Multi-Relational Graph Convolution Network for Stock Movement Prediction

Zihao Zhou[1], Le Zhang[2], Rui Zha[1], Qiming Hao[1], Tong Xu[✉1], Di Wu[3], and Enhong Chen[1]
[1]School of Computer Science and Technology, University of Science and Technology of China
[2]Baidu Research, Baidu Inc
[3]Hainan Century Frontier Asset Management Co. Ltd
Email: {zihaozhou, zr990210, hqm97}@mail.ustc.edu.cn, zhangle0202@gmail.com,
{tongxu, cheneh}@ustc.edu.cn, dwu@centuryfrontier.com

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892482

***Abstract*—Stock movement prediction aims at predicting the future price trends of stocks, which plays an important role in quantitative investing. Existing approaches toward this direction mainly focus on modeling the historical sequential information, while the fine-grained relationships among stocks (e.g., belonging to the same industry or concept) were largely neglected. To tackle this limitation, in this paper, we propose a Multi-Relational Graph Convolution Network (MRGCN) framework for stock movement prediction, which incorporates the fine-grained multiple relationships into stock representation. Specifically, we first extract the temporal and static information for each stock from the historical series and corporation descriptions respectively. Then, we construct two pre-defined graphs based on domain knowledge and a self-adaptive graph to capture both explicit and implicit relationships among stocks. Along this line, the graph convolution network with attention mechanism is adopted on the multi-relational graph to generate the structural representation for each stock, and an embedding reconstruction module is further designed to refine the representation. Finally, we make predictions by integrating both temporal and structural embedding of stocks. Experiments on real-world China A-share market evince the superior performance of MRGCN compared to other baselines.**

***Index Terms*—Stock Movement Prediction, Graph Convolution Networks, Financial Data Mining.**

I. Introduction

The prediction of the stock movement has received growing interest as one of the most important tasks in quantitative investing. The target is to predict the trends (e.g., raise or fall) of stock price, which can help the decision-making in the complex fast-changing stock market for investors.

Although the efficient market hypothesis states that every asset is fairly priced and it's impossible for investors to purchase undervalued stocks or sell stocks at inflated prices [1], [2], yet market inefficiencies may exist due to information asymmetries. Evidence such as earning momentum [3] and price momentum [4] has been discovered, proving that relying on technical analysis of past stock prices can still obtain excess returns. However, the stock market is highly volatile and non-stationary [5], which is caused by a variety of factors such as national policies and emergencies. As a result, predicting the trend of stock prices from a single perspective is not enough.

Plenty of deep neural network models regard stock movement prediction as a time series modelling problem, relying on

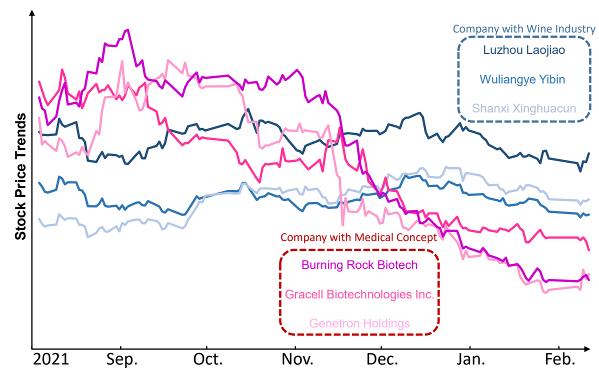

Fig. 1. An example of the stock movement trends, where some belong to the same industry (wine), and the others hold the same concept (medicine).

an effective process of indicator selection, these models indeed perform better than traditional machine learning algorithms [6]–[9]. However, considering that the stock market has only been for hundreds of years, the number of time series for medium and low frequency stock data is too limited to model every market trend of human civilization. Thus, the predictive capabilities of dominant paradigm are influenced due to the overfitting problem.

To this end, it's necessary to bring in external information such as news and social media, which can help us make precise prediction. However, this kind of information update quickly and is difficult to process. It's significant to find a static and convenient external information.

Often, there are rich associations in the complex interaction among companies. Research shows that the rich relationships among stocks contain valuable signals that facilitate stock trend prediction [10]. For instance, Figure 1 shows that stocks belong to the same industry or hold the same concepts exhibit synchronised trends, this enlighten us that relational information between stocks may be a good external information. In fact, researchers use relational data to model explicit associations between stocks recently. However, current work only considers a single relation between stocks, or considers multiple relationships that exist among stocks without taking the effects of these relationships dynamically, which make an inadequate utilization for different relationships.

Still, there are some main challenges in modelling these relationships between stocks. First, it's difficult to collect various relationships between stocks such as *subsidiary, competition* and *cooperation* due to confidentiality agreements and so on. Second, the relationship between companies varies dynamically, two companies may shift from a competitive to a cooperative partnership, which is difficult to track explicitly. Third, encoding a specific relationship into graph appropriately is challenging. Fourth, improper fusion of temporal representations and graph representations may lead to unsatisfactory results due to low signal-to-noise ratio in stock market data.

To address these challenges, we propose a novel framework, namely Multi-Relational Graph Convolution Network (MRGCN). For the first challenge, we design a self-adaptive graph that can automatically discover implicit relationships among stocks. For the second one, the relationship is given different weights to track the dynamic relationship changes throughout the attention mechanism, if the relationship varies, the weight varies. For the third one, we encode the explicit relationship with prior financial knowledge, making it explainable. For the fourth one, we find an effective way to combine temporal and structural embeddings, we separate temporal embeddings from graph convolution operation, which preserve the temporal dependencies effectively.

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

- We design a novel multi-graph module to model relations among stocks. In this phase, in addition to industry and concept graphs based on prior knowledge, a data-driven self-adaptive graph is developed to investigate the underlying relationships. Besides, the attention mechanism is adopted to track dynamic relationship and generate the structural embeddings.
- We explore a way to enhance the graph representation of stocks. Specifically, we get the initial graph representation from the description documents. By reconstructing it, the final graph representation can conserve the inherent characteristics of the stocks.
- We find a better way to combine temporal representation and graph representation, which can effectively improve the result. Experiments on China A-share market have verified our MRGCN framework's validity.

II. Related work

A. Technical Analysis

Conventional models for stock movement prediction mainly regard it as a sequence modelling problem, these models focus on technical analysis, which intend to predict the stock price trends through technical indicators, including volatility, moving average and momentum, etc., and determine investment strategies based on the trend. Traditional statistical methods use auto regressive models such as ARIMA [11], ARCH [12], GARCH [13]. However, such methods have poor fitting ability for highly nonlinear and unstable financial data. In recent years, machine learning methods such as Support Vector Machine (SVM) [14] and Random Forest (RF) [15] have made good performance in stock prediction. With the rapid development of deep learning, deep neural networks have showed their powerful modeling capabilities, which enables them to discover the hidden features from complex stock data. Recurrent neural network (RNNs), which inputs input features and hidden state features into the same structure at each time step, can capture the long-term dependence. However, Vanilla RNN has the vanishing gradient problem. So RNN-based models are more popular in time series modelling [16], [17]. For instance, Zhang *et al.* [18] proposed State Frequency Memory (SFM) to represent sequence information as a combination of different state-frequency to predict stock price changes. Qin *et al.* [19] proposed a dual-stage attention-based recurrent neural network (DA-RNN), which introduces the attention mechanism to select the corresponding features and the corresponding hidden layer states for time series prediction. However, these models assumed that each stock is isolated, without utilizing the information on collaborative interactions between stocks.

B. Market Relation Modeling

Recently, researchers have paid a great deal of attention to the correlations between stocks. In specific, graph-structured data has been utilized to exploit the associations among companies. For instance, Chen *et al.* [20] built a shareholding graph to incorporate information of neighborhood companies for predicting the stock price of the target company. Feng *et al.* [21] proposed a model called temporal graph convolutional (TGC), which combines industry and Wikipedia relations to modify stock sequence embeddings in a time-aware manner. Kim *et al.* [22] proposed a hierarchical attention network (HATS) which selectively aggregates information on different relation types for stock market prediction. Ye *et al.* [23] proposed Multi-GCN, which encodes multiple kinds of relationships into graphs and uses GCN to extract the shared information among stocks based on these pre-defined graphs. Ying *et al.* [24] proposed a time-aware graph relational attention network (TRAN) to capture time-aware correlation strength.

Although graph-based methods for stock movement prediction have made great progress, current work is mainly limited by financial domain knowledge, hard-to-get relational data, and dynamic relationships. In this paper, we focus on modeling the effects of explicit and implicit associations among stocks, with taking these differently and dynamically.

III. Preliminary

A. Problem Formulation

For all stocks, let $\mathcal{S} = \{s_1, s_2, \ldots, s_v\}$ denote v individual stocks, $\mathcal{D} = \{d_1, \ldots, d_v\}$ denote the description documents of corporations that issue the stocks. Each stock is associated with historical sequence data $\mathcal{X}_t^s = \left\{[x_{t-T+1}^s, \ldots, x_t^s], s \in \mathcal{S}\right\} \in \mathbb{R}^{T\times K}$, where x_t^s is a set of input features of stock s on the t-th trading day, T is the length of the sequence and K is the dimension of input features. We denote a graph as $\mathcal{G}=(\mathcal{S},\mathcal{E})$ where $\mathcal{E}$ is the set of edges that represent the relations of stocks.

Definition 1: Stock Relation Graph. Since there are various relations between the stocks, we construct a multi-relational graph to denote the stock relation graph precisely. It can be split into three subgraphs via pre-defined financial knowledge and data-driven learning. The multi-relational graph can be denoted as $\mathcal{G}=\{\mathcal{G}_{ind},\mathcal{G}_{cpt},\mathcal{G}_{adp}\}$, where $\mathcal{G}_{ind}=\{\mathcal{S},\mathcal{E}_{ind}\}$, $\mathcal{G}_{cpt}=\{\mathcal{S},\mathcal{E}_{cpt}\}$, $\mathcal{G}_{adp}=\{\mathcal{S},\mathcal{E}_{adp}\}$ denote different subgraphs on $\mathcal{G}$, the adjacency matrix derived from these subgraphs is denoted by $\mathcal{A}_{ind},\mathcal{A}_{cpt},\mathcal{A}_{adp}\in\mathbb{R}^{v\times v}$ respectively.

Definition 2: Stock Movement Prediction. We formulate stock movement prediction as a ternary classification problem, for it is more practical to distinguish between small-scale and large-scale ups and downs. We define the return ratio $r_s^t=\frac{p_s^t-p_s^{t-1}}{p_s^{t-1}}$ where p_s^t is the adjusted closing price[1] on day t, it reflects a stock's true value by taking any corporate actions such as stock splits and dividends under consideration. In our study, the price movement of stocks from $t-1$ to t is defined as follows:

$$y_t=\begin{cases}\text{up}, & r_t\geq\tau_{rising}\\ \text{neutral}, & \tau_{falling}<r_t<\tau_{rising}\\ \text{down}\ , & r_t\leq\tau_{falling}\end{cases}\tag{1}$$

Following [25], we set $\tau_{rising}=0.55\%$ and $\tau_{falling}=-0.50\%$ to balance the portion of positive and negative samples.

Given a stock relation graph $\mathcal{G}$, historical sequences $\mathcal{X}_t=\{\mathcal{X}_t^{s_1},\ldots,\mathcal{X}_t^{s_v}\}$ and the description documents $\mathcal{D}$, our problem is to learn a function f which is able to forecast its trend on the next day. The mapping relation is described as follows:

$$[\mathcal{X}_t,\mathcal{D},\mathcal{G}]\xrightarrow{f}\hat{Y}_{t+1},\tag{2}$$

where $\hat{Y}_{t+1}=\left\{\hat{y}_{t+1}^{s_1},\ldots,\hat{y}_{t+1}^{s_v}\right\}$ is the predicted price movement of stocks on day $t+1$.

TABLE I
MAIN MATHMATICAL NOTATIONS USED IN OUR PAPER

Symbol	Description
$\mathcal{S}$	Set of stocks
$\oplus$	The matrix concatenating operation
$\odot$	The Hadamard product
α	A hyperparameter that balance different loss terms
$\mathcal{G}$	Multi-relational graph where $\mathcal{G}=\{\mathcal{G}_{ind},\mathcal{G}_{cpt},\mathcal{G}_{adp}\}$
$\mathcal{G}_{ind},\mathcal{G}_{cpt}$	Relational graph built by pre-defined knowledge
$\mathcal{G}_{apt}$	Relational graph learned from the data
$\mathcal{X}_t$	Features of all the stocks on day t
Y_t	The ground truth of the movement of all the stocks on day t
$\mathbf{H}^{(0)}\in\mathbb{R}^{V\times L}$	Encoder embeddings for all the stocks
$\mathbf{H}^l_{\mathcal{R}_i}\in\mathbb{R}^{V\times L}$	The representation of the model's l-th layer on relation $\mathcal{R}_i$

IV. Methodology

In this section, we will introduce the technical details of the proposed MRGCN framework. As shown in Figure 2, MRGCN framework mainly contains four components, namely *temporal feature extraction*, *static feature extraction*, *multi-relational graph feature fusion* and *stock movement prediction*, respectively. Table I presents some important mathematical notations used throughout this paper.

A. Temporal Feature Extraction

Obviously, historical sequence data plays an important role in predicting the future trends of stocks. Inspired by the recent success of Long Short-Term Memory (LSTM) [26] on learning the temporal representation of sequential data, we apply LSTM model to capture temporal dependency of each stock from its historical sequence. Specifically, LSTM is a recurrent neural network designed for long-term dependency problems [27]. It aggregates historical information over time, which can overcome the problem of vanishing gradients and capture the long-term dependencies of time series better. Formally, given the historical sequence of stock s, $\mathcal{X}_t^s=\left\{[x_{t-T+1}^s,\ldots,x_t^s],s\in\mathcal{S}\right\}$, we individually input it into the LSTM networks, the process can be formulated as follows:

$$\begin{aligned}z_t^s&=\tanh\left(W_zx_t^s+Q_zh_{t-1}^s+b_z\right),\\ i_t^s&=\sigma\left(W_ix_t^s+Q_ih_{t-1}^s+b_i\right),\\ g_t^s&=\sigma\left(W_gx_t^s+Q_gh_{t-1}^s+b_g\right),\\ c_t^s&=g_t^s\odot c_{t-1}^s+i_t\odot z_t^s,\\ o_t^s&=\sigma\left(W_ox_t^s+Q_oh_{t-1}^s+b_o\right),\\ h_t^s&=o_t^s\odot\tanh\left(c_t^s\right),\end{aligned}\tag{3}$$

where $W_z,W_i,W_g,W_o\in\mathbb{R}^{U\times K}$ and $Q_z,Q_i,Q_g,Q_o\in\mathbb{R}^{U\times U}$ are weight matrices and U is the number of hidden units, $b_z,b_i,b_g,b_o\in\mathbb{R}^U$ are bias vectors, c^t and h^t denote cell state vector and hidden state vector, $tanh$ and σ denote tanh and sigmoid activation function, respectively.

However, the LSTM network may lose useful information about the header of time series when faced with a long-term sequence. If the model can't capture the key feature information in time, it will affect the performance of prediction. To solve this problem, we introduce a temporal attention mechanism to weigh the importance of the features in different days. Given the hidden representation $\bar{h}_t=[h_{t-T},\ldots,h_{t-1}]\in\mathbb{R}^{U\times T}$, an unified embedding $\widetilde{h_t}$ is aggregated by calculating the attention weight β_t, the process can be formulated as:

$$\widetilde{h_t}=\sum_k\beta_kh_k,\beta_k=\frac{\exp\left(h_k^TW\bar{h}_t\right)}{\sum_i\exp\left(h_i^TW\bar{h}_t\right)},\tag{4}$$

where W is the transformation matrix to be learned. Thus, we generate the final sequence features of all stocks on day t, which are denoted as $\widetilde{h_t^{s_1}},\widetilde{h_t^{s_2}},\ldots,\widetilde{h_t^{s_v}}\in\mathbb{R}^U$.

B. Static Feature Extraction

To represent the static feature of each stock, we utilize the corporation description document, which contains valuable information of the corporation, such as business details. However, the number of description documents is small and the necessary annotations are lacking, which may hinder the model training. Recently, with the popularity of transfer learning, BERT [28] has shown its unique advantages in several tasks

[1]https://www.investopedia.com/terms/a/adjusted_closing_price.asp

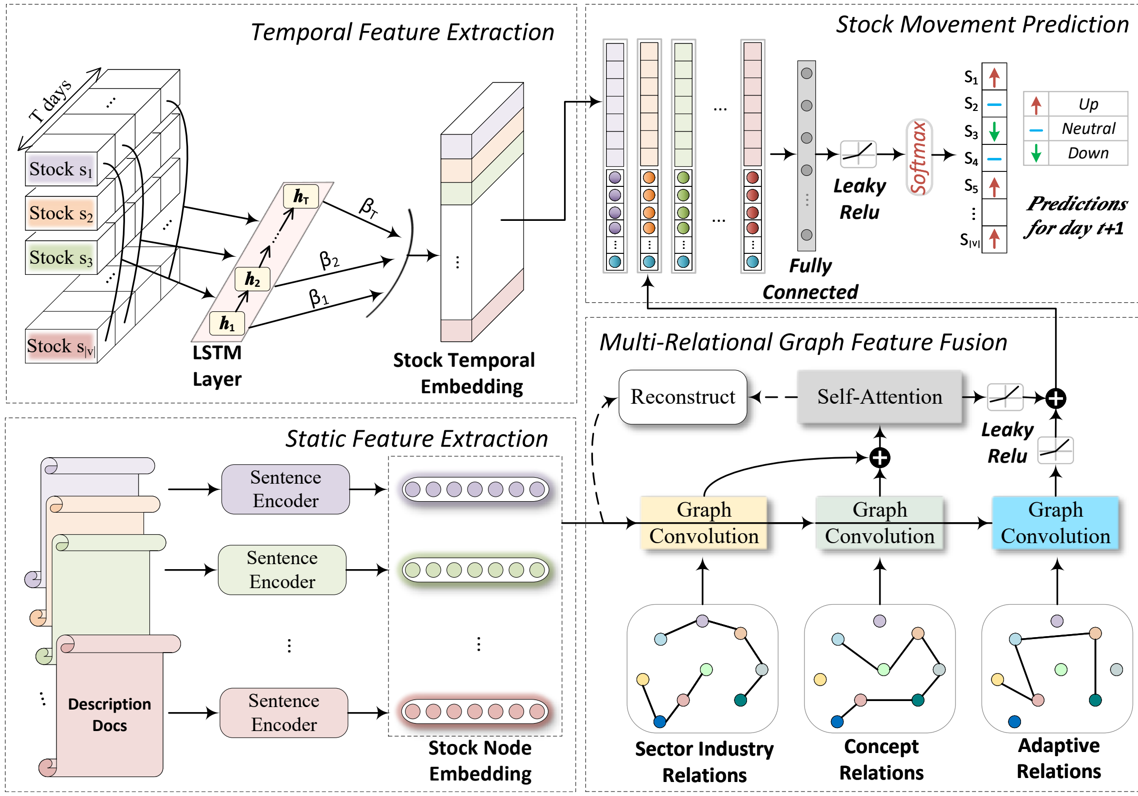

Fig. 2. The overall architecture of our MRGCN framework, which mainly contains four components, namely *temporal feature extraction*, *static feature extraction*, *multi-relational graph feature fusion* and *stock movement prediction*.

such as natural language inference. Inspired by Sentence-BERT [29] model, we extract the initial stack embeddings for each stock s, $\overline{\mathbf{H}^{(0)}} = \left\{\overline{\mathbf{H}_{s_1}^{(0)}}, \ldots, \overline{\mathbf{H}_{s_v}^{(0)}}\right\} \in \mathbb{R}^{\mathcal{V}\times\widetilde{L}}$, where $\widetilde{L}$ is the dimension of the output embeddings. In order to preserve the key features in the description document, we further reduce the embedding dimension by using Principal Component Analysis (PCA) [30]. Therefore, the description documents of all stocks $\mathcal{D}$ are encoded as $\mathbf{H}^{(0)} = \left\{\mathbf{H}_{s_1}^{(0)}, \ldots, \mathbf{H}_{s_v}^{(0)}\right\} \in \mathbb{R}^{\mathcal{V}\times L}$ to describe the attributes of the stocks.

C. Multi-Relational Graph Construction

As is shown in Figure 1, there are complex correlations among a set of stocks. To capture this cross effect in the stock market, we extract three types of relationships based on prior financial knowledge and data-driven automatic learning. Along this line, we attempt to introduce Graph Neural Networks (GNNs), which are prevalent in many real-world applications [31], [32]. Specifically, for each type of relationship $r \in \mathcal{R}$, graph $\mathcal{G}_r$ is built with its adjacency matrix $\mathcal{A}_r = (a_{ij})_{\mathcal{V}\times\mathcal{V}}$, element a_{ij} in $\mathcal{A}_r$ indicates the connection strength between stock i and stock j.

1) Industry Graph $\mathcal{G}_{ind}$: There is considerable evidence pointing to a lead-lag effect in the equity markets, in which some firms' stock prices show a delayed reaction to price innovations of other firms [33]. For example, returns of small firms are correlated with past returns of big firms, but not vice versa. Hou [10] confirmed that the lead-lag effect exists in an intra-industry predominantly, and it can be explained by industry information diffusion hypothesis. Since the lead-lag effect on inter-industry is rather weak [10], we only consider it in the same industry. To model the lead-lag effect, we use *register capital (C)* to measure the firm size, the influence from company i to company j is quantified as $a_{ij} = \frac{C_i}{C_j}$.

2) Concept Graph $\mathcal{G}_{cpt}$: The concept of the stock market refers to a class of stocks with common characteristics. For example, the concept of the Olympic Games refers to a class of companies that have business opportunities with the Olympic Games. In the stock market, the inherent meaning of the concept can be regarded as a market consensus. As is shown in Figure 1, the stocks which own same concepts tend to be correlative with each other. To model this effect, we collect all the concepts which a stock owns, and weight the edge by the number of common concepts. Specifically, if company i owns M_i concepts, company j owns M_j concepts and they share N_{ij} concepts, then the edge from i to j is weighted as $a_{ij} = \frac{\mathrm{N_{ij}}}{\mathrm{M_i}}$. In order to reduce the influence of the noise, we set a threshold $\mathcal{T}_{cpt}$, if $N_{ij} < \mathcal{T}_{cpt}$, the connection between company i and j would be discarded.

3) Self-adaptive Graph $\mathcal{G}_{adp}$: Indeed, the various relationships between stocks are difficult to collect. Inspired by Graph WaveNet [34], we design a self-adaptive matrix $\mathcal{A}_{adp}$ learned by the data to discover the latent relationships between stocks. We introduce two node embedding dictionaries $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{\mathcal{V}\times L_c}$, which are learnable parameters with randomly initialized, the self-adaptive matrix is formulated as:

$$\mathcal{A}_{adp} = \operatorname{SoftMax}\left(\operatorname{LeakyReLU}\left(\mathbf{E}_1\mathbf{E}_2^T\right)\right). \tag{5}$$

D. Multi-Relational Graph Feature Fusion

Afterwards, we adopt the graph convolution network on the constructed graphs to generate representation for nodes by aggregating the neighboring information. In order to capture stock shared information from various relations, follow [35], we design our graph convolution network with two layers. Both pre-defined relations and self-learned hidden relations are encoded as follows:

$$\begin{aligned}\mathbf{H}_\zeta^{(1)} &= \boldsymbol{\rho}\left[\widehat{\mathcal{A}}_\zeta \mathbf{H}^{(0)} \boldsymbol{W}_\zeta^{l_1}\right], \zeta \in \mathcal{R}_{[ind,cpt,adp]}, \\ \mathbf{H}_\zeta^{(2)} &= \boldsymbol{\rho}\left[\widehat{\mathcal{A}}_\zeta (\mathbf{H}^{(0)} \oplus \mathbf{H}_\zeta^{(1)}) \boldsymbol{W}_\zeta^{l_2}\right], \zeta \in \mathcal{R}_{[ind,cpt,adp]},\end{aligned} \tag{6}$$

where $\widehat{\mathcal{A}} \in \mathbb{R}^{\mathcal{V}\times\mathcal{V}}$ denote the normalized adjacency matrix, $W_\zeta^{l_1} \in \mathbb{R}^{L\times L}$ and $W_\zeta^{l_2} \in \mathbb{R}^{2L\times L}$ is the subgraph parameter matrix of relation type $\mathcal{R}_\zeta$, $\rho(\cdot)$ denotes the activation function.

To jointly aggregate information from different relations [36], we first concatenate encoded prior knowledge relation representations on $\mathcal{G}_{ind}$ and $\mathcal{G}_{cpt}$ as follows:

$$\mathbf{H}_{pre}^{(2)} = \mathbf{H}_{\mathcal{R}_{ind}}^{(2)} \oplus \mathbf{H}_{\mathcal{R}_{cpt}}^{(2)}. \tag{7}$$

Then, to coordinate the importance of the two pre-defined relations, we introduce the self-attention mechanism to update the representation of nodes. Specifically, we build the query matrix Q, key matrix K, and value matrix V, the process can be expressed as follows:

$$\begin{aligned}Q &= \mathbf{H}_{pre}^{(2)} \cdot W^Q, \\ K &= \mathbf{H}_{pre}^{(2)} \cdot W^K, \\ V &= \mathbf{H}_{pre}^{(2)} \cdot W^V,\end{aligned} \tag{8}$$

where $W^Q \in \mathbb{R}^{d_{\text{model}}\times d_k}$, $W^K \in \mathbb{R}^{d_{\text{model}}\times d_k}$, $W^V \in \mathbb{R}^{d_{\text{model}}\times d_v}$ are parameter matrix to be learned.

Afterwards, we calculate the vector representation by way of scaled dot-product attention as follows:

$$\mathbf{H}_{attn} = \operatorname{SoftMax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V. \tag{9}$$

Finally, the overall stock node representations are generated by combining the output results from the self-attention mechanism and encoded representations from self-adaptively learned relationships, which can be described as follows:

$$\mathbf{H}_{fu} = \operatorname{LeakyReLU}\left[\mathbf{H}_{attn} \oplus \mathbf{H}_{\mathcal{R}_{adp}}^{(2)}\right]. \tag{10}$$

E. Graph Embedding Reconstruction

As the low signal-to-noise ratio is ubiquitous in finical data, it may largely limit the prediction performance. To solve this problem, we design a graph embedding reconstruction module to make sure the generated representation of each stock can preserve the initial stock embedding. Specifically, the reconstructed loss can be represented as follows:

$$\mathcal{L}_r = \frac{\left\|\left(\mathbf{H}_{attn} - \mathbf{H}^{(0)}\right)\right\|_F^2}{|\mathcal{V}|}. \tag{11}$$

In our reconstruction module, the focus of our work is to reconstruct the initial representation of stocks in the decoder part. As can be seen, once the module has the ability to restore the initial representations, the inherent properties of stocks can be considered valid preservation.

F. Prediction Layer

At last, we integrate the temporal representation learned from historical sequence data and the structural representation learned from multi-relational graph to make prediction. Specifically, we concatenate these two representations, and feed them into a fully connected layer to predict the movement of each stock, the process can be formulated as follows:

$$\widehat{y}_t^s = \sigma\left(W_2' \operatorname{LeakyReLU}\left(W_1'\left(\widetilde{h_t^s} \oplus \mathbf{H}_{fu}^s\right) + b_1'\right) + b_2'\right), \tag{12}$$

where σ denotes the sigmoid function, W_1' and W_2' are are weight matrices, b_1' and b_2' are bias vectors.

The overall objective is defined by combining the cross entropy loss and graph embedding reconstruction constraint loss as follows:

$$\begin{aligned}\mathcal{L}_{cse} &= -\sum_{i=1}^{|\mathcal{V}|}\sum_{j=1}^{3} y_{ij} \ln\left(\widehat{y}_{ij}\right), \\ \mathcal{L} &= \mathcal{L}_{cse} + \alpha\mathcal{L}_r,\end{aligned} \tag{13}$$

where y_{ij} is the ground truth of the stock movement, $\widehat{y}_{ij}$ is the estimated probability of different stock movement. $|\mathcal{V}|$ is the total number of stocks, α is a hyper-parameter to balance the cross entropy loss and the graph embedding reconstruction constraint loss.

V. Experimental Setup

A. Data Collection

To demonstrate the effectiveness of our model, we collect the Chinese stock data including stock historical sequence, description document, industry and concept information from *Wind-Financial Terminal*[2].The details are as follows:

Stock Sets. We collected the stocks from China A-share market with a historical data range from 03/01/2016 to 12/01/2020. Almost 4100 stocks were gathered, covering the majority of Chinese stocks. Following [21], we performed a filtering on the stocks by removing the stocks which have been traded on less than 98% of all trading days to avoid the effects of abnormal temporal patterns. As a consequence, we got 1602 stocks for our experiment.

Stock Historical Sequence. We extracted four price attributes per day, including the adjusted *open, high, low price*, and *close price*. To reduce the impact of short-term volatility of the price, we calculated four more trading indicators, namely, 5, 10, 20, and 30 days moving averages. In addition, the entire sequence data was strictly chronologically divided into three periods for training, validation and evaluation, details of the division are shown in Table II.

Stock Description Document. We collected the description document of the company that issued each stock. It includes

[2]https://www.wind.com.cn/

TABLE II
STATISTICS OF THE DATASETS

Stock Sets	# Stocks # Period	1,602 2016.3-2020.12
Stock Historical Sequence	# Training Days # Validation Days # Evaluation Days	791 114 228
Stock Description Document	# Descriptions # Words	1,602 512,912
Stock Relation Graph	# Nodes # $\mathcal{G}_{ind}$ Edges # $\mathcal{G}_{cpt}$ Edges	1,602 87,462 92,090

the basic company information, such as its main business, values and development goals.

Stock Relation Graph. We collected two types of critical pre-defined relational data: the industry which the stock belongs to and the concept that the stock holds. Finally, we got the 75 kinds of industries and 2085 concepts, which were used to construct the multi-relational graph. Table II shows the summary statistics for the stock relation graph.

B. Training Setup

All experiments were performed on a GeForce RTX 2080 GPU. Specially, the length of sequence is set to $T = 10$, and we trained our model using Adam optimizer with an initial learning rate of 0.01. In addition, the embedding dimension of description documents is 64, the hidden size of LSTM is 128, and the embedding size in GCN is 64. We set the mini-batch size as 64 and α as 0.1 respectively. Besides, we also conducted a standard student's t-test for the pair of MRGCN and each baseline to validate the improvement significance.

C. Evaluation Metrics

Since we regard stock movement prediction as a ternary classification problem, we adopt the classification macro-averaged accuracy, precision, recall, and F1-score as evaluation metrics. We calculate precision, recall and F1-score for each class i as follows:

$$\begin{aligned} \text{Precision}_i &= \frac{TP_i}{TP_i + FP_i}, \\ \text{Recall}_i &= \frac{TP_i}{TP_i + FN_i}, \\ \text{F1-score}_i &= \frac{2 \times Recall_i \times Precision_i}{Recall_i + Precision_i}, \end{aligned} \tag{14}$$

where TP_i is the true positive of class i, TN_i is the true negative of class i, FP_i is the false positive of class i, and FN_i is the false negative of class i. Then, the macro-averaged metrics can be calculated as follows:

$$\begin{aligned} \text{Precision}_{macro} &= \frac{\sum_{i=1}^{3} \text{Precision}_i}{3}, \\ \text{Recall}_{macro} &= \frac{\sum_{i=1}^{3} \text{Recall}_i}{3}, \\ \text{F1-score}_{macro} &= \frac{\sum_{i=1}^{3} \text{F1-score}_i}{3}. \end{aligned} \tag{15}$$

D. Baselines

We compare the proposed MRGCN with the following baselines, which make up a broad set of traditional classification models, deep sequence models and stock relationship-based models as follows:

- **LR:** Logistic regression (LR) models the probabilities for a discrete outcome given an input variable.
- **SVM [14]:** Support Vector Machine (SVM) tries to get the best effect of classifying by maximizing data margin as a traditional machine learning method.
- **RF [15]:** Random Forest (RF) adopts the strategy of integrating a number of decision trees to classify as an ensemble learning method.
- **LSTM [37]:** Long Short-Term Memory (LSTM) constructs a recurrent neural network framework with a classifier to learn the historical information.
- **DA-RNN [19]:** This method utilizes an attention mechanism to select the corresponding features and the corresponding hidden layer states respectively for time series prediction.
- **GCN [35]:** This method encodes stock historical sequence with a LSTM network at first and then feeds the last hidden state of LSTM into GCN to explore interactions between stocks.
- **TGC [21]:** This method devises a new component called temporal graph convolution to encode the temporal information into the relation-strength in a time-sensitive way.
- **MRGCN-Ind:** A variant of our model where only the industry relation is utilized.
- **MRGCN-Cpt:** A variant of our model where only the concept relation is utilized.
- **MRGCN-Adp:** A variant of our model where only the adaptive relation is utilized.
- **MRGCN-NoRe:** A variant of our model w/o the module of graph embedding reconstruction.

VI. RESULTS AND ANALYSIS

A. Performance Comparison with Baselines

The evaluation results of different methods are reported in Table III, where the p-value is less than 0.01 in all cases to show that improvements are statistically significant.

On this basis, we can get the following observations: 1) Relationship-based models, such as *TGC*, outperform traditional shallow architectures and deep sequence models significantly. The result clearly demonstrates the benefit of introducing the relational information in stock movement prediction. 2) The proposed *MRGCN* outperforms all the compared methods including *TGC*, which demonstrates the superiority of our model on capturing multiple relations among stocks. 3) Compared with *GCN* and *TGC*, *MRGCN-Ind* achieves better results, which suggests that taking the structural and temporal information into account simultaneously can lead to the best performance. Meanwhile, *DA-RNN* outperforms *GCN*, indicating that feeding temporal embeddings to graph is ineffective,

TABLE III
THE EXPERIMENTAL RESULTS OF DIFFERENT METHODS ON STOCK MOVEMENT PREDICTION.

Input Feature	Models	Accuracy	Precision	Recall	F1
Historical Records	LR	37.58%	36.42%	35.69%	32.52%
	SVM	39.53%	36.75%	36.02%	33.05%
	RF	39.86%	37.21%	36.47%	33.25%
	LSTM	40.50%	38.47%	37.16%	34.37%
	DA-RNN	41.68%	39.57%	37.84%	36.01%
Historical Records & Corporation Relationships	GCN	40.98%	39.23%	37.42%	35.57%
	TGC	41.25%	39.76%	37.05%	36.32%
	MRGCN-Ind	41.74%	39.85%	37.83%	37.33%
	MRGCN-Cpt	41.61%	39.79%	37.45%	37.56%
	MRGCN-Adp	41.17%	39.35%	37.40%	36.98%
	MRGCN-NoRe	41.98%	39.83%	37.91%	37.80%
	MRGCN	**42.37%**	**40.25%**	**38.52%**	**38.36%**

which may be caused by the high level of noise in China A-share market. 4) *MRGCN-Ind* and *MRGCN-Cpt* show similar results, while *MRGCN-Adp* performs worse relatively. This result demonstrates that the single pre-defined graph is better than the adaptive graph in the stock market. However, the integrated model *MRGCN* has the best performance, indicating that prior knowledge graph and self-adaptive graph can be complementary. 5) Compared with *MRGCN-NoRe*, we can see the effectiveness of the graph embedding reconstruction module, indicating that we can reduce the interference of the noise by reconstructing graph embeddings.

B. Visualization and Case Study

To explore the association between the pre-defined graph and the self-adaptive graph, we visualized the normalized adjacency matrix of three graphs in Figure 3, from which we have some interesting findings.

Given a target stock, the self-adaptive graph can learn the weights to indicate the stocks that are closely associated with it. Here, we presented a case study for the stock *Luxshare*. We picked out the top-5 weight stocks, as is displayed in Figure 4, we found that they are related to apple products mostly. That is to say, they own the concept of "apple", which our relation data of concepts doesn't exist. This demonstrates that the self-adaptive graph can capture implicit relations to some extent.

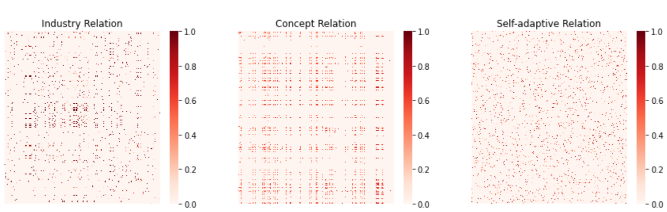

Fig. 3. The visualization of adjacency matrices

C. Parameter Analysis: Probing Sensitivity

We changed several hyperparameters to explore the sensitivity of MRGCN, including the size of historical length T, the size of α in the loss function.

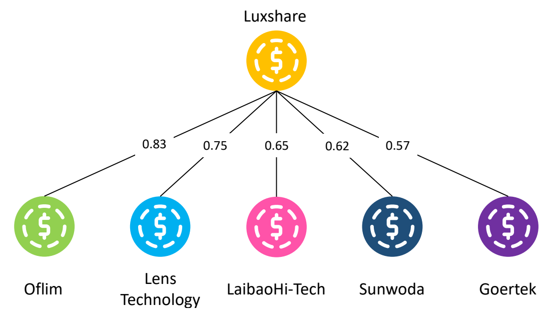

Fig. 4. The implicit relation digged from self-adaptive graph

1) Impact of the size of Historical Length T: We analyzed MRGCN's prediction performance with varying the historical sequence length T in Figure 5 and observed that our model performs better over the long sequence.

2) Impact of the size of α in Loss Function: To analyze the influence of reconstructing graph embeddings, we conducted experiments on different size of α, specifically $[0.01, 0.1, 0.5, 1]$ as is depicted in Figure 5 and found our model performs well with $\alpha = 0.1$.

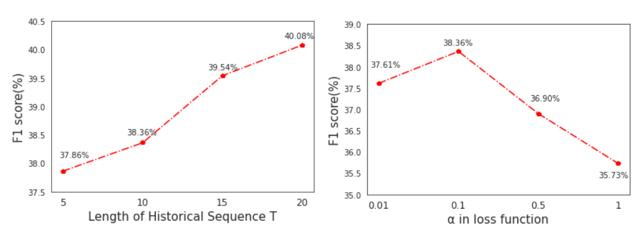

Fig. 5. Sensitivity to parameter T and α

VII. CONCLUSION

In this paper, we proposed a multi-relational graph convolution network framework for stock movement prediction, which represented each stock node comprehensively by leveraging the fine-grained multiple relationships among stocks. For temporal view, we extracted temporal embedding by temporal attention mechanism to learn time-varying dependencies. For spatial view, we constructed two pre-defined graphs based on domain knowledge and a self-adaptive graph to learn the explicit and implicit relationships among stocks. Meanwhile, the spatial attention mechanism was adopted to track dynamic relationships and generate structural embeddings. We further conserved the inherent properties of stocks by refining the representations through a embedding reconstruction module. Experiments on China A-share market validated the effectiveness of our model. In the future, we will explore dynamic graphs of stock relations to learn dynamic spatial dependencies.

ACKNOWLEDGMENT

This research was partially supported by grants from the National Natural Science Foundation of China (Grant No.62072423).

REFERENCES

[1] E. F. Fama, "The behavior of stock-market prices," *The journal of Business*, vol. 38, no. 1, pp. 34–105, 1965.

[2] ——, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[3] R. Ball and P. Brown, "An empirical evaluation of accounting income numbers," *Journal of accounting research*, pp. 159–178, 1968.

[4] N. Jegadeesh and S. Titman, "Returns to buying winners and selling losers: Implications for stock market efficiency," *The Journal of finance*, vol. 48, no. 1, pp. 65–91, 1993.

[5] K. Adam, A. Marcet, and J. P. Nicolini, "Stock market volatility and learning," *The Journal of Finance*, vol. 71, no. 1, pp. 33–82, 2016.

[6] W. Jiang, "Applications of deep learning in stock market prediction: recent progress," *Expert Systems with Applications*, vol. 184, p. 115537, 2021.

[7] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua, "Enhancing stock movement prediction with adversarial training," *arXiv preprint arXiv:1810.09936*, 2018.

[8] H. Liu and B. Song, "Stock trends forecasting by multi-layer stochastic ann bagging," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2017, pp. 322–329.

[9] C. Chen, L. Zhao, J. Bian, C. Xing, and T.-Y. Liu, "Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2376–2384.

[10] K. Hou, "Industry information diffusion and the lead-lag effect in stock returns," *The Review of Financial Studies*, vol. 20, no. 4, pp. 1113–1138, 2007.

[11] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, 2014, pp. 106–112.

[12] F. Fornari and A. Mele, "Sign-and volatility-switching arch models: theory and applications to international stock markets," *Journal of Applied Econometrics*, vol. 12, no. 1, pp. 49–65, 1997.

[13] P. H. Franses and D. Van Dijk, "Forecasting stock market volatility using (non-linear) garch models," *Journal of Forecasting*, vol. 15, no. 3, pp. 229–235, 1996.

[14] R. K. Nayak, D. Mishra, and A. K. Rath, "A naïve svm-knn based stock market trend reversal analysis for indian benchmark indices," *Applied Soft Computing*, vol. 35, pp. 670–680, 2015.

[15] L. Khaidem, S. Saha, and S. R. Dey, "Predicting the direction of stock market prices using random forest," *arXiv preprint arXiv:1605.00003*, 2016.

[16] S.-Y. Shih, F.-K. Sun, and H.-y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Machine Learning*, vol. 108, no. 8, pp. 1421–1441, 2019.

[17] J. Wang, T. Sun, B. Liu, Y. Cao, and H. Zhu, "Clvsa: A convolutional lstm based variational sequence-to-sequence model with attention for predicting trends of financial markets," *arXiv preprint arXiv:2104.04041*, 2021.

[18] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 2141–2149.

[19] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[20] Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1655–1658.

[21] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–30, 2019.

[22] R. Kim, C. H. So, M. Jeong, S. Lee, J. Kim, and J. Kang, "Hats: A hierarchical graph attention network for stock movement prediction," *arXiv preprint arXiv:1908.07999*, 2019.

[23] J. Ye, J. Zhao, K. Ye, and C. Xu, "Multi-graph convolutional network for relationship-driven stock movement prediction," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 6702–6709.

[24] X. Ying, C. Xu, J. Gao, J. Wang, and Z. Li, "Time-aware graph relational attention network for stock recommendation," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2281–2284.

[25] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, "Hierarchical multi-scale gaussian transformer for stock movement prediction." in *IJCAI*, 2020, pp. 4640–4646.

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

[27] L. Chen, Z. Li, W. He, G. Cheng, T. Xu, N. J. Yuan, and E. Chen, "Entity summarization via exploiting description complementarity and salience," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[29] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[30] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[31] L. Zhang, D. Zhou, H. Zhu, T. Xu, R. Zha, E. Chen, and H. Xiong, "Attentive heterogeneous graph embedding for job mobility prediction," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2192–2201.

[32] L. Zhang, T. Xu, H. Zhu, C. Qin, Q. Meng, H. Xiong, and E. Chen, "Large-scale talent flow embedding for company competitive analysis," in *Proceedings of The Web Conference 2020*, 2020, pp. 2354–2364.

[33] A. W. Lo and A. C. MacKinlay, "When are contrarian profits due to stock market overreaction?" *The review of financial studies*, vol. 3, no. 2, pp. 175–205, 1990.

[34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

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

[36] L. Chen, Z. Li, Y. Wang, T. Xu, Z. Wang, and E. Chen, "Mmea: entity alignment for multi-modal knowledge graph," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2020, pp. 134–147.

[37] D. M. Nelson, A. C. Pereira, and R. A. De Oliveira, "Stock market's price movement prediction with lstm neural networks," in *2017 International joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1419–1426.