Concept Prerequisite Relation Prediction by Using Permutation-Equivariant Directed Graph Neural Networks *

Xiran Qu XRQU@MAIL.NWPU.EDU.CN
Xuequn Shang SHANG@NWPU.EDU.CN
Yupei Zhang† YPZHAANG@NWPU.EDU.CN
School of Computer Science, Northwestern Polytechnical University, Xi'an, 710129, China.
Big data Storage and Management MIIT Lab, Xi'an, 710129, China.

Editor: Editor's name

Abstract

This paper studies the problem of CPRP, concept prerequisite relation prediction, which is a fundamental task in using AI for education. CPRP is usually formulated into a link-prediction task on a relationship graph of concepts and solved by training the graph neural network (GNN) model. However, current directed GNNs fail to manage graph isomorphism which refers to the invariance of non-isomorphic graphs, reducing the expressivity of resulting representations. We present a permutation-equivariant directed GNN model by introducing the Weisfeiler-Leman test into directed GNN learning. Our method is then used for CPRP and evaluated on three public datasets. The experimental results show that our model delivers better prediction performance than the state-of-the-art methods.

Keywords: Concept Prerequisite Relation, permutation-equivariant GNNs, Weisfeiler-Leman Test, Directed Graph Learning, AI for Education.

1. Introduction

With the continuous advancement of dissemination methods, an increasing number of educational resources are becoming available for people to learn Fischer et al. (2020). Therefore, finding prerequisite relationships among concepts has become an important issue requiring investigation in the field of AI for education Pan et al. (2017); Roy et al. (2019). Generally, this Concept Prerequisite Relation Prediction task, CPRP, is modeled as the link prediction problem in many studies Sun et al. (2022); Roy et al. (2019). Applications of CRPR involve material recommendation Guan et al. (2023), learning path planning Shi et al. (2020), and optimization of problem-solving paths Le et al. (2023).

There are numerous approaches to solving the link prediction problem, including probabilistic models, spectral clustering, evolutionary algorithms, and deep graph learning models Kumar et al. (2020). Currently, graph neural networks (GNNs) have become the benchmark method and presented state-of-the-art performance in link prediction Cai et al. (2021), including many GNN-based solutions to CPRP Roy et al. (2019); Jia et al. (2021); Sun et al. (2022); Mazumder et al. (2023). To improve the capability of GNNs, Long et al. (2022) and Chamberlain et al. (2022) proposed to pre-training node features through the method of

* This work was supported in part by National Natural Science Foundation of China (62272392, U22A2025) and the funding for Teaching & Learning Reform at NPU (2023JGZ14).
† Corresponding Author

© 2023 X. Qu, X. Shang & Y. Zhang.

graph reconstruction and utilizing subgraph sketches to pass messages in subgraph GNNs, aiming to enhance the capabilities of GNNs and reduce computational costs. Thess models mitigated the expressive limitations, such as the inability to count triangles and distinguish automorphic nodes. Persistent homology was also adopted in the work of Yan et al. (2021) to extract topological information from graphs, which was integrated with node features to enhance the expressive power of GNNs for link prediction. Besides, the Weisfeiler-Leman test was introduced by Morris et al. (2019) to improve the capability of differentiating graph isomorphism in undirected GNN learning Huang et al. (2022).

However, the CPRP problem is usually formulated into the directed-link prediction in a directed graph Sun et al. (2022). Rather than the undirected graph, the edges can indicate the prerequisite relation, describing the flow of information from one node to another. The difference makes undirected GNNs not directly applicable to directed graphs. Hence, Salha et al. (2019) designed a new gravity-inspired decoder to extend the graph Autoencoders, while Wu et al. (2019) adopted two weight matrixes for forward edges and backward edges, respectively, to perform message passing between graph nodes. Nevertheless, there is a lack of studies on improving the expressive power of direct GNNs for CPRP.

In this paper, we extended the Weisfeiler-Leman test-based GNN model, i.e., SpeqNets recently introduced by Morris et al. (2022), into the directed graphs for CPRP. The proposed framework contributes to learning the permutation-equivariant direct GNNs and improving the prediction performance of CPRP by distinguishing non-isomorphism graphs. Experimental results on three public datasets manifest that our method performs better than the state-of-the-art methods Sun et al. (2022); Jia et al. (2021); Roy et al. (2019).

2. CPRP: Concept Prerequisite Relation Prediction

The problem of CPRP refers to predicting prerequisite relations between knowledge concepts involved in learning Sun et al. (2022). For example, one should learn the knowledge concept (KC) of "conditional probability distribution" before learning "Bayesian theory." As usual, CPRP can be formulated into the directed-link prediction in a KC graph.

2.1. Problem formulation

Denote by $G = (V, E)$ a directed KC graph with a vertex set $V = \{v_1, v_2, \ldots, v_N\}$, where v_i represents a KC in our study, and an edge set $E = \{e_{ij}\}_{1 \leq i,j \leq N}$, where e_{ij} is the prerequisite relation $v_i \rightarrow v_j$ meaning that the KC v_i is a prerequisite KC v_j. Denote by $A \in R^{N \times N}$ the adjacency matrix of G with each element being 0 or 1. CPRP on G can be written into

$$\mathcal{P}(v_p, v_q) = \mathcal{M}(\mathcal{G}(v_p), \mathcal{G}(v_q)) \tag{1}$$

where v_p and v_q are two KCs; $\mathcal{P}$ is the probability of whether the relation e_{pq} exists; $\mathcal{G}$ is a representation model that integrates KC information into a vector; the function $\mathcal{M}$ aims to obtain the existing probability of the prerequisite relation e_{pq}.

2.2. Previous Methods

Early works of CPRP usually extracted handcrafted features for $\mathcal{G}$, such as the contextual and structural features Pan et al. (2017), while recent works are focused on designing deep-learning models, such as Siamese networks Roy et al. (2019) and GNNs Sun et al. (2022).

Inspired by the promising performance, we implemented $\mathcal{G}$ by training a GNN model in this study. GNNs aim to learn node representations in a graph by iteratively aggregating the neighborhood features. In each layer, the feature of the node v is updated by merging the information transmitted from its neighbors, expressed as

$$\mathbf{f}_v^{(t)} = \mathcal{F}_{mer}^{W_1^{(t)}}(\mathbf{f}_v^{(t-1)}, \mathcal{F}_{agg}^{W_2^{(t)}}(\{\mathbf{f}^{(t-1)}(u)|u \in \mathcal{N}(v)\})) \tag{2}$$

where $\mathbf{f}_v^{(t)}$ indicates the feature vector of the node v at the t-th layer; $\mathcal{F}_{mer}^{W_1^t}$ is the merging function with learned parameters $W_1^{(t)}$ and $\mathcal{F}_{agg}^{W_2^{(t)}}$ is the aggregating function with network parameters $W_2^{(t)}$; $\mathcal{N}(v)$ delivers the neighbors of node v Wu et al. (2020).

With the node representations from GNNs, many methods can estimate the prerequisite relations by employing similarity metrics or classical classifiers for $\mathcal{M}$ Liang et al. (2018). However, the previous studies fail to consider the problem of graph isomorphism of the KC graphs, leading to low expressive powers for CPRP.

3. Our CPRP Method

To achieve fine representations of KCs, our method adopts the well-known Weisfeiler-Leman test Morris et al. (2021) to guide the GNN training in the KC graph G. With the obtained KC representations, the Siamese network computes the link probability, shown in Fig. 1.

3.1. Weisfeiler-Leman Test

Denote by $S = \{(s_1, s_2, ..., s_k)|s_i \in V, i \in I_k\}$ a k-tuple S of vertices in the KC graph G, where I_k indicates the first k natural numbers; s_i is the i-th element of S specified to a $v_j \in V$. Let $V^k(G)$ be the collection of all k-tuples from the graph G. Weisfeiler-Leman (WL) test is to assign labels to each tuple in $V^k(G)$ and then iteratively relabel these tuples by merging their neighborhood labels. Here, the j-th neighborhood of the tuple S is yielded by replacing its j-th element with every node $v \in V$, i.e., $\mathcal{N}_j^k(S, v) = \{(s_1, s_2, ..., s_{j-1}, v, s_{j+1}, ...s_k)|v \in V\}$ Morris et al. (2021). Based on all neighborhoods of S, the WL test usually uses a predefined route to compute a new label for the merged node in the i-th iteration, i.e.,

$$l_i(S) = \{\{(C_i^k(\mathcal{N}_1^k(S, v), \mathcal{N}_2^k(S, v), ..., \mathcal{N}_k^k(S, v)|v \in V\}\} \tag{3}$$

where $l_i(S)$ represents the obtained label and C_i^k indicates a predefined function that maps all tuples V^k to new labels. Then, the iterative labeling for S can be expressed as

$$C_i^k(S) = Relabel((C_{i-1}^k(S), l_i(S))) \tag{4}$$

where $Relabel(a, b)$ is to achieve $a \leftarrow b$.

The k-WL test is a powerful tool for distinguishing isomorphic graphs. Let G and H be two graphs. If the number of k-tuples with a specific label differs between graphs G and H at any iteration, then the two graphs are non-isomorphic. Specifically, 1-WL employs the 1-hop neighbors for $\mathcal{N}$ in Eq. (3). With the increase of k, the k-WL algorithm can become more capable of distinguishing non-isomorphic graphs.

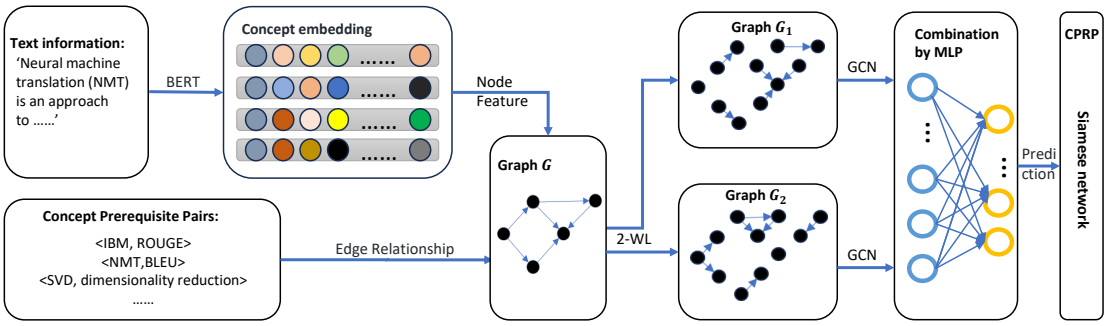

Figure 1: Workflow of the proposed method for CPRP.

3.2. Main Steps

3.2.1. KC Graph Construction

The first step of the proposed method is to achieve the node features for the given directed KC graph G using the pre-trained BERT [1]. More specifically, the textual descriptions of KCs from the datasets or Wikipedia were obtained and fed into BERT to extract the KC embedding for V. With the given E, we achieved the KC graph G Devlin et al. (2018).

3.2.2. Weisfeiler Leman Guided Directed GNNs

On the resulting graph G, we proposed the directed GNN model guided by the k-WL test to achieve KC representations in this step. To be different from undirected graphs, we denote by $\mathcal{N}_{out}(v)$ the set of all out-neighbors of the node v. For two connected nodes v and w in a directed graph, if there exists an edge pointing from v to w, node w is defined as an out-neighbor of node v. In the proposed method, we redefined the k-tuple as follows,

$$\widehat{V}(G)^k := \{(s_1, s_2, ..., s_k) | s_i \in V, i \in I_k; \exists j \in I_{i-1} : s_i \in \mathcal{N}_{out}(s_j) \cup \{s_j\}\}. \tag{5}$$

The feature representation of k-tuple $(s_1, s_2, ..., s_k)$ is here given by $[\mathbf{n_1} : \mathbf{n_2} : ... : \mathbf{n_k}]$, where $\mathbf{n_i}$ denotes the embedding of node s_i; $[\cdot : \cdot]$ denotes the concatenation of vectors. For $\forall S \in \widehat{V}(G)^k$, the j-th out-neighbor of S can be cast as:

$$\widehat{\mathcal{N}}_j^k(S) := \{(s_1, s_2, ..., s_{j-1}, v, s_{j+1}, ...s_k) | v \in \mathcal{N}_{out}(s_j)\} \tag{6}$$

For all k-tuples, the neighbor relationships generated on the j-th element are used to construct a graph representation as G_j. For all graphs $G_i (i \in I_k)$, each node in the graph G_i represents a k-tuple in $\widehat{V}(G)^k$. For all graphs G_i, graph neural networks are constructed separately for training. After each layer of training is completed, the features of the same k-tuple on different graphs G_i are fused using multilayer perceptron (MLP). The expression of k-tuple S at the t-th layer is as follows:

$$\widehat{\mathbf{f}}_S = \mathcal{F}_{MLP}(\mathcal{F}_S^{(1)}, \mathcal{F}_S^{(2)}, ..., \mathcal{F}_S^{(k)}) \tag{7}$$

1. https://huggingface.co/bert-large-uncased

where $\widehat{\mathbf{f}}_S$ is defined as the representation of k-tuple S at layer t. For $\forall i \in I_k$, $\mathcal{F}_S^{(i)}$ denotes the representation of k-tuple S in graph G_i after passing through the GCN Kipf and Welling (2016) designed for graph G_i. $\mathcal{F}_{MLP}$ denotes the multilayer perceptron. After several layers of learning, we obtain the representations of all k-tuples.

The representations of k-tuples obtained through GNN training are finally distributed to the nodes of G using an average allocation scheme, shown as follows:

$$\mathbf{x}_j^{(i)} = \frac{1}{\#\{S|\forall S \in \widehat{V}(G)^k, s_i = v_j\}} \sum_{\{S|\forall S \in \widehat{V}(G)^k, s_i = v_j\}} \mathbf{h}_S \tag{8}$$

where s_i represents the i-th element in the k-tuple S. $\mathbf{x}_j^{(i)}$ represents the transmission of feature expressions of k-tuples back to the representations of concept v_j, based on the i-th element in each k-tuple. $\mathbf{h}_S$ denotes the feature of k-tuple S. $\#\{...\}$ denotes the number of elements in the set $\{...\}$. For $\forall i$, all $\mathbf{x}_j^{(i)}$ vectors are merged and fed into a multi-layer perceptron to obtain node representations in G.

$$\mathbf{f}_j = \mathcal{F}_{MLP}([\mathbf{x}_j^{(1)} : \mathbf{x}_j^{(2)} : ... : \mathbf{x}_j^{(k)}]) \tag{9}$$

where $\mathbf{f}_j$ represents the learned concept v_j representations.

3.2.3. Prediction Network

After obtaining the KC representations, the Siamese network is employed here to predict the probability that the concept v_p is a prerequisite concept of concept v_q in Eq. (1). Let $\tilde{\mathbf{e}}_p$ and $\tilde{\mathbf{e}}_q$ be the representations of v_p and v_q from the two feed-forward networks with shared weights in the Siamese network. The probability between v_p and v_q is achieved via

$$\mathcal{P}(v_p, v_q) = \sigma\left(W\left[\tilde{\mathbf{e}}_i : \tilde{\mathbf{e}}_j : \tilde{\mathbf{e}}_i - \tilde{\mathbf{e}}_j : \tilde{\mathbf{e}}_i \odot \tilde{\mathbf{e}}_j : 1\right]\right) \tag{10}$$

where σ represents the sigmoid operator and $\odot$ is Hadamard product. Finally, we used the cross-entropy loss to compute the training loss for the deep framework.

4. Experiments

We used LectureBank Li et al. (2019), University Courses Liang et al. (2017), and ML of MOOCs Pan et al. (2017) to evaluate the performance and compare to the state-of-the-art methods, including binary classification models (SVM, LR, NB, RF) Pan et al. (2017), RefD Liang et al. (2015), GAE Li et al. (2019), VGAE Li et al. (2019), PREREQ Roy et al. (2019), CPRL Jia et al. (2021), and Conlearn Sun et al. (2022). Besides, we employed precision, recall, and F1-score as evaluation metrics to measure the performance.

For BERT, we leveraged a combination of course lecture information and Wikipedia for concept description extraction in the MOOC and University Courses datasets. For the LectureBank dataset, we utilized the text information from the Wikipedia URLs of each concept in the dataset. When extracting vectors from texts, the max token size of BERT was set to 256 for all three datasets.

For our method, the parameter k of k-WL was set to 2. We used Adam as the optimizer with a learning rate of 0.00002 for all experiments. The batch size was set to 256 for the

Table 1: Comparison between the results of the baseline model and our model

Datasets	Metric	NB	SVM	LR	RF	RefD	GAE	VGAE	PREREQ	CPRL	ConLearn	Ours
MOOCs	Precision	0.577	0.668	0.748	0.375	0.784	0.293	0.266	0.448	0.800	0.895	**0.915**
	Recall	0.623	0.577	0.270	0.669	0.188	0.733	0.647	0.592	0.642	0.850	**0.860**
	F1-score	0.599	0.619	0.397	0.481	0.303	0.419	0.377	0.510	0.712	0.872	**0.887**
LectureBank	Precision	0.670	0.857	0.744	0.855	0.666	0.462	0.417	0.590	**0.861**	0.831	0.857
	Recall	0.640	0.692	0.744	0.681	0.228	0.811	0.575	0.502	0.858	0.960	**0.960**
	F1-score	0.655	0.766	0.744	0.758	0.339	0.589	0.484	0.543	0.860	0.891	**0.906**
University Courses	Precision	0.478	0.796	0.595	0.739	**0.919**	0.450	0.470	0.468	0.689	0.611	0.822
	Recall	0.649	0.635	0.546	0.480	0.415	0.886	0.694	0.916	0.760	**0.966**	0.74
	F1-score	0.550	0.707	0.569	0.582	0.572	0.597	0.560	0.597	0.723	0.749	**0.778**

MOOC dataset and LectureBank dataset and 512 for the University Course dataset. The models were trained for 4000 epochs for all experiments until the loss stabilized. As for the baseline methods, we used default parameters as in their original implementations.

For all three datasets, we selected 80% of the concept prerequisite pairs as the training set and 20% of the concept prerequisite pairs as the test set. Negative samples were generated by randomly selecting unrelated phrase pairs from the vocabulary, along with the reverse pairs of the original positive samples. The results are recorded in Table 1.

From Table 1, we can draw the following observations: the four binary classifiers, i.e., NB, SVM, LR, and RF, perform weak on the three datasets due to hand-crafted features, as well as RefD; GAE, VGAE, and PREREQ exploit the prerequisite relation information between KCs, resulting in improved performance; the CPRL method fails to delve into the textual information behind the concepts, leading to better performance; finally, both our algorithm and ConLearn extract prior information using the large-scale language model BERT and yield the best evaluations. Importantly, compared to ConLearn, our method uses 2-WL to integrate the structural information of the graph deeply, resulting in the best performance in terms of F1-score. All observations manifest that the introduced WL test into direct GNN is effective for CPRP.

5. Conclusion

This paper proposes a directed graph neural network based on the Weisfeiler-Leman algorithm to address the CPRP problem. Our method leverages BERT for KC text embeddings and redefines the k-tuple in the directed KC graph. Then, the 2-WL test is implemented to train a permutation-equivariant GNN. With the KC representation from GNN, the Siamese network computes the prediction probability of a KC link. Extensive experiments on three datasets demonstrate the superiority of the proposed method, achieving a more advanced performance than the state-of-the-art approaches of CPRP. Our future work will consider more evaluation results and topological information on the graph.

References

Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5103–5113, 2021.

Benjamin Paul Chamberlain, Sergey Shirobokov, Emanuele Rossi, Fabrizio Frasca, Thomas Markovich, Nils Hammerla, Michael M Bronstein, and Max Hansmire. Graph neural networks for link prediction with subgraph sketching. *arXiv preprint arXiv:2209.15486*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Christian Fischer, Zachary A Pardos, Ryan Shaun Baker, Joseph Jay Williams, Padhraic Smyth, Renzhe Yu, Stefan Slater, Rachel Baker, and Mark Warschauer. Mining big data in education: Affordances and challenges. *Review of Research in Education*, 44(1): 130–160, 2020.

Quanlong Guan, Fang Xiao, Xinghe Cheng, Liangda Fang, Ziliang Chen, Guanliang Chen, and Weiqi Luo. Kg4ex: An explainable knowledge graph-based approach for exercise recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 597–607, 2023.

Zhongyu Huang, Yingheng Wang, Chaozhuo Li, and Huiguang He. Going deeper into permutation-sensitive graph neural networks. In *International Conference on Machine Learning*, pages 9377–9409. PMLR, 2022.

Chenghao Jia, Yongliang Shen, Yechun Tang, Lu Sun, and Weiming Lu. Heterogeneous graph neural networks for concept prerequisite relation learning in educational data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2036–2047, 2021.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Ajay Kumar, Shashank Sheshar Singh, Kuldeep Singh, and Bhaskar Biswas. Link prediction techniques, applications, and performance: A survey. *Physica A: Statistical Mechanics and its Applications*, 553:124289, 2020.

Thanh Le, Ngoc Huynh, and Bac Le. Knowledge graph embedding by projection and rotation on hyperplanes for link prediction. *Applied Intelligence*, 53(9):10340–10364, 2023.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6674–6681, 2019.

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*, pages 1668–1674, 2015.

Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, and C Giles. Recovering concept prerequisite relations from university course dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C Lee Giles. Investigating active learning for concept prerequisite learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yahui Long, Min Wu, Yong Liu, Yuan Fang, Chee Keong Kwoh, Jinmiao Chen, Jiawei Luo, and Xiaoli Li. Pre-training graph neural networks for link prediction in biomedical networks. *Bioinformatics*, 38(8):2254–2262, 2022.

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*, pages 1787–1796, 2023.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

Christopher Morris, Matthias Fey, and Nils M Kriege. The power of the weisfeiler-leman algorithm for machine learning with graphs. *arXiv preprint arXiv:2105.05911*, 2021.

Christopher Morris, Gaurav Rattan, Sandra Kiefer, and Siamak Ravanbakhsh. Speqnets: Sparsity-aware permutation-equivariant graph networks. In *International Conference on Machine Learning*, pages 16017–16042. PMLR, 2022.

Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, 2017.

Sudeshna Roy, Meghana Madhyastha, Sheril Lawrence, and Vaibhav Rajan. Inferring concept prerequisite relations from online educational resources. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9589–9594, 2019.

Guillaume Salha, Stratis Limnios, Romain Hennequin, Viet-Anh Tran, and Michalis Vazirgiannis. Gravity-inspired graph autoencoders for directed link prediction. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 589–598, 2019.

Daqian Shi, Ting Wang, Hao Xing, and Hao Xu. A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. *Knowledge-Based Systems*, 195:105618, 2020.

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)*, pages 118–126. SIAM, 2022.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353, 2019.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, and Chao Chen. Link prediction with persistent homology: An interactive view. In *International conference on machine learning*, pages 11659–11669. PMLR, 2021.