Co-Embedding Network Nodes and Hierarchical Labels with Taxonomy Based Generative Adversarial Networks

Carl Yang, Jieyu Zhang, Jiawei Han
University of Illinois, Urbana Champaign
{jiyang3, jieyuz2, hanj}@illinois.edu

***Abstract*—Network embedding aims at transferring node proximity in networks into distributed vectors, which can be leveraged in various downstream applications. Recent research has shown that nodes in a network can often be organized in latent hierarchical structures, but without a particular underlying taxonomy, the learned node embedding is less useful nor interpretable. In this work, we aim to improve network embedding by modeling the *conditional node proximity* in networks indicated by node labels residing in real taxonomies. In the meantime, we also aim to model the *hierarchical label proximity* in the given taxonomies, which is too coarse by solely looking at the hierarchical topologies. To this end, we propose TAXOGAN to co-embed network nodes and hierarchical labels, through a hierarchical network generation process. Particularly, TAXOGAN models the child labels and network nodes of each parent label in an individual embedding space while learning to transfer network proximity among the spaces of hierarchical labels through stacked network generators and embedding encoders. To enable robust and efficient model inference, we further develop a hierarchical adversarial training process. Comprehensive experiments and case studies on four real-world datasets of networks with hierarchical labels demonstrate the utility of TAXOGAN in improving network embedding on traditional tasks of node classification and link prediction, as well as novel tasks like conditional proximity search and fine-grained taxonomy layout.**

***Index Terms*—conditional network embedding, hierarchical network embedding, generative adversarial networks**

I. INTRODUCTION

Representation learning has become the backbone of various tasks in artificial intelligence [1]. Unsupervised learning is often the default setting due to the desired generalizability. However, many recent works in various fields have demonstrated the profit of leveraging limited label data to learn representations that are not only powerful for the corresponding predictive objectives, but also transferrable to other related tasks [2], [3], [4], [5], [6]. Among them, hierarchical labels residing in given taxonomies have been widely used for natural language processing and bioinformatics, which are especially useful for the tasks of hypernym modeling and hierarchical classification [7], [8], [9], [10], [11], [12], [13]. In their essence, these methods jointly learn the representations of objects and labels in a shared latent space. The objects they model often have rich features, but they do not directly interact with each other.

As for representation learning on networks of interconnected objects (nodes), intensive research has been done on the modeling of both plain networks without node features [14], [15], [16], [17], [18], [19] and content-rich networks with node attributes and/or labels [20], [21], [3], [22], [23]. Recently, the notion of taxonomy has been explored by pioneering research [24], [25], which assume and seek for the latent hierarchical structure underlying the seemingly flatly connected nodes. However, without proper reference to a particular underlying taxonomy, the learned network embedding is still limited to global network mining tasks and uninterpretable without further analysis [26].

Thanks to the vast effort in taxonomy construction from both the research community [27], [28] and industry[1][2][3], increasing amount of network data nowadays can be readily associated with existing taxonomies (Sec. III.1), which provides great opportunities for enhancing network embedding (Sec. III.2) and enabling novel network mining tasks (Sec. III.3). Meanwhile, the rich relational data in networks may also help in better modeling and interpreting the existing taxonomies (Sec. III.4).

Consider a toy example in Figure 1, which consists of an author network (*e.g.*, given by DBLP[4]) and a research topic taxonomy (*e.g.*, given by ACM[5]). Author-author links can be generated *w.r.t.* co-authorships, while author-label links can be generated by keyword matching between the topic names in the taxonomy and the published papers of the authors. In this work, we stress the importance of two novelly observed properties, *i.e.*, *conditional node proximity* and *hierarchical label proximity*.

Conditional node proximity. While existing works on network embedding mostly consider network proximity within the same set of nodes, we argue that node proximity should be conditionally measured within the proper context. For example, on the left side of Figure 1, consider the proximity between C. Faloutsos and J. Kleinberg (particularly, in comparison to that between C. Faloutsos and J. Han). When working on Graph Mining (Graph) problems, C. Faloutsos and J. Kleinberg share more important coauthors like J.

[1]https://feedonomics.com/amazon-category-taxonomy/
[2]https://www.ncbi.nlm.nih.gov/books/NBK21100/
[3]https://wiki.dbpedia.org/services-resources/ontology
[4]https://dblp.uni-trier.de/
[5]https://dl.acm.org/ccs/ccs.cfm

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00081

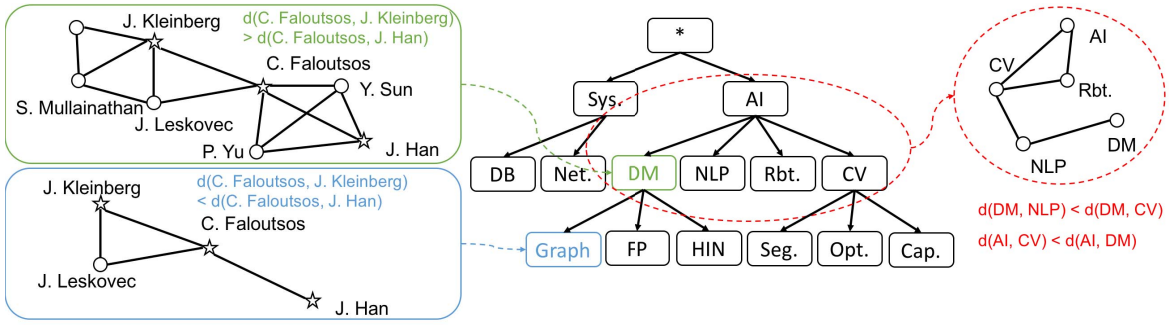


Fig. 1. **Toy example of TAXOGAN: Authors in a publication network are naturally connected to a research topic taxonomy. Through proper modeling of *conditional node proximity* (on the left side) and *hierarchical label proximity* (on the right side), we aim to leverage author node proximity in the network to capture topic label proximity in the taxonomy, which in turn can benefit the learning of both author and topic representations in a closed loop.**

Leskovec, thus resulting in a smaller distance. However, when working on broader problems in Data Mining (DM), they find their own coauthors like S. Mullainathan and J. Han from different fields, hence resulting in a larger distance. As such, under different conditions, node proximity can be rather different and even contradictory.

As we will show in more details in Sec. II, although a given taxonomy naturally allows for the construction of various conditional subnetworks, the modeling of conditional node proximity is non-trivial. This is because modeling all conditional subnetworks separately will prohibit the leverage of node interactions across different subnetworks and suffer from data sparsity, but modeling all conditional subnetworks together in a flat way will lead to a cluttered embedding space violating the hierarchical label relations.

Hierarchical label proximity. Although we assume the existence of given taxonomies for particular networks, where node labels are organized in tree-structured hierarchies, the actual distribution and relative distance of these labels in the embedding space is unknown. For example, consider the four labels CV, NLP, Rbt. and DM on the right side of Figure 1. Although they are all child labels of the parent label AI, the distances among these siblings as well as their distances to AI might be rather different, which is impossible to understand by solely looking at the taxonomy structure itself. In this work, we propose to leverage the rich relational information from the networks to model the fine-grained proximity among the hierarchical labels. Continue with our example. Since authors working on Rbt. may overlap or collaborate more with those working on CV than DM, the distance between Rbt. and CV should be smaller than that between Rbt. and DM. Moreover, compared with authors working on DM, authors working on CV might more often study the core problems of AI. As a consequence, the distance between AI and CV should be smaller than that between AI and DM.

As we will discuss more in Sec. II, proper modeling of the hierarchical label proximity can further help in regularizing the network embedding of all nodes. However, the task is again non-trivial, as the embedding distances in different hierarchical levels should not be modeled in the same space, but how proximity information can be transferred across the different spaces is unclear.

Present work. We propose TAXOGAN to co-embed network nodes and hierarchical labels, which leverages stacked generative adversarial nets to model the conditional node proximity and hierarchical label proximity in networks associated with label taxonomies. Specifically, TAXOGAN models a hierarchical network generation process, where a network generator is devised at each parent label in the taxonomy to model the children network induced by the corresponding child labels and labeled nodes in the original network. Moreover, a learnable network encoder is devised at each child label to enable the learning of proximity transfer from the embedding spaces of children to parents in a fine-to-abstract manner along the actual label paths in the taxonomy. Finally, we device hierarchical adversarial learning to achieve efficient and robust model inference.

The main contributions of this work are:

1) We propose and formulate the novel problem of co-embedding network nodes and hierarchical labels, where we particularly model the two novel important properties of conditional node proximity and hierarchical label proximity (Sec. II.1).
2) We develop TAXOGAN to simultaneously improve network node embedding and enable hierarchical label embedding, by learning to transfer proximity information among different label induced conditional subnetworks according to the taxonomy structure (Sec. II.2-3).
3) We prepare four datasets by linking real-world networks with publicly available taxonomies and conduct thorough experiments regarding two traditional network mining tasks. Significant improvements on hierarchical node classification (11%–70%) and competitive performance on general link prediction compared with the state-of-the-art demonstrate the power of TAXOGAN on improving the quality of network embedding (Sec. III.1-2).

4) We design two novel tasks of *conditional proximity search* and *fine-grained taxonomy layout*, and conduct insightful case studies to demonstrate the unique utility and interpretability of TAXOGAN (Sec. III.3-4).

II. CO-EMBEDDING NETWORK NODES AND HIERARCHICAL LABELS

A. Problem Formulation

Input. We take the input of a network $\mathcal{N} = \{\mathcal{V}, \mathcal{E}, \mathcal{Y}\}$ and a taxonomy $\mathcal{T} = \{\mathcal{L}, \mathcal{H}\}$, where $\mathcal{V} = \{v_i\}_{i=1}^{N}$ is the set of nodes, $\mathcal{E}$ is the set of node-node links, $\mathcal{Y}$ is the set of label-node assignments, $\mathcal{L} = \{l_j\}_{j=1}^{M}$ is the set of labels, and $\mathcal{H}$ is the set of label-label links. For simplicity, we consider uniform undirected node-node links in $\mathcal{E}$, while our model easily generalizes to networks with weighted directed links. By the definition of taxonomy, label-label links in $\mathcal{H}$ are uniform and directed, pointing from parent labels to child labels. Our model works for taxonomies both in tree and DAG structures.

$\mathcal{Y}$ serves as the bridge between $\mathcal{N}$ and $\mathcal{T}$, where for each node $v_i \in \mathcal{V}$, $\mathbf{y}_i$ is the set of labels assigned to v_i. In this work, we require all labels in $\mathbf{y}_i$ to also appear in $\mathcal{L}$, but $\mathbf{y}_i$ can be empty or include any combination of multiple labels. In other words, we only consider node labels organized in a given taxonomy, while we allow the links between the network and the taxonomy to be flexible (likely also weak and noisy). Due to the rapid development of taxonomy construction methods and the growing availability of real-world taxonomies, many networks can be naturally connected with existing taxonomies, which leads to an urge in developing proper models for the joint modeling of both worlds.

Output. To effectively capture the interactions among nodes in the network and labels in the taxonomy, we propose to co-embed $\mathcal{V}$ and $\mathcal{L}$. Therefore, the output of TAXOGAN consists of an $(N \times K)$-dim embedding matrix $\mathbf{U}$ for $\mathcal{V}$ and an $(M \times K)$-dim embedding matrix $\mathbf{Q}$ for $\mathcal{L}$. As we will show later, although we embed $\mathcal{V}$ and $\mathcal{L}$ as the same dimension, their proximity is preserved in different projected spaces, which is necessary for capturing the conditional node proximity under different contexts. Moreover, the projected spaces are connected via learnable transformation functions, which effectively learns to transfer proximities along parent-child label links and captures the hierarchical label proximity.

B. Preliminaries

Heterogeneous graph embedding. A naïve way to jointly model $\mathcal{N}$ and $\mathcal{T}$ is to use a heterogeneous graph, where labels are flattened in the taxonomy. PTE [2] provides a vanilla formula to embed such graphs. In our case, consider nodes $\mathcal{V}$ in $\mathcal{N}$ as *words* with undirected co-occurrence links and labels $\mathcal{L}$ in $\mathcal{T}$ as *documents* connected by directed *citation links*. A heterogeneous graph of $\mathcal{V}$ and $\mathcal{L}$ can be embedded according to the following objective

$$J_{PTE} = J_{vv} + J_{vl} + J_{ll}, \tag{1}$$

where

$$J_{vv} = -\sum_{e_{ij} \in \mathcal{E}} w_{ij} \log \mathcal{G}(v_i, v_j),$$
$$\text{with} \quad \mathcal{G}(v_i, v_j) = \frac{\exp(\mathbf{u}_i'^T \cdot \mathbf{u}_j)}{\sum_{v_k \in \mathcal{V}} \exp(\mathbf{u}_k'^T \cdot \mathbf{u}_j)}; \tag{2}$$
$$J_{vl} = -\sum_{y_{ij} \in \mathcal{Y}} w_{ij} \log \mathcal{G}(v_i, l_j),$$
$$\text{with} \quad \mathcal{G}(v_i, l_j) = \frac{\exp(\mathbf{u}_i'^T \cdot \mathbf{q}_j)}{\sum_{v_k \in \mathcal{V}} \exp(\mathbf{u}_k'^T \cdot \mathbf{q}_j)}; \tag{3}$$
$$J_{ll} = -\sum_{h_{ij} \in \mathcal{H}} w_{ij} \log \mathcal{G}(l_i, l_j),$$
$$\text{with} \quad \mathcal{G}(l_i, l_j) = \frac{\exp(\mathbf{q}_i'^T \cdot \mathbf{q}_j)}{\sum_{l_k \in \mathcal{L}} \exp(\mathbf{q}_k'^T \cdot \mathbf{q}_j)}. \tag{4}$$

Each $\mathcal{G}(o_i, o_j)$ models the probability of generating a linked from object (node/label) o_i to object o_j. Following the setting of Skip-gram adapted to network embedding [14], [15], we use $\mathbf{U}$/$\mathbf{Q}$ as the target embedding and $\mathbf{U}'$/$\mathbf{Q}'$ as the context embedding, which allows explicit modeling of the second-order proximity as proposed in LINE [15].

To optimize Eq. 1, stochastic gradient descent with the techniques of edge sampling and negative sampling [15] can be leveraged. However, the random negative sampling process does not leverage the graph structures at all, which leads to inefficient and unstable training.

Adversarial graph embedding. To enable efficient and robust graph embedding, GraphGAN [16] was proposed based on the concept of adversarial learning [29]. Instead of randomly sampling negative pairs of objects (objects without links), GraphGAN constructs a link discriminator $\mathcal{D}$ and a fake link generator $\mathcal{G}$, and iteratively optimizes the following objective function by allowing $\mathcal{D}$ and $\mathcal{G}$ to play a two-player *minimax* game

$$\min_{\theta_G} \max_{\theta_D} J_{gGAN} = \sum_{v_i \in \mathcal{V}} \Big(\mathbb{E}_{v \sim p_{true}(\cdot, v_i)} \big[\log \mathcal{D}(v, v_i; \theta_D) \big] + \mathbb{E}_{v \sim \mathcal{G}(\cdot, v_i; \theta_G)} \big[\log \big(1 - \mathcal{D}(v, v_i; \theta_D) \big) \big] \Big). \tag{5}$$

Empowered by the novel graph softmax function, $\mathcal{G}$ is able to efficiently generate strong negative samples on-the-fly during training in a graph-structure-aware way [16]. Note that, by sharing the target and context embedding in both $\mathcal{G}$ and $\mathcal{D}$, GraphGAN does not explicitly consider second-order proximity as in PTE and LINE [15]. However, since $\mathcal{G}$ and $\mathcal{D}$ still maintain two sets of embedding, which takes charge of generating and discriminating links respectively, GraphGAN manages to outperform LINE on classic network embedding tasks by significant margins.

In another line of research, complex generative adversarial nets (GAN) have been rapidly developed in domains like computer vision and natural language processing. Particularly, we notice the SGAN model developed for hierarchical image

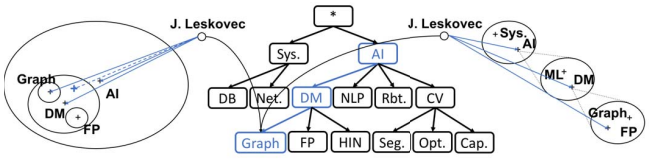


Fig. 2. **Illustration of the main challenges: Modeling network nodes and hierarchical labels all together in a single space leads to a cluttered embedding space violating the underlying hierarchy, while simply partitioning them into separate spaces ignores label correlations and results in parameter redundancy.**

representation learning [30], which consists of a top-down stack of GANs learned to generate high-level to low-level image representations in a hierarchical fashion. While the tasks of image representation and network representation are naturally different, we find essential connections between their task and ours, due to the consideration of underlying hierarchical structures.

C. TaxoGAN

In this work, we aim at co-embedding network nodes and hierarchical labels. To understand the main challenges of this task, let us take a look at Figure 2, where an author node J. Leskovec has three labels Graph, DM and AI in the research topic taxonomy. In this simple case, on one hand, if we do not consider the hierarchical structure of labels and put them all in a *single* space, the author embedding will eventually lie somewhere in the middle of the three label embeddings (as marked by the blue '+' sign), which violates the label hierarchy and results in underfitting. On the other hand, if we simply use *separate* spaces to embed the nodes and labels under each parent label, the model will ignore the rich correlations among labels in the hierarchy, bringing in massive redundant parameters and leading to overfitting.

To address the above challenges, we propose TaxoGAN to co-embed network nodes and hierarchical labels through a hierarchical network generation process, where a network generator is devised at each parent label in the taxonomy to model the subnetwork of nodes and child labels, and a network encoder is devised at each child label to learn the transferrable proximity across levels in the taxonomy. The generator and encoder are jointly trained through efficient and robust hierarchical adversarial learning, where a network discriminator is devised in each embedding space to enforce correct node-node and node-label proximity. In the following, we motivate and describe each component of TaxoGAN in details.

Label-wise subnetwork generator: jointly model node and label proximities in conditional subnetworks. To properly model conditional node proximity and respect the label hierarchy, we propose to generate a specific node-label network under each parent (non-leaf) label in the taxonomy. Let l_p denote an arbitrary parent label in $\mathcal{T}$, and $\mathcal{L}_p$ denote the set of all immediate child labels of l_p. Then $\mathcal{V}_p$ is the subset of $\mathcal{V}$ consisting of all nodes with label l_p or labels in $\mathcal{L}_p$. A *conditional subnetwork* $\mathcal{B}_p$ is constructed from $\mathcal{V}_p$, $\mathcal{L}_p$ as well as the node-node links $\mathcal{E}_p$ among nodes $\mathcal{V}_p$ and node-label links $\mathcal{Y}_p$ between nodes $\mathcal{V}_p$ and labels $\mathcal{L}_p$.

$\mathcal{B}_p$ acts as a bridge between node proximity and label proximity under the condition of l_p. In the corresponding embedding space $\mathcal{S}_p$, $\mathcal{V}_p$ and $\mathcal{L}_p$ can then be arranged in a flat way. To learn the node embedding $\mathbf{U}^p$ and label embedding $\mathbf{Q}^p$ in the space of $\mathcal{S}_p$, we devise a subnetwork generator $\mathcal{G}$ to enforce $\mathcal{E}_p$ and $\mathcal{Y}_p$ based on the softmax function as follows

$$\mathcal{G}(v_j, v_i|l_p) = \frac{\exp(\mathbf{u}_j^{pT} \cdot \mathbf{u}_i^p)}{\sum_{v_k \in \mathcal{V}_p} \exp(\mathbf{u}_k^{pT} \cdot \mathbf{u}_i^p)}, \tag{6}$$

$$\mathcal{G}(l_s, v_i|l_p) = \frac{\exp(\mathbf{q}_s^{pT} \cdot \mathbf{u}_i^p)}{\sum_{l_k \in \mathcal{L}_p} \exp(\mathbf{q}_k^{pT} \cdot \mathbf{u}_i^p)}. \tag{7}$$

Following LINE [15], we can use negative sampling to compute the softmax in Eq. 6, since the number of nodes $|\mathcal{V}_p|$ can be quite large even in the subnetworks. However, since the number of child labels $|\mathcal{L}_p|$ is often quite small, we can directly compute the softmax in Eq. 7 for better label accuracy. Note that, in each conditional subnetwork, there exist no direct links among labels. Thus, the fine-grained relative distances among child labels under each parent label are learned based on the corresponding network structure, which cannot be inferred from the taxonomy structure itself.

Cross-level learnable encoder: proximity transfer and parameter sharing in the taxonomy. The generator $\mathcal{G}$, without the consideration of label correlations and transferrable information in the taxonomy, can either model all conditional subnetworks essentially in a single embedding space or separately in independent spaces. The key difference lies in the computation of $\mathbf{U}^p$ and $\mathbf{Q}^p$. Since in each conditional subnetwork $\mathcal{B}_p$, we co-embed nodes $\mathcal{V}_p$ and labels $\mathcal{L}_p$ in the space $\mathcal{S}_p$, $\mathbf{U}^p$ and $\mathbf{Q}^p$ can be computed from $\mathbf{U}$ and $\mathbf{Q}$ in the same way. Without loss of generality, we will focus our discussion on the computation of $\mathbf{U}^p$.

Particularly, if $\mathbf{U}^p = \mathbf{U}$, which is shared across all conditional subnetworks, all nodes and labels are essentially flatly arranged in a single embedding space of $\mathbf{U}$, which violates the label hierarchy, resulting in clutter embedding space and underfitting. Otherwise, if we compute a completely different $\mathbf{U}^p$ for each conditional subnetwork, the subnetworks are modeled in independent spaces, which ignores label correlations, leading to large parameter redundancy and overfitting.

As a remedy to this trap, we propose to compute each $\mathbf{U}^p$ as an *encoded version* of $\mathbf{U}$, *i.e.*, $\mathbf{U}^p = \mathcal{A}(\mathbf{U}, l_p)$, so as to essentially transfer proximities captured by different subnetwork generators in the taxonomy. However, since the semantic information in taxonomies is coarse, it is hard to decide how to exactly transfer the proximities. For example, consider the sibling labels of NLP and CV under parent AI. Since NLP communities might be *tighter* than CV as including

less diverse subtopics, it should transfer stronger proximity signals. That is, in the subspace of AI, authors close in the subspace of NLP should be closer than those close in the subspace of CV. To capture such subtle semantics in the taxonomy, we require the encoder $\mathcal{A}$ to be *learnable* and *label-dependent*. To this end, we leverage the simple but powerful nonlinear fully connected feedforward neural network (FNN) to model $\mathbf{U}^p$ as

$$\mathbf{U}^p = \mathcal{A}(\mathbf{U}, l_p) = \text{ReLU}(\mathbf{A}_p\mathbf{U}) + \mathbf{b}_p, \tag{8}$$

where $\mathbf{A}_p$ and $\mathbf{b}_p$ are the learnable parameters in the encoder at l_p.

Learning a separate encoder function at each child label does not really leverage the hierarchical structure of $\mathcal{T}$ and still leads to large parameter spaces. To this end, we get motivated by the idea of hierarchical image representation learning [30], which leverages stacked encoders to guide the generation of image representations from high (abstract) to low (detailed) levels. In our scenario, since nodes in the network are connected with labels in the taxonomy, they can also be described by representations at multiple granularities [25]. Therefore, we propose to parameterize $\mathcal{A}$ as *nested embedding transformations* following the hierarchy paths along the taxonomy. For any label l_p, let $l_p \to \ldots \to l_j \to l_i$ denote the path from l_p to a certain leaf label l_i. We have

$$\mathbf{U}^p = \mathcal{A}(\mathbf{U}, l_p) = \mathcal{A}_p(\cdots \mathcal{A}_j(\mathbf{U}, l_j) \cdots, l_p). \tag{9}$$

Note that, the number of parameters in $\mathcal{A}$ grows linearly with the number of labels $|\mathcal{L}|$ in the taxonomy. However, since the main purpose for using $\mathcal{A}$ is to compute multi-granularity node embeddings and separate labels on different levels, it is reasonable to share the parameters of $\mathcal{A}$ among all labels on the same levels of the taxonomy, which reduces the model complexity of $\mathcal{A}$ to $\log|\mathcal{L}|$, and further alleviates possible overfitting due to sparse data in certain subspaces.

Adversarial network discriminator: enable efficient and robust learning. Through subnetwork generation and learnable encoding, we essentially manage to partition the whole network and taxonomy into a set of conditional subnetworks with proper proximity transfer functions. Following the classic heterogeneous network embedding framework of Eq. 1, we formulate the overall objective of TAXOGAN into

$$J_{\text{TAXOGAN}} = J_{vl} + \lambda_1 J_{vv} + \lambda_2 J_{ll}, \tag{10}$$

where each of J_{vv}, J_{vl} and J_{ll} is only slightly different from those in Eq. 1 by replacing the global generator $\mathcal{G}$ with conditional generators and embedding encoders defined in Eq. 6-9.

In practice, we find the joint training of generator networks $\mathcal{G}$ and encoder networks $\mathcal{A}$ to be often inefficient and unstable. Inspired by recent advances in adversarial learning [16], [17], [18], [19], we propose to improve the efficiency and robustness of model inference, by designing a novel hierarchical adversarial network discriminator $\mathcal{D}$. Specifically, each of J_{vv}, J_{vl} and J_{ll} can be optimized through a two-player minimax game defined in Eq. 5, with the corresponding designs of $\mathcal{G}$ and $\mathcal{A}$ defined in Eq. 6-9 and $\mathcal{D}$ defined as follows, which measure the log-probability of node-node and node-label links.

$$\mathcal{D}(v_j, v_i|l_p) = \frac{1}{1+\exp(-\mathbf{u}_j^{pT}\mathbf{u}_i^p)}, \tag{11}$$

$$\mathcal{D}(l_s, v_i|l_p) = \frac{1}{1+\exp(-\mathbf{q}_s^{pT}\mathbf{u}_i^p)}. \tag{12}$$

As illustrated in Figure 3, for each node v_i in $\mathcal{N}$, we consider a bottom-up node encoding process together with a top-down network generation process. $\mathbf{u}_0 = \mathbf{u}$ is the lowest level node embedding, capturing raw node proximity in $\mathcal{N}$. At each parent label l_p in $\mathcal{T}$, the encoder network $\mathcal{A}$ computes a transformed embedding $\mathbf{u}^p$, which ideally can best characterize the embedding of $\mathcal{V}_p$ and $\mathcal{L}_p$ in the conditional embedding space $\mathcal{S}_p$. To achieve this goal, the generator network $\mathcal{G}$ takes $\mathbf{u}_p$ as input and generates the most misleading linked node $\hat{v}_j$ from $\mathcal{V}_p$ and the most relevant label $\hat{l}_s$ from $\mathcal{L}_p$ based on Eq. 6 and 7. The discriminator network $\mathcal{D}$ then tries to differentiate $\hat{v}_j$ and $\hat{l}_s$ from the true linked nodes and labels by maximizing Eq. 10 *w.r.t.* the above equations.

Note that, for stable model training, we find it important for $\mathcal{G}$ and $\mathcal{D}$ to maintain two different sets of node and label embeddings, *i.e.*, $\mathbf{U}'$, $\mathbf{Q}'$ for $\mathcal{G}$ and $\mathbf{U}$, $\mathbf{Q}$ for $\mathcal{D}$, which correspond to the *context embedding* and *target embedding* in [14], [15], respectively. Also note that, since we partition the whole network into series of subnetworks, some node-node links across different subnetworks cannot be directly modeled, but they nonetheless carry important proximity information. To deal with this, we add a global node-node proximity module on the base embedding of nodes $\mathcal{U}$, which is implemented by exactly following [16].

Algorithm 1 TAXOGAN Training

1: **procedure** TAXOGAN-TRAIN
2: **Input:** network $\mathcal{N}$, taxonomy $\mathcal{T}$, embedding dimension K, #batches b_{vl}, b_{vv}, b_{ll}, batch size s, negative sampling rate n
3: **while** not converge **do**
4: Sample a parent label l_p and construct the subnetwork $\mathcal{B}_p$
5: **for** $t \leftarrow 1$ to b_{vv} **do**
6: Update $\mathbf{U}'^p$ and $\mathbf{U}^p$ by training $\mathcal{G}_{vv}$, $\mathcal{D}_{vv}$, $\mathcal{A}$
7: **end for**
8: **for** $t \leftarrow 1$ to b_{ll} **do**
9: Update $\mathbf{Q}'^p$ and $\mathbf{Q}^p$ by training $\mathcal{G}_{ll}$, $\mathcal{D}_{ll}$, $\mathcal{A}$
10: **end for**
11: **for** $t \leftarrow 1$ to b_{vl} **do**
12: Update $\mathbf{U}'^p$, $\mathbf{U}^p$, $\mathbf{Q}'^p$, $\mathbf{Q}^p$ by training $\mathcal{G}_{vl}$, $\mathcal{D}_{vl}$, $\mathcal{A}$
13: **end for**
14: **end while**
15: **return** $\mathbf{U}$, $\mathbf{Q}$ and $\mathcal{A}$
16: **end procedure**

Training algorithm. Finally, with the neural architectures of generator $\mathcal{G}$, discriminator $\mathcal{D}$ and encoder $\mathcal{A}$ defined, we describe the detailed joint training process of TAXOGAN in Algorithm 1.

In algorithm 1, in Line 6, the design and training of $\mathcal{G}_{vv}$ and $\mathcal{D}_{vv}$ in each subnetwork is the same as in the plain networks

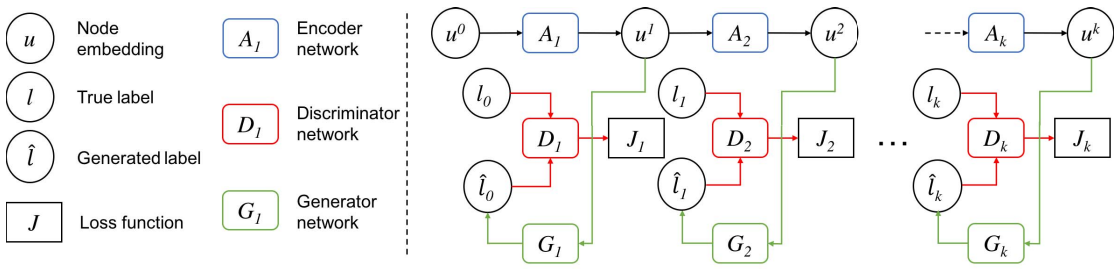


Fig. 3. **TAXOGAN overview: A framework for the adversarial learning of hierarchical network embedding.**

of [16]; in Line 9, the training of $\mathcal{G}_{ll}$ and $\mathcal{D}_{ll}$ are very similar to those of $\mathcal{G}_{vv}$ and $\mathcal{D}_{vv}$, only by substituting $\mathbf{U}$ with $\mathbf{Q}$, and $\mathcal{A}$ is shared for $\mathbf{U}/\mathbf{U}'$ and $\mathbf{Q}/\mathbf{Q}'$. Since the sampling of v and l is discrete, all generator networks are trained by policy gradient [31]. For example, the gradient of J_{vl} conditioned on l_p *w.r.t.* $\mathcal{G}$ is computed as

$$\begin{aligned}
&\nabla_{\mathbf{U}',\mathbf{Q}'} J_{vl}|l_p \\
=&\nabla_{\mathbf{U}',\mathbf{Q}'} \sum_{l_j \in \mathcal{L}_p} \mathbb{E}_{l_j \sim \mathcal{G}(\cdot,v_i|l_p)}[\log(1-\mathcal{D}(l_j,v_i|l_p)] \\
=&\sum_{l_j \in \mathcal{L}_p} \mathbb{E}_{l_j \sim G(\cdot,v_i|l_p)} \\
&[\nabla_{\mathbf{U}',\mathbf{Q}'} \log \mathcal{G}(l_j,v_i|l_p) \log(1-\mathcal{D}(l_j,v_i|l_p))].
\end{aligned}$$

Training the generator networks $\mathcal{G}$ results in the update of $\mathbf{U}'$ and $\mathbf{Q}'$, while training the discriminator networks $\mathcal{D}$ results in the update of $\mathbf{U}$ and $\mathbf{Q}$. For stability concern, we fix $\mathcal{A}$ during the training of $\mathcal{G}$, and only update it while training $\mathcal{D}$.

In each iteration, the complexity of Line 5-7 is $O(b_{vv}snd_NK)$, Line 8-10 is $O(b_{ll}snd_TK)$, Line 11-13 is $O(b_{vl}snd_TK)$, where d_N is the average node degree in $\mathcal{N}$ and d_T is the average number of child labels of each non-leaf label in $\mathcal{T}$. b_{vv}, b_{ll} and b_{vl} are set to balance the trade-off among the three objectives and reflect the weighing parameters λ_1 and λ_2 in Eq. 10. Considering convergence to be reached after a constant number of iterations over all nodes, the overall complexity of TAXOGAN is bounded by the $N \log N$ complexity of global network embedding same as [16].

We implement TAXOGAN with Pytorch. As we can observe from the experimental results, the variance across different trains of TAXOGAN on the same data is not large. We further inspect the loss curves and conclude that the training process of TAXOGAN is stable. The code is published on GitHub[6].

III. EXPERIMENTS

A. Experimental Settings

Datasets. We construct four datasets of real-world networks with explicit taxonomies.

- **DBLP:** We collect the author network[7] with the research topic taxonomy[8]. Undirected uniform links in the network are generated based on coauthorships. A label in the taxonomy is assigned to an author if her/his papers mentions the keyword.
- **Yelp:** We collect the business network[9] with the category taxonomy[10]. Undirected uniform links in the network are generated based on common customers who posted reviews for both businesses. Label assignments are given in the original dataset.
- **FreeBase:** We collect the entity network[11] with the type taxonomy[12]. Undirected uniform links in the network are generated if two entities appear together in any triplet of facts. Labels are assigned by retrieving the nested entity types.
- **PubMed:** We collect the protein network[13] with the disease taxonomy[14]. Undirected uniform links in the network are generated if mentions of two proteins appear in any same sentence. Labels are assigned by surface name matching.

Datasets	Network		Taxonomy	
	#nodes	#links	#labels	#levels
DBLP	81,389	208,711	268	4
Yelp	14,573	55,243	438	4
FreeBase	30,180	53,632	18	3
PubMed	9,619	25,655	87	2

TABLE I
STATISTICS OF THE FOUR REAL-WORLD DATASETS WE USE.

Compared algorithms. We compare with three groups of network embedding algorithms from the state-of-the-art to comprehensively evaluate the performance of TAXOGAN.

- *Plain network embedding:* We compare with DeepWalk [14] and GraphGAN [16]. DeepWalk is the most pioneering and popular Skip-gram based network embedding algorithm, while GraphGAN represents the state-of-the-art plain network embedding models leveraging adversarial learning. We run both algorithms on the original networks by ignoring the taxonomies.
- *Attributed and labeled network embedding:* We compare with PTE [2] and GraphSage [32]. PTE is an extension of

[6]https://github.com/JieyuZ2/TaxoGAN
[7]https://dblp.uni-trier.de/xml/
[8]https://dl.acm.org/ccs/ccs_flat.cfm
[9]https://www.yelp.com/dataset
[10]https://www.yelp.com/developers/documentation/v3/all_category_list
[11]http://freebase-easy.cs.uni-freiburg.de/dump/
[12]http://dbpedia.org/page/Taxonomy
[13]ftp://ftp.ncbi.nih.gov/pub/taxonomy
[14]ftp://ftp.ncbi.nlm.nih.gov/

the popular LINE [15] algorithm to networks with attributes and labels. We treat taxonomies as flat label networks, and run PTE on the bipartite networks of nodes and labels. GraphSage represents the state-of-the-art attributed and labeled network embedding models. We regard all labels as flat node attributes and train GraphSage in the link prediction fashion.

- *Taxonomy aware network embedding:* We compare with Poincare [24] and Nethiex [25], which are the most recent network embedding algorithms assuming latent node taxonomies. Since they do not work with explicit taxonomies, we run both of them on the original networks as in their original settings.

We also conduct comprehensive ablation study by comparing four different TAXOGAN variants: (1) TAXOGAN-sin is the model with a single embedding space; (2) TAXOGAN-sep is the model with separate embedding spaces; (3) TAXOGAN-noadv is the model without adversarial training; (4) TAXOGAN is our full model.

Evaluation protocols. We evaluate all algorithms on two fundamental tasks: node classification and link prediction.

For node classification, since we consider hierarchical labels in taxonomies in this work, we focus on the setting of level-by-level classification. Given the learned embedding of training nodes and the label taxonomy, we further train a linear SVM at each parent label to classify the testing node *w.r.t.* the current child labels. During testing, each node thus can be assigned to a path in the label taxonomy, a testing node-label pair (v, l) is correct if the predicted label path of v includes l. All TAXOGAN models except for TAXOGAN-sin use the corresponding embeddings in each level, while the other models all use a single embedding across all levels. We randomly split the set of labeled nodes into training and testing sets with the ratio of 4:1 for five times and compute the testing F1 of each node-label pair. We aggregate the pair-wise F1 scores by each node to compute the micro F1 and by each label to compute the macro F1.

We consider standard link prediction in the same way as in [14], [15]. Predicted links are ranked by the cosine distance among the node embedding vectors. All TAXOGAN models use the shared base embedding $\mathcal{U}$ for link prediction. We randomly split the set of all links in the network into training and testing sets with the ratio of 4:1 for five times and compute the standard AUC and MRR scores.

Parameter settings. The implementations of all compared algorithms are provided by their original authors, and all model hyper-parameters are tuned to the best via standard five-fold cross validation. For TAXOGAN, we use the same parameters for all datasets. After a rough grid search, we empirically set the loss weighing parameters λ_1 and λ_2 to 0.1, embedding dimension τ to 50, batch size s to 64 and learning rate to 10^{-4}. All batch numbers b's are set to 128 and negative sampling rate n is set to 5.

B. Quantitative Evaluations

Table II presents the performance of compared algorithms on hierarchical node classification. The improvements of TAXOGAN over the second runners all passed the significance t-test with p-value 0.01. Since the classification at each level in the label taxonomy is multi-class, and deeper labels are harder to be correctly predicted (if any precedent label is predicted wrong, the label path can never reach the correct label), the absolute F1 values are all pretty low. Dataset like Yelp has a lot of deep but narrow labels, which are hard to correctly predict, and the mistakes largely impact the macro F1, whereas dataset like PubMed has a lot of shallow but wide labels, and the mistakes largely impact the micro F1. Thus the suite of datasets and metrics provides a comprehensive evaluation towards the compared algorithms.

The baselines have varying performance across different datasets, while PTE and GraphSage often perform better due to the leverage of labeled data during training. By considering latent hierarchies, Poincare and Nethiex perform better than DeepWalk and GraphGAN in many cases, but their learned latent hierarchies do not always perfectly match the reality and even lead to worse performance in some cases like on DBLP.

Overall, TAXOGAN constantly outperforms all compared algorithms in all cases, with significant margins over the best baseline ranging from 11% to 70%, and the scores all passed t-test with p-value 0.05, demonstrating its superior effectiveness and generalizability. In particular, the improvements of TAXOGAN are more significant when the numbers of labels are larger and the hierarchies of labels are deeper, like with DBLP and Yelp, which supports the appropriate design of our model to leverage the explicit hierarchical structure of associative labels. Note that, while the unsupervised baselines (DeepWalk, GraphGAN, Poincare and Nethiex) do not have access to the node labels in the taxonomy, PTE and GraphSage use the exact same labels as TAXOGAN. This shows TAXOGAN to be effective in modeling hierarchical label spaces, as we will further demonstrate in the ablation study.

For ablation study, our TAXOGAN-sin model has close performance towards the best baselines like PTE, because they are indeed similar only by the difference in adversarial training; our TAXOGAN-sep model does not always outperform TAXOGAN-sin, indicating that even if the evaluation protocol of level-by-level classification may favor multiple embeddings, simply using separate embeddings is not good enough and can harm the performance due to problems like subnetwork sparsity and overfitting, and TAXOGAN-sep is extremely hard to train due to redundant parameters and large memory cost; our TAXOGAN-noadv model is the nested space model without adversarial training, which outperforms TAXOGAN-sep with significant margins, corroborating the effectiveness of our model design with connected subspaces through base and transformed embeddings; our TAXOGAN model further outperforms TAXOGAN-noadv, directly showing the advantage of our novel hierarchical adversarial training technique.

Model	Micro F1				Macro F1			
	DBLP	Yelp	FreeBase	PubMed	DBLP	Yelp	FreeBase	PubMed
DeepWalk	11.07 ± 0.61	26.24 ± 0.84	26.41 ± 1.12	10.94 ± 1.06	13.11 ± 0.81	4.54 ± 0.97	28.11 ± 1.06	39.37 ± 0.36
GraphGAN	16.10 ± 0.55	26.40 ± 1.21	25.97 ± 0.85	13.68 ± 1.28	16.19 ± 0.71	4.90 ± 1.04	26.65 ± 0.43	40.35 ± 0.44
PTE	16.42 ± 0.47	33.73 ± 0.93	50.27 ± 1.40	12.71 ± 1.64	18.61 ± 0.67	5.47 ± 0.39	28.19 ± 0.31	40.74 ± 0.87
GraphSage	18.72 ± 1.18	29.06 ± 0.29	45.77 ± 0.60	12.05 ± 1.17	16.65 ± 0.72	9.43 ± 1.03	24.06 ± 0.90	36.39 ± 1.09
Poincare	13.87 ± 0.51	29.02 ± 1.12	30.43 ± 1.29	12.73 ± 1.90	18.49 ± 0.51	4.25 ± 1.08	28.57 ± 0.37	40.09 ± 0.33
Nethiex	10.06 ± 0.56	19.44 ± 1.53	35.39 ± 1.37	12.22 ± 1.31	13.86 ± 0.54	4.06 ± 1.03	24.75 ± 0.52	40.83 ± 0.78
TAXOGAN-sin	20.56 ± 0.25	34.88 ± 0.42	65.36 ± 0.59	11.81 ± 1.13	30.44 ± 0.63	13.33 ± 0.39	32.21 ± 0.32	39.86 ± 0.63
TAXOGAN-sep	25.80 ± 1.01	28.47 ± 1.04	63.46 ± 0.46	11.98 ± 0.42	33.37 ± 0.45	11.63 ± 0.95	29.41 ± 0.98	39.86 ± 0.74
TAXOGAN-noadv	29.52 ± 0.79	39.83 ± 1.09	65.79 ± 1.07	16.31 ± 0.22	30.13 ± 0.62	12.74 ± 0.93	31.55 ± 0.62	40.05 ± 0.98
TAXOGAN	**31.97 ± 1.44**	**41.37 ± 0.58**	**65.98 ± 0.98**	**20.11 ± 1.41**	**36.42 ± 0.57**	**15.19 ± 0.72**	**36.62 ± 0.95**	**40.89 ± 0.63**

TABLE II
PERFORMANCE OF ALL COMPARED ALGORITHMS ON HIERARCHICAL NODE CLASSIFICATION.

Model	AUC				MRR			
	DBLP	Yelp	FreeBase	PubMed	DBLP	Yelp	FreeBase	PubMed
DeepWalk	83.40 ± 0.26	87.93 ± 0.43	64.93 ± 0.35	69.15 ± 1.18	81.41 ± 0.66	68.46 ± 0.57	63.07 ± 0.98	40.37 ± 0.38
GraphGAN	83.76 ± 0.09	88.51 ± 0.28	65.00 ± 0.67	68.19 ± 1.31	81.03 ± 0.59	67.81 ± 0.41	63.03 ± 0.63	41.35 ± 0.44
PTE	75.47 ± 0.15	89.10 ± 0.26	63.16 ± 0.52	71.46 ± 0.86	77.77 ± 0.56	68.85 ± 0.25	62.75 ± 0.41	**42.74 ± 0.87**
GraphSage	82.63 ± 0.22	85.33 ± 0.56	66.53 ± 0.51	68.20 ± 1.21	76.47 ± 0.34	62.91 ± 0.15	62.65 ± 0.11	36.39 ± 1.09
Poincare	84.06 ± 0.15	91.60 ± 0.16	68.86 ± 0.35	71.68 ± 0.80	81.79 ± 0.48	68.51 ± 0.62	63.11 ± 0.12	41.09 ± 0.33
Nethiex	84.41 ± 0.07	92.70 ± 0.26	69.75 ± 0.68	**71.78 ± 0.28**	81.13 ± 0.46	69.16 ± 0.54	63.05 ± 0.35	40.83 ± 0.78
TAXOGAN-sin	84.14 ± 0.06	92.31 ± 0.31	67.14 ± 0.41	68.00 ± 0.74	79.99 ± 0.46	69.59 ± 0.71	63.64 ± 0.97	41.89 ± 1.11
TAXOGAN-sep	84.17 ± 0.14	87.47 ± 0.34	63.29 ± 0.65	68.60 ± 0.64	80.84 ± 0.38	68.36 ± 0.95	62.65 ± 0.68	40.96 ± 0.74
TAXOGAN-noadv	84.56 ± 0.15	92.22 ± 0.39	66.73 ± 0.66	68.95 ± 0.33	80.27 ± 0.26	69.31 ± 0.41	63.34 ± 0.59	40.05 ± 0.98
TAXOGAN	**85.02 ± 0.25**	**92.92 ± 0.44**	**70.48 ± 0.32**	70.02 ± 1.03	**82.32 ± 0.28**	**69.70 ± 0.57**	**64.33 ± 0.49**	42.05 ± 0.98

TABLE III
PERFORMANCE OF ALL COMPARED ALGORITHMS ON STANDARD LINK PREDICTION.

Table III presents the performance of compared algorithms on standard link prediction. Note that, the main goal of TAXOGAN is hierarchical node classification by design, where we leverage network structures to compute the node embeddings as inputs of the hierarchical classifiers. As a result, the base embeddings that we use for the link prediction experiments are mostly decided by the plain network structures and only get slightly influenced during the training of the hierarchical GAN model. Nonetheless, such fine tuning *w.r.t.* hierarchically structured labels is shown to be useful for global (unconditional) link prediction, which leads to very competitive performance compared to the strongest baselines, further corroborating the general utility of TAXOGAN. It is reasonable to expect TAXOGAN to further excel on datasets where links are also generated under different conditions.

We measure the runtimes of all compared algorithms on a server with one GeForce GTX TITAN X GPU and two Intel Xeon E5-2650V3 10-core 2.3GHz CPUs. We observe the runtimes of TAXOGAN to be similar to GraphGAN and GraphSage, while slightly larger than other baselines like PTE and DeepWalk.

C. Conditional Proximity Search

To illustrate how TAXOGAN is able to capture both global and conditional proximity on networks with hierarchical labels, we select four of the many well-known researchers from different fields related to data mining and extract their hierarchical embeddings computed by TAXOGAN on DBLP.

In Table IV, for each author pair, we present their predicted proximity based on the cosine similarity between their global base embeddings as well as the topic-wise conditional embeddings of the top five research topics where the pair is embedded most closely. As we can observe, (1) the global proximity computed by TAXOGAN reflect the reality, where authors working on more similar topics in general are embedded closer. Meanwhile, (2) the conditional proximities are even more accurate and telling, since they provide essential insights into which particular research topics a given pair of authors are likely to collaborate on. Such knowledge, while directly facilitating the unique application of conditional proximity search as we advocate in this work, is hard to gather without proper joint modeling of the network and associated taxonomy.

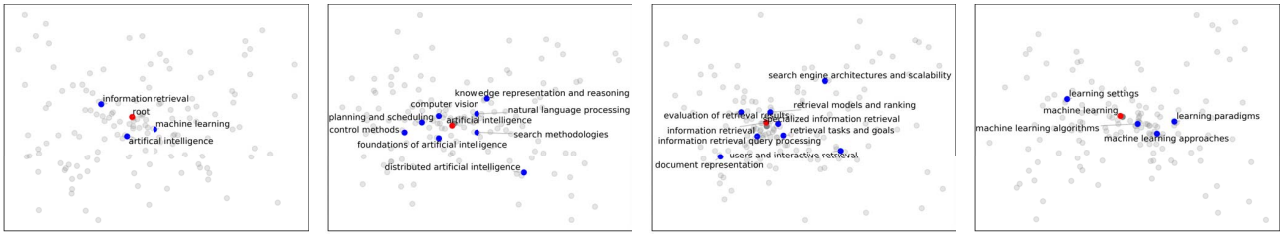
D. Fine-Grained Taxonomy Visualization

Another novel application of TAXOGAN is fine-grained taxonomy visualization, which is enabled by our unique leverage of node proximity in networks associated with the taxonomies. As an example, we visualize the embedding spaces (Figure 4, reduced to 2-dim by standard PCA) of four label-induced subnetworks from DBLP, corresponding to the labels root, AI, IR, and ML. Grey dots are nodes in the conditional subnetworks, while red and blue dots are the parent and child labels, respectively. Since many labels have quite similar textual names, such fine-grained label representations are hard to generate by existing methods like word embedding.

As we can observe, the results are highly interpretable and insightful, which provide knowledge about the relative distances among labels. For example, in the AI subnetwork, labels closest to AI include CV, NLP and foundations of AI, while the closest pairs of labels include knowledge rep.–NLP, CV–planning, planning–control, *etc*. While existing taxonomies mostly only include a label skeleton, such label

Jiawei Han & Christos Faloutsos	Jiawei Han & Jure Leskovec	Jiawei Han & Yoshua Bengio
global (0.8772)	global (0.7401)	global (0.6558)
knowledge rep. and reasoning (0.8983)	*information system applications* (0.8238)	*retrieval tasks and goals* (0.6174)
collaborative and social computing (0.8884)	*machine learning approaches* (0.7671)	*machine learning approaches* (0.5717)
data mining (0.8579)	*collaborative and social computing* (0.7019)	*foundations of AI* (0.5634)
information system applications (0.7960)	*users and interactive retrieval* (0.6784)	*document representation* (0.5328)
spatial-temporal systems (0.7280)	*knowledge rep. and reasoning* (0.5381)	*scheduling and planning* (0.5079)
Christos Faloutsos & Jure Leskovec	Christos Faloutsos & Yoshua Bengio	Jure Leskovec & Yoshua Bengio
global (0.8883)	global (0.7939)	global (0.7710)
collaborative and social computing (0.9229)	*foundations of AI* (0.7468)	*enterprise information systems* (0.7632)
specialized information retrieval (0.8864)	*enterprise information systems* (0.7020)	*foundations of AI* (0.7124)
information system applications (0.8664)	*collaborative and social computing* (0.6569)	*machine learning approaches* (0.6846)
search methodologies (0.8624)	*retrieval models and ranking* (0.5428)	*planning and scheduling* (0.6437)
machine learning (0.7989)	*computer vision* (0.4958)	*search methodologies* (0.6128)

TABLE IV
PAIR-WISE GLOBAL AND CONDITIONAL SIMILARITY AMONG FOUR RESEARCHERS JOINTLY LEARNED BY TAXOGAN.



(a) Base Embedding Space (b) Artificial Intelligence (c) Information Retrieval (d) Machine Learning

Fig. 4. Visualization of the hierarchical label spaces learned by TAXOGAN given the network and label taxonomy (zoom in for clear view).

embeddings are valuable towards the understanding of subtle label relations, and likely useful for more downstream tasks like taxonomy refinement and others involving machine learning on taxonomies.

IV. RELATED WORK

Network Embedding. Network embedding, particularly node embedding, aims to transform node proximity into embedding distance in low-dimensional vector space. Along with the recent success of neural networks, many powerful models have been developed for network embedding, such as the graph context preserving models based on Skip-gram [14], [15], [16], [17], [18], [19]. They are mostly developed for plain networks, where the learned representations capture network proximity among nodes up to certain orders.

On top of them, more recent works aim to stress auxiliary information associated with networks such as attributes, labels, and communities. Most of them can be arranged into two groups. The first models the embedding process as semi-supervised or multi-modal learning, by jointly utilizing the network proximity while optimizing particularly designed auxiliary task objectives, such as attribute prediction [20], [22], [21], classification [33], [23], [3], [34], clustering [35], [36], [37], *etc*. The second group reorganizes the networks based on auxiliary data into particular structures such as multi-view networks [38] and heterogeneous networks [39]. They model and weigh multiple measures of proximity among the same set of nodes. Different from them, we leverage the auxiliary data of hierarchical labels and combine the ideas behind both groups to formulate multiple optimization objectives.

Until very recently, the notion of taxonomy has been brought to network embedding. With the assumption that network nodes can be organized in an underlying taxonomy, [24] proposed to preserve node proximity on the tree-structured taxonomy in a hyperbolic space for parsimonious representations, while [25] proposed to partition the embedding vectors into segments to model multi-granularity node proximity. Both methods are shown to be advantageous for improving the quality of general unsupervised network embedding. Different from them, we leverage the knowledge from existing taxonomies, which provide additional opportunities to further improve the embedding quality, while naturally enabling more novel applications with valuable interpretability. During the writing of this paper, we notice a recent work sharing similar spirits with us by jointly embedding KB instances and ontology concepts [40]. However, they only compute two embedding spaces by ignoring subtler conditional proximities and their KB embedding models do not work in the network embedding setting as we consider.

Taxonomy Modeling. Taxonomy has attracted tremendous attention from both research community and industry, due to its fundamental utility in various real-world applications [28], [27]. In industry, enterprises have been manually constructing large taxonomies of products, services and so on for decades. In academia, while most existing research focuses hypernym-hyponym pair extraction through lexical patterns [41], [42] or supervised classification [10], [11], recent works also construct topic taxonomies based on hierarchical clustering [27], [43]. As closest to us, [44] jointly performs clustering and ranking on text-rich networks to extract hierarchical topic structures. However, existing works on taxonomy construction seldom consider the relative proximity from children to parents and the varying proximity among siblings.

Another line of research related to taxonomy is hierarchical classification [13], [12], [7]. To correctly classify objects to paths of hierarchical labels on a taxonomy, the models usually need to implicitly capture the distributions of child classes under parent class. Some recent works based on word embedding techniques also aim to capture the fine-grained hierarchical relations among parent and child classes through complex neural networks [8], [9]. However, since their ultimate goal is classification, the implicitly captured label distributions are not readily useful and interpretable.

V. Conclusion

To the best of our knowledge, we are the first to jointly model networks and taxonomies. By stressing the important properties of conditional node proximity and hierarchical label proximity, we develop TaxoGAN, which computes high-quality network embedding under the guidance of hierarchical labels, while in turn produce fine-grained label embedding. Extensive experimental results and interpretable case studies demonstrate the advantages of TaxoGAN in both traditional network mining tasks and unique novel applications.

Acknowledgement

Research was sponsored in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and SocialSim Program No. W911NF-17-C-0099, National Science Foundation IIS-19-56151, IIS-17-41317, IIS 17-04532, and IIS 16-18481, and DTRA HDTRA11810026. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *TPAMI*, vol. 35, no. 8, pp. 1798–1828, 2013.
[2] J. Tang, M. Qu, and Q. Mei, "Pte: Predictive text embedding through large-scale heterogeneous text networks," in *KDD*, 2015.
[3] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.
[4] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *TPAMI*, vol. 38, no. 7, pp. 1425–1438, 2016.
[5] Y. Ma, E. Cambria, and S. Gao, "Label embedding for zero-shot fine-grained named entity typing," in *COLING*, 2016.
[6] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, R. Henao, and L. Carin, "Joint embedding of words and labels for text classification," in *ACL*, 2018.
[7] M. Alsuhaibani, T. Maehara, and D. Bollegala, "Joint learning of hierarchical word embeddings from a corpus and a taxonomy," in *AKBC*, 2018.
[8] K. A. Nguyen, M. Köper, S. S. im Walde, and N. T. Vu, "Hierarchical embeddings for hypernymy detection and directionality," in *EMNLP*, 2017.
[9] I. Vulić and N. Mrkšić, "Specialising word vectors for lexical entailment," in *ACL*, 2018.
[10] T. L. Anh, Y. Tay, S. C. Hui, and S. K. Ng, "Learning term embeddings for taxonomic relation identification using dynamic weighting neural network," in *EMNLP*, 2016.
[11] Z. Yu, H. Wang, X. Lin, and M. Wang, "Learning term embeddings for hypernymy identification," in *IJCAI*, 2015.
[12] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *WWW*, 2018.
[13] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *ICML*, 2018.
[14] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *KDD*, 2014.
[15] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *WWW*, 2015.
[16] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "Graphgan: Graph representation learning with generative adversarial nets," in *AAAI*, 2018.
[17] W. Yu, C. Zheng, W. Cheng, C. C. Aggarwal, D. Song, B. Zong, H. Chen, and W. Wang, "Learning deep network representations with adversarially regularized autoencoders," in *KDD*, 2018.
[18] Q. Dai, X. Shen, L. Zhang, Q. Li, and D. Wang, "Adversarial training methods for network embedding," in *WWW*, 2019.
[19] H. Gao, J. Pei, and H. Huang, "Progan: Network embedding via proximity generative adversarial network," in *KDD*, 2019.
[20] Z. Meng, S. Liang, H. Bao, and X. Zhang, "Co-embedding attributed networks," in *WSDM*, 2019.
[21] X. Huang, J. Li, and X. Hu, "Accelerated attributed network embedding," in *SDM*, 2017.
[22] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "Anrl: Attributed network representation learning via deep neural networks." in *IJCAI*, 2018.
[23] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding," in *WSDM*, 2017.
[24] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *NIPS*, 2017.
[25] J. Ma, P. Cui, X. Wang, and W. Zhu, "Hierarchical taxonomy aware network embedding," in *KDD*, 2018.
[26] N. Liu, X. Huang, J. Li, and X. Hu, "On interpretation of network embedding via taxonomy induction," in *KDD*, 2018.
[27] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, and J. Han, "Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering," in *KDD*, 2018.
[28] J. Park, B. J. Hescott, and D. K. Slonim, "Towards a more molecular taxonomy of disease," *Journal of Biomedical Semantics*, vol. 8, no. 1, p. 25, 2017.
[29] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *arXiv preprint arXiv:2001.06937*, 2020.
[30] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, "Stacked generative adversarial networks," in *CVPR*, 2017.
[31] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *AAAI*, 2017.
[32] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017.
[33] Z. Yang, W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings," in *ICML*, 2016.
[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.
[35] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*, 2016.
[36] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *NIPS*, 2017.
[37] Y. Jia, Q. Zhang, W. Zhang, and X. Wang, "Communitygan: Community detection with generative adversarial nets," in *WWW*, 2019.
[38] M. Qu, J. Tang, J. Shang, X. Ren, M. Zhang, and J. Han, "An attention-based collaboration framework for multi-view network representation learning," in *CIKM*, 2017.
[39] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *KDD*, 2017.
[40] J. Hao, M. Chen, W. Yu, Y. Sun, and W. Wang, "Universal representationlearning of knowledge bases by jointly embedding instances and ontological concepts." KDD, 2019.
[41] M. Jiang, J. Shang, T. Cassidy, X. Ren, L. M. Kaplan, T. P. Hanratty, and J. Han, "Metapad: Meta pattern discovery from massive text corpora," in *KDD*, 2017.
[42] N. Nakashole, G. Weikum, and F. Suchanek, "Patty: a taxonomy of relational patterns with semantic types," in *EMNLP*, 2012.
[43] D. Downey, C. Bhagavatula, and Y. Yang, "Efficient methods for inferring large sparse topic hierarchies," in *ACL*, 2015.
[44] C. Wang, J. Liu, N. Desai, M. Danilevsky, and J. Han, "Constructing topical hierarchies in heterogeneous information networks," *Knowledge and Information Systems*, vol. 44, no. 3, pp. 529–558, 2015.