervised learning for cognitive engagement classification in online course discussions. *Knowledge-Based Systems*, *259*, Article 110053.

Liu, Z., Liu, Q., Chen, J., Huang, S., Gao, B., Luo, W., & Weng, J. (2023). Enhancing deep knowledge tracing with auxiliary tasks. In *Proceedings of the ACM Web Conference 2023* (pp. 4178–4187).

Liu, Z., Liu, Q., Chen, J., Huang, S., & Luo, W. (2023). simpleKT: A simple but tough-to-beat baseline for knowledge tracing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, kigali, rwanda, May 1-5, 2023*.

Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J., & Luo, W. (2022). pyKT: A Python Library to benchmark deep learning based knowledge tracing models. In *Advances in Neural Information Processing Systems, 2022, NeurIPS 2022, New Orleans, la, USA, November 28 - December 9, 2022*.

Liu, S., Liu, S., Liu, Z., Peng, X., & Yang, Z. (2022). Automated detection of emotional and cognitive engagement in MOOC discussions to predict learning achievement. *Computers & Education*, *181*, Article 104461.

Liu, Z., Xu, G., Liu, T., Fu, W., Qi, Y., Ding, W., Song, Y., Guo, C., Kong, C., & Yang, S. (2020). Dolphin: a spoken language proficiency assessment system for elementary education. In *Proceedings of The Web Conference 2020* (pp. 2641–2647).

Ma, H., Yang, Y., Qin, C., Yu, X., Yang, S., Zhang, X., & Zhu, H. (2024). HD-KT: Advancing robust knowledge tracing via anomalous learning interaction detection. In *Proceedings of the ACM on Web Conference 2024* (pp. 4479–4488).

Mao, S., Zhan, J., Wang, Y., & Jiang, Y. (2023). Improving knowledge tracing via considering two types of actual differences from exercises and prior knowledge. *IEEE Transactions on Learning Technologies*, *16*(3), 324–338.

Mavroforakis, M. E., & Theodoridis, S. (2006). A geometric approach to support vector machine (SVM) classification. *IEEE Transactions on Neural Networks*, *17*(3), 671–682.

Nakagawa, H., Iwasawa, Y., & Matsuo, Y. (2021). Graph-based knowledge tracing: Modeling student proficiency using graph neural networks. *Web Intelligence*, *19*(1–2), 87–102.

Narducci, F., Basile, P., Musto, C., Lops, P., Caputo, A., de Gemmis, M., Iaquinta, L., & Semeraro, G. (2016). Concept-based item representations for a cross-lingual content-based recommendation process. *Information Sciences*, *374*, 15–31.

Nguyen, H., Wang, Y., Stamper, J., & McLaren, B. M. (2019). Using knowledge component modeling to increase domain understanding in a digital learning game. *International Educational Data Mining Society*.

Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems, December 4-9, 2017, Long Beach, CA, USA* (pp. 6338–6347).

Pandey, S., & Karypis, G. (2019). A self attentive model for knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, montréAl, Canada, July 2-5, 2019*.

Pandey, S., & Srivastava, J. (2020). RKT: Relation-aware self-attention for knowledge tracing. In *CIKM '20: the 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020* (pp. 1205–1214).

Pardos, Z. A., & Heffernan, N. T. (2011). KT-IDEM: Introducing item difficulty to the knowledge tracing model. *vol. 6787*, In *User modeling, adaption and personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. proceedings* (pp. 243–254).

Patel, R., & Thakkar, P. (2022). Addressing item cold start problem in collaborative filtering-based recommender systems using auxiliary information. *vol. 2*, In *IOT with Smart Systems: proceedings of ICTIS 2022* (pp. 133–142). Springer.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in Neural Information Processing Systems, December 7-12, 2015, Montreal, Quebec, Canada* (pp. 505–513).

Pliakos, K., Joo, S.-H., Park, J. Y., Cornillie, F., Vens, C., & Van den Noortgate, W. (2019). Integrating machine learning into item response theory for addressing the cold start problem in adaptive learning systems. *Computers & Education*, *137*, 91–103.

Press, O., Smith, N. A., & Lewis, M. (2022). Train short, test long: Attention with linear biases enables input length extrapolation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Qin, Z., Sun, W., Deng, H., Li, D., Wei, Y., Lv, B., Yan, J., Kong, L., & Zhong, Y. (2022). CosFormer: Rethinking softmax in attention. In *The tenth international conference on learning representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests*. ERIC.

Shen, S., Huang, Z., Liu, Q., Su, Y., Wang, S., & Chen, E. (2022). Assessing student's dynamic knowledge state by exploring the question difficulty effect. In *SIGIR '22: the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022* (pp. 427–437).

Shen, S., Liu, Q., Chen, E., Huang, Z., Huang, W., Yin, Y., Su, Y., & Wang, S. (2021). Learning process-consistent knowledge tracing. In *KDD '21: the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021* (pp. 1452–1460).

Sikström, S. (2002). Forgetting curves: implications for connectionist models. *Cognitive Psychology*, *45*(1), 95–152.

Stamper, J., & Pardos, Z. A. (2016). The 2010 KDD cup competition dataset: Engaging the machine learning community in predictive learning analytics. *Journal of Learning Analytics*, *3*(2), 312–316.

Steif, P., & Bier, N. (2014). Oli engineering statics-fall 2011, february 2014.

Su, Y., Liu, Q., Liu, Q., Huang, Z., Yin, Y., Chen, E., Ding, C. H. Q., Wei, S., & Hu, G. (2018). Exercise-enhanced sequential modeling for student performance prediction. In *Proceedings of the Thirty-second AAAI Conference on Artificial Intelligence* (pp. 2435–2443).

Sun, J., Yu, F., Liu, S., Luo, Y., Liang, R., & Shen, X. (2023). Adversarial bootstrapped question representation learning for knowledge tracing. In *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023, Ottawa, Canada, 29 October 2023- 3 November 2023* (pp. 8016–8025).

Sun, J., Zou, R., Liang, R., Gao, L., Liu, S., Li, Q., Zhang, K., & Jiang, L. (2022). Ensemble knowledge tracing: Modeling interactions in learning process. *Expert Systems with Applications*, *207*, Article 117680.

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2023). Efficient transformers: A survey. *ACM Computing Surveys*, *55*(6), 109:1–109:28.

Tong, H., Wang, Z., Zhou, Y., Tong, S., Han, W., & Liu, Q. (2022). Introducing problem schema with hierarchical exercise graph for knowledge tracing. In *SIGIR '22: the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, madrid, Spain, July 11 - 15, 2022* (pp. 405–415).

Tseng, A., Yu, T., Liu, T. J. B., & Sa, C. D. (2023). Coneheads: Hierarchy aware attention. In *Advances in Neural Information Processing Systems, NeurIPS 2023, New orleans, LA, USA, December 10 - 16, 2023*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31th International Conference on Neural Information Processing Systems* (pp. 6000–6010).

van der Velde, M., Sense, F., Borst, J. P., & Rijn, H. v. (2024). Large-scale evaluation of cold-start mitigation in adaptive fact learning: Knowing "what" matters more than knowing "who". *User Modeling and User-Adapted Interaction*, 1–25.

Villano, M. (1992). Probabilistic student models: Bayesian belief networks and knowledge space theory. In *Intelligent Tutoring Systems: Second International Conference* (pp. 491–498).

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. (2020). Instructions and guide for diagnostic questions: The neurips 2020 education challenge. arXiv preprint arXiv:2007.12061.

Wang, Y., Li, D., Xu, Y., & Wang, H. (2022). Online hybrid kernel learning machine with dynamic forgetting mechanism. In *International Conference on Emerging Networking Architecture and Technologies* (pp. 273–285). Springer.

Wang, W., Ma, H., Zhao, Y., & Li, Z. (2024). Pre-training question embeddings for improving knowledge tracing with Self-supervised Bi-graph Co-contrastive learning. *ACM Transactions on Knowledge Discovery from Data*, *18*(4), 74:1–74:20.

Wang, X., Peng, Z., Wang, S., Yu, P. S., Fu, W., Xu, X., & Hong, X. (2020). CDLFM: cross-domain recommendation for cold-start users via latent feature mapping. *Knowledge and Information Systems*, *62*, 1723–1750.

Wei, T., & Chow, T. W. (2023). FGCR: Fused graph context-aware recommender system. *Knowledge-Based Systems*, *277*, Article 110806.

Wu, Z., Huang, L., Huang, Q., Huang, C., & Tang, Y. (2022). SGKT: Session graph-based knowledge tracing for student performance prediction. *Expert Systems with Applications*, *206*, Article 117681.

Wu, L., Quan, C., Li, C., Wang, Q., Zheng, B., & Luo, X. (2019). A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems (TOIS)*, *37*(2), 1–29.

Xu, S., Ge, Y., Li, Y., Fu, Z., Chen, X., & Zhang, Y. (2023). Causal collaborative filtering. In *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR 2023, Taipei, Taiwan, 23 July 2023* (pp. 235–245).

Xue, F., He, X., Wang, X., Xu, J., Liu, K., & Hong, R. (2019). Deep item-based collaborative filtering for Top-N Recommendation. *ACM Transactions on Management Information Systems*, *37*(3), 33:1–33:25.

Yang, Y., Shen, J., Qu, Y., Liu, Y., Wang, K., Zhu, Y., Zhang, W., & Yu, Y. (2020). GIKT: A graph-based interaction model for knowledge tracing. *vol. 12457*, In *Machine learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020* (pp. 299–315).

Yang, S., Yu, X., Tian, Y., Yan, X., Ma, H., & Zhang, X. (2023). Evolutionary neural architecture search for transformer in knowledge tracing. In *Advances in Neural Information Processing Systems, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yeung, C., & Yeung, D. (2018). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale, London, UK, June 26-28, 2018* (pp. 5:1–5:10).

Yin, Y., Dai, L., Huang, Z., Shen, S., Wang, F., Liu, Q., Chen, E., & Li, X. (2023). Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023* (pp. 855–864).

Zhang, J., Shi, X., King, I., & Yeung, D. (2017). Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017* (pp. 765–774).

Zhao, J., Bhatt, S. P., Thille, C., Gattani, N., & Zimmaro, D. (2020). Cold start knowledge tracing with attentive neural turing machine. In *L@S'20: Seventh ACM Conference on Learning @ Scale, Virtual Event, USA, August 12-14, 2020* (pp. 333–336).