CAN-SIN: A Cross-Layer Heterogeneous Academic Network with Semantic Information

Yufei Tian*, Hong Hu*, Yuejiang Li*, H. Vicky Zhao*, and Yan Chen†
* Department of Automation, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, P. R. China
† School of Cyberspace Security, University of Science and Technology of China, Hefei, Anhui, P. R. China.

***Abstract*—In this paper, we focus on incorporating the semantic information into the structure of academic networks to enrich the dimensionality of extracted features. We propose a cross-layer scholar-paper network that can capture the characteristics of heterogeneous academic networks. In addition, we leverage the BERT model, which is widely used in the realm of natural language processing (NLP), to integrate the semantic information of the scholar papers. We also introduce a new concept, "close collaborator", to tackle data leakage issues. This can be used in many downstream tasks such as automatic detection of conflict of interests among scholars. Extensive experiments on two datasets show that our enhanced cross-layer model is both effective and lightweight, and outperforms three strong baselines. Further analysis shows that our model successfully combines the semantic information and the topology of the whole network.**

***Keywords*—Heterogeneous information network (HIN), academic network, graph convolutional networks (GCN), link prediction, pre-trained language model**

I. Introduction

Heterogeneous information networks (HINs) that differentiate node and edge types are almost everywhere in our daily life: social networks, road traffic, academic collaboration, etc. By modeling the entities (nodes) and their interactions as graphs, researchers have extended the theory of network analysis by integrating it with machine learning and network embedding algorithms. Recently, the success of deep learning has boosted in the graph domain. Specifically, the graph neural networks (GNNs) are utilize to solve different tasks, such as similarity search [1], node clustering [2], [3] and link prediction[4], [5] so as to capture the hidden information behind the data with non-Euclidean structure. Among the aforementioned tasks, link prediction plays an important role in recommendation systems like e-commerce (Amazon, Taobao) and social media (Facebook, Twitter, Weibo) platform.

In this paper, we focus on academic networks, which is of crucial significance. By analyzing the academic networks, we can provide paper recommendations to scholars, explore the cooperation mode of different academic teams, and help journals to analyze the conflicts of interest between scholars.

With the recent trend of graph convolutional networks (GCNs) [6], there are several attempts to combine GCNs with heterogeneous information networks, such as [7], [8], [9]. To model heterogeneity in academic networks (papers, authors, institutions, venues, etc.), the latest Heterogeneous Graph Transformer (HGT) model [9] adopts node-type and edge-type dependent parameters to characterize the heterogeneous attention over each edge, empowering HGT to maintain dedicated representations for different types of nodes and edges. However, these existing works on heterogeneous networks only distinguish the types of different nodes and edges and ignore that information from different domains can enrich the extracted features. As for the study of academic networks, current research mainly perform network analysis using the structure

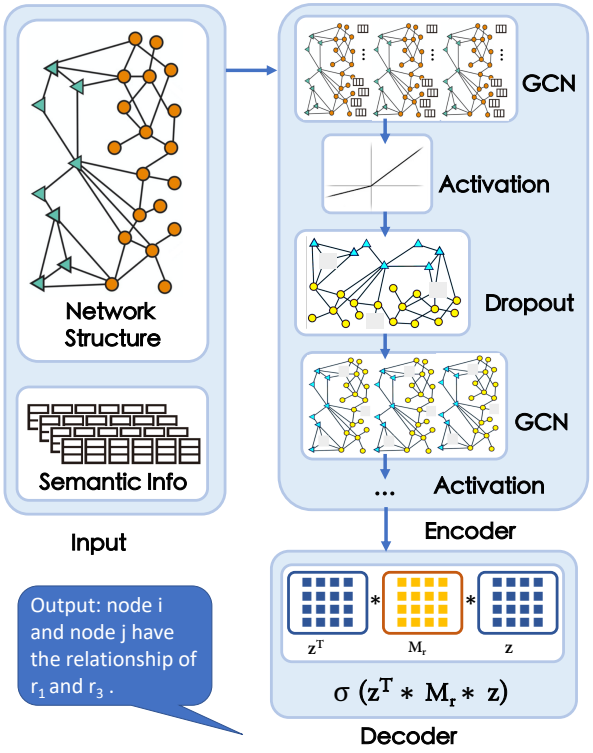

Fig. 1: An overview of the proposed heterogeneous architecture. The input consists of a cross-layer network and the semantic information, which is then fed into a GCN encoder. The encoder is responsible of learning the hidden representation **z** for each node, and the decoder calculates whether certain relationships exists between two given nodes based on **z**.

This work is supported by the National Key Research and Development Program of China (2017YFB1400100).

features such as meta-path [1] and the more advanced graph neural network [7], [8]. They have the following characteristics and disadvantages:

1) Current models, trained on deep neural networks, are often of high complexity, which demand strong computing power and cost lots of training time.
2) Manual definition of meta path is labor-intensive and hard to generalize.
3) Current works on academic networks do not include semantic information (such as title and abstract of an article), which can provide rich connotations for the model from another perspective.

In light of all these, we propose a lightweight cross-layer model that takes advantage of both network topology and semantic information, as shown in Fig. 1. We summarize our contributions as follow:

1) We enrich the dimension of information in traditional heterogeneous networks by introducing the content information of the article, and propose the **C**ross-layer **A**cademic **N**etwork with **S**emantic **IN**formation (**CANSIN**).
2) We collect and construct two datasets of scholars with different time spans, including the institutions of scholars, close collaborators, paper citation information, papers published in journals, paper titles and abstract contents, etc.
3) Our simulation results show that the proposed model is both effective and lightweight. Given the aforementioned dataset, compared with other heterogeneous graph models such as HGT, our proposed **CANSIN** model gains an average increase of 5% in prediction accuracy while using 90% fewer parameters.

II. Related Works

A. Metapath in Academic Network

To capture the features of heterogeneous academic graphs, one of the classical paradigms is to define and utilize meta paths to model heterogeneous structures, such as PathSim [1]. The meta-path is a path that defines the coincidence relationship between two objects in the network. Given a pattern with entity $\mathcal{A}$ and relation $\mathcal{R}$, the path form of A_i to A_{l+1} is denoted as:

$$A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}. \tag{1}$$

The meta-path defines a complex relationship between $A_1, A_2, \ldots, A_{l+1}$:

$$R = R_1 \circ R_2 \circ \cdots \circ R_l \tag{2}$$

Where $\circ$ represents the composition operator.

For example, authors can be connected via the "Author→Paper→Author" (APA) meta-path and the "Author→Paper→Venue→Paper→Author" (APVPA) mata-path, while authors and venues can be linked via the "Author→Paper→Venue" (APV) mata-path. However, the manual design of meta paths requires specific domain knowledge, and is thus difficult to generalize to other problems.

B. Network Embedding

Network embedding is usually designed to represent the nodes in a network as vectors. Embedding methods usually are based on the assumption that the similarity between nodes should be reflected in the learned feature representations. Popular methods include matrix decomposition [10], [11], DeepWalk [12], large-scale information network embedding [13] and node2vec [14]. With the contained attribute information and topological information of nodes in the network, network embeddings can be applied to tasks such as classification, clustering, prediction and generation.

C. Graph Neural Networks and Heterogeneity

The research of GNNs, especially on heterogeneous information networks, has attracted much attention in the fields of machine learning and data mining, and is closely related to our work.

Specifically, the Graph Convolutional Network (GCN) [6] is targeted at generating the representation of node v from both its own feature x_v and its neighbor's ($N(v)$) feature x_u, where $u \in N(v)$, so as to extract the whole representation of the node by stacking several graphic convolutional layers. Each layer encapsulates the hidden representation of the node by aggregating feature information from its neighbors. Through the K-stack step, the final hidden representation of each node can contain the characteristic information of the neighbor up to K. Based on GCN, an unsupervised learning framework called Variational Graph Auto-encoder (VGAE) is introduced in [15]. A VGAE framework encodes nodes in the Graph into a hidden D-dimension vector space using a GCN encoder, and then reconstructs the original Graph data based on the encoded hidden information using decoder. This model makes use of latent variables and is capable of learning interpretable latent representations for undirected graphs.

Recently, in view of graph neural networks' (GNNs) success [16], [17], there are several attempts to adopt deep neural networks to heterogeneous graphs [7], [8], [9]. A latest architecture, Heterogeneous Graph Transformer (HGT) [9], models Web-scale dynamic heterogeneous graphs on Open Academic Graph (OAG) [18]. Similar to previous works, the goal of HGT is to aggregate the information of source node s to obtain the context representation of target node t. This process can be decomposed into three parts: 1) heterogeneous mutual attention, 2) heterogeneous message passing, and 3) target-specific aggregation. The obtained representation for each node can be then utilized in many downstream tasks such as node clustering and link prediction.

D. Sentence Embedding and Pre-trained Model

In word embedding, each word is mapped to a low-dimensional vector. The characteristics of each word are preserved in the latent space by aligning similarity in corpus with vector dot product similarity.

Usually, a pre-trained model on a large corpus needs to be introduced for generating word and sentence embeddings. Popular word-level pre-trained models include word2vec [19]

and GloVe [20]. Word embedding achieves huge success, but still fails to extract semantics and logics at sentence or paragraph level. Recently, researchers start to focus on pre-training at higher level. Some successful attempts include Embedding from Language Models (ELMo) [21] proposed by AllenNLP and GPT model [22] proposed by OpenAI.

Bidirectional Encoder Representation form Transformers (BERT) [23] is a revolutionary language representation model proposed by Google AI, which is the first truly bidirectional unsupervised model pre-trained on massive plain text data. BERT is trained with two tasks: (1) masked language modeling, i.e., fill in the blanks in one sentence; (2) next sentence prediction, i.e., determine whether one sentence is the next sentence of another. In summary, BERT model generates a vector representation for each sentence input. This vector is contextualized, and maintains the semantics of the sentence, which ca be applied to downstream tasks, such as link prediction. It brings natural language processing (NLP) to a new era by making its massively pre-trained language model readily available to all researchers, saving time, resources and knowledge.

III. The Proposed CANSIN Framework

A. Modeling of Cross-layer Academic Networks

Motivation Taking a certain scholar C as an example, suppose we have the following information:

- the institution in which scholar C works, and possibly his specific laboratory as affiliations;
- the articles he has published, illustrated as orange circles outlined with solid black lines in Fig. 2;
- the basic information of the paper, such as the year of publication, the journal, the title of the article and the abstract information;
- the references of this article, namely the lines inside the orange circle in Fig. 2; and
- interaction between scholar C and other scholars, such as close collaborators.

Based on the above information, we can build the academic network in Fig. 2 with a cross-layer model and abundant semantic information.

Fig. 2 is a schematic diagram of the modelling of our academic network. The model is divided into two layers, with blue-green triangles at the top layer representing the network of scholars and orange circles below forming the network of academic papers.

The upper layer Each blue-green triangle (node) in the top network refers to one scholar. For each link in this layer and its two corresponding scholars, depending on their affiliations and their closeness in cooperation, their relationship can be classified into three types (r_1, r_2, r_3):

- r_1- they are affiliated with the same institute but not the same lab;
- r_2- they are in the same lab in the same institute;
- r_3- they have co-authored at least n articles in m consecutive years and are close collaborators.

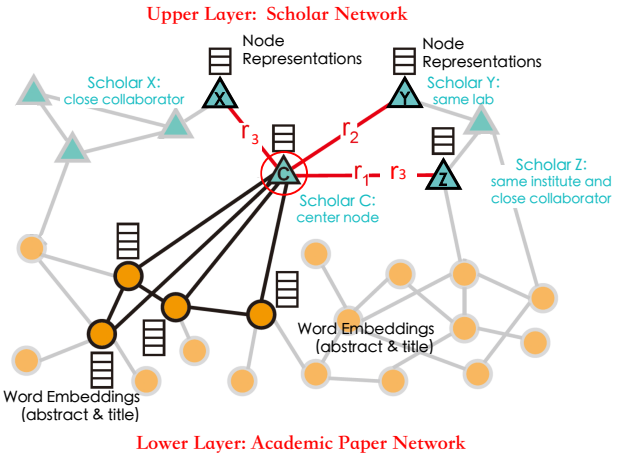

Fig. 2: A two-layer academic network where both semantic information and network structure are considered. Blue-green triangles at the upper layer form the network of scholars, and orange circles at the lower layer form the network of academic papers. The two networks are connected by a publishing relationship.

Note that r_1 and r_2 are mutually exclusive, but r_3 and can be superimposed on r_1 or r_2. Take node C in Fig. 2 as an example, from left to right he/she:

- works closely with scholar X, so that only r_3 exists between node C and X;
- works in the same lab as scholar Y, but have not established a close collaboration relationship, so that only r_2 exists between node C and Y; and
- belongs to the same institution with Scholar Z but not in the same lab, and are close collaborators, so that both r_1 and r_3 exist between node C and Z.

The lower layer Each orange node in the lower layer corresponds to a published scholar paper, and each node is represented by word vectors of the corresponding article's title and abstract, initially generated using BERT. A link between two nodes in the lower layer represents paper-paper citation relationship (r_5), and a link connecting a node in the upper layer and a node in the lower layer represents the scholar-paper publication relationship (r_4).

B. Task Definition

In total, there are five types of links defined in the network. Namely, three relationships among scholars (r_1, r_2, r_3), author-paper publishing relationship (r_4), and paper-paper citation relationship (r_5). The task of our work is to predict all five possible link relationships in cross-layer academic networks.

We formally define our task as follow: for each node $v_i \in V = \{V_{scholar}, V_{paper}\}$, we have the labeled edges (relationships) $r^{ij} \in R = \{r_1, r_2, r_3, r_4, r_5\}$ with its neighbour $v_j \in N_{v_i}$, forming a graph $G = (V, R)$. For each triplet (v_i, r^{ij}, v_j) and all possible relation types $r^{ij} \in \{r_1, r_2, r_3, r_4, r_5\}$, the

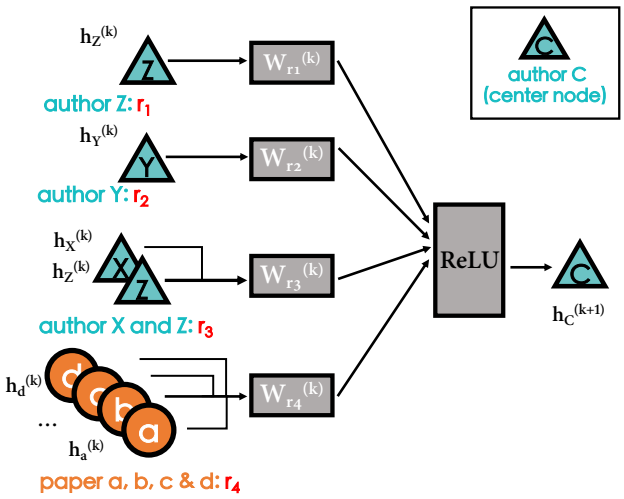

Fig. 3: One layer of our GCN encoder that is responsible of updating the hidden state of node C at the (k+1)-th layer based on the representations at the k-th layer. The annotations are consistent with Fig. 2 and Eq. 4.

model should generate a binary prediction

$$P_r^{ij} = P(r^{ij}|G, x(v_i), x(v_j)) \in \{0, 1\} \tag{3}$$

as accurately as possible, to predict whether or not r^{ij} exists as r_1, r_2, r_3, r_4, and r_5 respectively. Here $x(v_i)$ and $x(v_j)$ are the feature representations node of v_i and v_j. Among them, the feature vectors of scholars are randomly initialized, while the those of academic paper are initialized by the word vectors obtained from BERT [23].

C. System Design

Our system design is inspired by a multi-drug framework, Decagon [24], which is proposed to predict polypharmacy side effects with graph convolutional networks. Motivated by the Decagon model, we propose an encoder-decoder based **C**ross-layer **A**cademic **N**etwork with **S**emantic **IN**formation (**CANSIN**).

Graph Convolutional Encoder In the graph encoder, the proposed model takes the cross-layer network G and its additional node feature vector $\mathbf{x}_v$ as input, and generates a d dimensional node embedded $\mathbf{z}_i \in \mathbb{R}^d$ for each node in the graph (scholars and academic papers).

Our graph encoder is a stack of two GCN layers. The encoder propagates the adjacent node feature information between the graph edges while considering the type of edge $r \in r_1, r_2, r_3, r_4, r_5$). As is shown in Fig. 3, each single layer of this neural network model adopts the following form:

$$\mathbf{h}_i^{(k+1)} = ReLU\left(\sum_r \left[\sum_{j \in \mathcal{N}_r^i} c_r^{ij} \mathbf{W}_r^{(k)} \mathbf{h}_j^{(k)} + c_r^i \mathbf{h}_i^{(k)}\right]\right), \tag{4}$$

where $h_i^{(k+1)}$ is the hidden state of v_i at the (k+1)-th layer, $d^{(k)}$ is the dimension of the k-th layer, and c_r^{ij}, c_r^i are normalization factors. Weight $\mathbf{W}_r^{(k)}$, dependent on r, is the trainable matrix to be learned. $\mathcal{N}_r^i$ is the set of v_i's neighbour that connected via r, with $r \in \{r_1, r_2, r_3, r_4, r_5\}$ indicating the edge type. By taking sum over $\mathcal{N}_r^i$ and r, the graph encoder updates h_i^k by aggregating the feature vectors of its neighbors depending on relation types.

Such operation is then applied K times in a row so that the encoder can effectively convolve the information in the K-th neighborhood into the embedded representation of the current node. At the same time, for each type of link relationship, a specific transformation matrix is trained to keep the heterogeneity.

Graph Decoder Based on the node embedding obtained from the above graph encoder, the decoder reconstructs labeled edges. Specifically, it generates a probability for each possible edge based on the node vector of the last hidden layer from the encoder ($\mathbf{z}_i$).

For each candidate triplet $\left(v_i, r^{ij}, v_j\right)$, the decoder predicts its likelihood using a scoring function $g\left(v_i, r^{ij}, v_j\right)$. According to $\mathbf{z}_i$ and $\mathbf{z}_j$, the scoring function g yields a continuous score ($\in [0, 1]$) indicating the probability that nodes v_i and v_j interact through the candidate relationship type r^{ij}. The decoder can be written as

$$p_r^{ij} = \sigma(g\left(v_i, r^{ij}, v_j\right)) = \sigma(\mathbf{z}_i^T \mathbf{M}_r \mathbf{z}_j), \tag{5}$$

where $\mathbf{M}_r$ is a type-dependent trainable parameter matrix that models interactions between two hidden representations, and σ is a sigmoid function that introduces non-linearity to the model.

Finally, we need to binarize the continuous probability that falls between 0 and 1 for prediction and evaluation purposes. We set the threshold as 0.5 and map all p_r^{ij}s ($r \in \{r_1, r_2, r_3, r_4, r_5\}$) to binary ones.

Loss Function During the training step, we adopt cross-entropy loss to optimize the parameters. The loss function can be written as follows:

$$J_r(i, j) = -\log P_r^{ij} - E_{n \sim p_r ij} \log\left(1 - P_r^{in}\right). \tag{6}$$

For each training iteration, only one type of relation r is optimized. Specifically, iteration No. $(5N + k)$ is responsible for relation type $r = r_k$, where N is an arbitrary natural number and $k \in \{1, 2, 3, 4, 5\}$.

The final loss function of all relation types $\mathcal{R}_k, k \in \{1, 2, 3, 4, 5\}$ can be written as:

$$J = \sum_{r \in \mathcal{R}_k} J_r(i, j). \tag{7}$$

D. Evaluation Metrics

We choose three evaluation indexes most suitable for link prediction and our specific task: AUROC, AUPRC, and accuracy, where acccuracy is defined as the number of correct predictions over the total sample size. AUROC is the area under the ROC curve, and AUPRC is calculated as the area under the

PRC curve. The ROC curve shows the tradeoff between true positive rate (TPR, on the x-axis) and false positive rate (FPR, on the y-axis) at different decision thresholds. The x-axis of the PRC is precision and the y-axis is recall.

IV. Experimental Setup

A. Dataset construction

For data constraction, we leverage the Aminer citation dataset[1], which includes academic papers from DBLP, ACM, etc. In order to obtain the corresponding cross-layer academic network, we preprocess and then manually annotate the citation dataset on Aminer:

1) Close Collaboration: Since all co-authorship information has been included in author-paper links, we construct a "close collaboration" relationship, which is impossible to directly infer from existing links. To this end, we include temporal information - if and only if scholar A and Scholar B have coauthored at least m articles for n consecutive years, they are defined as close collaborators. In this experiment, we set n to 2 and m to 2 in order to avoid data leakage issues.

2) Affiliation: Although the Aminer dataset provides the author's affiliation information, the descriptions are all unstructured text provided by individual scholars. Even for the same laboratory from the same university, the texts filled out by authors vary. We therefore use keyword matching to group authors in coarse granularity first, and then manually mark specific labs affiliated to more than 70 major research institutions.

3) Overall Statistics: We obtained two datasets of different sizes in the data mining field, one is the academic network with papers published on SIGKDD from 2002 to 2007, the other is the academic network with papers published on several conferences in the data mining field from 2002 to 2010. The specific information is as follows:

i. SIGKDD
- Including academic papers published at SIGKDD only
- year: 2002 $\sim$ 2007
- 1699 authors and 876 articles
- 70 influential institutions and 8 specific laboratories
- author-author relationship: 733 affiliation relations (r_1 and r_2) and 202 close collaboration relations (r_3)
- 2639 author-paper relations (r_4)
- 780 paper-paper citations (r_5)

ii. Data Mining (DM)
- Including academic papers published at several data mining related venues.
- year: 2002 $\sim$ 2010
- 6887 authors and 3695 articles
- 74 influential institutions and 34 specific laboratories
- author-author relationship: 1836 affiliation relations (r_1 and r_2) and 1036 close collaboration relations (r_3)
- 11110 author-paper relations (r_4)
- 3350 paper-paper citations (r_5)

[1]https://www.aminer.cn/citation

TABLE I: Hyper-parameter settings

Name	Value
Negative sample size	1
Dropout rate	0.1
Learning rate	0.075
Batch size	32
Train: valid: test	77.5:7.5:15

B. Baseline Models

Our experiment has three baseline models: PathPredict, Heterogeneous Graph Transformer (HGT) and Decagon. In this section, we first introduce the PathPredict method, and then discuss some adjustments to the latter two, which have already been introduced in Section II and III, for fair comparison.

PathPredict We follow the steps of PathPredict, a supervised metapath based relationship prediction model proposed in [25]. First, metapath based topological features are extracted from the heterogeneous academic network, such as path count and random walk. Then, we use a supervised logistic regression model to learn the best weights associated with different features in deciding the relationships.

HGT In addition to the attention mechanism and information transmission and integration, HGT also applies 1) relative temporal encoding (RTE) to process the temporal information, and 2) sub-sampling techniques to deal with web-scale networks efficiently. Considering the fact that our dataset is static and relatively small in size, the RTE and sub-sampling techniques are not introduced. The adjusted HGT model is composed of heterogeneous mutual attention, message passing and aggregation.

Decagon Considering the fact that a large number (963) of side effects of polypharmacy are sparsely and unevenly distributed in the drug-protein network, the original Decagon model adopts AP@50 (Average percision at top 50) as evaluation index. For the link prediction task in our academic network, where there are a total of 5 possible relationships, AP@50 is no longer suitable. For each of the five relationships, we chose AUROC (area under ROC curve), AUPRC (area under PRC curve) and accuracy as our evaluation metrics, as mentioned in Section III-D.

C. Implementation Details

We manually tune the hyper-parameters and list the best values of our hyper-parameters in TABLE I.

In order to find the best dimension of the two hidden layers in the decoder step, we try three combinations, i.e. 16-8, 32-16 and 64-32. We then plot the training loss, validation accuracy and validation AUC to select the best combination of hidden dimensions. Due to space limit, we only show one representative group in Fig. 4. As can be seen, the accuracy and AUC values on the validation set of the first model, with dimension 16-8, are about 80% and 75%, while the training error is at least 0.3. The accuracy and AUC of the second model, with the dimension of 32-16, are 80% $\sim$ 85%. The training loss is also lower. The performance of the last model,

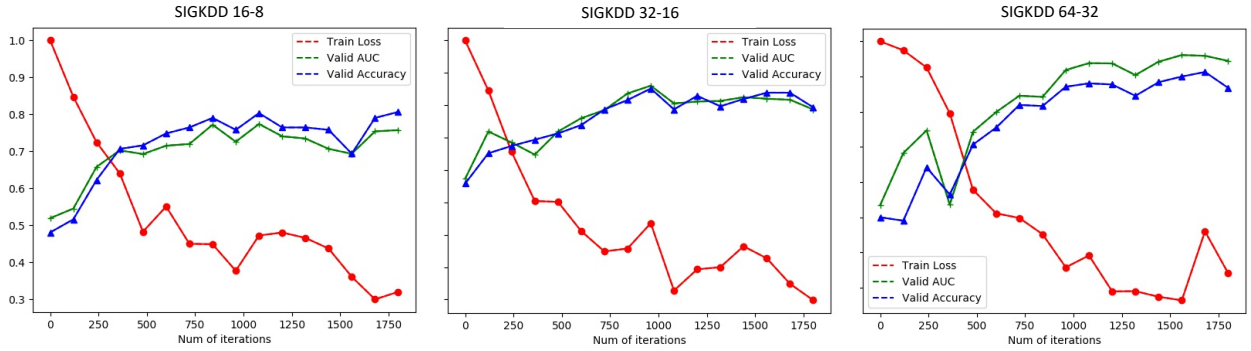

Fig. 4: Learning curves of close collaborator relationship on the SIGKDD dataset

TABLE II: Performance on SIGKDD dataset

Model performance on SIGKDD dataset		CANSIN	PathPredict	HGT	Decagon
r_1	AUROC	**0.927**	0.826	0.923	0.889
	AUPRC	**0.924**	0.806	0.885	0.871
	Accuracy	**0.853**	0.717	0.827	0.781
r_2	AUROC	**0.714**	0.613	0.645	0.701
	AUPRC	0.645	0.632	0.662	**0.724**
	Accuracy	0.609	0.520	0.565	**0.620**
r_3	AUROC	**0.908**	0.819	0.861	0.805
	AUPRC	**0.911**	0.812	0.849	0.801
	Accuracy	**0.870**	0.738	0.801	0.779
r_4	AUROC	**0.851**	0.724	0.784	0.790
	AUPRC	**0.858**	0.727	0.741	0.782
	Accuracy	**0.805**	0.652	0.758	0.754
r_5	AUROC	**0.785**	0.760	0.701	0.710
	AUPRC	**0.741**	0.660	0.570	0.576
	Accuracy	**0.799**	0.682	0.773	0.761

TABLE III: Performance on Data Mining (DM) dataset

Model performance on DM dataset		CANSIN	PathPredict	HGT	Decagon
r_1	AUROC	**0.956**	0.857	0.937	0.943
	AUPRC	**0.952**	0.846	0.945	0.933
	Accuracy	**0.876**	0.754	0.858	0.851
r_2	AUROC	0.724	0.635	**0.744**	0.617
	AUPRC	0.685	0.609	**0.801**	0.666
	Accuracy	**0.661**	0.551	0.550	0.571
r_3	AUROC	**0.972**	0.823	0.905	0.913
	AUPRC	**0.969**	0.816	0.908	0.908
	Accuracy	**0.923**	0.732	0.862	0.866
r_4	AUROC	**0.840**	0.717	0.787	0.776
	AUPRC	**0.835**	0.693	0.738	0.747
	Accuracy	**0.808**	0.649	0.790	0.775
r_5	AUROC	**0.857**	0.744	0.799	0.816
	AUPRC	**0.773**	0.696	0.698	0.712
	Accuracy	**0.844**	0.733	0.815	0.806

with dimension of 64-32, outperforms the former two by hitting over 85% on accuracy and 90% on AUC.

Based on our simulation results, we then conclude that as the dimension of the hidden layer increases, the model effect gradually gets better. However, when we further increase the model dimension, the improvement in performance is not significant. Therefore, a 64-32 combination of hidden layer dimensions is ideal.

V. Results and Analysis

A. Main results

Recall that we construct two datasets from the Aminer citation dataset and repeat all the experiments four times to avoid random factors. For each dataset, the mean of three indexes: AUROC, AUPRC and accuracy of all models are reported in TABLE II and TABLE III, respectively..

As can be seen, our **CANSIN** model achieves satisfactory results on most relation prediction tasks. For example, the accuracy of the our model in relationship (r_1) reached 85.3% and 87.5%, which is 7.2% and 2.5% higher than Decagon. The accuracy of the enhanced model in the scholar-paper authorship (r_4) is 80.5% and 80.8%, i.e. 5.0% and 3.3% higher than Decagon, and 4.6% and 1.7% higher than HGT. PathPredict on the other hand gets relatively low scores as it is merely a regression model of observed metapath features. Among the five tasks of link prediction, our proposed cross-layer **CANSIN** model is consistently better than the other three baseline models, with exception only for the r_2 relationship. We notice that the total sample size of relationship r_2 is in the order of hundreds, with even fewer samples in the test set, meaning that one single correct (or wrong) prediction can have a significant impact on the final performance.

Based on the results in TABLE II and TABLE III, we observe that

1) Our proposed graph convolutional encoder and type-dependent decoder are the best at capturing node features and link prediction tasks.
2) After adding semantic information into heterogeneous network (**CANSIN**), the performance for various relation types are improved. It indicates that semantic information (one representative type of multi-source information) is highly useful for current settings.

The significance of semantic information After adding semantic information to articles, the improvement in the smaller dataset (SIGKDD) is much more obvious than that in larger dataset (Data Mining). A possible explanation is that compared with the smaller dataset, the Data Mining dataset contains richer amount of direct and latent information, hence the semantic

information from an unseen domain makes less contribution to the overall richness of data. Namely, for smaller datasets, the introduction of semantic information will greatly increase the dimensionality of input data and thereby improves the accuracy of the model more significantly.

B. Complexity Analysis

To further understand the advantages and disadvantages of **CANSIN** compared to other deep learning models, we compute the number of parameters of **CANSIN**, HGT can Decagon in TABLE IV. Note that although **CANSIN** leverages the embeddings obtained from BERT, we exclude the parameters of pre-trained BERT because the operation to obtain the sentence embedded representation is one-off and should be regarded as the initialization or pre-processing part.

TABLE IV: Statistics of model parameter size

Model	CANSIN	Decagon	HGT
No. Params	0.546 M	0.504 M	7.44 M

The number of parameters of the enhanced model is 546,000, which is about 40,000 more than that of the Decagon model, but it is far smaller than the 7.44-million parameter size of the HGT model. In other words, the complexity and training time of **CANSIN** or Decagon is less than 10% of that of HGT.

C. Comparison and Discussion

According to section V-A and section V-B, both HGT and Decagon are suitable for cross-network link prediction. The modified HGT is composed of message passing and aggregation, and the advanced attention mechanism. On the other hand, the **CANSIN**/Decagon model is a simple combination of graph encoder and decoders. The HGT model, despite its more complex design and a larger number of parameters (a stronger fitting ability), performs moderately in this task – its prediction performance on 5 out of 10 relation types is better than that of the Decagon model. In this section, we further analyze the features of **CANSIN**, Decagon and HGT.

First, the attention mechanism and parameter size of HGT does not give full play to its advantages. In theory, the addition of attention should intuitively help HGT to learn the weight of each feature, and thus help HGT to more accurately locate important information. However, the real-world meaning of node connections in the current network is very clear (close collaborator, co-authorship, reference, paper publication relationship, etc.). Whether two nodes are interconnected and the type of link can naturally transfer the significance carried in each node.

Second, in our academic network, there are more than one edge types between certain nodes. Our type-dependent bilinear decoder in the **CANSIN** model is specially designed to predict the existence of multiple relation types. Besides, the possible 5 edge types are more densely and uniformly distributed. Our proposed **CANSIN** model is most suitable for such topology. HGT does not support high dimensional heterogeneity in edge types, while the Decagon model is targeted at link prediction on a skewed dataset with more than 900 candidates.

Lastly, the original HIN models (our two baselines) only considers network features, while the **CANSIN** model takes advantage of both network features and semantic meanings. Based on the results shown in TABLE II and TABLE III, **CANSIN** has an average advantage of 5.9% over Decagon and an average advantage of 5.0% over HGT. We can conclude that the content of each article is definitely crucial because it contributes to a more diverse dataset with richer perspectives.

D. Visualization of Article Embeddings

To get a more intuitive understanding of the learning process, we select 2 groups of articles and plot the change of their node embeddings before and after the training stage. Specifically, we map the 768-dimensional vectors to 2-dimensional space using the t-SNE method, and use arrows to indicate the direction of changes. For the first group, we select one academic paper as center node, along with its nine neighbours: three references (Ref $1 \sim 3$ in Fig. 5(a)) and six future works that cite this article (Future work $1 \sim 6$ in Fig. 5(a)). For the second group, we randomly select 10 articles that do not each other and plot the changes of embeddings for each node in Fig. 5(b).

For each node representation, the initial embedding (obtained from pre-trained BERT) is purely semantic oriented. while the final node embedding contains both semantic information of the articles and the structure of whole academic network. As can be seen in Fig. 5(a), the sub-graph becomes more concentrated because the 10 nodes are all inter-connected. Their representations becomes closer after training, where the model learns to incorporate the network topology with the original semantic features. In contrast, the nodes in Fig. 5(b) move much more stochastically, as they are randomly selected and do not have any correlation, meaning that our model indeed learns the topology of the academic network in a correct way.

VI. CONCLUSION

This paper focuses on link prediction tasks in academic networks by fusing the semantic information with cross-layer networks. We obtain the embeddings of titles and abstracts of academic papers using BERT, and then incorporate these contextualized embeddings to the encoder-decoder-based **CANSIN** model. In addition, our model is not limited to the realm of academic networks, and can be easily extended to other domains, such as social networks and molecular networks as long as there are extra features at hand.

To the best of our knowledge, we are the first to define *close collaborators* based on temporal information. This concept can not be simply derived from the affiliation or cooperative papers and effectively avoids data leakage problems. Finally, the enhanced cross-layer model, **CANSIN**, achieves good results on the above datasets – the accuracy of **CANSIN** is in average 5.9% (or 5.0%) higher than that of Decagon (or HGT). The complexity analysis also shows that **CANSIN** is lightweight and costs relatively small computational resources.

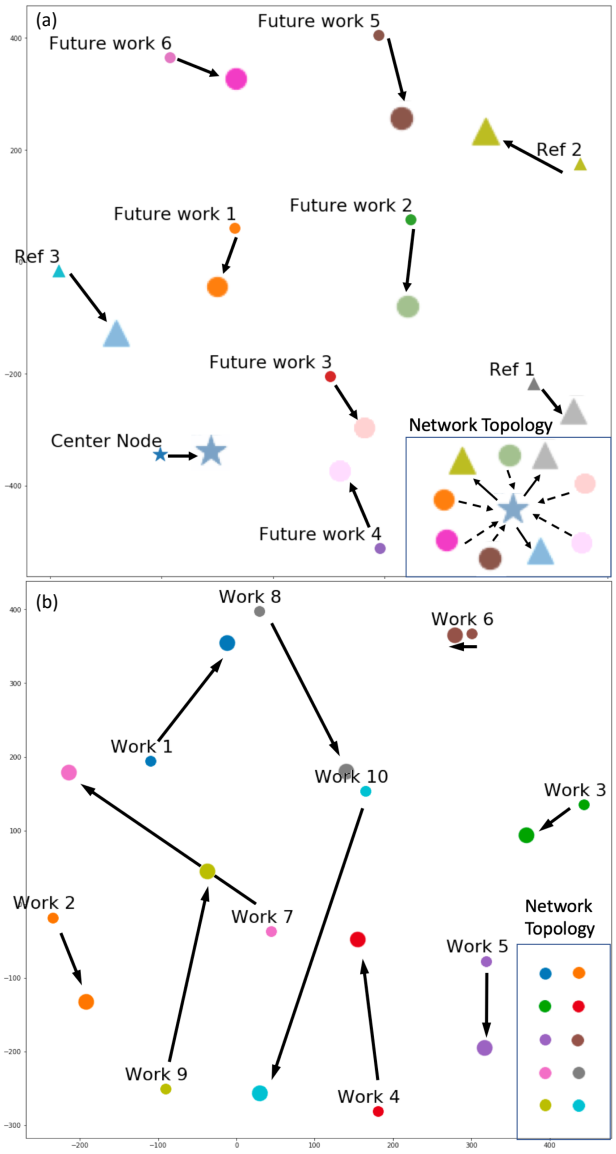

Fig. 5: Comparison3 of how the node embeddings of (a) 10 *inter-connected* and (b) 10 *randomly sampled* articles change before and after training. We map the 768-dimensional vectors to 2-dimensional space using the t-SNE method, and use arrows to indicate the direction of changes. The bigger stars/triangles/dots represent the updated embeddings while the smaller ones represent the initial ones. We also plot the network topology on the right-bottom corner of each sub-figure. The solid arrows link the center article with its references, and the dotted arrows link the center article with future works that cite itself. Comparatively, the points in sub-figure (a) gather together after training, while the nodes move stochastically in sub-figure (b).

Future directions include building dynamic cross-layer networks and exploring real-world applications of the **CANSIN** model. For example, the link prediction of close collaborators automatically analyzes whether there is conflict of interest between two scholars, which can therefore facilitate the automatic assignment of conference/journal reviewers.

REFERENCES

[1] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[2] M. Ji, J. Han, and M. Danilevsky, "Ranking-based classification of heterogeneous information networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1298–1306.

[3] P. Bangcharoensap, T. Murata, H. Kobayashi, and N. Shimizu, "Transductive classification on heterogeneous information networks with edge betweenness-based normalization," in *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 2016, pp. 437–446.

[4] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "Shine: Signed heterogeneous information network embedding for sentiment link prediction," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018, pp. 592–600.

[5] F. Liu and S.-T. Xia, "Link prediction in aligned heterogeneous networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2015, pp. 33–44.

[6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[7] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[8] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 793–803.

[9] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of The Web Conference 2020*, 2020, pp. 2704–2710.

[10] X. Shen, S. Pan, W. Liu, Y.-S. Ong, and Q.-S. Sun, "Discrete network embedding," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3549–3555.

[11] H. Yang, S. Pan, P. Zhang, L. Chen, D. Lian, and C. Zhang, "Binarized attributed network embedding," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 1476–1481.

[12] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[13] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proceedings of the 24th international conference on world wide web*, 2015, pp. 1067–1077.

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[15] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[16] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[18] F. Zhang, X. Liu, J. Tang, Y. Dong, P. Yao, J. Zhang, X. Gu, Y. Wang, B. Shao, R. Li *et al.*, "Oag: Toward linking large-scale heterogeneous entity graphs," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2585–2595.

[19] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2013, pp. 746–751.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[21] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *arXiv preprint arXiv:1802.05365*, 2018.

[22] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[23] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[24] M. Zitnik, M. Agrawal, and J. Leskovec, "Modeling polypharmacy side effects with graph convolutional networks," *Bioinformatics*, vol. 34, no. 13, pp. i457–i466, 2018.

[25] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. Han, "Co-author relationship prediction in heterogeneous bibliographic networks," in *2011 International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2011, pp. 121–128.