

Generalized Taxonomy-Guided Graph Neural Networks

Yu Zhou¹, Di Jin¹, Jianguo Wei¹, Dongxiao He¹, Zhizhi Yu^{1*} and Weixiong Zhang²

¹College of Intelligence and Computing, Tianjin University, Tianjin, China

²Department of Health Technology and Informatics, The Hong Kong Polytechnic University, Kowloon, Hong Kong

{zhouyu_, jindi, jianguo, hedongxiao, yuzhizhi}@tju.edu.cn, weixiong.zhang@polyu.edu.hk

Abstract

Graph neural networks have been demonstrated to be effective analytic apparatus for mining network data. Most real-world networks are inherently hierarchical, offering unique opportunities to acquire latent, intrinsic network organizational properties by utilizing network taxonomies. The existing approaches for learning implicit hierarchical network structures focus on introducing taxonomy to graph neural networks but often run short of exploiting the rich network semantics and structural properties in the taxonomy, resulting in poor generalizability and reusability. To address these issues, we propose generalized Taxonomy-Guided Graph Neural Networks (TG-GNN) to integrate taxonomy into network representation learning. We first construct a taxonomy representation learning module that introduces the concept of ego network to propagate and aggregate rich semantic and structural information in the taxonomy. We then design a taxonomy-guided Markov mechanism, which encapsulates taxonomy knowledge in pairwise potential functions, to refine network embeddings. Extensive experiments on various real-world networks illustrate the effectiveness of TG-GNN over the state-of-the-art methods on scenarios involving incomplete taxonomies and inductive settings.

1 Introduction

Network systems are ubiquitous in real life. With the recent success of neural networks, many network embedding techniques have been proposed for network analysis by transforming node representations into a low-dimensional space. Among these methods are Graph Neural Networks (GNNs) [Kipf and Welling, 2017; Velickovic *et al.*, 2018], which are best known for their ability to obtain node embeddings by propagating and aggregating network features across given networks. The GNN methods have been broadly adopted for diverse applications like natural language processing [Huang and Carley, 2019; Qu *et al.*, 2020], traffic forecasting [Bai *et al.*, 2021; Bui *et al.*, 2022], recommendation systems [He *et al.*, 2020; Wang *et al.*, 2019a; Ying *et al.*, 2018], cybersecurity [Liu *et al.*, 2021; Lu and Li, 2020].

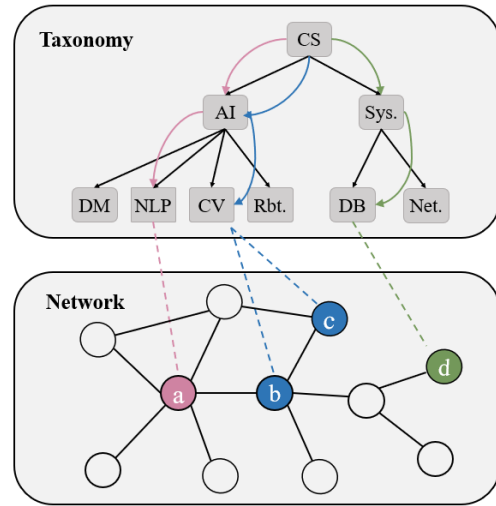


Figure 1: A toy example of a paper citation network with the research topic taxonomy.

et al., 2020; Wang *et al.*, 2019a; Ying *et al.*, 2018], cybersecurity [Liu *et al.*, 2021; Lu and Li, 2020].

The classical GNNs and their variants [Kipf and Welling, 2017; Hamilton *et al.*, 2017; Velickovic *et al.*, 2018] iteratively update node representations through passing messages among neighboring nodes. However, many real-world networks are inherently hierarchical, and the neighborhood structures of nodes are often arranged based on the underlying implicit taxonomies that dictate system organizational structures. Fig. 1 shows an example of a citation network based on a taxonomy of research topics. Categories in taxonomy can be viewed as groups of highly related entities, and subcategories prescribe a hierarchical structure. Correspondingly, nodes representing network entities tend to form implicit hierarchical communities within which they share varying degrees of similarity. For example, nodes **a** and **c** are first-order neighbors of node **b**, and node **d** is a second-order neighbor of node **b**. In computer science, **a**, **b**, **c** and **d** are all computer science-related papers, while by a detailed classification, papers **a**, **b** and **c** belong to a general category of AI, and papers **b** and **c** are categorized to the specialized computer vision field. Thus, for paper **b**, paper **c** is the most similar or closest neighbor, followed by **a**, and then **d**. In

*Corresponding author

this toy example, paper **b** has different granularities of similarity with its neighbors, indicating an implicit hierarchical structure. Unfortunately, the existing GNN methods fail to capture the latent hierarchical structure in this example, resulting in distortions when embedding real-world networks with hierarchical structures.

Several algorithms and models have been designed to capture latent hierarchical structures within networks. For example, it has been attempted to infer network hierarchies and use estimated hierarchical structures to facilitate network representation learning [Chami *et al.*, 2019; Ma *et al.*, 2018; Lin *et al.*, 2022]. However, the quality of inferred hierarchical structures depends on the network, and the results lack interpretability. Another line of attempt is to introduce network-specific taxonomy to graph neural networks to learn network hierarchical structures, thereby improving representation learning. For instance, TaxoGAN [Yang *et al.*, 2020] co-embeds networks and taxonomies by computing network embedding using hierarchy labels to generate fine-grained label embedding. TaxoGNN [Xu *et al.*, 2022] jointly learns the taxonomy and network representation by aligning taxonomy knowledge with the network structure through information distillation. However, these methods overlook the rich semantic and structural information in the given taxonomy. They are difficult to generalize and reuse, especially in scenarios where taxonomy is partially or completely missing.

To address these issues, we propose generalized Taxonomy-Guided Graph Neural Networks (TG-GNN) to incorporate network-specific taxonomies to integrate taxonomy guidance into network representation learning. TG-GNN comprises two key modules: taxonomy representation learning and network representation learning. The taxonomy representation module constructs a position-aware ego network for each category and introduces a position-enhanced graph neural network to capture taxonomical information. It generates pseudo-training data based on the given taxonomy. The network representation module incorporates a taxonomy-guided Markov mechanism to encapsulate taxonomy knowledge in pairwise potential functions to refine network embedding, ensuring effective guidance of the taxonomy knowledge space toward the network data space. Extensive experiments illustrate that our proposed TG-GNN significantly outperforms the state-of-the-art methods on various real-world networks, including scenarios with incomplete taxonomy and inductive settings.

2 Preliminaries

2.1 Terms and Notations

Network. Consider a network $G = \{V, E, X\}$, with a set of nodes $V = \{v_1, \dots, v_n\}$, a set of edges $E = \{e_{ij}\} \subseteq V \times V$, and a node attribute matrix X . The topological structure of G is represented by an adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, where $a_{ij} = 1$ if nodes v_i and v_j are connected, or $a_{ij} = 0$ otherwise.

Taxonomy. A taxonomy $T = (C, L)$ organized in a tree-like form related to a network G can be formulated by a directed acyclic graph, where a node $c_i \in C$ represents a category and a directed edge $\langle c_i, c_j \rangle \in L$ indicates a category-category

relation, expressing that c_i serves as the ‘‘parent’’ of c_j and c_j as the ‘‘child’’ of c_i in the taxonomy.

Network with Taxonomy. Each node v_i in the network G can be attached to a category c_{i,d_i} in the taxonomy T , and thus labeled with a category path $(c_{i,1}, c_{i,2}, \dots, c_{i,d_i})$ from root category $c_{i,1}$ to category c_{i,d_i} . D is the depth of the taxonomy. If c_{i,d_i} is a leaf category, $d_i = D$, otherwise $d_i < D$. We may extend a path until its length becomes D .

2.2 Markov Random Fields

Markov Random Field (MRF) [Picka, 2006] is a probability distribution P over variables defined by an undirected graph. It is characterized by an energy function E consisting of unary potentials ϕ and pairwise potentials ψ :

$$E(Y; A, X) = \sum_{v_i} \phi(y_i) + \sum_{(v_i, v_j) \in E} \psi(y_i, y_j), \quad (1)$$

where y_i and y_j are the labels of node v_i and node v_j , respectively. The unary term $\phi(y_i)$ for node v_i measures the cost for having label y_i , and the pairwise term $\psi(y_i, y_j)$ represents the cost that v_i and v_j have labels y_i and y_j , respectively. The MRF can be estimated by using the Gibbs distribution to compute the posterior probability of label Y given network topology A and node attributes X , defined as:

$$P(Y|A, X) = \frac{1}{Z} \exp(-E(Y; A, X)), \quad (2)$$

where Z is the partition function for normalization. The best label assignments \hat{Y} can then be obtained:

$$\hat{Y} = \arg \max_Y P(Y|A, X). \quad (3)$$

3 Methodology

In the sequel, we first briefly discuss the new TG-GNN method, whose architecture is sketched in Fig. 2. We then present the two learning modules.

3.1 Overview of Taxonomy-Guided Graph Neural Networks

To effectively integrate taxonomy guidance into network representation learning, we propose generalized Taxonomy-Guided Graph Neural Networks (TG-GNN, Fig. 2). TG-GNN comprises two key modules: 1) The taxonomy representation learning module that constructs a position-aware ego network to capture the semantic and structural information in the taxonomy. 2) The network representation learning module employs a graph attention mechanism to extract semantic and topological information from the network. It then passes this information to the taxonomy-guided Markov component to optimize the outcome of labeling.

3.2 Taxonomy Representation Learning

The taxonomy representation learning module captures the rich semantic and structural information in the taxonomy and provides the learned taxonomy knowledge to the network representation learning module. We first employ Word2Vec [Mikolov *et al.*, 2013] to represent individual categories of

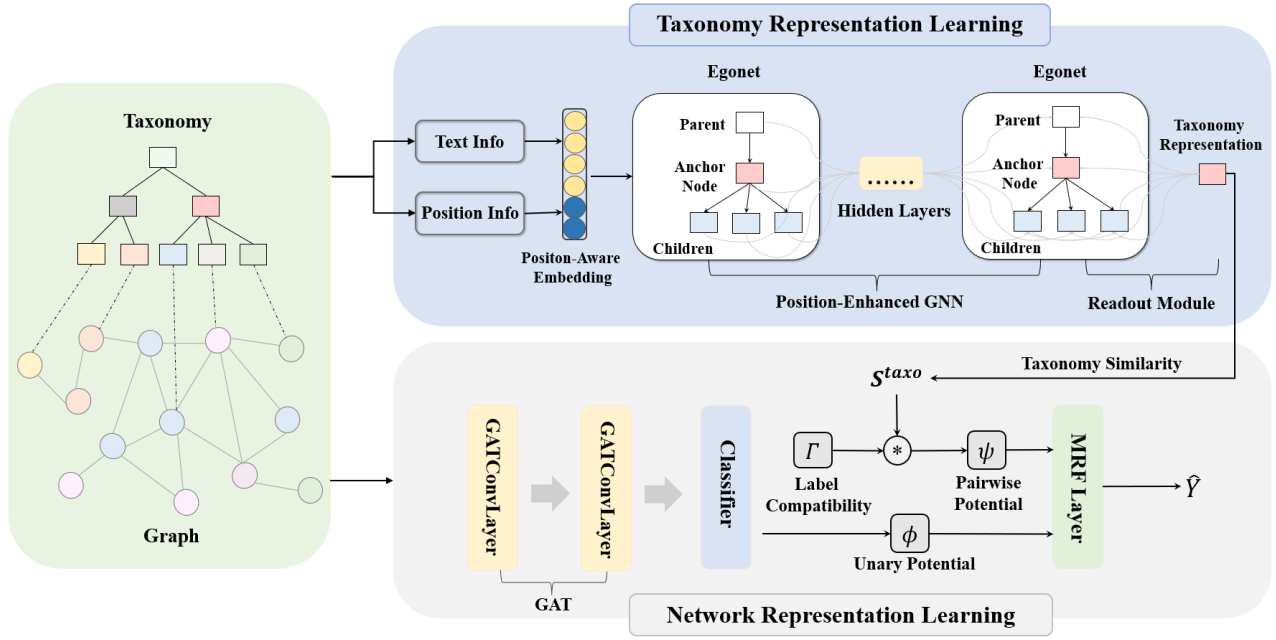


Figure 2: The architecture of TG-GNN.

the taxonomy tree as the taxonomy features. We then construct an ego network for each node within the taxonomy, enhancing the node embedding with the node position information. We model the taxonomy representation using a position-enhanced graph neural network.

Ego Network. It is a network constructed for each node to capture parent-child and hierarchical relationships. Since the edges in a taxonomy are unidirectional, information can only flow from the top to the bottom of the taxonomy tree, and parent nodes cannot learn information from their child nodes. However, as the surface names of parent and child nodes in the taxonomy might differ, there is no direct connection. For example, in the ACM computing classification, where “Natural Language Processing” serves as the parent category of “Machine Translation” without an explicit surface name connection, their initial embeddings reside far apart in the low-dimensional representation space. To address this issue, we consider each taxonomy category as an anchor concept and build an ego network that includes itself, its parent category, and subcategories so as to better model taxonomy categories.

Position-aware Embedding. In addition to modeling the semantic information in taxonomy, we introduce three position types to nodes: the anchor, parent and child nodes. Such positional information can be used to distinguish different ego networks and separate nodes with different identities (i.e., anchor vs. parent vs. child nodes) during information propagation and aggregation. Consider a graph neural network with K layers. We represent the attributes of taxonomy node c_i in the k -th ($k = 1, 2, \dots, K$) layer of GNN by $h_i^{(k)}$, and the position embedding of node c_i in the k -th layer of the graph neural network is represented as $p_i^{(k)}$, then the position-aware embedding of node c_i at layer k is

$$h_i^{\prime(k)} = h_i^{(k)} \parallel p_i^{(k)}. \quad (4)$$

where \parallel represents the concatenation operation.

Position-Enhanced Graph Neural Network. Given an anchor node c_i and its ego network, motivated by [Shen *et al.*, 2020], we use a position-enhanced graph neural network to generate the final representation for the anchor node. Specifically, we use position-enhanced graph neural networks and a neighborhood aggregation strategy to iteratively update the representation of node c_i by aggregating the representations of its own and its neighbors $N(c_i)$, i.e., $N(c_i) \cup c_i$, short noted as $\tilde{N}(c_i)$. After K iterations, a node’s representation captures structural information within its K -hop neighborhood. Formally, we define a position-enhanced graph neural network with K layers as follows:

$$h_i^{(k+1)} = \text{AGG}^{(k+1)}(\{h_j^{\prime(k)} | c_j \in \tilde{N}(c_i)\}), \quad (5)$$

where $k \in \{0, \dots, K-1\}$. Subsequently, we use a multi-head graph attention network with M attention heads to instantiate the aggregation function:

$$h_i^{\prime(k+1)} = \parallel_{m=1}^M \rho \left(\sum_{c_j \in \tilde{N}(c_i)} \alpha_{ij}^{(k)} W_m^{(k)} h_j^{\prime(k)} \right), \quad (6)$$

where \parallel represents the concatenation operation, ρ is a non-linear function, e.g., ReLU, $W_m^{(k)}$ is the m -th weight matrix in the m -th attention head, and $\alpha_{ij}^{(k)}$ is the importance of the characteristics of nodes c_j to c_i , which can be defined as:

$$\alpha_{ij}^{(k)} = \frac{\exp(\gamma(a^{(k)} [W^{(k)} h_i^{\prime(k)} \parallel W^{(k)} h_j^{\prime(k)}]))}{\sum_{c_r \in \tilde{N}(c_i)} \exp(\gamma(a^{(k)} [W^{(k)} h_i^{\prime(k)} \parallel W^{(k)} h_r^{\prime(k)}]))}, \quad (7)$$

where $a^{(k)}$ and $W^{(k)}$ are both learnable parameters, $\gamma(\cdot)$ is another nonlinear function (such as LeakyReLU), and \parallel represents the concatenation operation.

Readout Function. After obtaining the final representation $h_i^{(K)}$ of each node, we employ the readout function to generate the representation h_G of the ego network G and use h_G as the final embedding of the anchor node:

$$h_G = \text{READOUT}(h_i^{(K)} | v_i \in G), \quad (8)$$

where the READOUT function is a weighted element-wise mean function, with weights related to node positions:

$$\text{READOUT}(\cdot) = \sum_{v_i \in G} \frac{\log(1 + \exp(\alpha_{p_i}))}{\sum_{v_j \in G} \log(1 + \exp(\alpha_{p_j}))} h_i^{(K)}, \quad (9)$$

where α_{p_i} is a parameter indicating the importance of position p_i .

3.3 Network Representation Learning

The network representation learning module first uses graph neural networks to extract the semantic and topological information of the network. Subsequently, inspired by [He *et al.*, 2018], we design a taxonomy-guided network-specific Markov mechanism to encapsulate node-related taxonomy knowledge by the pairwise potential function to direct the learning process.

Taxonomy-guided Markov Mechanism. We employ the unary potential function $\phi(y_i)$ as the interface between GNN and MRF, which allows the amalgamation of the network’s learned semantics and topology into one MRF layer. Formally, the unary potential function can be defined as:

$$\phi(y_i) = -\log p(y_i), \quad (10)$$

where $p(y_i)$ is the probability that node v_i has label y_i whose initial value comes from GNN’s result. Then, inspired by [Dong *et al.*, 2023], we employ the pairwise potential function of the MRF to encapsulate node-related taxonomy information. For each node v_i , it is assumed to have an associated taxonomy path $C_i = (c_{i,1}, c_{i,2}, \dots, c_{i,d_i})$, and the taxonomy path representation H_i is the sequential concatenation of category representations in path $H_i = h_{i,1} \| h_{i,1} \| \dots \| h_{i,d_i}$. We utilize the cosine similarity to quantify the taxonomy correlation between nodes:

$$S_{i,j}^{taxo} = \cos(w_{i,j} \odot H_i, w_{i,j} \odot H_j), \quad (11)$$

where \odot represents the Hadamard product, $w_{i,j}$ is a learnable parameter matrix. H_i and H_j are the taxonomy path representations of node v_i and node v_j respectively, which are learned by the taxonomy representation learning module. From the above definition, it can be seen that $S_{i,j}^{taxo}$ reflects the strength of the similarity between taxonomies associated with node v_i and node v_j . Therefore, the pairwise potential function can be defined as:

$$\psi(y_i, y_j) = S_{i,j}^{taxo} \Gamma(y_i, y_j), \quad (12)$$

where $\Gamma(y_i, y_j)$ is the label compatibility, which equals one if $y_i \neq y_j$ or 0 otherwise. By using the pairwise potential function, node pairs with similar taxonomy information but inconsistent labels will be penalized. The more similar the taxonomy information H_i and H_j of node v_i and node v_j is,

the greater the value of $S_{i,j}^{taxo}$ becomes. Correspondingly, the penalty increases when the value of $\psi(y_i, y_j)$ grows. Therefore, we incorporate the guidance of the taxonomy knowledge space into the network data space via the pairwise potential function of the Markov random field.

Mean Field Approximate. As the distribution of $P(Y|A, X)$ is difficult to compute precisely and efficiently, we employ the mean-field theory [Koller and Friedman, 2009] to approximate $P(Y|A, X)$ by an approximate distribution $Q(Y|A, X)$, which minimizes the KL-divergence $D(Q||P)$. $Q(Y|A, X)$ is decomposable, which can be represented as:

$$Q(Y) = \sum_{v_i \in V} q(y_i), \quad (13)$$

$$q(y_i) = \frac{1}{Z_i} \left\{ - \left(\phi(y_i) + \alpha \sum_{(v_i, v_j) \in E} \psi(y_i, y_j) \right) \right\}, \quad (14)$$

where Z_i is a partition function for normalization.

Then, we stack K MRF layers to obtain the matrix form of Eq. (14), which can be updated iteratively as:

$$Q^{(k)} = \text{softmax} \left(\log(Q^{(0)}) - \alpha S^{taxo} Q^{(k-1)} \Gamma \right), \quad (15)$$

where k represents the k -th MRF layer, the inputs to the MRF layer are the label prediction probabilities $Q^{(0)} = p(y_i) \in \mathbb{R}^{n \times d_L}$ from the GNN, and d_L is the dimensionality of node labels. $S^{taxo} \in \mathbb{R}^{n \times n}$ represents the taxonomy correlation between nodes, and $\Gamma(y_i, y_j) \in \mathbb{R}^{d_L \times d_L}$ is the label compatibility matrix.

3.4 Learning and Inference

For the taxonomy representation learning, given a taxonomy $T = (C, L)$, we generate self-supervised data and train the model using a self-supervised training method. Specifically, for each category c_q in the taxonomy, we extract its parent node c_p as a positive example (c_q, c_p) with the label of 1 and construct M negative pairs $\{(c_q, c_r^1), (c_q, c_r^2), \dots, (c_q, c_r^M)\}$ by randomly select M other nodes c_r^l ($l = 1, 2, \dots, M$) with the label of 0, which are neither the parents nor the descendants of c_q . During the training process, c_q serves as the query node and c_p and c_r^l as anchor nodes. Given the constructed self-supervised binary tuples, the model is tasked to predict the labels for these tuples. Given the constructed self-supervised binary tuples, the model is tasked to predict the labels for these tuples. We use InfoNCE loss [van den Oord *et al.*, 2018] to predict the labels for these tuples in the self-supervised data $D_s = \{(c_q, c_p), (c_q, c_r^1), (c_q, c_r^2), \dots, (c_q, c_r^M)\}$ to train the model:

$$\mathcal{L}_{taxo} = -\log \frac{f(c_q, c_p)}{\sum_{(c_q, c_j) \in D_s} f(c_q, c_j)}, \quad (16)$$

where $f(\cdot)$ is the matching module, functioning as a log-bilinear model, $j = 1, 2, \dots, M + 1$, and \mathcal{L}_{taxo} is the cross entropy of classifying the positive pair (c_q, c_p) correctly.

For the network representation learning, we design a cross-entropy loss function for the node classification:

$$\mathcal{L}_{net} = - \sum_{v_i \in V_L} y_i \log \hat{y}_i, \quad (17)$$

where V_L denotes the node set with labels, y_i is the ground truth, and \hat{y}_i is the predicted label distribution of node v_i from the last MRF layer.

4 Experiments

Here, we discuss the experiment setup, compare the proposed TG-GNN with the best existing methods for node classification and visualization, analyze different components of TG-GNN by an ablation study, and present a cold start experiment and parameter analysis.

4.1 Experimental Setup

Datasets. We used four real-world networks with taxonomies (Table 1). Aminer¹ is a co-author network with the research topic taxonomy², where node labels represent the types of conferences authors participate in. Patent³ is a patent citation dataset with the patent taxonomy, where the range of the patent generality index determines the node labels. PubMed⁴ is a protein network with the disease taxonomy⁵, where a clustering algorithm on the protein network determines the node label. Yelp⁶ is a business network with the category taxonomy⁷, where node labels are determined by a clustering algorithm on the business network, which can be viewed as product themes. Notably, the PubMed and Yelp taxonomies are incomplete, with only about 20% and 66% of the network nodes containing taxonomy information, respectively.

Datasets	Aminer	Patent	PubMed	Yelp
#Nodes	13,900	7,313	9,619	14,573
#Edges	58,869	16,097	25,655	55,243
#Labels	4	2	4	3
#Taxonomy Categories	184	42	86	420
#Taxonomy Layers	4	2	3	4

Table 1: Statistics of datasets.

Baseline Methods. We evaluate the performance of TG-GNN by comparing it with seven state-of-art algorithms.

- **GCN** [Kipf and Welling, 2017] and **GAT** [Veličković *et al.*, 2018] iteratively update node representations through message passing among neighboring nodes.
- **HGCN** [Chami *et al.*, 2019] and **Poincaré** [Nickel and Kiela, 2017] extend GCNs into hyperbolic geometry to infer latent hierarchies from networks and use estimated hierarchies to facilitate network representation learning.
- **Tag2Guass** [Wang *et al.*, 2019b] introduces flat labels to enhance network embedding, which ignores the hierarchical relationship between the labels.

¹<https://www.aminer.cn/data/?nav=openDataAcademic-Social-Network>

²<https://dl.acm.org/ccs>

³<https://www.nber.org/research/data/us-patents>

⁴<ftp://ftp.ncbi.nih.gov/pub/taxonomy>

⁵<ftp://ftp.ncbi.nlm.nih.gov/>

⁶<https://www.yelp.com/dataset>

⁷https://www.yelp.com/developers/documentation/v3/all_category_list

- **TaxoGAN** [Yang *et al.*, 2020] and **TaxoGNN** [Xu *et al.*, 2022] introduce taxonomy explicitly into GNNs and jointly learn the taxonomy and graph representations.

Implementation Details. We used the recommended parameter settings from the respective codes for all baseline methods. In taxonomy representation learning, we employed a self-supervised training approach to acquire representations for all taxonomy categories. Our approach uses a two-layer position-enhanced GAT, with the first layer comprising four attention heads and the second layer using one attention head. Both layers utilize 200-dimensional semantic embeddings and 50-dimensional position embeddings, with a dropout rate of 0.1 applied to the input feature vectors. Additionally, we utilized the Adam optimizer with an initial learning rate of 0.001. In network representation learning, we utilized two GAT layers along with three MRF layers and set the dropout ratio to 0.5 to prevent overfitting. Each dataset is divided into three distinct sets: training, validation, and testing, with allocation proportions of 60%, 20%, and 20%, respectively.

4.2 Application 1 – Node Classification

We compared the performance of these methods in a node classification task. We report the mean Accuracy and F1 Score along with the standard deviation of 10 independent trials with different random seeds.

As listed in Table 2, our model outperforms all compared baseline methods on all four datasets. Specifically, TG-GNN achieves up to 1.46%, 3.61%, 6.27% and 21.07% better accuracy than the best baseline method on Aminer, Patent, PubMed and Yelp datasets, respectively. TG-GNN also achieves up to 1.36%, 3.92%, 2.65% and 20.88% higher F1 Score than the best baseline method on these four datasets. TG-GNN improves more significantly than the existing methods when network taxonomies are incomplete, such as PubMed and Yelp. Surprisingly, when taxonomies are incomplete, taxonomy-guided methods TaxoGAN and TaxoGNN perform poorly, even underperforming traditional data-driven GNN models like GCN and GAT. These results suggest that the existing methods are not robust nor generalizable to networks with incomplete taxonomies. In contrast, our new TG-GNN is highly robust and generalizable to all types of networks we considered, particularly those with incomplete taxonomies. Moreover, our new method performs better than HGCN and Poincaré that implicitly model network hierarchies. This result suggests that it is necessary to explicitly introduce taxonomy and fully exploit semantic and structural information of the taxonomy.

4.3 Application 2 – Visualization

To demonstrate the effectiveness of our new taxonomy representation learning method, we used t-SNE [van der Maaten and Hinton, 2008] to downscale the learned taxonomy representations to a two-dimensional space. Fig.3 shows the representation visualization of the research topic taxonomy related to Aminer. It can be observed that embeddings generated by Word2Vec rely solely on the textual names of the taxonomic categories and fail to capture the intricate hierarchical relationships among them. Taxonomic categories are scattered

Methods	Aminer		Patent		PubMed		Yelp	
	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
GCN	84.60 ± 0.21	71.28 ± 0.45	50.96 ± 1.60	44.43 ± 5.60	64.85 ± 0.89	61.60 ± 0.83	60.35 ± 0.57	58.96 ± 1.90
GAT	85.96 ± 0.09	74.39 ± 0.19	51.30 ± 1.30	45.90 ± 4.00	67.02 ± 0.37	61.64 ± 0.51	59.85 ± 0.73	51.77 ± 1.50
Poincaré	81.03 ± 0.12	63.98 ± 0.30	58.63 ± 1.90	58.23 ± 1.90	54.50 ± 0.19	40.62 ± 0.41	43.65 ± 0.12	28.05 ± 2.00
HGCN	87.59 ± 0.16	79.59 ± 0.23	55.21 ± 3.70	60.61 ± 3.10	77.64 ± 0.39	77.82 ± 0.33	61.59 ± 1.60	62.82 ± 0.67
Tag2Guass	84.93 ± 0.27	73.61 ± 0.41	55.44 ± 2.20	55.05 ± 2.50	68.94 ± 0.95	68.28 ± 1.00	61.49 ± 0.83	61.06 ± 0.93
TaxoGAN	83.33 ± 1.56	70.41 ± 2.48	<u>60.11 ± 2.42</u>	<u>59.78 ± 2.53</u>	53.50 ± 0.73	37.30 ± 0.84	44.13 ± 1.13	27.03 ± 1.16
TaxoGNN	85.45 ± 0.22	74.84 ± 0.40	58.97 ± 1.40	58.93 ± 1.40	55.41 ± 0.71	49.36 ± 1.70	<u>65.05 ± 0.93</u>	<u>64.24 ± 1.10</u>
TG-GNN	89.05 ± 0.18	80.95 ± 0.33	63.72 ± 1.40	63.70 ± 1.40	83.91 ± 2.10	80.47 ± 3.30	86.12 ± 2.10	85.12 ± 2.50

Table 2: Node classification results with mean value and standard deviation in terms of Accuracy (%) and F1 Score (%). Bold and underline are adopted to display the best and the second best results.

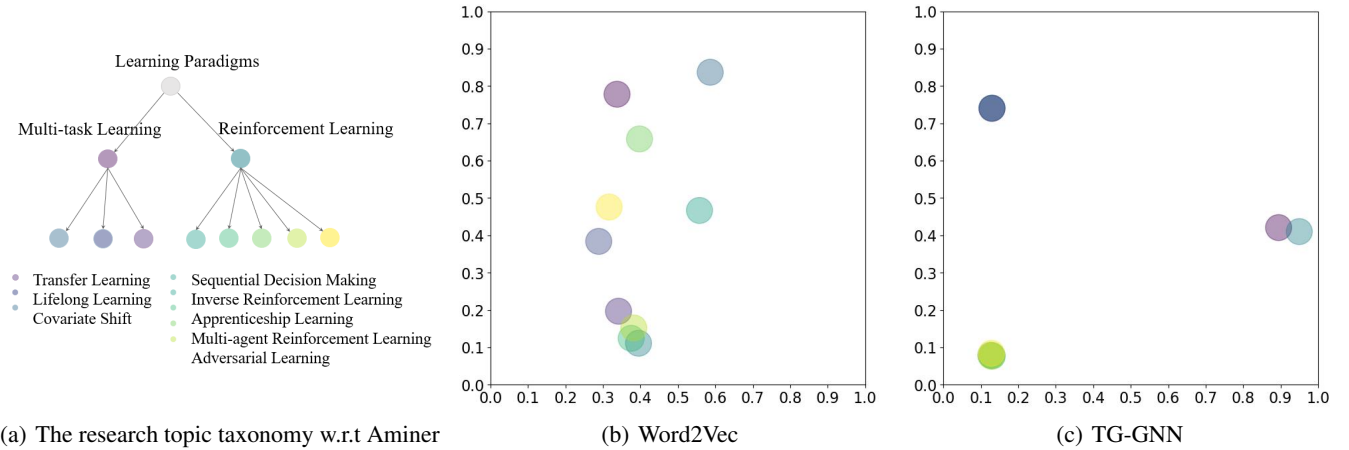


Figure 3: Figure (a) is part of the research topic taxonomy w.r.t Aminer. Figure (b) and Figure (c) are embedding visualizations modeled by Word2Vec and our TG-GNN, respectively.

disorderly around the embedding space. In contrast, the embeddings generated by the position-aware taxonomy learning module result in closer representations for categories within the same branch and more distant representations for categories in separate branches. This demonstrates that our taxonomy representation learning module can effectively acquire hierarchical structural information within the taxonomy.

4.4 Ablation Study of TG-GNN Components

We conducted an ablation study to appreciate the validity and contribution of each component of our new method. We considered three variants of TG-GNN: 1) **TG-GNN-S^{net}** removes the guidance of taxonomy knowledge. Specifically, it modifies the pairwise potential function ψ of the MRF layer by changing S^{taxo} to S^{net} to calculate the similarity between network node attributes instead of the correlation of taxonomy information between nodes. 2) **TG-GNN-P** removes position embeddings when learning taxonomy representations. 3) **TG-GNN-T** removes the entire taxonomy representation learning module and uses the initial taxonomy representation learned by Word2Vec.

We report the results on node classification in Table 3. As shown, TG-GNN outperforms TG-GNN-S^{net} in all cases, suggesting that it is highly preferable to integrate taxonomy

guidance in graph neural networks. Moreover, TG-GNN-P with no position information performs inferior to TG-GNN, indicating that position information helps learn the structure information of the taxonomy tree. Finally, TG-GNN-T does not perform as well as the complete TG-GNN, demonstrating the effectiveness of thoroughly mining the rich semantic and structural information in taxonomy.

4.5 Cold Start Analysis

To assess and demonstrate the generalizability of our new method in the presence of missing data in networks and taxonomies, we designed a cold-start experiment on Aminer and Yelp. We divided each network in Aminer and Yelp into two parts in a 1:1 ratio, i.e., an observed network A and a test network B . We trained the model on network A and tested it on network B , the nodes and edges in test network B are not used for training. For the Aminer networks, all nodes in networks A_{Aminer} and B_{Aminer} have relevant taxonomy. For the Yelp networks, all nodes in an observed network A_{Yelp} have a taxonomy, whereas in a test network B_{Yelp} , more than two-thirds (68%) of the nodes have no taxonomy information. Therefore, the model’s performance on B_{Yelp} is suitable for evaluating the generalizability of our method on networks with network and taxonomy information missing. TG-GNN

Methods	Aminer		Patent		PubMed		Yelp	
	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score
TG-GNN-S ^{net}	88.31	79.63	63.45	63.42	83.24	79.38	86.55	86.33
TG-GNN-P	88.80	80.67	63.12	63.08	84.10	80.62	86.51	86.41
TG-GNN-T	89.04	80.93	62.35	62.27	82.38	78.16	85.59	83.91
TG-GNN	89.13	81.10	63.58	63.55	84.14	80.81	86.67	86.54

Table 3: Ablation studies on four datasets.

outperforms the four baseline methods on these four networks of the two datasets (Fig. 4), showing that our method has better inductive capability on unseen networks.

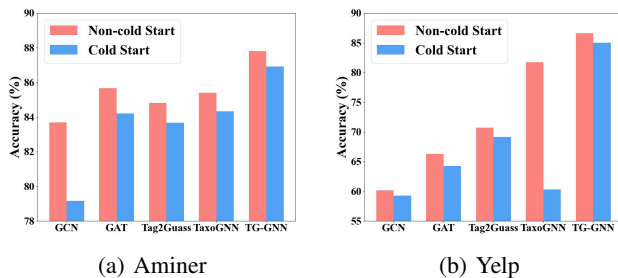


Figure 4: Cold start on Aminer and Yelp.

4.6 Parameter Analysis

We analyzed the impact of the number of MRF layers on model performance. As shown in Fig. 5, as the number of MRF layers increases, the overall performance improves initially but then declines. Therefore, stacking an appropriate number of MRF layers is beneficial for enhancing the guidance of taxonomy knowledge. However, an excessive number of MRF layers may weaken the contribution of taxonomy knowledge. Therefore, considering the balance between model performance and efficiency, we chose to use three MRF layers in TG-GNN.

5 Related Work

Several prior works are related to our research, particularly classical GNNs and taxonomy-guided graph neural networks.

Classical Graph Neural Networks. Graph Neural Networks apply deep neural networks to graph-structured data. GCN [Kipf and Welling, 2017] extends Convolutional Neural Networks to handle graph-structured data by message propagation through graph neighborhoods using graph convolution operations. GraphSAGE [Hamilton *et al.*, 2017] learns node embeddings by sampling and aggregating features from a node’s local neighborhood, especially for optimizing large-scale networks. GAT [Velickovic *et al.*, 2018] utilizes attention mechanisms to differentially weigh the importance of neighboring nodes, enabling selective attention to various parts of the neighborhood during message passing. However, these classical GNNs all fail to consider the implicit network hierarchical structure, inevitably resulting in information distortion on real-world graphs with hierarchical structures.

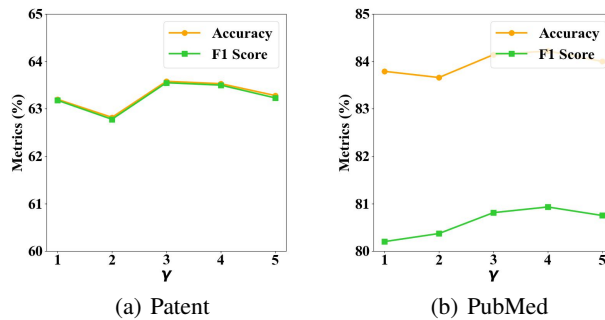


Figure 5: Parameter sensitivity on Patent and PubMed.

Taxonomy-Guided Graph Neural Networks. The concept of taxonomy has been recently introduced to graph neural networks. For example, HGCN [Chami *et al.*, 2019] and Poincaré [Nickel and Kiela, 2017] extend GCNs to incorporate hyperbolic geometry to represent latent hierarchical structures. NetHiex [Ma *et al.*, 2018] divides node embeddings to capture multi-granularity node proximity. However, these approaches ignore network taxonomies, even when they are available, resulting in less interpretable node embeddings. In contrast, TaxoGAN [Yang *et al.*, 2020] introduces taxonomy explicitly into GNNs, which co-embeds nodes and hierarchical labels via a hierarchical network generation process. TaxoGNN [Xu *et al.*, 2022] jointly learns the taxonomy and graph representations, which aligns taxonomy knowledge with the graph through information distillation. Despite the above advancements, these methods neglect the rich semantic and structural information in taxonomy. Moreover, these methods are difficult to generalize and reuse, especially when taxonomy is incomplete.

6 Conclusion

We proposed generalized Taxonomy-Guided Graph Neural Networks (TG-GNN) that integrate taxonomy guidance into network representation learning. Specifically, we constructed position-aware ego networks and used a position-enhanced graph neural network to capture both the rich semantic and structural information in the given taxonomy. We then incorporated a taxonomy-guided Markov mechanism to encapsulate taxonomy knowledge to guide network representation learning. Extensive experimental results and interpretable case studies demonstrated the advantages of TG-GNN.

Acknowledgements

This work is supported by National Key Research and Development Program of China (2023YFC3304503, 2023YFB2603904), National Natural Science Foundation of China (92370111, 62276187, 62272340).

References

- [Bai *et al.*, 2021] Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. A3T-GCN: attention temporal graph convolutional network for traffic forecasting. *International Society for Photogrammetry and Remote Sensing*, 10(7):485, 2021.
- [Bui *et al.*, 2022] Khac-Hoai Nam Bui, Jiho Cho, and Hong-suk Yi. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3):2763–2774, 2022.
- [Chami *et al.*, 2019] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems*, pages 4869–4880, 2019.
- [Dong *et al.*, 2023] Yiqi Dong, Dongxiao He, Xiaobao Wang, Yawen Li, Xiaowen Su, and Di Jin. A generalized deep markov random fields framework for fake news detection. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 4758–4765, 2023.
- [Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 1024–1034, 2017.
- [He *et al.*, 2018] Dongxiao He, Xinxin You, Zhiyong Feng, Di Jin, Xue Yang, and Weixiong Zhang. A network-specific markov random field approach to community detection. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 306–313, 2018.
- [He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.
- [Huang and Carley, 2019] Binxuan Huang and Kathleen M. Carley. Syntax-aware aspect level sentiment classification with graph attention networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5468–5476, 2019.
- [Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2017.
- [Koller and Friedman, 2009] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models - Principles and Techniques*. 2009.
- [Lin *et al.*, 2022] Lu Lin, Ethan Blaser, and Hongning Wang. Graph embedding with hierarchical attentive membership. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining*, pages 582–590, 2022.
- [Liu *et al.*, 2021] Yang Liu, Xiang Ao, Zidi Qin, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Pick and choose: A gnn-based imbalanced learning approach for fraud detection. In *WWW '21: The Web Conference 2021*, pages 3168–3177, 2021.
- [Lu and Li, 2020] Yi-Ju Lu and Cheng-Te Li. GCAN: graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 505–514, 2020.
- [Ma *et al.*, 2018] Jianxin Ma, Peng Cui, Xiao Wang, and Wenwu Zhu. Hierarchical taxonomy aware network embedding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1920–1929, 2018.
- [Mikolov *et al.*, 2013] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations*, 2013.
- [Nickel and Kiela, 2017] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 6338–6347, 2017.
- [Picka, 2006] Jeffrey D. Picka. Gaussian markov random fields: Theory and applications. *Technometrics*, 48(1):146–147, 2006.
- [Qu *et al.*, 2020] Meng Qu, Tianyu Gao, Louis-Pascal A. C. Xhonneux, and Jian Tang. Few-shot relation extraction via bayesian meta-learning on relation graphs. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7867–7876, 2020.
- [Shen *et al.*, 2020] Jiaming Shen, Zhihong Shen, Chenyan Xiong, Chi Wang, Kuansan Wang, and Jiawei Han. Taxoexpan: Self-supervised taxonomy expansion with position-enhanced graph neural network. In *WWW '20: The Web Conference 2020*, pages 486–497, 2020.
- [van den Oord *et al.*, 2018] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [van der Maaten and Hinton, 2008] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.
- [Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations*, 2018.

- [Wang *et al.*, 2019a] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 165–174, 2019.
- [Wang *et al.*, 2019b] Yun Wang, Lun Du, Guojie Song, Xiaojun Ma, Lichen Jin, Wei Lin, and Fei Sun. Tag2gauss: Learning tag representations via gaussian distribution in tagged networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 3799–3805, 2019.
- [Xu *et al.*, 2022] Lingjun Xu, Shiyin Zhang, Guojie Song, Junshan Wang, Tianshu Wu, and Guojun Liu. Taxonomy-enhanced graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2270–2279, 2022.
- [Yang *et al.*, 2020] Carl Yang, Jieyu Zhang, and Jiawei Han. Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks. In *20th IEEE International Conference on Data Mining*, pages 721–730, 2020.
- [Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.