

# MDGNN: Multi-relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction

Hao Qian<sup>1</sup>, Hongting Zhou<sup>1</sup>, Qian Zhao<sup>1</sup>, Hao Chen<sup>1</sup>, Hongxiang Yao<sup>2</sup>,  
Jingwei Wang<sup>1</sup>, Ziqi Liu<sup>1</sup>, Fei Yu<sup>1</sup>, Zhiqiang Zhang<sup>1</sup>, Jun Zhou<sup>1\*</sup>

<sup>1</sup>Ant Group, Hangzhou, China

<sup>2</sup>Alibaba Group, Hangzhou, China

{qianhao.qh, zhouhongting.zht, zq317110, chuhu.ch, wangjingwei.wjw, ziqiliu, lingyao.zzq, jun.zhoujun}@antgroup.com  
henry.yhx@alibaba-inc.com

## Abstract

The stock market is a crucial component of the financial system, but predicting the movement of stock prices is challenging due to the dynamic and intricate relations arising from various aspects such as economic indicators, financial reports, global news, and investor sentiment. Traditional sequential methods and graph-based models have been applied in stock movement prediction, but they have limitations in capturing the multifaceted and temporal influences in stock price movements. To address these challenges, the Multi-relational Dynamic Graph Neural Network (MDGNN) framework is proposed, which utilizes a discrete dynamic graph to comprehensively capture multifaceted relations among stocks and their evolution over time. The representation generated from the graph offers a complete perspective on the interrelationships among stocks and associated entities. Additionally, the power of the Transformer structure is leveraged to encode the temporal evolution of multiplex relations, providing a dynamic and effective approach to predicting stock investment. Further, our proposed MDGNN framework achieves the best performance in public datasets compared with state-of-the-art (SOTA) stock investment methods.

## Introduction

The stock market is a crucial component of the financial system, offering investors a marketplace to trade shares of a wide range of assets. Nevertheless, predicting the movement of stock prices is challenging due to the dynamic and intricate relations arising from various aspects. The active trading behaviors of investors, such as buying and selling, drive the fluctuations in stock prices. Additionally, the stock market is influenced by several factors, including economic indicators, financial reports, global news, political events, investor sentiments, and many others. Hence, it's indispensable to integrate comprehensive and multifaceted relations to capture the dynamics of the stock markets accurately.

Two lines of research have been applied to stock movement prediction. Traditional sequential methods (Hochreiter and Schmidhuber 1997; Chung et al. 2014; Feng et al. 2019a; Lin et al. 2021; Zhang, Aggarwal, and Qi 2017) propose to capture the temporal patterns of stock movement

by optimizing the temporal dependency encoder, which employs sequential extraction techniques (Jain and Medsker 1999; Devlin et al. 2019). Nevertheless, the majority of these methods still assume that stocks are independent of each other and overlook the influence of complex relations. In addition, graph-based models (Xu et al. 2021a; Chen and Robert 2021a; Wang et al. 2021a; Sawhney et al. 2021; Wang et al. 2022) incorporate heterogeneous information explicitly from data or implicitly mine it from textual data to capture the interdependence of stocks by designing various graph representation methods. However, these approaches could still be dissatisfactory due to the following two issues.

(1) **Multifacetedness.** The stock price movement is influenced not only by a single factor but also by multiple relations among stocks, industries, investment banks, etc. For example, changes in the stock prices in a particular industry can be caused by a variety of factors, such as high product demand, new government policies, the rise of raw material costs, and negative earnings reports from large companies. Similarly, investment banks can influence stock prices in numerous ways, including conducting research on companies, releasing positive or negative reports, and trading shares. Therefore, accurately predicting the movement of stock prices requires consideration of the multifaceted relations among stocks. Previous graph-based methods for stock investment prediction have only utilized single relations between stocks, ignoring the potential of incorporating other complex relations as auxiliary information.

(2) **Temporal.** The movement of stock prices and the multifaceted relations among stocks are not static but exhibit temporal evolution. Stock prices can change rapidly due to external factors such as economic conditions, political events, and regulatory changes, while internal factors such as company earnings and industry performance can also influence the movement of stock prices over time. Relationships among stocks change over time due to factors such as investment banks trading stocks, common shareholders co-holding stocks, and companies releasing products into new industries. Hence, accurately predicting the movement of stock prices and anticipating the impacts of these changes requires a dynamic approach that considers the historical trends and evolving relationships among stocks.

To address the aforementioned issues, we introduce a novel framework to underline the multifacetedness and tem-

\*Corresponding author

poral influences in stock investment prediction and propose a **Multi-relational Dynamic Graph Neural Network (MDGNN)**. Overall, we utilize the discrete dynamic graph framework to tackle the stock investment prediction. Specifically, to comprehensively capture the multifacetedness nature of stocks, we construct each graph snapshot with daily stock information and relationship data, which is then analyzed with a multi-relational graph embedding layer. The generated representation from the multi-relational graph offers a thorough and complete perspective on the interrelationships among stocks and associated entities. Additionally, we leverage the power of the Transformer structure to encode the temporal evolution of multiplex relations, providing a dynamic and effective approach to predicting stock investment. Our contributions are summarized as follows:

- We discuss the multifacetedness and temporal in the context of stock investment prediction tasks. We also provide insights on modeling complex stock relations based on empirical evidence.
- We propose to capture the multifaceted and temporal evolution nature of stocks with a multi-relational dynamic graph and generate a comprehensive representation of the stock market.
- We perform extensive experiments on public datasets to verify the superiority of our proposed framework. With detailed analysis, we demonstrate the effectiveness of the multi-relational dynamic graph in tackling the stock investment prediction task.

## Related Work

**Stock Trend Prediction.** In quantitative trading, the ability to anticipate stock trends is crucial. To accomplish this task, a multi-factor model is commonly employed, as detailed in Nagel’s recent work (Nagel 2021). This model considers several influential factors from an econometrics standpoint, including trading volumes and prices, as well as company-specific fundamental data like earnings per share and debt ratio. The obtained prediction results can then be utilized for long/short trading in the real market through either rule- or learning-based methods.

When utilizing learning-based methods, it’s a common practice to start with linear regression (Gu, Kelly, and Xiu 2020). Moreover, (Roy et al. 2015) utilized ordinary least squares equipped with regularization, such as ridges and lasso, to overcome the over-fitting issues. However, linear models have limitations in capturing complicated patterns in stock price trends. To overcome this limitation, attempts have been made to incorporate more complex learning techniques. XGBoost (Han, Kim, and Enke 2023) based method is developed and evaluated through an empirical analysis of companies listed on the NASDAQ. Neural-network-based LSTM (Nelson, Pereira, and De Oliveira 2017) is utilized in predicting future trends of stock prices and shows potential to tackle the challenge of an immensely complex, chaotic, and dynamic environment for the stock market.

**Dynamic Graph Neural Networks.** The above-mentioned methods for predicting stock trends primarily concentrate on individual stocks and disregard the interdependence and

resulting interactions between various stocks. For instance, stocks that belong to the same supply chain are interrelated due to profit transmission. Other types of relations comprise being held by identical institutions, mentioned in the same news article, and so forth.

In recent years, GNN has gained great success owing to its powerful capability of representing complex relations. Traditional GNN methods (e.g., GCN (Kipf and Welling 2017), GAT (Veličković et al. 2018)) are mostly based on static graphs where nodes and edges don’t change over time. However, many real-world relations (e.g., financial transactions, social relations) are continuously evolving, in which dynamic graphs are indispensable to capture the advancing relations. Several approaches have been proposed to represent dynamic graphs.

Recently, some studies have started utilizing relationship information and graph learning to predict stock trends. One such approach is Relational Stock Ranking (RSR) (Feng et al. 2019b), which incorporates sector and supply chain relation information into its temporal graph convolution. This allows RSR to handle the impact between different stocks by encoding stock relations in a time-sensitive way. Another approach is Multi-Graph Recurrent Network (MGRN) (Chen and Robert 2021b), which utilizes even more relationships such as historical price. This is calculated by the correlation coefficient of two stocks’ daily return time series. The multi-graph embedding combined with text embedding extracted from the news is then fed into an LSTM network to predict the stock trend. Hierarchical Adaptive Temporal-Relational Network (HATR) (Wang et al. 2021b) takes this further by introducing topicality associations in graph modeling. Additionally, Concept-oriented shared information for stock trend forecasting (HIST) (Xu et al. 2021b) proposed to mine hidden relations by designing a hidden concept module. This approach successfully mined information beyond that carried by predefined concepts. Evolvegcnn (Pareja et al. 2020) exploits the combination of graph convolution and RNN (Jain and Medsker 1999) to capture both the topological structures and temporal relations.

## Preliminary

**Definition 1. Multi-relational Graph.** A multi-relational graph is defined as a tuple  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ , where  $\mathcal{V}$  is the set of nodes,  $\mathcal{R}$  is a set of relations, and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \times \mathcal{R}$  is a set of directed edges. Each edge is denoted as a triplet  $(u, v, r)$ , indicating a relation  $r \in \mathcal{R}$  between nodes  $u$  and  $v$ .

**Definition 2. Problem Formulation.** Given that the relationships between stocks are multifaceted and changing on a daily basis, we propose a Dynamic Graph Neural Network (DGNN) to capture and represent them. Let  $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_T\}$  represent DGNN, where  $\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathcal{R}_t)$  is a multi-relational graph snapshot at trading day  $t$  and  $T$  is the total number of snapshots. For a stock node  $v_{it} \in \mathcal{V}_t$ , the closing price at a trading day  $t$  is denoted as  $p_{it}$ . The ground-truth label of stock  $v_{it}$  on trading day  $t$  based on the return between two consecutive trading days is denoted as  $y_{it} = \frac{p_{i,t+1} - p_{it}}{p_{it}} - \text{benchmark}_t$ , in which  $\text{benchmark}_t$  is the return of the benchmark index on trading day  $t$ .

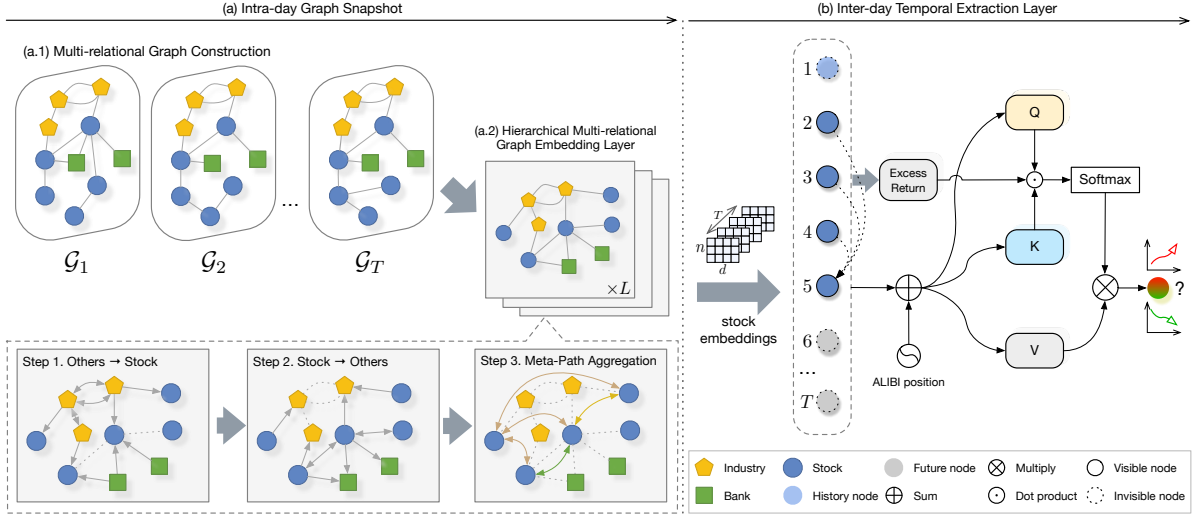


Figure 1: The overview architecture of the MDGNN Model.

We formulate the stock prediction as a node regression task that utilizes DGNN to learn a scoring function,  $f(\mathcal{G}; \Theta)$ , parameterized by  $\Theta$ . The scoring function is usually optimized by minimizing the loss function as:

$$\mathcal{L} = \sum_{\mathcal{N}} \ell\{\mathbf{Y}, f(\mathcal{G}; \Theta)\}, \quad (1)$$

where  $\mathcal{N}$  is the set of training samples, and  $\ell$  is the loss computed from each sample.

### Algorithm Design

In the following sections, we will describe the architecture of the MDGNN model as depicted in Figure 1, which includes the Intra-day layer, the Inter-day Temporal Extraction layer, and the prediction layer.

#### Intra-day Graph Snapshot

In this section, we outline the framework for capturing node representations from graph snapshots generated on each trading day. The framework comprises two key components: the construction of the multi-relation graph and the graph embedding layer.

**Multi-relational Graph Construction** The performance of a single stock is influenced by a wide range of factors beyond its individual characteristics. As the stock markets are complex and multifaceted, a singlet relation is not sufficient to depict the intricate relations. As such, it is important to consider the correlations between stocks from comprehensive relations so that a more accurate picture of the overall performance of stock markets can be depicted.

To tackle the intricacy of stock markets, we integrate relations from industry, investment banks, and stock pairs to establish a multi-relational graph. This approach enables us to reveal these complex connections and offer a deeper understanding of the underlying dynamics of the financial system.

(1) *Industry Graph*: The performance of a company and its corresponding stock is closely tied to the industry in which it operates. For instance, as an industry is growing rapidly, companies in that industry are likely to experience increased demand for their products or services, which will

lead to higher revenue and profits. As a result, the stock prices of these companies are likely to increase. Besides, the products manufactured by a company can either serve as raw materials for another industry or depend on the raw materials produced by another industry. Therefore, any increase in the cost of raw materials can result in an increase in the expenses of companies. Additionally, government regulations and policies toward industries can significantly impact the associated companies. To be specific, we represent the stock as  $\mathcal{S}$  and the industry as  $\mathcal{I}$ , while the connection between them is denoted as  $\mathcal{E}_{\mathcal{SI}}$ . This connection contains features that encode the aforementioned supply, demand, competition, and regulatory connections to account for the impact transmission from the industry.

(2) *Investment Bank Graph*: Investment banks greatly impact stock because they provide a wide range of services related to stock markets. Investment banks often act as market makers for stocks, meaning they provide liquidity to the market by buying and selling stocks on a regular basis. In addition, investment banks provide research reports on stocks regarding the company's financial performance, industry trends, and other factors. These relationships allow investment banks to have an impact on the price of the stock. We extract the buy, sell, research, and advisory relations from investment banks to capture the wield significant influence over stock prices. We denote the investment bank as  $\mathcal{B}$  and the connection between that and stock as  $\mathcal{E}_{\mathcal{SB}}$ , which incorporates the intricate aforementioned relations.

(3) *Stock Graph*: Stocks have a great impact on other stocks because of the interconnectedness of the stock market and the various factors that can affect stock prices. A company's earnings can lead to increased or reduced demand for its stock, as well as other companies in the same industry or sector. In some cases, companies can be held by the same owners, in which the co-holding relations offer a means of gauging the correlation among stocks. Additionally, common shareholders may engage in simultaneous buying or selling of a company's stock during a specific period of time. Therefore, the performance of stocks can be positively or negatively correlated with one another. To capture the interconnected nature of the stock market, we iden-

tify relationships between stocks based on factors such as sector, ownership, and co-holding relations. These relationships are denoted as  $\mathcal{E}_{SS}$ .

To create the multiplex relations mentioned earlier, we start by gathering daily trading data and textual data such as macroeconomic reports, financial news, financial statements, and research reports from TuShare<sup>1</sup>. We then employ financial lexicons and syntactic methods, as suggested by (Wang, Wang, and Li 2020), to build the edges between pairs of entities. Through intricate data preparation and extraction processes, we construct a multi-relational graph that integrates stocks, industries, and investment banks as nodes and multiplex relations as edges.

#### Hierarchical Multi-relational Graph Embedding Layer

As stated above, we construct a multi-relational graph from different relationships associated with stocks. This enables us to capture complex representations of the relationships between stocks, leading to a more comprehensive understanding of stock investment modeling. Concretely, we define a few meta-paths starting from stock nodes, such as “Stock-Stock (*SS*)”, “Stock-Bank-Stock (*SBS*)”, and “Stock-Industry-Industry-Stock (*SIIS*)”. As both nodes and edges have a distinct impact on stock nodes, we propose a hierarchical graph embedding layer that can aggregate and propagate information. Besides, edge features are crucial in graph-based models as they encode essential information about the relationships between nodes. For instance, in the context of stock market prediction, the features that encode supply, demand, competition, and regulatory connections between an industry and its corresponding stocks can provide valuable insights into the future trends of stocks.

Concretely, we utilize an attention mechanism when aggregating information from neighborhood nodes, allowing the model to attend to distinctive attributes of the edges and the nodes they connect. As demonstrated in (Veličković et al. 2018), using multi-head attention in the graph attention mechanism is advantageous. Hereby, the attention weight of the  $k$ -th head when aggregating the neighbors of target node  $v_i$  is conducted as follows:

$$\alpha_{ij}^k = \frac{\exp(\text{LeakyReLU}(a^T[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_j || \mathbf{W}\mathbf{e}_{ij}]))}{\sum_{j' \in \mathcal{N}_i} \exp(\text{LeakyReLU}(a^T[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_{j'} || \mathbf{W}\mathbf{e}_{ij'}]))}, \quad (2)$$

where  $\mathcal{N}_i$  denotes the neighborhood nodes of the target node  $i$ ,  $||$  represents the concatenation operation, and  $\mathbf{W}$  is the shared projection matrix. Moreover, we aggregate representations from multiple heads and use average pooling to update the target node’s representation as follows:

$$\mathbf{h}_i = \sigma \left( \frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j \right), \quad (3)$$

where  $\mathbf{W}^k$  is the shared projection matrix and  $K$  is the total number of heads. Consequently, we obtain the stock representations from each meta-path. Specifically, we define the stock representations from the meta-paths “*SS*”, “*SBS*”, and “*SIIS*” as  $\mathbf{h}_{i1}$ ,  $\mathbf{h}_{i2}$ , and  $\mathbf{h}_{i3}$ .

<sup>1</sup><https://tushare.pro/>

However, combining these representations effectively can be a challenging task. Attention mechanisms provide a solution by allowing the model to selectively focus on the most relevant representations for the target node. By assigning different attention weights to each representation, the model can effectively combine and aggregate the information from multiple meta-paths. This process not only captures the most critical information but also helps reduce the noise and redundancy in the representations.

Specifically, we design a relation-aware graph module that aggregates node and relation features from a multi-relational graph in an adaptive manner as follows:

$$\mathbf{h}_{v_i} = \sigma \left( \sum_{j=1}^3 \text{Softmax}(\mathbf{W}\mathbf{h}_{ij}) \mathbf{h}_{ij} \right), \quad (4)$$

where  $\mathbf{W}$  is a learnable matrix and  $\mathbf{h}_{v_i}$  is the representation of node  $v_i$  after incorporating multiple relations among nodes. By assigning different attention weights to each representation, the model can prioritize the most informative representations, enhancing the model’s ability to capture important patterns and relationships. Moreover, analyzing these weights makes it possible to understand the importance of each relation or meta-path in the final prediction. This interpretability is crucial for understanding the reasoning behind the model’s decision-making process.

To enhance the representations of stock nodes, we stack multiple hierarchical multi-relational graph embedding layers. The first layer captures local information, while subsequent layers capture increasingly global information. Hence, as we stack multiple graph layers, nodes that are distant from the originating node will be impacted. This facilitates the modeling of the intricate relationships involved in the transmission of stock information. This also enables the model to learn complex patterns and relationships in the graph, leading to improved performance. Specifically, we stack  $L$  GNN layers to obtain the representation of each stock node  $v$  on a trading day  $t$ , denoted as  $\mathbf{h}_{vt}$ , using the final GNN layer.

#### Inter-day Temporal Extraction Layer

Although the node representation is obtained through the graph embedding layer from each graph snapshot, the semantics of stocks and the relationships between them are constantly evolving. For instance, every day, securities companies adjust their positions by selling the stocks of a company purchased the previous day while buying stocks of companies that have not been purchased. Furthermore, the features of stocks (e.g., the momentum, volatility, and yield factors) are also changing due to the impact of market changes every day. Therefore, it is essential to capture the dynamic nature of nodes and the evolving relations among graph snapshots in the temporal order.

To tackle the aforementioned challenge, we have developed a temporal extraction module that employs the transformer structure (Vaswani et al. 2017). This module enables the extraction of the temporal evolution of graph snapshots’ propagation by taking in the representation of the target nodes within a time window.

Concretely, let  $\mathbf{H}_{v,t-\delta:t} = \{\mathbf{h}_{vt'} | t - \delta \leq t' \leq t\}$  represent the node’s representation, from trading day  $t$  and look-

ing back to the preceding  $\delta_t$  trading days. Hereby, the window size  $\delta_t$  is a hyperparameter. To facilitate with the structure, we transform  $h_{v,t-\delta_t:t}$  into query  $\mathbf{Q}$  and key  $\mathbf{K}$ , and value  $\mathbf{V}$  as follows:

$$\mathbf{Q}, \mathbf{K}, \mathbf{V} = \mathbf{W}_\mathbf{Q} \mathbf{H}_{\mathbf{v},t-\delta_t:t}, \mathbf{W}_\mathbf{K} \mathbf{H}_{\mathbf{v},t-\delta_t:t}, \mathbf{W}_\mathbf{V} \mathbf{H}_{\mathbf{v},t-\delta_t:t}, \quad (5)$$

where  $\mathbf{W}_\mathbf{Q}$ ,  $\mathbf{W}_\mathbf{K}$ , and  $\mathbf{W}_\mathbf{V}$  are trainable weight matrices of query, key, and value, respectively.

In addition, the temporal dependency of graph snapshots is crucial for modeling the dependency in the node’s representation. Over time, the historical relationship between stocks and current price changes will diminish, making stock prices more susceptible to the influence of recent events. Hereby, we leverage the relative position method proposed in ALIBI (Press, Smith, and Lewis 2022) that adds a static, non-learnable bias to the query-key dot product. It introduces an inductive bias in favor of recent events, as it imposes a penalty on attention scores between distant query-key pairs. Moreover, the penalty increases in proportion to the distance between a key and a query.

Moreover, we also employ the forward mask to prevent positions in the input sequence from attending to subsequent positions during the self-attention mechanism. It’s applied to the attention mechanism’s softmax operation to mask out the future positions, ensuring that each position can only attend to the previous positions.

We calculate the dot product of query and key vectors to capture information between any node pair via a self-attention network, in which the multiplicative operation efficiently captures complex feature interactions. Then we apply the softmax function to scale the attention weight before multiplying it with the corresponding value vector.

$$\mathbf{Z} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + m \cdot \mathbf{P} + \mathbf{M}\right)\mathbf{V}, \quad (6)$$

where  $m$  is a slope parameter,  $\mathbf{P}$  is the position bias introduced from ALIBI, and  $\mathbf{M}$  is the forward mask matrix.  $\mathbf{Z}$  combines the significant evolving patterns extracted from the historical data between time period  $t - \delta_t$  and  $t$ .  $\mathbf{z}_{vt}$  denotes the representation of stock node  $v$  on trading day  $t$ .

### Prediction Layer

In stock investment prediction, we aim to estimate the probability  $\hat{y}_{vt}$  that a given stock will yield a positive return on trading day  $t$  based on the stock’s representation  $\mathbf{z}_{vt}$  as:

$$\hat{y}_{vt} = \sigma(\mathbf{W}_1 \mathbf{z}_{vt} + \mathbf{b}_1). \quad (7)$$

where  $\mathbf{W}_1$  and  $\mathbf{b}_1$  are trainable matrices and bias;  $\sigma$  is the sigmoid activation function.

## Experiments

To validate the efficacy of our proposed method, we conducted extensive experiments using real Chinese stock market data. This section presents a comprehensive overview of our experimental setup, including the comparison baseline, dataset, features, graph relationships, and evaluation metrics. We then showcase the primary experimental results, followed by ablation, hyperparameter studies, and a case study of our proposed module.

## Experiment Setup

**Datasets.** First, we construct datasets using the CSI100 and CSI300 indices of China’s stock market with details in Table 1. Next, we extract a set of 42-dimensional features, which includes 25-dimensional market performance features such as opening price, closing price, change percentage, volatility, and turnover rate, 12-dimensional company valuation features such as P/E ratio, P/B ratio, and P/S ratio, 4-dimensional company categorical features, and 1-dimensional institutional consensus expectations feature. All of these features are normalized prior to analysis.

Table 1: Detailed statistics of the datasets.

	# stocks	# investment banks	# industries	# edges
CSI100	100	196	97	18,950,706
CSI300	300	202	191	62,500,988

**Backtest.** We use the timeframe from 01/01/2020 to 02/31/2023 for backtest. The training cycle is set at half a year, meaning that we train the model every six months, resulting in a total of seven models in the experimental set cycle. The training set utilizes data and labels from the preceding six months, while the validation set employs those from last month. The model utilizes fixed parameter values for prediction during the following six months.

**Baselines.** To show the performance of our proposed model, we compare MDGNN with SOTA methods. We select the following models as the baseline for comparison: (1) Traditional time series modeling methods (MLP, LSTM (Hochreiter and Schmidhuber 1997), Transformer (Devlin et al. 2019)) leverage only the intrinsic characteristics of the stock and do not consider the accompanying graph information. (2) Homogeneous graph methods (GCN (Kipf and Welling 2017), GAT (Velićković et al. 2018)) solely utilize relationships between stocks and do not incorporate other types of multiplex relationship information. (3) Heterogeneous graph methods (RGCN (Schlichtkrull et al. 2017), HAN (Han, Kim, and Enke 2023), HGT (Hu et al. 2020)) exploit heterogeneous relationships but lack the capability to mine dynamic relationships. (4) Dynamic graph methods (EvolveGCN (Pareja et al. 2020), HTGNN (Fan et al. 2021)) incorporate the mining of dynamic graph relationships as a fundamental aspect of their approach.

**Metrics.** We utilize Information Coefficient (IC) (Li et al. 2019), Information Ratio (IR), Cumulative Return (CR), and Precision@K (Järvelin and Kekäläinen 2000) as evaluation metrics. IC evaluates the overall ranking performance, and IR divides the excess return of a portfolio by its tracking error. CR is the accumulated portfolio return based on the prediction score. Precision@K evaluates whether the excess returns of TopK stocks outperform the benchmark index.

**Implementation Details.** Our experiment is trained with Nvidia V100 GPU, and all models are built using PyTorch. The hidden size was set to 128, the number of GNN layers is 2, and the window size is 10. The training and validation sets are kept consistent across all models. To ensure that all models receive sufficient training, we train each for 500 epochs and implement an early stopping strategy with patience of 20. This approach is adopted to ensure thorough training and

Table 2: Results of methods on public datasets. The last row in each dataset indicates the percentage of improvements gained by the proposed method w.r.t the best-performed baseline. Prec@k is a shortened form of Precision@k.

Methods	CSI 100						CSI 300					
	IC	IR	CR	Prec@3	Prec@10	Prec@30	IC	IR	CR	Prec@3	Prec@10	Prec@30
MLP	0.0027 (2.25e-03)	0.0282 (2.26e-02)	0.1166 (8.10e-03)	0.4889 (8.74e-03)	0.4849 (2.13e-03)	0.4751 (8.17e-04)	0.0039 (9.42e-04)	0.0314 (1.53e-02)	0.1721 (1.08e-02)	0.4952 (1.04e-02)	0.5004 (2.02e-03)	0.4958 (1.01e-03)
LSTM	0.0040 (1.27e-03)	0.0335 (1.31e-02)	0.1289 (1.90e-03)	0.4915 (4.59e-03)	0.4892 (1.01e-03)	0.4808 (2.70e-04)	0.0049 (6.84e-04)	0.0345 (1.09e-02)	0.1859 (1.29e-02)	0.5008 (6.68e-03)	0.5015 (2.23e-03)	0.4958 (1.99e-03)
Transformer	0.0058 (2.50e-03)	0.0422 (1.51e-02)	0.1383 (7.47e-02)	0.5131 (6.51e-03)	0.5106 (2.95e-03)	0.4987 (3.22e-03)	0.0063 (1.95e-03)	0.0442 (1.28e-02)	0.2122 (1.14e-01)	0.5041 (7.49e-03)	0.5078 (4.32e-03)	0.5065 (6.03e-03)
GAT	0.0031 (9.08e-04)	0.0274 (7.63e-03)	0.1534 (2.45e-02)	0.4959 (9.72e-03)	0.4931 (3.25e-03)	0.4812 (2.31e-03)	0.0066 (1.50e-03)	0.0454 (2.46e-02)	0.2653 (2.42e-02)	0.4897 (6.30e-03)	0.5064 (3.05e-03)	0.4991 (3.00e-04)
GCN	0.0038 (1.34e-03)	0.0305 (9.36e-03)	0.1616 (8.64e-03)	0.4983 (7.75e-03)	0.4960 (1.60e-03)	0.4927 (2.33e-03)	0.0075 (9.85e-04)	0.0674 (3.80e-02)	0.2816 (2.88e-02)	0.4984 (1.23e-02)	0.5077 (7.78e-03)	0.5055 (2.04e-03)
RGCN	0.0104 (1.29e-03)	0.0578 (7.47e-03)	0.1912 (2.84e-02)	0.5050 (6.57e-03)	0.5024 (4.31e-03)	0.4985 (2.59e-03)	0.0090 (1.69e-03)	0.0845 (1.42e-02)	0.3743 (5.32e-02)	0.5083 (7.89e-03)	0.5159 (5.14e-03)	0.5104 (1.85e-03)
HAN	0.0108 (4.08e-04)	0.0525 (2.69e-03)	0.2267 (2.48e-02)	0.5068 (3.55e-03)	0.5043 (3.20e-03)	0.4997 (3.25e-03)	0.0086 (4.68e-03)	0.0848 (4.53e-02)	0.3511 (5.72e-02)	0.5185 (3.68e-03)	0.5166 (6.68e-03)	0.5112 (4.53e-03)
HGT	0.0112 (1.35e-03)	0.0657 (7.46e-03)	0.2384 (1.98e-02)	0.5198 (3.80e-03)	0.5127 (1.19e-03)	0.5036 (4.72e-03)	0.0115 (2.05e-03)	0.0874 (1.17e-02)	0.4108 (5.65e-02)	0.5051 (1.56e-03)	0.4937 (9.48e-03)	0.4923 (6.93e-03)
EvolveGCN	0.0065 (3.54e-04)	0.0538 (3.18e-03)	0.1815 (2.81e-02)	0.5090 (2.90e-03)	0.5043 (2.83e-03)	0.4961 (2.26e-03)	0.0080 (3.46e-04)	0.0811 (4.69e-03)	0.3215 (6.09e-02)	0.5012 (5.27e-03)	0.4989 (9.59e-03)	0.4830 (3.11e-03)
HTGNN	0.0118 (3.76e-03)	0.0724 (2.45e-02)	0.2643 (8.23e-02)	0.5288 (1.16e-02)	0.5116 (5.55e-03)	0.5039 (3.54e-03)	0.0192 (7.59e-04)	0.1773 (9.94e-03)	0.4653 (7.03e-02)	0.5205 (1.01e-02)	0.5142 (1.24e-03)	0.5126 (1.12e-03)
MDGNN	<b>0.0123</b> (2.75e-03)	<b>0.0746</b> (1.59e-02)	<b>0.2741</b> (8.11e-02)	<b>0.5436</b> (6.30e-03)	<b>0.5236</b> (3.15e-03)	<b>0.5081</b> (3.22e-03)	<b>0.0322</b> (2.43e-03)	<b>0.2488</b> (4.19e-03)	<b>0.9828</b> (1.13e-02)	<b>0.5281</b> (6.90e-03)	<b>0.5276</b> (5.60e-03)	<b>0.5232</b> (3.01e-03)

optimal performance for each model.

## Experiment Result

The results of our proposed method, as well as the other baseline models, are presented in Table 2 for CSI100 and CSI300 datasets. Our model outperforms all other methods across all metrics. Based on these experimental findings, we draw the following conclusions:

1) **Time series modeling and static graphs:** Traditional time series modeling methods, such as MLP, primarily rely on the intrinsic node features of the stock, while LSTM/Transformer places greater emphasis on temporal features. However, homogeneous graph-based approaches, such as GAT and GCN, only consider node features and stock connections. Despite their inferior performance on the CSI100 dataset, these methods outperform Transformer on the larger CSI300 dataset. This observation underscores the significance of establishing connections between stocks.

2) **Heterogeneous and dynamic graphs:** The inclusion of diverse heterogeneous graph information in algorithms, such as RGCN, HAN, and HGT, has led to notable performance enhancements over prior approaches. Additionally, we compared time series heterogeneous graph-based methods such as EvolveGCN and HTGNN, which are designed for temporal and multi-relational graphs, respectively. Our findings suggest that the incorporation of both temporal and multi-relational graph information can yield further improvements in performance.

3) **Our proposed method:** Our proposed MDGNN algorithm, leveraging enhanced modules to capture information from the distinctive multi-relational graph structure of stocks, surpasses previous time series heterogeneous graph-based algorithms on both datasets. Moreover, the performance improvement is more pronounced in the CSI300 dataset compared to the CSI100 dataset. This outcome can be attributed to the inclusion of additional institutional and industry nodes, which results in a larger training graph and

enables more effective information propagation. These findings provide further evidence of the effectiveness of constructing graphs for stock trend prediction.

## Ablation Study

**Effect of Components.** To validate the design choices in our proposed framework, we perform an ablation experiment by removing four components individually: edge weight (w/o edge), meta-path (w/o meta-path), hierarchical aggregation (w/o aggregation), and temporal extraction layer (w/o temporal). The experiment is performed on the CSI300 dataset, and the results are presented in Table 3. We observe that the removal of the meta-path module results in the most significant decrease in performance, thereby confirming the effectiveness of the multi-relational graph in our framework.

Table 3: The results of the effect of components.

	IC	IR	CR	Prec@30
w/o edge	0.0268	0.2155	0.8950	0.5152
w/o meta-path	0.0216	0.1723	0.7502	0.5076
w/o aggregation	0.0303	0.2392	0.9402	0.5227
w/o temporal	0.0286	0.2226	0.8745	0.5215
MDGNN	0.0322	0.2488	0.9828	0.5232

**Effect of Relations.** To further confirm the effectiveness of each relationship in our multi-relational graph, we present the results in Table 4. Here,  $SS$ ,  $SB$ ,  $SI$ , and  $II$  refer to the relationships between stocks and stocks, stocks and investment banks, stocks and industries, and industries and industries, respectively. The default connection between different node types is bidirectional, and if the required edges in the meta-path are removed, the corresponding meta-path will also be removed. Our findings indicate that the performance is comparable to EvolveGCN when using only the  $SS$  edge in the homogeneous graph. After introducing the  $SB$  and  $SI$  edges, the performance improves to some extent, thereby confirming our basic assumption of constructing a

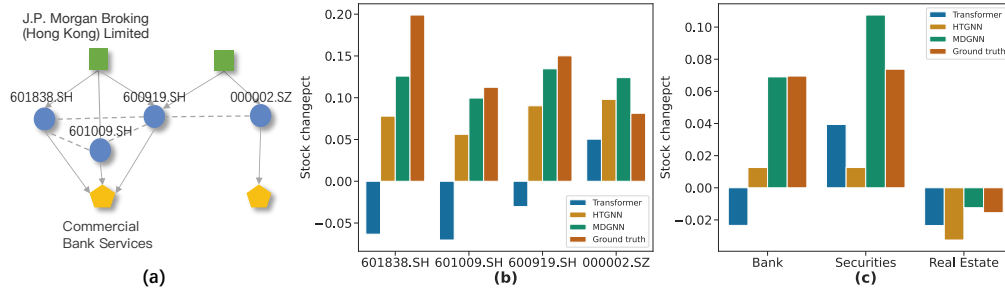


Figure 2: The results of the case study.

multi-relational graph, namely that the stock price changes of stocks held by the same investment bank or belonging to the same industry exhibit a certain degree of consistency. Moreover, after introducing the  $\mathcal{II}$  edge, the significance of connecting similar industries is validated, as it is more beneficial for information transmission among stocks belonging to similar industries. Furthermore, the introduction of the connection between investment banks yields a more significant effect than the connection between industries, as the former brings about greater differences in information when multiple investment banks hold a stock, whereas it can only belong to one industry.

Table 4: The results of the effect of relations.

$SS$	$SB$	$SI$	$II$	IC	IR	CR	Prec@30
✓	-	-	-	0.0217	0.1727	0.7372	0.5128
✓	✓	-	-	0.0264	0.2092	0.8220	0.5203
✓	-	✓	-	0.0210	0.1632	0.7133	0.5101
✓	-	✓	✓	0.0217	0.1802	0.7755	0.5134
✓	✓	✓	-	0.0283	0.2300	0.9074	0.5208
✓	✓	✓	✓	0.0322	0.2488	0.9828	0.5232

## Case Study

A research report on investment bank holdings reveals a rising credit pulse trend from December 2021 to March 2022, accompanied by an increase in the proportion of bank holdings by international investment institutions. To analyze this trend, we focus on Chengdu Bank (601838.SH) and its sub-graph, which consists of four stock nodes: Chengdu Bank (601838.SH), Nanjing Bank (601009.SH), Jiangsu Bank (600919.SH), and Vanke A (000002.SZ). The first three stocks belong to the commercial banking service industry and are held by the same institution. The last stock belongs to the real estate industry and is held by a different institution, which also holds both Vanke A and Jiangsu Bank.

Figure 2(b) shows the average stock change rates for the four selected stocks between January 4 and 17, 2022. Traditional temporal models predict negative change rates for most bank-related stocks, except for 000002.SZ. However, the heterogeneous temporal model enables the upward trend to propagate through multiple relation graphs, influencing the change rates of all bank-related stocks and leading to an increase in the change rate of 600919.SH. The MDGNN model outperforms other models, demonstrating its strong ability to handle multiple relation graphs. Further analysis of the average stock change rates of three key industries during this period is shown in Figure 2(c). These industries include banking, securities, and real estate, comprising 22, 7, and

27 stocks, respectively. The findings indicate that the utilization of multiple relational graphs has a more pronounced influence on banks and real estate than securities. This observation can be attributed to the consistent trends observed among stocks within the same industry, as well as the larger number of stocks present within these industries, which facilitates the effective propagation of graph information.

## Hyperparameter Study

We also design some experiments to check the sensitivity of hyperparameters. In Figure 3(a), the changes in cumulative return are depicted across different window sizes. It appears that increasing the window size improves the effect, but only up to a certain limit for information capture. Similarly, as the number of GNN layers increases in Figure 3(b), the effect also improves gradually, but an excessively high complexity can lead to a decline in performance.

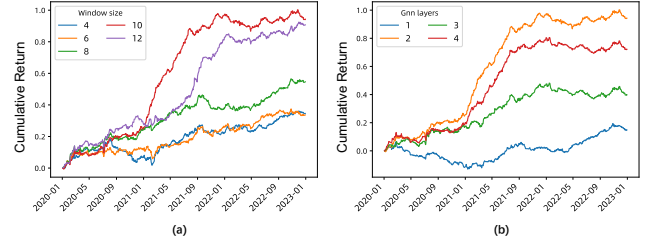


Figure 3: The results of hyperparameter study.

## Conclusion

In this work, we formally define the multifacetedness and temporal patterns of stocks through empirical analysis for the first time and propose a novel hierarchical multi-relational dynamic graph framework for modeling stock investment prediction. Our approach involves constructing a multi-relational graph for each trading day and generating a set of discrete graph snapshots within the specified look-back window size. In terms of the intra-day graph snapshot, we design a hierarchical multi-relational graph embedding layer to first aggregate the neighbor nodes within a specific meta-path and then adaptively integrate the stock representation from distinct meta-paths. Furthermore, we incorporate the transformer structure to aggregate the temporal evolving patterns of stocks. We demonstrate the effectiveness and robustness of our proposed framework through extensive experiments. In the future, we would like to study MDGNN with contrastive learning methods for stock investment prediction and improve performance even further.



## References

- Chen, Q.; and Robert, C. Y. 2021a. Graph-Based Learning for Stock Movement Prediction with Textual and Relational Data. In *The Journal of Financial Data Science*.
- Chen, Q.; and Robert, C.-Y. 2021b. Graph-based learning for stock movement prediction with textual and relational data. *arXiv preprint arXiv:2107.10941*.
- Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 4171–4186.
- Fan, Y.; Ju, M.; Zhang, C.; Zhao, L.; and Ye, Y. 2021. Heterogeneous Temporal Graph Neural Network. *arXiv:2110.13889*.
- Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; and Chua, T.-S. 2019a. Enhancing Stock Movement Prediction with Adversarial Training. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, 5843–5849. International Joint Conferences on Artificial Intelligence Organization.
- Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; and Chua, T.-S. 2019b. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2): 1–30.
- Gu, S.; Kelly, B.; and Xiu, D. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5): 2223–2273.
- Han, Y.; Kim, J.; and Enke, D. 2023. A machine learning trading system for the stock market based on N-period Min-Max labeling using XGBoost. *Expert Systems with Applications*, 211: 118581.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.
- Hu, Z.; Dong, Y.; Wang, K.; and Sun, Y. 2020. Heterogeneous Graph Transformer. In *Proceedings of The Web Conference 2020, WWW '20*, 2704–2710. New York, NY, USA: Association for Computing Machinery. ISBN 9781450370233.
- Jain, L. C.; and Medsker, L. R. 1999. *Recurrent Neural Networks: Design and Applications*. USA: CRC Press, Inc., 1st edition. ISBN 0849371813.
- Järvelin, K.; and Kekäläinen, J. 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00*, 41–48. New York, NY, USA: Association for Computing Machinery. ISBN 1581132263.
- Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Li, Z.; Yang, D.; Zhao, L.; Bian, J.; Qin, T.; and Liu, T.-Y. 2019. Individualized Indicator for All: Stock-Wise Technical Indicator Optimization with Stock Embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19*, 894–902. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362016.
- Lin, H.; Zhou, D.; Liu, W.; and Bian, J. 2021. Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21*, 1017–1026. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383325.
- Nagel, S. 2021. *Machine learning in asset pricing*, volume 8. Princeton University Press.
- Nelson, D. M.; Pereira, A. C.; and De Oliveira, R. A. 2017. Stock market's price movement prediction with LSTM neural networks. In *2017 International joint conference on neural networks (IJCNN)*, 1419–1426. Ieee.
- Pareja, A.; Domeniconi, G.; Chen, J.; Ma, T.; Suzumura, T.; Kanezashi, H.; Kaler, T.; Schardl, T.; and Leiserson, C. 2020. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 5363–5370.
- Press, O.; Smith, N. A.; and Lewis, M. 2022. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. *arXiv:2108.12409*.
- Roy, S. S.; Mittal, D.; Basu, A.; and Abraham, A. 2015. Stock market forecasting using LASSO linear regression model. In *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014*, 371–381. Springer.
- Sawhney, R.; Agarwal, S.; Wadhwa, A.; Derr, T.; and Shah, R. R. 2021. Stock Selection via Spatiotemporal Hypergraph Attention Network: A Learning to Rank Approach. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1): 497–504.
- Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; van den Berg, R.; Titov, I.; and Welling, M. 2017. Modeling Relational Data with Graph Convolutional Networks. *arXiv:1703.06103*.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. *International Conference on Learning Representations*.
- Wang, H.; Li, S.; Wang, T.; and Zheng, J. 2021a. Hierarchical Adaptive Temporal-Relational Modeling for Stock Trend Prediction. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 3691–3698. International Joint Conferences on Artificial Intelligence Organization. Main Track.
- Wang, H.; Li, S.; Wang, T.; and Zheng, J. 2021b. Hierarchical Adaptive Temporal-Relational Modeling for Stock Trend Prediction. In *IJCAI*, 3691–3698.



Wang, H.; Wang, T.; Li, S.; Zheng, J.; Guan, S.; and Chen, W. 2022. Adaptive Long-Short Pattern Transformer for Stock Investment Selection. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 3970–3977. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Wang, H.; Wang, T.; and Li, Y. 2020. Incorporating Expert-Based Investment Opinion Signals in Stock Prediction: A Deep Learning Framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01): 971–978.

Xu, W.; Liu, W.; Wang, L.; Xia, Y.; Bian, J.; Yin, J.; and Liu, T.-Y. 2021a. HIST: A Graph-based Framework for Stock Trend Forecasting via Mining Concept-Oriented Shared Information. *arXiv preprint arXiv:2110.13716*.

Xu, W.; Liu, W.; Wang, L.; Xia, Y.; Bian, J.; Yin, J.; and Liu, T.-Y. 2021b. Hist: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. *arXiv preprint arXiv:2110.13716*.

Zhang, L.; Aggarwal, C.; and Qi, G.-J. 2017. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, 2141–2149. New York, NY, USA: Association for Computing Machinery. ISBN 9781450348874.