Pre-training on Large-Scale Heterogeneous Graph

Xunqiang Jiang[1], Tianrui Jia[1], Yuan Fang[2], Chuan Shi[1,3], Zhe Lin[3], Hui Wang[3]

[1]Beijing University of Posts and Telecommunications [2]Singapore Management University [3]Peng Cheng Laboratory, Shenzhen, China

Abstract

Graph neural networks (GNNs) emerge as the state-of-the-art representation learning methods on graphs and often rely on a large amount of labeled data to achieve satisfactory performance. Recently, in order to relieve the label scarcity issues, some works propose to pre-train GNNs in a self-supervised manner by distilling transferable knowledge from the unlabeled graph structures. Unfortunately, these pre-training frameworks mainly target at homogeneous graphs, while real interaction systems usually constitute large-scale heterogeneous graphs, containing different types of nodes and edges, which leads to new challenges on structure heterogeneity and scalability for graph pre-training. In this paper, we first study the problem of pre-training on a large-scale heterogeneous graph and propose a novel pre-training GNN framework, named PT-HGNN. The proposed PT-HGNN designs both the node- and schema-level pre-training tasks to contrastively preserve heterogeneous semantic and structural properties as a form of transferable knowledge for various downstream tasks. In addition, a relation-based personalized PageRank is proposed to sparsify a large-scale heterogeneous graph for efficient pre-training. Extensive experiments on one of the largest public heterogeneous graphs demonstrate that our PT-HGNN significantly outperforms various state-of-the-art baselines.

Definition of a Heterogeneous Graph

A heterogeneous graph, denoted by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, \phi, \varphi\}$, is a form of graph, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges, respectively. It is also associated with a node-type mapping function $\phi : \mathcal{V} \to \mathcal{A}$ and an edge-type mapping function $\varphi : \mathcal{E} \to \mathcal{R}$, where $\mathcal{A}$ and $\mathcal{R}$ denote the sets of node and edge types such that $|\mathcal{A}| + |\mathcal{R}| > 2$. Moreover, the network schema $T_{\mathcal{G}} = (\mathcal{A}, \mathcal{R})$ of a heterogeneous graph specifies the type constraints on the nodes and their relations, which can guide the exploration of heterogeneous structural contexts on the graph. Figure 1(a) shows an example of heterogeneous graph and its network schema with four types of node and edge.

PT-HGNN Framework

In this work, we introduce our pre-training framework PT-HGNN on a large-scale heterogeneous graph. More specifically, we first elaborate on the design of the pre-training tasks for a heterogeneous graph. To preserve the heterogeneity, we propose both node- and schema-level pre-training tasks to respectively utilize node relations and the network schema, which encourages the GNN to capture heterogeneous semantic and structural properties. Second, we present our edge sparsification strategy on a large-scale heterogeneous graph for pre-processing. To avoid unwanted biases toward certain types of node, we propose a relation-based personalized PageRank to retain the most useful graph structures, in order to accelerate the pre-training procedure. Figure 1 shows the overall framework of the proposed PT-HGNN.

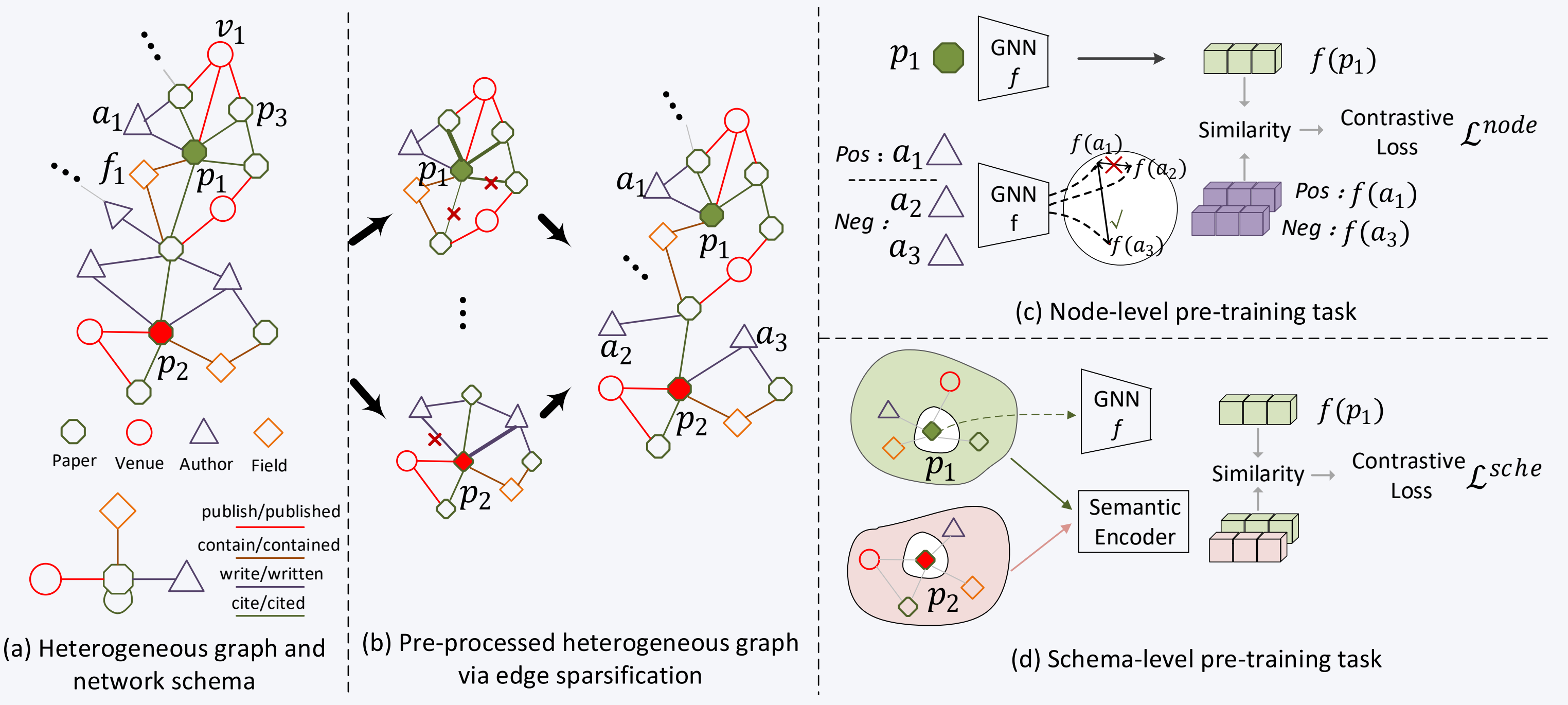


Figure 1:The overall framework of PT-HGNN.

Node-level Pre-training Task

Here, we design a node-level pre-training task to encode the semantics, which allows us to model pairwise relations between different types of node. A relation between two nodes conveys important semantic information about them. On one hand, a positive triple $\langle u, R, v\rangle$ in a heterogeneous graph $\mathcal{G}$ means that nodes $u \in \mathcal{V}$ and $v \in \mathcal{V}$ are linked via a specific relation $R \in \mathcal{R}$ on $\mathcal{G}$. We construct the negative samples in a relation-specific manner and consistent with the alignment principle. In specific, for a given positive triplet $\langle u, R, v\rangle$, we define the negative samples for a relation R as:

$$\mathcal{N}^{node}_{\langle u,R,v\rangle} = \left\{\langle u, R, v^-\rangle \mid \phi(v) = \phi(v^-), (u, v^-) \notin \mathcal{E}, Sim(v, v^-) \le \delta\right\}, \quad (1)$$

where Sim is a function to measure the similarity between the node representations, and δ is a threshold for filtering out too similar nodes in violation of the alignment principle.

Schema-level Pre-training Task

In order to incorporate the high-order heterogeneous semantic and structural contexts in a heterogeneous graph, a natural idea is to utilize high-order semantic patterns.

We utilize the network schema to strike a balance between capturing the heterogeneity and achieving efficiency on a large-scale heterogeneous graph. We generate schema instances to construct the positive and negative samples, to complement the first-order node-level samples.

Formally, given a schema instance containing node u, let u be the target node and the other nodes in the instance be the context nodes of u. Based on this definition, we consider the context nodes of target node u as the schema-level positive samples for u, denoted $\mathcal{P}^{sche}_u$, given by

$$\mathcal{P}^{sche}_u = \bigcup_{s \in I(u)} s \backslash \{u\}, \quad (2)$$

where $I(u)$ denotes the set of all schema instances containing node u.

To construct the schema-level negative samples for target node u, we follow two approaches: 1) if two network schema instances are generated from two different target nodes of the same type, we treat them as negative schema samples of each other; and 2) we design a dynamic queue for storing the negative network schema samples.

$$\mathcal{N}^1_u = \{\mathcal{P}^{sche}_{u^-} \mid u^- \in \mathcal{V}_B, u \ne u^-, \phi(u) = \phi(u^-)\}. \quad (3)$$

$$\mathcal{N}^2_u = \{\mathcal{P}^{sche}_v \mid \phi(u) = \phi(v), v \in \mathcal{V}^{t-1}_B\}, \quad (4)$$

where $\mathcal{V}^{t-1}_B$ is the node set of the previous batch. It is worth noticing that the queue is initialized as an empty set in the beginning and updated during the training procedure. Therefore, we obtain the overall schema-level negative samples as follows:

$$\mathcal{N}^{sche}_u = \mathcal{N}^1_u \cup \mathcal{N}^2_u. \quad (5)$$

Sparsification of Large-Scale Heterogeneous Graph

To pre-train a better GNN, the key is to leverage a large-scale heterogeneous graph. We resort to offline sparsification to retain the most important edges in the graph. Inspired by previous studies, personalized PageRank can be utilized to preserve more effective neighborhood. We can obtain a dense matrix Π^R with pairwise personalized PageRank scores for each relation R, in contrast to the sparse adjacency matrix A^R under the same relation. Note that the values in Π^R represent the pairwise influence between all pairs of nodes in relation R, which typically are highly localized. Spatial localization allows us to simply truncate small values of Π^R and recover sparsity. Similar to previous studies, we use the top-k entries with the highest mass per column and set all other entries to zero in Π^R.

The edge sparsification for the heterogeneous graph $\mathcal{G}$ will be done as a pre-processing step. Subsequently, we obtain the sparsified graph $\mathcal{G}'$ for pre-training, which can accelerate the aggregation operation of GNNs.

Traditional Experiment

We conduct experiments on Open Academic Graph (OAG), which is consitituted by papers (P), authors (A), venues (V), institutes (I), fields (F) and their relations with 178 million nodes and 2.236 billion edges. As far as we know, this is the largest publicly available heterogeneous graph. To test the generalization ability and transferability of the proposed pre-training framework, we also construct four representative domain-specific subgraphs from OAG: Computer Science (CS), Material Science (Mater), Engineering (Engin) and Chemistry (Chem).We consider the prediction of Paper–Field, Paper–Venue, and Author Name Disambiguation (Author ND) as three downstream tasks used in prior works. We only show the result on the whole OAG dataset here.

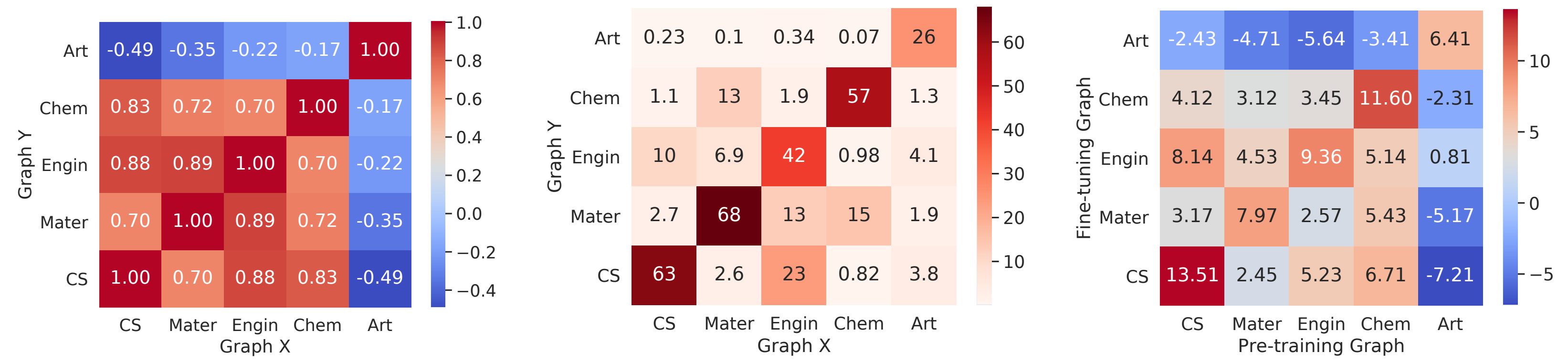
Dataset	Downstream Task		No pre-train	EdgePred	DGI	ContextPred	GraphCL	GPT-GNN	PT-HGNN	Improv.
OAG	Paper–Field	NDCG	32.33±0.36	38.03±0.33	37.12±0.42	38.40±0.49	39.32±0.30	<u>40.76</u>±0.40	**42.33**±0.62	3.85%
		MRR	28.15±0.48	44.23±0.56	42.96±0.43	43.15±0.55	45.65±0.60	<u>45.70</u>±0.41	**47.29**±0.49	3.48%
	Paper–Venue	NDCG	42.28±0.50	43.25±0.61	<u>44.23</u>±0.53	43.07±0.74	42.66±0.66	44.05±0.75	**47.13**±0.68	6.56%
		MRR	22.76±0.37	23.40±0.35	24.38±0.35	24.12±0.42	25.03±0.48	<u>25.19</u>±0.45	**26.75**±0.57	6.19%
	Author ND	NDCG	76.52±1.13	78.01±0.86	77.98±0.93	77.88±0.72	78.11±0.93	<u>79.33</u>±0.87	**79.99**±0.92	0.83%
		MRR	54.65±0.53	58.00±0.63	58.30±0.48	57.49±0.60	58.22±0.53	<u>59.08</u>±0.52	**61.32**±0.55	3.79%

Table 1:Performance comparison between our method and baselines (best result in bold, and second best underlined).

Transfer Experiment

We investigate how the network structures affect the ability of knowledge transfer from pre-training to fine-tuning, and examine the applicability of our proposed pre-training strategies. We compute the correlation between graphs using a series of commonly used graph property metrics, so as to quantify the structural difference between graphs. Figure 2(a) shows a heatmap of the pairwise correlation among five representative graphs. This difference in network structures is also verified by the citation coefficient, which measures the percentage of publications in graph Y that have citations in graph X, as shown in Figure 2(b).

Next, we evaluate the model performance by pre-training with one graph and fine-tuning with another for the Author ND task. The MRR improvement of our proposed method over the one without pre-training is shown in Figure 2(c).



(a) Correlation based on graph properties (b) Citation coefficient (%) (c) MRR gain (%) over no pre-training

Figure 2:Visualization of pairwise correlation and knowledge transfer among five graphs.

Conclusion

In this work, we take the first attempt to pre-train GNNs on a large-scale heterogeneous graph, and introduce a pre-training framework named PT-HGNN. First, to preserve the heterogeneity, we propose both node- and schema-level pre-training tasks to utilize node relations and the network schema, respectively, which enables the GNN to capture heterogeneous semantics and structural properties. Second, to pre-train on large-scale heterogeneous graphs, we present an edge sparsification strategy via relation-based personalized PageRank, which retains meaningful graph structures while accelerating the pre-training procedure. Extensive experiments on one of the largest heterogeneous graphs, OAG, demonstrate the superior ability of our PT-HGNN to transfer knowledge to various downstream tasks via pre-training.