sustainability

MDPI

Article

DCKT: A Novel Dual-Centric Learning Model for Knowledge Tracing

Yixuan Chen, Shuang Wang, Fan Jiang, Yaxin Tu and Qionghao Huang *

Key Laboratory of Intelligent Education Technology and Application of Zhejiang Province, Zhejiang Normal University, Jinhua 321004, China
* Correspondence: 2018010055@m.scnu.edu.cn

Abstract: Knowledge tracing (KT), aiming to model learners' mastery of a concept based on their historical learning records, has received extensive attention due to its great potential in realizing personalized learning in intelligent tutoring systems. However, most existing KT methods focus on a single aspect of knowledge or learner, not paying careful attention to the coupling influence of knowledge and learner characteristics. To fill this gap, in this paper, we explore a new paradigm for the KT task by exploiting the coupling influence of knowledge and learner. A novel model called Dual-Centric Knowledge Tracing (DCKT) is proposed to model knowledge states through two joint tasks of knowledge modeling and learner modeling. In particular, we first generate concept embeddings in abundant knowledge structure information via a pretext task (knowledge-centric): unsupervised graph representation learning. Then, we deeply measure learners' prior knowledge the knowledge-enhanced representations and three predefined educational priors for discriminative feature enhancement. Furthermore, we design a forgetting-fusion transformer (learner-centric) to simulate the declining trend of learners' knowledge proficiency over time, representing the common forgetting phenomenon. Extensive experiments were conducted on four public datasets, and the results demonstrate that DCKT could achieve better knowledge tracing results over all datasets via a dual-centric modeling process. Additionally, DCKT can learn meaningful question embeddings automatically without manual annotations. Our work indicates a potential future research direction for personalized learner modeling, which is of both accuracy and high interpretability.

Keywords: knowledge tracing; learner modeling; prerequisite inferences; forgetting behaviors; transformer; personalized learning

Citation: Chen, Y.; Wang, S.; Jiang, F.; Tu, Y.; Huang, Q. DCKT: A Novel Dual-Centric Learning Model for Knowledge Tracing. *Sustainability* **2022**, *14*, 16307. https://doi.org/10.3390/su142316307

Academic Editor: Eila Jeronen

Received: 21 November 2022
Accepted: 2 December 2022
Published: 6 December 2022

1. Introduction

The past few decades have witnessed the rapid development of online education platforms to improve learning efficiency while minimizing the cost of education [1], such as massive open online courses (MOOCs) and intelligent tutoring systems (ITS). Knowledge tracing (KT) [2] is an essential task in online education platforms. Given learners' past learning records, it aims to track and quantify their knowledge state over time to make accurate predictions on future performance. Concretely, supposing there are a set of t discrete time indices, we use the following generic model to represent a learner's hidden knowledge state and historical performance:

$$f(h_{t-1}) = h_t, \qquad g(h_t) = r_t, \tag{1}$$

where the hidden variable h_t denotes the learner's knowledge state at time step t, and the binary value $r_t \in \{0, 1\}$ denotes the predicted learner's response at the current question (with 1 representing a correct answer and 0 representing an incorrect answer). $f(\cdot)$ and $g(\cdot)$ are the two functions that characterize the learners' knowledge evolution and predict their future responses, respectively. Once the knowledge proficiency is precisely estimated through KT, learners can make up for their weaknesses in time and thus maximize the

learning outcome. Due to its great potential for personalized learning, KT attracts increasing interest and is widely used in the scientific and educational communities [3–5].

Research efforts on KT tasks usually focus on a single aspect of knowledge or learners for different purposes, and Table 1 briefly summarizes these models from the two aspects, respectively. On the one side, classical KT models, such as Bayesian knowledge tracing (BKT) [2], deep knowledge tracing (DKT) [6], and DKT-forget [7], concentrate on mining learners' interaction information and estimate hidden knowledge states from their learning performance data, and pay less attention to knowledge estimation. On the other side, some models pay more attention to knowledge modeling. For example, prerequisite-driven deep knowledge tracing (PDKT-C) [8], structure-based knowledge tracing (SKT) [9], PQR-LKA [10], and AKT [4] highlight the importance of knowledge structures or the need to learn embedding representations with plentiful domain knowledge but assess learners' knowledge in simple ways, such as a simple RNN. However, knowledge and learner characteristics have a combined effect during the process of learners' cognition and knowledge growth; ignoring the knowledge factor or the learner factor will lead to a decrease in the prediction accuracy of knowledge tracing, and only combining knowledge and learner factors can make more accurate predictions (we show this in the experiment section).

Table 1. A comparison of knowledge tracing models.

KT Model	Knowledge Component		Learner Component	
	Knowledge Structure	Question Rank	Forgetting	Prior
BKT [2]	×	×	×	×
DKT [6]	×	×	×	×
DKT-forget [7]	×	×	✓	×
PDKT-C [8]	✓	×	×	×
SKT [9]	✓	×	×	×
PQRLKA [10]	✓	×	×	×
AKT [4]	×	✓	✓	×
DCKT (ours)	✓	✓	✓	✓

From the perspective of *learner modeling* [3,11] in education theory, knowledge tracing is a typical learner modeling technique whose process involves human knowledge and learning. For better illustration, we give an example of knowledge tracing in Figure 1. The middle part depicts the learning process, where a learner practices a sequence of questions $\{q_1, q_2, \ldots, q_8\}$ associated with a concept set $\{c_1, c_2, c_3, c_4\}$, and after this, the learner is informed whether their answers are correct or not. In addition to these observable phenomena, there are implicit but non-negligible details in a KT task. As shown in the upper dotted box, the learner's knowledge state of all concepts constantly changes during learning, and the whole process reflects their cognitive evolution. Moreover, the learner's knowledge proficiency of a specific concept shows a declining trend since their last practice, which is attributed to human memory decay in cognitive science, known as forgetting behaviors. The bottom dotted box in Figure 1 shows the latent knowledge structure, where the colored undirected lines represent association relations and the black directed lines represent prerequisite relations. We can observe that the knowledge components are linked by multiple relations, including association relations between questions and concepts and prerequisite relations within concepts.

Therefore, it is necessary to give equal importance to the characteristics of knowledge and learners and to integrate the process of knowledge modeling and learner modeling effectively for knowledge tracing. Although both aspects are somewhat involved, previous deep-learning-based KT methods cannot meet this requirement due to three major challenges. First, the knowledge structure, which is inextricably linked to modeling domain knowledge, is inherent and implicit in the KT scenario. For instance, each question may relate to multiple concepts, and different concepts also have potential correlations, thus making it difficult to learn the complex relational dependencies between these knowledge

components. Yang et al. [12] propose a graph-based Interaction model for Knowledge Tracing (GIKT) to learn the graph embeddings of questions and skills from high-order relations. However, the defined relationships between concepts and questions rely on many expert annotations. Second, prior knowledge is the basis of learners' differentiated knowledge proficiency and an important criterion for evaluating personalized learning, which needs to be measured based on learning performance data. Third, knowledge decline is an inevitable phenomenon in the learning process, commonly attributed to forgetting behaviors. Hence, it is also a huge challenge to be solved in KT tasks.

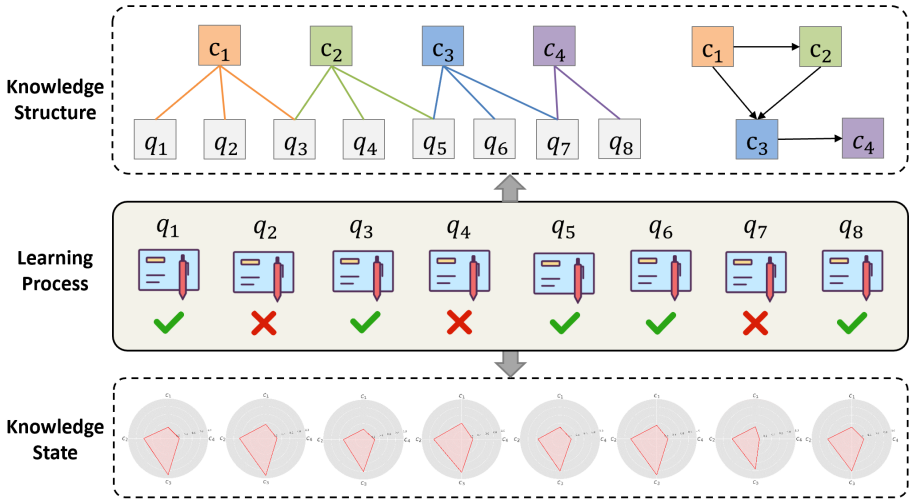


Figure 1. An example of knowledge tracing.

To address the above challenges, we propose a novel KT framework called Dual-Centric Knowledge Tracing (DCKT) to integrate the two subtasks of knowledge discovery and knowledge tracing. The purpose of the former is to serve the latter. Specifically, since knowledge structure is implicit and static, we exploit knowledge structure features as crucial domain information to enhance the KT task. Following this idea, concept embeddings are generated via a well-designed pretext task, which constructs the knowledge structure through an unsupervised representation learning method without needing manual labels. In particular, we compute a transition probability matrix from the large-scale learning logs based on specific statistics. A concept prerequisite graph is constructed with the matrix, and high-order relations between concepts in the concept prerequisite graph are learned with graph neural networks (GNNs). Notably, skill-level KT datasets identify each question by its underlying concept based on the Q-matrix, combined with graph representation learning for prerequisite relationships within concepts. Thus, the produced concept embeddings contain a wealth of knowledge structure information. Then, to measure learners' prior knowledge, we generate knowledge-enhanced embedding representations and represent learners' knowledge proficiency using three predefined educational priors to enhance the discriminative features. Therefore, we are more capable of capturing learners' personalized traits at a finer-grained level from long-term behaviors. Finally, for modeling forgetting behaviors over a long study period, we design a forgetting-fusion transformer to determine the rate of learners' knowledge decline over time.

To sum up, the main contributions of this work are as follows:

- We propose a novel KT model, namely DCKT, which combines the task of knowledge discovery with knowledge tracing and leverages the former to benefit the latter, i.e., a knowledge-centric module, called concept graph representation learning, and a learner-centric module, called KT, with fine-grained forgetting behaviors modeling.
- We explore an unsupervised representation learning method that automatically infers domain prerequisites and learns graph representations for concepts, which can be leveraged to enhance knowledge tracing.

- We design a novel forgetting-fusion transformer to model the forgetting behaviors of learners with exponential decay attention to quantifying the forgetting effect during learning.
- We conduct extensive experiments to evaluate the performance of our proposed DCKT model on four public KT datasets. The results demonstrate the effectiveness of DCKT in concept prerequisite inferences and knowledge tracing.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 lists important problem definitions and notations in the research. Section 4 introduces our proposed KT model. Section 5 details the four research questions and the experiment settings. Section 6 presents the experiment results and discusses these research questions. Finally, the paper concludes in Section 7.

2. Related Work

2.1. Knowledge Tracing

Generally speaking, existing KT work can be divided into two categories: traditional statistics-based models and deep-learning-based models. The first class of traditional KT models is Bayesian knowledge tracing (BKT) [2], which uses a probabilistic graphical model such as hidden Markov models (HMMs) to track the latent knowledge state. BKT-based methods assume that the current knowledge state is determined by the state at the previous time step. They model the knowledge state as a set of binary latent variables based on *mastery learning* [13]. The second traditional KT category comprises factor analysis models with logistic regression, such as the Additive Factor Model (AFM) [14], Performance Factor Analysis (PFA) [15], and Knowledge Tracing Machine (KTM) [16]. The key idea of these models is to predict the response performance by learning a logistic function, which considers a wide range of factors such as learner, concept, item, or learning environment. Although both statistical KT models have good interpretability, the limitations of manual tag reliance and oversimplified assumptions prevent them from mining the complex knowledge state of learners.

With the huge breakthrough of deep learning in various fields, Piech et al. [6] introduced deep learning techniques into KT for the first time. Deep Knowledge Tracing (DKT) employs recurrent neural networks (RNNs) or their variant Long Short-Term Memory (LSTM) on learning interaction sequences and models the knowledge state as a high-dimensional hidden state at each time step, showing great potential for learning performance prediction. Dynamic Key-Value Memory Networks (DKVMN) [17] use a memory network to enrich the hidden variable representations in the KT task. They design two matrices for tracking knowledge state over time, with a static matrix called key to store the latent concepts underlying all questions and a dynamic matrix called value to store and update the mastery level of each concept through reading and writing operations. To model complex learning behaviors in real-world education scenarios, many variants focus on integrating rich features as side information for KT. For the personalized modeling of learners, Deep Knowledge Tracking for Dynamic Student Classification (DKT-DSC) [18] extends DKT by clustering learners with similar ability levels using the K-means clustering algorithm and incorporating the clustering results into the model input. In addition to these student-specific factors, some work explores the inclusion of knowledge characteristics to enhance the KT task. For example, PDKT-C [8] leverages the prerequisite relations between latent concepts as additional constraints, and EERNN [19], EKT [20], and MathBERT [21] incorporate textual features of questions as additional input to the KT model.

Various emerging techniques have been recently applied to tackle the KT problem. Inspired by the powerful capabilities of Transformer [22] in time series analysis, Self-Attentive Knowledge Tracing (SAKT) [23] introduced an attention mechanism into KT for the first time. Later, Context-aware Attentive Knowledge Tracing(AKT) [4] modified the original scaled dot-product attention and proposed monotonic attention to learn context-aware representations. It computes attention weights for questions by simulating the forgetting effect as a time distance measure. However, although the common idea of

attention-based KT models is to learn attention weights of key knowledge components, most works ignore the influence of learners' personalization characteristics on their learning. In this regard, Convolutional Knowledge Tracing (CKT) [24] leverages convolutional neural networks (CNNs) to model the individualization of learners based on their individualized prior knowledge and learning rates. Considering various graph structures that naturally exist in KT, graph neural networks (GNNs) are designed to process these graph-structured data mining relational structures for better embedding representations, including the GKT [25], GIKT [12], SGKT [26], and Bi-CLKT [27] models. While knowledge tracing for deep learning has shown promising performance results, limited work explicitly defines the KT task from knowledge and learner perspectives and emphasizes the combined role of both in the modeling process.

2.2. Concept Prerequisite Inferences in KT

Due to the fundamental role that concepts play in human cognitive processes [28], the inferences of concept prerequisites have been studied in various educational contexts. For example, Wang et al. [29] leverage prerequisites to construct concept maps from textbooks. Pan et al. [30] design a representational learning-based method and different leveraged features to infer the prerequisite relation between course concepts in MOOCs. To alleviate the manually labeled reliance on course prerequisites, Roy et al. [31] propose a new supervised learning method capable of identifying unknown concept prerequisites with labeled concept prerequisite data and course prerequisites.

For knowledge modeling, many deep learning KT models automatically attempt to infer concept prerequisite relationships. Chen et al. [32] propose a novel algorithm named COMMAND to simultaneously learn a concept prerequisite graph and a student model from performance data, which models the concept prerequisite relations as a Bayesian network through a two-stage learning process. To address the data sparsity issue, PDKT-C [8] advocates for incorporating knowledge structure information into the KT model, especially the prerequisite relations between pedagogical concepts. It first models prerequisites as ordered pairs, then combines them with a proper mathematical formulation to serve as model constraints. Inspired by the success of GNNs in relation learning, GKT [25] utilizes the graph-structured nature of knowledge as a relational inductive bias and reformulates the KT task as a time series node-level classification problem in GNNs. This work proposed statistics-based and learning-based approaches to construct latent knowledge graphs, where nodes represent concepts and edges represent the dependency relation between concepts, such as similarity and prerequisite relations. Unlike the graph data in GKT, which only involve a single relation between concepts, SKT [9] captures multiple relations between concepts and learns graph embeddings through information propagation. An increasing number of KT models extract knowledge structures to enrich embedding representations, but there is limited work considering the static nature of knowledge and serving domain knowledge as an important supplement to dynamic KT tasks.

2.3. Forgetting Behaviors in KT

In cognitive psychology studies [33,34], there is broad evidence showing that forgetting behaviors significantly impact learners' knowledge proficiency and post-learning performance. Moreover, the well-known *Ebbinghaus forgetting curve theory* [35] shows that learners tend to forget what they have learned at an exponentially decaying rate. Therefore, forgetting modeling is highly active in many KT models. Nedungadi and Remya [36] extended BKT by incorporating forgetting behaviors into their model, which is viewed as knowledge decline over time and measured by an exponentially decaying function. To characterize more complex forgetting behaviors in the entire sequence, DKT-forget [7] adds three types of forgetting features that reflect both the learning and forgetting effects to the DKT model. Similar to the idea of DKT-forget, a probabilistic matrix factorization model called Knowledge Proficiency Tracing (KPT) [37] captures knowledge-state dynamics over time based on the forgetting curve and learning curve theories. A recent attempt at a forgetting-aware KT is the Deep Graph Memory Network (DGMN) [38] model, which uti-

lizes GNNs to learn forgetting behavior dynamically. DGMN differs from previous models in that a dynamic graph is built for identifying mutual relationships among concepts to model forgetting behaviors over the latent concept space. Though the above models attach great importance to the phenomenon of forgetting, they ignore the dynamic influence of knowledge decline and proficiency level change on human memory retention during learning, which limits the ability to capture nonmonotonic forgetting behaviors.

3. Preliminaries

3.1. Problem Definition

An online education platform encompasses a set of learner L, a wide range of knowledge components, including a set of questions $Q = \{q_1, q_2, \cdots, q_N\}$, and a set of concepts $C = \{c_1, c_2, \cdots, c_M\}$. In a KT task, the learning process is typically viewed as a composition of interactions between learners and knowledge components across consecutive time steps, which is explicitly reflected by learners' question-answering records. Along this line, knowledge tracing can be reasonably formulated as a sequence prediction problem. We denote a learner's learning sequence with t time steps as $X_t = (\{q_1, r_1\}, \{q_2, r_2\}, \cdots, \{q_t, r_t\})$. Here, $q_i \in Q$ refers to the question answered at time step i, and $r_i \in \{0, 1\}$ indicates whether the question q_i has been answered correctly, with 0 representing wrong and 1 representing correct. Important definitions are given as follows:

Definition 1 (Q-matrix). *A Q-matrix $Q \in \mathbb{R}^{N \times M}$ is a binary matrix that describes correlations between all the questions Q and concepts C, which is typically predefined by domain experts. If question q_i is related to the concept c_j, then $Q_{ij} = 1$; otherwise $Q_{ij} = 0$.*

Definition 2 (Concept Prerequisite Graph). *A concept prerequisite graph is represented as $G = (V, E, \mathbf{X})$, where $V = \{c_1, c_2, \cdots, c_M\}$ is the set of M distinct concept nodes. These concepts share prerequisite dependencies denoted as $E \in V \times V$; $\mathbf{X} \in \mathbb{R}^{M \times D}$ represents the node feature matrix, and D is the feature dimension. The topology of the graph is defined as the adjacency matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$, where $\mathbf{A}_{c_i, c_j} = 1$ means concept $c_i \in C$ is the prerequisite of concept $c_j \in C$, and $\mathbf{A}_{c_i, c_j} = 0$ otherwise.*

Definition 3 (Knowledge Tracing). *Given a learner's learning sequence $X_t = (x_1, x_2, ..., x_t)$ and the next question q_{t+1}, the objective of the KT task is to assess the learner's evolving knowledge state over time and predict the probability of q_{t+1} being answered correctly at the time step $t + 1$.*

Like the traditional skill-level KT method, this work denotes every question by its underlying concept through a question-to-concept mapping. A list of important notations used in DCKT is presented in Table 2.

Table 2. A list of important notations.

Variable	Description
L	a set of learners
Q	a set of questions
C	a set of concepts
G	a concept prerequisite graph
$\mathbf{A}$	the adjacency matrix of the graph G
$\mathbf{T}$	the transition probability matrix of the graph G
c_i	the concept i
q_t	the question at time t
x_t	the learning interaction at time t
$\mathbf{c}_i$	an embedding representation of concept c_i
$\mathbf{q}_t$	an embedding representation of question q_t
$\mathbf{x}_t$	an embedding representation of learning interaction x_t
r_t	learner's ground-truth response to the question q_t

Table 2. *Cont.*

Variable	Description
$\hat{r}_t$	the model-predicted learner's response to the question q_t
h_t	learner's hidden knowledge state at time t
$\mathbf{E}_K$	a embedding matrix of learner's personalized prior

3.2. Predefined Embeddings

To realize the main goal of knowledge tracing, we consider the following input elements: concepts, questions, answers, and interactions. In DCKT, all embeddings are associated with the concept embeddings, which are randomly initialized as $\mathbf{E}_C \in \mathbb{R}^{M \times D}$, where D represents the embedding dimension. After concept graph representation learning, the trained concept embeddings are mapped to the question embedding matrix $\mathbf{E}_Q \in \mathbb{R}^{t \times D}$. For easy calculation and unified representation, we convert the response r_i to a zero vector with the same D dimension as $\mathbf{r}_i \in \mathbb{R}^D$. The exactness of a learner's responses greatly affects the knowledge state assessment, so we distinguish between wrong and correct response representations. The learning interaction representation $\mathbf{x}_i \in \mathbb{R}^{2D}$ are defined as:

$$\mathbf{x}_i = \begin{cases} [\mathbf{q}_i \oplus \mathbf{r}_i], & \text{if } r_i = 0, \\ [\mathbf{r}_i \oplus \mathbf{q}_i], & \text{if } r_i = 1, \end{cases} \tag{2}$$

where $\oplus$ denotes the concatenation operation. We represent the embedding matrix of learning interactions (LI) as $LI \in \mathbb{R}^{t \times 2D}$.

4. The DCKT Model

This section introduces our proposed DCKT in detail, which consists of two modules: Unsupervised Graph Representation Learning (knowledge-centric module) and KT with Fine-grained Forgetting Behaviors Modeling (learner-centric module). Figure 2 shows the model architecture of DCKT.

4.1. Unsupervised Graph Representation Learning

This module aims to discover the latent knowledge graph structure and incorporate concept prerequisites as domain knowledge in preparation for the subsequent tasks.

4.1.1. Knowledge Structure Construction

In a KT task, knowledge components consist of learning sequences, e.g., concepts and questions, regularly organized following some inherent rules. For example, questions typically evolve from relatively elementary ones to advanced ones. Only when learners master the underlying prerequisite concepts do they have the knowledge base to master subsequent ones. Thus, fully exploring the prerequisite concept structure is crucial for modeling learners' knowledge. However, existing prerequisite inferences methods suffer from heavy expert-labeled reliance and the data sparsity problem. Inspired by unsupervised learning to dig out informative knowledge from the data themselves without relying on manual annotation, we aim to construct the underlying knowledge graph automatically via an unsupervised learning approach based on domain-related statistics.

Considering that the learning sequence order explicitly reflects a concept prerequisite, we learn the latent knowledge structure with the given Q-matrix and large-scale learning sequences in a data-driven manner. These question representations are constructed from the Q-matrix and concept map, enabling the integration of question-distinctive information and the dependency relationships between questions and concepts, but ignoring the inner correlations between latent concepts. Inspired by previous work [25], we first mine the implicit knowledge structure from the massive training datasets, thereby representing the concept relations as a transition probability matrix $\mathbf{T} \in \mathbb{R}^{M \times M}$. Here, $\mathbf{T}_{i,j} = \frac{n_{i,j}}{\sum_M n_{i,M}}$ if $i \neq j$; else, it equals 0; $n_{i,j}$ counts the total number of the unidirectional occurrences from concept

c_i to concept c_j. Then, we define the adjacency matrix of the concept prerequisite relations by $\mathbf{A}_{i,j} = 1$ if $\mathbf{T}_{i,j} \neq 0$; else it is 0.

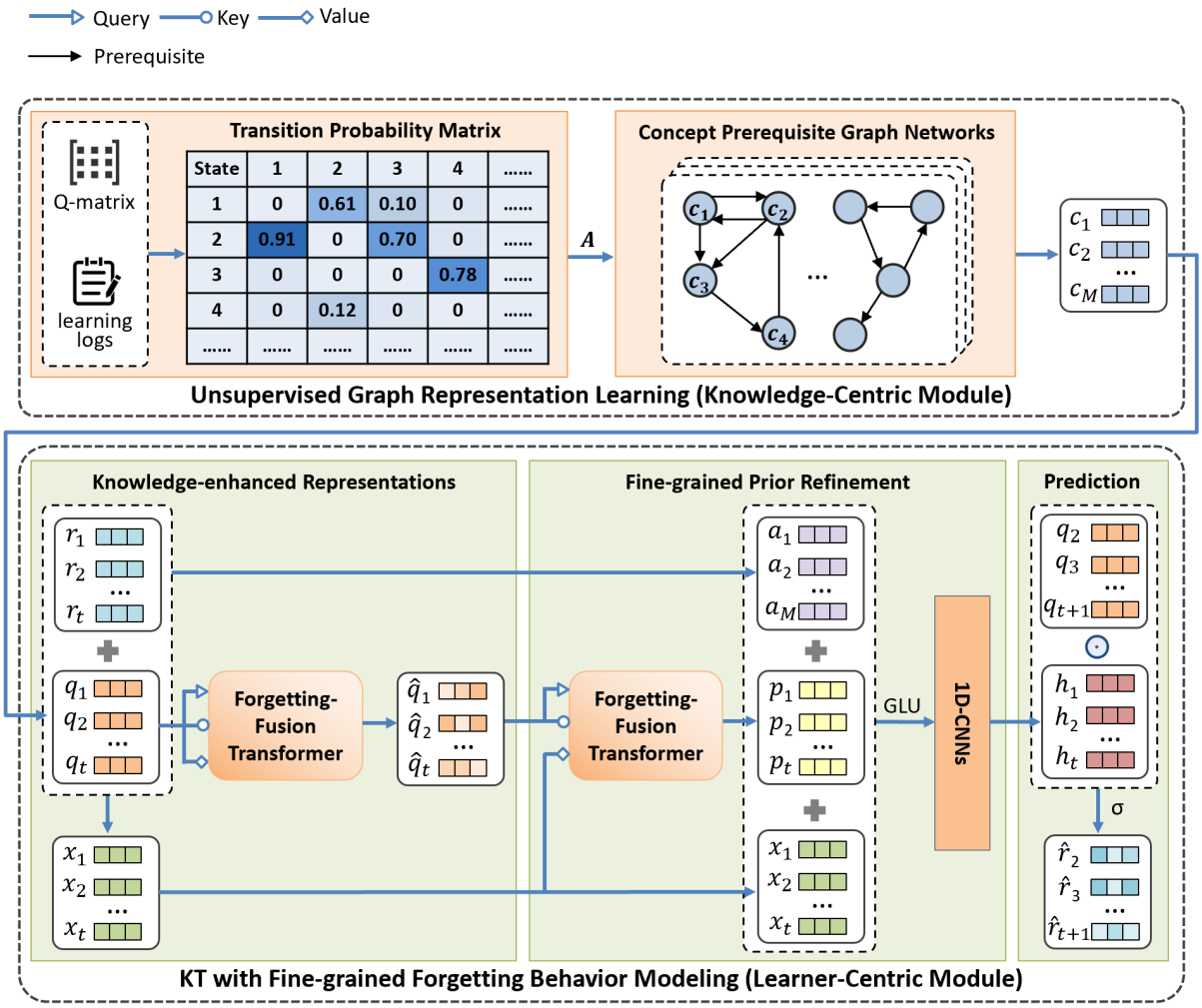


Figure 2. Overview of the DCKT model, where the top box represents unsupervised graph representation learning (knowledge-centric module), and the bottom box represents KT with fine-grained forgetting behaviors modeling (learner-centric module).

4.1.2. Concept Graph Representation

From the viewpoint of data structure, knowledge concepts have a potential graph-structured nature and are worthy of further exploration. After representing the concept prerequisite structure with the matrix $\mathbf{A}$, we construct the global prerequisite graph of all concepts as $G = (C, E, \mathbf{X})$, where the feature matrix $\mathbf{X}$ is randomly initialized by distinct concepts. To preserve the directions of prerequisite relations and extract high-order information in the graph, we leverage the graph neural network with edge multilayer perceptron (GNN-MLP) [39] to encode concept embeddings. It aggregates and propagates each message by applying an MLP to the concatenation of the source and target state, node representations are updated with the current concept node $\mathbf{c}_i$ and its neighboring node representations $\mathbf{N}_i$ using the following definition:

$$\mathbf{c}_i^{(t+1)} = \sigma\Big(\sum_{j \in \mathbf{N}_i \cup \{i\}} \frac{1}{|\mathbf{N}_i|} \cdot MLP_\ell(\mathbf{c}_i^{(t)} \| \mathbf{c}_j^{(t)})\Big) \tag{3}$$

where $MLP_\ell(\cdot)$ is the message passing function of the ℓ-th MLP layer, and $\|$ represents the concatenation operation. After graph learning, the concept embedding representations at time step t are updated as $\mathbf{c}_t$.

4.2. KT with Fine-Grained Forgetting Behaviors Modeling

To realize the primary goal of predicting learning performance and establishing personalized profiles of learners, we design a hierarchical framework to implement the downstream task of knowledge tracing.

4.2.1. Knowledge-Enhanced Representations

We update each question embedding with the obtained concept embedding matrix by its corresponding concept embedding. Thus, question embedding $\mathbf{q}_t$ at time step t is represented as its underlying concept embedding $\mathbf{c}_t$. Likewise, we update the response embedding $\mathbf{r}_t$ and interaction embedding $\mathbf{x}_t$ using the trained concept embedding $\mathbf{c}_t$. In this way, concept prerequisite information can be incorporated into model input, but deep-level contextual dependencies between question embeddings are still unexplored. Inspired by Transformer's excellent performance in parallelization and representation learning [22], we use a modified version called forgetting-fusion transformer for long-range relation learning, which is introduced in detail in the following section. We employ the forgetting-fusion transformer on the past question embeddings to further enhance global dependency learning between these prerequisite-enhanced embeddings. Specifically, the global-aware question representation $\hat{\mathbf{q}}_i \in \mathbb{R}^D$ is constructed by packing all the question embeddings $\{\mathbf{q}_1, \ldots, \mathbf{q}_i\}$ together into matrices Q, K, and V:

$$\hat{\mathbf{q}}_i = f_{ForgetAtt}(\mathbf{q}_1, \ldots, \mathbf{q}_i, \theta_1), \qquad i \in (1, t-1) \tag{4}$$

where $f_{ForgetAtt}(\cdot)$ is the attention function of our forgetting-fusion transformer, and θ_1 is a trainable global scalar initialized randomly and learned automatically during the training process.

4.2.2. Fine-Grained Prior Refinement

After extracting the complex global dependencies among questions, a second problem arises: what we have learned remains in general information, which may be deficient in distinguishing the knowledge mastery levels of learners. Moreover, the large receptive field of a transformer may result in fitting to some irrelevant features, but ignore highly discriminative features that could have a more significant impact on the prediction results. To deal with these potential issues, we augment the impact of learners' personalized characteristics by concatenating predefined educational prior from three aspects: Attempt Times (AT), Long-range Performance (LP), and Learning Interactions (LI), respectively. Despite its simplicity, our experiment results show its great potential for personalization and interpretability in the KT task.

Attempt Times (AT): A learner's proficiency level on the current question is strongly associated with their historical attempts related to concepts. Accordingly, we use AT to count the total number of times each learner answers a question relating to a specific concept, which is defined as follows:

$$AT = count(q^m) \tag{5}$$

where $m \in (1, M)$ refers to the concept m underlying the current question, and $count(q^m)$ represents the total number of times the learner answered question q^m.

Long-range Performance (LP): It is widely accepted that learning performance is roughly equivalent to historical interactions. In fact, the implicit connections between past questions and interactions have a non-negligible impact on learners' behavioral performance. On the one hand, learning performance is strongly associated with question similarity. For example, a learner tends to achieve similar performance on questions related

to the same concept. On the other hand, how learners interact with past questions greatly affects their performance to the current question, because historical interactions reflect the evolution of knowledge proficiency. Based on the two key factors of question similarity and learning interactions, we leverage the global-aware question embeddings $\{\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_i\}$ and interaction embeddings $\{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_i\}$ for a more fine-grained analysis of learning behaviors.

Although learning performance can be assessed using questions and interactions, we still face the inherent KT challenge of forgetting behaviors modeling. To meet the requirements of both dependency learning and forgetting modeling, we adopt a forgetting-fusion transformer with a unique implementation. Unlike traditional practice, where query, keys, and values correspond to the same item, we tune the forgetting-fusion transformer to better satisfy our needs by setting question embeddings as query and keys and interaction embeddings as values. The embedding representation of the learner's long-range performance at time step i is calculated by:

$$\mathbf{p}_i = f_{ForgetAtt}(\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_i, \hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_i, \theta_2) \tag{6}$$

where θ_2 is a global scalar specifically trained for the learning performance encoder. Thus, we obtain the embedding matrix of long-range performance $LP \in \mathbb{R}^{t \times 2D}$.

Then, we concatenate the embedding matrices of AT, LP, and LI. Here, we use GLU [40] to handle the concatenation to reduce gradient dispersion and nonlinear activation. The final outcome of a learner's personalized prior $\mathbf{E}_K \in \mathbb{R}^{t \times 2D}$ is denoted as:

$$\mathbf{E}_K = GLU(AT \oplus LP \oplus LI) \tag{7}$$

Although the forgetting-fusion transformer can extract global relationship dependencies of a long learning sequence, it does not perform well in capturing the more fine-grained dynamics of the knowledge-state evolution. To compensate for this defect, we employ a one-dimensional convolution neural network (1D-CNN) [24] on the learning sequence for a high-level learning behaviors analysis. The sliding window is the key element of the 1D-CNN for feature mapping, where learning interactions are segmented at a fixed length. The critical local features are refined from the continuous time series in a way that can learn discriminative features from the prior concatenation $\mathbf{E}_K$. Then, the output of the 1D-CNN is fed into GLU for a nonlinear transformation. To accelerate the training process, residual connections [41] are added from the input to the output between each convolutional block. Finally, we build the hierarchical convolutional neural networks by stacking the previously mentioned N identical convolution blocks. The convolutional operation of the ℓ-th CNN layer can be simply expressed as:

$$h_t^l = h_t^{l-1} + GLU(1D\text{-}CNN(h_t^{\ell-1})) \tag{8}$$

After local feature extraction by 1D-CNN, the knowledge state at time step t is updated as $\hat{h}_t$, which stands for the learner's knowledge proficiency level and can be further used to predict their future performance.

4.2.3. Prediction

The last module of DCKT predicts the learners' learning performance on the next question. When given the question q_{t+1} to solve, a learner searches for the relevant knowledge concept within an established cognitive horizon, which was modeled as the current knowledge state $\hat{h}_t$ after the t-th learning interaction. Therefore, we first apply the dot product of $\hat{h}_t$ and the next question embedding $\mathbf{q_{t+1}}$, then set up a sigmoid function to generate the future performance representation:

$$\hat{r}_{t+1} = \sigma\left(\hat{h}_t \cdot \mathbf{q}_{t+1}\right) \tag{9}$$

The output $\hat{r}_{t+1} \in [0, 1]$ represents the predicted probability of the learner correctly answering question q_{t+1}.

4.3. Forgetting-Fusion Transformer

Knowledge tracing is essentially a time series task whose datasets naturally arise from real-world educational applications and are recorded over a fixed sampling interval. Transformer [22] has a powerful sequential processing capability by virtue of its core component, positional encoding (P.E.), which can incorporate positional information in an input sequence and process the modified input in parallel. However, P.E. vectors record the location information of items in the input sequence, and embedding representations encode contextual information about the items, whereas simply adding the two cannot simulate the complex patterns of human forgetting behaviors, all of which place higher demands on the modeling of forgetting behaviors during the learning process. The key to solving these issues is a precise quantification of learners' forgetting effect in line with cognitive science studies, forgetting curve, etc.

As illustrated in Figure 3, we modify the original Transformer by fusing the forgetting behaviors with the scaled dot-product attention. Similar to the commonly used attention function in Transformer, we compute the dot products of the query with all keys, scale the dot products by $\frac{1}{\sqrt{d_k}}$, and finally obtain the weights of values via a softmax function. The biggest difference is that in place of the position encoding, we design a forgetting module to depict the overall forgetting effect of the learning process and adapt it to the attention weights. The output matrix is obtained from the following:

$$ForgetAttention(Q, K, V) = softmax(effect * \frac{QK^{\mathsf{T}}}{\sqrt{d_k}})V \tag{10}$$

where *effect* stands for the forgetting module output, d_k is the dimension of keys, and $*$ and T represent multiply and transpose operations, respectively.

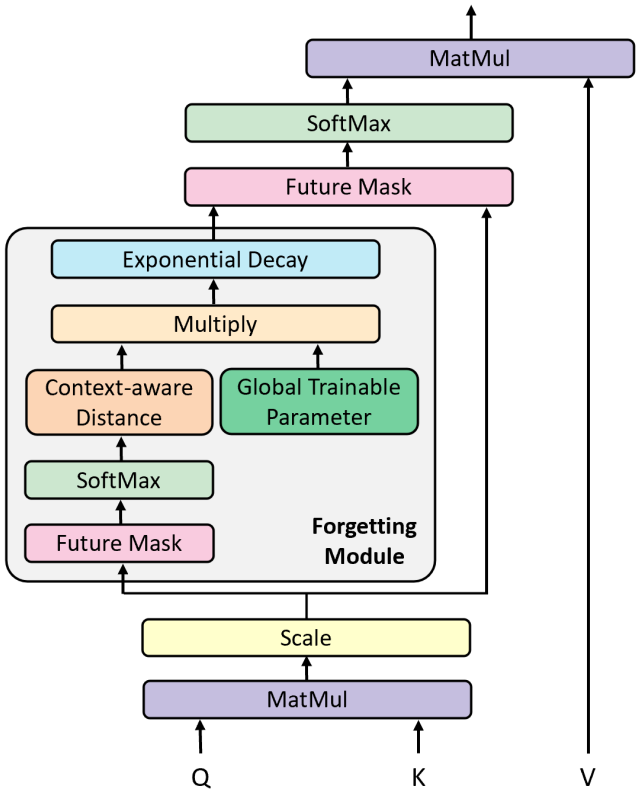


Figure 3. Illustration of the forgetting-fusion scaled dot product.

According to the relevant cognitive science study in [42], we design exponential decay attention to measure the forgetting effect. To weigh the importance of questions in combination with the inevitable forgetting, we consider two critical elements: context-

aware distance $d(t, \tau)$ [4] and question difficulty parameter θ. Specifically, each question's attention weight is calculated by its global importance to the entire learning sequence and the time intervals between past questions. The trainable global parameter θ, which indicates a global question difficulty, controls the exponential decay rate throughout the model training process. The calculation of the forgetting effect is simply expressed as:

$$effect = \exp(-\theta \cdot d(t, \tau)) \tag{11}$$

4.4. Objective Function

All the parameters are learned in the training process by minimizing the cross-entropy log loss between the predicted label $\hat{r}_t$ and the ground-truth response label r_t. We use the following objective function to optimize our model:

$$L = -\sum_{t=1}^{T}(r_t log\hat{r}_t + (1 - r_t)log(1 - \hat{r}_t)) \tag{12}$$

5. Experiments

In this section, to evaluate our proposed DCKT model; we present the experiment settings by answering the following research questions:

RQ1: Can our proposed DCKT model outperform other state-of-the-art KT models?

RQ2: How do different components in DCKT affect the final performance prediction?

RQ3: Does the pretext task for knowledge modeling in DCKT help to learn the meaningful representations of questions?

RQ4: How does DCKT precisely track the knowledge state compared with other KT models for personalized learner modeling?

5.1. Datasets

We use four real-world public datasets to evaluate the effectiveness of DCKT. Table 3 summarizes the general statistics for each dataset. Details of all datasets are as follows:

Table 3. Dataset Statistics.

Dataset	Students	Questions	Concepts	Records	Avg.len
ASSIST2009	4151	16,891	110	325,637	78
ASSIST2012	29,018	53,091	265	6,123,270	93
ASSIST2015	19,840	100	-	683,801	34
ASSISTChall	1709	3162	102	942,816	552

- ASSISTments2009 (https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data (accessed on 20 November 2022)) (ASSIST2009) is one of the most widely used benchmark datasets for KT tasks [43]. We conduct experiments using the latest updated skill-builder dataset, which removes duplicated records and facilitates data modeling. It contains a total of 325,637 records from 4151 learners associated with 16,891 distinct questions and 110 concepts.
- ASSISTments2012 (https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-withaffect (accessed on 20 November 2022)) (ASSIST2012) is the largest version of the ASSISTments datasets, which was collected from the ASSISTments online education platform during the 2012–2013 period. This dataset consists of 6,123,270 interactions, with 29,018 learners answering 53,091 questions.
- ASSISTments2015 (https://sites.google.com/site/assistmentsdata/home/2015-assistments-skill-builder-data (accessed on 20 November 2022)) (ASSIST2015) is composed of 708,631 response records over 100 distinct concepts produced by 19,917 students in 2015. The biggest difference between ASSIST2015 and previous versions of the ASSISTments datasets is that it provides no metadata or concept. Despite the increasing

number of records, ASSIST2015 has the lowest average number of records per learner at around 36.

- ASSISTment Challenge (https://sites.google.com/view/assistmentsdatamining/dataset (accessed on 20 November 2022)) (ASSISTChall) was publicly released from the 2017 ASSISTments data mining competition and has the most informative descriptions of all the ASSISTments datasets. In addition, it contains the most interactions, with 942,816 learning records, ranking first in terms of the number of records per learner ratio.

5.2. Baseline Methods

To answer research question 1, we compare the performance of DCKT against several well-known KT methods. To ensure the fairness of the comparison, we adopt the best parameter configurations for all methods. A summary of the baseline methods is as follows:

- DKT [6] introduces deep learning techniques into knowledge tracing for the first time. It utilizes an RNN or LSTM to model the knowledge state as a high-dimensional hidden state in the learning process.
- DKVMN [17] uses a memory network to enrich the hidden variable representation of DKT. Such a memory structure consists of two matrices: a static matrix called key to store all the concepts and a dynamic matrix called value to store and retrieve the mastery level of each concept through reading and writing operations.
- SAKT [23] is the first attentive knowledge tracing model based on the Transformer architecture. The attention mechanism is used for weighing the importance of past questions relative to the entire learning sequence, thereby predicting learning performance on the current question.
- CKT [24] utilizes a CNN to model learners' individualization for KT. It measures a learner's personalization in terms of the learner's personalized prior knowledge and learning rates during their learning process.
- AKT [4] uses a context-aware attention mechanism to learn the context-aware representations of exercises and answers. Unlike the scaled dot-product attention used in SAKT, AKT devises a modified monotonic attention version to simulate the forgetting effect by exponentially decaying attention weights.

5.3. Ablation Study of DCKT

To answer research question 2, we designed an ablation study with different variants of our proposed model to evaluate the impact of each component on the final prediction results. These variants are as follows:

- DCKT-NoPreq: This variant randomly initializes the concept embeddings to replace the knowledge-centric unsupervised representation learning module in DCKT, which learns concept representations by extracting the latent prerequisite relations. This variant aims to examine the effectiveness of concept representation learning combined with prerequisite discovery.
- DCKT-NoPrior: This variant removes all the components concerned with prior knowledge. We simply use the interactions to compute the learner's knowledge state. This variant evaluates the impact of the learner's personalized prior on the final results of DCKT.
- DCKT-NoTrans: This variant adopts the basic design of DCKT except for all operations by forgetting-fusion transformer, including question and long-range learning performance, which are replaced by the regular dot-product attention. This variant evaluates the impact of our forgetting-fusion transformer on the performance of DCKT.
- DCKT-NoForget: This variant is built by removing the forgetting module in the forgetting-fusion transformer. Compared with DCKT-NoTrans, this variant can further evaluate the impact of the forgetting module on the performance of the forgetting-fusion transformer.

5.4. Implementation Details

5.4.1. Dataset Preprocessing

We first preprocess the learning records at each time step for all datasets. For computational efficiency purposes, each dataset has a maximum input sequence length proportional to its average sequence length. If sequences are longer than the fixed length, we split them into several subsequences, while shorter ones are padded up to the fixed length.

5.4.2. Training Settings

To ensure the reliability of the experiment results, we perform standard 5-fold cross-validation over all the datasets. For each fold, we split 80% of learners into the training set and validation set, and the remaining 20% as the testing set. For empirical evaluation, we tune the hyper-parameters on the training set, choose the best-performing model on the validation set, and evaluate it on the testing set.

In our training settings, all learnable parameters are randomly initialized using the Xavier initialization [44] and optimized using the *Adam* gradient decent algorithm [45]. As for important hyper-parameter settings, a dropout rate with a keep probability of 0.2 is set to prevent overfitting, and the number of epochs is 80 for all datasets. For the ASSIST2009, ASSIST2012, ASSIST2015, and ASSISTChall datasets, the parameter batch sizes are set to 10, 20, 25, and 15, respectively. A series of experiments were conducted to determine the hyper-parameters of the forgetting-fusion transformer, including the number of attention heads $h = 8$ and the output dimension $d_{model} = 512$. Thus, the dimensions of queries, keys, and values are $d_q = d_k = d_v = d_{model}/h = 48$, and the inner-layer dimension of position-wise feed-forward networks $d_{ff} = 2048$. Our code is implemented with *TensorFlow 1.x* in Python on a Linux server with NVIDIA GeForce RTX 2080Ti GPUs.

6. Results and Discussion

In this section, we present the experiment results and discuss the important findings from our experiments.

6.1. Learning Performance Prediction (RQ1)

Learning performance prediction assesses a learner's future performance on specific questions, where the predicted binary-valued responses indicate whether the learner has mastered these questions. Thus, we considered it a binary classification task. To evaluate the performance predictions in the KT task, we use Area Under Curve (AUC) as the evaluation metric and compare DCKT with several state-of-the-art KT methods using the average AUC results across five test folds. Table 4 reports the AUC results of all methods over four public datasets, and Figure 4 visualizes the average AUC values with bar plots.

Table 4. The AUC results of all KT methods over four datasets.

Model	ASSIST2009	ASSIST2012	ASSIST2015	ASSISTChall
DKT [6]	0.8170	0.7286	0.7310	0.7213
DKVMN [17]	0.8093	0.7228	0.7276	0.7108
SAKT [23]	0.7520	0.7233	0.7212	0.6605
CKT [24]	0.8248	0.7310	0.7359	0.7263
AKT [4]	0.8169	0.7555	0.7828	0.7282
DCKT (ours)	**0.8250**	**0.7665**	**0.8530**	**0.7886**

The experiment results indicate that DCKT outperforms all other baselines over the four datasets. In comparison with the state-of-the-art methods, DCKT gains average AUC improvements of 1.1%, 7.1%, and 6.1% on the ASSIST2012, ASSIST2015, and ASSISTChall datasets, respectively. In addition, we also noticed some interesting findings. First, we can observe that DCKT achieves significant performance on ASSIST2015 and ASSISTChall, which reflects its strong ability to extract meaningful information in long sequences. It

also outperforms pure question-labeled KT datasets without considering question–concept relations. Second, DCKT achieves only slight improvements on the ASSIST2009 and ASSIST2012 datasets, which can be attributed to the complexity of the latent knowledge structure among datasets, as the two datasets contain a larger number of questions, which is a great challenge to knowledge tracing.

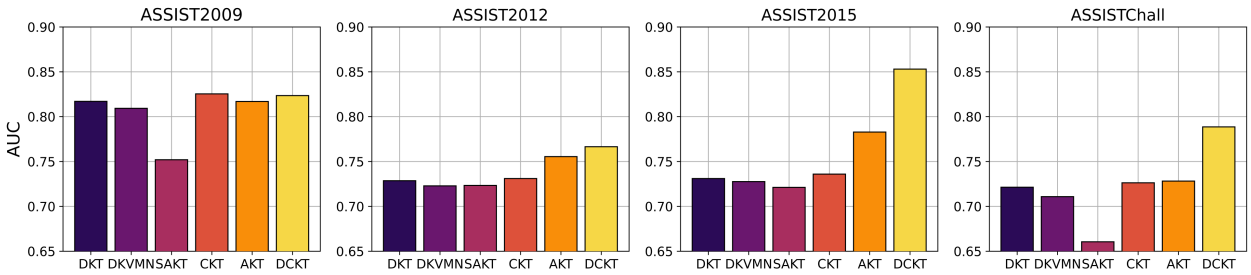


Figure 4. The average AUC values of all KT methods over four datasets.

6.2. Ablation Study (RQ2)

Table 5 summarizes the average AUC results for all variants of DCKT, each of which is an essential component of the complete model. From Table 5, we can draw some important conclusions. First, we can observe the impact of concept prerequisite inferences by comparing DCKT and DCKT-NoPreq, demonstrating DCKT's ability to model knowledge and learn valuable representations. Second, for the ASSIST2009, ASSIST2012, ASSIST2015, and ASSISTChall datasets, DCKT achieves a statistically significant performance upon DCKT-NoPrior by margins of 6.2%, 4.9%, 15.1%, and 11.1%, respectively. This phenomenon suggests that the personalized prior refinement module plays a crucial role in knowledge-state modeling. It can learn meaningful tokens from large-scale learning logs, reflecting the knowledge proficiency level unique to each learner. Third, the comparison of DCKT with DCKT-NoTrans further proves the forgetting-fusion transformer's superior performance, which benefits from its powerful global relation learning ability. Finally, the impact of the forgetting mechanism can be observed by comparing DCKT-NoTrans with DCKT-NoForget. The clear performance gap of these two models over all datasets, especially ASSIST2015 and ASSISTChall, demonstrates the necessity to incorporate forgetting behaviors in learner modeling.

Table 5. Ablation study of DCKT and its variants over four datasets.

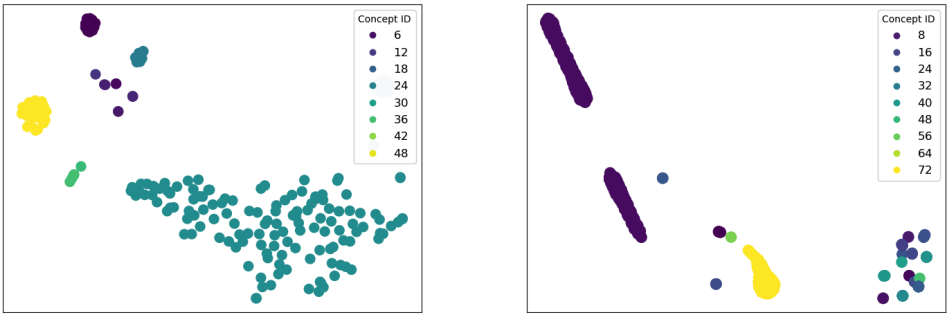
Model	Component				Dataset			
	Preq	Trans	Forget	Prior	ASSIST2009	ASSIST2012	ASSIST2015	ASSISTChall
DCKT	✓	✓	✓	✓	**0.8250**	**0.7665**	**0.8530**	**0.7886**
DCKT-NoPreq	×	✓	✓	✓	0.8154	0.7631	0.8335	0.7723
DCKT-NoPrior	✓	×	×	×	0.7608	0.7172	0.7024	0.6773
DCKT-NoTrans	✓	×	×	✓	0.8139	0.7321	0.7151	0.6960
DCKT-NoForget	✓	✓	×	✓	0.8160	0.7528	0.7877	0.7384

6.3. Question Clustering (RQ3)

In DCKT, the question embeddings are initialized with question identifiers. The feature weight matrix is obtained through unsupervised graph representation learning, so the learned embeddings are supposed to integrate concept prerequisites and question information. To assess the significance of the question embeddings learned by DCKT, we randomly select 200 questions in the ASSIST2009 and ASSISTChall datasets, respectively, and visualize the multidimensional embeddings of questions using T-SNE [46] in Figure 5.

Following the principle that questions underlying the same concept are labeled with the same color, we made some interesting observations. First, as shown in Figure 5,

the cluster results in the two datasets show that DCKT can learn question embedding representations well, where questions with the same concept are mostly distributed in the same cluster. Second, similar concepts with more relevant meanings are clustered at close range in the latent embedding space. For example, there are eight distinct concepts in the visualization results for ASSIST2009 in Figure 5a. The largest cluster with concept ID 24, which represents *"Addition and Subtraction Fractions"*, is close to the cluster with concept ID 36, which means *"Unit Rate"* operation. This phenomenon is consistent with the inner knowledge structure. However, the purple clusters with I.D.s {6,12} that correspond to the relatively unrelated concepts *"Stem and Leaf Plot"* and *"Circle Graph"* are the furthest away from all the other clusters. In summary, the clustering results intuitively describe the complex and implicit relationships between concepts and questions, which can provide important references for knowledge discovery.



(**a**) (**b**)

Figure 5. Clustering results of question embeddings learned by DCKT in two datasets: (**a**) ASSIST2009; (**b**) ASSISTChall, where the color of the question nodes refers to the underlying concept to which they belong.

6.4. Knowledge-State Visualization (RQ4)

To accomplish the goal of personalized learner modeling, we examine the effectiveness of DCKT in tracing knowledge state in terms of accuracy and plausibility. Figure 6 shows three visualization cases of the traced knowledge-state results from the same learning sequence, a fragment of a learner's interactions taken from the ASSISTChall dataset. From Figure 6, we draw some important findings that can help build a personalized profile of the learner.

The first case demonstrates that our proposed DCKT can achieve more accurate performance prediction results than the CKT model. As shown in Figure 6a; a heatmap visualizes the prediction probabilities of the learner answering questions correctly for the CKT and DCKT models. The horizontal axis refers to a learning sequence taken from the dataset ASSISTChall, where the learner has answered 23 questions on five concepts. Here, every question is denoted as a tuple consisting of its underlying concept and the correctness of the learner's answer. On the one hand, DCKT performs better in extracting personalized prior knowledge from the learner's practice history. We can observe that DCKT achieves significantly better predictions than CKT in the latter part of the learning sequence because a longer learning sequence implies a more abundant prior of the learner. On the other hand, DCKT also performs excellently in simulating the learner's forgetting behaviors during the learning process. For instance, the learner practiced questions corresponding to the same concept c_{50} in time steps 10 and 12–17. Still, all were answered wrongly, mainly due to the forgetting effect resulting from multiple intervals before reviewing concept c_{50}. In contrast, DCKT produces lower probabilities for these questions, as the forgetting-fusion transformer enables it to extract the forgetting features of human memory.

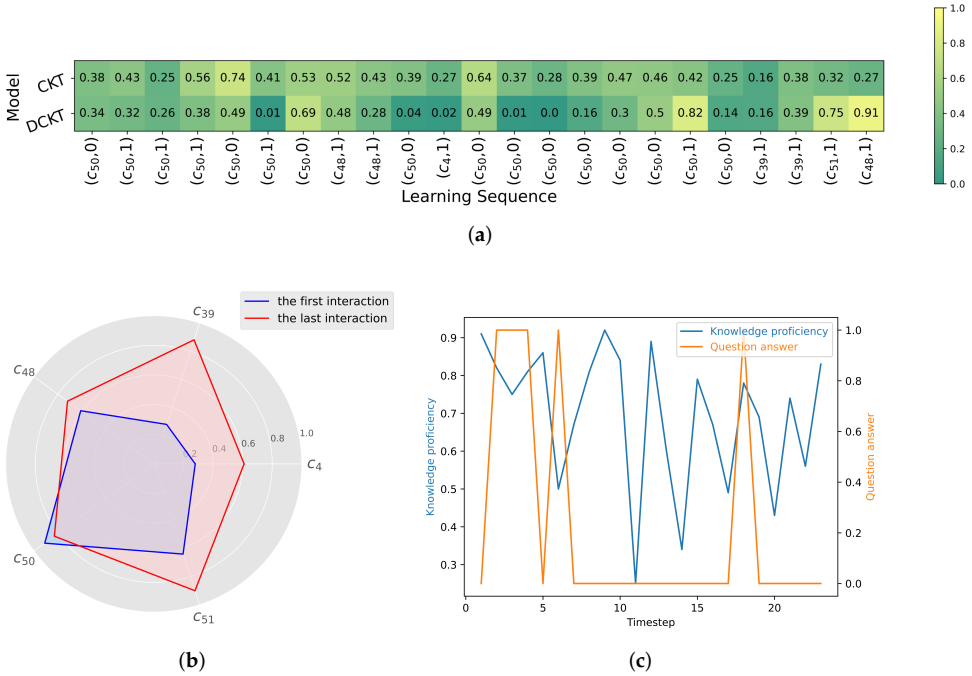


Figure 6. Visualization cases of a learner's knowledge state tracked by DCKT, using a learning sequence taken from the ASSISTChall dataset, where the learner answered 23 questions on five concepts. In (**a**), a heatmap compares the prediction probabilities of the learner answering questions correctly for the CKT and DCKT models. (**b**) is a radar chart that gives a before-and-after comparison of the learner's knowledge proficiency in the learning process. (**c**) depicts the mutual relationship between the learner's evolving proficiency on concept c_{50} and their answers.

For the second case, to obtained a reasonable explanation of the learner's knowledge state; a radar chart that describes the evolving process of the learner's knowledge proficiency is shown in Figure 6b. From the changing region between the first and last interactions in their learning, we notice an overall improvement in the knowledge proficiency levels for all concepts, except concept c_{50}. To determine the reasons for the learning regression, as shown in Figure 6c, we mine the mutual relationship between the learner's proficiency in concept c_{50} and their answers to related questions. We can see that when the learner correctly answered questions corresponding to concept c_{50}, their proficiency with idea c_{50} also increased. However, it is a nonmonotonic relationship affected by many potential factors, such as the practice of related concepts, review, forgetting behaviors, etc., which all affect the learner's knowledge state differently. While not all visualizations of the learners' knowledge state are precise in an intelligent education scenario, these findings can support personalized learner assessment and targeted instructional improvements.

7. Conclusions and Future Work

In this paper, we explore the coupling influence of knowledge concept prerequisite/relationships (i.e., knowledge-centric) and learners' forgetting behaviors (i.e., learner-centric) in promoting the performance of KT tasks. We thus proposed a novel KT model, named DCKT. Specifically, we leverage an unsupervised representation learning method to construct a prerequisite graph, and learn concept embeddings as a pretext task (knowledge-centric). Then, these learned embeddings are employed as input for the downstream task to perform knowledge tracing. As for the common forgetting behaviors, we designed a forgetting-fusion transformer to measure the forgetting effect during the learning process (learner-centric). Extensive experimental results over four public datasets prove that DCKT can outperform all other methods of learning performance prediction. Moreover, the visualization results show that DCKT can not only learn valuable embedding representations for knowledge components but also models an accurate and reasonable knowledge state

for learners. Our work points out a potential research avenue to advance the KT task by exploiting the complementary effects of knowledge and learner, but effective ways to combine the two need to be further explored.

For future work, we will explore more research opportunities for knowledge discovery and learner personalization modeling. For instance, we may use multimodal datasets or integrate educational contexts to enrich embedding representations for questions and concepts. Furthermore, we intend to pretrain question representations in a self-supervised learning manner that can automatically generate labels. Finally, for modeling knowledge states dynamically, we will investigate how to fully exploit dynamic information in the massive interaction records.

Author Contributions: Conceptualization, Y.C.; data curation, Y.C.; formal analysis, Y.C.; funding acquisition, Q.H.; investigation, S.W., F.J. and Y.T.; methodology, Y.C. and Q.H.; validation, Y.C. and Y.T.; visualization, Y.C. and S.W.; writing—original draft, Y.C. and Q.H.; writing—review and editing, Y.C., S.W., F.J., Y.T. and Q.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Research and Development Program of Zhejiang Province (No. 2021C03141, 2022C03106) and the Open Research Fund of College of Teacher Education, Zhejiang Normal University (No. jykf22006).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki. Ethical review and approval were waived because permission to conduct this government-funded project in China was granted.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nguyen, T. The effectiveness of online learning: Beyond no significant difference and future horizons. *MERLOT J. Online Learn. Teach.* **2015**, *11*, 309–319.
2. Corbett, A.T.; Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.* **2005**, *4*, 253–278. [CrossRef]
3. Pelánek, R. Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques. *User Model. User-Adapt. Interact.* **2017**, *27*, 313–350. [CrossRef]
4. 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, Virtual Event, 6–10 July 2020; pp. 2330–2339.
5. Liu, Q.; Shen, S.; Huang, Z.; Chen, E.; Zheng, Y. A survey of knowledge tracing. *arXiv* **2021**, arXiv:2105.15106.
6. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.J.; Sohl-Dickstein, J. Deep knowledge tracing. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–12.
7. Nagatani, K.; Zhang, Q.; Sato, M.; Chen, Y.Y.; Chen, F.; Ohkuma, T. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In Proceedings of the The World Wide Web Conference, San Francisco, CA, USA, 13 May 2019; pp. 3101–3107.
8. Chen, P.; Lu, Y.; Zheng, V.W.; Pian, Y. Prerequisite-Driven Deep Knowledge Tracing. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 39–48.
9. Tong, S.; Liu, Q.; Huang, W.; Hunag, Z.; Chen, E.; Liu, C.; Ma, H.; Wang, S. Structure-Based Knowledge Tracing: An Influence Propagation View. In Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), Sorrento, Italy, 17–20 November 2020; pp. 541–550. [CrossRef]
10. Gan, W.; Sun, Y. Prerequisite-driven Q-matrix Refinement for Learner Knowledge Assessment: A Case Study in Online Learning Context. *arXiv* **2022**, arXiv:2208.12642.
11. Mandin, S.; Guin, N. Basing learner modelling on an ontology of knowledge and skills. In Proceedings of the 2014 IEEE 14th International Conference on Advanced Learning Technologies, Athens, Greece, 7–10 July 2014; pp. 321–323.
12. Yang, Y.; Shen, J.; Qu, Y.; Liu, Y.; Wang, K.; Zhu, Y.; Zhang, W.; Yu, Y. GIKT: A Graph-Based Interaction Model for Knowledge Tracing. In *Machine Learning and Knowledge Discovery in Databases*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 299–315.
13. Corbett, A. Cognitive Mastery Learning in the Act Programming Tutor. Adaptive User Interfaces. AAAI SS-00-01. 2000. Available online: https://aaai.org/Library/Symposia/Spring/ss00-01 (accessed on 20 November 2022).
14. Cen, H.; Koedinger, K.; Junker, B. Comparing Two IRT Models for Conjunctive Skills. In *Intelligent Tutoring Systems*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 796–798.

15. Pavlik, P.I.; Cen, H.; Koedinger, K.R. Performance Factors Analysis—A New Alternative to Knowledge Tracing. In Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling, Brighton, UK, 6–10 July 2009; IOS Press: Amsterdam, The Netherlands, 2009; pp. 531–538.
16. Vie, J.J.; Kashima, H. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 750–757. [CrossRef]
17. 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, Perth, Australia, 3–7 April 2017; pp. 765–774.
18. Minn, S.; Yu, Y.; Desmarais, M.C.; Zhu, F.; Vie, J.J. Deep Knowledge Tracing and Dynamic Student Classification for Knowledge Tracing. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 1182–1187.
19. 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, New Orleans, LA, USA, 2–7 February 2018; Volume 32. [CrossRef]
20. Liu, Q.; Huang, Z.; Yin, Y.; Chen, E.; Xiong, H.; Su, Y.; Hu, G. EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 100–115. [CrossRef]
21. Shen, J.T.; Yamashita, M.; Prihar, E.; Heffernan, N.; Wu, X.; Graff, B.; Lee, D. Mathbert: A pre-trained language model for general nlp tasks in mathematics education. *arXiv* **2021**, arXiv:2106.07340.
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All you Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
23. Pandey, S.; Karypis, G. A Self-Attentive model for Knowledge Tracing. *arXiv* **2019**, arXiv:1907.06837.
24. Shen, S.; Liu, Q.; Chen, E.; Wu, H.; Huang, Z.; Zhao, W.; Su, Y.; Ma, H.; Wang, S. Convolutional Knowledge Tracing: Modeling Individualization in Student Learning Process. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, 25–30 July 2020; pp. 1857–1860.
25. Nakagawa, H.; Iwasawa, Y.; Matsuo, Y. Graph-based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Thessaloniki, Greece, 14–17 October 2019; pp. 156–163. [CrossRef]
26. Wu, Z.; Huang, L.; Huang, Q.; Huang, C.; Tang, Y. SGKT: Session graph-based knowledge tracing for student performance prediction. *Expert Syst. Appl.* **2022**, *206*, 117681. [CrossRef]
27. Song, X.; Li, J.; Lei, Q.; Zhao, W.; Chen, Y.; Mian, A. Bi-CLKT: Bi-graph contrastive learning based knowledge tracing. *Knowl.-Based Syst.* **2022**, *241*, 108274. [CrossRef]
28. Laurence, S.; Margolis, E. Concepts and Cognitive Science. In *Concepts: Core Readings*; Margolis, E., Laurence, S., Eds.; MIT Press: Cambridge, MA, USA, 1999; pp. 3–81.
29. Wang, S.; Ororbia, A.; Wu, Z.; Williams, K.; Liang, C.; Pursel, B.; Giles, C.L. Using prerequisites to extract concept maps fromtextbooks. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, Indianapolis, IN, USA, 4–28 October 2016; pp. 317–326.
30. Pan, L.; Li, C.; Li, J.; Tang, J. Prerequisite relation learning for concepts in moocs. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 1447–1456.
31. Roy, S.; Madhyastha, M.; Lawrence, S.; Rajan, V. Inferring concept prerequisite relations from online educational resources. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 9589–9594.
32. Chen, Y.; González-Brenes, J.P.; Tian, J. Joint Discovery of Skill Prerequisite Graphs and Student Models. In Proceedings of the International Conference on Educational Data Mining (EDM), Raleigh, NC, USA, 29 June–2 July 2016.
33. Pavlik, P.I.; Anderson, J.R. Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect. *Cogn. Sci.* **2005**, *29*, 559–586. [CrossRef] [PubMed]
34. Averell, L.; Heathcote, A. The form of the forgetting curve and the fate of memories. *J. Math. Psychol.* **2011**, *55*, 25–35. [CrossRef]
35. Ebbinghaus, H. Memory: A Contribution to Experimental Psychology. *Ann. Neurosci.* **2013**, *20*, 155–156. [CrossRef] [PubMed]
36. Nedungadi, P.; Remya, M. Incorporating forgetting in the personalized, clustered, bayesian knowledge tracing (pc-bkt) model. In Proceedings of the 2015 International Conference on Cognitive Computing and Information Processing (CCIP), Noida, India, 3–4 March 2015; pp. 1–5.
37. Huang, Z.; Liu, Q.; Chen, Y.; Wu, L.; Xiao, K.; Chen, E.; Ma, H.; Hu, G. Learning or forgetting? A dynamic approach for tracking the knowledge proficiency of students. *Acm Trans. Inf. Syst. (TOIS)* **2020**, *38*, 1–33. [CrossRef]
38. Abdelrahman, G.; Wang, Q. Deep Graph Memory Networks for Forgetting-Robust Knowledge Tracing. *IEEE Trans. Knowl. Data Eng.* **2022**, *early access*. [CrossRef]
39. Brockschmidt, M. GNN-FiLM: Graph Neural Networks with Feature-wise Linear Modulation. In Proceedings of the International Conference on Machine Learning, Virtual, 13–18 July 2020; Volume 119, pp. 1144–1152.
40. Dauphin, Y.N.; Fan, A.; Auli, M.; Grangier, D. Language modeling with gated convolutional networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 933–941.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

42. Reddy, S.; Labutov, I.; Banerjee, S.; Joachims, T. Unbounded Human Learning: Optimal Scheduling for Spaced Repetition. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016.
43. Feng, M.; Heffernan, N.; Koedinger, K. Addressing the assessment challenge with an online system that tutors as it assesses. *User Model. User-Adapt. Interact.* **2009**, *19*, 243–266. [CrossRef]
44. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
45. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
46. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.