PAGTPRP: POSITION-AWARE GRAPH TRANSFORMER FOR PREREQUISITE RELATIONSHIP PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting prerequisite relationships is crucial in educational applications such as curriculum planning and intelligent tutoring. Although unified modeling of concept and document prerequisite prediction has been achieved, existing approaches still face limitations in integrating global structural context and edge-weight information when modeling heterogeneous prerequisite networks. We propose a **P**osition-Aware **G**raph **T**ransformer for **P**rerequisite **R**elationship **P**rediction (PAGTPRP). Specifically, PAGTPRP firstly generates embeddings for knowledge concepts and documents and constructs a global heterogeneous graph, which contains two types of nodes (documents and knowledge concepts) and three types of edges ("document-document", "concept-concept", and "document-concept"). In addition, we calculate the edge weights and use the graph Transformer to encode them as relational encodings, which are integrated into the multi-head self-attention mechanism. Finally, we employ a Siamese network to more accurately predict the prerequisite relationships. On LectureBank and UCD, we compare PAGTPRP with several baseline methods. The experimental results demonstrate that PAGTPRP achieves excellent performance in predicting prerequisite relationships. Our code is available at https://anonymous.4open.science/r/PAGTPRP-4E44.

1 INTRODUCTION

Prerequisite relations encode hierarchical dependencies among learning objects and are fundamental to learning and knowledge organization (Yu et al., 2024; Wang & Liu, 2016). A learner is generally expected to master prerequisite knowledge before moving to subsequent material; for example, basic arithmetic supports later study of algebra. Beyond an individual candidate pair, related concepts and documents can provide additional structural context for prerequisite prediction. Prerequisite information supports curriculum planning (Limongelli et al., 2015; Yu et al., 2024), reading-list generation (Gordon et al., 2016; Dong et al., 2024), knowledge tracing (Chen et al., 2018; Huang et al., 2024), and cognitive diagnosis (Gao et al., 2023).

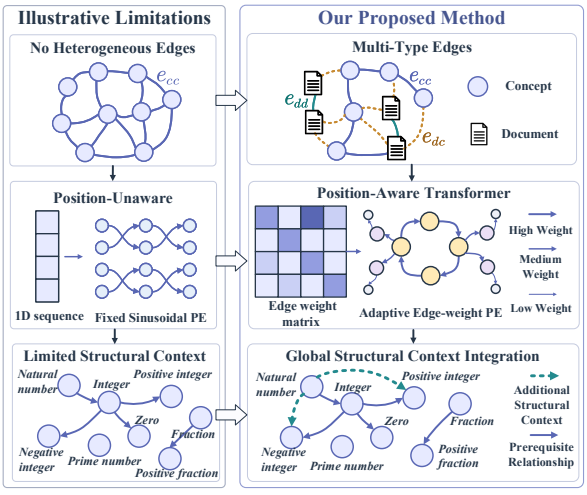


Figure 1: Schematic illustration of limited versus global structural context and edge-weight-aware representation learning. Dashed curves indicate contextual associations, not inferred prerequisite relations.

Existing studies broadly include feature-engineering and deep-learning approaches. Feature-based methods, such as SVM, RF, and RefD (Liang et al., 2018; 2015), exploit manually designed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

semantic or structural cues, but can be limited in capturing complex representations. Graph-based neural methods learn from concepts, documents, and their relations; representative work addresses concept-level prediction (Sun et al., 2024; Zhang et al., 2025a;b) and joint concept/document modeling (Zhang et al., 2025c).

Although heterogeneous modeling and global context have been explored in prerequisite prediction, an important design question is how continuous association strengths should guide node interactions. A related question is how to incorporate context beyond individual candidate pairs. Figure 1 illustrates these design choices through heterogeneous relations, edge-weight-aware attention, and the integration of broader structural context.

To address these questions, we propose a Position-Aware Graph Transformer for Prerequisite Relationship Prediction (PAGTPRP). We construct a weighted heterogeneous graph containing concept and document nodes with document–document, concept–concept, and document–concept edges. The Transformer maps continuous edge weights into relational encodings and adds them to self-attention logits. The resulting node representations combine semantic compatibility with broader relational context. An ordered-pair prediction MLP then uses node pairs to infer directed prerequisite relations at both concept and document levels.

The primary contributions are outlined as follows:

- We propose a new heterogeneous graph transformer model based on the encoding of edge-weight information and construct a global heterogeneous graph to predict concept and document prerequisite relationships.
- We use the graph Transformer to encode continuous edge weights as relational encodings and integrate them into self-attention as additive biases. This allows association strengths to guide node interactions when learning representations of knowledge objects.
- Our approach is assessed alongside existing baseline models using two publicly available datasets, LectureBank and UCD, and the results of the experiments confirm the benefits of our model.

2 Related Work

2.1 Feature Engineering-Based Prerequisite Relationship Extraction

Early prerequisite graphs were largely specified by domain experts, an approach that was difficult to scale (Gasparetti et al., 2015). Later studies automatically inferred prerequisite relations using semantic, structural, and Wikipedia-derived features (Li et al., 2020). Talukdar & Cohen (2012) combined Wikipedia link structure with crowd-sourced annotations, while Sayyadiharikandeh et al. (2019) designed Wikipedia features for binary classification. Other work explored active learning and feature design (Liang et al., 2018), and additional signals for prerequisite classification (Zhou & Xiao, 2019). These studies use manually designed semantic or structural features.

2.2 Deep Learning-Based Prerequisite Relationship Extraction

Deep learning methods increasingly construct concept graphs and learn node representations with neural networks. PREREQ learns concept representations using Pairwise-link LDA and predicts prerequisite relations with a Siamese neural network (Roy et al., 2019). ConLearn combines pre-trained language models with graph/self-attention modeling (Sun et al., 2022). MHAVGAE uses multi-head attention over resource–concept graphs (Zhang et al., 2022), while MTN extracts and fuses multi-view concept features (Wang et al., 2024). EBCPL derives evidence from educational data and performs relation extraction over that evidence (Zhang et al., 2024). LECDPR combines LLM-enhanced semantic representations with concept–document interaction modeling (Xiao et al., 2026). GKROM combines neighborhood graph attention, a global linear transformation, and multi-objective relation prediction (Zhang et al., 2025c). PAGTPRP instead maps continuous association weights into learnable biases added directly to global self-attention logits over concept and document nodes. The distinction lies in how association strengths guide representation learning, while prerequisite direction is learned through ordered-pair prediction.

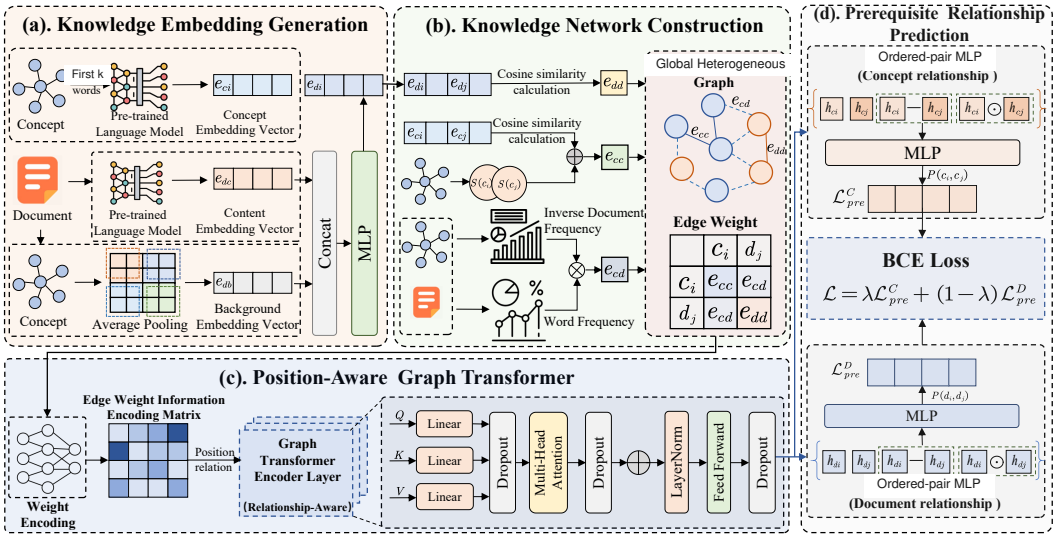


Figure 2: The overall structure of PAGTPRP. In panel (a), e_{dc} and e_{db} denote document content and background embeddings. Panel (d) uses final node representations $\mathbf{h}_o$ from $\mathbf{X}_L$.

3 Problem Statement

We introduce the main notation used throughout the paper. Let $C = \{c_1, c_2, \ldots, c_m\}$ denote the set of knowledge concepts and $D = \{d_1, d_2, \ldots, d_n\}$ the set of learning documents, where each document contains one or more concepts. We construct a concept–document heterogeneous graph $G = (V, E)$ with $V = C \cup D$ and three edge types. **Document–document** edges e_{dd} connect semantically related documents and enrich document representations. **Concept–concept** edges e_{cc} represent semantic relatedness between concepts. **Document–concept** edges e_{dc} measure the relevance of a concept to a document.

Research Definition: Given documents and their associated concepts, our goal is to learn $\mathcal{F}_\theta : (C \times C) \cup (D \times D) \to \{0, 1\}$ to predict whether o_i is a prerequisite of o_j for a concept or document pair. Concept–document edges provide contextual associations, not prediction targets. Specifically, $\mathcal{F}_\theta(o_i, o_j) = 1$ indicates that o_i is a prerequisite of o_j, and 0 otherwise.

4 Methodology

PAGTPRP contains four components (Figure 2): knowledge embedding generation, heterogeneous knowledge-network construction, a position-aware graph Transformer, and ordered-pair prerequisite prediction. The first builds semantic concept/document representations; the second constructs three weighted edge types; the third injects edge-weight-derived relational encodings into self-attention; and the final component predicts ordered prerequisite pairs.

4.1 Knowledge Embedding Generation

For concepts and documents, we first obtain semantic embeddings that serve as node features for the heterogeneous graph. For a knowledge concept $c_i \in C$, we use the first k words of its Wikipedia article as the textual description and encode it with a pretrained language model:

$$\mathbf{e}_{c_i} = f_{pre}(c_i), \tag{1}$$

where $\mathbf{e}_{c_i} \in \mathbb{R}^d$, d is the shared node embedding dimension, and $f_{\text{pre}}(\cdot)$ denotes the pretrained encoder.

For a document $d_i \in D$, we use both a content embedding and a background embedding. The content embedding is directly obtained from the pretrained model,

$$\mathbf{e}_{d_i}^{\text{content}} = f_{pre}(d_i), \tag{2}$$

while the background embedding summarizes its associated concepts. Let $C(d_i) \subseteq C$ be the nonempty concept set of document d_i; we average their embeddings:

$$\mathbf{e}_{d_i}^{\mathrm{bg}} = \frac{1}{|C(d_i)|} \sum_{c \in C(d_i)} \mathbf{e}_c. \tag{3}$$

The two views are complementary: $\mathbf{e}_{d_i}^{\mathrm{content}}$ captures the text, whereas $\mathbf{e}_{d_i}^{\mathrm{bg}}$ captures concept-level background. We concatenate them and apply an MLP to obtain the final document representation,

$$\mathbf{e}_{d_i} = f([\mathbf{e}_{d_i}^{\mathrm{content}}; \mathbf{e}_{d_i}^{\mathrm{bg}}]), \tag{4}$$

where $[\cdot;\cdot]$ denotes concatenation and $f(\cdot)$ is a nonlinear transformation. The MLP projects document features to the shared d-dimensional node space, so $\mathbf{e}_{c_i}, \mathbf{e}_{d_i} \in \mathbb{R}^d$. For concepts without Wikipedia pages, GPT-4-selected related entries supply only the textual descriptions; the original node identities and pair labels are retained. The selected entries are fixed before training and evaluation. Counts are given in Appendix A, with the prompt and details in Appendix C.1.

4.2 Knowledge Network Construction

In traditional representation learning approaches, concepts and documents are often treated as unrelated entities. To integrate heterogeneous information, we construct a global heterogeneous graph containing document and concept nodes with document–document, concept–concept, and document–concept edges. The edge weights characterize semantic, structural, or relevance-based associations rather than directly encoding prerequisite direction, consistent with prior prerequisite-learning studies using similarity- or relevance-based signals for representation learning (Liang et al., 2018; Li et al., 2021; Zhang et al., 2025c).

For the "document-document" edges, we calculate the edge weights by computing the cosine similarity (Liang et al., 2015) between the embeddings of two documents. This cosine score characterizes similarity in embedding space and provides contextual associations between documents. The weight for "document-document" edges is given by this formula:

$$A_{d_i,d_j} = \cos(\mathbf{e}_{d_i}, \mathbf{e}_{d_j}). \tag{5}$$

Rather than relying on a single perspective, we consider both how close concepts are in semantic space and how much overlap exists between their associated sets. For content similarity, we use cosine similarity between concept embeddings. Specifically, we employ BERT-base-uncased to encode the textual descriptions of concepts, producing 768-dimensional representations. The resulting concept embeddings are then used to compute cosine similarity; a higher score indicates greater semantic relatedness between the two concepts. For the latter, inspired by Liang et al. (2015), "a concept can be represented by the set of all related concepts in the concept space," we calculate the Jaccard similarity (Liang et al., 2015). Here, following Liang et al. (2015), related concepts are the other Wikipedia concepts referenced in its entry. If no reference set is available, we use $S(c_i) = \varnothing$. Therefore, for a pair of concepts (c_i, c_j), let $S(c_i)$ and $S(c_j)$ represent the sets of related concepts for the two concepts, respectively. The weight for "concept-concept" edges is given by:

$$A_{c_i,c_j} = \beta \cdot \frac{|S(c_i) \cap S(c_j)|}{|S(c_i) \cup S(c_j)|} + (1-\beta) \cdot \cos(\mathbf{e}_{c_i}, \mathbf{e}_{c_j}), \tag{6}$$

where $\beta \in [0,1]$ balances the two similarities; the Jaccard term is defined as zero when the union is empty. $|S(c_i) \cap S(c_j)|$ is the number of external concepts shared by both concepts; it refers to the total number of third-party concepts linked from the Wikipedia pages of both concepts. The quantity $|S(c_i) \cup S(c_j)|$ is the total number of external concepts related to at least one of the two concepts.

We use TF-IDF to emphasize concepts frequent within a document but relatively rare across the corpus. It measures the importance of a term to a document within a collection by taking into account both the term's frequency in the document and its distribution across the entire corpus. The weight A_{d_i,c_j} for the "document-concept" edges is given by:

$$A_{d_i,c_j} = \mathrm{TF}(c_j, d_i) \times \mathrm{IDF}(c_j), \tag{7}$$

where $\mathrm{TF}(c_j, d_i)$ is the term frequency of concept c_j in document d_i, calculated as:

$$\mathrm{TF}(c_j, d_i) = \frac{f_{c_j,d_i}}{N_{d_i}}, \tag{8}$$

where f_{c_j,d_i} is the number of occurrences of concept c_j in document d_i, and N_{d_i} is the total number of words in document d_i. The inverse document frequency $\text{IDF}(c_j)$ for concept c_j is calculated as:

$$\text{IDF}(c_j) = \log\left(\frac{N}{N_{c_j}}\right), \tag{9}$$

where N is the total number of documents in the corpus, and N_{c_j} is the number of documents containing concept c_j.

4.3 Position-Aware Graph Transformer

In PAGTPRP, position awareness denotes pairwise relational information derived from continuous edge weights, rather than absolute sequence positions or shortest-path coordinates. Each weight A_{ij} is mapped to a learnable relational encoding $B_{ij}^{(l)}$, which serves as an additive bias in self-attention.

We first stack the aligned document and concept embeddings to construct the node embedding matrix $\mathbf{X}_0 \in \mathbb{R}^{(n+m)\times d}$, where n and m denote the numbers of documents and concepts, respectively, and d is their shared node embedding dimension. At the same time, we construct the corresponding edge-weight matrix $\mathbf{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ using the weights defined above, with $A_{c_j,d_i} = A_{d_i,c_j}$. Within-type entries, including the diagonal, follow Eqs. (5)–(6); document–concept entries with zero term frequency are zero.

We set the Transformer model to contain L blocks; then the multi-head self-attention mechanism is employed to calculate $\mathbf{Q}_{l,k}$, $\mathbf{K}_{l,k}$, and $\mathbf{V}_{l,k}$. Let $\mathbf{X}_{l-1}$ denote the node representation matrix from the $(l-1)$-th layer, which serves as the input to the l-th Transformer layer. Then, $\mathbf{Q}_{l,k}$, $\mathbf{K}_{l,k}$, and $\mathbf{V}_{l,k}$ for the k-th attention head at the l-th layer are given by:

$$\mathbf{Q}_{l,k} = \mathbf{X}_{l-1}\mathbf{W}_{l,k}^{Q}, \quad \mathbf{K}_{l,k} = \mathbf{X}_{l-1}\mathbf{W}_{l,k}^{K}, \quad \mathbf{V}_{l,k} = \mathbf{X}_{l-1}\mathbf{W}_{l,k}^{V}. \tag{10}$$

Here, $\mathbf{W}_{l,k}^{Q}, \mathbf{W}_{l,k}^{K}, \mathbf{W}_{l,k}^{V} \in \mathbb{R}^{d\times d_k}$ are learnable parameter matrices.

For each node pair (o_i, o_j), we map its association weight A_{ij} to a learnable relational encoding. Collecting these encodings yields the attention bias matrix $\mathbf{B}^{(l)} \in \mathbb{R}^{(n+m)\times(n+m)}$, defined entry-wise as:

$$B_{ij}^{(l)} = \text{ReLU}(w_p^{(l)} A_{ij} + b_p^{(l)}), \tag{11}$$

where $w_p^{(l)}$ and $b_p^{(l)}$ are scalar trainable parameters used at layer l, shared across its attention heads. Adding $\mathbf{B}^{(l)}$ to the attention logits gives the output of head k at layer l:

$$\mathbf{X}_{l,k} = \text{Attention}^{(l)}(\mathbf{Q}_{l,k}, \mathbf{K}_{l,k}, \mathbf{V}_{l,k}) = \text{softmax}\left(\frac{\mathbf{Q}_{l,k}\mathbf{K}_{l,k}^{\top}}{\sqrt{d_k}} + \mathbf{B}^{(l)}\right)\mathbf{V}_{l,k}. \tag{12}$$

Here, $d_k = \frac{d}{H}$ represents the dimension of each attention head, and H represents the number of attention heads. Then, the outputs of all attention heads are concatenated. The process can be represented as:

$$\mathbf{X}_l^c = \text{Concat}(\mathbf{X}_{l,1}; \mathbf{X}_{l,2}; \cdots; \mathbf{X}_{l,H}). \tag{13}$$

Subsequently, to stabilize the training process, we apply layer normalization and add residual connections:

$$\widetilde{\mathbf{X}}_l = \text{LayerNorm}\left(\mathbf{X}_{l-1} + \mathbf{X}_l^c\right). \tag{14}$$

Finally, the feedforward network further processes features to obtain the final output of the l-th layer:

$$\mathbf{X}_l = \Phi\left(\widetilde{\mathbf{X}}_l\mathbf{W}_1 + \mathbf{b}_1\right)\mathbf{W}_2 + \mathbf{b}_2. \tag{15}$$

Here, $\Phi(\cdot)$ represents the GELU activation function; $\mathbf{W}_1$ and $\mathbf{W}_2$ are trainable parameters, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias terms. The resulting output $\mathbf{X}_l$ retains the graph-structural information, thereby preparing the representation for the next layer or the final prediction.

Ultimately, $\mathbf{X}_L \in \mathbb{R}^{(n+m)\times d}$ contains the final node representations, integrating global graph structure, neighborhood semantics, and edge-weight-derived relational information for prerequisite prediction.

4.4 PREREQUISITE RELATIONSHIP PREDICTION

We predict prerequisite relations over ordered node pairs with an ordered-pair MLP. Let $\mathbf{h}_o$ be the row of $\mathbf{X}_L$ corresponding to node o. For a concept pair (c_i, c_j), where c_i is the candidate prerequisite of c_j, the logit is

$$\text{logit}_{ij} = \text{MLP}([\mathbf{h}_{c_i}; \mathbf{h}_{c_j}; \mathbf{h}_{c_i} - \mathbf{h}_{c_j}; \mathbf{h}_{c_i} \odot \mathbf{h}_{c_j}]), \tag{16}$$

where $\odot$ is the Hadamard product; the same pair-feature construction applies to document pairs using $\mathbf{h}_{d_i}$ and $\mathbf{h}_{d_j}$. Reversing the pair changes both concatenation order and the signed difference, so logit_{ij} and logit_{ji} can differ; graph weights model association strength, while prerequisite direction is learned at this ordered-pair stage, as in GKROM (Zhang et al., 2025c). The predicted probability is

$$P(c_i, c_j) = \sigma(\text{logit}_{ij}), \tag{17}$$

where σ is the sigmoid function. Let $\mathcal{P}_C \subseteq C \times C$ and $\mathcal{P}_D \subseteq D \times D$ be the labeled ordered-pair sets, $N_C = |\mathcal{P}_C|$, and $N_D = |\mathcal{P}_D|$. Each $y_{ij} \in \{0, 1\}$ is the ground-truth label. The concept-level binary cross-entropy loss is

$$\begin{aligned}\mathcal{L}^C_{\text{preq}} = -\frac{1}{N_C} \sum_{(c_i,c_j)\in\mathcal{P}_C} \Big[& y_{ij} \log\left(P(c_i, c_j)\right) \\ & + (1 - y_{ij}) \log\left(1 - P(c_i, c_j)\right)\Big].\end{aligned} \tag{18}$$

The document-level loss is defined analogously:

$$\begin{aligned}\mathcal{L}^D_{\text{preq}} = -\frac{1}{N_D} \sum_{(d_i,d_j)\in\mathcal{P}_D} \big[& y_{ij} \log\left(P(d_i, d_j)\right) \\ & + (1 - y_{ij}) \log\left(1 - P(d_i, d_j)\right)\big],\end{aligned} \tag{19}$$

The joint objective is

$$\mathcal{L} = \lambda \mathcal{L}^C_{\text{preq}} + (1 - \lambda)\mathcal{L}^D_{\text{preq}}, \tag{20}$$

where $\lambda \in [0, 1]$ balances the two tasks. The complete procedure is summarized in Algorithm 1 in Appendix B.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets. We evaluate PAGTPRP on two public benchmarks. UCD (Liang et al., 2017) collects computer-science course information from 377 U.S. universities. LectureBank (Li et al., 2019) contains undergraduate and graduate computer-science educational resources collected from multiple online platforms. Table 1 summarizes the dataset statistics; further descriptions are provided in Appendix C.2.

Table 1: Dataset statistics. $|C|/|D|$ denote the numbers of concepts/documents; $|C^+_{pre}|/|C^-_{pre}|$ denote positive/negative concept pairs.

Dataset	$\lvert C\rvert$	$\lvert D\rvert$	$\lvert C^+_{pre}\rvert$	$\lvert C^-_{pre}\rvert$
UCD	407	654	1,007	1,007
LectureBank	246	277	601	601

Baseline Methods. We compare with PREREQ, HGT, ConLearn, MHAVGAE, SeHGNN, HGAPNet, LCPRE, MTN, GKROM, and LECDPR. All baselines are rerun on our data splits; no GKROM-reported scores are reused. References and descriptions are provided in Appendix C.3.

Evaluation Metrics. We report ACC, F1, and AUC for concept- and document-level prerequisite prediction. Table 2 reports means and sample standard deviations over five random seeds; Tables 3 and 4 and Figure 3 report single-run results.

Implementation Details. We independently repartition each dataset 8:1:1 into train/validation/test sets and use identical splits for PAGTPRP and all baselines. During training, test nodes and their incident edges are excluded from graph construction and message passing, so the test graph remains unseen and does not contribute to representation learning. Negatives are constructed from

reversed positive pairs and randomly sampled non-positive pairs (Sun et al., 2024); the former provides direction-sensitive negatives and the latter covers general non-prerequisite relations. PAGTPRP is trained for 30 epochs with Adam, a learning rate of 10^{-4}, a batch size of 32, two Transformer layers, and a weight decay of 6×10^{-5}. The loss weight λ is 0.6 and 0.65 on UCD and LectureBank, respectively. BERT is used to initialize node representations, and all experiments are implemented in PyTorch on an RTX 4090D GPU. Additional preprocessing details, including the fixed replacement strategy for concepts without Wikipedia pages, are provided in Appendix C.1.

Table 2: Concept prerequisite prediction on LectureBank and UCD. Results are mean ± sample standard deviation (%) over five random seeds. The best and second-best means are bold and underlined, respectively. Gains over the best baseline are in percentage points (pp).

Method	LectureBank			UCD		
	ACC	AUC	F1	ACC	AUC	F1
PREREQ	45.95 ± 5.08	43.39 ± 5.25	52.04 ± 6.60	49.90 ± 2.72	48.42 ± 2.70	60.24 ± 5.05
HGT	76.20 ± 0.37	87.26 ± 0.19	79.31 ± 0.44	79.31 ± 1.01	86.20 ± 0.52	79.20 ± 0.94
ConLearn	79.01 ± 1.50	87.02 ± 0.28	78.19 ± 1.89	81.39 ± 1.24	86.98 ± 0.71	79.92 ± 1.30
MHAVGAE	75.04 ± 0.37	84.76 ± 0.61	75.83 ± 0.56	71.88 ± 1.54	79.55 ± 0.61	70.72 ± 2.28
SeHGNN	80.00 ± 0.69	86.05 ± 0.73	80.71 ± 0.36	80.59 ± 3.08	86.07 ± 2.30	80.95 ± 2.37
HGAPNet	81.32 ± 1.25	88.42 ± 0.44	81.51 ± 1.16	81.39 ± 0.57	90.06 ± 0.12	80.08 ± 0.68
LCPRE	78.02 ± 1.99	83.83 ± 1.68	77.32 ± 2.20	80.50 ± 1.43	85.76 ± 1.79	80.02 ± 1.56
MTN	80.50 ± 2.38	87.20 ± 2.69	80.68 ± 3.00	81.19 ± 2.16	89.46 ± 1.37	80.02 ± 2.71
GKROM	81.65 ± 1.08	88.59 ± 0.68	81.71 ± 1.02	81.19 ± 1.05	89.62 ± 0.41	79.70 ± 1.19
LECDPR	$\underline{84.30} \pm 1.66$	$\underline{91.87} \pm 0.92$	$\underline{84.72} \pm 1.41$	$\underline{87.03} \pm 2.14$	$\underline{92.62} \pm 0.80$	$\underline{86.58} \pm 2.47$
PAGTPRP (Ours)	$\mathbf{85.46} \pm 3.90$	$\mathbf{93.95} \pm 2.71$	$\mathbf{85.72} \pm 4.02$	$\mathbf{87.33} \pm 1.42$	$\mathbf{94.77} \pm 1.07$	$\mathbf{87.07} \pm 1.46$
Gain (pp)	1.16↑	2.08↑	1.00↑	0.30↑	2.15↑	0.49↑

5.2 Overall Performance

Table 2 reports concept prerequisite prediction over five random seeds. PAGTPRP achieves the highest mean ACC, AUC, and F1 on both LectureBank and UCD. LECDPR is the strongest baseline across all six comparisons. Relative to LECDPR, PAGTPRP improves ACC, AUC, and F1 by 1.16, 2.08, and 1.00 percentage points on LectureBank, and by 0.30, 2.15, and 0.49 percentage points on UCD, respectively. The largest gains on both datasets are in AUC, indicating improved ranking of prerequisite versus non-prerequisite pairs.

These results support the effectiveness of combining heterogeneous relational context with edge-weight-aware global attention. However, the ACC and F1 gains over LECDPR are modest, and PAGTPRP exhibits greater variability on LectureBank. We therefore interpret the differences as improvements in mean performance, without claiming statistical significance. Section 5.5 examines the contributions of individual components through ablation experiments.

Table 3: Document prerequisite prediction (single run). Best and second-best scores are bold and underlined. Gains in percentage points (pp) are computed from the displayed scores.

Method	LectureBank			UCD		
	ACC	AUC	F1	ACC	AUC	F1
DPR-ED	0.545	0.536	0.566	0.574	0.560	0.616
EPR-VL	0.590	0.587	0.573	0.590	0.608	0.591
GKROM	<u>0.695</u>	<u>0.750</u>	<u>0.683</u>	<u>0.695</u>	<u>0.756</u>	<u>0.632</u>
PAGTPRP(Ours)	**0.822**	**0.864**	**0.823**	**0.746**	**0.785**	**0.775**
Gain	12.7	11.4	14.0	5.1	2.9	14.3

5.3 Document Prerequisite Prediction

Document prerequisite prediction is important for organizing instructional materials into a coherent and progressive learning sequence. Because a document can contain multiple concepts, document-level dependencies provide complementary supervision to concept-level prerequisite learning. We therefore compare PAGTPRP with EPR-VL (Xiao et al., 2022), DPR-ED (Gasparetti, 2022), and GKROM (Zhang et al., 2025c) on document–document prerequisite prediction.

Table 3 shows that PAGTPRP consistently outperforms the document-level baselines on LectureBank and UCD. The improvements indicate that document representations benefit from jointly incorporating document–document dependencies, document–concept associations, and concept–concept structure rather than modeling document pairs in isolation. This supports the effectiveness of the unified heterogeneous graph for document prerequisite prediction.

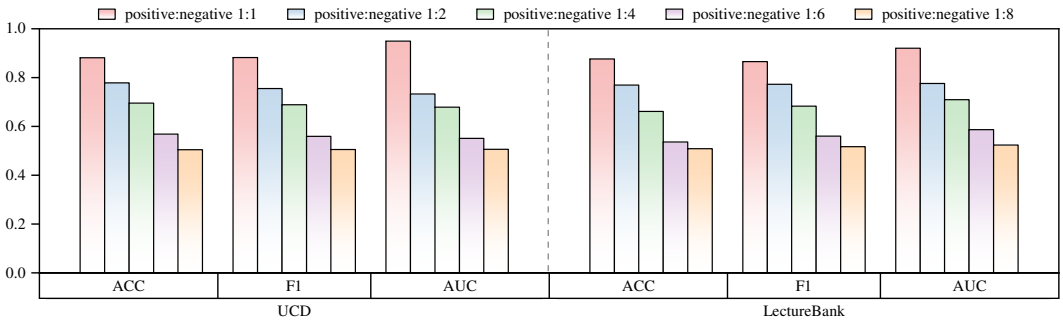


Figure 3: Prerequisite relation prediction in sparse scenarios on UCD and LectureBank (single run).

5.4 Prerequisite Relation Prediction in Sparse Scenarios

Prerequisite annotations are often sparse in educational data, so we evaluate PAGTPRP under different prerequisite densities. The normal setting uses a 1:1 positive-to-negative ratio; negatives are obtained from reversed positive pairs or randomly sampled unrelated concept pairs. To simulate increasingly sparse supervision, the hard settings use ratios of 1:2, 1:4, 1:6, and 1:8. One negative is the reversed positive pair and the remaining negatives are generated by replacing one concept in a positive pair.

Figure 3 shows that performance decreases as the number of negatives grows and available prerequisite supervision becomes sparser. Nevertheless, PAGTPRP maintains comparatively strong predictive performance because the global heterogeneous graph can aggregate complementary signals from structurally dispersed concepts, providing additional context when direct prerequisite annotations are limited.

Table 4: Concept prerequisite prediction in the ablation study (single run).

Method	LectureBank			UCD		
	ACC	AUC	F1	ACC	AUC	F1
Ours	**0.867**	**0.931**	**0.878**	**0.886**	**0.946**	**0.885**
Ours w/o BE	0.843	0.927	0.863	0.851	0.938	0.848
Ours w/o PE	0.826	0.905	0.844	0.816	0.914	0.817
Ours w/o BEPE	0.801	0.909	0.841	0.801	0.919	0.819
Ours w/o DP	0.836	0.901	0.868	0.856	0.933	0.859

5.5 Ablation Experiment

To evaluate the effectiveness of each module in PAGTPRP, we designed four ablation experiments: (1) Removing the document background embedding (BE). (2) Removing the position encoding (PE).

(3) Simultaneously removing the document background embedding and position encoding (BEPE). (4) Removing the document prerequisite relationships (DP).

Table 4 shows that every ablated component contributes to performance. Removing BE consistently degrades results, confirming the value of document background information. This concept-level summary is intended to reduce dependence on incidental wording in the document text and emphasize concepts central to its subject matter. Removing PE produces a larger drop, highlighting the importance of edge-weight-aware relational encoding. Removing BE and PE together yields the lowest ACC on both datasets, while removing DP also hurts the metrics, indicating that document relations provide useful complementary information for concept prerequisite prediction.

Specifically, removing PE lowers ACC by 4.1 and 7.0 percentage points on LectureBank and UCD, respectively, whereas removing BE lowers it by 2.4 and 3.5 points. Removing DP lowers ACC by 3.1 and 3.0 points, respectively. These comparisons support the contributions of relational encoding and document-level supervision within the single-run setting of this ablation.

The dot-product term in Eq. (12) measures compatibility between learned node features, whereas $\mathbf{B}^{(l)}$ supplies pairwise association information. Thus, similar feature compatibility can yield different attention weights when the relational encodings differ. The bias guides information aggregation across the graph; it is not itself a prerequisite probability and does not determine prerequisite direction. BE and PE consequently operate at different stages: BE enriches the initial document features, whereas PE modulates subsequent attention-based aggregation.

5.6 Hyperparameter Analysis

As shown in Figure 4, we evaluated the model's performance on the UCD dataset under different hyperparameter settings (attention heads, loss weight, transformer layers, and learning rate).

The best configuration uses 8 attention heads, a loss weight of $\lambda = 0.6$, 2 Transformer layers, and a learning rate of 10^{-4}. Too few heads restrict the representation space, whereas too many add redundancy and computation. The loss weight of 0.6 provides the best balance between objectives, and two layers are sufficient to capture the relevant dependencies without unnecessary depth. The learning-rate results similarly show that overly large updates destabilize optimization, while smaller rates slow convergence.

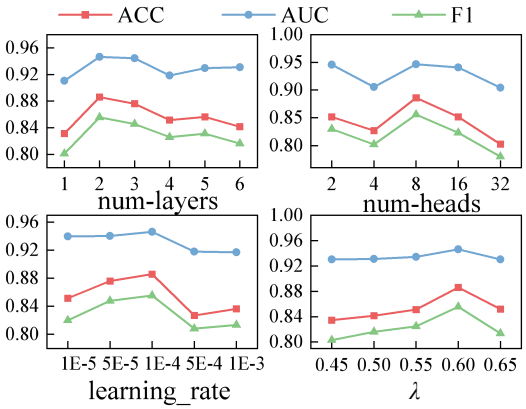


Figure 4: Hyperparameter analysis of PAGTPRP.

6 Conclusion

We presented PAGTPRP for joint concept- and document-level prerequisite prediction on a weighted heterogeneous graph. The model transforms continuous edge weights into relational encodings that guide global self-attention alongside semantic compatibility. Ordered-pair prediction learns prerequisite direction from the resulting node representations. Experiments on LectureBank and UCD support the effectiveness of the framework, while ablation studies examine the contributions of its components.

GenAI Usage Disclosure

To enhance the manuscript's clarity and style, we used a generative language model (GPT) to assist with grammar correction, academic expression, and sentence fluency, based on the authors' original draft. We also used GPT-4 during data preprocessing to identify semantically close substitutes with existing Wikipedia pages for seven concepts in LectureBank and ten in UCD that lacked corresponding entries; these substitutions were fixed before training and evaluation. However, all core ideas, including the main (technical) contributions, method design, experimental setup, and analysis, were entirely conceived and developed by the authors. All figures, including method diagrams and experimental results, were created solely by the authors without any assistance from generative models.

REFERENCES

Penghe Chen, Yu Lu, Vincent W. Zheng, and Yang Pian. Prerequisite-driven deep knowledge tracing. In *IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018*, pp. 39–48. IEEE Computer Society, 2018.

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.

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 *SIGIR*, pp. 983–992. ACM, 2023.

Fabio Gasparetti. Discovering prerequisite relations from educational documents through word embeddings. *Future Gener. Comput. Syst.*, 127:31–41, 2022.

Fabio Gasparetti, Carla Limongelli, and Filippo Sciarrone. Exploiting wikipedia for discovering prerequisite relationships among learning objects. In *2015 International Conference on Information Technology Based Higher Education and Training, ITHET 2015, Lisbon, Portugal, June 11-13, 2015*, pp. 1–6. IEEE, 2015.

Jonathan Gordon, Linhong Zhu, Aram Galstyan, Prem Natarajan, and Gully Burns. Modeling concept dependencies in a scientific corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (eds.), *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pp. 2704–2710. ACM / IW3C2, 2020.

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 Jianfei Cai, Mohan S. Kankanhalli, Balakrishnan Prabhakaran, Susanne Boll, Ramanathan Subramanian, Liang Zheng, Vivek K. Singh, Pablo César, Lexing Xie, and Dong Xu (eds.), *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pp. 3151–3159. ACM, 2024.

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, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 6674–6681. AAAI Press, 2019.

Irene Li, Alexander R. Fabbri, Swapnil Hingmire, and Dragomir R. Radev. R-VGAE: relational-variational graph autoencoder for unsupervised prerequisite chain learning. In Donia Scott and Núria Bel (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 1147–1157. International Committee on Computational Linguistics, 2020.

Irene Li, Vanessa Yan, Tianxiao Li, Rihao Qu, and Dragomir R. Radev. Unsupervised cross-domain prerequisite chain learning using variational graph autoencoders. In *ACL/IJCNLP (2)*, pp. 1005–1011. Association for Computational Linguistics, 2021.

Chen Liang, Zhaohui Wu, Wenyi Huang, and C. Lee Giles. Measuring prerequisite relations among concepts. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1668–1674. The Association for Computational Linguistics, 2015.

Chen Liang, Jianbo Ye, Zhaohui Wu, Bart Pursel, and C. Lee Giles. Recovering concept prerequisite relations from university course dependencies. In *AAAI*, pp. 4786–4791. AAAI Press, 2017.

Chen Liang, Jianbo Ye, Shuting Wang, Bart Pursel, and C. Lee Giles. Investigating active learning for concept prerequisite learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 7913–7919. AAAI Press, 2018.

Carla Limongelli, Fabio Gasparetti, and Filippo Sciarrone. Wiki course builder: A system for retrieving and sequencing didactic materials from wikipedia. In *2015 International Conference on Information Technology Based Higher Education and Training, Lisbon, Portugal, June 11-13, 2015*, pp. 1–6. IEEE, 2015.

Debjani Mazumder, Jiaul H. Paik, and Anupam Basu. A graph neural network model for concept prerequisite relation extraction. In Ingo Frommholz, Frank Hopfgartner, and Rodrygo L. T. Santos (eds.), *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, United Kingdom, October 21-25, 2023*, pp. 1787–1796. ACM, 2023.

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, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019,Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 9589–9594. AAAI Press, 2019.

Mohsen Sayyadiharikandeh, Jonathan Gordon, José Luis Ambite, and Kristina Lerman. Finding prerequisite relations using the wikipedia clickstream. In Sihem Amer-Yahia and Mohammad Mahdian (eds.), *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pp. 1240–1247. ACM, 2019.

Hao Sun, Yuntao Li, and Yan Zhang. Conlearn: Contextual-knowledge-aware concept prerequisite relation learning with graph neural network. In Arindam Banerjee and Zhi-Hua Zhou (eds.), *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM 2022, Alexandria, VA, USA, April 28-30, 2022*, pp. 118–126. SIAM, 2022.

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 Edoardo Serra and Francesca Spezzano (eds.), *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, pp. 2168–2177. ACM, 2024.

Partha P. Talukdar and William W. Cohen. Crowdsourced comprehension: Predicting prerequisite structure in wikipedia. In Joel R. Tetreault, Jill Burstein, and Claudia Leacock (eds.), *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*, pp. 307–315. The Association for Computer Linguistics, 2012.

Shuting Wang and Lei Liu. Prerequisite concept maps extraction for automaticassessment. In Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao (eds.), *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pp. 519–521. ACM, 2016.

Zhichun Wang, Yifeng Shao, Boci Peng, Bangui Li, Yun Li, Qianren Wang, and Nijun Li. Multi-view transformer-based network for prerequisite learning in concept graphs. In Gianluca Demartini, Katja Hose, Maribel Acosta, Matteo Palmonari, Gong Cheng, Hala Skaf-Molli, Nicolas Ferranti, Daniel Hernández, and Aidan Hogan (eds.), *The Semantic Web - ISWC 2024 - 23rd International Semantic Web Conference, Baltimore, MD, USA, November 11-15, 2024, Proceedings, Part I*, volume 15231 of *Lecture Notes in Computer Science*, pp. 67–86. Springer, 2024.

Kui Xiao, Youheng Bai, and Yan Zhang. Extracting precedence relations between video lectures in moocs. In *ICMR*, pp. 608–614. ACM, 2022.

Kui Xiao, Lele Zheng, Xiaoxue He, Miao Zhang, Zhifang Huang, and Yan Zhang. LECDPR: LLM enhancement and concept-document interactive modeling for prerequisite relation prediction. In

Diego Calvanese (ed.), *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence, IJCAI-26*, pp. 5092–5100. International Joint Conferences on Artificial Intelligence Organization, 8 2026. doi: 10.24963/ijcai.2026/567. URL https://doi.org/10.24963/ijcai.2026/567. Main Track.

Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, and Dongrui Fan. Simple and efficient heterogeneous graph neural network. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 10816–10824. AAAI Press, 2023.

Shenbao Yu, Yifeng Zeng, Fan Yang, and Yinghui Pan. Causal-driven skill prerequisite structure discovery. In *AAAI*, pp. 20604–20612. AAAI Press, 2024.

JiaYu Zhang, XiaoYan Zhang, XiaoFeng Du, and TianBo Lu. EBCPL: A novel evidence-based method for concept prerequisite relation learning. In Makoto Onizuka, Jae-Gil Lee, and Kejing Lu (eds.), *Database Systems for Advanced Applications - 29th International Conference, DASFAA 2024, Gifu, Japan, July 2-5, 2024, Proceedings, Part II*, volume 14851 of *Lecture Notes in Computer Science*, pp. 390–405. Springer, 2024.

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 K. Selcuk Candan, Huan Liu, and Leman Akoglu (eds.), *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pp. 1377–1385. ACM, 2022.

Miao Zhang, Jiawei Wang, Jinying Han, Kui Xiao, Zhifei Li, Yan Zhang, Hao Chen, and Shihui Wang. DGCPL: dual graph distillation for concept prerequisite relation learning. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 8366–8374. ijcai.org, 2025a.

Miao Zhang, Jiawei Wang, Kui Xiao, Zhifang Huang, Zhifei Li, and Yan Zhang. Enhancing weak supervision for concept prerequisite relation learning. *IEEE Trans. Big Data*, 11(5):2643–2656, 2025b.

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 Toby Walsh, Julie Shah, and Zico Kolter (eds.), *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pp. 1638–1646. AAAI Press, 2025c.

Yang Zhou and Kui Xiao. Extracting prerequisite relations among concepts in wikipedia. In *International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14-19, 2019*, pp. 1–8. IEEE, 2019.

A Embedding Generation Details

A small number of concepts have no corresponding Wikipedia page (seven in LectureBank and ten in UCD). For these cases, GPT-4 selects a semantically related Wikipedia entry to supply substitute text during preprocessing. The original concept nodes and their prerequisite labels remain unchanged; selected substitute entries are fixed across training and evaluation. The exact prompt and additional implementation details are provided in Appendix C.1.

B Complete PAGTPRP Algorithm

Algorithm 1 summarizes Section 4; J denotes the Jaccard term in Eq. (6).

Algorithm 1 Position-Aware Graph Transformer for Prerequisite Prediction

Input: C, D, Wikipedia entries, encoder BERT, labeled pairs $\mathcal{P}_C, \mathcal{P}_D$
Output: Prerequisite predictions for concept/document pairs and loss $\mathcal{L}$

Embeddings:
$\mathbf{e}_{c_i} \leftarrow f_{\text{pre}}(c_i)$ # Eq.(1)
$\mathbf{e}_{d_i}^{\text{content}} \leftarrow f_{\text{pre}}(d_i), \quad \mathbf{e}_{d_i}^{\text{bg}} \leftarrow \text{avg}_{c \in C(d_i)}(\mathbf{e}_c), \quad \mathbf{e}_{d_i} \leftarrow \text{MLP}([\mathbf{e}_{d_i}^{\text{content}}; \mathbf{e}_{d_i}^{\text{bg}}])$ # Eqs.(2-4)

Graph construction:
Document-document: $A_{d_i d_j} \leftarrow \cos(\mathbf{e}_{d_i}, \mathbf{e}_{d_j})$ # Eq.(5)
Concept-concept: $A_{c_i c_j} \leftarrow \beta \cdot J(S(c_i), S(c_j)) + (1-\beta)\cos(\mathbf{e}_{c_i}, \mathbf{e}_{c_j})$ # Eq.(6)
Document-concept: $A_{d_i c_j} \leftarrow$ TF-IDF # Eqs.(7-9)
Embedding matrix $\mathbf{X}_0$, adjacency $\mathbf{A}$

Position-aware Transformer:
for $l = 1, \ldots, L$ **do**
  $B_{ij}^{(l)} \leftarrow \text{ReLU}(w_p^{(l)} A_{ij} + b_p^{(l)})$ # Eq.(11)
  for $k = 1, \ldots, H$ **do**
    $\mathbf{Q}_{l,k}, \mathbf{K}_{l,k}, \mathbf{V}_{l,k} \leftarrow \mathbf{X}_{l-1}\mathbf{W}_{l,k}^Q, \mathbf{X}_{l-1}\mathbf{W}_{l,k}^K, \mathbf{X}_{l-1}\mathbf{W}_{l,k}^V$ # Eq.(10)
    $\mathbf{X}_{l,k} \leftarrow \text{softmax}(\mathbf{Q}_{l,k}\mathbf{K}_{l,k}^\top / \sqrt{d_k} + \mathbf{B}^{(l)})\mathbf{V}_{l,k}$ # Eq.(12)
  end for
  $\mathbf{X}_l^c \leftarrow \text{Concat}(\mathbf{X}_{l,1}; \ldots; \mathbf{X}_{l,H})$ # Eq.(13)
  $\widetilde{\mathbf{X}}_l \leftarrow \text{LayerNorm}(\mathbf{X}_{l-1} + \mathbf{X}_l^c)$ # Eq.(14)
  $\mathbf{X}_l \leftarrow \text{GELU}(\widetilde{\mathbf{X}}_l \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$ # Eq.(15)
end for

Prediction: Let $\mathbf{h}_o$ be the row of $\mathbf{X}_L$ corresponding to node o.
For concept pair: logit $\leftarrow \text{MLP}([\mathbf{h}_{c_i}; \mathbf{h}_{c_j}; \mathbf{h}_{c_i} - \mathbf{h}_{c_j}; \mathbf{h}_{c_i} \odot \mathbf{h}_{c_j}])$ # Eq.(16)
$P \leftarrow \sigma(\text{logit})$ # Eq.(17)
Same for document pairs

Loss:
$\mathcal{L}_{\text{preq}}^C$ and $\mathcal{L}_{\text{preq}}^D$ via binary cross-entropy # Eqs.(18-19)
$\mathcal{L} \leftarrow \lambda \mathcal{L}_{\text{preq}}^C + (1-\lambda)\mathcal{L}_{\text{preq}}^D$ # Eq.(20)

Return predictions and $\mathcal{L}$

C Experimental Details

C.1 Implementation Details

For concepts without Wikipedia pages, GPT-4 selects substitute Wikipedia entries used only for their textual descriptions. The prompt is: "Given a target concept, your task is to recommend the single

most semantically similar concept that has a corresponding Wikipedia page. The recommendation must be a direct synonym or a highly specific technical equivalent. Only output the name of the concept." These substitutions are fixed during training and evaluation; original concept identities and labels are retained. The weight decay is 6×10^{-5}; the task-balance parameter λ is 0.6 and 0.65 for UCD and LectureBank, respectively. BERT is used to initialize node representations. We evaluate on LectureBank and UCD, and include LECDPR (Xiao et al., 2026) among the concept-level baselines. All reported baseline scores are from our independent reruns on the same 8:1:1 splits as PAGTPRP. Table 2 reports five-seed means and sample standard deviations; the document prediction, ablation, and sparse-scenario results are from single runs. Experiments were run in PyTorch on a Linux server with an RTX 4090D GPU.

C.2 Dataset Descriptions and Statistics

The two benchmark datasets used in the experiments are described below; their statistics are summarized in Table 1 in the main paper.

University Course Dataset (UCD) (Liang et al., 2017): This dataset compiles course information from 377 universities in the United States, covering 376 courses in the field of computer science.

LectureBank (Li et al., 2019): This dataset originates from an online education platform. The original data were collected from multiple online education platforms through automated crawling and manual cleaning, focusing on publicly available course resources at the undergraduate and graduate levels in Computer Science.

C.3 Baseline Descriptions

HGT (Hu et al., 2020) develops an attention mechanism that is aware of node types. In the process of computing attention scores, the features of nodes are modified and assigned weights based on their types, allowing for improved capture of semantic relationships among nodes of various types.

SeHGNN (Yang et al., 2023) adopts a single-layer structure with long metapaths and integrates a Transformer-based semantic fusion section to fuse multi-source semantic features by precomputing neighbor aggregation, which achieves a simple network structure.

PREREQ (Roy et al., 2019) uses Pairwise-link LDA to learn concept representations and a Siamese neural network to predict concept prerequisite relationships.

ConLearn (Sun et al., 2022) merges the benefits of pre-trained language models such as BERT with graph neural networks in order to efficiently learn prerequisite relationships among concepts derived from sparse and unstructured educational data.

MHAVGAE (Zhang et al., 2022) presents a multi-head attention framework designed to enhance the ability to capture relationships among nodes across various dimensions, thereby increasing the precision of predictive outcomes.

HGAPNet (Mazumder et al., 2023) learns the heterogeneous feature representations and attention weights of nodes to effectively handle relationships between different types of nodes.

LCPRE (Sun et al., 2024) captures temporal concept context using sparse graphs and learning paths for prerequisite prediction.

MTN (Wang et al., 2024) extracts multi-view features through three components: a Resource-view Transformer, a Relation-view Transformer, and a Text-view Transformer. Finally, an attention-based aggregation method is adopted to fuse these features for predicting prerequisite relations.

GKROM (Zhang et al., 2025c) constructs a graph structure to represent knowledge and uses graph mining techniques to predict the relationships between concepts.

LECDPR (Xiao et al., 2026) uses two-stage LLM prompting to enrich concept and document representations, and models concept–document interactions through three propagation mechanisms. Random masking and contrastive learning align semantic and relational representations, which are adaptively fused for prerequisite prediction.