# Hierarchical Label Inference Incorporating Attribute Semantics in Attributed Networks

Junliang Li[1,2], Yajun Yang[1,2†], Qinghua Hu[1], Xin Wang[1] and Hong Gao[3]
[1]*College of Intelligence and Computing, Tianjin University, Tianjin, China*
[2]*State Key Laboratory of Communication Content Cognition, Beijing, China*
[3]*College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China*
[1]{lijunliang, yjyang, huqinghua, wangx}@tju.edu.cn, [3]honggao@zjnu.edu.cn

2023 IEEE International Conference on Data Mining (ICDM) | 979-8-3503-0788-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDM58522.2023.00129

***Abstract*—Node attribute label inference is an important problem in attributed networks. Most existing works assume that node labels are at a single level, but in practice, the attribute labels can always be organized in a hierarchical structure according to their semantics. In this paper, we propose a novel hierarchical label inference model for attributed networks. Specifically, we propose a triple attention mechanism to extract fine-grained label semantics from three levels: hierarchical, sibling and global. Next, we propose the semantic fully-connected layer to explicitly exploit label semantics for attribute inference. We also propose semantic label propagation to enhance the interaction between the label semantics and the attributed network, and this interaction enables nodes in the attributed network to realise the proximity assumption at the label semantic level. Finally, we combine the semantic fully-connected layer with semantic label propagation for top-down hierarchical attribute inference. Extensive experiments demonstrate the superiority of our model.**

***Index Terms*—attribute inference, hierarchical inference, label semantics**

## I. Introduction

Inferring attributes of nodes in networks (e.g., social networks) is an important problem. Most existing work on attribute inference always assumes that all labels are at one level, e.g., inferring the interests of users in social networks [1]–[3]. In these works, there is no hierarchical relationship between attributes, such as "military" and "classical music", both of which are at the same level. However, in some cases, attributes can be organized in a hierarchy through their semantics. For example, in an academic network, the research interest labels of different users can be organized in a tree-like structure in Fig. 1. Note that "CV" is subfield of "AI" and thus "AI" is the parent of "CV" in this figure. From top to bottom, it is a gradual refinement of the research fields and the lower-level attribute labels are the fine-grained subfield of the upper-level ones. Our task is to infer all attribute labels of a user from top to bottom. We call this type of task hierarchical attribute label inference or hierarchical multi-label classification (HMC).

Existing hierarchical attribute inference methods are mainly classified into local and global methods. Local methods [4]–[6] generate a classifier for each level separately. Local methods ignore the semantic hierarchy of attributes, which can lead to severe inconsistency in hierarchical inference tasks [7]. Also, the local methods are computationally expensive when there are many attribute levels. The objective of the global methods is to develop a single global model for all attributes in the hierarchy. Recent global methods have introduced various strategies to exploit the structural information of top-down paths, such as recursive regularization [8], reinforcement learning [9], and meta-learning [10]. However, the above methods still exploit hierarchical structures in a simple way, thus ignoring fine-grained label relevance information.

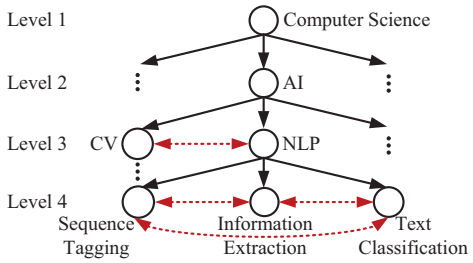


Fig. 1. An example of attribute semantic hierarchy graph.

Currently, only a small amount of work (e.g. [7] and [11]) study hierarchical attribute inference in networks. An attributed network is one where each node has attributes, organized in an Attribute Semantic Hierarchy Graph (ASHG) as depicted in Fig. 1. The MLI method in [7] infers hierarchical attributes for unknown users by collecting nearby user attributes through maximum entropy random walk, adjusting results using ASHG structure. However, [7] only considers proximity in the attributed network, excluding node features in attribute inference, possibly affecting inference accuracy. Although [11] takes node features as input and designs a correction mechanism using ASHG, this correction mechanism still simply exploits the explicit hierarchy of ASHG (parent-child) and does not take into account deeper semantics (e.g., the importance of labels to each other). Furthermore, both [7] and [11] separate the attributed network from the ASHG and do not consider enabling them to interact for more reasonable inference. Based on the above analysis, we need to address the following new requirements when designing hierarchical attribute inference mechanisms for attributed networks: (**Q1**) How to mine the finer-grained label semantics in ASHG? (**Q2**) How to use label semantics for attribute inference? (**Q3**) How can the attributed network be made to interact with ASHG?

†Corresponding author

979-8-3503-0788-7/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDM58522.2023.00129



Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on August 29,2025 at 16:59:59 UTC from IEEE Xplore. Restrictions apply.

To address these three problems, we propose a **H**ierarchical **L**abel **I**nference (HLI) model for attributed networks. Firstly, to address **Q1**, we propose a triple attention mechanism to mine fine-grained semantics in ASHG. We end up with label embeddings that contain rich semantics. Secondly, to address **Q2**, we propose semantic fully-connected layer (SFC), which performs attribute inference for each level by using attributed network node embeddings and label semantics as input and classification weights, respectively. Thirdly, to address **Q3**, we propose semantic label propagation (SLP), which is an approach that combines label semantics and attributed network structure for attribute inference. Finally, we combine SFC and SLP and propose a novel top-down hierarchical inference mechanism.

The main contributions of our work can be summarized as follows:

- We propose a triple attention mechanism to extract the richer semantic information between labels. Compared to previous work that uses only structural information, our model can extract different types of importance between labels.
- We propose two approaches for attribute inference, semantic fully-connected layer (SFC) and semantic label propagation (SLP). SFC directly uses label semantics for attribute inference, while SLP combines label semantics and attributed network structure to satisfy the proximity assumption in attributed network. We also propose a top-down hierarchical attribute inference mechanism combining SFC and SLP, such that lower-level inference can benefit from the results of higher-level inference.
- We conduct extensive experiments to validate the effectiveness of our proposed model on real datasets.

## II. Problem Statement

An attributed network, denoted as $G = (V, E, \boldsymbol{X}, \Sigma)$, is modeled as a directed graph, where $V$ and $E$ are the set of the nodes and edges in $G$ respectively. $\boldsymbol{X} \in \mathbb{R}^{n \times f}$ represents the node feature matrix, where $n$ is the number of nodes in $G$, and every node has $f$ features. $\Sigma$ is a set of attribute labels and $l$ is a function mapping from $V_s$ to $\Sigma$, where $V_s$ is a subset of $V$, called "attribute-known" node set, in which every node $v_i \in V_s$ has a set of attribute labels $l(v_i) = \{l_1, \cdots, l_k\}$, where $l_i$ is a parent of $l_{i+1}$ in semantic for every $1 \leq i \leq k-1$. It means that the attribute information about each node $v_i \in V_s$ is known, while the attribute information about each node $v_j \notin V_s$ is unknown.

**Definition 1** (Attribute Semantic Hierarchy Graph, ASHG)**.** *An Attribute Semantic Hierarchy Graph $H = (\Sigma, T)$ is a directed graph, where the label set $\Sigma$ and $T$ are the node set and edge set respectively. All the nodes in $\Sigma$ are distinct attribute labels and are organized in a tree-like structure with $k$ different levels. $\Sigma_i$ is the set of the labels at the $i$-th level and thus $\Sigma = \bigcup_{1 \leq i \leq k} \Sigma_i$. Attribute labels in $\Sigma_{i+1}$ are the refinement of the corresponding parent nodes in $\Sigma_i$. Every edge $(l_i, l_j) \in T$ means $l_i$ is a parent of $l_j$ in semantic or $l_i$ and $l_j$ has the same semantical parent.* □

Fig. 1 illustrates the attribute semantic hierarchy graph of an academic network. For a researcher $v_i$ whose interest is "Sequence Tagging", the research interest label set of $v_i$ is $l(v_i) = \{$"CS", "AI", "NLP", "Sequence Tagging"$\}$.

**Definition 2** (Hierarchical Attribute Label Inference)**.** *Given a network $G$ with an attribute-known node set $V_s$, and an attribute semantic hierarchy graph $H$, the problem of hierarchical attribute label inference is to determine the attribute label set $l(v_i) = \{l_1, \cdots, l_k\}$ for every node $v_i \in V \setminus V_s$, where $l_i \in \Sigma_i$ is an attribute label at $i$-th level of $H$ and $l_i$ is the parent of $l_{i+1}$ in semantic.* □

## III. Proposed Model

As shown in Fig. 2, HLI is divided into three phases: representation learning, attribute inference and objective optimization. Representation learning is used to extract attributed network node embeddings and label embeddings containing rich semantic information. Attribute inference consists of semantic fully-connected layer (SFC) and semantic label propagation (SLP), and aims to use label semantics for attribute inference. Finally, we perform model optimisation using cross-entropy loss.

### A. Representation Learning

*1) ASHG Representation Learning:* We designed a triple attention mechanism to extract the rich label semantics in ASHG. Hierarchical and sibling-level attention is used to extract two different label semantics, and global-level attention is used to combine the two semantics.

**Hierarchical-level Attention.** The design of hierarchical-level attention is inspired by varying importance of parent and child nodes for a label. For example, in learning the embedding of the "NLP" node in Fig. 1, "AI" and "Sequence Tagging" are the coarser and finer-grained semantics of "NLP", respectively. If our task is to infer the specific research direction of NLP researchers, then perhaps "Sequence Tagging" is more important for "NLP" node.

Assuming that the feature of node $h_i$ in ASHG $H = (\Sigma, T)$ is $\boldsymbol{f}_i \in \mathbb{R}^b$, we first convert the feature vector of node $h_i$ as follows to obtain a more potent expression: $\boldsymbol{f}_i^{'} = \boldsymbol{W}\boldsymbol{f}_i$, where $\boldsymbol{W} \in \mathbb{R}^{d \times b}$ is the learnable weight and $\boldsymbol{f}_i^{'} \in \mathbb{R}^d$ is the transformed feature of node $h_i$. Next, the weight between node $h_i$ and its hierarchical neighbor $h_j$ are defined as:

$$a_{i,j} = \boldsymbol{\nu}_\Phi \cdot \cos(\boldsymbol{f}_i^{'}, \boldsymbol{f}_j^{'}), h_j \in N_i^\Phi \cup \{h_i\} \tag{1}$$

$$\alpha_{ij} = \mathrm{softmax}(a_{i,j}) = \frac{\exp(a_{i,j})}{\sum_{h_k \in N_i^\Phi \cup \{h_i\}} \exp(a_{i,k})} \tag{2}$$

where $\boldsymbol{\nu}_\Phi \in \mathbb{R}^3$ is the trainable parameter and $\cos(\boldsymbol{f}_i^{'}, \boldsymbol{f}_j^{'}) = \boldsymbol{f}_i^{'\top}\boldsymbol{f}_j^{'} / \|\boldsymbol{f}_i^{'}\| \|\boldsymbol{f}_j^{'}\|$ with the $L_2$ norm $\|\boldsymbol{f}_i^{'}\|$. Where $N_i^\Phi$ denotes the hierarchical neighbor nodes of node $h_i$. $a_{i,j} \in \mathbb{R}$ denotes the importance of node $h_i$'s hierarchical neighbor $h_j$ to node $h_i$. It is worth noting that the $\cos(\cdot, \cdot)$ function is symmetric, i.e., $\cos(\boldsymbol{f}_i^{'}, \boldsymbol{f}_j^{'}) = \cos(\boldsymbol{f}_j^{'}, \boldsymbol{f}_i^{'})$. However, this symmetry does not make sense in the real world. For example, in social networks, nodes with low influence are easily influenced by nodes with



Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on August 29,2025 at 16:59:59 UTC from IEEE Xplore. Restrictions apply.

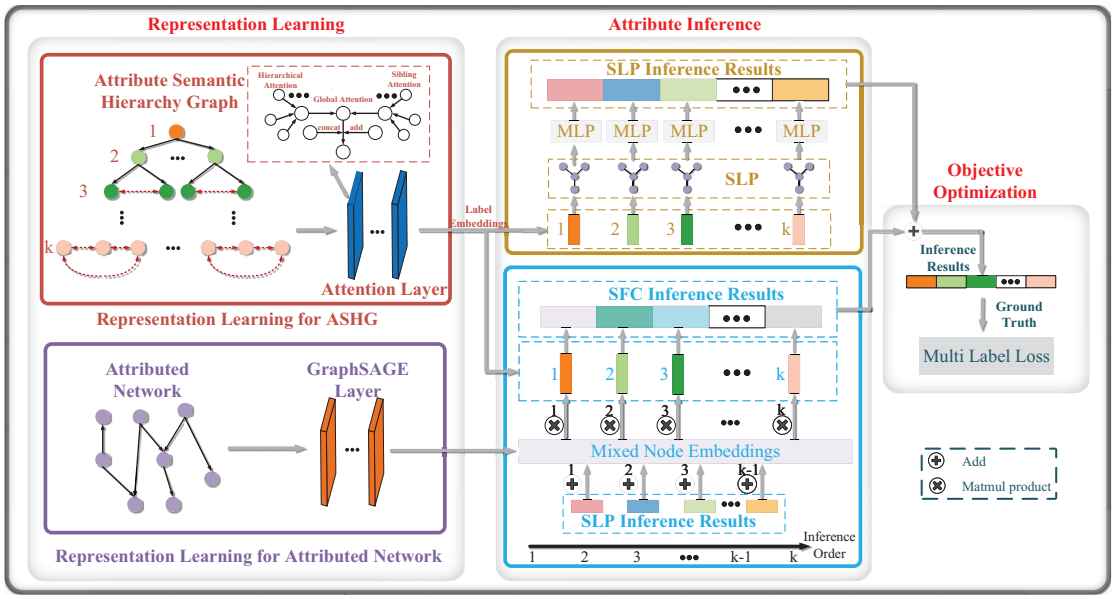


Fig. 2. Framework of HLI. Firstly, HLI learns node embeddings from ASHG and attributed network utilizing the triple attention mechanism and GraphSage respectively. Then, based on the attribution network node embeddings and label embeddings, HLI performs attribute inference using SLP and SFC. Finaly, we utilize multi-label loss to optimize our model.

high influence, and vice versa. So we use $\boldsymbol{\nu}_\Phi$ to realize this asymmetry. After obtaining the importance of all hierarchical neighbor nodes of node $h_i$, we normalize them using the softmax function to obtain the hierarchical-level attention score $\alpha_{ij}$. Then we can obtain the aggregated features of node $h_i$'s hierarchical neighbors:

$$\boldsymbol{z}_i^\Phi = \sigma\Big(\sum_{h_j \in N_i^\Phi \cup \{h_i\}} \alpha_{i,j} \cdot \boldsymbol{f}_j^{'}\Big) \tag{3}$$

where $\sigma$ denotes the activation function and $\boldsymbol{z}_i^\Phi \in \mathbb{R}^d$ denotes the embedding learned from node $h_i$'s hierarchical neighbors.

**Sibling-level Attention.** It is necessary to distinguish the differences between sibling labels under the same parent node. For example, "CV", "NLP" and "Time Series Analysis" are all subfields of "AI", and "CV" should be considered more in relation to NLP if the majority of authors in the academic network study the field of "Visual Question Answering". In this paper, we introduce sibling-level attention to learn the differences between sibling nodes.

The weights between node $h_i$ and its sibling neighbor node $h_j$ are defined as:

$$b_{i,j} = \boldsymbol{\nu}_\Psi \cdot \cos(\boldsymbol{f}_i^{'}, \boldsymbol{f}_j^{'}), h_j \in N_i^\Psi \cup \{h_i\} \tag{4}$$

$$\beta_{i,j} = \frac{\exp(b_{i,j})}{\sum_{h_k \in N_i^\Psi \cup \{h_i\}} \exp(b_{i,k})} \tag{5}$$

where $\boldsymbol{\nu}_\Psi \in \mathbb{R}^3$ is a trainable parameter and $N_i^\Psi$ denotes the sibling neighbor nodes of node $h_i$. $b_{i,j} \in \mathbb{R}$ denotes the importance of node $h_i$'s sibling neighbor $h_j$ to node $h_i$. After obtaining the importance of all the sibling neighbor nodes of node $h_i$, we use the softmax function to normalize all weights in order to obtain the sibling-level attention score $\beta_{i,j}$. Then

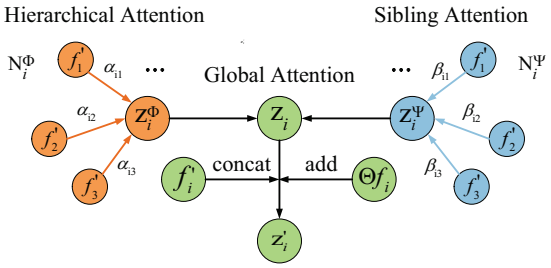


Fig. 3. Feature aggregation process with attention mechanism. Hierarchical-level and sibling-level attention are applied to node $h_i$'s hierarchical neighbors $N_i^\Phi$ and sibling neighbors $N_i^\Psi$ to obtain $\boldsymbol{z}_i^\Phi$ and $\boldsymbol{z}_i^\Psi$, respectively. $\boldsymbol{z}_i^\Phi$ and $\boldsymbol{z}_i^\Psi$ are finally aggregated by global-level attention to obtain node $h_i$'s updated feature $\boldsymbol{z}_i^{'}$.

we can obtain the aggregated features of node $h_i$'s sibling neighbors:

$$\boldsymbol{z}_i^\Psi = \sigma\Big(\sum_{h_j \in N_i^\Psi \cup \{h_i\}} \beta_{i,j} \cdot \boldsymbol{f}_j^{'}\Big) \tag{6}$$

where $\boldsymbol{z}_i^\Psi \in \mathbb{R}^d$ denotes the embedding learned from node $h_i$'s sibling neighbors.

**Global-level Attention.** To obtain the final embedding of node $h_i$, we need to combine the embedding learned from its hierarchical neighbors with the embedding learned from its sibling neighbors. To retain more information, we directly concatenate the two embeddings to obtain the final representation:

$$\boldsymbol{z}_i = \boldsymbol{W}_g \cdot (\boldsymbol{z}_i^\Phi || \boldsymbol{z}_i^\Psi) \tag{7}$$

where $\boldsymbol{W}_g \in \mathbb{R}^{d \times 2d}$. To stabilize the learning process for the three attention mechanisms, we used multi-head mechanism.



Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on August 29,2025 at 16:59:59 UTC from IEEE Xplore. Restrictions apply.

Specifically, we repeated the above process $K$ times and then averaged the $K$ results to obtain the final results:

$$\boldsymbol{z}_i = \sum_{t=1}^{K} (\boldsymbol{W}_g \cdot (\boldsymbol{z}_i^{\Phi} || \boldsymbol{z}_i^{\Psi})) \tag{8}$$

Finally, considering the importance of the feature of the node itself, the feature of node $h_i$ are updated as follows:

$$\boldsymbol{z}_i^{'} = \sigma(\boldsymbol{M}[\boldsymbol{z}_i || \boldsymbol{f}_i^{'}] + \boldsymbol{\Theta} \boldsymbol{f}_i) \tag{9}$$

where $\boldsymbol{z}_i^{'} \in \mathbb{R}^d$ denotes the final embedding of node $h_i$. $\boldsymbol{M} \in \mathbb{R}^{d \times 2d}$ and $\boldsymbol{\Theta} \in \mathbb{R}^{d \times b}$ are learnable weights. The reason to consider adding $\boldsymbol{\Theta} \boldsymbol{f}_i$ is because it ensures that the output of each node is different to alleviate over-smoothing. To better understand the execution process of attention, we have made a brief illustration in Fig. 3.

*2) Attributed Network Representation Learning:* Attributed networks are usually large in size. GraphSAGE [12] solves the problem of GCN [13] memory explosion by neighbor sampling and is suitable for large-scale networks. Therefore, GraphSAGE is used in this paper for representation learning of attributed networks. Let the initial feature of node $v_i$ in $G = (V, E, \boldsymbol{X}, \Sigma)$ is $\boldsymbol{x}_i$. After multiple layers of message passing and aggregation, the features $\boldsymbol{X} \in \mathbb{R}^{n \times f}$ of all nodes in $G$ are updated to $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times d}$.

### *B. Attribute Inference*

In this section, we introduce the mechanisms used for attribute inference. Let ASHG $H$ output be $\boldsymbol{Z} \in \mathbb{R}^{|\Sigma| \times d}$ and attributed network $G$ output be $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times d}$ after the node features of $H$ and $G$ have been updated, where $d$ is the embedding dimension. We hypothesize that attribute inference consists of two components: **semantic fully-connected layer inference (SFC)** and **semantic label propagation (SLP)**, where the latter complements the former and provides a priori knowledge to the former.

**Semantic Fully-Connected Layer Inference.** $\boldsymbol{Z} \in \mathbb{R}^{|\Sigma| \times d}$ is the embedding matrix of all attribute labels in ASHG, which contains rich label semantic information. Thus, similar to the fully-connected layer classifier $\boldsymbol{y} = \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}$, in this paper we use the attribute labels embedding $\boldsymbol{Z}$ as weights for attribute inference. Specifically, we treat the label embedding $\boldsymbol{Z}_{\Sigma_i} \in \mathbb{R}^{|\Sigma_i| \times d}$ of level $i$ in ASHG as a classifier and multiply it with $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n \times d}$ to obtain the prediction scores $\boldsymbol{O}_i^{sfc} \in \mathbb{R}^{n \times |\Sigma_i|}$ for level $i$ of all nodes in $G$:

$$\boldsymbol{O}_i^{sfc} = \boldsymbol{\mathcal{X}} \boldsymbol{Z}_{\Sigma_i}^{\top},\ 1 \le i \le k \tag{10}$$

where $|\Sigma_i|$ denotes the number of labels in level $i$ of ASHG.

**Semantic Label Propagation.** Although the attribute label embedding $\boldsymbol{Z} \in \mathbb{R}^{|\Sigma| \times d}$ contains rich label semantic information, this semantic information is only extracted based on ASHG and does not consider the interactions between entities in the attributed network $G$. SFC simply multiplies the two embeddings and does not consider the naive assumption that the labels of adjacent entities in $G$ should be more similar.

Label propagation (LP) [14] is a classical semi-supervised learning model based on graph structure, which is based on a simple assumption that two nodes on the same edge tend to have similar labels. We assume that the initialized label matrix of all nodes in $G$ at level $i$ is $\boldsymbol{Y}_i^{(0)}$. Formally, the $t+1$-th iteration of label propagation is defined as follows:

$$\boldsymbol{Y}_i^{(t+1)} = \lambda \cdot \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{Y}_i^{(t)} + (1-\lambda) \boldsymbol{Y}_i^{(t)} \tag{11}$$

where $\boldsymbol{A}$ and $\boldsymbol{D}$ denote the adjacency matrix and the diagonal degree matrix of $G$, respectively. $\lambda$ is a hyper-parameter controlling the smoothness of node updates. Let $\boldsymbol{Y}_i^{(t)}(v_j) \in \mathbb{R}^{|\Sigma_i|}$ denote the predicted score of node $v_j \in V$ in $\boldsymbol{Y}_i^{(t)}$. In this paper, we use label semantic embeddings $\boldsymbol{Z}$ instead of traditional one-hot encoding [15] to initialize the label matrix, and then use MLP to obtain the prediction scores when the propagation is finished. This way can effectively utilize meaningful label semantic information, and the model is trainable, so it can effectively alleviate the disadvantages of iterative instability and low accuracy. Specifically, the label matrix of level $i$ of all nodes in $G$ is initialized as $\tilde{\boldsymbol{Y}}_i^{(0)}$:

$$\tilde{\boldsymbol{Y}}_i^{(0)}(v_j) = \begin{cases} \boldsymbol{Z}_{\Sigma_i}(v_j) \in \mathbb{R}^d,\ \forall v_j \in V_s. \\ (0, ...0, ...0) \in \mathbb{R}^d,\ \forall v_j \in V \setminus V_s \end{cases} \tag{12}$$

where $\boldsymbol{Z}_{\Sigma_i}(v_j)$ denotes the embedding of the label of the $i$-th level of node $v_j \in V_s$. After $K_p$ iterations, we use a MLP to obtain the prediction scores of all nodes as follows:

$$\boldsymbol{O}_i^{slp} = \mathrm{softmax}(\mathrm{MLP}(\tilde{\boldsymbol{Y}}_i^{(K_p)})),\ 1 \le i \le k \tag{13}$$

where $\boldsymbol{O}_i^{slp} \in \mathbb{R}^{n \times |\Sigma_i|}$ denotes the prediction scores of the $i$-th level of all nodes in $G$.

**Combination and Chain Residual Inference.** Now we will combine SFC and SLP for full hierarchical attribute label inference. Since the inference results of the upper level can provide guidance to the lower level inference process, we follow top-down hierarchical inference. Specifically, the final prediction scores $\boldsymbol{O}_1 \in \mathbb{R}^{n \times |\Sigma_1|}$ for the first level of all nodes in $G$ are obtained by weighted summation of the above two components:

$$\boldsymbol{O}_1 = (1-\alpha) \boldsymbol{O}_1^{sfc} + \alpha \boldsymbol{O}_1^{slp} \tag{14}$$

where $\alpha$ is the hyperparameter that controls the proportion of semantic label propagation. Next, the prediction score $\boldsymbol{O}_i \in \mathbb{R}^{n \times |\Sigma_i|}$ of the $i$-th ($1 < i \le k$) level is defined as follows:

$$\boldsymbol{O}_i = (1-\alpha)(\tilde{\boldsymbol{O}}_i^{sfc}) + \alpha \boldsymbol{O}_i^{slp},\ \tilde{\boldsymbol{O}}_i^{sfc} = \tilde{\boldsymbol{\mathcal{X}}}_i \boldsymbol{Z}_{\Sigma_i}^{\top} \tag{15}$$

Unlike $\boldsymbol{\mathcal{X}}$ in the calculation of $\boldsymbol{O}_1^{sfc} = \boldsymbol{\mathcal{X}} \boldsymbol{Z}_{\Sigma_1}^{\top}$, the inference results of level $i-1$ are incorporated in $\tilde{\boldsymbol{\mathcal{X}}}_i$. Specifically, the expression of $\tilde{\boldsymbol{\mathcal{X}}}_i$ is as follows:

$$\tilde{\boldsymbol{\mathcal{X}}}_i = (1-\alpha_i-\beta) \boldsymbol{\mathcal{X}} + \alpha_i \boldsymbol{W}_i \tilde{\boldsymbol{Y}}_{i-1}^{(K_p)} + \beta \boldsymbol{W}_0 \boldsymbol{X} \tag{16}$$

where $\alpha_i$ and $\beta$ are two hyperparameters. The three components of $\tilde{\boldsymbol{\mathcal{X}}}_i$ are as follows: 1) $(1-\alpha_i-\beta)\boldsymbol{\mathcal{X}}$. 2) $\alpha_i \boldsymbol{W}_i \tilde{\boldsymbol{Y}}_{i-1}^{(K_p)}$. This component represents the result of the semantic propagation from the upper level, which is used to guide the inference in



Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on August 29,2025 at 16:59:59 UTC from IEEE Xplore. Restrictions apply.

TABLE I
STATISTICS OF DATASETS

| Dataset | Graph | Nodes | Edges | Levels | Labels |
|---|---|---|---|---|---|
| IS | Attri. Net. | 17225 | 180378 | 4 | 1+10+29+82 |
| | ASHG | 122 | 642 | | |
| ACS | Attri. Net. | 33332 | 451150 | 4 | 1+8+41+213 |
| | ASHG | 263 | 2390 | | |

this level. The larger $\alpha_i$ means that the inference process of the lower level is more affected by the upper level. 3) $\beta \boldsymbol{W}_0 \boldsymbol{X}$. $\boldsymbol{X}$ represents the initial features of all nodes in attributed network, and this component will be added to $\tilde{\boldsymbol{\mathcal{X}}}_i$ as the initial residual to prevent over-smoothing. In our experiments, we let $\alpha = 0.3$, $\alpha_i = 0.1$ and $\beta = 0.2$.

### C. Objective

Considering label inference as essentially a classification task, we consider minimizing the cross-entropy loss function:

$$\mathcal{L}_s = \sum_{i=1}^{k} \mathrm{CrossEntropyLoss}(\boldsymbol{O}_i, \boldsymbol{y}_i) \tag{17}$$

where $\boldsymbol{y}_i$ represents the ground truth of the $i$-th level of labeled nodes in attributed network. Eventually, we define the total loss by linearly combining these two component losses: $\mathcal{L} \leftarrow \mathcal{L}_s + \gamma \mathcal{L}_{reg}$, where $\gamma \mathcal{L}_{reg}$ denotes the regularization term to alleviate overfitting. We optimize using the Adam optimizer to update all the parameters mentioned earlier.

## IV. EXPERIMENTS

### A. Experimental Setup

*1) Dataset:* The dataset used in this article is an updated version of the public Amazon reviews dataset[1] released in 2014. In this study, we create attributed network and ASHG using product metadata. Our task is to infer the multi-level attribute labels of the product nodes in the attributed network.

Nodes in the attribute network denote products, and links denote two products being purchased together. Nodes in ASHG denote the category labels of the products. We use the Hashing Vectorizer algorithm to extract the feature vectors from the product descriptions. We utilize node2vec [16] to obtain the initial feature vectors for every node in ASHG. The statistical information is shown in Table I. The abbreviations "IS" and "ACS" stand for "Industrial and Scientific" and "Arts Crafts and Sewing" respectively, representing two different broad categories. In Table I, "1+10+29+82" indicates that the number of labels in each level of the ASHG is 1, 10, 29 and 82, respectively.

*2) Evaluation Metrics:* Let $V_u = V \setminus V_s$ denotes the set of nodes with unknown attribute labels in the attributed network. We used Accuracy (Acc), H-Precision (HP), H-Recall (HR), H-F1 (HF1), Jaccard Distance (JD) and Hamming Loss (HL) to evaluate the performance of all models.

$$\mathrm{Acc} = \frac{1}{|V_u|} |\{v_i | v_i \in V_u \wedge P_{v_i} = T_{v_i}\}| \tag{18}$$

$$\mathrm{HP} = \frac{1}{|V_u|} \sum_{v_j \in V_u} \frac{1}{k} \sum_{i=1}^{k} \frac{|P_i^{v_j} \cap Y_i^{v_j}|}{|P_i^{v_j}|} \tag{19}$$

$$\mathrm{HR} = \frac{1}{|V_u|} \sum_{v_j \in V_u} \frac{1}{k} \sum_{i=1}^{k} \frac{|P_i^{v_j} \cap Y_i^{v_j}|}{|Y_i^{v_j}|} \tag{20}$$

$$\mathrm{HF1} = \frac{2 \times \mathrm{HP} \times \mathrm{HR}}{\mathrm{HP} + \mathrm{HR}}, \ \mathrm{JD} = \frac{1}{|V_u|} \sum_{v_i \in V_u} \frac{|P_{v_i} \cap Y_{v_i}|}{|P_{v_i} \cup Y_{v_i}|} \tag{21}$$

$$\mathrm{HL} = \frac{1}{|V_u|} \sum_{v_i \in V_u} \frac{|\mathrm{XOR}(P_{v_i}, Y_{v_i})|}{k} \tag{22}$$

where $P_{v_i}$ and $T_{v_i}$ ($|P_{v_i}| = |T_{v_i}| = k$) are the set of predicted result and ground-truth attribute labels of $v_i \in V_u$ separately. $P_i^{v_j}$ is the set consisting of the attributes predicted for $v_j$ in level $i$ and all their ancestor attributes, similarly $Y_i^{v_j}$ is the set of ground-truth attributes in level $i$ and all their ancestor attributes. Notably, $|Y_i^{v_j}| = |P_i^{v_j}|$, i.e., HP=HR=HF1. Therefore, in the following experiments, we only show the calculation results of HF1. For all metrics except Hamming Loss, a larger value means better performance.

*3) Comparative models and Parameter Settings:* It is worth noting that the number of first level labels is 1 for both datasets, so we only need to predict the last three labels for each node in the attributed network. We compare our model HLI with 5 methods including the traditional machine learning model Decision Tree (Dec), 2 classical GNN models (GCN [13] and GAT [17]) and 2 HMC methods (HMC-LMLP [4] and MLI [7]). For GCN and GAT, after using GNN to obtain the embeddings of the nodes in attributed network, we input them into three MLPs to obtain three labels for each node. More detailed parameter settings can be found in github[2].

### B. Full-supervised Attribute Label Inference

We first evaluate the performance of HLI in the task of full-supervised hierarchical attribute label inference. For each dataset, we randomly split the nodes into 60%, 20%, and 20% for training, validation and testing.

Table II reports the experimental results on the 2 datasets. We observe that HLI achieves the best performance on all metrics for both datasets. Dec and HMC-LMLP are all less effective as they can only utilize the node features and cannot extract structural information. Due to its ability to efficiently extract both structural and feature information simultaneously, the GNN-based method performs better overall. HLI extracts fine-grained semantics using attention mechanisms, while enhancing the interaction between the label and attributed networks, and therefore performs best. Notably, MLI exploits both the attributed network structure and the coarse-grained semantics of the labels and performs second only to HLI, indicating the importance of exploiting the semantics of the labels.

### C. Semi-supervised Attribute Label Inference

In real life, only a small fraction of the nodes in the attributed network may have labels due to data privacy and other

[1]https://nijianmo.github.io/amazon/index.html

[2]https://github.com/ki-ljl/HLI



Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on August 29,2025 at 16:59:59 UTC from IEEE Xplore. Restrictions apply.

TABLE II
FULL-SUPERVISED HIERARCHICAL ATTRIBUTE LABEL INFERENCE RESULTS (%)

| Dataset | Metric | Dec | GCN | GAT | HMC-LMLP | MLI | HLI |
|---|---|---|---|---|---|---|---|
| IS | Acc | 58.43 | 77.76 | 76.80 | 63.11 | 79.18 | **82.32** |
| | HF1 | 83.27 | 94.41 | 94.12 | 87.12 | 95.14 | **95.72** |
| | JD | 67.72 | 84.79 | 84.14 | 72.24 | 86.47 | **88.36** |
| | HL | 56.93 | 24.71 | 25.83 | 48.57 | 21.50 | **18.61** |
| ACS | Acc | 62.01 | 69.30 | 71.28 | 61.73 | 75.67 | **77.48** |
| | HF1 | 85.65 | 91.92 | 92.17 | 88.33 | 93.37 | **94.02** |
| | JD | 71.26 | 79.24 | 80.25 | 72.60 | 83.49 | **84.62** |
| | HL | 49.73 | 33.95 | 32.56 | 46.38 | 27.01 | **25.26** |

TABLE III
SEMI-SUPERVISED HIERARCHICAL ATTRIBUTE LABEL INFERENCE RESULTS (%)

| Dataset | Metric | Dec | GCN | GAT | HMC-LMLP | MLI | HLI |
|---|---|---|---|---|---|---|---|
| IS | Acc | 43.10 | 70.25 | 66.18 | 50.75 | 73.95 | **79.35** |
| | HF1 | 77.24 | 92.59 | 91.91 | 84.93 | 93.58 | **94.58** |
| | JD | 55.89 | 80.11 | 77.53 | 64.80 | 82.91 | **86.04** |
| | HL | 77.07 | 32.35 | 36.08 | 59.25 | 27.27 | **22.62** |
| ACS | Acc | 35.03 | 64.85 | 64.25 | 30.75 | 65.83 | **67.15** |
| | HF1 | 76.31 | 90.88 | 89.64 | 82.70 | 90.73 | **91.26** |
| | JD | 51.59 | 76.16 | 74.81 | 53.85 | 77.17 | **78.25** |
| | HL | 83.13 | 39.08 | 42.38 | 44.98 | 37.08 | **35.03** |

reasons. To explore the performance of HLI in this scenario, we conduct semi-supervised experiments. Specifically, we performe a standard fixed train/validation/test split on the two datasets, with 100 nodes per class (fourth level of labels) for training, 2,000 nodes for validation and 4,000 nodes for testing.

Table III reports the experimental results on the 2 datasets. As can be seen, in the semi-supervised scenario, HLI still performs the best among all metrics in both datasets. Although fewer labels are available for model training, HLI can still use ASHG (SFC and SLP) to indirectly leverage more label information, so HLI can still achieve a solid performance.

### D. Ablation Experiment

To demonstrate the effectiveness of the components in HLI, we created the following 4 variants: (1) $\mathbf{HLI}_h$: which does not include hierarchical-level attention. (2) $\mathbf{HLI}_s$: which does not include sibling-level attention. (3) $\mathbf{HLI}_{sfc}$: which does not use SFC and only performs hierarchical attribute inference based on SLP. (4) $\mathbf{HLI}_{slp}$: which does not use SLP and only performs hierarchical attribute inference based on SFC.

Fig. 4 reports the results of the ablation experiments. We can observe that HLI outperforms the 4 variants mentioned above, which indicates that all four modules of HLI can individually improve the model performance.

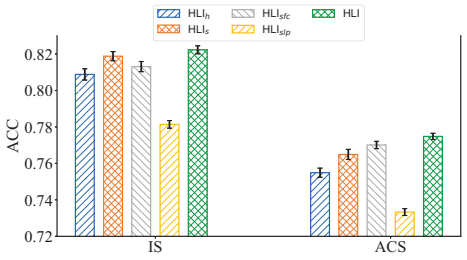


Fig. 4. Ablation study in full-supervised scenario.

## V. CONCLUSION

In this paper, we propose a novel hierarchical attribute inference model for attributed networks. We propose to use a triple attention mechanism to mine more fine-grained label semantics than previous work that only exploits structural information between labels. We propose both SFC and SLP approaches to hierarchical attribute inference and combine them to perform top-down hierarchical attribute inference. Extensive experiments demonstrate the effectiveness of our model.


## ACKNOWLEDGEMENT

This work was supported by the State Key Laboratory of Communication Content Cognition Funded Project No. A32003, National Natural Science Foundation of China No. U22A2025 and No. 61972275.


## REFERENCES


[1] F. Zarrinkalam, M. Kahani, and E. Bagheri, "Mining user interests over active topics on social networks," *Information Processing & Management*, vol. 54, no. 2, pp. 339–357, 2018.

[2] C. Budak, A. Kannan, R. Agrawal, and J. Pedersen, "Inferring user interests from microblogs," *AAAI ICWSM*, 2014.

[3] J. Xu and T.-C. Lu, "Inferring user interests on tumblr," in *Social Computing, Behavioral-Cultural Modeling, and Prediction: 8th International Conference, SBP 2015, Washington, DC, USA, March 31-April 3, 2015. Proceedings 8*, pp. 458–463, Springer, 2015.

[4] R. Cerri, R. C. Barros, and A. C. de Carvalho, "Hierarchical multi-label classification for protein function prediction: A local approach based on neural networks," in *2011 11th International Conference on Intelligent Systems Design and Applications*, pp. 337–343, IEEE, 2011.

[5] F. K. Nakano, R. Cerri, and C. Vens, "Active learning for hierarchical multi-label classification," *Data Mining and Knowledge Discovery*, vol. 34, pp. 1496–1530, 2020.

[6] Y. Yan and S.-J. Huang, "Cost-effective active learning for hierarchical multi-label classification.," in *IJCAI*, pp. 2962–2968, 2018.

[7] H. Zhang, Y. Yang, X. Wang, H. Gao, and Q. Hu, "Mli: A multi-level inference mechanism for user attributes in social networks," *ACM Transactions on Information Systems (TOIS)*, 2022.

[8] S. Gopal and Y. Yang, "Recursive regularization for large-scale classification with hierarchical and graphical dependencies," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 257–265, 2013.

[9] Y. Mao, J. Tian, J. Han, and X. Ren, "Hierarchical text classification with reinforced label assignment," *arXiv preprint arXiv:1908.10419*, 2019.

[10] J. Wu, W. Xiong, and W. Y. Wang, "Learning to learn and predict: A meta-learning approach for multi-label classification," *arXiv preprint arXiv:1909.04176*, 2019.

[11] M. Romero, J. Finke, and C. Rocha, "A top-down supervised learning approach to hierarchical multi-label classification in networks," *Applied Network Science*, vol. 7, no. 1, pp. 1–17, 2022.

[12] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[13] M. Welling and T. N. Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[14] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," 2002.

[15] C. Yang, J. Liu, and C. Shi, "Extract the knowledge of graph neural networks and go beyond it: An effective knowledge distillation framework," in *Proceedings of the web conference 2021*, pp. 1227–1237, 2021.

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

[17] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, p. 20, 2017.




Authorized licensed use limited to: TIANJIN UNIVERSITY. Downloaded on August 29,2025 at 16:59:59 UTC from IEEE Xplore. Restrictions apply.