Meta-GNN: Metagraph Neural Network for Semi-supervised learning in Attributed Heterogeneous Information Networks

Aravind Sankar[1], Xinyang Zhang[1], and Kevin Chen-Chuan Chang[1]

[1]Department of Computer Science, University of Illinois, Urbana-Champaign, USA
{*asankar3,xz43,kcchang*}*@illinois.edu*

***Abstract*—Heterogeneous Information Networks (HINs) comprise nodes of different types inter-connected through diverse semantic relationships. In many real-world applications, nodes in information networks are often associated with additional attributes, resulting in Attributed HINs (or AHINs). In this paper, we study semi-supervised learning (SSL) on AHINs to classify nodes based on their *structure*, *node types* and *attributes*, given limited supervision. Recently, Graph Convolutional Networks (GCNs) have achieved impressive results in several graph-based SSL tasks. However, they operate on homogeneous networks, while being completely agnostic to the semantics of typed nodes and relationships in real-world HINs.**

In this paper, we seek to bridge the gap between semantic-rich HINs and the neighborhood aggregation paradigm of graph neural networks, to generalize GCNs through metagraph semantics. We propose a novel *metagraph* convolution operation to extract features from local metagraph-structured neighborhoods, thus capturing semantic higher-order relationships in AHINs. Our proposed neural architecture *Meta-GNN* extracts features of diverse semantics by utilizing multiple metagraphs, and employs a novel metagraph-attention module to learn personalized metagraph preferences for each node. Our semi-supervised node classification experiments on multiple real-world AHIN datasets indicate significant performance gains of 6% Micro-F_1 on average over state-of-the-art AHIN baselines. Visualizations on metagraph attention weights yield interpretable insights into their relative task-specific importance.

I. Introduction

Graph-based semi-supervised learning (SSL) is an important machine learning paradigm that aims to classify unlabeled nodes in a graph, given a small subset of labeled nodes [1]. The key objective is to avoid the expensive cost of human data labeling by leveraging more accessible unlabeled data. Graph-based SSL is prevalent in various data mining applications such as profiling users in social networks [2], [3], categorizing publications in bibliographic networks [4] and modeling user interests in recommender systems [5], [6].

Classical SSL techniques employ smoothness assumptions to propagate labels through explicit graph-based regularization [1] under the hypothesis that directly linked nodes share labels (homophily). However, links in real-world networks often go beyond similarity to indicate semantics such as relationships between various objects [7]. Recent advances in graph convolutional networks (GCNs) [8]–[10] jointly consider graph structure and node attributes. Specifically, GCNs [9] characterize a *target* node of interest through local neighborhood aggregation of node attributes and have achieved impressive gains in several SSL tasks. However, GCNs are designed for homogeneous networks - which are representative of singular type of nodes and relationships.

Many real-world applications manifest as Heterogeneous Information Networks (HINs) containing nodes of multiple types inter-connected in diverse semantic relationships. For instance, bibliographic networks (such as DBLP) with author, paper, and venue nodes, include co-authorships, co-citations, publishing in the same venue, etc., as different semantic relationships. HINs often include node attributes, *e.g.*, “users” in Facebook have age, gender, location and employer attributes, while “publications” in DBLP are described through their text content. Although discrete attributes such as location and publication venue may be considered as distinct node types, complex numerical attributes and text content preclude such type representations. To enrich the information content of HINs with attributes, we consider *Attributed Heterogeneous Information Networks* or AHINs [11].

Semi-supervised learning in AHINs is significantly more challenging in comparison to both homogeneous networks and HINs. For instance, to classify an author in DBLP, her co-authors and published venues illustrate *context* nodes that both provide relevant features. Yet, these context nodes possess diverse attributes and play different roles due to (a) diverse node types such as venues versus co-authors or (b) varying structural orientations amongst the same node type, such as cited versus referenced publications. Thus, it is necessary to consider the *interplay of structure and attributes, in conjunction with HIN semantics*, to accurately extract relevant features from local neighborhoods for classification.

Inspired by the recent success of GCNs, we aim to unify the local neighborhood aggregation paradigm of Graph Neural Networks with semantic-rich AHINs. We generalize graph convolutions to AHINs through *metagraphs*. In HINs, metapaths and metagraphs are widely used to encode different semantic relationships, with applications in various data mining tasks [4], [12]. We propose metagraph convolutions to aggregate features from local neighborhoods specified by the

ASONAM '19, August 27-30, 2019, Vancouver, Canada
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6868-1/19/08/...$15.00
http://dx.doi.org/10.1145/3341161.3342859

metagraph structure. We identify two key interesting insights enabled by metagraph convolutions:

- **Semantic High-order locality:** Metagraphs specify semantic relationships via certain sub-substructures, thus providing a principled framework capturing *semantic* higher-order locality (in contrast to immediate neighbors), such as authors (target) connecting to other authors (context) through coauthored papers.
- **Precise semantic role:** Metagraphs enable accurate semantic role discrimination of local context nodes based on their node types and structural connection patterns, such as roles of co-authors (context) and publication venues (context) in author classification.

HINs comprise multiple metagraphs expressing diverse semantic relationships with varying task-specific relevance, *e.g.*, metagraphs relating authors to venues (M_1 in Fig. 1(a)) are strongly indicative of their research areas, while co-authorship relations (M_3 in Fig. 1(a)) are more informative in identifying research groups. The relevance of each metagraph may further depend on the specific node under consideration. This poses a new challenge of automatically learning personalized task-specific metagraph preferences per node.

To this end, we introduce a novel metagraph-attention module to learn personalized metagraph preferences. Attention mechanisms have recently achieved great success in many natural language processing tasks, such as machine translation [13], etc. The key objective of attentions is to learn a function that focuses on the most relevant parts of the input, to compute an aggregate representation. In contrast to the usual application of attention at the granularity of nodes [10], we apply attention at a higher abstraction level (metagraphs) to enable dual benefits: (a) interpretable insights into their discriminative power, and (b) computational efficiency in comparison to node-level attention. We summarize our key contributions below:

- We introduce a novel generalization of graph convolutions to Attributed Heterogeneous Information Networks, by utilizing metagraphs to define the local context or receptive field around a target node of interest.
- We propose a novel neural architecture *Meta-GNN* that employs multiple metagraph convolutional layers, along with an attention mechanism to learn personalized metagraph preferences for each node.
- Our experiments on multiple real-world datasets demonstrate the effectiveness of *Meta-GNN* in achieving state-of-the-art semi-supervised classification performance.

II. Related work

Our work is related to graph-based SSL literature in both homogeneous and heterogeneous information networks.

Graph-based SSL (Homogeneous): Semi-supervised learning on graphs is a well-studied problem in recent years with two broad classes of techniques: (a) explicit graph-based regularization methods and (b) graph embedding methods. Classical methods in the first category include variants of label propagation [1] and manifold regularization [14]. A thorough survey of classical SSL methods can be found at [15]. Graph embedding methods are designed to learn unsupervised node representations by predicting local graph neighborhoods [16]–[18], followed by downstream supervised classification using the learnt embeddings. Though these approaches are universal, the embeddings are learnt independent of the underlying learning task.

TABLE I. Comparison of *Meta-GNN* with existing on the aspects of node attributes, node types and node labels

Aspect	Attributes	Types	Labels
GCN [9], GAT [10]	Yes	No	Yes
metapath2vec [23]	No	Metapath	No
LP-metagraph [24]	No	Metagraph	Yes
HCC [25], CLN [26]	Partial	Metapath	Yes
Meta-GNN	**Yes**	**Metagraph**	**Yes**

Recently, graph neural networks have been introduced to solve graph classification [19] and semi-supervised node classification tasks [8], [9], [20]–[22]. Our work is closely related to GCN [9], which aggregates neighborhood features with equal importance. GraphSAGE [21] generalizes GCN to explore a large family of expressive aggregators, while GAT [10] employs attention mechanisms to learn different weights for context nodes via pairwise feature correlations. However, all these techniques are applicable only on homogeneous networks by design. In contrast, we focus on AHINs, where modeling heterogeneous semantics is crucial to learn suitable neighborhood aggregation functions.

Graph-based SSL (Heterogeneous): Metapaths [7] and metagraphs [27] encode HIN semantics to generalize label propagation methods to HINs [24], [28]. metapath2vec [23] and others [29], [30] learn unsupervised node representations by utilizing metapath-guided node similarity measures. HCC [25] and Column Networks (CLN) [26] use metapaths along with node attributes of the target type, for node classification (often called collective classification).

However, all these techniques are limited to modeling semantic node proximity through metapaths or metagraphs, and can at best use attributes of a single node type (such as CLN), for node classification, *i.e.*, existing methods cannot incorporate the attributes of all node types in general AHINs to learn prediction models. In contrast, *Meta-GNN* generalizes graph convolutions by extracting local features from diverse local neighborhoods defined by metagraphs.

We qualitatively compare *Meta-GNN* with representative methods across all settings on three key distinct aspects of our problem: (a) node attributes (b) node types (HIN semantics) and (c) node labels (task supervision). Table. I compares *Meta-GNN* with GCN (homogeneous graph CNN), metapath2vec (HIN embedding), HCC, CLN (HIN metapath-based SSL) and LP-metagraph (HIN metagraph-based SSL). Our proposed model *Meta-GNN* captures HIN semantics through a novel metagraph convolutional framework that leverages node attributes for semi-supervised learning.

III. Problem Formulation

In this section, we introduce preliminary concepts and formally define the problem of semi-supervised learning on Attributed Heterogeneous Information Networks (AHINs).

Definition 1: An **Attributed Heterogeneous Information Network** (AHIN) is a graph $G = (V, E, X)$ with node type mapping: $l : V \mapsto \mathcal{L}$ where $V = \{v_1, ..., v_N\}$ is the set of N nodes, E is the set of links, $\mathcal{L}$ is the set of node types, and $X \in \mathbb{R}^{N \times D}$ is the attribute matrix describing all nodes.

Since each node type may belong to a different feature space, we concatenate the features of all types (with zero padding) to obtain a joint feature space with D attributes. Now, we introduce the concept of *metagraphs* in AHINs.

A metagraph is a subgraph pattern describing the relationship between a pair of *target* and *context* nodes, *e.g.*, metagraph M_3 (Fig. 1(a)) illustrates a co-author relationship of a target node A with context node A_2 through *auxiliary* paper node P_1. In contrast to existing metagraph definitions, we explicitly specify the connecting *auxiliary* nodes since they provide useful attributes for classification. In Fig. 1(a), the node subscript gives the node index, *e.g.*, P_1 and P_2 are two different nodes of type P. A metagraph M is a subgraph comprising a designated target node t_M, context node c_M and remaining auxiliary nodes B_M. We formally define metagraphs as follows:

*Definition 2: A **metagraph** M with **target** node t_M, **context** node c_M is defined as $M = (V_M, E_M, t_M, c_M, B_M)$ with node type mapping $l_M : V_M \mapsto \mathcal{L}$ where V_M is the set of nodes with $t_M, c_M \in V_M$, E_M is the set of links and $B_M = V_M - \{t_M, c_M\}$ is the set of auxiliary nodes.*

We define an instance S_u of metagraph M with target node u in G, as a subgraph induced by M with u as the target node. Fig. 1(b) illustrates an author node a characterized via different instances of metagraph M_1 in G with a as target. The formal definition of a metagraph instance is given by:

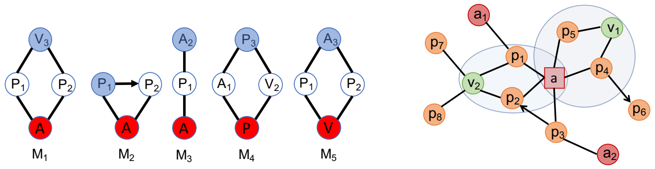
*Definition 3: A **metagraph instance** $S_u = (V_S, E_S)$ of metagraph M with target node u is a subgraph of G where $V_S \subseteq V$ and $E_S \subseteq E$, such that there exists a bijective node mapping $\psi_S : V_S \mapsto V_M$ satisfying conditions on (i) target node: $u \in V_S$, $\psi_S(u) = t_M$ (ii) node types: $\forall x \in V_S$, $l(x) = l_M(\psi_S(x))$ and (iii) edge set: $\forall x, y \in V_S$, $(x, y) \in E_S$ if and only if $(\psi_S(x), \psi_S(y)) \in E_M$.*

AHINs naturally contains multiple metagraphs to completely specify the associated semantics. In this paper (similar to most works in HINs), we assume a set of metagraphs as input, specified by domain experts familiar with the HIN schema, to capture the relationships between different node types. We now formally define the problem of semi-supervised node classification on AHINs below as:

*Definition 4: **Semi-supervised node classification in AHINs**. Given an AHIN $G = (V, E, X)$ with labeled nodes $\mathcal{Y}_L$, learn a model M to classify the nodes into K classes.*

IV. Metagraph Convolutional Neural Network

In this section, we review graph convolution networks [9] to present necessary background for metagraph convolutions.



(a) Metagraphs in DBLP

(b) Example graph

Fig. 1: **(a) Metagraphs (Author (A), Paper (P) and Venue (V)) with target and context nodes marked in red and blue respectively (b) Illustrating instances of M_1 for target a. Lower-case letters (*e.g.*, a) denote nodes in G while upper-case letters (*e.g.*, A) correspond to node types.**

A. Graph Convolutional Layer

Graph Convolutional Networks define *receptive fields* to extract features from local regions of interest around a target node, *e.g.*, GCN [9] defines receptive fields as the set of immediate first-order neighbors in G. A single GCN layer operates on the adjacency matrix A (representing G) and the attribute matrix X to extract and aggregate first-order neighborhood features by assigning equal importance to each neighboring node. The operation of a GCN layer is given as:

$$H = \sigma(\hat{A}XW) \quad \hat{A} = D^{-1/2}AD^{-1/2} + I_N$$

where X is the input attribute matrix, $W \in \mathbb{R}^{D \times F}$ is the weight transformation matrix, $\hat{A}$ is a normalized adjacency matrix, and $\sigma(\cdot)$ is a non-linear activation.

B. Metagraphs for Convolution

The heterogeneous neighborhood around a target node of interest u is generalized beyond immediate neighbors through *metagraph*-based receptive fields.

Heterogeneous Receptive Field: We define the heterogeneous receptive field around target node u *w.r.t* metagraph M as the set of all instances of M with u as the target. Metagraphs enable explicit description of multiple semantic relationships of a target node u in its local neighborhood. Here, the metagraph M serves as the *structure* or *template* that spans the neighborhood around u to define the M*-receptive field* through its different instances. For example, the receptive field around an author node a is characterized via different instances of metagraph M_1 in G with a as target (illustrated in Fig. 1(b)). We define an M-receptive field as:

*Definition 5: The M**-receptive field** around target node u is the set of all instances of metagraph M in G with target node u, denoted by I_u^M.*

In other words, heterogeneous M-receptive fields define semantic high-order neighbors, as specified by metagraph structures. In contrast to GCN [9] which uses homogeneous node sets as receptive fields, M-receptive fields contain nodes of different types in each heterogeneous metagraph instance. This poses new challenges since the nodes in each instance possess distinct semantics, which precludes the application of a naive type-agnostic aggregation.

Thus, it is necessary to clearly delineate the *roles* of different nodes in each instance of M, to accurately capture the

TABLE II. Notations

Symbol	Dimensions	Description
N	Scalar	Number of nodes
D	Scalar	Number of input features
F	Scalar	Number of filters per metagraph
K_M	Scalar	Number of unique semantic roles in metagraph M
U	Scalar	Number of metagraphs per layer
X	$N \times D$	Input Feature Matrix
H_l^j	$N \times F$	Activations at Conv unit j (metagraph M_j) in layer l
$\mathcal{A}^M$	$K_M \times N \times N$	Adjacency tensor for metagraph M
W^M	$(K_M{+}1){\times}D{\times}F$	Weight tensor for metagraph M

semantics of various node types. Furthermore, it is critical to note that even two nodes of the same type in a metagraph may not be equivalent, *e.g.,* the roles of nodes P_1 and P_2 relative to target A in metagraph M_2 (Fig. 1(a)), are quite different, despite being of the same type. To address this challenge, we define the concept of *semantic roles* to distinguish the roles of different nodes in a metagraph based on their types and structural orientations.

Semantic roles: The semantic roles of context and auxiliary nodes in metagraph M are determined by the concept of automorphic equivalence, *i.e.*, two nodes u and v (other than target t_M) are automorphically equivalent if exchanging the two nodes through permutations does not affect the relationships among all nodes in the graph. For instance, in Fig. 1(a), nodes P_1 and P_2 share the same semantic roles in M_1, while the roles of P_1 and P_2 in M_2 are distinct. We formally define the concept of semantic roles as:

Definition 6: Given metagraph $M = (V_M, E_M, t_M, c_M)$, nodes $u, v \in V_M - \{t_M\}$ share **semantic roles** *w.r.t.* target t_M if $\exists$ an automorphic mapping, $\Omega_M : V_M \mapsto V_M$ such that $\Omega_M(u) = v, \Omega_M(v) = u$ and $\Omega_M(x) = x \; \forall x \in V_M - \{u, v\}$.

The semantic roles of various nodes *w.r.t.* target node t_M in metagraph M can be easily deduced from their structure and types. Hence, we simply define a role mapping function $\phi_M : V_M - \{t_M\} \mapsto \{1, \ldots, K_M\}$ to return the semantic role *w.r.t* target t_M in metagraph M, where K_M is the number of unique semantic roles. Since roles are metagraph-specific, K_M is at most 1+$|B_M|$. Next, we define a compact tensor to represent node connectivity in G through metagraph M.

Metagraph-adjacency Tensor: $\mathcal{A}^M$ is a tensor of K_M matrices denoting the occurrences of nodes in each unique semantic role k over all instances of M in G. $\mathcal{A}^M_{kij}$ is the number of instances with v_j in role k and v_i as target:

$$\mathcal{A}^M_{kij} = \sum_{S_{v_i} \in I^M_{v_i}, v_j \in V_S - \{v_i\}} I\Big(\phi_M(\psi_S(v_j)) = k\Big)$$

where $I(\cdot)$ is the indicator function. We define a diagonal matrix $\mathcal{D}^M \in \mathbb{R}^{N \times N}$ for each metagraph M where each entry $\mathcal{D}^M_{ii}$ stores the number of metagraph instances with node i as target, *i.e.*, $\mathcal{D}^M_{ii} = |I^M_{v_i}| = L_i \; \forall 1 \leq i \leq N$.

C. Metagraph-based Convolution

In this section, we define metagraph convolutions by utilizing the concepts of M-receptive fields and semantic roles to extract local features around a target node of a specific type. For ease of explanation, we first consider a single metagraph M with target type $T = l(t_M)$ applied at a target node $v_i \in V$ with $l(v_i) = T$. We denote by F, the number of filters per metagraph. Now, we introduce a metagraph filter operating on a single feature ($D = 1$).

Metagraph Filter: A metagraph filter (on M) is defined by weight w_0 for target t_M and weight vector $\mathbf{w} \in \mathbb{R}^{K_M}$ for the K_M roles, *i.e.*, each weight in $\mathbf{w}$ differentiates the semantic roles of context and auxiliary nodes in M.

Metagraph Convolutional Unit: The features of all nodes connected through metagraph M are weighted according to their semantic roles and normalized by the diagonal matrix which reduces the bias introduced by highly connected nodes, resulting in the following expression:

$$h^M(v_i) = \sigma\left(w_0 x_i + \frac{1}{\mathcal{D}^M_{ii}} \sum_{j=1}^{N} \sum_{k=1}^{K_M} w_k \mathcal{A}^M_{kij} x_j\right) \qquad (1)$$

where x_i and x_j refer to the features of nodes v_i and v_j respectively, $h^M(v_i)$ is the output of convolution at node v_i and $\sigma(\cdot)$ is an activation function, such as $ReLU(\cdot) = \max(0, \cdot)$. Thus, the metagraph convolutional unit for metagraph M generates a new node representation by assigning different weights according to the semantic roles of context and auxiliary nodes in each instance, followed by a mean aggregation over multiple instances.

We extend Eqn. 1 for input matrix $X \in \mathbb{R}^{N \times D}$ with N nodes, D features and F filters per metagraph, to get:

$$H^M = \sigma\left(XW_0^M + (\mathcal{D}^M)^{-1} \sum_{k=1}^{K_M} \mathcal{A}^M_k X W^M_k\right) \qquad (2)$$

where W^M is the weight tensor for metagraph M and H^M is the output of the metagraph convolutional unit.

D. Semantic Metagraph Fusion

The node representations computed by metagraph convolutions across multiple metagraphs, must be suitably weighted according to the underlying learning task. An immediate solution is to use a set of U linear weights to combine the outputs of different metagraphs. However, this approach fails to learn personalized metagraph preferences.

We are inspired by recent advances in *attention* mechanisms [13], that enable focusing on the most relevant inputs, while learning an aggregate representation. We introduce a metagraph-attention module to semantically fuse the representations learnt from individual metagraphs by attending over the outputs of different metagraph convolutional units.

Metagraph-Attention: We use a scaled dot-product attention function to compute the output at node v_i, as given by:

$$h(v_i) = \sum_{k=1}^{U} \alpha_{k,i} h^k(v_i) \quad \alpha_{k,i} = \frac{\exp(e_{k,i})}{\sum_{j=1}^{U} \exp(e_{j,i})} \qquad (3)$$

where $e_{k,i} = a(h^k(v_i), z) = \frac{z^T h^k(v_i)}{\sqrt{|z|}}$ and $\alpha_{k,i}$ are normalized attentional co-efficients that indicate the importance of meta-

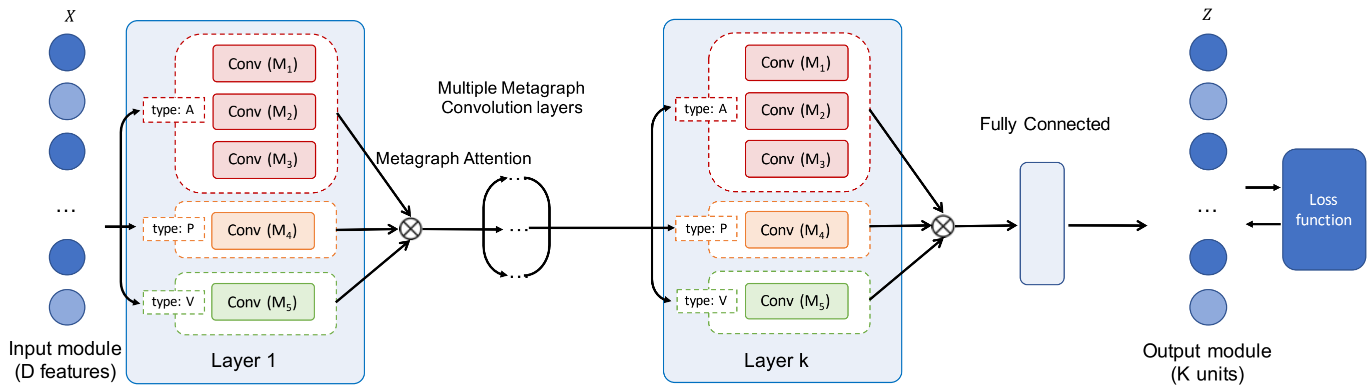


Fig. 2: Illustration of the deep neural architecture of *Meta-GNN* for metagraphs in Fig. 1(a)

graph M_k to node v_i, $h^k(v_i)$ is the output of convolutional unit for M_k (Eqn. 2), and z is an attention vector shared across all metagraphs. z can be viewed as a representation of a fixed semantic query "which is the most informative metagraph" compared against the features extracted from each metagraph. A higher value of $\alpha_{k,i}$ indicates greater importance of metagraph M_K for node v_i. Note that the metagraph attentional coefficients vary across nodes, thus learning personalized metagraph preferences.

E. *Meta-GNN Architecture*

Meta-GNN comprises multiple stacked convolutional layers followed by a fully connected layer. Each convolutional layer has U metagraph convolutional units, each corresponding to a metagraph. Fig. 2 illustrates the architecture of *Meta-GNN* for the sample metagraphs in Fig. 1(a). In each layer, the output of all convolutional units are fused via metagraph-attention, to feed as input to the next layer, *i.e.*, the output at layer l denoted by H_l, is computed by Eqn. 3 where the input to the first layer is the attribute matrix $H_0 = X$.

For a K-class node classification setting, the output layer has K units and applies the softmax activation function to obtain predictions $Z \in \mathbb{R}^{N\times K}$. We use the cross-entropy loss function for optimization, as described below:

$$L = -\sum_{l\in Y_L}\sum_{k=1}^{K} Y_{lk}\log Z_{lk} \tag{4}$$

where Y_L is the set of node indices that have labels. For multi-label classification, we instead use K sigmoid units and apply the binary cross-entropy loss function.

F. *Complexity analysis*

We analyze the complexity of *Meta-GNN* in two parts:

Pre-Computation of $\mathcal{A}^M$***:*** The Metagraph-Adjacency Tensor, which is independent of the architecture, is pre-computed for all metagraphs. In this paper, we focus on metagraphs of up to 3 nodes. The cost of computing $\mathcal{A}^M$ for triangles is $O(|E|^{1.5})$ [31]. For non-triangle 3-node metagraphs, each pair of neighbors can be examined for all nodes, giving a complexity of $\Theta(\sum_j d_j^2)$ (d_j is the degree of node v_j), with superior efficient algorithms in practice [32]. For larger metagraphs, subgraph matching can be used with approximate sampling strategies for practical efficiency.

Dataset	$\vert V\vert$	$\vert E\vert$	$\vert\mathcal{L}\vert$	Classes
DBLP-A	11,170	24,846	3	4
DBLP-P	35,770	131,636	3	10
Movie	10,441	99,509	4	6

TABLE III. Statistics of three AHIN datasets

Model Training: The complexity of a single layer is a function of the number of metagraphs U (typically < 5) and density of each $\mathcal{A}^M$, given by $O(\sum_{i=1}^{U}|\mathcal{A}^{M_i}|DF)$. In practice, the number of roles K_M is at most 3 and the role-specific matrices are sparser than the original adjacency matrix, giving an average-case complexity $O(U|E|DF)$. Thus, we observe linear scaling with U in comparison to GCN which has $O(|E|DF)$. An efficient implementation of *Meta-GNN* in Tensorflow [33] is publicly available[1].

V. Experiments

We conduct experiments on multiple real-world graph datasets to evaluate the performance of Meta-GNN on semi-supervised node classification in AHINs.

A. *Datasets*

We conduct experiments on three real-world datasets, whose statistics are shown in Table. III:

- **DBLP-A**: This is a bibliographic network composed of 3 node types: author (A), paper (P) and venue (V), connected by three link types: P→P, A-P and P-V. We use a subset of DBLP [7] with text attributes of papers to classify authors based on their research areas.
- **DBLP-P**: This dataset has the same schema as DBLP-A, but the task is to classify research papers into 10 categories, which are obtained from Cora [34].
- **Movie**: We use MovieLens [35] to create an AHIN with 4 node types: movie (M), user (U), actor (A) and tag (T) linked by 4 types: U-M, A-M, U-T and M-T, with attributes available for actors and movies. The classification task is movie genre prediction, which corresponds to a multi-label prediction scenario.

[1] https://github.com/aravindsankar28/Meta-GNN

TABLE IV. Semi-supervised node classification results (Micro-F_1 and Macro-F_1) on DBLP-A, DBLP-P and Movie. - indicates *not applicable*, x indicates *does not scale*.

Method	DBLP-A		DBLP-P		Movie	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**	**Micro-F1**	**Macro-F1**
DCNN	69.68	69.20	x	x	49.91	46.72
GCN	81.34	81.29	71.30	53.14	55.67	53.85
Graph-CNN	72.89	73.04	65.58	52.85	52.07	49.53
GAT	82.96	81.27	71.98	60.65	58.62	58.06
LP-Metapath	82.77	82.86	62.15	52.41	-	-
LP-Metagraph	83.03	83.11	62.97	57.08	-	-
HCC	61.62	59.75	61.56	58.12	53.38	26.25
CLN	80.99	80.94	66.71	52.19	57.11	47.71
metapath2vec	83.37	83.43	70.10	61.89	48.61	48.58
metapath2vec + Attributes	83.37	83.43	71.22	64.53	60.00	60.00
Meta-GNN	**89.12**	**87.38**	**74.58**	**66.76**	**64.72**	**62.34**

B. *Baselines:*

In our experiments, we compare against 10 algorithms in total, split across two categories. Since Meta-GNN generalizes homogeneous Graph CNNs to AHINs, we first compare against four recent graph CNN methods:

- **DCNN** [8]: Diffusion convolutions through weighted feature aggregation on depth-wise neighborhoods.
- **GCN** [9]: Graph convolutions as layer-wise linear aggregation of the first-order neighborhood.
- **Graph-CNN** [20]: Graph convolution filters as polynomials of functions of the graph adjacency matrix.
- **GAT [10]**: Graph Attention Network which utilizes attentional neighborhood aggregation functions.

All Graph CNNs are provided one-hot encodings of node types as additional attributes to make a fair comparison. Next, we compare against multiple strong AHIN baselines:

- **LP-Metapath** [28]: Metapath-specific Laplacians for joint label propagation and metapath weight learning.
- **LP-Metagraph** [24]: SSL algorithm on HINs based on an ensemble of metagraph guided random walks.
- **Column Network (CLN)** [26]: Deep neural network for node classification in multi-relational graphs.
- **HCC** [25]: Iterative collective classification method that exploits dependencies based on multiple metapaths.
- **metapath2vec** [23]: Skip-gram to learn node embeddings in HINs using metapath-based random walks, which feed into a logistic regression classifier.
- **metapath2vec + Attributes** [23]: We extend metapath2vec by concatenating the node embeddings with target node attributes before classification.

C. *Experimental setup:*

We sample 10% labeled nodes for training, 10% for validation and rest for testing. We repeat this process 10 times and report the mean performance in terms of both Micro-F_1 and Macro-F_1. We train a 3-layer Meta-GNN with two metagraph convolutional layers (hidden layer sizes of 64 and 32) with ReLU activations followed by a fully connected layer. We apply L_2 regularization with $\lambda = 10^{-5}$ for the weight matrices and use dropout of 0.5 in each layer. We train our model for a maximum of 200 epochs using Adam [36] with windowed early stopping (10 epochs) on the validation set. For node types that do not have attributes, we use 1-hot encoded inputs. For *Meta-GNN*, we use all *relevant* 3-node metagraphs that indicate semantic closeness based on the graph schema. We also ignore metagraphs with low occurrence frequency based on empirical thresholds. We provide details of these metagraphs online[2]. Similarly, we provide all relevant metapaths and metagraphs as input for the remaining AHIN baselines. Note that LP-Metapath and LP-Metagraph are not applicable for multi-label classification.

D. *Experimental results:*

From Table. IV, we observe relative gains of 7% and 4% (Micro-F_1) for *Meta-GNN* over other graph CNN models while gaining 6% and 4% overall in DBLP-A and DBLP-P respectively. Since venues are the most discriminative features in DBLP-A, even metagraph-based label-propagation methods do quite well. In contrast, the attributes of papers and their citations/references are more informative in DBLP-P, hence GCN, GAT and attributed-augmented metapath2vec perform better. In DBLP-P, DCNN does not scale due to $O(N^2)$ space complexity. In Movie, *Meta-GNN* performs the best with 10% gain in Micro-F_1 over graph CNN models, while improving 8% overall. Metapath2vec + Attributes is the strongest baseline, combining unsupervised representation learning on HINs with node attributes to learn a downstream classifier, while GAT is strongest graph CNN method as expected. *Meta-GNN* achieves significant performance improvements on all datasets, which illustrates the benefits of jointly modeling HIN semantics and attributes in an end-to-end framework, for achieving state-of-the-art performance.

E. *Ablation Study*

In this section, we conduct an ablation study to examine the relative importance of various key components (or aspects) involved in the design of *Meta-GNN*.

- **Node Attributes:** To analyze the benefits of using node attributes in AHINs, we instead use one-hot encoded vectors for each node (indicating their identity).

[2] https://sites.google.com/site/metagnn/

TABLE V. Ablation study analyzing impact of *node attributes*, *semantic roles* and *metagraph attention*

Method	**DBLP-A**		**DBLP-P**		**Movie**	
	Mic-F1	**Mac-F1**	**Mic-F1**	**Mac-F1**	**Mic-F1**	**Mac-F1**
(1) **Default**	**89.12**	**87.38**	**74.58**	**66.76**	**64.72**	**62.34**
(2) **No Attributes**	86.24	83.93	66.39	58.51	55.11	51.28
(3) **No semantic roles**	87.81	85.75	72.41	64.49	62.09	60.42
(4) **No Attention**	86.44	86.49	73.86	65.61	62.41	60.64

We observe noticeable performance drops on DBLP-P and Movie datasets demonstrating the importance of attributes towards achieving state-of-the-art performance. Note that this variant consistently outperforms attribute-agnostic baselines (LP-Metapath, LP-Metagraph and metapath2vec), which demonstrates the capability of *Meta-GNN* in capturing complex structural patterns despite the absence of node attributes.

- **Semantic Roles:** We examine the choice of defining explicit semantic roles for metagraph convolutions. We replace the metagraph-adjacency tensor of K_M roles with a single role-agnostic matrix merely indicating the existence of different nodes in a metagraph, while using the same set of metagraphs for consistency.

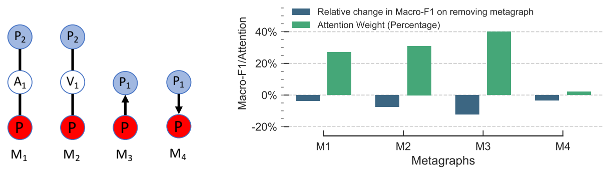
  We observe 2-3% reduction in performance across all datasets in the absence of semantic roles, thus validating the design choice of explicit role differentiation.
- **Metagraph-Attention:** To investigate the power of the metagraph-attention module in learning personalized metagraph preferences, we replace attention with a simple weighted combination of metagraphs.

  The removal of metagraph attention leads to a 2% performance drop on average, illustrating the benefits of learning personalized metagraph preferences.

F. Qualitative Analysis

In this section, we analyze the attention weights learned by *Meta-GNN*. Let us consider the research paper classification task in DBLP-P. To illustrate, we use 4 metagraphs (depicted in Fig. 3(a)) to represent different ways to relate a paper node with shared authors and venues (M_1 and M_2), references (M_3) and citations (M_4). We use a two-layer network for training, and report the average metagraph attention weights (percentages) assigned to different paper nodes. We remove each metagraph one at a time and re-train the model to observe the drop in performance. Fig. 3(b) depicts the attention weight of each metagraph along with the relative decrease in Macro-F_1 score on removing that particular metagraph.

Overall, the average attention for a metagraph is positively correlated with the decrease in performance on removing the metagraph, *i.e.*, our model assigns high attention weights for metagraphs that are crucial for good discrimination. From Fig. 3(b), we observe that metagraphs M_2 and M_3 have significantly larger weights, which is fairly intuitive since the references and venues of a paper are more strongly indicative of its research area/category in comparison to its authors or citations. Thus, the learned attention weights may be used to filter out low-weight metagraphs in practice.



(a) Sample metagraphs in DBLP-P dataset (b) Attention weights and relative performance drop on removing metagraph (in percent)

Fig. 3: **Qualitative analysis of metagraph attention weights. Relative changes in Macro-F_1 on metagraph removal are shown alongside attention weights. Higher weights are learnt for metagraphs whose removal leads to greater performance drop.**

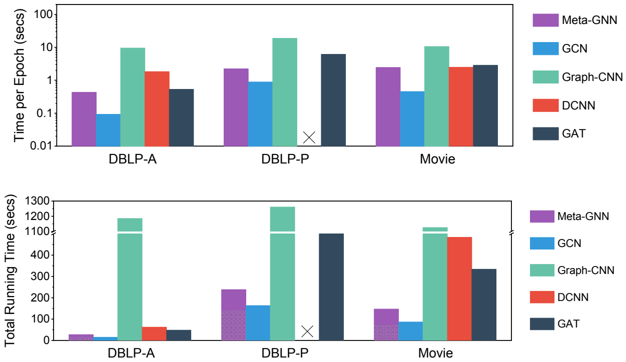


Fig. 4: **Efficiency comparison of graph CNNs across datasets. Shaded region denotes pre-computation time of *Meta-GNN*. Upper: running time per epoch. Lower: total running time.**

G. Computational Efficiency

We report running times on an Intel(R) Xeon(R) CPU E5-2699 v4 2.20 GHz system with 8 cores and 64 GB memory. We use CPU for comparison since many baseline implementations do not fit in GPU memory. We compare *Meta-GNN* with four neural methods, GCN, DCNN, Graph-CNN and CLN on all three datasets, as shown in Fig. 4.

Fig. 4(a) illustrates the training time per epoch of each model. We find that *Meta-GNN* is quite efficient in comparison to existing Graph CNNs and comes second only to GCN, which is expected since the complexity scales linearly with the number of metagraphs (Sec. IV-F). *Meta-GNN* is also faster than GAT, due to its application of metagraph-level attention instead of neighbor-level attention used in GAT.

We also compare the total running time of different models till convergence (Fig. 4(b)). We observe that *Meta-GNN* converges much faster than other graph CNN methods owing to its effective use of HIN semantics. Note that the running time of our model includes the cost of pre-computation. Although the pre-computation cost is a noticeable (but not substantial) portion of the total time, *Meta-GNN* is reasonably close to GCN and significantly faster than GAT as its rapid convergence trades off the cost of pre-computation.

VI. Conclusion and Future Work

In this paper, we study semi-supervised learning on Attributed Heterogeneous Information Networks. We introduce a novel metagraph convolution operation to model high-order locality and distinguish semantic roles of nodes in local heterogeneous neighborhoods. We propose a novel neural architecture *Meta-GNN* that employs multiple convolutional layers, each augmented with an attention module to learn personalized metagraph preferences for each node. Experimental results on multiple real-world datasets demonstrate significant gains over several state-of-the-art baselines.

We find several interesting and concrete directions for future work. A direct motif-based extension to operate on homogeneous networks, yields encouraging results [37]. Firstly, Our framework can be readily generalized beyond GCN to incorporate expressive neighborhood aggregation functions, such as pooling [21] and attentional [10] aggregations. Secondly, Neighborhood sampling [21] strategies can facilitate scaling of *Meta-GNN* to very large graphs. Finally, we also plan to examine the effect of larger metagraphs, especially in domains with more complex heterogeneous interactions.

VII. Acknowledgement

This material is based upon work supported by the National Science Foundation IIS 16-19302 and IIS 16-33755, Zhejing University ZJU Research 083650, Futurewei Technologies HF2017060011, Futurewei Technologies 094013, UIUC OVCR CCIL Planning Grant 434S34, and UIUC CSBS Small Grant 434C8U, and UIUC Advanced Digital Sciences Center Faculty Grant. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the funding agencies.

References

[1] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, 2004, pp. 321–328.

[2] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 817–826.

[3] A. Krishnan, A. Sharma, and H. Sundaram, "Improving latent user models in online social media," *arXiv preprint arXiv:1711.11124*, 2017.

[4] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao, "Graph regularized transductive classification on heterogeneous information networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2010, pp. 570–586.

[5] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: a neural approach for poi recommendation," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1245–1254.

[6] A. Krishnan, A. Sharma, A. Sankar, and H. Sundaram, "An adversarial approach to improve long-tail performance in neural collaborative filtering," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM '18. New York, NY, USA: ACM, 2018, pp. 1491–1494. [Online]. Available: http://doi.acm.org/10.1145/3269206.3269264

[7] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[8] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 1993–2001.

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

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[11] X. Li, Y. Wu, M. Ester, B. Kao, X. Wang, and Y. Zheng, "Semi-supervised clustering in attributed heterogeneous information networks," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1621–1629.

[12] J. Hu, R. Cheng, K. C.-C. Chang, A. Sankar, Y. Fang, and B. Y. Lam, "Discovering maximal motif cliques in large heterogeneous information networks," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 746–757.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[15] X. Zhou and M. Belkin, "Semi-supervised learning," in *Academic Press Library in Signal Processing*. Elsevier, 2014, vol. 1, pp. 1239–1269.

[16] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[17] 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*. ACM, 2016, pp. 855–864.

[18] A. Sankar, A. Krishnan, Z. He, and C. Yang, "Rase: Relationship aware social embedding," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019.

[19] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *CoRR*, vol. abs/1506.05163, 2015.

[20] F. P. Such, S. Sah, M. Dominguez, S. Pillai, C. Zhang, A. Michael, N. Cahill, and R. Ptucha, "Robust spatial filtering with graph convolutional neural networks," *arXiv preprint arXiv:1703.00792*, 2017.

[21] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 1025–1035.

[22] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "Dynamic graph representation learning via self-attention networks," *arXiv preprint arXiv:1812.09430*, 2018.

[23] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 135–144.

[24] H. Jiang, Y. Song, C. Wang, M. Zhang, and Y. Sun, "Semi-supervised learning over heterogeneous information networks by ensemble of meta-graph guided random walks." IJCAI, 2017.

[25] X. Kong, P. S. Yu, Y. Ding, and D. J. Wild, "Meta path-based collective classification in heterogeneous information networks," in *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 2012, pp. 1567–1571.

[26] T. Pham, T. Tran, D. Q. Phung, and S. Venkatesh, "Column networks for collective classification." in *AAAI*, 2017, pp. 2485–2491.

[27] Y. Fang, W. Lin, V. W. Zheng, M. Wu, K. C.-C. Chang, and X.-L. Li, "Semantic proximity search on graphs with metagraph-based learning," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 277–288.

[28] M. Wan, Y. Ouyang, L. Kaplan, and J. Han, "Graph regularized meta-path based transductive regression in heterogeneous information network," in *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 2015, pp. 918–926.

[29] T.-y. Fu, W.-C. Lee, and Z. Lei, "Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 2017, pp. 1797–1806.

[30] Y. Shi, H. Gui, Q. Zhu, L. Kaplan, and J. Han, "Aspem: Embedding learning by aspects in heterogeneous information networks," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 144–152.

[31] M. Latapy, "Main-memory triangle computations for very large (sparse (power-law)) graphs," *Theoretical Computer Science*, vol. 407, no. 1-3, pp. 458–473, 2008.

[32] L. Lai, L. Qin, X. Lin, and L. Chang, "Scalable subgraph enumeration in mapreduce," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 974–985, 2015.

[33] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[34] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning," *Information Retrieval Journal*, vol. 3, pp. 127–163, 2000.

[35] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2016.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[37] A. Sankar, X. Zhang, and K. C.-C. Chang, "Motif-based convolutional neural network on graphs," *arXiv preprint arXiv:1711.05697*, 2017.