HGBER: Heterogeneous Graph Neural Network With Bidirectional Encoding Representation

Yanbei Liu, Lianxi Fan, Xiao Wang, *Member, IEEE*, Zhitao Xiao, Shuai Ma, *Senior Member, IEEE*, Yanwei Pang, *Senior Member, IEEE*, and Jerry Chun-Wei Lin, *Senior Member, IEEE*

***Abstract*— Heterogeneous graphs with multiple types of nodes and link relationships are ubiquitous in many real-world applications. Heterogeneous graph neural networks (HGNNs) as an efficient technique have shown superior capacity of dealing with heterogeneous graphs. Existing HGNNs usually define multiple meta-paths in a heterogeneous graph to capture the composite relations and guide neighbor selection. However, these models only consider the simple relationships (i.e., concatenation or linear superposition) between different meta-paths, ignoring more general or complex relationships. In this article, we propose a novel unsupervised framework termed Heterogeneous Graph neural network with bidirectional encoding representation (HGBER) to learn comprehensive node representations. Specifically, the contrastive forward encoding is firstly performed to extract node representations on a set of meta-specific graphs corresponding to meta-paths. We then introduce the reversed encoding for the degradation process from the final node representations to each single meta-specific node representations. Moreover, to learn structure-preserving node representations, we further utilize a self-training module to discover the optimal node distribution through iterative optimization. Extensive experiments on five open public datasets show that the proposed HGBER model outperforms the state-of-the-art HGNNs baselines by 0.8%–8.4% in terms of accuracy on most datasets in various downstream tasks.**

***Index Terms*— Bidirectional encoding representation, heterogeneous graph neural network, heterogeneous graph representation learning, multiple meta-paths.**

I. Introduction

HETEROGENEOUS graphs, which are capable of modeling various types of nodes and diverse interactions between them, also known as heterogeneous information network, have become ubiquitous in real-world scenarios, ranging from bibliographic networks [1], social networks [2] to biological networks [3]. For example, as shown in Fig. 1, a bibliographic network (i.e., academic network) contains three types of nodes (author, paper, and venue) and two types of edges (author-write-paper and conference-publish-paper). Meanwhile, these basic relations can be further derived for more complex semantics over the heterogeneous graph (e.g., author-write-paper-conference-publish-paper). It has been well recognized that heterogeneous graphs are powerful models that are able to embrace rich semantics and structural information in real world data. Recently, heterogeneous graph neural networks (HGNNs) have received considerable research attention, because they are able to effectively combine the mechanism of message passing with complex heterogeneity, so that the complex structures and rich semantics can be well captured. So far, HGNNs have significantly promoted the development of heterogeneous network analysis toward real-world applications, e.g., recommend system [4], security system [5], and information retrieval [6].

Most of the existing HGNNs typically utilize multiple meta-paths to capture composite relations and guide neighbor selection. The meta-path [7] is an ordered sequence of node types and edge types defined on the network model that describes the composite relationship between the involved node types. As shown in Fig. 1, Author-Paper-Author (APA) and Author-Paper-Venue-Paper-Author (APVPA) are meta-paths describing two different relations among authors. The APA meta-path associates two coauthors, while the APVPA meta-path associates two authors who publish papers in the same venue. Therefore, we can view a meta-path as high-order proximity between nodes. The meta-path-based HGNNs can be roughly grouped into concatenation-based methods [8], [9] and attention-based methods [10], [11], [12]. For the convenience of description, we refer to node representations learned from a single meta-specific graph and the union of all meta-specific graphs as meta-level and

Manuscript received 26 January 2022; revised 24 July 2022 and 18 October 2022; accepted 23 December 2022. Date of publication 6 January 2023; date of current version 9 July 2024. This work was supported in part by the China Postdoctoral Science Foundation under Grant 2022M712370, in part by the Tianjin Natural Science Foundation Project under Grant H2021202008, in part by the Program for Innovative Research Team in University of Tianjin under Grant TD13-5034, and in part by the Open Project of Anhui Provincial Key Laboratory of Multimodal Cognitive Computation, Anhui University under Grant MMC202108. *(Corresponding author: Zhitao Xiao.)*

Yanbei Liu and Zhitao Xiao are with the School of Life Sciences, Tiangong University, Tianjin 300387, China, and also with the Tianjin Key Laboratory of Optoelectronic Detection Technology and Systems, Tianjin 300387, China (e-mail: liuyanbei@tiangong.edu.cn; xiaozhitao@tiangong.edu.cn).

Lianxi Fan is with the School of Electronics and Information Engineering, Tiangong University, Tianjin 300387, China (e-mail: 1831095395@tiangong.edu.cn).

Xiao Wang is with the School of Software, Beihang University, Beijing 100083, China (e-mail: xiaowang@bupt.edu.cn).

Shuai Ma is with the School of Computer Science and Engineering, Beihang University, Beijing 100083, China (e-mail: mashuai@buaa.edu.cn).

Yanwei Pang is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: pyw@tju.edu.cn).

Jerry Chun-Wei Lin is with the Department of Computer Science, Western Norway University of Applied Sciences, 5063 Bergen, Norway (e-mail: jerrylin@ieee.org).

Digital Object Identifier 10.1109/TNNLS.2022.3232709

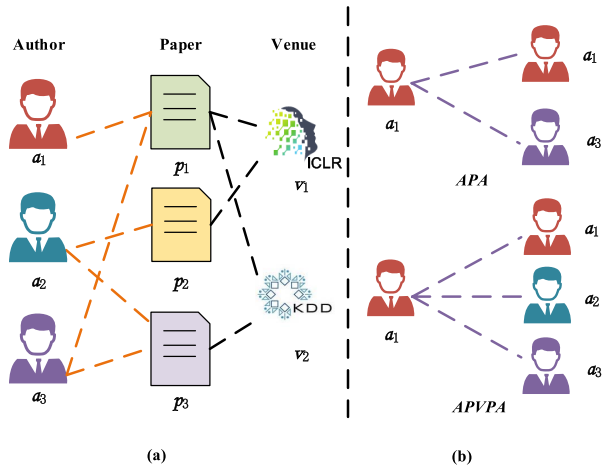

Fig. 1. Illustrative example of a heterogeneous citation network. (a) It consists of three types of nodes and two types of link relationships (heterogeneous citation network). (b) Author a_1 and its neighbors based on two meta-paths, Author-Paper-Author (APA) and Author-Paper-Venue-Paper-Author (APVPA) (meta-path-based node neighbors).

semantic-level node representations, respectively. Specifically, the concatenation-based methods first learn meta-level node representations for each meta-path and then capture the final node representations by directly concatenating semantic information of meta-paths. Such methods ignore the underlying association between different meta-paths. The attention-based methods mainly adopt the attention mechanism to assign different weights for each meta-level representation, and linearly superimpose to learn final node representations. Such methods can learn node representations with better flexibility, yet they are prone to preserve only the linear relationship between semantic-level and meta-level node representations [10].

Although these meta-path-based methods have achieved promising results, there are still some limitations. The first one is that it is difficult to explore the inherent association due to the complex and diverse relationships between semantic-level node representations and meta-level node representations. These existing models only consider the simple concatenation or linear superposition relationships between them. However, their relationships may be complex or nonlinear, i.e., the change of semantic-level node representations does not correspond with constant change in meta-level node representations, but with more complex change than in a linear relationship. The second one is that node representations obtained by concatenation-based and attention-based methods are not concise, but redundant. For example, as shown in Fig. 1, the scholar a_1 has the same neighbor a_3 via APA and APVPA meta-paths, so there is overlapping information between two meta-level node representations learned by these methods. The third limitation is that most meta-path based methods acquire the semantic-level node representations in the supervised setting, so they need additional labeled data to train the model.

To this end, we propose heterogeneous graph neural network with bidirectional encoding representation (HGBER), an unsupervised neural network model that addresses the above challenges to learn node representations. Different from traditional meta-path based heterogeneous graph representation learning methods, which linearly aggregate meta-level node representations to learn semantic-level node representations, the proposed method can learn the optimal semantic-level node representations by the bidirectional encoding networks, which introduces a neural network model to explore more general relationships between semantic-level node representations and meta-level node representations. Furthermore, we refine semantic-level node representations to discover the optimal node distribution by a self-training module. Finally, extensive experiments are conducted to show the superior performance of HGBER compared with the state-of-the-art baselines. The contributions of our work are summarized as follows.

1) We propose a general framework for heterogeneous graph representation learning by aggregating various meta-paths, which is capable of exploring nonlinear or more complex relationships among meta-level node representations in an unsupervised setting.
2) We propose HGBER, a novel heterogeneous graph neural network method that handles graph heterogeneity by utilizing bidirectional encoding representation, which can learn concise meta-level node representations and comprehensive semantic-level node representations for the heterogeneous graph. Moreover, the bidirectional encoding representation and self-training modules are jointly designed to acquire optimal node representations.
3) We conduct extensive experiments to evaluate the performance of HGBER in terms of the representation capacity and generalization ability on five public datasets. The results show its superiority by comparing with the state-of-the-art methods.

The remainder of this article is organized as follows. Section II presents the related work. Related concept, including graph representation learning and related content of heterogeneous graph, are briefly reviewed in Section III. Details of our proposed approach are presented in Section IV. In Section V, we present experimental results that demonstrate the effectiveness of our model on a variety of real-world datasets. Conclusions are drawn in Section VI.

II. Related Work

The related study includes graph neural networks and heterogeneous graph learning as follows.

A. Graph Neural Networks

Recently, with the success of deep learning, graph neural networks (GNNs) [13] have gained a lot of attention in graph representation learning. Different from previous graph embedding models, the main idea of GNNs is to aggregate the feature information from node's local neighbors via neural networks, which can combine the node attribute information and corresponding structural information to learn its new representations. Most successful GNNs are based on supervised learning including GCN [14], GAT [15], GraphRNN [16], AdaGCN [17], AP-GCN [18], DigGCN [19], and AS-GCN [20]. The unsupervised learning GNNs can

be mainly divided into two categories, i.e., random walk-based [21], [22], [23], [24], [25] and mutual information-based [26], [27].

B. Heterogeneous Graph Learning

To handle the heterogeneity of graphs, some methods have been proposed in recent years. Metapath2vec [7] formalizes the random walks based on meta-paths to obtain heterogeneous neighborhoods of a node and utilizes Skip-gram model to learn the network structure. HIN2Vec [28] learns the embedding vectors of nodes and meta-paths simultaneously while performing prediction tasks. From the perspective of attributed graphs, SHINE [29] utilizes multiple deep auto-encoders to extract users highly nonlinear representations while preserving the structure of original networks. HAN [10] leverages the attention mechanism in heterogeneous graph learning to learn the semantic information from multiple meta-paths defined connections. Based on meta-level and semantic-level structure, MAGNN [30] takes intermediate nodes of meta-paths into account. GTN [31] proposes to automatically identify useful connections. HGT [32] is designed for web-scale heterogeneous networks. In unsupervised setting, HetGNN [33] samples a fixed size of neighbors, and fuses their features by using LSTMs. Inspired by the mutual information-based learning model, HDGI [12] effectively learns node representations by maximizing the local–global mutual information. NSHE [34] focuses on network schema, and preserves pairwise and network schema proximity simultaneously. HeCo [35] employs cross-view (network schema view and meta-path view) contrastive mechanism to collaboratively learn high-level node embeddings. However, the above methods cannot fully exploit the general relationships among meta-paths to learn comprehensive node representations.

III. Preliminary

In this section, we first formalize the heterogeneous graph representation learning problem, and then summarize main notations involved in this article.

Definition 1 (Graph Representation Learning): Given a homogeneous graph $G = (V, E, \mathbf{X})$, where V is a node set, E is a link set, and $\mathbf{X}$ denotes a matrix of node attributes. Graph representation learning aims to learn low-dimensional node feature matrix $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_N\} \in \mathbb{R}^{N\times d}$ with $d \ll N$ that integrates the node attributes and structural information, where N is the number of nodes and d is the dimension of node features.

Definition 2 (Heterogeneous Graph): A heterogeneous graph is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, which consists of a node set $\mathcal{V}$, a link set $\mathcal{E}$ and a node attribute matrix. Respectively, each $v \in \mathcal{V}$ and $e \in \mathcal{E}$ are associated with their type mapping functions $\tau(v) : \mathcal{V} \to T_{\mathcal{V}}$ and $\phi(e) : \mathcal{E} \to T_{\mathcal{E}}$, where $T_{\mathcal{V}}$ and $T_{\mathcal{E}}$ represent the type sets of vertices and edges satisfying $|T_{\mathcal{V}}| + |T_{\mathcal{E}}| > 2$. It is a homogeneous graph with the same node and edge types when $|T_{\mathcal{V}}| = 1$ and $|T_{\mathcal{E}}| = 1$. As shown in Fig. 1, three node types of paper, author, and venue and two types of link relationships constitute a heterogeneous graph.

TABLE I
Main Notations Used in This Article

Notation	Meaning
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$	Heterogeneous graph
$O_1 \xrightarrow{P_1} O_2 \cdots \xrightarrow{P_K} O_{K+1}$	Meta-paths
$G = (V, E, \mathbf{X})$	Homogeneous graph
$\{G_1, G_2 \cdots G_K\}$	Meta-specific graphs
$\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_K\}$	Meta-level representation set
$\tilde{\mathcal{H}} = \left\{\tilde{\mathbf{H}}_1, \tilde{\mathbf{H}}_2, \cdots, \tilde{\mathbf{H}}_K\right\}$	Reversed reconstruction feature set
$\mathbf{H} \in \mathbb{R}^{N\times d}$	Semantic-level representation
$\mathbf{s}_k$	Representation vector in DGI

Definition 3 (Meta-Relation): In a heterogeneous graph, each edge is related to its meta-relation. For example, $< \tau(s), \phi(e), \tau(t) >$ represents a meta-relation for an edge $e = (s, t)$ linked from node s to node t. Moreover, the graph schema $T_{\mathcal{G}} = (T_{\mathcal{V}}, T_{\mathcal{E}})$ of heterogeneous graph $\mathcal{G}$ consists of all meta-relations, where the node types $T_{\mathcal{V}}$ denotes node set and edge types $T_{\mathcal{V}}$ is edge set.

Definition 4 (Meta-Path): Meta-path scheme is defined as a sequence of meta-relations over graph schema. More details, a meta-path denotes as $O_1 \xrightarrow{P_1} O_2, \ldots, \xrightarrow{P_K} O_{K+1}$, where $P = P_1 + P_2 + \cdots + P_K$ is a composite relation between objects $O_1 \to O_{K+1}$. For example, in ACM dataset,[1] the meta-path PAP represents *Paper* $\xrightarrow{\text{Author}}$ *Paper* relation. To facilitate analysis, we can decompose the original heterogeneous graph into multiple homogeneous meta-specific graphs denoted as $\{G_1, G_2, \ldots, G_K\}$ according to the meta-paths. For the node set $\mathcal{V}$ with K nodes, it has the corresponding representation in each meta-specific graph, and the composed representation set is denoted as $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_K\}$. Main notations used in this article are listed in Table I.

IV. Methodology

In this section, we present HGBER for learning comprehensive node representations. As shown in Fig. 2, it mainly consists of three parts. First of all, the original heterogeneous graph is divided into multiple meta-specific graphs according to the meta-paths. Secondly, bidirectional encoding representation learning over meta-specific graphs is employed for learning node representations, including two steps: contrastive forward encoding networks are firstly performed to extract meta-level node representations corresponding to the meta-paths. Then, multiple reversed encoding networks are introduced to encode these meta-level node representations into semantic-level node representations. Note that, due to the associated nonlinear networks, more general relationships among different meta-paths can be learned. Finally, the final learned node representations are refined by a self-training module for the next task.

A. Contrastive Forward Encoding Network

After decomposing the original heterogeneous graph into multiple homogeneous meta-specific graphs according to meta-paths, we utilize contrastive forward encoding networks

[1] http://dl.acm.org/

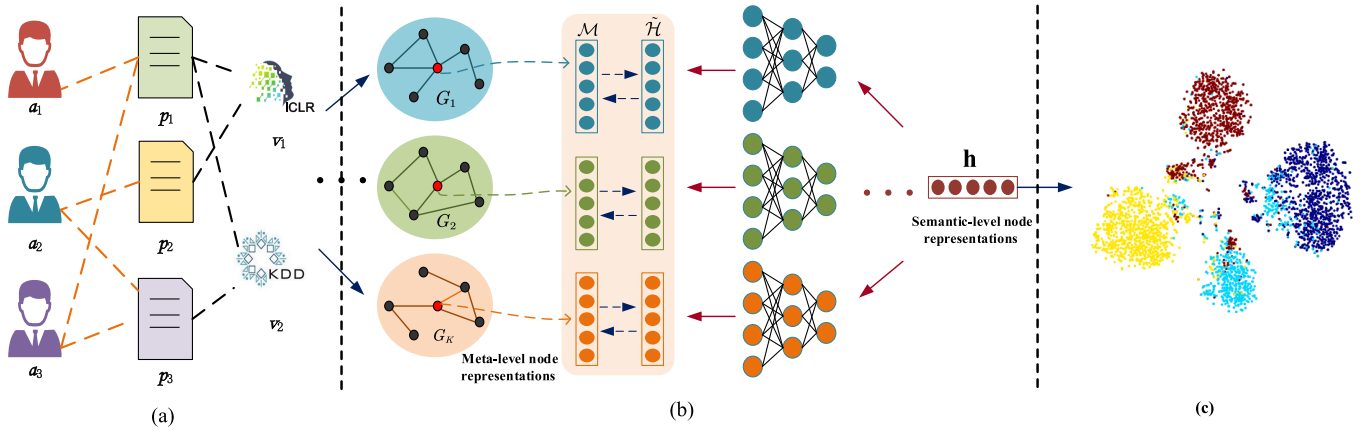

Fig. 2. Illustration of the proposed HGBER. The key component is the bidirectional encoding representation learning, which is composed of the contrastive forward encoding networks for meta-level node representations and the reversed encoding networks for semantic-level node representations. The proposed model jointly learns concise meta-level node representations for each meta-specific graph and comprehensive semantic-level node representations which can be mapped to reconstruct each meta-specific graph. (a) Heterogeneous network. (b) Bidirectional encoding representation learning. (c) Task.

to obtain meta-level node representations corresponding to each subgraph. Here two kinds of contrastive forward encoding networks are considered in this article. The first one is variational graph autoencoder (VGAE) [23], which uses GCN to encode the node attribute information and reconstruct the link information of the graph. The second one is the deep graph informax (DGI) [26] model by introducing the theory of maximizing mutual information. Being a self-supervised method, DGI can also learn the node representations without labels like VGAE. We introduce the details of the proposed framework as follows.

1) Variational Graph Autoencoder: Given an undirected unweight heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ with N nodes and K meta-paths, we denote K homogeneous meta-specific graphs as $\{G_1, G_2, \ldots, G_K\}$ and the adjacency matrix set as $\mathcal{A} = \{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k\}$, where $\mathbf{A}_k \in \mathbb{R}^{N\times N}$ corresponds to the kth meta-specific graph. It is worth mentioning that all meta-specific graphs share the same node attributes. Then, taking a meta-path as an example, the corresponding VGAE used in this article are parameterized as follows:

$$q(\mathbf{M}|\mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q(\mathbf{m}_i|\mathbf{X}, \mathbf{A}) \tag{1}$$

where $q(\mathbf{m}_i|\mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{m}_i|\mu_i, \mathrm{diag}(\sigma_i^2))$, $\mu = \mathrm{GCN}_\mu(\mathbf{X}, \mathbf{A})$ is the matrix of mean vectors μ_i same as $\log\sigma = \mathrm{GCN}_\sigma(\mathbf{X}, \mathbf{A})$ and $\mathbf{M}$ is the output of the VGAE model. The two-layer GCN model is defined as

$$\mathrm{GCN}(\mathbf{X}, \mathbf{A}) = \tilde{\mathbf{A}}\mathrm{ReLU}\big(\tilde{\mathbf{A}}\mathbf{X}\boldsymbol{\Theta}_0\big)\boldsymbol{\Theta}_1 \tag{2}$$

where $\boldsymbol{\Theta}$ is the parameter matrix of GCN. And $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is normalized Laplace matrix, where D is the diagonal matrix consisting of the row sums of $\mathbf{A}$ along its diagonals. $\mathrm{ReLU}(*) = \max(0, *)$ is the active function.

Given the attribute and adjacency matrices of a graph, VGAE can encode them into the matrix of mean vectors μ and variance vectors σ. The learned representation $\mathbf{m}$ can be sampled from the normal distribution determined by these two factors. An inner product is used to constrain two representation vectors, which can be seen as a decoding process

$$p(\mathbf{A}|\mathbf{M}) = \prod_{i=1}^{N}\prod_{j=1}^{N} p\big(A_{ij}|\mathbf{m}_i, \mathbf{m}_j\big) \tag{3}$$

where $p(A_{ij} = 1|\mathbf{m}_i, \mathbf{m}_j) = \mathrm{sigmoid}(\mathbf{m}_i^T\mathbf{m}_j)$, $\mathrm{sigmoid}(*)$ is the logistic sigmoid function. Therefore, we can minimize the following variational lower bound

$$\mathcal{L}_1 = \sum_{k=1}^{K} \mathbb{E}_{q(\mathbf{M}_k|\mathbf{X},\mathbf{A}_k)}\big[\log p\big(\mathbf{A}_k|\mathbf{M}_k\big)\big] - \mathrm{KL}\big[q\big(\mathbf{M}_k|\mathbf{X}, \mathbf{A}_k\big)||p(\mathbf{M}_k)\big] \tag{4}$$

where $p(\mathbf{M}) = \prod_i \mathcal{N}(\mathbf{m}_i|0, \mathbf{I})$ is a Gaussian prior. $\mathrm{KL}[q(*)||p(*)]$ is the Kullback-Leibler divergence between $q(*)$ and $p(*)$.

VGAE model is performed for the corresponding meta-specific graphs. Then we can obtain the meta-level representation set $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_K\}$, where $\mathbf{M}_k$ corresponds to the kth meta-specific graph.

2) Deep Graph Informax: The DGI model applies mutual information maximization theory to graph representation learning. It can be expressed as a mapping function $\mathcal{F}$. We can obtain the meta-level representation set $\mathcal{M} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_K\}$, where $\mathbf{M}_k = \mathcal{F}_k(\mathbf{X}, \mathbf{A}_k)$ corresponds to the kth meta-specific graph. $\mathbf{M}_k$ belongs to the space of $\mathbb{R}^{N\times d_k}$, where d_k is the feature dimension.

In addition, the DGI model relies on maximizing local mutual information. Taking a meta-path as an example, it constrains the learned meta-level node representations $\mathbf{M}$ to be able to capture the global information of the corresponding meta-specific graph. Firstly, taking $\varphi(\mathbf{X}, \mathbf{A})$ as the basic node representation encoding network, the preliminary node representations in the graph can be calculated. Note that $\varphi(\mathbf{X}, \mathbf{A})$ can be any suitable graph learning framework. In this article, we still adopt the GCN framework with two layers. The node representations are denoted $\mathbf{M} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N\}$. Then, the graph-level information is summarized as a vector $\mathbf{s}$ through a global information encoder $\mathcal{R}(\mathbf{M})$. In particular,

we perform a direct average function to aggregate the graph-level information, which is represented by

$$\mathbf{s} = \mathcal{R}(\mathbf{M}) = \varepsilon\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{m}_i\right) \tag{5}$$

where $\varepsilon(*)$ is the logistic sigmoid function.

To constrain the maximum local mutual information, the discriminator represented by $\mathcal{D}(*, \mathbf{s})$ is to determine positive and negative samples taking the graph-level information vector $\mathbf{s}$ as the baseline, where $*$ is the sample representations to be determined. $\mathcal{D}$ can be a binary classification network, and its output are probability scores. Specifically, the positive sample representations are defined as the original node representations $\mathbf{M}$, and the negative ones $\tilde{\mathbf{M}} = \{\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \ldots, \tilde{\mathbf{m}}_N\}$ are extracted from the reconstructed meta-specific graph represented by $\tilde{G}$ with M nodes through a shuffle function $\mathcal{C}(G)$. In our application, a direct function $\mathcal{C}$ is to shuffle the attribute matrix of the graph G. The objective function is defined as a binary cross-entropy loss as follows:

$$\mathcal{L}_1 = \frac{1}{K(N+M)}\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\mathbb{E}_{G_k}\left[\log\mathcal{D}(\mathbf{m}_{ki}, \mathbf{s}_k)\right] + \sum_{k=1}^{K}\sum_{j=1}^{M}\mathbb{E}_{\tilde{G}}\left[\log\left(1-\mathcal{D}\left(\tilde{\mathbf{m}}_{kj}, \mathbf{s}_k\right)\right)\right]\right). \tag{6}$$

By minimizing the above objective function, mutual information can be maximized. The learned node representations $\mathcal{M}$ are the meta-level node representations.

B. Reversed Encoding Network

After obtaining the meta-level node representation set $\mathcal{M}$ of a heterogeneous graph, our goal is to learn comprehensive node representations that combine all meta-paths. We denote the semantic-level node representations as $\mathbf{H} \in \mathbb{R}^{N\times d}$, where d is the dimension of node features. To this end, the reversed encoding networks realize the assumption that each meta-level node representation $\mathbf{M}$ can be degraded from the comprehensive common semantic-level node representations $\mathbf{H}$. The three-layer fully connected neural networks are employed to model the degradation process as shown in Fig. 2. Specifically, we map $\mathbf{H}$ onto the meta-level node representation $\mathbf{M}_k$ with reversed encoding network $\text{MLP}(\mathbf{H}; \mathbf{\Phi}_k) = \tilde{\mathbf{H}}_k$, where $\tilde{\mathbf{H}}_k \subseteq \tilde{\mathcal{H}}$ is the reversed reconstruction features and $\mathbf{\Phi}_k$ is the weight matrix. Then, combining multiple meta-paths, the objective of reversed encoding networks is defined as

$$\mathcal{L}_2 = \frac{1}{2}\sum_{k=1}^{K}\|\mathbf{M}_k - \tilde{\mathbf{H}}_k\|_F^2. \tag{7}$$

In our model, we jointly learn meta-level node representations for each meta-specific graph and semantic-level node representations using bidirectional encoding networks. Its objective function can be summarized as

$$\mathcal{L} = \mathcal{L}_1 + \lambda\mathcal{L}_2 \tag{8}$$

where λ is a hyperparameter to balance the forward encoding stage and reversed encoding stage. After optimizing the above objective function, we can obtain the semantic-level node representations, which can aggregate information from each meta-path and dock specific tasks.

C. Self-Training Module

The framework proposed above aggregates the information of heterogeneous graphs (including node attributes, link relationships and heterogeneous graph structure) into the semantic-level node representations $\mathbf{H}$. However, the node representations obtained in this way are not necessarily optimal, since they just follow the local structure of the original graph data. And it is not rigorous that believing the distribution of the original node attributes is optimal. We expect that the learned semantic-level node representations $\mathbf{H}$ have better structure and adaptively seek the optimal distribution, that is, the nodes within the same category are gathered densely, and the boundaries between different categories are distinct. Therefore, it is worth adding some items to constrain the distribution of nodes in the process of learning $\mathbf{H}$. Inspired by DCE [36], we introduce a self-training objective function to optimize the node distribution. It is essentially a clustering constraint that can be directly added to the objective function as an item, so that the network can directly output a list of categories, which is different from K-Means algorithm [37]. By utilizing highly confident nodes as soft labels to supervise the learning process, the self-training scheme constraint $\mathbf{H}$ is to minimize the following function

$$\mathcal{L}_3 = \text{KL}[T||R] = \sum_i\sum_j t_{ij}\log\frac{t_{ij}}{r_{ij}} \tag{9}$$

where $\text{KL}[T||R]$ is the Kullback–Leibler divergence between T and R. In addition, R is the distribution of the soft labels and r_{ij} indicates the similarity measured by Student's t-distribution between node embedding $\mathbf{h}_i$ and centroid μ_j

$$r_{ij} = \frac{\left(1+||\mathbf{h}_i-\mu_j||^2/\alpha\right)^{-\frac{\alpha+1}{2}}}{\sum_{j'}\left(1+||\mathbf{h}_i-\mu_{j'}||^2/\alpha\right)^{-\frac{\alpha+1}{2}}} \tag{10}$$

where $\mathbf{h}_i \in \mathbf{H}$ denotes the learned semantic-level node representations, r_{ij} can be interpreted as a soft clustering assignment, α are the degrees of freedom of the Student's t-distribution. Due to the unsupervised setting, we set $\alpha = 1$. It is worth noting that in the first calculation, K-Means clustering algorithm of $\mathbf{H}$ is required to initialize μ. In (9), t_{ij} is computed by the first raising r_{ij} to the second power to get a denser distribution

$$t_{ij} = \frac{r_{ij}^2/f_j}{\sum_{j'} r_{ij}^2/f_{j'}} \tag{11}$$

where $f_j = \sum_j r_{ij}$ is the soft cluster frequency. Further, the final cluster label of node i can be obtained by the following:

$$c_i = \arg\max_j r_{ij}. \tag{12}$$

The distribution of R will be denser by minimizing (9). Its training strategy is same as a form of self-training, and the distribution T can be considered a target. This learning

manner can iteratively learn from high-confidence predictions to improve the initial estimate and low confidence nodes.

Combining the contrastive forward encoding network, reversed encoding network, and self-training module, the final objective function can be summarized as

$$\mathcal{L} = \mathcal{L}_1 + \lambda \mathcal{L}_2 + \gamma \mathcal{L}_3 \tag{13}$$

where γ and λ are hyperparameters to balance three terms. Our model has the following three merits: firstly, the intrinsic information of each meta-specific graph is automatically extracted with the contrastive forward encoding networks. Secondly, the degradation process involved in the reversed encoding networks ensures the intrinsic information from each meta-specific graph are encoded into the comprehensive semantic-level node representations. Thirdly, it further performs a self-training module to discover the optimal node distribution. The above three merits of the proposed model provide conditions for learning comprehensive common node representations.

D. Model Optimization

The proposed method has the following variables to be optimized: the weights $\mathbf{\Theta}$ of the contrastive forward encoding networks, the weights $\mathbf{\Phi}$ of the reversed encoding networks, the cluster centers μ in self-training part and the semantic-level node representations $\mathbf{H}$. We firstly pre-train the contrastive forward encoding network to obtain the preliminary meta-level representation set $\mathcal{M}$. Then, we iteratively update parameters of the reversed encoding network to acquire the preliminary H. Further, the initial cluster centers μ are obtained by performing K-Means clustering algorithm on $\mathbf{H}$. A self-training module is employed to acquire the structure-preserving node representations. Finally, the above steps are updated in turn to obtain the optimal solution. More details are given as follows.

1) Update Contrastive Forward and Reversed Encoding Networks: To update the contrastive forward and reversed encoding networks, we just need minimize (8) using Adam algorithm [38]. It is worth noting that the reversed encoding network needs a better mapping target as initialization, so we pre-train the DGI to obtain preliminary meta-level node representations before updating the parameters of reversed encoding networks. Specifically, taking the DBLP dataset as an example, we pre-train both the GAE network and the DGI network for 120 iterations. Then after randomly initializing $\mathbf{H}$, joint optimization can be performed. So far, we could obtain the preliminary semantic-level node representations $\mathbf{H}$.

2) Update Cluster Center μ: The self-training part is a structural constraint on the semantic-level node representations $\mathbf{H}$. The distribution of $\mathbf{H}$ can be changed with the optimization process. To acquire a better initialized $\mathbf{H}$, we update the parameters of the entire network step by step. The above process of obtaining the preliminary $\mathbf{H}$ can be regarded as the initialization of the following step. In (9), the gradients of $\mathcal{L}_3$ with respect to cluster center μ are

$$\frac{\partial \mathcal{L}_3}{\partial \mu_j} = 2\sum_{i=1}^{N}\left(1 + ||h_i - \mu_i||^2\right)^{-1}\left(r_{ij} - t_{ij}\right)\left(h_i - \mu_j\right). \tag{14}$$

Algorithm 1 Model Training for HGBER

Require: Heterogeneous graph: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, meta-paths: $O_1 \xrightarrow{P_1} O_2 \cdots \xrightarrow{P_K} O_{K+1}$;
Ensure: Node representations $\mathbf{H}$ and cluster labels;
Initialize the weights $\mathbf{\Theta}$ of the contrastive forward encoding networks and $\mathbf{\Phi}$ of the reversed encoding networks, semantic-level node representations $\mathbf{H}$ and cluster centers μ;
Adopt K-Means algorithm to obtain initial μ;
while *not converge* **do**
  // **Training bidirectional encoding networks**
  Update weights of the forward encoding networks;
  Update weights of the reversed encoding networks;
  // **Self-training module for node representations**
  Update the cluster center μ;
  Calculate the target distribution T;
  Update the semantic-level node representations $\mathbf{H}$;
end while
return Node representations $\mathbf{H}$ and cluster labels.

Using the stochastic gradient descent (SGD) method [39], the cluster center μ can be updated by

$$\mu_j = \mu_j - \eta \frac{\partial \mathcal{L}_3}{\partial \mu_j} \tag{15}$$

where η is the learning rate.

The next step is to update the target distribution T. According to (10) and (11), the updated distribution T can be calculated. The distributions T and R are updated alternately by iterative optimization. In addition, we can also obtain the cluster to which node belongs according to Q. The cluster label corresponding to node i can be calculated by

$$\text{Label}_i = \arg\max_j r_{ij} \tag{16}$$

where r_{ij} is obtained by (10). So far, the overall network optimization process can be summarized as updating the parameters of DGI to obtain preliminary meta-features, updating the parameters of the reversed encoding networks to obtain preliminary semantic-level node representations $\mathbf{H}$ and updating the cluster center μ and target distribution T. To clarify, we summarize the optimization process in Algorithm 1.

V. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of the proposed HGBER. Specifically, we demonstrate its advantages over other methods in different tasks, including node classification, node clustering, and visualization of node representations. In addition, we also discuss the obtained experimental results, verify, and analyze the role of the various components of the proposed method.

A. Datasets

The following five public datasets are used to verify the effectiveness of the proposed method.

ACM[2]*:* ACM is a paper network contrived from the ACM dataset using papers published in ACM Knowledge Discovery and Data Mining (KDD), SIGMOD, SIGCOMM, MobiCOMM, and VLDB. ACM dataset consists of 3025 papers (*P*), 5835 authors (*A*), and 56 subjects (*S*). The papers are divided into three categories: *database, wireless communication, data mining*. Depending on the conferences linked to the papers, we classify the papers into the appropriate categories. The attribute information of papers is extracted from keywords. Two meta-paths PAP, PSP are extracted for use with experiments.

DBLP[3]*:* DBLP is an integrated database of English-language literature in the field of computer with the results of research as the author as the core. In this article, a subset of DBLP which consists of 14 328 papers (*P*), 4057 authors(*A*), 20 conferences(*C*), and 8789 terms(*T*) is extracted. We divide the authors into four categories according to their research areas: *database, data mining, machine learning, information retrieval*. Depending on the conferences linked to the authors, we classify the authors into the appropriate research areas. The attribute information of authors is extracted from keywords. Three meta-paths, APA, APCPA, and APTPA are extracted for use with experiments.

IMDB[4]*:* IMDB is a link dataset built with permission from the Internet Movie Data (IMDB). A subset of IMDB which contains 4780 movies (*M*), 5841 actors (*A*), and 2269 directors (*D*) is extracted. We divide the movies into three categories: *Action, Comedy, Drama* according to their genre. The attribute information of movies is extracted from plots. Two meta-paths *MAM*, *MDM* are extracted for use with experiments.

Yelp[5]*:* The Yelp dataset contains 2614 businesses (*B*), 1286 users (*U*), two reviews (*R*), two services (*S*), and nine rating levels (*L*). The business nodes are labeled by their category. The node features are constructed by the bag-of-words representation of the related keywords. Four meta-paths, BUB, BRB, BSB, and BLB, are extracted for use with experiments.

Aminer[6]*:* The Aminer dataset contains 6564 papers (*P*), 13 329 authors (*A*), 35 890 reference (*R*), which is a subset extracted from the original dataset. The paper nodes are divided into four classes. Two meta-paths *PAP*, *PRP* are extracted for use with experiments.

The details of each dataset are summarized in Table II.

B. Comparison Method

To verify the effectiveness of the proposed method, we compare HGBER with the following unsupervised and supervised methods.

Unsupervised Methods:

1) *DeepWalk [21]:* DeepWalk is a random walk based graph representation learning method. It is designed for the homogeneous graph. Here we test all the meta-paths for it and report the best performance.
2) *Metapath2vec [7]:* Metapath2vec is a heterogeneous graph embedding method based on random walk and skip-gram.
3) *HERec [40]:* HERec is a heterogeneous graph embedding method. It introduces a type-constraint strategy to filter the sequence of nodes, and learns node representations for each meta-path using skip-gram.
4) *FeaCon-V:* FeaCon-V method concatenates the features of each meta-path learned from VGAE network.
5) *FeaCon-D:* FeaCon-D method concatenates the features of each meta-path learned from DGI network.
6) *HDGI [12]:* HDGI is an unsupervised representation learning method for heterogeneous graphs, which uses the meta-path to modal the structure involving semantics and utilizes semantic-level attention mechanism to capture individual node local representations.
7) *HEAD [41]:* HEAD is an unsupervised representation learning method, which introduces the adversarial disentangler to separate the distinct, informative factors of variations in node semantics formulated by meta-paths.

Supervised Methods:

1) *GCN [14]:* GCN is a semi-supervised graph convolutional network designed for homogeneous graphs. Here we test all the meta-paths for it and report the best performance.
2) *GAT [15]:* GAT is a semi-supervised method for homogeneous graphs, which introduces an attention mechanism. Here we test all the meta-paths for it and report the best performance.
3) *HAN [10]:* HAN is a semi-supervised embedding method for heterogeneous graphs that introduces two levels of attention mechanism, and can be combined with meta-specific graphs constructed by multiple meta-paths for learning.

The Proposed Methods:

The proposed HGBERs contains two variants as follows:

1) *HGBER-V:* The first variant of proposed HGBER is to take the VGAE model as contrastive forward encoding network.
2) *HGBER-D:* The second variant of proposed HGBER is to take the DGI model as contrastive forward encoding network.

C. Experimental Setup

For the unsupervised methods, node representations are learned from the full dataset without label information. For the supervised methods such as GCN, GAT, and HAN, all datasets are divided into training set, validation set, and test set according to the same ratio as in [10]. The logistic regression (LR) classifier is employed as the basic classifier. A tenfold cross-validation strategy is applied to evaluate all comparison methods. Indicators Micro-F1 and Macro-FI are used as metrics.

For the node clustering, the K-Means algorithm is performed to verify the clustering performance of node representations. The same ground truth as in the node classification

[2]http://dl.acm.org/

[3]https://dblp.uni-trier.de

[4]http://komarix.org/ac/ds/

[5]https://www.yelp.com/dataset

[6]https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding

TABLE II
STATISTICS OF DATASETS

Dataset	Node-type	Nodes	Edge-type	Edges	Feature	Meta-path
ACM	Paper(P)	3025	Paper-Author	9744	1830	*PAP*
	Author(A)	5835	Paper-Subject	3025		*PSP*
	Subject(S)	56				
IMDB	Movie(M)	4780	Movie-Actor	14340	1232	*MAM*
	Actor(A)	5841	Movie-Director	4780		*MDM*
	Director(D)	2269				
DBLP	Author(A)	4057	Author-Paper	19645	334	*APA*
	Paper(P)	14328	Paper-Conference	14328		*APCPA*
	Conference(C)	20	Paper-Term	88420		*APTPA*
	Term(T)	8789				
YELP	Business(B)	2614	Business-User	30838	2614	*BUB*
	User(U)	1286	Business-Review	2614		*BRB*
	Review(R)	2	Business-Service	2614		*BSB*
	Service(S)	2	Business-Level	2614		*BLB*
	Level(L)	9				
Aminer	Paper(P)	6564	Paper-Author	18007	128	*PAP*
	Author(A)	13329	Paper-Reference	58831		*PRP*
	Reference(R)	35890				

TABLE III
NODE CLASSIFICATION RESULTS

Dataset	ACM		DBLP		IMDB		YELP		Aminer	
Metric	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DeepWalk	0.738	0.741	0.862	0.848	0.483	0.453	0.694	0.498	0.633	0.619
Metapath2vec	0.736	0.738	0.906	0.908	0.488	0.451	0.718	0.508	0.639	0.606
HERec	0.738	0.739	0.911	0.907	0.509	0.476	0.720	0.513	0.698	0.655
FeaCon-V	0.877	0.881	0.910	0.915	0.538	0.506	0.722	0.537	0.652	0.642
FeaCon-D	0.877	0.872	0.911	0.888	0.533	0.423	0.727	0.529	0.637	0.664
HDGI	0.884	0.884	0.912	0.904	0.598	0.501	0.721	0.527	0.649	0.596
HEAD	0.870	0.872	0.913	0.910	0.573	0.490	**0.746**	**0.550**	0.775	0.659
GCN	0.883	0.882	0.917	0.908	0.546	0.518	0.721	0.525	0.766	0.617
GAT	0.871	0.873	0.919	0.911	0.569	0.529	0.727	0.530	0.773	0.656
HAN	0.905	0.906	0.920	0.917	0.585	0.543	0.723	0.535	0.747	0.620
HGBER-V	0.923	0.915	0.927	0.918	**0.599**	**0.551**	0.731	0.534	0.830	0.687
HGBER-D	**0.934**	**0.937**	**0.929**	**0.925**	0.587	0.546	0.740	0.547	**0.846**	**0.726**

is adopted. Indicators such as clustering accuracy (ACC), normalized mutual information (NMI), and adjusted rand index (ARI) [42] are used as metrics, in which the accuracy used in our clustering experiments is defined as

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n} \Gamma(\mathbf{l}_i, \mathrm{map}(\mathbf{r}_i))}{n} \tag{17}$$

where $\mathbf{r}_i$ and $\mathbf{l}_i$ are predicted cluster label and ground-truth label of sample $\mathbf{x}_i$, respectively. $\Gamma(x, y) = 1$ if $x = y$, or otherwise $\Gamma(x, y) = 0$. map$(\cdot)$ is a permutation map function mapping the cluster label into the class labels. Then the best map can be obtained by Kuhn–Munkres algorithm [42].

In the visualization experiment, through mapping the representations to a two-dimensional space, we draw the position of each node in the two-dimensional coordinate system. Here we utilize t-SNE [43] algorithm to visualize the author representations on DBLP dataset and color the nodes based on their research areas. As baselines, the original node attribute features and the concatenated meta-level node representations are selected for visualization. As comparisons, the node representations learned by VGAE, DGI, Metapath2vec, and HDGI are visualized.

D. Results and Analysis

1) Node Classification: The results of the node classification are shown in Table III. The best results of each indicator are marked in bold. As an unsupervised representation learning method, it can be observed that the proposed HGBERs (i.e., HGBER-V and HGBER-D) achieve the best results on 4 out of 5 datasets. Specifically, as a traditional homogeneous graph learning method, DeepWalk learns the node representations over each meta-path, but it cannot combine semantic information from different meta-paths, so it does not performs well compared with heterogeneous graph representation methods, e.g. Metapath2vec and HERec. With the powerful fitting ability of neural network, graph neural network-based methods, FeaCon-V, FeaCon-D, HDGI, and HEAD, perform better than the above traditional models on most datasets. Due to the use of strong prior knowledge (i.e., label information), the supervised methods, such as GCN, GAT, and HAN, achieve better results. The proposed HGBERs exceed these methods on most datasets, which indicates its powerful capabilities of node representations. It is worth mentioning the proposed HGBERs do not outperform HEAD on YELP dataset, achieving the second best results. The possible reason is that classes in YELP are severely imbalanced (R:S:L:U:B = 2:2:9:1286:2614), so link relationships between nodes in meta-specific graphs based on BRB, BSB, and BLB meta-paths are too simple. The obtained meta-level node representations are not discriminative and cannot provide rich information to final semantic-level node representations, while HEAD can learn discriminative information from each meta-specific graph by utilizing the adversarial learning manner. In addition, compared with HAN, the proposed HGBERs with the bidirectional encoding representation learning, capture the rich semantics successfully and show their superiority. Through the above analysis, we can

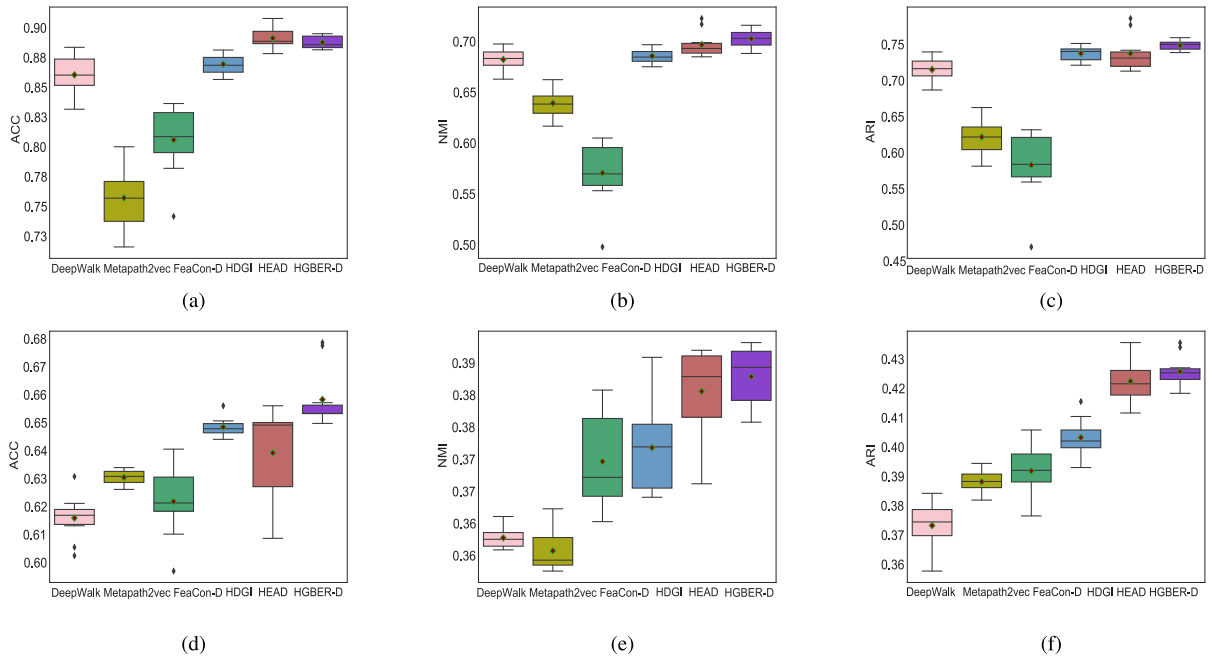

Fig. 3. Boxplots of clustering results of different methods on DBLP and YELP datasets. (a) DBLP. (b) DBLP. (c) DBLP. (d) YELP. (e) YELP. (f) YELP.

find that the proposed HGBERs could jointly learn concise meta-level node representations for each meta-specific graph and comprehensive common semantic-level node representations. For two variants, HGBER-V adopts the manner of generative learning and needs to reconstruct the original graph data. It tends to learn node-level features and is more suitable for link prediction task. Unlike HGBER-V, HGBER-D adopts the manner of contrastive learning and only needs to learn semantic-level features to classify nodes. It is more suitable for the tasks of node classification and graph classification.

2) Node Clustering: The results of the node clustering experiment are shown in Table IV. The best results of each indicator are marked in bold. It can be seen that the proposed HGBERs obtain the best performance on all datasets. Especially on ACM, they have obvious advantages compared with other methods. Specifically, without the aggregation of meta-level node representations, DeepWalk method does not achieve good results. It only performs well on DBLP, which may be because the selected meta-path contains richer discriminative node information. Two heterogeneous graph methods Metapath2vec and HERec both have better performance in terms of clustering NMI and ARI. Based on the graph neural network, FeaCon-V, FeaCon-D, HDGI, and HEAD achieve better results, which indicates that they could integrate node attribute and link information of the heterogeneous graph. Combining the advantages of the contrastive forward encoding and reversed encoding networks, the proposed HGBERs gain superior performance compared with other methods. To further visualize the superiority of our model, we plot clustering ACC on two datasets as boxplot shown in Fig. 3. It can be seen that HGBER is superior to other methods and is more stable.

3) Visualization: To show the pros and cons of node representations more intuitively, we conduct visualization experiment as shown in Fig. 4. After mapping the representations to a two-dimensional space, we draw the position of each node in the two-dimensional coordinate system. t-SNE [43] algorithm is adopted to visualize node representations on DBLP dataset. Each point in the figure indicates an author and its color represents the research area. As baselines, the original node attribute features and the concatenated meta-level node representations learned by VGAE and DGI are selected for visualization. The node representations learned by Metapath2vec, HDGI, HGBER*sel*, and HGBERs are also visualized. Specifically, the Rawdata method shows the distribution of the original node attributes. Its visual result is messy since the information used is not comprehensive. Although VGAE and DGI show better performance for four categories, but they still have messy results. For two heterogeneous methods Metapath2vec and HDGI, their results are much better, indicating that they can fuse the graph structural information with the node attributes, but there is no clear cluster boundary. From Fig. 4(f) and (g), we can see that the distribution of node representations presents an excellent effect that different classes are distributed more clearly and compactly. Due to the use of the self-training module, the nodes in the same cluster are more closely compared with HGBER*sel* (i.e., the HGBER-D without the self-training module), which verifies the effectiveness of node distribution optimization.

4) Ablation Experiment: Taking the performance of HGBER-D on the DBLP dataset as an example, we show the clustering ACC and NMI of the ablation experiment in Fig. 5 to verify the effectiveness of each part of the proposed framework. The methods to participate in the comparison are Rawdata, HGBERm1, HGBERm2, HGBERm3, HGBER*sel*, and HGBER. HGBER*sel* is the HGBER-D without considering self-training module, which serves as a baseline to verify the effectiveness of the self-training module. HGBERm1, HGBERm2 and HGBERm3 correspond to three meta-paths (i.e., APA, APCPA, and APTPA) based methods without reversed encoding network. From Fig. 5, we can see that

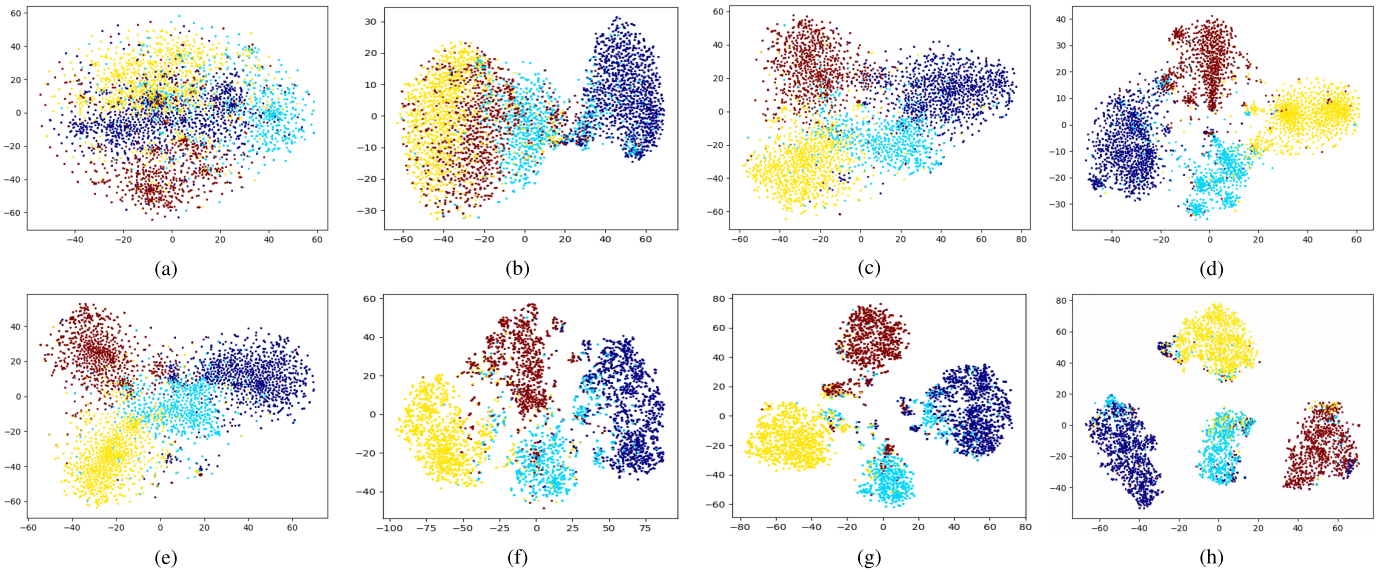

Fig. 4. Visualization on DBLP. (a) Rawdata. (b) VGAE. (c) DGI. (d) Metapath2vec. (e) HDGI. (f) HGBER*sel*. (g) HGBER-V. (h) HGBER-D.

TABLE IV
NODE CLUSTERING RESULTS

Dataset	ACM			DBLP			IMDB			YELP			Aminer		
Metric	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
DeepWalk	0.588	0.295	0.288	0.823	0.709	0.756	0.382	0.009	0.006	0.618	0.358	0.386	0.501	0.224	0.212
Metapath2vec	0.589	0.311	0.310	0.756	0.713	0.785	0.387	0.012	0.017	0.629	0.355	0.387	0.506	0.308	0.256
HERec	0.633	0.421	0.367	0.765	0.714	0.795	0.389	0.012	0.016	0.638	0.351	0.401	0.493	0.278	0.202
FeaCon-V	0.682	0.452	0.404	0.723	0.571	0.556	0.428	0.044	0.039	0.632	0.366	0.402	0.505	0.301	0.221
FeaCon-D	0.786	0.536	0.521	0.804	0.568	0.565	0.436	0.035	0.039	0.633	0.370	0.399	0.497	0.245	0.255
HDGI	0.718	0.544	0.496	0.881	0.715	0.766	0.443	0.037	0.041	0.638	0.375	0.401	0.506	0.281	0.237
HEAD	0.809	0.567	0.435	0.889	0.716	0.734	0.441	0.034	0.042	0.639	0.380	0.422	0.515	0.316	0.271
HGBER-V	0.891	0.658	0.705	**0.899**	**0.721**	**0.796**	0.447	0.047	0.046	0.653	**0.387**	**0.425**	0.503	0.333	0.285
HGBER-D	**0.893**	**0.659**	**0.707**	0.884	0.718	0.785	**0.451**	**0.048**	**0.046**	**0.658**	0.375	0.421	**0.524**	**0.338**	**0.296**

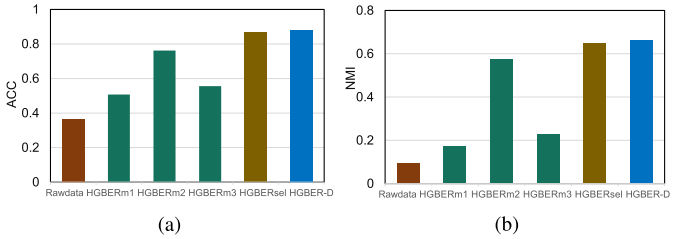

Fig. 5. Ablation experimental results on DBLP. (a) Clustering ACC. (b) Clustering NMI.

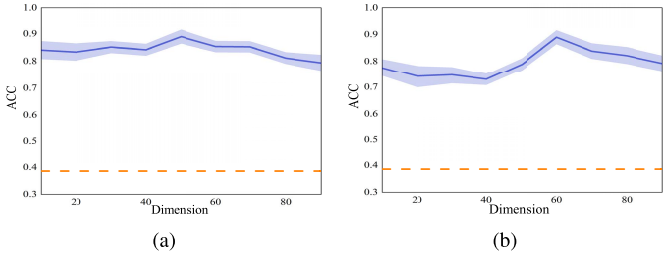

Fig. 6. Clustering accuracy on different dimensions of the node representations on DBLP. (a) Meta-level node representations. (b) Semantic-level node representations.

HGBER-D achieves the best performance, indicating that the proposed method can effectively aggregate the information of various types of link relations and attributes of the node itself, and optimize the distribution of node representations on the basis of HGBER*sel*. Moreover, the average training time of HGBER-D and DGI is 8.25 min versus 40.36 s under a linux machine powered by an Intel(R) Core(TM) i9-9900k CPU at 3.60 GHz CPU and a GeForce RTX 2080 TI GPU cards. Because we utilize three meta-paths on DBLP, compared with DGI, the proposed HGBER-D consists of three forward encoding network modules, three reversed encoding network modules, and a self-training module in the experiments.

5) Dimension Analysis: To determine dimensions of the meta-level and the semantic-level node representations, taking HGBER-D as an example, we study the sensitivity of the dimensional parameter and report clustering ACC on DBLP dataset. As shown in Fig. 6, the 50-dimensional meta-level node representations and the 60-dimensional semantic-level node representations could achieve the best experimental results. Note that the red dotted line represents the clustering results by using the raw features of nodes.

6) Hyperparameter Sensitivity: We investigate the sensitivity of two main parameters in our algorithm, i.e., λ and γ in (13). The parameter tuning on DBLP for our HGBER-D is plotted as shown in Fig. 7. We tune the two parameters from $\{0, 0.1, 0.2, 0.3, \ldots, 0.9, 1\}$ and report the clustering accuracy. In fact, for learning representation $\mathbf{H}$, we tune λ from greater than zero. It can be observed that with the increase of two parameters, the clustering accuracy is also improved, which indicates that the effectiveness of bidirectional encoding representation learning and self-training module. When both λ and γ are set to 0.8, the accuracy reaches the best.

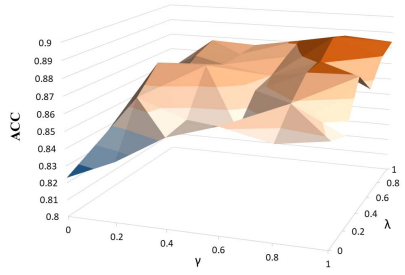

Fig. 7. Parameter sensitivity on DBLP.

VI. Conclusion

In this article, we investigated a novel and challenging problem of heterogeneous information fusion for heterogeneous graph representation learning. Different from the existing HGNNs methods that only consider the simple concatenation or linear superposition relationships between different meta-paths, we proposed HGBER, a general framework for heterogeneous graph representation learning by aggregating various meta-paths, which aims to learn comprehensive node representations by using bidirectional encoding representation. Specifically, HGBER first employs contrastive forward encoding to extract node representations on a set of meta-specific graphs corresponding to meta-paths, and then performs reversed encoding to learn comprehensive node representations. Moreover, a self-training module is designed to discover the optimal node distribution, so as to further improve the performance of HGBER. The experimental results on five public datasets demonstrate the effectiveness of the proposed method on node classification and node clustering tasks.

References

[1] W. Hu et al., "Strategies for pre-training graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–22.

[2] T. Zhang, Q. Wu, and J. Yan, "Learning high-order graph convolutional networks via adaptive layerwise aggregation combination," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 26, 2021, doi: 10.1109/TNNLS.2021.3119958.

[3] A. P. Davis et al., "The comparative toxicogenomics database: Update 2017," *Nucleic Acids Res.*, vol. 45, no. D1, pp. D972–D978, Jan. 2017.

[4] S. Fan et al., "Metapath-guided heterogeneous graph neural network for intent recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2478–2486.

[5] Y. Fan, S. Hou, Y. Zhang, Y. Ye, and M. Abdulhayoglu, "Gotcha—Sly malware! *Scorpion* a metagraph2vec based malware detection system," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 253–262.

[6] F. Li, Y. Li, C. Shang, and Q. Shen, "Fuzzy knowledge-based prediction through weighted rule interpolation," *IEEE Trans. Cybern.*, vol. 50, no. 10, pp. 4508–4517, Oct. 2020.

[7] Y. Dong, N. V. Chawla, and A. Swami, "Metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 135–144.

[8] J. Tang, M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2015, pp. 1165–1174.

[9] L. Xu, X. Wei, J. Cao, and P. S. Yu, "Embedding of embedding (EOE): Joint embedding for coupled heterogeneous networks," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, Feb. 2017, pp. 741–749.

[10] X. Wang et al., "Heterogeneous graph attention network," in *Proc. World Wide Web Conf.*, May 2019, pp. 2022–2032.

[11] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, "Leveraging meta-path based context for Top-n recommendation with a neural co-attention model," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1531–1540.

[12] Y. Ren, B. Liu, C. Huang, P. Dai, L. Bo, and J. Zhang, "Heterogeneous deep graph infomax," 2019, *arXiv:1911.08538*.

[13] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016, *arXiv:1606.09375*.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

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

[16] J. You, R. Ying, X. Ren, W. Hamilton, and J. Leskovec, "GraphRNN: Generating realistic graphs with deep auto-regressive models," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5708–5717.

[17] K. Sun, Z. Zhu, and Z. Lin, "AdaGCN: Adaboosting graph convolutional networks into deep models," 2019, *arXiv:1908.05081*.

[18] I. Spinelli, S. Scardapane, and A. Uncini, "Adaptive propagation graph convolutional network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 6, no. 4, pp. 930–944, Sep. 2020.

[19] M. Lei, P. Quan, R. Ma, Y. Shi, and L. Niu, "DigGCN: Learning compact graph convolutional networks via diffusion aggregation," *IEEE Trans. Cybern.*, vol. 52, no. 2, pp. 912–924, Feb. 2022.

[20] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive sampling towards fast graph representation learning," 2018, *arXiv:1809.05343*.

[21] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.

[22] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[23] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016, *arXiv:1611.07308*.

[24] A. García-Durán and M. Niepert, "Learning graph representations with embedding propagation," 2017, *arXiv:1710.03059*.

[25] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, *arXiv:1706.02216*.

[26] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–46.

[27] M. Shi, Y. Tang, and X. Zhu, "Topology and content co-alignment graph convolutional learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 1–9, Dec. 2021.

[28] T.-Y. Fu, W.-C. Lee, and Z. Lei, "HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1797–1806.

[29] H. Wang, F. Zhang, M. Hou, X. Xie, M. Guo, and Q. Liu, "SHINE: Signed heterogeneous information network embedding for sentiment link prediction," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 592–600.

[30] X. Fu, J. Zhang, Z. Meng, and I. King, "MAGNN: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proc. Web Conf.*, Apr. 2020, pp. 2331–2341.

[31] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, "Graph transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 11960–11970.

[32] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proc. Web Conf.*, Apr. 2020, pp. 2704–2710.

[33] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, "Heterogeneous graph neural network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 793–803.

[34] J. Zhao, X. Wang, C. Shi, Z. Liu, and Y. Ye, "Network schema preserving heterogeneous information network embedding," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1–7.

[35] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 1726–1736.

[36] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[37] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-means clustering algorithm," *J. Roy. Stat. Soc. C Appl. Statist.*, vol. 28, no. 1, pp. 100–108, Jan. 1979.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[39] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[40] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.
[41] R. Wang, C. Shi, T. Zhao, X. Wang, and Y. F. Ye, "Heterogeneous information network embedding with adversarial disentangler," *IEEE Trans. Knowl. Data Eng.*, early access, Jul. 13, 2021, doi: 10.1109/TKDE.2021.3096231.
[42] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 28, no. 1, 2014, pp. 1–7.
[43] L. van der Maaten, "Accelerating t-SNE using tree-based algorithms," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3221–3245, Oct. 2014.

Yanbei Liu received the Ph.D. degree from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2017.

He is currently an Associate Professor with Tiangong University, Tianjin. He has published more than 30 peer reviewed papers in top venues (e.g., IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON MEDICAL IMAGING (TMI), AAAI Conference on Artificial Intelligence (AAAI), and International Joint Conference on Artificial Intelligence (IJCAI)). His current research interests include graph representation learning, data mining, and machine learning.

Lianxi Fan received the B.S. and M.S. degrees from the School of Electrical and Electronic Engineering, Tiangong University, Tianjin, China, in 2018 and 2021, respectively.

His research interests include graph neural network, data mining, and machine learning.

Xiao Wang (Member, IEEE) received the Ph.D. degree from Tianjin University, Tianjin, China, in 2016.

He was a Post-Doctoral Researcher with Tsinghua University, Beijing, China, and an Associate Professor with the Beijing University of Posts and Telecommunications, Beijing. He is currently an Associate Professor with the School of Software, Beihang University, Beijing. He has published 80 peer reviewed papers in top venues [e.g., ACM Knowledge Discovery and Data Mining (KDD), International World Wide Web Conferences (WWW), NeurIPS, AAAI Conference on Artificial Intelligence (AAAI), International Joint Conference on Artificial Intelligence (IJCAI) and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE)]. His current research interests include data mining, social network analysis, and machine learning.

Dr. Wang obtains the WWW 2021 Best Paper Awards Nomination, the ICDM 2021 Best Student Paper Awards Runner-up, 2022 the World Artificial Intelligence Conference Youth Outstanding Paper Nomination Award, and the ACM China Rising Star Nomination Awards.

Zhitao Xiao received the B.E. and M.E. degrees from the Hebei University of Technology, Tianjin, China, in 1994 and 1997, respectively, and the Ph.D. degree from Tianjin University, Tianjin, in 2003.

He is currently a Professor with Tiangong University, Tianjin. His current research interests include pattern recognition, medical image processing, and intelligent signal processing.

Shuai Ma (Senior Member, IEEE) received the Ph.D. degrees from Peking University, Beijing, China, in 2004, and the University of Edinburgh, Edinburgh, U.K., in 2010.

He was a Post-Doctoral Research Fellow with the Database Group, University of Edinburgh, a Summer Intern with Bell Laboratories, Murray Hill, NJ, USA, and a Visiting Researcher with MSRA. He is currently a Professor with the School of Computer Science and Engineering, Beihang University, Beijing. His current research interests include database theory and systems, and big data.

Dr. Ma was a recipient of the Best Paper Award for VLDB 2010 and the Best Challenge Paper Award for WISE 2013. He has been an Associate Editor of the *VLDB Journal* since 2017.

Yanwei Pang (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the University of Science and Technology of China, Hefei, China, in 2004.

He is currently a Professor with Tianjin University, Tianjin, China. His current research interests include object detection, image recognition, image processing, deep learning, and their applications in self-driving cars, visual surveillance, human–machine interaction, and biometrics. He has published more than 120 scientific articles, including 32 IEEE TRANSACTIONS papers, in these fields.

Jerry Chun-Wei Lin (Senior Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, in 2010.

He is currently a Full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 500+ research articles in refereed journals and international conferences.

Dr. Lin is the fellow of IET (FIET) and ACM Distinguished Member (Scientist).