DGCPL: Dual Graph Distillation for Concept Prerequisite Relation Learning

Miao Zhang[1,2], **Jiawei Wang**[1,3], **Jinying Han**[1,3], **Kui Xiao**[1,3*],
Zhifei Li[1,2], **Yan Zhang**[1,3], **Hao Chen**[1,3], **Shihui Wang**[1,3]
[1]School of Computer Science, Hubei University, China
[2]Hubei Key Laboratory of Big Data Intelligent Analysis and Application, China
[3]Key Laboratory of Intelligent Sensing System and Security, Ministry of Education, China
zhangmiao@hubu.edu.cn, {wangjw, hanjy}@stu.hubu.edu.cn
{xiaokui, zhifei1993, zhangyan, ch, wsh}@hubu.edu.cn

Abstract

Concept prerequisite relations determine the learning order of knowledge concepts in one domain, which has an important impact on teachers' course design and students' personalized learning. Current research usually predicts concept prerequisite relations from the perspective of knowledge, and rarely pays attention to the role of learners' learning behavior. We propose a **D**ual **G**raph Distillation Method for **C**oncept **P**rerequisite Relation **L**earning (DGCPL). Specifically, DGCPL constructs a dual graph structure from both the knowledge and learning behavior perspectives, and captures the high-order knowledge features and learning behavior features through the concept-resource hypergraph and the learning behavior graph respectively. In addition, we introduce a gated knowledge distillation to fuse the structural information of concept nodes in the two graphs, so as to obtain a more comprehensive concept embedding representation and achieve accurate prediction of prerequisite relations. On three public benchmark datasets, we compare DGCPL with eight graph-based baseline methods and five traditional classification baseline methods. The experimental results show that DGCPL achieves state-of-the-art performance in learning concept prerequisite relations. Our code is available at https://github.com/wisejw/DGCPL.

1 Introduction

As the increasing of learners number on online learning platforms, the learning resources also become more abundant [O'Dea and Stern, 2022]. However, the diversity of resources often leads to difficulties in choice for learners. The main reason for this is that the dependency between the core concepts in different learning resources is unclear. Concept prerequisite relations are the type of knowledge dependency, which indicates that the learning order of core concepts in a domain determines the learning order of knowledge. For example, in mathematics, understanding the concept of "*Markov chain*" requires prior knowledge of "*Probability*" and "*Stochastic process*". Thus, Concept prerequisite relations play an indispensable role in various educational scenarios, such as learning path planning [Yu *et al.*, 2024; Wang and Liu, 2016], knowledge tracing [Huang *et al.*, 2024; Yin *et al.*, 2023], cognitive diagnosis [Song *et al.*, 2023; Gao *et al.*, 2023], and learning resource recommendation [Dong *et al.*, 2024; Wang *et al.*, 2022a].

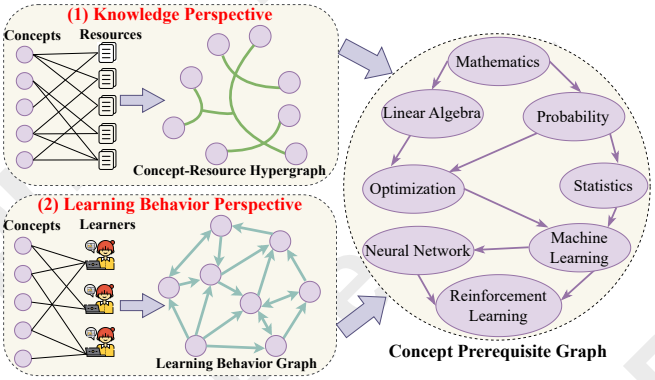

Figure 1: Learning concept prerequisite relations from the perspective of knowledge and the perspective of learning behavior.

Early research with machine learning for concept prerequisite relations [Talukdar and Cohen, 2012; Liang *et al.*, 2015] primarily rely on manual feature extraction of concept pairs for concept modeling. This involves designing a set of metrics to capture the semantic and structural information between concepts, but the process is both tedious and time-consuming. With the rise of graph neural networks, researchers begin to explore the intrinsic relations between knowledge concepts in graphs [Sun *et al.*, 2022; Zhang *et al.*, 2022a]. However, previous research mostly infer prerequisite relations between knowledge concepts from the knowledge perspective [Zhang *et al.*, 2025]. Few research approach the problem from the perspective of learning behavior. In fact, if many learners, after studying concept B, immediately look up the definition of concept A, there is likely a prerequisite relation between these two concepts. Existing research on learning behaviors [Sayyadiharikandeh *et al.*, 2019; Hu *et al.*, 2021] predominantly adopts traditional machine learning methods. While these studies recognize the importance of learning behaviors, they have yet to explore advanced

*Corresponding author.

deep learning methods for deeper insights.

However, current graph-based methods usually only focus on pairwise relations of concept-concept or concept-resource. They do not address the set-level relations between a learning resource and multiple knowledge concepts, which may lead to the omission of key information. On the other hand, few research [Sayyadiharikandeh *et al.*, 2019; Hu *et al.*, 2021] use learning behavior features to predict prerequisite relations, but these still rely on traditional machine learning methods. How to combine advanced deep learning methods and predict prerequisite relations from both the knowledge and learning behavior perspectives is an urgent research question, as shown in Figure 1.

To this end, we propose a **D**ual **G**raph Distillation method for **C**oncept **P**rerequisite Relation **L**earning (DGCPL), which aims to construct a dual graph model from both the knowledge and learning behavior perspectives to capture concept prerequisite relations. Specifically, the Concept-Resource Hypergraph represents the higher-order knowledge relations between concepts and resources. The Learning Behavior Graph describes the learner's behavior on knowledge concepts. To effectively integrate information from different perspectives, we design gated knowledge distillation to combine the knowledge structure features of concepts and dynamic behavior features of learners, and adaptively adjust and optimize two types of information to identify prerequisite relations.

The main contributions of this paper are as follows:

- We propose a Dual Graph Distillation method for concept prerequisite relation learning, which predicts concept prerequisite relations by constructing a dual graph from the knowledge and learning behavior perspectives.
- We introduce gated knowledge distillation that adaptively integrates higher-order knowledge relations and learning behavior features information in the dual graph through a gating mechanism, resulting in more comprehensive representations of concept embeddings.
- On three publicly available benchmark datasets, we compare DGCPL with eight graph-based baseline methods and five traditional classification baseline methods. The experimental results demonstrate that DGCPL achieves state-of-the-art performance.

2 Related Work

2.1 Concept Prerequisite Relation Learning

Talukdar and William pioneer the task of extracting concept prerequisite relations from Wikipedia [Talukdar and Cohen, 2012], laying the foundation for subsequent research. Early machine learning methods, such as RefD [Liang *et al.*, 2015], propose a metric based on citation distance; Sayyadiharikandeh et al. are the first to analyze learning behavior using Wikipedia Clickstream data [Sayyadiharikandeh *et al.*, 2019]; and PREREQ [Roy *et al.*, 2019] employs siamese networks to identify prerequisite relations between concepts. These research further advance the domain. In the context of graph neural networks, research gradually shifts to graph modeling. ConLearn [Sun *et al.*, 2022] combines concept graphs with self-attention mechanisms to learn prerequisite relations in a context-aware manner. MHAVGAE [Zhang *et al.*, 2022a], HGAPNet [Mazumder *et al.*, 2023], and LCPRE [Sun *et al.*, 2024] extract prerequisite relations by constructing heterogeneous concept-resource graphs, offering new insights for modeling complex concept relations.

2.2 Knowledge Distillation

Knowledge Distillation [Hinton *et al.*, 2015] is a model compression technique, which aims to transfer the knowledge of a complex teacher model to a lightweight student model. To integrate diverse data more effectively, the DGEKT model [Cui *et al.*, 2024] in the knowledge tracing research field incorporates a gating mechanism into online knowledge distillation to dynamically adjust the knowledge transferred from the teacher model to the student model. Traditional knowledge distillation achieves knowledge transfer by minimizing the difference between the outputs of the teacher and student models. However, many subsequent studies [Romero *et al.*, 2015; Zagoruyko and Komodakis, 2017; Pechác *et al.*, 2024] have found that extracting representation features from the intermediate layers of the teacher model can further improve the effectiveness of distillation. Additionally, the introduction of online knowledge distillation [Yang *et al.*, 2023; Gong *et al.*, 2023] and self-distillation [Zhang *et al.*, 2022b; Li *et al.*, 2024] has opened new directions for the development of knowledge distillation.

Remarks. We construct a dual graph structure from the perspectives of knowledge and learning behavior, using hypergraph and directed graph to model the concept-resource hypergraph and the learning behavior graph, respectively. Then we use gated knowledge distillation to integrate high-order knowledge relations between concepts and resources with learning behavior features.

3 Problem Statement

In this section, we present the key terms and research definition used in this paper. Let $\mathcal{C} = \{c_1, c_2, c_3, ..., c_m\}$ represent the set of knowledge concepts, and $\mathcal{R} = \{r_1, r_2, r_3, ..., r_n\}$ represent the set of learning resources, where m and n denote the total number of knowledge concepts and learning resources, respectively. A knowledge concept refers to a core knowledge unit in a specific domain. And a learning resource refers to an independent learning object, typically a chapter in a textbook or the subtitles of a course video.

Concept-Resource Hypergraph. In the hypergraph $G_H = (V, E_H)$, nodes represent concepts, and hyperedges represent resources. Here, $V = \{v_1, v_2, ..., v_m\}$ denotes the set of nodes, and $E_H = \{h_1, h_2, ..., h_n\}$ denotes the set of hyperedges. The relation between hyperedges and nodes represents the containment relation between resources and concepts.

Learning Behavior Graph. In the directed graph $G_L = (V, E_L)$, nodes represent concepts, and edges represent the navigation behavior of learners between concepts. E_L denotes the set of edges in the graph, where $l_{ij} \in E_L$ represents the number of times learners navigate from the definition document of concept c_i to the definition document of concept c_j within a certain time range.

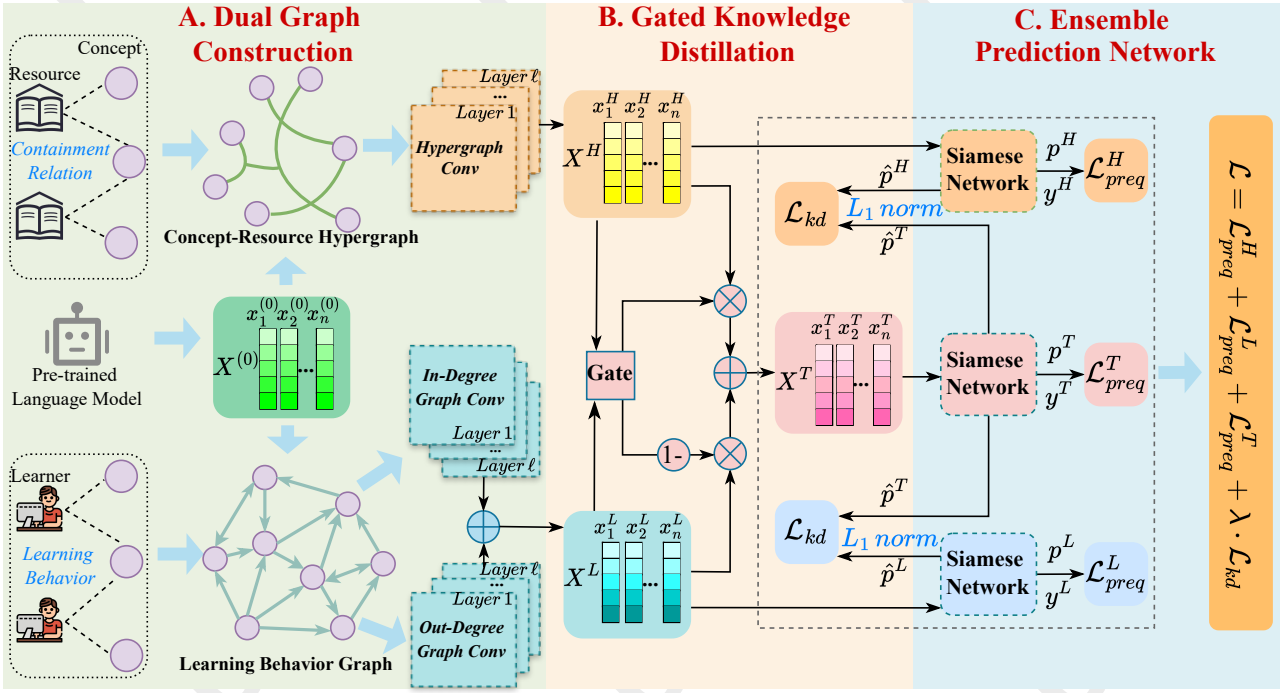

Figure 2: The overall structure of the proposed DGCPL.

Here, the concept definition documents are not the same as the learning resources mentioned above. It is often challenging to directly obtain clickstream data from learning resources. Therefore, We uses the Wikipedia page articles corresponding to concepts as their definition documents to collect learning behavior data and construct the learning behavior graph. Clickstream data only includes concept pairs with more than 10 navigation instances in Wikipedia. Clearly, this clickstream data represents learning learning behavior with distinct group characteristics, which better indicates the potential relation between two concepts. Such dynamic learning behaviors can reflect the concept dependency relations.

Research Definition. Given a set of knowledge concepts $\mathcal{C}$ and a set of learning resources $\mathcal{R}$ within a specific domain, our goal is to learn a function $F_\theta : \mathcal{C} \times \mathcal{C} \rightarrow \{0, 1\}$ that can predict whether any pair of concepts $< c_i, c_j >$ in the domain has a prerequisite relation.

4 Methodology

In this paper, we introduce an innovative model designed to learn concept prerequisite relations using a dual graph distillation strategy. The structure of DGCPL consists of three main modules: dual graph construction, gated knowledge distillation, and ensemble prediction network. Figure 2 illustrates the overall architecture of our proposed DGCPL.

4.1 Dual Graph Construction

Concept-Resource Hypergraph

We adopt a hypergraph to capture the high-order knowledge relations between knowledge concepts and learning resources, thereby learning concept embeddings within the hypergraph. Specifically, we first construct hypergraph $G_H = (V, E_H)$. The hypergraph adjacency matrix is denoted as $\mathrm{H} \in \mathbb{R}^{\mathrm{n}\times\mathrm{m}}$. If a node v_i belongs to a hyperedge h_i, $H_{i,h_i} = 1$; otherwise, $H_{i,h_i} = 0$. Additionally, each hyperedge is assigned a weight w_h, and all these weights are stored in a diagonal matrix $\mathbf{W} \in \mathbb{R}^{m\times m}$. For simplicity, we set the weights of all hyperedges to 1. We update the embeddings of concept nodes using a Hypergraph Convolutional Network (HGCN) [Feng *et al.*, 2019],

$$\begin{aligned}\mathbf{x}_i^{(l+1)} =& \phi(\mathbf{D}^{-1/2}\mathbf{H}\mathbf{W}\mathbf{B}^{-1}\mathbf{H}^T\mathbf{D}^{-1/2}\mathbf{x}_i^{(l)}\Theta^{(l+1)} + \\ &+\mathrm{Linear}(\mathbf{x}_i^{(l)})). \end{aligned} \tag{1}$$

Here, $\mathbf{x}_i^{(l)} \in \mathbb{R}^{C_l}$ represents the embedding of the i-th node at the l-th layer. $\mathbf{D} \in \mathbb{R}^{n\times n}$ is the degree matrix of nodes, where the degree of node v_i is $D_{i,i} = \sum_{i=1}^{m} w_{h_i} H_{i,h_i}$. $\mathbf{B} \in \mathbb{R}^{m\times m}$ is the degree matrix of hyperedges, where the degree of hyperedge h_i is $B_{h_i,h_i} = \sum_{i=1}^{n} H_{i,h_i}$. $\Theta^{(l+1)} \in \mathbb{R}^{C_l \times C_{l+1}}$ is the learnable weight matrix at the $(l+1)$-th layer. ϕ denotes the ReLU activation function, and $(\cdot)^T$ denotes the transpose operation. To retain the original critical information, we add $\mathbf{x}^{(l)}$ as a residual and apply a linear transformation $linear(\cdot)$. The initial embedding $\mathbf{x}_i^{(0)} \in \mathbb{R}^{C_0}$ of each concept is generated by inputting the concept definition document into a pre-trained language model.

Finally, we obtain the concept embedding $\mathbf{x}_i^H \in \mathbb{R}^d$ from the concept-resource hypergraph, where d represents the dimension of the concept embeddings.

Learning Behavior Graph

In addition to the high-order knowledge relations between learning resources and knowledge concepts, learners' behav-

iors also have a significant impact on predicting concept prerequisite relations. This has been demonstrated in previous studies [Sayyadiharikandeh *et al.*, 2019; Hu *et al.*, 2021]. Therefore, we construct a learning behavior graph based on learning behaviors to predict concept prerequisite relations. Specifically, for a node v_i in the learning behavior graph $G_L = (V, E_L)$, we let $n_{j,i}$ represent the number of times learners navigate from node v_j to node v_i. The in-degree of node v_i with respect to node v_j is defined as:

$$A_{j,i}^{(in)} = \begin{cases} \frac{n_{j,i}}{\sum_{v_k \in V} n_{k,i}}, & \text{if } (v_j, v_i) \in E_L \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where V represents the set of all knowledge concepts in the domain. Similarly, let $n_{i,j}$ represent the number of times learners navigate from node v_i to node v_j. The out-degree of node v_i with respect to node v_j is defined as:

$$A_{i,j}^{(out)} = \begin{cases} \frac{n_{i,j}}{\sum_{v_k \in V} n_{i,k}}, & \text{if } (v_i, v_j) \in E_L \\ 0. & \text{otherwise} \end{cases} \quad (3)$$

In addition, we set $A_{i,i}^{(in)} = 1$ and $A_{i,i}^{(out)} = 1$. Then, we define the in-degree and out-degree of node v_i as follows:

$$deg_i^{(in)} = \sum_{v_k \in V} A_{k,i}^{(in)}, \ deg_i^{(out)} = \sum_{v_k \in V} A_{i,k}^{(out)}. \quad (4)$$

In the learning behavior graph, each edge has a direction, which makes information propagation more complex. Inspired by previous research [Cui *et al.*, 2024], we utilize a Directed Graph Convolutional Network (DGCN) [Tong *et al.*, 2020; Zhang *et al.*, 2021] to learn concept embeddings from the learning behavior graph. Nodes can not only propagate information to other nodes but also receive information from other nodes. This parallel bidirectional information propagation mechanism enables more accurate capture of concept dependency relations based on learners' dynamic behaviors.

For the in-degree directed graph (denoted as "$-$") of the learning behavior graph, we use DGCN to propagate information along the in-degree edges. The embedding of node at the $(l+1)$-th layer is calculated as follows:

$$\mathbf{x}_i^{(l+1),-} = \phi\Big(\sum_{v_j \in N(i)^-} \frac{A_{j,i}^{(in)}}{\sqrt{deg_i^{(in)} \cdot deg_j^{(out)}}} \Theta_N^{(l+1),-} \mathbf{x}_j^{(l)}\Big), \quad (5)$$

where $N(i)^-$ represents the in-degree neighbor set of node v_i (including node v_i itself). $\Theta_N^{(l+1),-}$ is the learnable weight matrix at the $(l+1)$-th layer in the in-degree directed graph. $\mathbf{x}_j^{(l)}$ represents the embedding of node j at the l-th layer. Similarly, the initial embedding of each concept, $\mathbf{x}_i^{(0)} \in \mathbb{R}^{C_0}$ is generated by inputting the concept definition document into a pretrained language model.

Similarly, to achieve information propagation along the out-degree edges, we also apply DGCN to the out-degree directed graph (denoted as "$+$"). The embedding of node v_i at the $(l+1)$-th layer is calculated as follows:

$$\mathbf{x}_i^{(l+1),+} = \phi\Big(\sum_{v_j \in N(i)^+} \frac{A_{i,j}^{(out)}}{\sqrt{deg_i^{(out)} \cdot deg_j^{(in)}}} \Theta_N^{(l+1),+} \mathbf{x}_j^{(l)}\Big), \quad (6)$$

where $N(i)^+$ represents the out-degree neighbor set of node v_i (including node v_i itself), and $\Theta_N^{(l+1),+}$ is the learnable weight matrix at the $(l+1)$-th layer in the out-degree directed graph. Then, for the entire learning behavior graph, we fuse the updated concept embeddings obtained from the in-degree directed graph and the out-degree directed graph. The embedding of node v_i at the $(l+1)$-th layer is defined as:

$$\mathbf{x}_i^{(l+1)} = \phi(\text{Linear}(\mathbf{x}_i^{(l+1),+}) + \text{Linear}(\mathbf{x}_i^{(l+1),-})). \quad (7)$$

Here, we apply a linear transformation to both embeddings separately and then add them element-wise. Additionally, there are various options for embedding fusion methods. Besides element-wise addition, common methods include embedding concatenation, element-wise maximum, and element-wise average. We provide a detailed experimental analysis of these embedding fusion methods in Appendix A. Finally, we obtain the concept embedding $\mathbf{x}_i^L \in \mathbb{R}^d$ from the learning behavior graph.

4.2 Gated Knowledge Distillation

To effectively integrate features from the knowledge perspective and the learning behavior perspective, we apply gated knowledge distillation to adaptively integrate information from the dual graph, getting more comprehensive concept embedding representations. Specifically, we treat the concept-resource hypergraph and the learning behavior graph as two independent student models. Subsequently, we regard these two graph modules as equivalent student models and employ a gating mechanism to achieve their effective integration, thereby constructing a stronger teacher model. The gating mechanism [Hochreiter and Schmidhuber, 1997] is a method for controlling information flow. It adaptively adjusts the information flow weights between the concept-resource hypergraph and the learning behavior graph based on the final node embeddings generated by the two student models, optimizing the model's decision process.

The concept node embeddings $\mathbf{x}_i^H$ and $\mathbf{x}_i^L$, generated from the concept-resource hypergraph and the learning behavior graph, respectively, contain both the textual semantic features and the graph topological structure features of the node. To construct the teacher model, we use a gating mechanism to integrate these two embeddings:

$$\mathbf{x}_i^T = g \odot \mathbf{x}_i^H + (1 - g) \odot \mathbf{x}_i^L, \quad (8)$$

here,

$$g = \sigma(\mathbf{W}_f[\mathbf{x}_i^H, \mathbf{x}_i^L] + \mathbf{b}_f). \quad (9)$$

When g is close to 1, $\mathbf{x}_i^H$ contributes more to the fused embedding; while when g is close to 0, $\mathbf{x}_i^L$ contributes more. σ represents the sigmoid activation function, and $\odot$ denotes the hadamard product. $\mathbf{W}_f$ and $\mathbf{b}_f$ are the learnable weight matrix and bias vector, respectively. Finally, we obtain the final embedding $\mathbf{x}_i^T \in \mathbb{R}^d$ of the teacher model.

The teacher model is constructed from the student models through online knowledge distillation, where both models are updated during training. The teacher model generates predictions based on the student model's output and feeds them back as supervision signals. This enables the teacher model to integrate complex concept structures and dynamic learning behavior, helping the student models learn richer features.

4.3 Ensemble Prediction Network

Learning concept prerequisite relations is essentially a binary classification task. After obtaining the final concept representations, a Siamese network [Bromley *et al.*, 1993] is used to predict whether concept c_i is a prerequisite for concept c_j. The embeddings of the two concepts are separately fed into two feedforward networks with shared weights:

$$\tilde{\mathbf{x}}_i = \mathrm{ReLU}(\mathbf{W}_1 \cdot \mathbf{x}_i + \mathbf{b}_1),\ \tilde{\mathbf{x}}_j = \mathrm{ReLU}(\mathbf{W}_1 \cdot \mathbf{x}_j + \mathbf{b}_1). \tag{10}$$

Here, $\mathbf{x}_i$ can be any of $\mathbf{x}_i^H$, $\mathbf{x}_i^L$, or $\mathbf{x}_i^T$. $\mathbf{W}_1$ and $\mathbf{b}_1$ are the weight matrix and bias vector, respectively. The two outputs are then concatenated for classification, which can be expressed as follows:

$$\mathbf{logit}_{ij} = \mathbf{W}_p \cdot [\tilde{\mathbf{x}}_i \| \tilde{\mathbf{x}}_j \| (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j) \| (\tilde{\mathbf{x}}_i \odot \tilde{\mathbf{x}}_j)] + \mathbf{b}_p, \tag{11}$$

$$p(c_i, c_j) = \sigma(\mathbf{logit}_{ij}), \tag{12}$$

where $\|$ represents embedding concatenation, and $\mathbf{W}_p$ and $\mathbf{b}_p$ are the parameterized weight matrix and bias vector, respectively. Through three Siamese networks with the same structure but different parameters, we can obtain the predicted probabilities of the two student models and the teacher model, denoted as $p^H(c_i, c_j)$, $p^L(c_i, c_j)$, and $p^T(c_i, c_j)$, respectively. For these three models, we use the binary cross-entropy loss function(BCE) to calculate the prediction loss:

$$\mathcal{L}_{preq} = \frac{1}{|\mathcal{D}|} \sum_{(c_i, c_j) \in \mathcal{D}} \mathrm{BCE}(p(c_i, c_j), y_{c_i c_j}), \tag{13}$$

where $\mathcal{D}$ represents the training dataset for concept prerequisite relations, and $|\mathcal{D}|$ is the size of the training set. $y_{c_i c_j} \in \{0, 1\}$ indicates ground truth. We denote the prediction losses of the three models as $\mathcal{L}_{preq}^H$, $\mathcal{L}_{preq}^L$, and $\mathcal{L}_{preq}^T$, respectively. Additionally, we set the classification prediction threshold γ to 0.50. When the predicted probability is greater than or equal to γ, the concept pair $< c_i, c_j >$ is considered to have a prerequisite relation; otherwise, it does not.

Next, to distill the knowledge of the integrated teacher model back to the student models, we first calculate the soft version of the predicted probabilities as temperature τ:

$$\hat{p}(c_i, c_j) = \sigma(\mathbf{logit}_{ij} / \tau), \tag{14}$$

where the temperature coefficient τ is set to 0.5. We obtain the soft versions of the predicted probabilities for the two student models and the teacher model, denoted as $\hat{p}^H(c_i, c_j)$, $\hat{p}^L(c_i, c_j)$, and $\hat{p}^T(c_i, c_j)$, respectively. Then, we calculate the distillation loss to encourage each student model to align its predictions with the teacher model's predictions:

$$\mathcal{L}_{kd} = \frac{1}{|\mathcal{D}|} \sum_{(c_i, c_j) \in \mathcal{D}} \left\| \hat{p}^T(c_i, c_j) - \hat{p}^H(c_i, c_j) \right\|_1 + \left\| \hat{p}^T(c_i, c_j) - \hat{p}^L(c_i, c_j) \right\|_1, \tag{15}$$

where $\|\cdot\|_1$ represents the L1 norm. In this case, we use the L1 norm instead of KL divergence. Studies [Wang *et al.*, 2022b] have shown that using the L1 norm as part of the loss function can more effectively constrain the model's prediction error, thereby improving the model's classification performance.

Finally, the overall loss function of the model is defined as:

$$\mathcal{L} = \mathcal{L}_{preq}^H + \mathcal{L}_{preq}^L + \mathcal{L}_{preq}^T + \lambda \cdot \mathcal{L}_{kd}, \tag{16}$$

where λ is a hyperparameter that balances the distillation loss.

Dataset	$\lvert\mathcal{C}\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{C}_{preq}^+\rvert$	$\lvert\mathcal{C}_{preq}^-\rvert$
UCD	407	654	1,007	1,007
LectureBank	246	277	601	601
MOOC	406	381	1,003	1,003

Table 1: Datasets statistics. $|\mathcal{C}|$ represents the number of concepts, and $|\mathcal{R}|$ represents the number of resources. $\left|\mathcal{C}_{preq}^+\right|$ and $\left|\mathcal{C}_{preq}^-\right|$ denote the number of positive and negative examples, respectively.

5 Experiment

5.1 Experimental Setup

Datasets. To evaluate the effectiveness of our proposed model, we select three public benchmark datasets.

- **University Course Dataset (UCD)**[1]: This dataset compiles course information [Liang *et al.*, 2017] from the computer science domain across 11 universities in the United States, covering various topics such as algorithm design, computer graphics, and neural networks.
- **LectureBank**[2]: This dataset [Li *et al.*, 2019] originates from online education platforms, covering five domains: natural language processing, machine learning, artificial intelligence, deep learning, and information retrieval.
- **MOOC**[3]: This dataset [Liang *et al.*, 2017] is derived from video playlists in the MOOC corpus and includes the subtitle texts of videos from 38 playlists in the computer science department.

Data Preprocessing. Since our model will utilize the user clickstream data from Wikipedia in 2019 as the learning behavior data for the learning behavior graph. In LectureBank, 74 concepts could not be matched to Wikipedia page names and were merged with others of the same meaning, so we removed them without affecting domain coverage. Additionally, we follow the method in previous research [Sun *et al.*, 2024] to generate negative samples equal in number to the positive samples, enhancing the model's robustness. Specifically, half of the negative samples were created through unrelated random sampling, while the other half are reversed pairs of the original positive samples. The final statistics of the three experimental datasets are shown in Table 1.

Baseline Methods. We compare our DGCPL with eight graph-based methods for predicting concept prerequisite relations: LCPRE [Sun *et al.*, 2024], HGAPNet [Mazumder *et al.*, 2023], MHAVGAE [Zhang *et al.*, 2022a], ConLearn [Sun *et al.*, 2022], R-VGAE(T) [Li *et al.*, 2020], R-VGAE(P) [Li *et al.*, 2020], VGAE [Kipf and Welling, 2016], and GAE [Kipf and Welling, 2016]. Additionally, we considered five traditional classification baseline methods: PREREQ [Roy *et al.*, 2019], RefD [Liang *et al.*, 2015], Random Forest (RF), Support Vector Machine (SVM), and Naive Bayes (NB). Please refer to Appendix B for detailed descriptions.

Evaluation Metrics. For the fair and extensive evaluation, we select three common evaluation metrics to assess the per-

[1]https://github.com/suderoy/PREREQ-IAAI-19
[2]https://github.com/Yale-LILY/LectureBank
[3]https://github.com/suderoy/PREREQ-IAAI-19

Method	UCD			LectureBank			MOOC		
	ACC	F1	AUC	ACC	F1	AUC	ACC	F1	AUC
NB	0.5495	0.5845	0.5369	0.5207	0.5323	0.5262	0.5124	0.5288	0.5357
SVM	0.5743	0.6161	0.5601	0.5455	0.5669	0.5361	0.5821	0.6000	0.5923
RF	0.6683	0.6455	0.7419	0.6777	0.6723	0.8245	0.7363	0.7135	0.8084
RefD	0.7620	0.7110	0.7520	0.3900	0.5570	0.4760	0.5870	0.4140	0.4900
PREREQ	0.5433	0.5866	0.6702	0.4975	0.5130	0.5557	0.5429	0.5746	0.6248
GAE	0.6642	0.6631	0.6955	0.6877	0.6864	0.7651	0.6721	0.6700	0.6995
VGAE	0.6933	0.6927	0.7552	0.6907	0.6898	0.7534	0.6664	0.6656	0.7144
R-VGAE(T)	0.6849	0.6618	0.7646	0.6660	0.6345	0.7942	0.5926	0.5435	0.6602
R-VGAE(P)	0.7369	0.7220	0.8325	0.5678	0.4714	0.8107	0.5344	0.4108	0.8730
ConLearn	0.7822	0.7684	0.8529	0.8017	0.7931	0.8541	0.7562	0.7200	0.8472
MHAVGAE	0.7875	0.7952	0.8645	0.7263	0.7401	0.8213	0.7475	0.7642	0.8759
HGAPNet	0.8200	0.8043	<u>0.8998</u>	0.8167	<u>0.8136</u>	**0.8803**	<u>0.8550</u>	<u>0.8497</u>	<u>0.9014</u>
LCPRE	<u>0.8366</u>	<u>0.8216</u>	0.8884	<u>0.8182</u>	0.8000	0.8514	0.8258	0.8223	0.8898
DGCPL (Ours)	**0.8564**	**0.8557**	**0.9053**	**0.8347**	**0.8246**	<u>0.8795</u>	**0.8756**	**0.8718**	**0.9236**
Improve rate	1.98% ↑	3.41% ↑	0.55% ↑	1.65% ↑	1.10% ↑	0.08% ↓	2.06% ↑	2.21% ↑	2.22% ↑

Table 2: Performance Comparison. The best performance is highlighted in **bold**, and the runner-up is <u>underlined</u>. ↑(↓) indicates the improvement (decline) of our model compared to the best baseline.

Method	UCD	LectureBank	MOOC
Ours DGCPL	**0.8557**	**0.8246**	**0.8718**
Ours w/o GKD	0.8374	0.8226	0.8528
Ours w/o CRHG	0.8177	0.8160	0.8168
Ours w/o LBG	0.8235	0.8130	0.8469

Table 3: The ablation study of DGCPL.

formance of all methods: Accuracy (ACC), F1 Score (F1), and Area Under the ROC Curve (AUC).

Implementation Details. We split each datasets into training, validation, and test sets with a ratio of 8:1:1. Our proposed model is trained using the Adam optimizer for a total of 50 epochs. The learning rate lr, batch size, and classification prediction threshold γ are set to 1E-4, 16, and 0.50. For the UCD, LectureBank, and MOOC datasets, the number of graph neural network layers ℓ is set to 3, 2, and 2; the weight decay is set to 1E-4, 1E-2, and 1E-3; and the distillation loss weight λ in the overall loss function is set to 1E-6, 1E-1, and 1E-5, respectively. We use BERT [Devlin *et al.*, 2019] as the pretrained language model. Detailed analysis of the impact of the pretrained language model is provided in Appendix C. All experiments is implemented on the Linux sever with one RTX 4090D GPU using the PyTorch framework.

5.2 Overall Performance

In Table 2, we present the comparative results of all methods on the three datasets. Our model generally outperforms existing baselines across all datasets. Although the AUC metric shows a slight decline on the LectureBank dataset, this minor difference may result from the complex concept relations or the imbalanced sample distribution in the dataset. However, DGCPL achieves significant improvements in AUC on the other two datasets, demonstrating its adaptability and robustness in handling different types of educational data.

In summary, we draw the following conclusions: (1) DGCPL significantly outperforms existing baseline models in all evaluation metrics on the three datasets, showcasing its strong capability in learning concept prerequisite relations. (2) Graph-based methods perform better than traditional methods because the relations between concepts, between concepts and resources, and between resources form a knowledge network, which provides a significant advantage. (3) The HGAPNet and LCPRE models perform as the next best methods. They fully leverage the complex interaction relations between concepts and resources, but lack the perspective of learning behavior.

5.3 Ablation Experiment

In this study, to evaluate the effectiveness of each module in DGCPL, we designed three variants and compared them with these variants in terms of F1. The ablation study results in Table 3 show that: (1) Removing the Gated Knowledge Distillation (GKD) leads to a slight performance decrease. This reduction in performance occurs because the model loses its ability to integrate knowledge across graphs. (2) Removing the Concept Resource Hypergraph (CRHG) along with the associated GKD significantly reduces performance on all datasets, highlighting the importance of concept embeddings from the knowledge perspective. (3) Removing the Learning Behavior Graph (LBG) along with the associated GKD results in a substantial performance drop. This group learning navigation behavior indeed reflects the prerequisite relations between concepts and deserves attention.

5.4 Quality Analysis

In a knowledge domain, the number of neighboring concepts (i.e., prerequisite concepts and successor concepts) varies across different concepts. To evaluate our model's performance under different numbers of neighbors, we conduct experiments on LectureBank, dividing concepts into low-degree, medium-degree, and high-degree groups based on the trichotomies of neighbor count. The pie chart in Figure 3 shows the proportion of concepts in each group. We

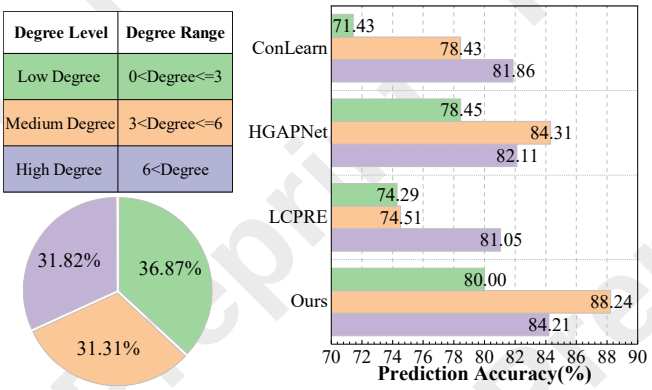

Figure 3: The quality analysis on the LectureBank.

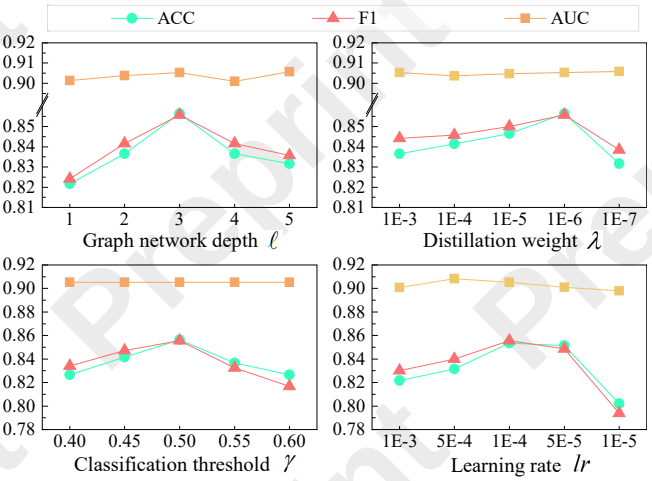

Figure 4: The hyperparameter analysis of DGCPL on the UCD.

compare our method with three advanced baselines on the test set. From the bar chart in Figure 3, we observe: (1) DGCPL outperforms the baselines in predicting low-degree, medium-degree, and high-degree concepts, demonstrating robustness in predicting sparse prerequisite relations. (2) For each model, the prediction performance for low-degree concepts is consistently the lowest. Current graph-based methods tend to degrade in performance when knowledge relations are sparse, highlighting the importance of considering the learning behavior perspective. However, Our model also excels in predicting low-degree concepts.

5.5 Hyperparameter Analysis

As illustrated in Figure 4, we evaluate the performance of the model on UCD under different hyperparameter settings. The results are summarized as follows: (1) The model performs best when the graph neural network depth ℓ is 3, as lower depths fail to capture complex graph structures, and deeper networks may overfit or excessively smooth information. (2) The distillation loss weight λ has an optimal range, with performance improving as the weight decreases from 1E-3 to 1E-6, but declining when it drops further to 1E-7 due to insufficient learning from the teacher model. (3) The classification prediction threshold γ of 0.50 achieves optimal performance; lower thresholds lead to an increase in false positives, while higher thresholds reduce the model's ability to correctly identify positive samples. (4) The optimal learning rate lr is 1E-4, which ensures fast and stable convergence. The same method is applied to select the best hyperparameter settings for the other datasets.

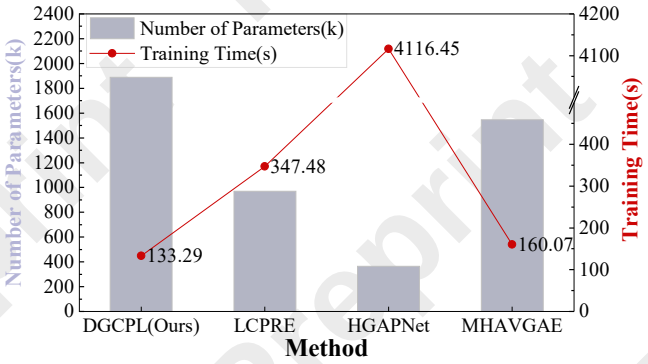

Figure 5: The training overhead analysis on the MOOC.

5.6 Training Overhead Analysis

In this study, we compare the model parameter sizes and training times between DGCPL and three advanced baselines on MOOC. As illustrated in Figure 5, we draw the following conclusions: (1) Despite the increased parameter count due to the dual graph structure, DGCPL achieves faster learning of better concept embeddings. The dual graph structure captures concept features from both knowledge structure and learning behavior perspectives, and gated knowledge distillation accelerates convergence, reducing the number of epochs needed for training. (2) LCPRE incurs significant overhead due to depth-first search for exploring concept-resource paths. (3) HGAPNet updates all node and edge features, slowing convergence, and its small batch size further hinders speed. (4) MHAVGAE has faster convergence, but the number of attention heads affects its optimization speed.

6 Conclusion

This paper proposes a dual graph distillation method for learning concept prerequisite relations. The method aims to construct a dual graph structure, including the concept-resource hypergraph and the learning behavior graph, from the perspective of knowledge and the perspective of learning behavior. Then we effectively integrate the dual graph through gated knowledge distillation to construct a more robust teacher model to predict concept prerequisite relations. This method not only leverages the high-order knowledge relations between concepts and resources, but also integrates the learner's learning behavior features, thereby enhancing the model's ability to understand and model concept prerequisite relations. Extensive experiments on three publicly benchmark datasets demonstrate that DGCPL achieves state-of-the-art performance for learning concept prerequisite relations.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62377009, 62407013 and 62207011.

References

[Bromley *et al.*, 1993] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a siamese time delay neural network. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference]*, pages 737–744. Morgan Kaufmann, 1993.

[Cui *et al.*, 2024] Chaoran Cui, Yumo Yao, Chunyun Zhang, Hebo Ma, Yuling Ma, Zhaochun Ren, Chen Zhang, and James Ko. DGEKT: A dual graph ensemble learning method for knowledge tracing. *ACM Trans. Inf. Syst.*, 42(3):78:1–78:24, 2024.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[Dong *et al.*, 2024] Yao Dong, Yuxi Liu, Yongfeng Dong, Yacong Wang, and Min Chen. Multi-knowledge enhanced graph convolution for learning resource recommendation. *Knowl. Based Syst.*, 291:111521, 2024.

[Feng *et al.*, 2019] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3558–3565. AAAI Press, 2019.

[Gao *et al.*, 2023] Weibo Gao, Hao Wang, Qi Liu, Fei Wang, Xin Lin, Linan Yue, Zheng Zhang, Rui Lv, and Shijin Wang. Leveraging transferable knowledge concept graph embedding for cold-start cognitive diagnosis. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 983–992. ACM, 2023.

[Gong *et al.*, 2023] Linrui Gong, Shaohui Lin, Baochang Zhang, Yunhang Shen, Ke Li, Ruizhi Qiao, Bo Ren, Muqing Li, Zhou Yu, and Lizhuang Ma. Adaptive hierarchy-branch fusion for online knowledge distillation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 7731–7739. AAAI Press, 2023.

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hu *et al.*, 2021] Cheng Hu, Kui Xiao, Zesong Wang, Shihui Wang, and Qifeng Li. Extracting prerequisite relations among wikipedia concepts using the clickstream data. In *Knowledge Science, Engineering and Management - 14th International Conference, KSEM 2021, Proceedings, Part I*, volume 12815 of *Lecture Notes in Computer Science*, pages 13–26. Springer, 2021.

[Huang *et al.*, 2024] Tao Huang, Xinjia Ou, Huali Yang, Shengze Hu, Jing Geng, Junjie Hu, and Zhuoran Xu. Remembering is not applying: Interpretable knowledge tracing for problem-solving processes. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024*, pages 3151–3159. ACM, 2024.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016.

[Li *et al.*, 2019] Irene Li, Alexander R. Fabbri, Robert R. Tung, and Dragomir R. Radev. What should I learn first: Introducing lecturebank for NLP education and prerequisite chain learning. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6674–6681. AAAI Press, 2019.

[Li *et al.*, 2020] Irene Li, Alexander R. Fabbri, Swapnil Hingmire, and Dragomir R. Radev. R-VGAE: relational-variational graph autoencoder for unsupervised prerequisite chain learning. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pages 1147–1157. International Committee on Computational Linguistics, 2020.

[Li *et al.*, 2024] Zheng Li, Xiang Li, Lingfeng Yang, Renjie Song, Jian Yang, and Zhigeng Pan. Dual teachers for self-knowledge distillation. *Pattern Recognit.*, 151:110422, 2024.

[Liang *et al.*, 2015] Chen Liang, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. Measuring prerequisite relations among concepts. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1668–1674. The Association for Computational Linguistics, 2015.

[Liang *et al.*, 2017] Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, and C. Lee Giles. Recovering concept prerequisite relations from university course dependencies. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4786–4791. AAAI Press, 2017.

[Mazumder *et al.*, 2023] Debjani Mazumder, Jiaul H. Paik, and Anupam Basu. A graph neural network model for concept prerequisite relation extraction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023*, pages 1787–1796. ACM, 2023.

[O'Dea and Stern, 2022] Xianghan (Christine) O'Dea and Julian Stern. Virtually the same?: Online higher education in the post covid-19 era. *Br. J. Educ. Technol.*, 53(3):437–442, 2022.

[Pechác *et al.*, 2024] Matej Pechác, Michal Chovanec, and Igor Farkas. Self-supervised network distillation: An effective approach to exploration in sparse reward environments. *Neurocomputing*, 599:128033, 2024.

[Romero *et al.*, 2015] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, pages 1–13, 2015.

[Roy *et al.*, 2019] Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. Inferring concept prerequisite relations from online educational resources. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 9589–9594. AAAI Press, 2019.

[Sayyadiharikandeh *et al.*, 2019] Mohsen Sayyadiharikandeh, Jonathan Gordon, José Luis Ambite, and Kristina Lerman. Finding prerequisite relations using the wikipedia clickstream. In *Companion of The 2019 World Wide Web Conference, WWW 2019*, pages 1240–1247. ACM, 2019.

[Song *et al.*, 2023] Lingyun Song, Mengting He, Xuequn Shang, Chen Yang, Jun Liu, Mengzhen Yu, and Yu Lu. A deep cross-modal neural cognitive diagnosis framework for modeling student performance. *Expert Syst. Appl.*, 230:120675, 2023.

[Sun *et al.*, 2022] Hao Sun, Yuntao Li, and Yan Zhang. Conlearn: Contextual-knowledge-aware concept prerequisite relation learning with graph neural network. In *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM 2022*, pages 118–126. SIAM, 2022.

[Sun *et al.*, 2024] Jingwen Sun, Yu He, Yiyu Xu, Jingwei Sun, and Guangzhong Sun. A learning-path based supervised method for concept prerequisite relations extraction in educational data. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024*, pages 2168–2177. ACM, 2024.

[Talukdar and Cohen, 2012] Partha P. Talukdar and William W. Cohen. Crowdsourced comprehension: Predicting prerequisite structure in wikipedia. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012*, pages 307–315. The Association for Computer Linguistics, 2012.

[Tong *et al.*, 2020] Zekun Tong, Yuxuan Liang, Changsheng Sun, David S. Rosenblum, and Andrew Lim. Directed graph convolutional network. *CoRR*, abs/2004.13970, 2020.

[Wang and Liu, 2016] Shuting Wang and Lei Liu. Prerequisite concept maps extraction for automaticassessment. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Companion Volume*, pages 519–521. ACM, 2016.

[Wang *et al.*, 2022a] Chao Wang, Hengshu Zhu, Peng Wang, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. Personalized and explainable employee training course recommendations: A bayesian variational approach. *ACM Trans. Inf. Syst.*, 40(4):70:1–70:32, 2022.

[Wang *et al.*, 2022b] Jing Wang, Xin Geng, and Hui Xue. Re-weighting large margin label distribution learning for classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):5445–5459, 2022.

[Yang *et al.*, 2023] Chuanguang Yang, Zhulin An, Helong Zhou, Fuzhen Zhuang, Yongjun Xu, and Qian Zhang. Online knowledge distillation via mutual contrastive learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(8):10212–10227, 2023.

[Yin *et al.*, 2023] Yu Yin, Le Dai, Zhenya Huang, Shuanghong Shen, Fei Wang, Qi Liu, Enhong Chen, and Xin Li. Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In *Proceedings of the ACM Web Conference 2023, WWW 2023*, pages 855–864. ACM, 2023.

[Yu *et al.*, 2024] Shenbao Yu, Yifeng Zeng, Fan Yang, and Yinghui Pan. Causal-driven skill prerequisite structure discovery. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 20604–20612. AAAI Press, 2024.

[Zagoruyko and Komodakis, 2017] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Conference Track Proceedings*, pages 1–13. OpenReview.net, 2017.

[Zhang *et al.*, 2021] Xitong Zhang, Yixuan He, Nathan Brugnone, Michael Perlmutter, and Matthew J. Hirn. Magnet: A neural network for directed graphs. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, pages 27003–27015, 2021.

[Zhang *et al.*, 2022a] Juntao Zhang, Nanzhou Lin, Xuelong Zhang, Wei Song, Xiandi Yang, and Zhiyong Peng. Learning concept prerequisite relations from educational data via multi-head attention variational graph auto-encoders. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining*, pages 1377–1385. ACM, 2022.

[Zhang *et al.*, 2022b] Linfeng Zhang, Chenglong Bao, and Kaisheng Ma. Self-distillation: Towards efficient and compact neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(8):4388–4403, 2022.

[Zhang *et al.*, 2025] Miao Zhang, Jiawei Wang, Kui Xiao, Shihui Wang, Yan Zhang, Hao Chen, and Zhifei Li. Learning concept prerequisite relation via global knowledge relation optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1638–1646. AAAI Press, 2025.