Corporate Relative Valuation Using Heterogeneous Multi-Modal Graph Neural Network

Yang Yang, Jia-Qi Yang, Ran Bao, De-Chuan Zhan, Hengshu Zhu, *Senior Member, IEEE*, Xiao-Ru Gao, Hui Xiong, *Fellow, IEEE*, and Jian Yang, *Member, IEEE*

Abstract—Corporate relative valuation (CRV) refers to the process of comparing a company's value from company products, core staff and other related information, so that we can assess the company's market value, which is critical for venture capital firms. Traditional relative valuation methods heavily rely on tedious and expensive human efforts, especially for non-publicly listed companies. However, the availability of information about company's invisible assets, such as patents, talent, and investors, enables a new paradigm to learn and evaluate corporate relative values automatically. Indeed, in this paper, we reveal that, the companies and their core members can natually be formed as a heterogeneous graph and the attributes of different nodes include semantically-rich multi-modal data, thereby we are able to extract a latent embedding for each company. The network embeddings can reflect domain experts' behavior and are effective for corporate relative valuation. Along this line, we develop a heterogeneous multi-modal graph neural network method, named HM^2, which deals with embedding challenges involving modal attribute encoding, multi-modal aggregation, and valuation prediction modules. Specifically, HM^2 first performs the representation learning for heterogeneous neighbors of the input company by taking relationships among nodes into consideration, which aggregates node attributes via linkage-aware multi-head attention mechanism, rather than multi-instance based methods. Then, HM^2 adopts the self-attention network to aggregate different modal embeddings for final prediction, and employs dynamic triplet loss with embeddings of competitors as the constraint. As a result, HM^2 can explore companies' intrinsic properties to improve the CRV performance. Extensive experiments on real-world data demonstrate the effectiveness of the proposed HM^2.

Index Terms—Corporate relative valuation, heterogeneous graph, multi-modal learning, linkage-aware

- *Yang Yang is with PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 211189, China. E-mail: yyang@njust.edu.cn.*
- *Jia-Qi Yang, Ran Bao, and De-Chuan Zhan are with Nanjing University, Nanjing 210023, China. E-mail: yangjq@lamda.nju.edu.cn, baorana@163.com, zhandc@nju.edu.cn.*
- *Hengshu Zhu is with Baidu Talent Intelligence Center, Baidu Inc, Beijing 100000, China. E-mail: zhuhengshu@baidu.com.*
- *Xiao-Ru Gao and Hui Xiong are with Management Science and Information Systems Department, Rutgers Business School, Rutgers University, Newark, NJ 07102 USA. E-mail: xg89@business.rutgers.edu, hxiong@rutgers.edu.*
- *Jian Yang is with the PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, and Jiangsu Key Lab of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China. E-mail: csjyang@njust.edu.cn.*

Manuscript received 18 June 2020; revised 30 April 2021; accepted 10 May 2021. Date of publication 14 May 2021; date of current version 7 December 2022.

This work was supported in part by NSFC under Grants 62006118, 91746301, 61773198, and 61632004, in part by Baidu TIC Open Fund, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20200460, in part by the Key Laboratory of Computer Network and Information Integration (Southeast University) Open Fund under Grant K93-9-2020-01, in part by Jiangsu Shuangchuang (Mass Innovation and Entrepreneurship) Talent Program, CAAIHuawei MindSpore Open Fund under Grant CAAIXSJLJJ-2021-014B, and in part by Young Elite Scientists Sponsorship Program by CAST.

(Corresponding author: De-Chuan Zhan.)

Recommended for acceptance by Y. Xia.
Digital Object Identifier no. 10.1109/TKDE.2021.3080293

1 Introduction

Recent years, we have witnessed the increasing popularity of applying machine learning models in software as a service (SAAS) and various enterprise applications, which greatly reduces the manual cost and improves the operating efficiency. For example, [1] proposed an intelligent job interview system, which can be applied in human resources management (HRM); [2] utilized the structure-aware convolution neural network for talent flow forecast, which can be introduced into enterprise resource planning (ERP); [3] applied neural networks for user recommendation, which can be practiced into customer relationship management (CRM), etc. Meanwhile, there also spring up many enterprise service companies based on artificial intelligence technologies, for example, UiPath[1] delivers data mining techniques for document management, contact center, human resources, supply chains, etc.; Pymetrics[2] combines artificial intelligence technology for intelligent recruitment, talent matching, etc. On the other hand, corporate valuation plays an important role in SAAS, which is to evaluate the relative value of companies, and establishes a critical basis for various pricing transactions in enterprise applications.

There exist several sophisticated corporate absolute valuation methods, for example, discounted cash flow method

1. http://www.uipath.com
2. https://www.pymetrics.ai/

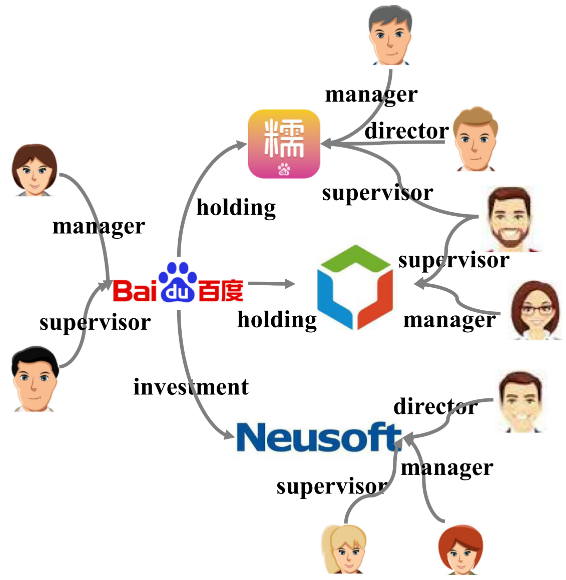


Fig. 1. (Best viewed in color.) Example of company structure. CRV usually considers two aspects of the company's structure: 1) affiliates, i.e., the relevant information of the company and its affiliates (for example, founded, acquired, and invested subsidiaries); 2) members, i.e., the relevant information of company's core member (for example, manager, supervisor, etc). Considering privacy, we use cartoon characters for replacement. Note that the companies and members can be regarded as entities, and the connections among them can be regarded as linkages, thereby all the data can naturally be constructed to a heterogeneous graph.

(DCF) [4], economic value added method (EVA) [5], real options method (ROA) [6] and price-to-sales method (PS) [7]. While these methods always require historical financial statements of the company, which are difficult to acquire, especially for non-publicly listed companies. On the other hand, some other corporate relative valuation methods are adopted. These methods usually rely on professionals to comprehensively consider the core resources, members, and competitors of the company, and then carry on the final valuation. Note that this type of methods can be used to estimate the company's value level without detailed financial statement analysis, whereas needs precise judgments and heavily relies on tedious and expensive manpower. With the economic development, the number of companies has increased dramatically, thereby it is undoubtedly difficult for venture capital firms to conduct large amount of company valuation screening on interested companies. In result, it urgently needs automatic or semi-automatic CRV technology by applying machine learning models.

As a matter of fact, relative valuation performed manually without financial statements always considers three factors: 1) the core resources of the company and its affiliates, such as the basic information, business conditions, and intellectual properties; 2) the information of core members of the company, such as member's background, resume, and influence; and 3) the valuation of competitors within the same industry. Naturally, as shown in Fig. 1, these companies and members construct a complex heterogeneous multi-modal graph. In detail, there are two types of nodes in the graph, i.e., companies and members. Meanwhile, attributes of nodes constitute multi-modal data, i.e., different types of nodes have various descriptions. Besides, there appear multiple types of graph linkages, i.e., company-company, company-member, and member-member. Therefore, by comprehensively modeling the corporate/personal attributes and the linkages among them, we can obtain new latent embeddings to describe the company, which can be further utilized in corporate valuation task. This learning procedure is also confirmed with professional domain experts' operation in reality.

Inspired by the observations above, we develop HM^2, a heterogeneous graph neural network for corporate relative valuation. HM^2 is a deep graph network, which can acquire discriminative embeddings of the company node by encoding heterogeneous neighbors comprehensively. Different from previous HGNNs, HM^2 can effectively capture the relationships among nodes, and design specific structural loss function to improve the final performance. In detail, based on obtained heterogeneous neighbors of the input company, HM^2 aggregates node attributes via the linkage-aware multi-head attention mechanism [8], which effectively incorporates the relationships into node embedding. Then, HM^2 utilizes adaptive weighted ensemble to aggregate multi-modal node embedding, which can capture modal interactions and get more descriptive capabilities. Moreover, the loss function includes the extra triplet loss, which considers the structure with competitors' embeddings except for normal company valuation loss, and aims to enhance the embedding presentation capability by multi-task operator. To the best of our knowledge, we are the first to formalize the corporate relative valuation into an inductive learning problem considering heterogeneous graph structure. To summarize, the main contributions are:

- We formalize the corporate relative valuation as the heterogeneous multi-modal graph structure, which includes heterogeneous nodes, linkages and multi-modal node attributes in specific;
- We develop HM^2, a heterogeneous multi-modal graph neural network, which considers heterogeneous nodes and linkages for node embeddings comprehensively, and combines specific structure loss for final prediction. In result, HM^2 can be effectively applied to corporate relative valuation;
- We conduct extensive experiments on collected real-world corporate valuation dataset, and our results demonstrate the effectiveness of HM^2.

2 Preliminaries

In this section, we declare our motivation, deliver the definition of CRV with heterogeneous company-member graph, and then introduce the adopted real-world data. In addition, we also introduce existing heterogeneous graph neural networks.

2.1 Motivation

In real applications, as shown in Fig. 1, relative valuation performed manually always considers two factors [9], [10]: 1) the company and its affiliates. 2) the core members of the company. Therefore, similar to the citation network (author-article) [11], [12], the companies and members can be

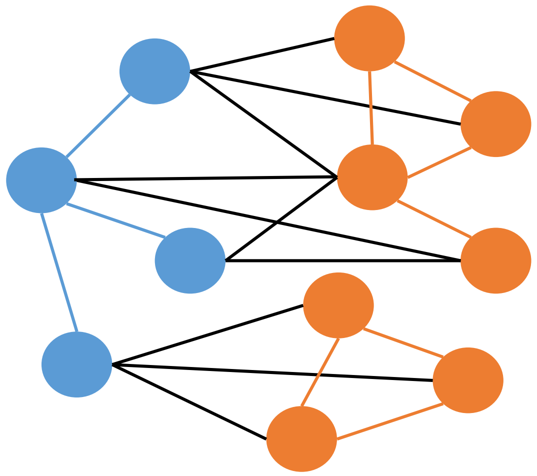


Fig. 2. (Best viewed in color.) The illustration of heterogeneous multi-modal graph. There exist various types of nodes, i.e., we utilize two types here for simplicity (blue and yellow). Meanwhile, the linkages among nodes are also with multiple types, i.e., blue, black, and yellow solid lines. Different types of nodes can be represented by various attributes.

regarded as entities, and the connections among them can be regarded as linkages. Consequently, the data naturally construct a complex heterogeneous multi-modal graph. Essentially, it is unstructured data for the reason that: 1) the graph size is arbitrary, the topological structure is complex, and there is no spatial locality like images; 2) the graph does not have a fixed order of nodes; and 3) the graph is dynamic and contains multi-modal features. If we directly concatenate the company embedding and the member embedding as a single example, the neighbor representation and structural information of the sample cannot be considered. Therefore, in this paper, we develop a deep heterogeneous graph method for CRV. The experimental comparison with traditional methods also verifies the effectiveness of the graph embedding.

2.2 Problem Definition

First, we formalize the definition of heterogeneous graph with multi-modal information.

Definition 1. Heterogeneous Multi-Modal Graph (HMMG). *(also known as Content-associated Heterogeneous Graph [13]) As shown in Fig. 2, HMMG is defined as a graph* $G = (V, E, C_V, C_E)$ *with node set* V*, linkage set* E*, node type set* C_V*, and linkage type set* C_E*. Different from homogeneous graph that nodes belong to a single type, HMMG owns multiple node types, and different types of nodes have various linkage representations. Moreover, attributes of different types of nodes constitute multi-modal data, i.e., different types of nodes have various raw dimensional attribute representations.*

With Definition 1, we can observe that the company penetration graph actually is an HMMG with two types of nodes. In detail, the node type set C_V includes: *company and member*, and the linkage type set C_E includes: *company-company, company-member and member-member*. Then, we can define the corporate relative valuation.

Problem 1. Corporate Relative Valuation (CRV). *CRV aims to estimate the relative valuation or value level, i.e., a regression or classification problem, without the financial statement data. CRV is widely used for startups and unlisted companies, and always considers the core resources, members, and competitors of the input company. Traditional CRV usually relies on experienced experts.*

In summary, we now define the CRV problem with HMMG representation learning. Without any loss of generality, we provide both approximate (coarse-grained) and accurate (fine-grained) valuations of the input companies in experiments.

Definition 2. Heterogeneous Multi-Modal Graph Network for Corporate Relative Valuation. *Given an HMMG* $G = (V, E, C_V, C_E)$*, each corporate node* v_i^1 *in* G *has its own attribute* $\mathbf{x}_i^1$*, and is with two ground truth, i.e.,* $\mathbf{y}_i^b \in R^{L_b}$ *denotes the business category, with* L_b *represents the dimension, and* $\mathbf{y}_i^p \in R^{L_p}$ *denotes the corporate relative valuation level, with* L_p *also denotes the dimension. Besides, each member node* v_j^2 *in* G *also has corresponding descriptions* $\mathbf{x}_j^2$*. The task is to design a model* f *that able to estimate corporate relative valuation level* $\mathbf{y}_i^p$ *of these companies, and the key challenge of* f *is to learn company's embedding, which encodes both structural relationships and node attributes.*

Note that the ambition is to estimate the corporate value, thereby we concentrate on the embedding of the company nodes, i.e., v_i^1, in this paper.

2.3 Data Descriptions

The real-world corporate valuation dataset is provided by our business partner, and consists of companies in the internet industry. There are several reasons to utilize the data from the internet industry: 1) With the development of the Internet, most of the recent emerging companies are belong to the internet industry, and serious data missing is a universal problem among companies in other different domains; 2) Considering the cost of data collection, Internet industry takes up the largest number of companies in the collected data; and 3) The heterogeneous graphs of internet companies have the relevance island problem with the heterogeneous graphs of other domains, i.e., few connections between two domains' heterogeneous graphs. Thereby it is difficult to consider all domains in one unified graph. Note that these companies are mostly startups or unlisted companies, and are in need of relative valuation. The data can be represented as HMMG in Definition 1 naturally. The original data will be published after permission.

In detail, the graph consists two types of nodes: company and core member, the corporate and member nodes are associated with their own attributes, i.e., company has 132 dimensional features, which cover basic information, legal proceedings, business conditions, intellectual property and so on. Member has 5 dimensional features, which are extracted from personal information. And member nodes are concatenated with 45 dimensional embeddings using node2vec [14]. Besides, the linkages constitutes three types: 1) *company-company* linkages have two predicates, i.e., *investment, acquisition*. We present the representation of linkage as investment ratio, and acquisition is denoted by 1. 2) *company-member* linkages have nine predicates, for example, chief executive officer (CEO), chief operating officer (COO), chief technology officer (CTO), chief financial officer (CFO)

TABLE 1
Datasets Used in This Work

Data	Node	Edge	CRV	BC
ICV	Company: 4362	Company-Company: 5106	4	7
		Company-Member: 13123		
	Member: 6877	Member-Member: 28224		

ICV denotes Internet corporate valuation, CRV represents corporate relative valuation level, and BC represents business class.

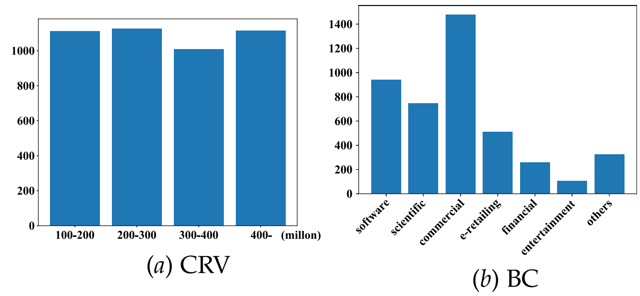


Fig. 3. Data visualization. (a) is the number of each class in relative valuation level (CRV) and (b) is the number of each class in business category (BC).

and related derivative positions. We apply one-hot encoding to the company-member linkage. 3) *member-member* linkages have one predicate, i.e., whether they belong to the same company. The main statistics of the dataset are shown in Table 1.

Moreover, the relative valuation level has 4 categories, i.e., 100 millions to 200 millions, 200 millions to 300 millions, 300 millions to 400 millions, and 400 millions above, and the business class (BC) contains 7 categories, i.e., software service, scientific research and technology service, commercial service, e-retailing service, financial service, entertainment service, and others. The visualizations of CRV and BC are shown in Fig. 3, and the figure reveals that the instances distribute evenly among valuation categories, but unbalanced among business categories. Therefore, the relevance of instances within each business field needs to be effectively considered. Note that we conduct experiments with the real corporate value after using log operator.

2.4 Heterogeneous Graph Neural Network

In this section, we present a generic definition of heterogeneous graph neural networks (HGNN). HGNN is mainly based on neighbor aggregation architecture, which emphasizes on processing different types of nodes respectively [13], [15], [16]. In detail, HGNN usually samples different types of neighbors for each input node, then encodes them respectively, and finally aggregates different embeddings into a uniform embedding. The key idea of HGNN is to process various types of neighbors for node v_i, which can be commonly expressed as [13], [15]

$$f_t(v_i) = \frac{1}{|N_t(v_i)|} \sum_{v' \in N_t(v_i)} \overrightarrow{LSTM}\{\mathbf{x}(v')\} \oplus \overleftarrow{LSTM}\{\mathbf{x}(v')\}, \tag{1}$$

where t denotes the node type, $N_t(\cdot)$ is the neighbor set of input node, $|N_t(\cdot)|$ is the set size, $\mathbf{x}(\cdot)$ represents the attribute of node, and $\oplus$ denotes concatenation. The single LSTM can be formulated as

$$\begin{aligned}
\mathbf{z}_i &= \sigma(W_z\mathbf{x}(v') + U_z\mathbf{h}_{i-1} + \mathbf{b}_z),\\
\mathbf{g}_i &= \sigma(W_g\mathbf{x}(v') + U_g\mathbf{h}_{i-1} + \mathbf{b}_g),\\
\mathbf{o}_i &= \sigma(W_o\mathbf{x}(v') + U_o\mathbf{h}_{i-1} + \mathbf{b}_o),\\
\hat{\mathbf{c}}_i &= tanh(W_c\mathbf{x}(v') + U_c\mathbf{h}_{i-1} + \mathbf{b}_c),\\
\mathbf{c}_i &= \mathbf{g}_i \odot \mathbf{c}_{i-1} + \mathbf{z}_i \odot \hat{\mathbf{c}}_i,\\
\mathbf{h}_i &= tanh(\mathbf{c}_i) \odot \mathbf{o}_i,
\end{aligned}$$

where $\mathbf{h}_i$ is the hidden state of ith node, $W_j, U_j, b_j \quad j \in \{z, g, o, c\}$ are learnable parameters, and $\mathbf{z}_i, \mathbf{g}_i, \mathbf{o}_i$ are forget gate vector, input gate vector, and output gate vector of ith node respectively. $\odot$ denoted element-wise product.

Here LSTM module employs Bi-LSTM [17] to capture deep feature interactions. The Bi-LSTM operates on an unordered content set, which is inspired by previous work [18] for aggregating unordered neighbors. In detail, the LSTM based module first transform different neighbors with the same type into a common embedding space, then employs the Bi-LSTM to accumulate deep feature representations of all neighbors, and utilizes a mean pooling operator over all hidden states to obtain the general content embeddings. HGNN establishes corresponding LSTM models for different types of nodes and fuses them to obtain the final feature embeddings. However, it is notable that Bi-LSTM in Equation (1) acts as a multi-instance learning operator, which only aggregates the information of heterogeneous neighbors, yet has not considered the linkages among neighbors. Therefore, Bi-LSTM may lose more information during feature embedding.

3 Proposed Method

The usage of HGNN for CRV task mainly faces the following challenges: 1) Nodes in heterogeneous graph connect to different types of neighbors, and the number of their neighbors varies, for example, member nodes usually contains more neighbors than company nodes. Thus, we need to design an effective neighbor sampling method to consider both the number and type of sampling comprehensively. 2) Heterogeneous neighbors contain different modal feature descriptions, and the linkages among homogeneous and heterogeneous neighbors are also inconsistent. Therefore, we need to design corresponding fusion networks for heterogeneous neighbors, and consider the linkages when learning embeddings. 3) Different types of neighbors contribute differently to node embeddings in heterogeneous graph, thus we need to adaptively learn the weights of heterogeneous nodes for final fusion.

Based on the considerations above, we formally present HM^2, which consists of there modules: 1) modal attribute encoding module, 2) multi-modal aggregation module, and 3) valuation loss module.

- *Modal Attribute Encoding Module:* This module encodes each type of neighbors respectively after neighbor sampling, i.e., single modal attribute embedding. The key idea is to take both nodes' attributes and their relations into consideration for learning overall embedding;
- *Multi-Modal Aggregation Module:* This module adaptively aggregates different types of neighbors and

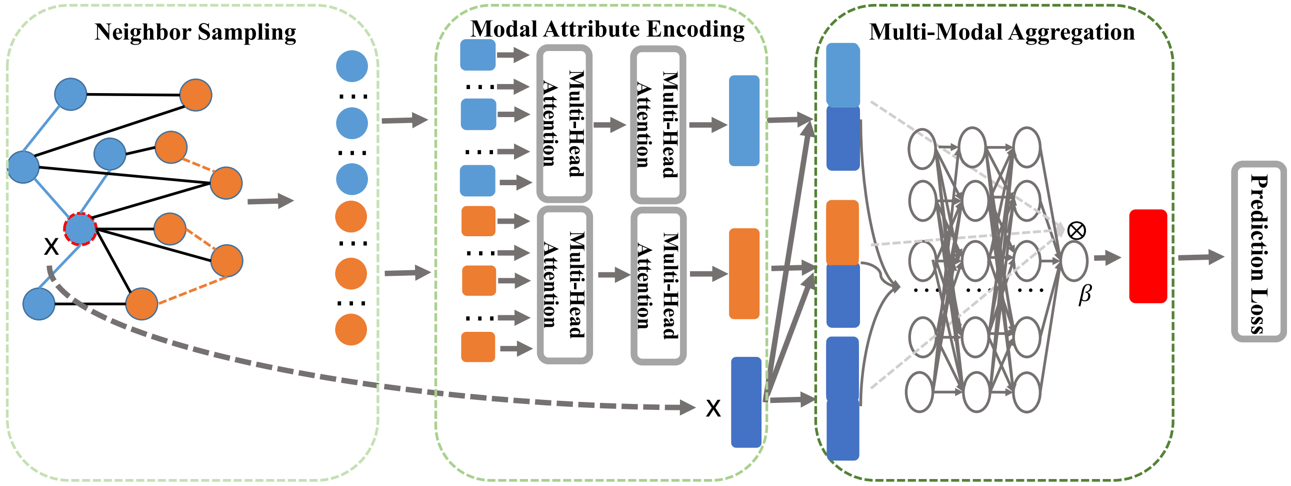


Fig. 4. The overall architecture of HM[2]. From left to right, HM[2] first samples fix sized neighbors, which include heterogeneous types. Then it encodes each modal attributes via deep network with multi-head attention mechanism, and aggregates multi-modal embedding through adaptive attention. Finally, it develops the loss via corporate valuation with structural triplet regularization.

input node itself, i.e., the multi-modal embedding weighted aggregation. The key idea is to learn adaptive weights for each modal information;

- *Valuation Loss Module:* This module considers the loss of valuation prediction, while incorporates the similarities of different corporate node embeddings in the same business category, which aims to regularize the consistence.

An overview of HM[2] is shown in Fig. 4. Specifically, for the input company node (i.e., dotted blue node), we will first sample heterogeneous neighbors (blue and yellow nodes) of the node. Second, we develop separate deep feature learning networks to aggregate information of neighbors, which combines multi-head attention mechanism considering various types of linkages in the learning process. Finally, we comprehensively consider the embeddings of input nodes and their neighbors using self-attention mechanism to acquire final embeddings, which are used for predicting the company's valuation. Table 2 provides the definition of symbols used in this paper.

3.1 Modal Attribute Encoding Module

The most critical component of graph neural networks (GNNs) [18] is to aggregate attributes of neighbors for representing input node. However, heterogeneous graphs have multiple types of nodes, rather than homogeneous type considered in previous methods. Therefore, the embedding of heterogeneous graph faces two challenges: a) sample heterogeneous neighbors for each node in HMMG; b) construct node encoder for each type of node in HMMG.

Neighbor Construction. The neighbor construction here aims to provide more useful structural auxiliary information for input node, so that learns more discriminative node embedding. The common method for neighbor sampling is to sample direct neighbors of each node, i.e., first-order neighbors. Nevertheless, first-order neighbors have several limitations as mentioned in [19]: 1) Be susceptible to interference. Nodes have limited first-order neighbors, thus noise neighbors with incorrect relations or attributes may impair the embedding; 2) Information loss. The effects of non-direct neighbors are lost by aggregating attributes of direct neighbors only, and limited neighbors may lead to insufficient embedding, for example, node A has five direct neighbors while node B only has three; and 3) Aggregation difficulty. Sampling direct neighbors is unsuitable for aggregating heterogeneous information that contains different modal features. Therefore, sampling only direct neighbors may play a negative role. Heterogeneous neighbors require different transformations to deal with various feature types and dimensions.

To solve this problem, inspired from [13], [14], HM[2] utilizes the random walk sampling for each node. In detail, it contains two steps:

- *Step 1:* Sampling fixed size l of neighbors with random walk sampling. In detail, the sampling process

TABLE 2
Description of Symbols

Sym.	Definition
V	set of nodes with different types, i.e., v^1 (company), v^2 (member)
E	set of edges with different types
C_V	T types of node: company and member
C_E	M types of linkage: company-company, company-member member-member
$\mathbf{x}$	attribute of each node, i.e., $\mathbf{x}^1 \in \mathbb{R}^{d_1}$ (company), $\mathbf{x}^2 \in \mathbb{R}^{d_2}$ (member)
$\mathbf{y}$	ground truth of each company node, i.e., $\mathbf{y}^b$ (business category), $\mathbf{y}^p$ (valuation level)
f_1	modal attribute encoding module
q_j^l	the embedding of jth node in lth layer
$\alpha_{i,j}^{l,h}$	the learnable weight between nodes i and j in lth layer with hth head
H_l	number of embedding aggregation head in lth layer
$p_{i,j}$	embedding of the directed edge between nodes i and j
$at_{i,j}$	predicate of edge between nodes i and j
f_2	multi-modal aggregating module

starts from input node $v_i^1 \in V$ (superscript 1 represents the company node), and iteratively random walks to neighbors of current node or returns to v_i^1 with a probability. The process ends until collecting l nodes, i.e., $N(v_i^1), |N(v_i^1)| = l$;
- *Step 2:* Grouping different types of neighbors. For each type, it selects top k_t (t is the node type) nodes from $N(v_i^1)$ according to frequency.

With the procedure above, HM[2] can collect all types of neighbors for each node, and the most frequently visited neighbors are selected. Notably, the number of each type of node in $N(v_i^1)$ is constrained, which ensures the balance of heterogeneous nodes. Note that HM[2] focuses on the embedding learning of corporate nodes, but the member-member linkage can help to collect high-order member neighbors for each node when constructing neighbor.

Modal Attribute Encoding. The majority of previous GNN models focus on homogeneous graphs [20], [21], [22], which ignore the impact of node type. However in HMMG, different types of neighbors contribute differently to node embeddings. For example, mature companies have stronger core resources, thus the attributes of corporate nodes have a greater impact, whereas the core members have a relatively large proportion of impact in several other companies for valuation. On the other hand, different types of nodes have various dimensional attributes, which contain inconsistent physical meanings. Therefore, it is unreasonable to directly aggregate heterogeneous neighbors as traditional GNN models. In other words, heterogeneous multi-modal neighbors require different embedding transformations. To solve this problem, [13], [15], [16] attempt to handle heterogeneous graph embedding with novel deep graph neural networks, in which heterogeneous multi-modal neighbors are encoded separately, and aggregated for final embedding. However, most of these methods only encode heterogeneous neighbors with multi-instance based process, without considering the relationships among nodes. But the correlations play an important role in traditional GNNs, i.e., a weighted metric in neighbor aggregation.

To model the relationships among neighboring nodes, inspired by recent work of attention mechanism [23], we propose a linkage-aware model f_1, rather than directly embedding aggregation. Specifically, f_1 considers two factors: 1) The relationships among nodes. Different relationships play various roles in embedding, for example, investment and acquisition represent different affiliations between two companies, and acquisition indicates stronger relation; and 2) The hierarchical embedding. Direct and non-direct neighbors have different impacts according to feature propagation process mechanism [24], i.e., direct neighboring nodes play relatively more important roles. In summary, f_1 can aggregate homogeneous neighboring attributes, considering the linkages among nodes comprehensively. Without any loss of generality, different types of nodes can have similar modal attribute encoding modules, i.e., corporate and member nodes have similar encoding structures except various dimensional input.

Therefore, as shown in the second part of Fig. 4, with sampled neighbors, tth type of neighbors of v_i^1 (company) are denoted as $N_t(v_i^1)$. We refer to the self-head attention mechanism, which performs embedding aggregation and attention computation simultaneously. Formally, the self-head attention aggregation can be formulated as

$$q_k^l = \sum_{j \in N_t(i) \cup \{i\}} \alpha_{k,j}^l q_j^{l-1}, \tag{2}$$

where l/k denotes hidden layer index ($l = 1, 2, \ldots, L$) and node index ($k \in N_t(i) \cup \{i\}$), $\alpha_{k,j}^l$ is a learnable weight between nodes k and j. q_j^{l-1} denotes the embedding of node j of $l-1$th layer output, where $q_j^0 = \Phi(\mathbf{x}_j)$ is the transformed representations in common space from raw attribute, i.e., $q^0 \in \mathbb{R}^d$. $\alpha_{k,j}^l$ acts as self-attention operator, which is a single layer forward neural network, and can be formalized as

$$\alpha_{k,j}^l = \frac{exp\Big((\omega_l^\top (\Psi(p_{k,j}) \| at_{k,j})[q_k^{l-1} \odot q_j^{l-1}]) \Big)}{\sum_{n \in N_t(i) \cup \{i\}} exp\big((\omega_l^\top (\Psi(p_{k,n}) \| at_{k,n})[q_k^{l-1} \odot q_n^{l-1}]) \big)}, \tag{3}$$

where $\omega_l^\top$ represents the weight matrix for lth layer, and $\odot$ denotes the vector point multiplication. $p_{k,j} \in \mathbb{R}^{d_p}$ denotes the one-hot representation of directed edge between nodes k and j, and $\Psi(\cdot)$ denotes mapping from raw linkage representation to its embedding. $at_{k,j} \in \mathbb{R}$ represents the link predicate, i.e., $at_{k,j}$ is the investment ratio for company-company linkages, and $at_{k,j} = 1$ for company-member and member-member linkages. $\|$ denotes concatenation operator, and $p_{k,j} = at_{k,j} = \xi$ if there exists no direct linkage between nodes k and j. ξ is always with a small value (i.e., 10^{-3} in experiment).

We can also extend f_1 to a more general architecture, in which each layer contains a variable number of attribute aggregation head. And multiple heads can promote the performance and optimization stability. Therefore, Equations (2) and (3) can be reformulated as

$$\begin{aligned} q_k^l &= \frac{1}{|H_l|} \sum_h \sum_{j \in N_t(i) \cup \{i\}} \alpha_{k,j}^{l,h} q_j^{l-1}, \\ \alpha_{k,j}^{l,h} &= \frac{exp\Big((\omega_{l,h}^\top (\Psi(p_{k,j}) \| at_{k,j})[q_k^{l-1} \odot q_j^{l-1}]) \Big)}{\sum_{n \in N_t(i) \cup \{i\}} exp\Big((\omega_{l,h}^\top (\Psi(p_{k,n}) \| at_{k,n})[q_k^{l-1} \odot q_n^{l-1}]) \Big)}, \end{aligned} \tag{4}$$

where h denotes the hth head, and H_l is the number of heads in lth layer. Consequently, we can formalize the final aggregated embedding output of tth type of neighboring nodes as

$$f_1^t(v_i^1) = \frac{1}{|N_t(i)|} \sum_{k \in N_t(i)} q_k^L. \tag{5}$$

Thus, $\alpha_{k,j}^l$ can well measure the relationships between input nodes and different types of neighbors, while considering the impact of direct and non-direct neighbors. f_1 computes the aggregated embeddings by performing a weighted aggregation of intermediate, and the learnable weight α can effectively overcome the two problems mentioned above.

3.2 Multi-Modal Aggregating Module

In this section, we aim to aggregate different modal embeddings for final representation. As shown in the third part of Fig. 4, different from concatenating multi-modal embedding directly [25], we turn to design a novel adaptive attention based network to capture more discriminative feature capability. Formally, the final representation of v_i^1 can be computed as

$$f_2(v_i^1) = \sum_{j \in \{T, i\}} \beta_j \hat{f}_1^j(v_i^1), \tag{6}$$

where β_j is the adaptive weights of each modal embedding, which aim to discover the relationships among different modalities. β_j can be formulated as

$$\beta_j = \frac{exp\{LeakyReLU(u^\top \hat{f}_1^j)\}}{\sum_k exp\{LeakyReLU(u^T \hat{f}_1^k)\}},$$

where LeakyReLU denotes leaky version of a Rectified Linear Unit, and $u \in R^{2d \times 1}$ is the parameter. $\hat{f}_1^j \in R^{2d}$ denotes the concatenated embeddings

$$\hat{f}_1^j = \begin{cases} f_1^j \| \Phi(\mathbf{x}_i^1), & when \quad j \neq i, \\ \Phi(\mathbf{x}_i^1) \| \Phi(\mathbf{x}_i^1), when & j = i. \end{cases}$$

where $\Phi(\mathbf{x}_i^1)$ is the mapping introduced in Section 3.1, and $\|$ denotes concatenation operator.

3.3 Model Training

To perform corporate relative valuation for input node, we train HM^2 from two aspects: 1) corporate relative valuation loss, and 2) heterogeneous graph representation structural loss. This constructs a multi-task learning approach, which can learn more discriminative representation. In result, the overall loss function is

$$\begin{aligned}
&\ell = \ell_m + \lambda \ell_b, \\
&\ell_m = -\sum_{i \in V^1} \sum_j \mathbf{1}\{\mathbf{y}_i^p = j\} log \frac{exp(\theta_j^\top f_2(v_i))}{\sum_k exp(\theta_k^\top f_2(v_i))}, \\
&\ell_b = \sum_{<i,j,k> \in \mathcal{T}} max\{0, \mu + d(f_2(v_i), f_2(v_j)) - d(f_2(v_i), f_2(v_k))\}, \\
&s.t. \quad y_i^b = y_j^b \neq y_k^b, \quad y_i^p = y_j^p = y_k^p,
\end{aligned} \tag{7}$$

where ℓ_m denotes the corporate valuation loss, θ is the fully connected layer to the prediction layer, μ represents the margin defined manually, and $d(\cdot)$ is the distance measurement function, which measures the distance between two node embeddings (we utilize the euclidean distance here for simplicity). Note that ℓ_b reflects the embedding effect between competitors considered by traditional domain experts, thus further regularizes the embedding structure in the same business category. Inspired by [26], we can corporate the random walk in graph to generate triplets $< i, j, k > \in \mathcal{T}$. In detail, we first generate a set of random walks in the HMMG. Then we collect node j with same the same business category y_i^b and valuation level y_i^p for node i in the walk sequence. Besides, we sample node k with the same valuation level y_i^p but different business category y_i^b for node i. For optimizing HM^2, we first sample a mini-batch of triplets at each iteration, and calculate the objective according to Equation (7). The model parameters are updated via the Adam optimizer [27]. And we utilize extra 10% randomly sampled data as early stop for better generalization. With the learned model, we can conduct inductive corporate relative valuation.

Algorithm 1. The Pseudo Code of HM^2

Input:

- Dataset: HMMG (V, E, C_V, C_E), attribute $\mathbf{x}$, ground truth $\mathbf{y}$;
- Parameter: λ;
- $\mathrm{maxIter}$: T, learning rate: l_r

Output:

- Classifiers: F

```
1:  Initialize HM^2 model parameters Θ;
2:  while stop condition is not triggered do
3:      for mini-batch of company node v^1 do
4:          Gather neighbour nodes n for each node in batch via
            random walk;
5:          Select neighbour company and member nodes with
            highest frequency, N^1(v^1) and N^2(v^1) respectively;
6:          Calculate f_1^t(v_i^1) according to Equation (5);
7:          Calculate f_2(v_i^1) according to Equation (6);
8:          Calculate ℓ_m;
9:          Sample triplets T and calculate ℓ_b;
10:         Calculate loss ℓ = ℓ_m + λℓ_b according to Equation (7);
11:         Update model parameters using gradient descent;
12:     end for
13: end while
```

The procedure of training HM^2 model can be summarized in Algorithm 1. Line 4 and line 5 correspond to our neighbour sampling module. Line 6 and Line 7 calculate the proposed modal attribute encoding and the multi-modal aggregating results respectively. Line 8 and line 9 calculate the classification loss and triplet margin loss respectively. In each epoch, HM^2 samples mini-batches of company nodes v^1 and update model parameters using gradient descent.

4 Experiments

In this section, we develop related experiments to validate the effectiveness of our proposed method.

- *Q1:* How do HM^2 and state-of-the-art comparison methods perform on real-world dataset? For example, the prediction performances in Definition 1.
- *Q2:* How do the components of HM^2 affect the estimation? For example, modal attribute encoding.
- *Q3:* How do various hyper-parameters in the approach affect performance? For example, the size of sampled neighbor and the embedding size.

We compare our HM^2 model with three traditional methods: SVM, MLP, KNN, and seven state-of-the-art graph methods: HetGNN [19], m2vec [28], ASNE [29], GraphSAGE (SAGE for simplicity) [18], GAT [25], HAN [30], and GATNE [31]. All the methods are given the best performance as [19]. The details are:

- *SVM*: A linear method that considers single modal features as input, in detail, we develop the attributes of corporate node as the input;

- *MLP*: A fully connected network that considers single modal features as input, in detail, we develop the attributes of corporate node as the input;
- *HetGNN*: A heterogeneous graph neural network model that constructs two modules to aggregate feature information of heterogeneous nodes respectively, in which the first module learns embeddings of heterogeneous contents with the LSTM module, and the second module aggregates embeddings of different neighboring types for obtaining the final node embedding [13];
- *m2vec*: A heterogeneous graph model that leverages meta-path based random walks in heterogeneous networks to generate heterogeneous neighborhoods, then extends the skip-gram model to facilitate the modeling of connected nodes. The model can preserve both the structures and semantics of the given heterogeneous network by maximizing the likelihood [28];
- *ASNE*: An attributed graph embedding method, which learns representations for nodes by preserving both the structural proximity (capturing the global network structure) and attribute proximity [29];
- *GraphSAGE (SAGE for simplicity)*: An inductive graph neural network model that leverages node feature to efficiently generate node embeddings for unseen data. The model generates embeddings through sampling and aggregating features from a node's local neighborhood by different neural networks, i.e., LSTM [18];
- *GAT*: A graph network model, which aggregates neighbors' information by masked self-attentions [25];
- *HAN*: A novel heterogeneous graph neural network based on the hierarchical attention, including node-level and semantic-level attentions;
- *GATNE*: A heterogeneous network that splits the overall node embedding into three parts: base, edge, and attribute embedding. The base embedding and attribute embedding are shared among edges of different types, while the edge embedding is computed by aggregation of neighborhood information with the self-attention mechanism;
- *FAME*: A heterogeneous network that maps the units from different modalities into the same latent space, which can preserve both attribute semantics and multi-type relations in the learned embeddings.

HetGNN, ASNE and m2vec are unsupervised graph node embedding learning methods. Thereby we use the source code provided by the authors, and modify our dataset to conform to their input formats. For m2vec, we employ three meta-paths, i.e., company-company, company-member-company, and company-member-member-company respectively. In addition, the walk length is set to 300. For ASNE, we employ the same content features of different modalities as HM^2 and concatenate them as general features besides the latent features. For GraphSAGE and GAT, we use the same input features and sampled neighbors set for each node as HM^2. With the learned embeddings, we train an MLP model to obtain a classifier or regressor as HM^2. GAT and SAGE are two transductive graph learning methods, we use a mask to indicate training and test nodes. For all the compared methods, we use the same early stop criterion.

4.1 Implementation

The number of embedding aggregation head H is 2, the dimension of edge embedding p is set as 32, and the hidden layer of node embedding q is set as 128 dimensions. f_1 is a two-layer modal attribute encoding module, and f_2 is two-layer fully connected network. The batch size is set as 32, triplet loss margin is 1.0 in experiment, random walk length is set as 300, and the probability of returning to the starting point is 0.05. The λ in Equation (7) is tuned with cross validation. The validation set is randomly selected from the training set by 10%. When the validation set loss does not decrease within 50 epochs, the training will be stopped. For comparing methods, we adjust the hyper-parameters according to the original paper to acquire their optimal results. We implement HM^2 on a server with GPU machines (Nvidia 2080ti).

4.2 Corporate Relative Valuation

To answer Q1, we design experiments to evaluate HM^2 on corporate relative valuation task. We give both approximate (coarse-grained range y) and accurate (fine-grained $\hat{y}$) valuations, in which $\hat{y}^p$ denotes the real value with log operator of corporate valuation.

Similar to traditional node classification task, we first use training data to build the model, then employ the learned model to predict nodes in test data. The ratio of training data is set to 10%, 30%, 50% and 70%, and the remaining nodes are used for test. As a multi-class classification problem, we use Accuracy (Acc), Recall (Rec), Precision (Pre), and F1-measure as the evaluation metric. Note that we use weighted average measures of Rec, Pre and F1 considering data imbalance problem. In addition, duplicated companies are removed from the experiments. Table 3 reports results of HM^2 and comparison methods. The results reveal that: 1) Graph embedding methods are superior than traditional linear method, which considers only the information of input node itself; 2) Most methods achieve good performance in the corporate relative valuation, which reflects the effectiveness of machine learning models on simulating the judgment of domain experts; 3) HM^2 achieves the best or comparable performance to comparison methods, which shows that HM^2 can encode effective node embedding for valuation task by considering the heterogeneous node attributes and linkages comprehensively; 4) HM^2 performs better than another attention based graph model GAT, because HM^2 considers heterogeneous neighbors and linkages comprehensively, and uses a more effective multi-head attention mechanism that validates the effectiveness of heterogeneous neighbor construction and fusion; 5) HM^2 performs better than HetGNN, which adopts LSTM to aggregate heterogeneous neighbors, and this indicates that attention mechanism can better employ the linkages; 6) HM^2 performs better than FAME, the reason is that HM^2 takes the relations as feature vectors, and utilizes an extra mapping function $\Psi(\cdot)$ for better learning the similarity between two nodes; and 7) With the increase of training data, the performance of HM^2 improves faster than other methods, for the reason that HM^2 employs the triplet loss by considering the embedding structure, and can better reflect the global structure of graphs with the increase of training data. Moreover, we regard the corporate relative

TABLE 3
Corporate Relative Valuation Prediction Results (y), Percentage Denotes Training Data Ratio

Metric		SVM	MLP	KNN	HetGNN	GAT	SAGE	ASNE	m2vec	HAN	GATNE	FAME	HM^2
Accuracy	10%	.273 ± .002	.247 ± .014	.326 ± .012	.307 ± .004	.313 ± .011	.304 ± .012	.289 ± .011	.299 ± .008	.310 ± .011	.304 ± .004	.322 ± .014	**.346 ± .008**
	30%	.305 ± .003	.305 ± .047	.333 ± .005	.353 ± .009	.346 ± .008	.364 ± .011	.359 ± .002	.342 ± .010	.347 ± .011	.330 ± .004	.349 ± .002	**.388 ± .004**
	50%	.336 ± .002	.340 ± .046	.347 ± .009	.377 ± .005	.378 ± .003	.395 ± .007	.387 ± .007	.360 ± .002	.393 ± .009	.346 ± .002	.380 ± .005	**.410 ± .008**
	70%	.367 ± .002	.374 ± .029	.349 ± .010	.393 ± .008	.407 ± .009	.413 ± .010	.399 ± .003	.388 ± .016	.403 ± .009	.357 ± .007	.390 ± .010	**.446 ± .007**
Precision	10%	.286 ± .003	.172 ± .097	.331 ± .010	.301 ± .009	.319 ± .313	.305 ± .013	.290 ± .014	.294 ± .010	.309 ± .011	.304 ± .003	.323 ± .011	**.351 ± .006**
	30%	.341 ± .006	.266 ± .095	.337 ± .004	.355 ± .011	.339 ± .010	.354 ± .011	.361 ± .004	.335 ± .008	.346 ± .007	.329 ± .005	.351 ± .004	**.395 ± .004**
	50%	.334 ± .005	.292 ± .102	.351 ± .009	.386 ± .007	.375 ± .003	.386 ± .007	.386 ± .013	.362 ± .005	.386 ± .014	.344 ± .003	.377 ± .007	**.405 ± .012**
	70%	.358 ± .004	.384 ± .021	.355 ± .013	.385 ± .012	.404 ± .009	.408 ± .010	.400 ± .002	.385 ± .016	.398 ± .013	.354 ± .006	.383 ± .011	**.428 ± .004**
Recall	10%	.273 ± .007	.247 ± .014	.326 ± .012	.307 ± .004	.313 ± .011	.304 ± .015	.289 ± .011	.299 ± .008	.310 ± .011	.304 ± .004	.322 ± .014	**.346 ± .008**
	30%	.305 ± .004	.305 ± .047	.333 ± .005	.353 ± .009	.346 ± .010	.364 ± .011	.359 ± .002	.338 ± .010	.347 ± .011	.330 ± .004	.349 ± .002	**.388 ± .004**
	50%	.336 ± .006	.340 ± .046	.346 ± .009	.377 ± .005	.378 ± .003	.395 ± .007	.384 ± .007	.358 ± .002	.393 ± .009	.346 ± .002	.380 ± .005	**.410 ± .012**
	70%	.367 ± .005	.374 ± .029	.349 ± .010	.393 ± .008	.407 ± .009	.413 ± .010	.399 ± .003	.388 ± .016	.403 ± .009	.357 ± .007	.390 ± .010	**.446 ± .005**
F1-measure	10%	.175 ± .003	.122 ± .038	.326 ± .011	.301 ± .010	.313 ± .011	.302 ± .012	.288 ± .013	.292 ± .016	.301 ± .010	.297 ± .009	.322 ± .013	**.340 ± .009**
	30%	.269 ± .004	.220 ± .085	.334 ± .004	.345 ± .016	.339 ± .008	.355 ± .012	.359 ± .003	.335 ± .012	.334 ± .018	.328 ± .005	.349 ± .004	**.376 ± .005**
	50%	.327 ± .003	.275 ± .085	.347 ± .009	.377 ± .006	.376 ± .003	.389 ± .007	.381 ± .009	.359 ± .004	.385 ± .010	.343 ± .003	.375 ± .009	**.400 ± .008**
	70%	.351 ± .004	.333 ± .053	.350 ± .011	.381 ± .015	.405 ± .009	.410 ± .010	.398 ± .003	.385 ± .017	.397 ± .013	.351 ± .007	.384 ± .011	**.424 ± .007**

The best results are highlighted in bold.

valuation as a regression problem, i.e., $\ell_m = \|\hat{y}_i^p - \theta^\top f_2(v)\|_2$. Table 4 reports the MSE results of HM^2 and comparison methods. The results reveal that: 1) The machine learning methods also have considerable performance on accurate relative valuation prediction; and 2) HM^2 also achieves the best or comparable performance, which is much better than comparison methods even under low training data ratio, i.e., HM^2 performs better with only 10% training data.

A notable phenomenon is that various methods do not have significant performance: 1) Even the performance of the best method are not significant; and 2) The promotions between deep methods and linear method, and the promotions between HM^2 and other deep graph models are not significant. This is because: 1) Considering the data privacy and field limitations, the amount of data is relatively small, which affects the training of deep models; and 2) Considering the feature missing and information insufficiency, the information contained in raw multi-modal data is limited.

4.3 Analysis of HM^2

To answer Q2, we design ablation studies for evaluation.

Ablation Study. To explore the role of each module in HM^2, we conduct extra ablation studies to evaluate performances of several variants, including:

- *HM^2-N*: The variant of HM^2 that only adopts the direct neighbors, without considering the higher-order neighbors;
- *HM^2-L*: The variant of HM^2 that replaces the transformer based attribute encoding module with traditional Bi-LSTM to encode heterogeneous node;
- *HM^2-FC*: The variant of HM^2 that replaces the transformer based attribute encoding module with fully connected network to encode heterogeneous node;
- *HM^2-R*: The variant of HM^2 that doesn't consider the relation embedding in Equation (3);
- *HM^2-A*: The variant of HM^2 that replaces attention based multi-modal aggregation module with directly concatenating multi-modal embeddings;
- *HM^2-B*: The variant of HM^2 that removes the triplet loss in Equation (7).

The results of prediction are reported in Tables 5 and 6. They reveal that: 1) HM^2 behaves better than HM^2-N, which demonstrates that neighbor sampling is effective for subsequent operation and embedding generalization; 2) HM^2 behaves better than HM^2-L and HM^2-FC, which shows that linkage-aware multi-head attention based encoding outperforms other methods without considering the linkages, and is beneficial for learning attribute interactions; 3) HM^2 behaves competitive to HM^2-A, which reveals that the self-attention mechanism has a slight advantage, and both kinds of neighbors have relative contributes for prediction; 4) HM^2 performances better than HM^2-R, which indicates the effectiveness of linkage representation in learning node embedding; and 5) HM^2 performs better than HM^2-B, which shows that triplet loss can take graph structure into full account to learn more discriminative embedding.

TABLE 4
Corporate Relative Valuation Results ($\hat{y}$), Percentage Denotes Training Data Ratio

Metric		SVM	MLP	KNN	HetGNN	GAT	SAGE	ASNE	m2vec	HAN	GATNE	FAME	HM^2
MSE	10%	4.277 ± .057	4.268 ± .212	4.188 ± .077	4.122 ± .033	5.247 ± .304	4.998 ± .176	3.822 ± .080	4.599 ± .081	4.496 ± .198	4.927 ± .060	4.796 ± .093	**3.919 ± .077**
	30%	4.029 ± .104	4.170 ± .250	4.165 ± .093	4.002 ± .046	4.225 ± .080	4.517 ± .096	3.823 ± .056	4.458 ± .024	4.207 ± .116	4.729 ± .029	4.607 ± .059	**3.481 ± .051**
	50%	3.818 ± .098	3.856 ± .114	4.143 ± .115	3.868 ± .072	3.872 ± .116	4.366 ± .207	3.638 ± .065	4.247 ± .051	3.970 ± .096	4.633 ± .022	4.561 ± .134	**3.432 ± .071**
	70%	3.523 ± .130	3.680 ± .219	4.127 ± .108	3.723 ± .039	3.453 ± .127	3.981 ± .043	3.487 ± .080	4.005 ± .045	3.858 ± .304	4.622 ± .030	4.475 ± .163	**2.951 ± .084**

The best results are highlighted in bold.

TABLE 5
Ablation Study (y), Percentage Denotes Training Data Ratio

Metric		HM^2-N	HM^2-L	HM^2-FC	HM^2-A	HM^2-B	HM^2-R	HM^2
Accuracy	10%	0.335 ± 0.003	0.329 ± 0.006	0.334 ± 0.009	0.341 ± 0.003	0.326 ± 0.006	0.334 ± 0.005	$\mathbf{0.346 \pm 0.008}$
	30%	0.366 ± 0.003	0.380 ± 0.006	0.347 ± 0.011	0.388 ± 0.002	0.380 ± 0.004	0.385 ± 0.005	$\mathbf{0.388 \pm 0.004}$
	50%	0.396 ± 0.004	0.389 ± 0.014	0.381 ± 0.016	0.381 ± 0.006	0.377 ± 0.008	0.396 ± 0.008	$\mathbf{0.410 \pm 0.008}$
	70%	0.427 ± 0.009	0.434 ± 0.026	0.420 ± 0.017	0.430 ± 0.011	0.424 ± 0.008	0.433 ± 0.006	$\mathbf{0.446 \pm 0.007}$
Precision	10%	0.333 ± 0.003	0.329 ± 0.031	0.337 ± 0.008	0.340 ± 0.002	0.337 ± 0.007	0.334 ± 0.005	$\mathbf{0.351 \pm 0.006}$
	30%	0.379 ± 0.005	0.375 ± 0.003	0.372 ± 0.012	0.393 ± 0.004	0.379 ± 0.004	0.388 ± 0.005	$\mathbf{0.395 \pm 0.004}$
	50%	0.385 ± 0.006	0.377 ± 0.016	0.382 ± 0.026	0.374 ± 0.005	0.377 ± 0.008	0.390 ± 0.016	$\mathbf{0.405 \pm 0.012}$
	70%	0.423 ± 0.008	0.421 ± 0.034	0.418 ± 0.023	$\mathbf{0.432 \pm 0.010}$	0.428 ± 0.009	0.425 ± 0.007	0.428 ± 0.004
Recall	10%	0.335 ± 0.003	0.329 ± 0.006	0.334 ± 0.009	0.341 ± 0.002	0.326 ± 0.010	0.334 ± 0.005	$\mathbf{0.346 \pm 0.008}$
	30%	0.366 ± 0.003	0.380 ± 0.006	0.347 ± 0.011	0.388 ± 0.004	0.380 ± 0.004	0.385 ± 0.005	$\mathbf{0.388 \pm 0.004}$
	50%	0.396 ± 0.004	0.389 ± 0.014	0.381 ± 0.016	0.381 ± 0.009	0.377 ± 0.008	0.396 ± 0.008	$\mathbf{0.410 \pm 0.012}$
	70%	0.427 ± 0.009	0.434 ± 0.025	0.420 ± 0.017	0.430 ± 0.013	0.424 ± 0.008	0.433 ± 0.006	$\mathbf{0.446 \pm 0.005}$
F1-measure	10%	0.331 ± 0.003	0.327 ± 0.022	0.334 ± 0.010	0.340 ± 0.003	0.329 ± 0.006	0.333 ± 0.004	$\mathbf{0.340 \pm 0.009}$
	30%	0.356 ± 0.004	0.371 ± 0.006	0.341 ± 0.013	$\mathbf{0.387 \pm 0.002}$	0.376 ± 0.004	0.382 ± 0.006	0.376 ± 0.005
	50%	0.388 ± 0.004	0.368 ± 0.028	0.347 ± 0.024	0.376 ± 0.006	0.357 ± 0.009	0.387 ± 0.013	$\mathbf{0.400 \pm 0.008}$
	70%	0.418 ± 0.010	0.423 ± 0.026	0.418 ± 0.023	0.425 ± 0.011	0.423 ± 0.007	0.423 ± 0.006	$\mathbf{0.424 \pm 0.007}$

The best results are highlighted in bold.

TABLE 6
Ablation Study ($\hat{y}$), Percentage Denotes Training Data Ratio

Metric		HM^2-N	HM^2-L	HM^2-FC	HM^2-A	HM^2-B	HM^2-R	HM^2
MSE	10%	5.223 ± 0.231	3.921 ± 0.559	5.304 ± 0.487	4.091 ± 0.658	5.808 ± 0.269	4.013 ± 0.216	$\mathbf{3.919 \pm 0.077}$
	30%	3.807 ± 0.104	4.027 ± 0.905	4.022 ± 0.151	3.801 ± 0.068	3.766 ± 0.086	3.605 ± 0.120	$\mathbf{3.481 \pm 0.051}$
	50%	3.731 ± 0.066	3.708 ± 0.261	3.800 ± 0.138	3.505 ± 0.034	3.895 ± 0.053	3.477 ± 0.076	$\mathbf{3.432 \pm 0.071}$
	70%	3.267 ± 0.065	3.167 ± 0.352	3.034 ± 0.162	3.114 ± 0.031	3.251 ± 0.068	3.014 ± 0.058	$\mathbf{2.951 \pm 0.084}$

The best results are highlighted in bold.

4.4 Influence of Linkages

In detail, the linkage type set of the CRV graph consists: 1) *company-company* linkages, 2) *company-member* linkages, and 3) *member-member* linkages. The main statistics are in Table 1 of the main body. Although HM^2 only focused on the embedding learning of company nodes in this paper, when collecting company nodes' neighbors, HM^2 utilizes the random walk sampling for neighbor construction of each node following [14]. Therefore, the company-member and member-member linkages can help collect high-order member neighbors for each node.

To explore the influence of linkages on embedding learning, we have conducted ablation studies to evaluate the mentioned relationships: 1) w/o m-m: HM^2 without member-member linkages, i.e., HM^2 that only samples the company's direct member neighbors; 2) w/o c-m: HM^2 without company-member linkages, i.e., HM^2 that does not sample company's member neighbors; 3) w/o e: HM^2 that does not distinguish edge types, i.e., the linkage is 1 if two nodes connected, otherwise is 0.

Table 7 records the results, and they reveal that: 1) the performance degradation of w/o m-m is not obvious, it indicates that the effect of member-member linkage is weak, for the reason that only the core members (i.e., the direct members) are necessary for company valuation, whereas the member-member linkage has less contribution to company node embedding learning; 2) the performance degradation of w/o c-m is more significant, because member neighbors are useful for learning more discriminative embedding of company node; 3) HM^2 is superior to w/o e, which indicates that it is more meaningful to consider the type of linkages when learning node embedding.

4.5 Hyper-Parameters Study

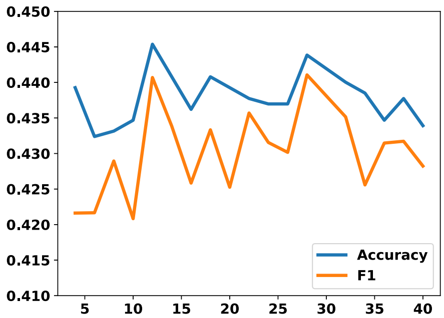
To answer Q3, we also develop hyper-parameter experiments to analyze the impacts of key parameters, i.e., the size of sampled neighbors set. We fix the ratio of training data to 70%, with all valuation level labels. The performances of HM^2 are shown in Fig. 5. Figs. 5a and 5b declare that, with the increase of neighbor size, all evaluation metrics first become better, i.e., accuracy and F1 increase and MSE decreases, and later turn worse after exceeding a certain size, i.e., around 13, which may be caused by the noise and weakly related neighbors. As is demonstrated in Fig. 5, the best neighbor size is $10 - 15$.

Furthermore, Fig. 6 shows the valuation performances of HM^2 embeddings with various dimensions, Fig. 6a reflects the classification task and Fig. 6b reflects the regression task. The dimension d varies from 32 to 256, the figures reveal that all evaluation criteria improve first, i.e., accuracy and F1 increase and MSE decreases, since better embeddings can be learned. However, the performance deteriorate when d further increases, i.e., after 128 dimension, this may be because of over-fitting.

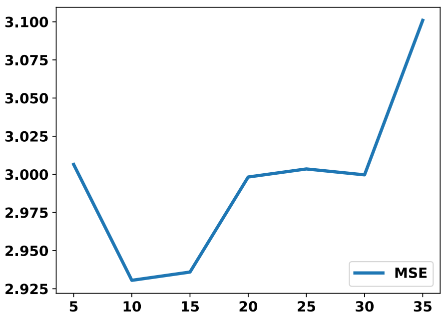
TABLE 7
Corporate Relative Valuation Prediction Results With Ablation Studies Considering Different Linkage Settings, Percentage Denotes Training Data Ratio

Metric		w/o m-m	w/o c-m	w/o e	HM^2
Accuracy	10%	.341 ± .006	.337 ± .005	.334 ± .005	**.346 ± .008**
	30%	.383 ± .005	.385 ± .007	.385 ± .005	**.388 ± .004**
	50%	.402 ± .010	.394 ± .006	.396 ± .008	**.410 ± .008**
	70%	.434 ± .005	.432 ± .005	.433 ± .006	**.446 ± .007**
Precision	10%	.340 ± .006	.339 ± .003	.334 ± .005	**.351 ± .006**
	30%	.385 ± .008	.388 ± .008	.388 ± .005	**.395 ± .004**
	50%	.399 ± .010	.395 ± .009	.390 ± .016	**.405 ± .012**
	70%	.426 ± .008	.428 ± .006	.425 ± .007	**.428 ± .004**
Recall	10%	.341 ± .006	.337 ± .005	.334 ± .005	**.346 ± .008**
	30%	.383 ± .005	.385 ± .007	.385 ± .005	**.388 ± .004**
	50%	.402 ± .010	.394 ± .006	.396 ± .008	**.410 ± .012**
	70%	.432 ± .005	.432 ± .005	.433 ± .006	**.446 ± .005**
F1-measure	10%	.338 ± .005	.337 ± .004	.333 ± .004	**.340 ± .009**
	30%	.372 ± .007	.382 ± .007	**.382 ± .006**	.376 ± .005
	50%	.395 ± .009	.390 ± .007	.387 ± .013	**.400 ± .008**
	70%	.423 ± .008	**.426 ± .006**	.423 ± .006	.424 ± .007
MSE	10%	3.909 ± 0.068	3.877 ± 0.039	4.013 ± 0.216	**3.919 ± 0.077**
	30%	3.541 ± 0.090	3.553 ± 0.095	3.605 ± 0.120	**3.481 ± 0.051**
	50%	3.451 ± 0.062	3.440 ± 0.027	3.477 ± 0.076	**3.432 ± 0.071**
	70%	3.047 ± 0.060	3.013 ± 0.018	3.014 ± 0.058	**2.951 ± 0.084**

The best results are highlighted in bold.



(a) Accuracy & F1

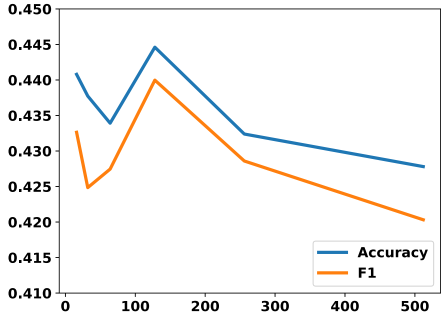


(b) MSE

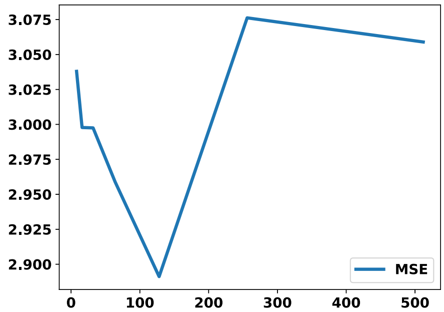
Fig. 5. Influence of sampled neighbor size, x-axis denotes the neighbor size and y-axis represents performance measure.

4.6 Case Study

Moreover, in order to analyze the interpretability of HM^2, we also give the attention visualization results of two unlisted company (MiaoQu and MayiJuniu software companies) by using HM^2. The visualization results are shown in Fig. 7, and it is notable that we only exhibit the relationships between input node and the neighbors, not the



(a) Accuracy & F1



(b) MSE

Fig. 6. Influence of embedding dimension, x-axis denotes the embedding dimension and y-axis represents performance measure.

topology structure. The first row displays MiaoQu company, and the second row illustrates MayiJuniu company.

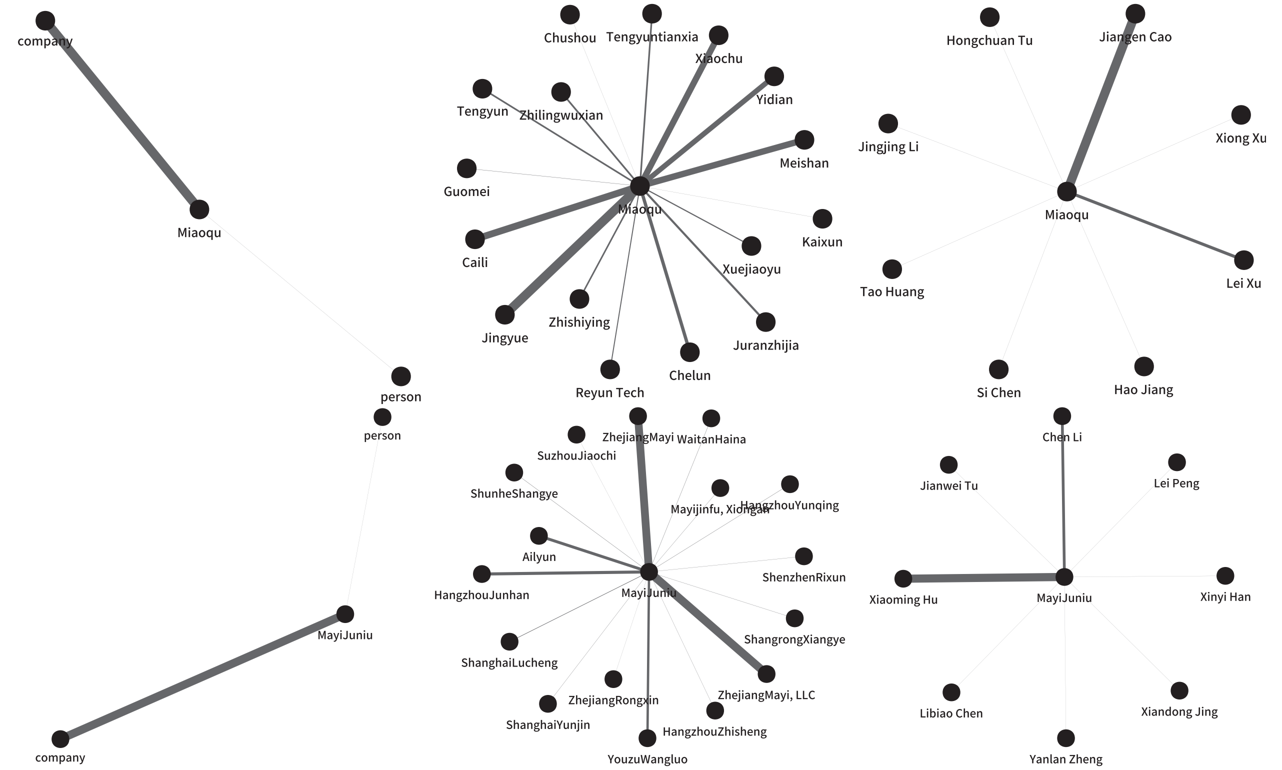
In the first row, Fig. 7a indicates that the relative valuation is strongly related to the company neighbors, i.e., $\alpha = 0.75$, which is reasonable, because unlisted company's value is generally strongly related to the company's industry information. Meanwhile, Figs. 7b and 7c reveal that the companies "XiaoChu", "Jingyue", "Meishan" (the weights are $0.18, 0.15, 0.13$) and member "JianGen Cao" (the weight is 0.74) have great impacts on the company, as these subsidiaries produced MiaoQu's main products, and JianGen Cao is the CEO of the company.

In the second row, Fig. 7a indicates that the relative valuation has relatively balanced correlations with the company and member neighbors, i.e., company attention weight $\alpha = 0.51$ and member attention weight $\alpha = 0.49$, which is reasonable, since the company has several influential members, "Xiaoming Hu" not only owns several influential companies, but also has connections with many core members of other listed company. Meanwhile, Figs. 7b and 7c reveal that the companies "ZhejiangMayi, LLC", "ZhejiangMayi" (the weights are $0.28, 0.27$) and member "Xiaoming Hu" (the weight is 0.75) have great impacts on the "MayiJuniu", as these companies are respectively investors and clients of the "MayiJuniu", and "Xiaoming Hu" is the CEO of the company.

5 Related Work

The related works include: 1) corporate valuation; 2) multi-modal aggregation; and 3) heterogeneous graph mining.

Corporate Valuation. Corporate valuation methods can be divided into two categories, i.e., relative valuation [32], [33], [34] and absolute valuation [4], [5]. Relative valuation always conducts a comparison with comparable companies (Trading Comps) or precedent transactions (Deal Comps).



(*a*) Overall (*b*) Company Neighbors (*c*) Member Neighbors

Fig. 7. Example of attention visualizations for two cases. (a) is attention visualizations of multi-modal aggregation module, (b) and (c) are attention visualizations of modal attribute encoding module.

Absolute valuation, which concentrates on the analysis of cash flow and the converting to current value, is a more complex refined forecast method. With the development of data mining technologies, there have been some attempts to use related techniques for valuation [35], [36]. However, these methods require full financial statements and stock information, which are difficult to obtain considering commercial privacy protection, especially for startups. Another effective method is to analyze the company's core resources and members, which are much easier to obtain from public information. But this method needs experienced experts.

Multi-Modal Aggregation. Multi-modal learning improves performance by leveraging heterogeneous multi-source data, in which modal sufficiency is one of the important principles. Traditional methods make full use of multi-modal data by directly aggregating multiple source information, i.e., early (i.e., feature-based) or late fusion (i.e., decision-based), for example, early fusion methods concatenated the multiple feature representations for final prediction. In contrast, late fusion methods utilize max/mean pooling to integrate multi-modal predictions. Theses approaches are based on the assumption that each modal can provide sufficient information for prediction. However, the information contained in various modalities is divergent, thus researchers turn to adopt weighted ensemble for acquiring a more reliable prediction. For example, [37] developed shot-variance and min-fusion schemes for both intra- and intermodal fusions; [38] utilized multiple kernel learning to integrate different modal information. Recently, with the development of deep learning and attention mechanism, many approaches attempted to self-learn the modal weights, for example, [39] incorporated feature-wise attention network to concatenate deep multi-modal embeddings for rumor detection; [40] combined self-attention to adaptively learn the weights for different modalities which is further used for prediction.

Heterogeneous Graph Mining. Graph learning [41] is one of the most popular data mining topics. Recently, with the advent of deep learning, graph neural networks [18], [22], [22], [41], which aggregate information from neighbors via neural networks, have been widely researched. Different from previous graph embedding models, which adopt linear methods, the key idea of graph neural networks is to aggregate feature information from node's neighbors via neural networks. For example, [18] proposed Graph-SAGE using neural networks, i.e., LSTM, to aggregate neighbors' feature information; [25] developed GAT to measure impacts of different neighbors via employing attention mechanism, and combine their impacts to obtain node embeddings. Most of these methods concentrate on homogeneous graph. However, as introduced in Section 2.2, company's core resources and members construct a heterogeneous graph with multi-modal attribute. To solve this problem, heterogeneous graphs mining has been proposed and applied widely, for example, [42] extracted topological features, and predicted citation relationship, [43] developed a co-attention deep network to leverage meta-path based context. Besides, [44], [45] designed heterogeneous networks to automatically preserve both attribute semantics and multi-type relations.

6 Conclusion

Considering the availability and deficiency of financial statements, corporate relate valuation, based on core resources, members, and competitors, plays an important role in entertainments services. Traditional CRV always relies on domain experts, which undoubtedly brings huge costs. Recent years, an increasing number of machine learning methods have been successfully applied in entertainments services. Notably, company's structure can be represented as a heterogeneous multi-modal graph, and the attributes on different types of nodes constitute multi-modal data. Therefore, we developed HM^2, an HGNN style method, which can aggregate node attributes via linkage-aware multi-head attention mechanism, rather than use multi-instance based methods without considering relationships among nodes. Meanwhile, HM^2 adopted additional triplet loss with embedding of competitors as the constraint to learn more discriminative features. Consequently, HM^2 can explore company intrinsic properties to improve CRV. Extensive experiments on real-world CRV data demonstrated the effectiveness of HM^2.

References

[1] C. Qin *et al.*, "DuerQuiz: A personalized question recommender system for intelligent job interview," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2165–2173.

[2] Y. Sun, F. Zhuang, H. Zhu, X. Song, Q. He, and H. Xiong, "The impact of person-organization fit on talent management: A structure-aware convolutional neural network approach," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1625–1633.

[3] M. Haldar *et al.*, "Applying deep learning to airbnb search," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1927–1935.

[4] M. C. Jensen, "Agency costs of free cash flow, corporate finance, and takeovers," *Amer. Econ. Rev.*, vol. 76, no. 2, pp. 323–329, 1986.

[5] J. Francis, "EVA [R] and value-based management: A practical guide to implementation.(book reviews)," *Accounting Rev.*, vol. 77, no. 1, pp. 228–229, 2002.

[6] T. A. Luehrman, *Investment Opportunities as Real Options: Getting Started on the Numbers*. Boston, MA, USA: Harvard Bus. Rev., 1998.

[7] K. L. Fisher, *The Only Three Questions That Count: Investing by Knowing What Others Don't*, vol. 22. Hoboken, NJ, USA: Wiley, 2007.

[8] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[9] T. Miloud, A. Aspelund, and M. Cabrol, "Startup valuation by venture capitalists: An empirical study," *Venture Capital*, vol. 14, no. 2/3, pp. 151–174, 2012.

[10] A. Kohn, "The determinants of startup valuation in the venture capital context: A systematic review and avenues for future research," *Manage. Rev. Quart.*, vol. 68, no. 1, pp. 3–36, 2018.

[11] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 990–998.

[12] R. He and J. J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proc. Int. Conf. World Wide Web*, 2016, pp. 507–517.

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

[14] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[15] F. Zhang *et al.*, "OAG: Toward linking large-scale heterogeneous entity graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2585–2595.

[16] J. Zhao *et al.*, "IntentGC: A scalable graph convolution framework fusing heterogeneous information for recommendation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2347–2357.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.

[19] Z. Liu, C. Chen, X. Yang, J. Zhou, X. Li, and L. Song, "Heterogeneous graph neural networks for malicious account detection," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 2077–2085.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[21] E. B. Khalil, H. Dai, Y. Zhang, B. Dilkina, and L. Song, "Learning combinatorial optimization algorithms over graphs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6348–6358.

[22] J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations*, 2013.

[24] B. Xiang, Z. Liu, J. Zhou, and X. Li, "Feature propagation on graph: A new perspective to graph representation learning," 2018, *arXiv:1804.06111*.

[25] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.

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

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

[28] Y. Dong, N. V. Chawla, and A. Swami, "metapath2vec: Scalable representation learning for heterogeneous networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 135–144.

[29] L. Liao, X. He, H. Zhang, and T.-S. Chua, "Attributed social network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 12, pp. 2257–2270, Dec. 2018.

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

[31] Y. Cen, X. Zou, J. Zhang, H. Yang, J. Zhou, and J. Tang, "Representation learning for attributed multiplex heterogeneous network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1358–1368.

[32] M. Ghaeli, "Price-to-earnings ratio: A state-of-art review," *Accounting*, vol. 3, no. 2, pp. 131–136, 2017.

[33] S. Agrawal, R. Monem, and M. Ariff, "Price to book ratio as a valuation model: An empirical investigation," *Finance India*, vol. 2, no. 10, pp. 333–344, 1996.

[34] J. P. O'Shaughnessy, *What Works on Wall Street: A Guide to the Best-Performing Investment Strategies of All Time*. New York, NY, USA: McGraw-Hill, 1998.

[35] E. Pagourtzi, K. Metaxiotis, K. Nikolopoulos, K. Giannelos, and V. Assimakopoulos, "Real estate valuation with artificial intelligence approaches," *Int. J. Intell. Syst. Technol. Appl.*, vol. 2, no. 1, pp. 50–57, 2007.

[36] G. Karami and S. BeikBoshrouyeh , "Corporate governance and equity valuation: The model by using artificial neural network," *J. Accounting Auditing Rev.*, vol. 18, no. 64, pp. 129–150, 2011.

[37] G. Evangelopoulos *et al.*, "Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention," *IEEE Trans. Multimedia*, vol. 15, no. 7, pp. 1553–1568, Nov. 2013.

[38] M. Gonen and E. Alpaydin, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, 2011.

[39] Z. Jin, J. Cao, H. Guo, Y. Zhang, and J. Luo, "Multimodal fusion with recurrent neural networks for rumor detection on microblogs," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 795–816.

[40] Y. Yang, K. Wang, D. Zhan, H. Xiong, and Y. Jiang, "Comprehensive semi-supervised multi-modal learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4092–4098.

[41] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 5, pp. 833–852, May 2019.

[42] Y. Sun, J. Han, C. C. Aggarwal, and N. V. Chawla, "When will it happen?: Relationship prediction in heterogeneous information networks," in *Proc. Int. Conf. World Wide Web*, 2012, pp. 663–672.

[43] 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. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1531–1540.

[44] Z. Liu, C. Huang, Y. Yu, B. Fan, and J. Dong, "Fast attributed multiplex heterogeneous network embedding," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 995–1004.

[45] B. Li, D. Pi, Y. Lin, I. A. Khan, and L. Cui, "Multi-source information fusion based heterogeneous network embedding," *Inf. Sci.*, vol. 534, pp. 53–71, 2020.

Yang Yang received the PhD degree in computer science from Nanjing University, Nanjing, China, in 2019. At the same year, he became a faculty member with the Nanjing University of Science and Technology, China. He is currently a professor with the School of Computer Science and Engineering. His research interests lie primarily in machine learning and data mining, including heterogeneous learning, model reuse, and incremental mining. He has published more than 10 papers in leading international journal/conferences. He serves as PC in leading conferences such as IJCAI, AAAI, ICML, NIPS, etc.

Jia-Qi Yang is currently working toward the MSc degree in the National Key Lab for Novel Software Technology, Department of Computer Science & Technology, Nanjing University, Nanjing, China. His research interests lie primarily in machine learning and data mining, including multi-modal learning.

Ran Bao is currently working toward the MSc degree in the National Key Lab for Novel Software Technology, Department of Computer Science & Technology, Nanjing University, Nanjing, China. His research interests lie primarily in machine learning and data mining.

De-Chuan Zhan received the PhD degree in computer science from Nanjing University, Nanjing, China, in 2010. At the same year, he became a faculty member with the Department of Computer Science and Technology, Nanjing University, China. He is currently a professor with the Department of Computer Science and Technology, Nanjing University. His research interests are mainly in machine learning, data mining, and mobile intelligence. He has published more than 20 papers in leading international journal/conferences. He serves as an editorial board member of the *IDA* and *International Journal of Applied Pattern Recognition*, and serves as SPC/PC in leading conferences such as IJCAI, AAAI, ICML, NIPS, etc.

Hengshu Zhu (Senior Member, IEEE) received the BE and PhD degrees in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2009 and 2014, respectively. He is currently a principal data scientist & architect with Baidu Inc. His general area of research is data mining and machine learning, with a focus on developing advanced data analysis techniques for innovative business applications. He has published prolifically in refereed journals and conference proceedings, including the *IEEE Transactions on Knowledge and Data Engineering* (TKDE), *IEEE Transactions on Mobile Computing* (TMC), *ACM Transactions on Information Systems* (ACM TOIS), *ACM Transactions on Knowledge Discovery from Data* (TKDD), ACM SIGKDD, ACM SIGIR, WWW, IJCAI, and AAAI. He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the KDD Cup-2019 Regular ML Track, and a founding co-chair of the first International Workshop on Organizational Behavior and Talent Analytics (OBTA) and the International Workshop on Talent and Management Computing (TMC), in conjunction with ACM SIGKDD. He was the recipient of the Distinguished Dissertation Award of CAS (2016), the Distinguished Dissertation Award of CAAI (2016), the Special Prize of President Scholarship for Postgraduate Students of CAS (2014), the Best Student Paper Award of KSEM-2011, WAIM-2013, CCDM-2014, and the Best Paper Nomination of ICDM-2014. He is the senior member the ACM and CCF.

Xiao-Ru Gao is currently working toward the PhD degree in marketing with Rutgers, the State University of New Jersey, New Brunswick, New Jersey. Her research interests lie primarily in data mining, social networking, especially influencer marketing, and brand image perception.

Hui Xiong (Fellow, IEEE) received the PhD degree from the University of Minnesota (UMN), Minneapolis, Minnesota. He is currently a full professor with the Rutgers, the State University of New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), ICDM Best Research Paper Award (2011), and the IEEE ICDM Outstanding Service Award (2017). He is a co-editor-in-chief of the *Encyclopedia of GIS*, an associate editor of the *IEEE Transactions on Big Data* (TBD), *ACM Transactions on Knowledge Discovery from Data* (TKDD), and *ACM Transactions on Management Information Systems* (TMIS). He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a program co-chair for the IEEE 2013 International Conference on Data Mining (ICDM), a general co-chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a program co-chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an ACM distinguished scientist.

Jian Yang (Member, IEEE) received the PhD degree in pattern recognition and intelligence systems from the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002. In 2003, he was a postdoctoral researcher with the University of Zaragoza, Zaragoza, Spain. From 2004 to 2006, he was a post-doctoral fellow with the Biometrics Centre, Hong Kong Polytechnic University, Hong Kong. From 2006 to 2007, he was a post-doctoral fellow with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA. He is currently a Chang-Jiang professor with the School of Computer Science and Engineering, NUST. He has authored more than 200 scientific papers in pattern recognition and computer vision. His papers have been cited more than 6,000 times in the Web of Science and 15,000 times in the Scholar Google. His current research interests include pattern recognition, computer vision, and machine learning. He is a fellow of IAPR. He is currently an associate editor of the *Pattern Recognition*, *Pattern Recognition Letters*, the *IEEE Transactions on Neural Networks and Learning Systems*, and *Neurocomputing*.