HGDL: Heterogeneous Graph Label Distribution Learning

Yufei Jin[†], **Heng Lian**[◇], **Yi He**[◇], **Xingquan Zhu**[†*]
[†]Dept. of Elec. Eng. & Computer Sci., Florida Atlantic University, Boca Raton, FL 33431, USA
[◇]Dept. of Data Science, William & Mary, Williamsburg, VA 23185, USA
yjin2021@fau.edu; hlian01@wm.edu; yihe@wm.edu; xzhu3@fau.edu

Abstract

Label Distribution Learning (LDL) has been extensively studied in IID data applications such as computer vision, thanks to its more generic setting over single-label and multi-label classification. This paper advances LDL into graph domains and aims to tackle a novel and fundamental *heterogeneous graph label distribution learning* (HGDL) problem. We argue that the graph heterogeneity reflected on node types, node attributes, and neighborhood structures can impose significant challenges for generalizing LDL onto graphs. To address the challenges, we propose a new learning framework with two key components: 1) proactive graph topology homogenization, and 2) topology and content consistency-aware graph transformer. Specifically, the former learns optimal information aggregation between meta-paths, so that the node heterogeneity can be proactively addressed prior to the succeeding embedding learning; the latter leverages an attention mechanism to learn consistency between meta-path and node attributes, allowing network topology and nodal attributes to be equally emphasized during the label distribution learning. By using KL-divergence and additional constraints, HGDL delivers an end-to-end solution for learning and predicting label distribution for nodes. Both theoretical and empirical studies substantiate the effectiveness of our HGDL approach. Our code and datasets are available at https://github.com/Listener-Watcher/HGDL.

1 Introduction

Definite supervision signals are often postulated in learning settings [3, 4]; yet, data generated from the real world tend to present inherent ambiguity, imposing challenges on assertive classifiers that predict instances into single or multiple classes. Label Distribution Learning (LDL) [5, 6, 7, 8, 9] has emerged to navigate label ambiguity by pursuing a mapping from instances to their class distributions. Each distribution quantifies the *descriptive degrees* of various classes given a specific instance.

However, the existing LDL studies mainly [10, 6, 7, 11] focus on independent and identically distributed (IID) data, such as images or texts, which do not generalize well on *graphs*. In fact, the topological structure underlying instances may provide invaluable information for label distribution learning. For example, in the task of urban planning, recent learning models have been employed to predict the point of interests (POIs) of local regions [12, 13, 14, 15]. LDL can further extend this task by providing the regional distributions over all POIs, which lends a finer-granular delineation of urban regional functionality instead of single- or multi-class classification. To wit, for a region that mixes four POIs (classes): housing, healthcare, education and worship, unlike other models assertively classify it into one or multiple POI(s), LDL model can provide insights of the functional degrees of all four POIs in this region, as shown in Figure 1. Nevertheless, existing LDL studies overlook the

[*]Corresponding author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

urban topology, which can be rendered from, e.g., the taxi services across regions [2], missing out critical city traffic patterns that are highly correlated with regional functionalities. For instance, the regions with balanced POI distributions (e.g., R_2 and R_3) are less likely to form connections with other nodes compared to regions heavily skewed towards a single class (e.g., R_4), as their residents enjoy fewer needs to travel to other regions for services such as education and healthcare.

In this paper, we aim to enable and generalize the label distribution learning paradigm in networked data. Two technical challenges confront our study. First, real-world graphs are mostly heterogeneous, comprising diverse types of nodes for better expressiveness. Graph heterogeneity complicates the message-passing between nodes of a specific type (e.g., residence), as the label distributions of those nodes are influenced by their neighboring nodes that may vary in terms of types, content, and topological features. Simply leveraging node embeddings generated from message-passing for LDL will thus not work well [16, 17]. To aid, although meta-path aggregation [18] is seemingly viable, it necessitates extensive domain knowledge and expertise to craft meta-paths for each node type with respect to their label distributions; given the combinatorial number of possible meta-paths in large heterogeneous graphs, searching for the optimal meta-path for LDL is costly, laborious, and time-demanding.

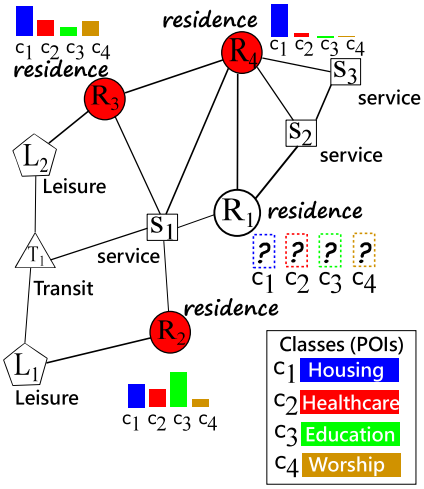


Figure 1: Motivating example of HGDL study, where each node is a local urban region [1] and edges represent taxi services [2] commuting between regions. Heterogeneous node types indicate disparate land use, including residence (R), service (S), leisure (L), and transit (T), among which R nodes are of our interest. Colored R nodes are with ground truth, which delineate their distributions over multiple point-of-interests (POIs), and each POI is deemed as a class/label. Our HGDL problem is to predict the label distribution of uncolored R nodes, enabling a precise delineation of regional urban functionality.

Second, graph topology and nodal features may suggest inconsistent label distributions, where nodes sharing similar contents are positioned far apart on the graph topology. The inconsistency is furthered in heterogeneous graphs, where nodes of the same type often connect through other intermediary types, resulting in substantial topological distances between them. Unlike traditional LDL that focuses on instance vectors only, an effective LDL model on graphs require harmonizing nodal contents with topological structures for a unified representation. The impact of distantly positioned nodes within a graph is substantially diminished, consequently steering the LDL model to prioritize individual nodal vectors, leading to compromised node representations in which the informative patterns embedded in their neighborhood structures are overlooked. Such patterns, which may significantly enhance label distribution predictions as illustrated in Figure 1, are neglected, undermining the LDL model effectiveness.

To overcome the challenges, we propose a new learning framework, coined *Heterogeneous Graph label Distribution Learning* (HGDL). Specifically, to tame the graph heterogeneity, HGDL learns the optimal graph topology of the target nodes from multiple homogeneous structures searched with various meta-path schemes through a tailored attention mechanism. The node embeddings are then generated by harmonizing the nodal features and the learned meta-path graph using a transformer architecture. A joint optimization objective is crafted based on the distance between true and predicted label distributions of the target nodes from their resultant embeddings, which unifies the learning of meta-path graph topology and the feature-topology harmonization function in an end-to-end fashion.

A key innovation of HGDL is that it changes existing heterogeneous graph learning paradigm from *reactive* (meaning that aggregation of different meta-paths are done ***after*** embedding learning from individual meta-path), to be *proactive* (meaning that aggregation are done ***before*** embedding learning). Combined with attention and transformer mechanisms to adjust individual meta-paths' interplay, and align with nodal features, HGDL deliver significantly better performance over alternatives. Our theoretical analysis assures that HGDL outperforms that of using an arbitrary meta-path graph, and HGDL's topology and feature consistency learning sparsifies network connectivity, intermediately encouraging tightens the error-bound, resulting in better model generalization.

Specific contributions of this paper are as follows:

1. This study pioneers the exploration of LDL problem in heterogeneous graphs. The learning problem enjoys practical implications such as for urban functionality delineation (presented in Sec 6.1) and, to our knowledge, has not yet been explored by any contemporary research.
2. We propose an end-to-end `HGDL` learning approach to jointly learn an optimal meta-path graph topology and align it with nodal features for consistent message-passing. Our approach is surprisingly simple yet effective, with its performance evidenced by theoretical underpinnings. Our approach and its analysis are presented in Sections 4 and 5, respectively.
3. Empirical study has been carried out over five graph datasets that span domains of biomedicine, scholarly network, business network, and urban planning. Experimental results substantiate the effectiveness of our approach over rival models, documented in Section 6.

2 Related Work

Label Distribution Learning (LDL) strives to learn a mapping from input to a distribution that profiles the descriptive degrees of classes associated with it [5, 10, 6, 7, 11]. Existing LDL methods fall into three categories, namely, problem transformation (PT), algorithm adaption (AA), and specialized algorithm (SA). PT methods transform LDL as multiple single-label learning tasks, using with label probabilities [19], and AA approaches revise mainstream learning algorithm to fit the LDL loss. SA algorithms are most commonly used because LDL learning is driven by new algorithm designs. Label correlation has been found to benefit the label distribution learning, where approaches were proposed to encode label correlation to a distance to measure the similarity of any two labels [6]. Later, low-rank approximation is used to construct label correlation matrix to capture the global label correlations [7] Instead of exploring common features for all labels, label-specific features [10] for each label are used to enhance the LDL model. Exploring feature-label and label-label correlation [9] has recently been studied in generalizable label distribution learning for cross domain learning. A Gaussian label distribution learning method [11] employs a parametric model underpinned by an optimization strategy assessing KL-divergence distance between Gaussian distributions, followed by a regression loss to normalize the KL-divergence distance. Noticing the difficulty to obtain ground-truth label distributions, Label Enhancement [20] is commonly used to recover label distributions from logical labels. Our research further push LDL to be generalized onto heterogeneous graphs, which have been overlooked by existing research. Although a recent study [21] explored using LDL in topological spaces, it focused on homogeneous graphs only and cannot work in the setting of more than one node type. Thus, the studied problem in [21] and its challenges differ from ours.

Heterogeneous Graph Neural Networks have drawn extensive attention in graph learning [16, 17, 22, 18, 23, 24], because the graph heterogeneity imposes considerable challenge to model the interplay among various node types, features, labels, and network topology. Using meta-path to aggregate information from different types of nodes/edges is a common approach for heterogeneous graph learning. HetGNN [17, 18] designs graph neural networks to encode features for each type of neighbors and then aggregates neighbors' representation with respect to different types. This provides a way for GNN to deal with heterogeneous graph structures and node attributes. HAN [25] introduces attention mechanisms to heterogeneous graph learning, where attentions are applied to embedding features learned from homogeneous networks, each created from a meta-path. By doing so, attentions serve as a weighting mechanism automatically determining the importance of each meta-path for learning. Using transformers for heterogeneous networks has also been investigated recently. For example. HGT [26] designs node- and edge-type dependent parameters to characterize the heterogeneous attention over each edge, allowing this method to learn representations for different types of nodes and edges. SeHGNN [27] proposes a transformer based semantic fusion module, allowing feature fusion from different meta-paths. Our research is fundamentally different from existing work in two aspects: 1) we study LDL learning for heterogeneous networks, and 2) we propose a new way to aggregate and align information for heterogeneous network.

3 Preliminaries

Notations. A heterogeneous graph is denoted by $G = \{V, E, X, Y\}$ associated with a node type mapping $\phi : V \mapsto \mathcal{T}^v$ and an edge type mapping $\varphi : E \mapsto \mathcal{T}^e$, with $\mathcal{T}^v$ and $\mathcal{T}^e$ the predefined and finite sets of nodes and edges, respectively, and $|\mathcal{T}^v| \geq 2$. Denote $t_\iota \in \mathcal{T}^v$ as the node type of our interest, and suppose in total n nodes are of this type. Without loss of generality, we have $\phi(v_1) = \ldots = \phi(v_n) = t_\iota$, and $V_{t_\iota} = \{v_1, \ldots, v_n\} \subset V$. We deem these n nodes as our *target nodes*, using a feature matrix $X \in \mathbb{R}^{n \times m}$ to describe their nodal contents, where each node contains

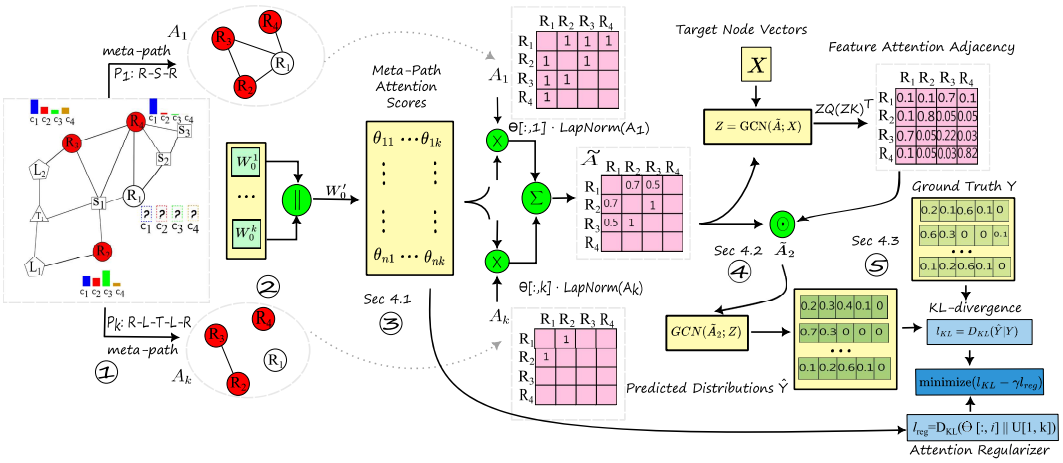


Figure 2: The proposed HGDL framework. Using k meta-paths, the heterogeneous network in ① is converted to k homogeneous meta-path graphs in ②. Topology homogenization in ③ proactively aggregates all k meta-path graphs, through learnable weight matrix $W_0^i \in \mathbb{R}^{n \times f}$ for each graph, and finally obtain attention $\Theta \in \mathbb{R}^{n \times k}$ across all graphs. Topology and feature consistency-aware graph transformer in ④ harmonizes the local and global consistencies. The objective function in ⑤ unifies loss and regularization terms to guide nodal label distribution learning.

an m-dimensional feature vector. A meta-path $\mathcal{P}$ is defined as a relational sequence in form of $t_1 \xrightarrow{r_1} t_2 \ldots \xrightarrow{r_i} t_i \xrightarrow{r_{i+1}} t_{i+1} \ldots \rightarrow t_l$ (abbreviated as $\mathcal{P} = (t_1 t_2 \ldots t_i t_{i+1} \ldots t_l)$, where $(t_i t_{i+1}) \in \mathcal{T}^e$ describes the composite relation between a pair of node types. By defining a meta-path $\mathcal{P}$ with same first and last node type as the target node type, *i.e.* $t_1 = t_l = t_\iota$, we can use $\mathcal{P}$ to convert a heterogeneous network as a meta-path graph concerning only the target node type, which shall be discussed later in Section 4.1.

Problem Statement. In our HGDL problem, the goal is to learn a predictive mapping $\hbar : (G, X) \mapsto Y$, where $Y \in [0, 1]^q$ is a distribution of descriptive labels over q classes. Let $y_{i,j} \in [0, 1]$ be the probability that the node v_i belongs to the j-th class, we have $\sum_{j=1}^{q} y_{i,j} = 1$. In this work, we follow a transductive learning regime [28] to allow the ground-truth label distributions known for a subset of target nodes $V_{tr} \subset V_{t_\iota}$ during training. Our learned mapping $\hbar$ is expected to generalize well so can make accurate prediction on the remaining target nodes $V_{t_\iota} \setminus V_{tr}$.

4 HGDL: The Proposed Approach

Overview. The proposed HGDL approach comprises three key components, as illustrated in Figure 2. First, for the target nodes belonging to the node type t_ι of interest, HGDL generates multiple homogeneous meta-path graphs based on their original locations on the heterogeneous graph through meta-paths; the optimal graph topology of this node type is then learned from the homogeneous graphs via attention mechanism (*Sec 4.1*). Second, HGDL learns the embeddings of the target nodes by harmonizing the information sourced from their feature space and the learned optimal topology using a transformer-like neural architecture (*Sec 4.2*). Third, HGDL minimizes the distance between the predicted and ground-truth label distributions based on the learned node embeddings. We tailor an objective function to unify the three components into one end-to-end optimization problem, in which the optimal graph topology, the harmonization function of the feature and topological information, and the target node label distribution are jointly learned (*Sec 4.3*).

4.1 Optimal Graph Topology Homogenization

For a heterogenous graph, by leveraging meta-path idea, multiple different meta-path homogeneous adjacency matrix can be obtained and they can be treated as multiple sources. Graph learning is about exchanging and updating information from neighbor nodes. A proper neighbor set is therefore important for a target node to learn correct distribution. Given multiple sources, each node will have

multiple neighbor sets to choose from for updating. Traditionally, embeddings are learned for all the neighbor sets, and then aggregation over embeddings is learned. Semantics over embeddings are hard to interpret and learn compared with directly learned from different neighbor sets.

To generate a meta-path graph from the original heterogeneous graph, interactions between the meta-path and the heterogeneous graph path are used. Two nodes v_i and v_j are connected in the meta-path graph, if there exists a path connecting them in the heterogeneous graph, and the path follows the meta-path. Given a meta path $\mathcal{P} = (t_1 \dots t_i t_{i+1} \dots t_l)$, we say that a path $p = (v_1, \dots, v_i, v_{i+1}, \dots, v_l)$ in graph G follows the meta-path $\mathcal{P}$, if $\forall i, \phi(v_i) = t_i$. Take graph in Fig. 1 as an example. Given meta-path $\mathcal{P} = (r\ s\ r)$, which defines node type $t_1 = r, t_2 = s, t_3 = r$. Path $p = (r_2, s_1, r_3)$ in the heterogeneous graph follows $\mathcal{P}$ because all nodes in the path p satisfy $\phi(r_2) = t_1 = r, \phi(s_1) = t_2 = s, \phi(r_3) = t_3 = r$. Because path $p = (r_2, s_1, r_3)$ follows the meta-path $\mathcal{P}$, an edge is used to connect r_2 and r_3 in the homogeneous meta-path graph constructed from $\mathcal{P}$. Indeed, each meta-path defines a specific way of information propagation in a heterogeneous network, with resulted meta-path graph capturing unique relationships between target nodes. While defining a single meta-path is relatively easy, there often exists many meta-paths; aggregating a variety of meta-path graphs to support the downstream learning task is non-trivial.

After searching the meta-paths connecting the nodes of target type t_ι, we generate a set of graphs $\mathcal{A} = \{A_1, \dots A_k\}$, in which each adjacency $A_i \in \{0, 1\}^{n \times n}$ captures the topological structure of the i-th meta-path-based homogeneous graph. Denoted by $A_i[p, q] = 1$ means that two target nodes v_p and v_q, with $\phi(v_p) = \phi(v_q) = t_\iota$, are connected by a meta-path; otherwise, $A_i[p, q] = 0$. Unlike existing studies [25] that yield target node embeddings through *reactive* meta-path aggregation, where they aggregate local neighborhood information for each $A_i \in \mathcal{A}$ to capture k separate meta-path topologies, our `HGDL` learns the optimal graph topology from $\mathcal{A}$ in a *proactive* fashion. Intuitively, `HGDL` learns node-level attention scores for various homogeneous graphs A_i, to respect the fact that the neighboring nodes may pass messages with varying importance levels in local neighborhoods, while the meta-paths walking across nodes of types other than the target t_ι. Revisit the motivating example demonstrated in Fig 1 where the residence nodes are deemed as the target, the meta-path linking through the service nodes dominates, as the residence nodes are more likely to be linked through service nodes instead of Transit and Leisure nodes. Specifically, the attention scores for the nodes in every $A_i \in \mathcal{A}$ are learned in a GAT regime [29], defined as:

$$\Theta = \text{softmax}\Big(\|_{A_i \in \mathcal{A}} \big\{\text{LapNorm}(A_i) W_0^i\big\} W_0^{'}\Big), \quad \tilde{A} = \sum_{i=1}^{k} \Theta[:, i] \cdot \text{LapNorm}(A_i), \tag{1}$$

where $\text{LapNorm}(A_i) = D_i^{-\frac{1}{2}}(A_i + I)D_i^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$ denotes Laplacian normalization of A_i, with D_i being A_i's degree matrix and I is an identity matrix. This term mitigates the imbalanced degree distribution of the meta-path graphs. Denoted by W_0^i and $W_0^{'}$ are the learnable GAT parameters, where $W_0^i \in \mathbb{R}^{n \times f}$ maps the meta-path topology of A_i onto an f-dimensional semantic space. The operator $\|_{A_i \in \mathcal{A}}\{\cdot\}$ concatenates all k resultant node embedding matrices from meta-path graphs $\mathcal{A}$, thereby producing an $\mathbb{R}^{n \times k \cdot f}$ lookup matrix, where each node is associated with a $k \cdot f$-dimensional embedding representation, and each latent f-dimension captures the local neighborhood structure of this node. Then, $W_0^{'} \in \mathbb{R}^{k \cdot f \times k}$ summarize the f-dimensional latent space into one coefficient through convex combination, resulting in attention logits, which are fed into $\text{softmax}(\cdot) = \exp(\cdot)/\sum_i \exp(\cdot)$ to yield the attention matrix $\Theta = [0, 1]^{n \times k}$. Take the i-th column vector of Θ, denoted by $\Theta[:, i] \in [0, 1]^n$, we have the attention scores of n target nodes for message-passing in the i-th meta-path graph A_i. We thus can deem $\tilde{A}$ in Eq. (1) as the learned optimal graph topology, which is an element-wise linear combination of k meta-path topologies with the attention scores broadcast onto all n nodes. Note, $\tilde{A}$ is asymmetric, namely $\tilde{A}[p, q] \neq \tilde{A}[q, p],\ \forall p \neq q$. This is because that the attention score of information aggregation from node v_p to v_q may be different from that from v_q to v_p, as their respective local neighborhood topologies naturally differ.

4.2 Local Topology and Global Feature Consistency-Aware Graph Transformer

After obtaining the optimal topology $\tilde{A}$ from all meta-path graphs A_i, the next question is how to harmonize it with the feature information to better the target node embeddings. The benefit of such harmonization is evident. Revisiting the urban network in Fig 1, we can envision that a pair of residence nodes tend to be associated with similar embedding vectors because they enjoy two types

of consistencies: i) *local neighborhood topology*: their residents tend to travel to similar functional regions for leisure or service purposes, and ii) *global feature space*: they share similar contents such as house types and number of residing families. These local and global consistencies complement with each other, as the residence nodes having similar contents can be topologically faraway from each other on the urban network, and vice versa.

To harmonize the local and global consistencies, we are inspired by the recent graph transformers [30] and observe that the feature attention suggests a global adjacency matrix, which can be incorporated into the message-passing process. We define the graph transformer block as follows.

$$Z = \text{ReLU}(\tilde{A}XW_1), \quad \tilde{A}_2 = \text{LapNorm}\Big(\text{softmax}\big((ZQ)(ZK)^\top \odot \tilde{A}\big)\Big), \tag{2}$$

where $Z \in \mathbb{R}^{n\times h}$ is the node embeddings learned from the optimal graph topology $\tilde{A}$ through local information aggregation, parameterized by $W_1 \in \mathbb{R}^{m\times h}$. Denoted by $\tilde{A}_2 \in [0,1]^{n\times n}$ is the normalized feature attention adjacency, where Q and $K \in \mathbb{R}^{h\times h}$ map the embedding matrix Z onto the latent query and key spaces, respectively, such that $(ZQ)(ZK)^\top$ calculates an $n \times n$ node-level attention matrix with respect to the feature space information. Instead of normalizing the attention score by the hidden dimension h, we penalize the feature attention adjacency through an element-wise production $\odot$ with the optimal meta-path topology $\tilde{A}$. The intuition behind Eq. (2) is that, for each target node, it aggregates information from those neighboring nodes only if their meta-path topology and feature space are both with high attention scores. In addition, Eq. (2) functions similarly to the edge dropout [31]; in lieu of randomly removing edges, we enforce a neighbor-set intersection, where the information is only propagated from the neighbors on which the feature space and meta-path topology both agree. Such an intersection sparsity thus lowers the degree of the resultant attention adjacency, thereby uplifting the learning efficacy, which will be substantiated later in Sec 5. Finally, denoted by $H = \text{LeakyReLU}(\tilde{A}_2 Z W_2) \in \mathbb{R}^{n\times h}$ are the resultant node embeddings, capturing both local topology and global feature consistencies, which is parameterized by weight $W_2 \in \mathbb{R}^{h\times h}$.

4.3 An End-to-End HGDL Objective Function

Based on the resultant target node embeddings H, we can predict their label distributions as $\hat{Y} = \text{softmax}(\tilde{A}_2 H W_3) \in [0,1]^{n\times q}$, where $\hat{Y}_i = \{\hat{y}_{i,j}\}_{j=1}^{q}$ is the predicted label distribution of node v_i, among which $\hat{y}_{i,j}$ denotes its predicted probability of belonging to the class j. The unified objective of our HGDL framework is defined as follows.

$$\min_{\{W_0^i\}_{i=1}^k, W_0', W_1, W_2, W_3, K, Q} \ell_{\texttt{HGDL}} = D_{\text{KL}}(Y\|\hat{Y}) - \gamma \cdot \Omega,$$
$$D_{\text{KL}}(Y\|\hat{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{q} y_{i,j} \cdot \log \frac{y_{i,j}}{\hat{y}_{i,j}}, \quad \Omega = \sum_{i=1}^{n} D_{\text{KL}}(\Theta[i,:] \parallel U[1,k]), \tag{3}$$

where the KL-divergence $D_{\text{KL}}(Y\|\hat{Y})$ gauges the discrepancy between the predicted and groundtruth label distributions of the target nodes [32]. The regularization term Ω gauges the distance between the attention scores of the i-th node across k meta-path typologies (denoted by $\Theta[i,:]$) and a uniform distribution $U[1,k]$. We note the minus sign before Ω, thus minimizing this term encourages a larger KL-distance, thereby avoiding the trivial uniform attention distribution (meaning that for each node, the learned attention weights from different meta-paths are encouraged to be as different as possible). γ is a tuned parameter to balance the two terms.

5 Analysis

We follow the PAC-Bayes regime to analyze the theoretical performance of our HGDL algorithm by deriving its generalization error bound. We proceed analysis based on the meta-path graph adjacency matrices $\mathcal{A} = \{A_1, \dots A_k\}$, which are searched from the heterogeneous graph G. Throughout the analysis, we assume the nodal feature representations to be residing in an ℓ_2-ball of radius B. We argue this a mild assumption, because in implementation we can leverage the batch-norm layers to normalize the resultant node embeddings, such that $\|\mathbf{h}_i^j\|_2 \leq B$, where $\mathbf{h}_i^j$ denotes the i-th node's embedding resulted from the j-th hidden layer.

Let $L_{\mathcal{G}}(\hbar)$ and $L_{(X,\tilde{A})}(\hbar)$ denote the *generalization risk* over a graph distribution $\mathcal{G}$ and the *empirical risk* on the target node samples and the learned meta-path topology $(X, \tilde{A})$, respectively, where $(X, \tilde{A}) \in G \overset{\text{iid}}{\sim} \mathcal{G}$. We can define:

$$L_{\mathcal{G}}(\hbar) = \mathbb{E}_{(X,\tilde{A})\sim\mathcal{G}}\mathbb{E}_{y_i\sim Y}\big[\ell\big(\hbar(X,\tilde{A})[i], y_i\big)\big],$$
$$L_{(X,\tilde{A})}(\hbar) = \frac{1}{n}\sum_{i=1}^{n}\big[\ell\big(\hbar(X,\tilde{A})[i], y_i\big)\big],$$

where $\ell(\cdot,\cdot)$ is a convex distance metric between two distributions that follows $|\ell(u,p) - \ell(u,q)| \leq (\sqrt{p}+1)\|p-q\|_2,\ \forall u,p,q \in \mathbb{R}^m$. Denoted by $\hbar(X,\tilde{A})[i] \in \mathbb{R}^q$ and $y_i \in [0,1]^q$ the predicted and ground-truth label distribution of the i-th target node, respectively. Implementing KL-divergence, we have $\ell\big(\hbar(X,\tilde{A})[i], y_i\big) = \sum_{j=1}^{q}\hbar(X,\tilde{A})[i,j]\ln(\hbar(X,\tilde{A})[i,j]/y_{i,j})$, where the predicted probability that node i belongs to the j-th class is denoted by $\hbar(X,\tilde{A})[i,j]$. By analyzing the performance of the learned meta-path graph topology $\tilde{A}$, we find that:

Theorem 1. *Let $\mathbb{E}[L_{(X,A_i)}(\hbar)]$ be the empirical risk of using the i-th meta-path graph $A_i \in \mathcal{A}$ for label distribution prediction. With the SGD step-size η, we have*

$$L_{(X,\tilde{A})}(\hbar) \leq \min_{A_i\in\mathcal{A}}\mathbb{E}[L_{(X,A_i)}(\hbar)] + \frac{\ln k}{\eta n} + \frac{\eta}{8}.$$

Remark 1. Theorem 1 indicates that the empirical risk of HGDL is no larger than the minimum empirical risk incurred by training label distribution learner on the optimal meta-path graph, as the error bound on the RHS reduces to $\mathcal{O}(1/n)$ with constant k and η. With Stochastic Gradient Descent (SGD) optimizer, larger number of target nodes n will lead to more training updates over them, diminishing the $\mathcal{O}(1/n)$ bound faster. This finding substantiates the tightness of our meta-path learning strategy for the optimal graph topology $\tilde{A}$.

Due to page limits, we defer the proof of Theorem 1 and the rest analysis to the Supplement. We then analyze the generalization error bound of HGDL and find that:

Theorem 2. *Let $\hbar \in \mathcal{H} : \mathcal{X}\times\mathcal{G} \to \mathbb{R}^q$ be an l-layer message-passing neural network with maximum hidden dimension k, of which the i-th layer is parameterized by W_i. Then for any $\delta,\gamma,B>0$ and $l>1$, with probability at least $1-\delta$ we have*

$$L_{\mathcal{G}}(\hbar) - L_{(X,\tilde{A})}(\hbar) \leq \frac{2(\sqrt{2q}+\sqrt{2})q}{\sqrt{n}}\max_{i\in[n],j\in[l]}\left\|\mathbf{h}_i^j\right\|_2$$
$$+ 3b\sqrt{\frac{\log 2/\delta}{2n}} + \mathcal{O}\left(\sqrt{\frac{B^2 d^{l-1}l^2 k\log(lk)\mathcal{D}(W_i)+\log\frac{nl}{\delta}}{\gamma^2 n}}\right),$$

where $\mathcal{D}(W_i) = \prod_{i=1}^{l}\|W_i\|_2^2 \cdot \sum_{i=1}^{l}\left(\|W_i\|_F^2 / \|W_i\|_2^2\right)$ bounds the hypothesis space and b is a constant.

Remark 2. We remark several key observations from Theorems 1 and 2. First, the generalization capability of the algorithm is negatively impacted by a higher dimensional label space q. Second, the robustness of HGDL decreases with larger B values, which gauges the magnitude of perturbations thus the inherent high data variance. Third, as the graph neural network architecture becomes deeper (larger l) or wider (larger k), the generalization risk increases, suggesting the potential risk of model overfitting. Forth, with larger n, the generalization error bound diminishes, which indicates that the meta-path topology can be better delineated with an increased number of target nodes on the graph.

Remark 3. By combining Theorems 1 and 2, we observe that the generalization error bound of HGDL using $\tilde{A}$ outperforms that of using an arbitrary meta-path graph A_i. Further, it is easy to verify that the maximum degree of $\tilde{A}$, denoted by d, is smaller than that of A_i, denoted by d_i, *i.e.,* $d \leq d_i$. $\forall i \in [k]$. This rationalizes our graph transformer design in Sec 4.2, where the enforced topology and feature consistency in Eq. (2) sparsifies network connectivity, thereby intermediately encourages better model generalization.

6 Experiments

6.1 Experiment Setup

Dataset	# node type	# nodes	# edges	# features	# labels
DRUG	4	40,786	1,737,890	191	28
ACM	5	20,200	104,976	1,903	14
DBLP	4	27,325	148,246	8,920	4
YELP	4	8,052,542	7,905,197	19	9
URBAN	4	1,434	42,857	155	10

Table 1: Summary of dataset statistics.

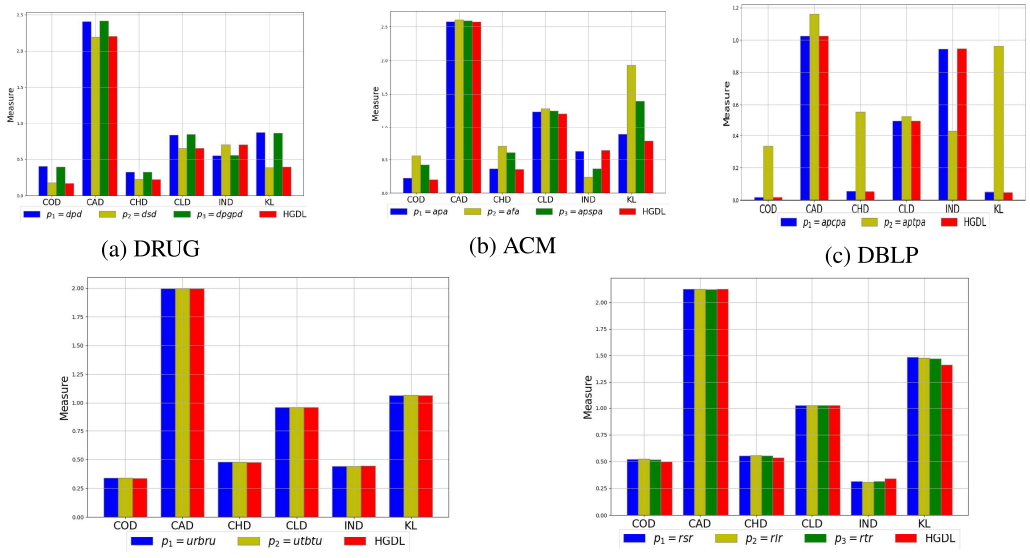
Benchmark Datasets To our best knowledge, no heterogeneous graph dataset with *ground-true label distributions* currently exists. To level the comparison study, we prepare five datasets with ground-truth node label distributions using existing heterogeneous graphs, including DBLP [33], ACM [33], YELP, DRUG [34], and URBAN [1]. Table 1 summarizes the data statistics. A detailed description on the dataset creation and preprocessing, as well as their domain and label semantic meanings, has been deferred to the Supplement B due to space limitation.

Compared Models In total six competitors are identified for comparative study. As no model directly resolving the HGDL problem exists, we employ the state-of-the-art heterogeneous graph neural networks and integrate them KL-divergence loss to learn label distributions of nodes. They include: 1) GCN_{KL}: A baseline that uses graph constructed from each meta-path to train a vanilla GCN [35], using KL-divergence as loss function, and reports the best meta-path result; 2) HAN_{KL}: This baseline uses HAN [25] to integrate embedding from different meta-paths; and 3) $\text{SeHGNN}_{\text{KL}}$: This baseline uses SeHGNN [27], a transformer based approach, to aggregate meta-paths embedding with KL-divergence loss function. For ablation study, we further include three variants reduced from our proposed HGDL method, which include: 4) $\text{HGDL}_{\neg\text{TH}}$: it removes HGDL's topology homogenization (Sec 4.1), which learns embedding from each meta-path graph and reports the best meta-path result; 5) $\text{HGDL}_{\neg\text{transformer}}$: it uses GCN instead of the transformer (Sec 4.2) to learn embedding to validate HGDL's transformer for embedding learning; and 6) HGDL_{ED}: it replaces HGDL's topology and feature consistence-aware graph transformer (Sec. 4.2) by using a random edge dropout method [31].

Evaluation Metrics To measure the discrepancy between two distributions, i.e., the predicted and true label distributions of target nodes, we identify six metrics: Cosine Distance (COD), Canberra Distance (CAD), Chebyshev Distance (CHD), Clark Distance (CLD); Intersection Score (IND), and Kullback-Leibler Divergence (KL). Their definitions and calculations are deferred to Supplement B.

Dataset	Model	COD↓	CAD↓	CHD↓	CLD↓	IND↑	KL↓	Win/Tie/Lose
DRUG	GCN_{KL}	$0.220_{\pm.025}$	$9.209_{\pm.740}$	$0.245_{\pm.031}$	$1.963_{\pm.134}$	$0.676_{\pm.029}$	$0.484_{\pm.075}$	4/2/0
	HAN_{KL}	$0.279_{\pm.023}$	$10.084_{\pm.823}$	$0.268_{\pm.027}$	$2.155_{\pm.148}$	$0.632_{\pm.025}$	$0.579_{\pm.055}$	6/0/0
	$\text{SeHGNN}_{\text{KL}}$	$0.286_{\pm.018}$	$10.178_{\pm.781}$	$0.267_{\pm.020}$	$2.166_{\pm.143}$	$0.640_{\pm.016}$	$0.600_{\pm.059}$	6/0/0
	HGDL	$\mathbf{0.168}_{\pm.019}$	$\mathbf{9.179}_{\pm.574}$	$\mathbf{0.217}_{\pm.017}$	$\mathbf{1.957}_{\pm.114}$	$\mathbf{0.710}_{\pm.020}$	$\mathbf{0.392}_{\pm.044}$	–
	$\text{HGDL}_{\neg\text{transformer}}$	$0.199_{\pm.014}$	$9.371_{\pm.679}$	$0.235_{\pm.017}$	$2.004_{\pm.137}$	$0.687_{\pm.021}$	$0.492_{\pm.059}$	4/2/0
	$\text{HGDL}_{\neg\text{TH}}$	$0.212_{\pm.023}$	$9.510_{\pm.602}$	$0.240_{\pm.018}$	$2.029_{\pm.110}$	$0.671_{\pm.020}$	$0.462_{\pm.050}$	4/2/0
	HGDL_{ED}	$0.204_{\pm.026}$	$9.602_{\pm.882}$	$0.239_{\pm.028}$	$2.040_{\pm.162}$	$0.681_{\pm.030}$	$0.574_{\pm.085}$	4/2/0
ACM	GCN_{KL}	$0.217_{\pm.007}$	$13.101_{\pm.014}$	$0.337_{\pm.012}$	$3.527_{\pm.057}$	$0.652_{\pm.013}$	$0.842_{\pm.072}$	5/1/0
	HAN_{KL}	$0.212_{\pm.005}$	$13.114_{\pm.009}$	$0.371_{\pm.008}$	$3.485_{\pm.023}$	$0.618_{\pm.008}$	$0.765_{\pm.015}$	5/1/0
	$\text{SeHGNN}_{\text{KL}}$	$0.247_{\pm.061}$	$13.141_{\pm.052}$	$0.371_{\pm.082}$	$3.492_{\pm.061}$	$0.617_{\pm.090}$	$0.924_{\pm.166}$	4/2/0
	HGDL	$\mathbf{0.203}_{\pm.004}$	$\mathbf{13.098}_{\pm.006}$	$\mathbf{0.351}_{\pm.004}$	$3.408_{\pm.035}$	$\mathbf{0.637}_{\pm.004}$	$\mathbf{0.753}_{\pm.025}$	–
	$\text{HGDL}_{\neg\text{transformer}}$	$0.211_{\pm.008}$	$13.099_{\pm.010}$	$0.358_{\pm.011}$	$3.403_{\pm.027}$	$0.630_{\pm.012}$	$0.777_{\pm.031}$	1/5/0
	$\text{HGDL}_{\neg\text{TH}}$	$0.223_{\pm.006}$	$13.130_{\pm.015}$	$0.361_{\pm.019}$	$3.423_{\pm.022}$	$0.631_{\pm.020}$	$0.879_{\pm.034}$	3/3/0
	HGDL_{ED}	$0.216_{\pm.007}$	$13.106_{\pm.005}$	$0.364_{\pm.006}$	$\mathbf{3.375}_{\pm.041}$	$0.624_{\pm.007}$	$0.801_{\pm.020}$	5/1/0
DBLP	GCN_{KL}	$0.031_{\pm.004}$	$2.852_{\pm.009}$	$0.091_{\pm.006}$	$1.647_{\pm.002}$	$0.908_{\pm.006}$	$0.114_{\pm.011}$	6/0/0
	HAN_{KL}	$0.025_{\pm.002}$	$2.819_{\pm.012}$	$0.071_{\pm.004}$	$1.633_{\pm.007}$	$0.929_{\pm.004}$	$0.082_{\pm.008}$	5/1/0
	$\text{SeHGNN}_{\text{KL}}$	$0.086_{\pm.140}$	$2.887_{\pm.170}$	$0.155_{\pm.208}$	$\mathbf{1.624}_{\pm.049}$	$0.842_{\pm.214}$	$0.252_{\pm.397}$	0/6/0
	HGDL	$\mathbf{0.019}_{\pm.002}$	$\mathbf{2.796}_{\pm.014}$	$\mathbf{0.057}_{\pm.005}$	$1.633_{\pm.005}$	$\mathbf{0.943}_{\pm.005}$	$\mathbf{0.057}_{\pm.011}$	–
	$\text{HGDL}_{\neg\text{transformer}}$	$0.025_{\pm.002}$	$2.828_{\pm.004}$	$0.074_{\pm.005}$	$1.642_{\pm.001}$	$0.925_{\pm.005}$	$0.090_{\pm.008}$	6/0/0
	$\text{HGDL}_{\neg\text{TH}}$	$0.020_{\pm.002}$	$2.808_{\pm.013}$	$0.062_{\pm.005}$	$1.637_{\pm.005}$	$0.937_{\pm.005}$	$0.070_{\pm.011}$	3/3/0
	HGDL_{ED}	$0.023_{\pm.001}$	$2.819_{\pm.005}$	$0.070_{\pm.003}$	$1.639_{\pm.003}$	$0.929_{\pm.003}$	$0.082_{\pm.006}$	6/0/0
YELP	GCN_{KL}	$\mathbf{0.342}_{\pm.014}$	$7.180_{\pm.125}$	$0.458_{\pm.015}$	$2.558_{\pm.031}$	$0.456_{\pm.016}$	$1.037_{\pm.044}$	0/6/0
	HAN_{KL}	$0.453_{\pm.163}$	$5.894_{\pm1.808}$	$0.569_{\pm.158}$	$\mathbf{2.226}_{\pm.461}$	$0.379_{\pm.118}$	$5.832_{\pm6.577}$	3/2/1
	$\text{SeHGNN}_{\text{KL}}$	$0.404_{\pm.106}$	$6.298_{\pm1.757}$	$0.522_{\pm.111}$	$2.343_{\pm.438}$	$0.413_{\pm.078}$	$3.993_{\pm5.426}$	0/6/0
	HGDL	$\mathbf{0.342}_{\pm.015}$	$7.177_{\pm.128}$	$\mathbf{0.457}_{\pm.016}$	$2.558_{\pm.031}$	$\mathbf{0.459}_{\pm.015}$	$\mathbf{1.034}_{\pm.041}$	–
	$\text{HGDL}_{\neg\text{transformer}}$	$\mathbf{0.342}_{\pm.014}$	$7.175_{\pm.128}$	$0.458_{\pm.016}$	$2.557_{\pm.031}$	$0.458_{\pm.015}$	$1.035_{\pm.039}$	0/6/0
	$\text{HGDL}_{\neg\text{TH}}$	$\mathbf{0.342}_{\pm.015}$	$\mathbf{7.173}_{\pm.126}$	$0.458_{\pm.017}$	$2.556_{\pm.031}$	$0.458_{\pm.016}$	$1.046_{\pm.051}$	0/6/0
	HGDL_{ED}	$0.348_{\pm.021}$	$7.221_{\pm.174}$	$0.463_{\pm.020}$	$2.565_{\pm.038}$	$0.453_{\pm.019}$	$1.070_{\pm.078}$	0/6/0
URBAN	GCN_{KL}	$0.485_{\pm.025}$	$8.318_{\pm.037}$	$0.536_{\pm.014}$	$\mathbf{2.773}_{\pm.009}$	$0.331_{\pm.011}$	$1.386_{\pm.069}$	2/4/0
	HAN_{KL}	$0.497_{\pm.017}$	$8.337_{\pm.029}$	$0.538_{\pm.010}$	$2.777_{\pm.008}$	$0.326_{\pm.005}$	$1.407_{\pm.054}$	4/2/0
	$\text{SeHGNN}_{\text{KL}}$	$0.497_{\pm.019}$	$8.336_{\pm.029}$	$0.537_{\pm.011}$	$2.776_{\pm.008}$	$0.327_{\pm.006}$	$1.409_{\pm.061}$	4/2/0
	HGDL	$\mathbf{0.467}_{\pm.023}$	$\mathbf{8.315}_{\pm.041}$	$\mathbf{0.517}_{\pm.012}$	$2.775_{\pm.009}$	$\mathbf{0.356}_{\pm.010}$	$\mathbf{1.340}_{\pm.065}$	–
	$\text{HGDL}_{\neg\text{transformer}}$	$0.497_{\pm.013}$	$8.340_{\pm.032}$	$0.537_{\pm.006}$	$2.777_{\pm.008}$	$0.325_{\pm.006}$	$1.409_{\pm.041}$	4/2/0
	$\text{HGDL}_{\neg\text{TH}}$	$0.500_{\pm.016}$	$8.338_{\pm.028}$	$0.540_{\pm.010}$	$2.776_{\pm.008}$	$0.321_{\pm.004}$	$1.414_{\pm.046}$	4/2/0
	HGDL_{ED}	$0.481_{\pm.028}$	$8.325_{\pm.031}$	$0.527_{\pm.012}$	$2.775_{\pm.007}$	$0.340_{\pm.017}$	$1.375_{\pm.067}$	1/5/0

Table 2: Mean $\pm$ standard deviation results of seven models on five datasets. Best results are bold, and $\uparrow$ (or $\downarrow$) indicates the higher (or lower) the better. Results are taken from 5 repeats. The win/tie/loss counts are suggested by the *paired t-test* at 90% confidence level.



(a) DRUG (b) ACM (c) DBLP

(d) YELP (e) URBAN

Figure 3: Comparisons between HGDL *vs.* results from a single meta-path (CAD and CLD are calculated in natural log for better visualization) for five datasets.

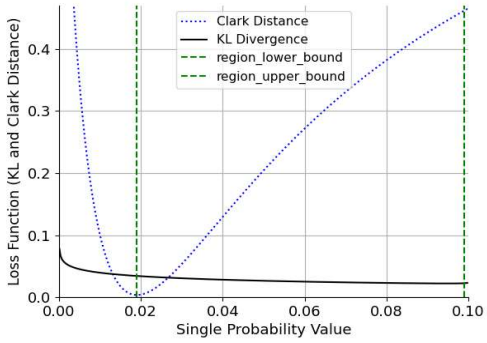


Figure 4: KL and CLD tradeoff function example. The estimated probability distribution is [x_1, x_2, 0.9] and true probability distribution is [0.05,0.05,0.9], with $x_1 + x_2 = 0.1$. Horizontal axis is the x_1 value and vertical axis is the loss for both CLD and KL divergence. Green dashed lines cover the tradeoff region where the CLD loss monotonically increases and KL-divergence decreases.

6.2 Results

Table 2 summarizes results of all methods. Overall, our HGDL wins in 99 out of 180 settings, among which on average 20 out of 30 settings excel in COD, CHD, IND, and KL metrics, 11 out of 30 in CAD, and 8 out of 30 in CLD. On DURG, ACM, DBLP, and URBAN datasets, HGDL outperforms its competitors in 83% settings in COD, CHD, IND, and KL metrics 46% in CAD, and 20% in CLD. Beyond its overall better comparative performance, we make the following observation on HGDL. First, HAN_{KL} and $\text{SeHGNN}_{\text{KL}}$ achieve better performance on YELP and DBLP dataset for CAD and CLD metrics, but not on the other datasets and metrics. This shows that existing meta-path based methods cannot learn distribution prediction well. In general, these models show similar performance on YELP dataset. We hypothesize that this is due to the lack of rich feature information on YELP, of which the dimension of nodal features is 19 which is minimum across all datasets. Second, HGDL achieves the best results in KL-divergence by a large margin across all settings. On average, HGDL have a 15% improvement compared with the second best result across all datasets in KL-divergence. Given that KL-divergence is the loss objective in our framework, we extrapolate that HGDL converges well in terms of minimizing the distribution distance. Same observation can

be drawn from the validation loss curve as shown in Supplement C Figures 5, 6, and 7. In addition, on metrics being strongly related to KL divergence including COD, CHD, and IND, our HGDL also enjoys significant performance improvement over other models. Among the metrics, CLD metrics shows a different patterns in terms of KL divergence, we show in Figure 4 that CLD and KL has a tradeoff region in small probability distribution and therefore caused such difference.

Third, the **ablation study** between HGDL and its variants, i.e., $\text{HGDL}_{\neg\text{transformer}}$, $\text{HGDL}_{\neg\text{TH}}$ and HGDL_{ED}, demonstrate clear benefits of topology homogenization and consistency-aware graph transformer in aggregating meta-paths and nodal features for LDL learning for heterogeneous graphs (more results are deferred to the Section E.1 in Supplement C; there, we observe that HGDL_{ed} has no improvement in KL-divergence with different edge drop rates compared to $\text{HGDL}_{\neg\text{transformer}}$, which is the model with 0 edge drop rate). We observe in Table 2 that $\text{HGDL}_{\neg\text{transformer}}$ shows comparable performance on ACM and YELP by tying HGDL in five and six metrics, respectively; however, HGDL outperforms it in all settings in other three datasets. Likewise, $\text{HGDL}_{\neg\text{TH}}$ ties HGDL across all metrics in YELP but is inferior to HGDL in all settings in other four datasets. HGDL_{ED} ties HGDL in six and five settings on YELP and URBAN, respectively, but is outperformed by HGDL for all other three datasets in all settings. The robust performance of HGDL can be attributed to two aspects. On the one hand, the improved results over those ablation variants suggest that our devised model components for proactive meta-path learning and attention modeling are indispensable. On other other hand, it substantiates the usefulness of our design that lets HGDL learn semantic fusion before the embedding learning. This end-to-end learning design provides a larger search space for embedding learning to find optimal solutions, whereas other methods that learn embedding and perform fusion in two independent stages may result in suboptimal node embeddings thus inferior LDL performance.

Fourth, even though the optimal meta-path choice may vary across different metrics and datasets, our HGDL that proactively learns to aggregate multiple meta-path graphs leads to the best performance in most cases. Figure 3 illustrates the performance from single meta-path graph and we can observe that our method outperforms the single best path results in all five datasets, with a larger improvement when the meta-path results are close (indicating each meta-path has similar information, *e.g.*, ACM dataset in Figure 3 (b)) and a smaller improvement when one meta-path is significantly inferior to others (*e.g.*, DBLP dataset Figure 3 (c) where p_1 outperforms p_2 with a large margin). These results validate the tightness of Theorem 1 by demonstrating the optimality of the learned meta-path graph in our HGDL method.

6.3 Scalability Analysis

Denote the total number of nodes, hidden dimension size, and number of meta-path by n, f, and k, respectively. The number of learnable parameters is $\mathcal{O}(n)$ for graph topology homogenization, because HGDL requires learning an adjacency matrix from all meta-path, which involves $knf + k^2 f$ training parameters (*i.e.* $\mathcal{O}(n)$ complexity). Inducing adjacency matrix from features, *i.e.* the 2nd stage, only requires $\mathcal{O}(1)$ number of learnable parameters, same as vanilla GCN. As a result, HGDL has $\mathcal{O}(n)$ complexity. The runtime performance is detailed in Appendix H.3.

7 Conclusion

This paper explored a novel graph learning setting, namely, heterogeneous graph label distribution learning. Our goal is to predict label distributions of target nodes in a heterogeneous graph, which enables a finer-granular delineation of node properties compared to traditional single- or multi-class node classification. We demonstrated that the topological heterogeneity and inconsistency impose unique challenge for generalizing LDL into networked data, and proposed HGDL to overcome them. Specifically, HGDL proactively aggregates meta-paths to achieve optimal graph topology homogenization through attention mechanism, followed by a transformer-based approach to ensure topology and feature consistency for learning node label distributions. We analyzed the PAC-Bayes error bound of HGDL, and the result suggests the superiority of our design over those models learned from a single meta-path graph. Empirical results on five benchmark datasets validated the tightness of our analysis and substantiate that HGDL significantly outperformed its competitors.

Acknowledgment

This work has been supported in part by the National Science Foundation (NSF) under Grant Nos. IIS-2236578, IIS-2236579, IIS-2302786, IIS-2441449, IOS-2430224, and IOS-2446522.

References

[1] N. L. Houssou, J.-l. Guillaume, and A. Prigent, "A graph based approach for functional urban areas delineation," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pp. 652–658, 2019.

[2] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri, "Modelling taxi drivers' behaviour for the next destination prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2980–2989, 2019.

[3] P. Kar and P. Jain, "Supervised learning with similarity functions," *NeurIPS*, vol. 25, 2012.

[4] A. Hefny, C. Downey, and G. J. Gordon, "Supervised learning for dynamical system learning," *NeurIPS*, vol. 28, 2015.

[5] X. Geng, "Label distribution learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1734–1748, 2016.

[6] X. Jia, W. Li, J. Liu, *et al.*, "Label distribution learning by exploiting label correlations," in *AAAI*, 2018.

[7] T. Ren, X. Jia, W. Li, and S. Zhao, "Label distribution learning with label correlations via low-rank approximation," in *IJCAI*, p. 3325–3331, 2019.

[8] J. Wang and X. Geng, "Theoretical analysis of label distribution learning," in *AAAI*, vol. 33, pp. 5256–5263, 2019.

[9] X. Zhao, L. Qi, Y. An, and X. Geng, "Generalizable label distribution learning," in *Proceedings of the 31st ACM International Conference on Multimedia (MM-23)*, p. 8932–8941, 2023.

[10] T. Ren, X. Jia, W. Li, L. Chen, and Z. Li, "Label distribution learning with label-specific features," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 3318–3324, International Joint Conferences on Artificial Intelligence Organization, 2019.

[11] H. Xu, X. Liu, Q. Zhao, Y. Ma, C. Yan, and F. Dai, "Gaussian label distribution learning for spherical image object detection," in *CVPR*, 2023.

[12] Z. Yao, Y. Fu, B. Liu, W. Hu, and H. Xiong, "Representing urban functions through zone embedding with human mobility patterns," in *IJCAI*, 2018.

[13] Y. Luo, F.-l. Chung, and K. Chen, "Urban region profiling via multi-graph representation learning," in *CIKM*, pp. 4294–4298, 2022.

[14] Y. Zheng, Y. Lin, L. Zhao, T. Wu, D. Jin, and Y. Li, "Spatial planning of urban communities via deep reinforcement learning," *Nature Computational Science*, vol. 3, no. 9, pp. 748–762, 2023.

[15] Y. Liu, J. Ding, Y. Fu, and Y. Li, "Urbankg: An urban knowledge graph system," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 4, pp. 1–25, 2023.

[16] Q. Lv, M. Ding, Q. Liu, Y. Chen, W. Feng, S. He, C. Zhou, J. Jiang, Y. Dong, and J. Tang, "Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks," in *SIGKDD*, pp. 1150–1160, 2021.

[17] C. Zhang, D. Song, C. Huang, A. Swami, and N. Chawla, "Heterogeneous graph neural network," in *Proc. of KDD*, pp. 793–803, 2019.

[18] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *WWW*, pp. 2331–2341, 2020.

[19] H. Borchani, G. Varando, C. Bielza, and P. Larranaga, "A survey on multi-output regression," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, 07 2015.

[20] Y. Wang, Y. Zhou, J. Zhu, X. Liu, W. Yan, and Z. Tian, "Contrastive label enhancement," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23)*, pp. 4353–4361, 2023.

[21] Y. Jin, R. Gao, Y. He, and X. Zhu, "Gldl: Graph label distribution learning," in *Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence*, 2024.

[22] Y. Dong, Z. Hu, K. Wang, Y. Sun, and J. Tang, "Heterogeneous network representation learning," in *Proc. of the 29th International Joint Conf. on Artificial Intelligence (IJCAI-20)*, pp. 4861–4867, 7 2020.

[23] Y. Jing, Y. Yang, X. Wang, M. Song, and D. Tao, "Amalgamating knowledge from heterogeneous graph neural networks," in *CVPR*, pp. 15709–15718, 2021.

[24] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *SIGKDD*, pp. 1726–1736, 2021.

[25] W. Xiao, J. Houye, S. Chuan, W. Bai, C. Peng, Y. P., and Y. Yanfang, "Heterogeneous graph attention network," *WWW*, 2019.

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

[27] X. Yang, M. Yan, S. Pan, X. Ye, and D. Fan, "Simple and efficient heterogeneous graph neural network," *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10816–10824, 2023.

[28] M. Wan, Y. Ouyang, L. Kaplan, and J. Han, "Graph regularized meta-path based transductive regression in heterogeneous information network," in *SDM*, pp. 918–926, 2015.

[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[30] V. P. Dwivedi and X. Bresson, "A generalization of transformer networks to graphs," 2021.

[31] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations (ICLR)*, 2020.

[32] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[33] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: Extraction and mining of academic social networks," in *KDD*, pp. 990–998, 2008.

[34] Y. Gu, S. Zheng, Q. Yin, R. Jiang, and J. Li, "REDDA: Integrating multiple biological relations to heterogeneous graph neural network for drug-disease association prediction," *Computers in Biology and Medicine*, 11 2022.

[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[36] G. Landrum, P. Tosco, B. Kelley, Ric, D. Cosgrove, sriniker, gedeck, R. Vianello, NadineSchneider, E. Kawashima, G. Jones, D. N, A. Dalke, B. Cole, M. Swain, S. Turk, AlexanderSavelyev, A. Vaucher, M. Wójcikowski, I. Take, V. F. Scalfani, D. Probst, K. Ujihara, guillaume godin, A. Pahl, R. Walker, J. Lehtivarjo, F. Berenger, jasondbiggs, and strets123, "rdkit/rdkit: 2023_09_4 (q3 2023) release," Jan. 2024.

[37] D. S. Himmelstein, "User-friendly extensions to mesh v1.0," Feb. 2016.

[38] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[39] Q. Mao, Z. Liu, C. Liu, and J. Sun, "Hinormer: Representation learning on heterogeneous information networks with graph transformer," 2023.

[40] N. Cesa-Bianchi and G. Lugosi, *Prediction, learning, and games*. Cambridge university press, 2006.

[41] P. L. Bartlett, D. J. Foster, and M. J. Telgarsky, "Spectrally-normalized margin bounds for neural networks," *NeurIPS*, vol. 30, 2017.

[42] B. Neyshabur, S. Bhojanapalli, and N. Srebro, "A pac-bayesian approach to spectrally-normalized margin bounds for neural networks," in *ICLR*, 2018.

[43] R. Liao, R. Urtasun, and R. Zemel, "A pac-bayesian approach to generalization bounds for graph neural networks," in *ICLR*, 2020.

[44] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of computational mathematics*, vol. 12, pp. 389–434, 2012.

[45] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 463–482, 2002.

[46] R. E. Schapire and Y. Freund, *Foundations of machine learning*, pp. 23–52. Mit Press, 2012.

[47] A. Maurer, "A vector-contraction inequality for rademacher complexities," in *ALT*, pp. 3–17, Springer, 2016.

[48] S. M. Kakade, K. Sridharan, and A. Tewari, "On the complexity of linear prediction: Risk bounds, margin bounds, and regularization," *NeurIPS*, vol. 21, 2008.

[49] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, p. 1, 2007.