Dynamic Graph Representation with Contrastive Learning for Financial Market Prediction: Integrating Temporal Evolution and Static Relations

Yunhua Pei[1][a], Jin Zheng[2][b] and John Cartlidge[2][c]
[1]*School of Computer Science, University of Bristol, U.K.*
[2]*School of Engineering Mathematics and Technology, University of Bristol, U.K.*
{*ge22472, jin.zheng, john.cartlidge*}*@bristol.ac.uk*

Keywords: Contrastive Learning, Financial Market Forecasting, Graph Neural Networks, Temporal Graph Learning.

Abstract: Temporal Graph Learning (TGL) is crucial for capturing the evolving nature of stock markets. Traditional methods often ignore the interplay between dynamic temporal changes and static relational structures between stocks. To address this issue, we propose the Dynamic Graph Representation with Contrastive Learning (DGRCL) framework, which integrates dynamic and static graph relations to improve the accuracy of stock trend prediction. Our framework introduces two key components: the Embedding Enhancement (EE) module and the Contrastive Constrained Training (CCT) module. The EE module focuses on dynamically capturing the temporal evolution of stock data, while the CCT module enforces static constraints based on stock relations, refined within contrastive learning. This dual-relation approach allows for a more comprehensive understanding of stock market dynamics. Our experiments on two major U.S. stock market datasets, NASDAQ and NYSE, demonstrate that DGRCL significantly outperforms state-of-the-art TGL baselines. Ablation studies indicate the importance of both modules. Overall, DGRCL not only enhances prediction ability but also provides a robust framework for integrating temporal and relational data in dynamic graphs. Code and data are available for public access.

1 INTRODUCTION

The goal of predicting stock movements attracts much attention, as success offers the opportunity to generate substantial investment returns. Previous research can generally be divided according to the input features used in the model. Historical price is the most common and basic input feature, including the open price, close price, highest price, lowest price, and trading volume (OHLCV). Many predictions have been made based solely on historical prices, including applying empirical mode decomposition with factorization machine-based neural networks (Zhou et al., 2019) and forming a stochastic recurrent network with seq2seq architecture and attention mechanism (Wang et al., 2019). More attempts include adding technical indicators such as achieving Long Short-Term Memory (LSTM) with the attention mechanism (Chen and Ge, 2019), creating multi-task RNNs with high-order Markov random fields (Li et al., 2019).

Graphs, which represent information through entities, attributes, and their relationships, have recently gained popularity. This is due to their effectiveness in illustrating the connections between stocks and their various attributes. By casting a graph, relevant information transfer between stocks and stock attributes from other channels can be studied. For instance, the attention mechanism is used to build a market knowledge graph to contain dual-type entities and mixed relationships (Zhao et al., 2022), GCNs with temporal graph convolution are performed on data in a rolling window (Matsunaga et al., 2019), and incorporating emotional factors within a non-stationary Markov chain model (Liu et al., 2022a). These graph-based models leverage the structure of GNNs to iteratively aggregate, showing potentially powerful performance compared with non-graph-based models.

Although previous research has delved into comprehensive investigations of the stock market, several key issues still need to be solved. The first issue is forecasting a single time series, which does not capture the overall market trend and leaves the relationships between stocks and the mechanisms influencing price transmission unexplored. The second one

[a] https://orcid.org/0000-0003-2906-0827
[b] https://orcid.org/0000-0002-1783-1375
[c] https://orcid.org/0000-0002-3143-6355

Pei, Y., Zheng, J. and Cartlidge, J.
Dynamic Graph Representation with Contrastive Learning for Financial Market Prediction: Integrating Temporal Evolution and Static Relations.
DOI: 10.5220/0013154700003890

In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025) - Volume 2*, pages 298-309
ISBN: 978-989-758-737-5; ISSN: 2184-433X

is that constructing stock market graphs relies heavily on prior knowledge, and is challenging to adapt automatically. The third is relations obtained from the knowledge base are not effectively utilized and are only used as tools for building graph networks.

To address these limitations, we propose a framework called Dynamic Graph Representation with Contrastive Learning (DGRCL), which consists of three main components, namely embedding enhancement, contrastive constrained training, and dynamic graph evolution. In embedding enhancement, we adaptively construct dynamic edges that follow Zipf's law. Subsequently, the enhanced features are derived from OHLCV, computed with refined distance calculations utilizing the Fourier transform. Using these enhanced features and dynamic edges, we continue to apply GCNs to these evolving graphs to extract the latent node embeddings. To further refine the embeddings and optimize predictions, we integrate company relationships as constraints within an adaptive contrastive learning module. This module employs contrastive loss in the latent space to maximize the consistency between two augmented views of the input graph, thereby ensuring more robust feature representation. Finally, our model captures dynamic changes in the graph over time by employing a time series forecasting architecture, which can utilize either a Gated Recurrent Unit (GRU) or LSTM. This enables DGRCL to generate accurate and temporally-aware predictions.

To the best of our knowledge, our work is the first to incorporate stock relations as constraints into contrastive learning models for stock prediction, especially those based on GNNs. The main contributions of this work can be summarised as follows:

- We adaptively construct dynamic edges that follow the objective laws, refining stock features through Fourier transform-based distance calculations. This allows the model to more effectively capture the intricate and dynamic relationships between stocks.
- We integrate company relationships as constraints within a contrastive learning module, employing contrastive loss in the latent space to maximize the consistency between two augmented views of the input graph. This ensures a robust feature representation that reflects the complex interdependencies among stocks.
- We conduct extensive experiments on real-world stock data from the US market. We conduct extensive experiments on real-world stock datasets with 2,763 stocks from two famous US markets (NASDAQ and NYSE). The experimental results demonstrate that DGRCL significantly outperforms state-of-the-art baselines in predicting the next trading day movement, with average improvements of 2.48% in classification accuracy, 5.53 in F1 score, and 6.67 in Matthew correlation coefficient.

2 RELATED WORK

2.1 GNNs

GNNs are designed with specific consideration for tasks involving graph data, such as node classification, graph classification, and link prediction. In GALSTM (Yin et al., 2022), an attention-based LSTM is constructed to learn weight matrices from Hawkes correlation graphs generated by different stocks to improve the stock selection winning rate while paying less attention to the optimal trading strategy. The most profitable stock prediction is also one task of portfolio management. Intra-sector relations, inter-sector relations, and stock-sector relations are considered in the previous study (Hsu et al., 2021) to find hierarchical influence among stocks and predict the most profitable stock. Although sector-level modeling is performed in their model, some fine-grained metadata with descriptions of listed company attributes deserves more attention. Stock description documents can also be introduced into the model. For example, Yahoo Finance is one of the channels for topic-based extraction models in TRAN (Gao et al., 2021) to make stock recommendations. Similar limitation, TRAN only constructs the stock graph as a static one. Current approaches need to focus on portfolio construction and selection, as well as on understanding the dynamic interactions between these portfolios.

2.2 Graph Contrastive Learning

Recently, contrastive representation learning on graphs has gained significant interest. It contrasts positive and negative sample pairs, aiming to bring similar samples closer and push dissimilar ones apart, focusing on solving the graph manual labeling problem in the real world. According to (Liu et al., 2022c), previous research can be categorized into cross-scale contrasting and same-scale contrasting. Cross-scale contrasting involves comparing elements at different scales. For example, MVGRL (Hassani and Khasahmadi, 2020) takes a multi-view approach by using a diffused graph as a global view and maximizing MI across both views and scales. DGI (Velickovic et al., 2019) contrasts patch-level and graph-level representations by maximizing mutual information, which helps propagate global information to local representations. Same-scale contrasting focuses on elements at the same scale, such as comparing graphs directly. Within this approach, methods are further divided into context-based and augmentation-based. Context-based methods typically use random walks to identify positive pairs. Augmentation-based methods, like GraphCL (You et al., 2020), generate positive pairs through various perturbations at the graph level. GCA (Zhu et al., 2021) performs node-level augmentations tailored to the graph's structure and attributes, while IGSD (Zhang et al., 2023) uses graph diffusion to create augmented views and employs a teacher-student framework.

3 PRELIMINARY

3.1 Notation

3.1.1 Graph Representation Learning

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the graph model where $\mathcal{V}$ is the set of nodes. The length of the time series is T, and $\boldsymbol{X}^i \in \mathbb{R}^{D \times T}$ is the covariate time-series data associated with node i, where each node is associated with D different covariate time series. Furthermore, at time step t, covariate values for node i is represented as $\mathbf{x}_t^i$, and $\boldsymbol{X}_T^i = \left\{\mathbf{x}_1^i, ..., \mathbf{x}_T^i\right\}$ is the set of time-series in graph $\mathcal{G}$. Two different nodes i and j that contain an edge $(i, j) \in \mathcal{E}$ without a direction, forming graph $\mathcal{G}$ an undirected one, encoding an explicit dependence between nodes. Let $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ denote this sparse adjacency matrix at a certain time. If $(i, j) \in \mathcal{E}$, then $\boldsymbol{A}_{ij}$ represents the weight of the edge (dependency) between node i and j, and $\boldsymbol{A}_{ij} = 0$ when $(i, j) \notin \mathcal{E}$ otherwise. Hence, the graph structure is represented by the adjacency matrix $\boldsymbol{A}$ and its corresponding time-series $\left\{\boldsymbol{X}_T^i | i = 1, ..., N\right\}$.

3.1.2 Graph Contrastive Learning

Given an input Graph $\mathcal{G}$, graph contrastive learning aims to learn the representations in node-level tasks in this work by maximizing the feature consistency between two augmented views of the input graph via contrastive loss in the latent space. Data augmentation $q(\cdot)$ is a common operation to obtain two views v_a and v_b for the same graph $\mathcal{G}$, including node dropping, edge perturbation, subgraph sampling, etc. With these views, latent representations z_a and z_b are then extracted by GNNs. Finally, given the latent representations, a contrastive loss is optimized to score the positive pairs $\left\{z_a^i, z_b^i\right\}$ higher compared to other negative pairs, including inter-view negative pairs $\left\{z_a^i, z_b^j\right\}$, and intra-view negative pairs $\left\{z_a^i, z_a^j\right\}$.

3.2 Problem Setting

The proposed framework solves the following graph-based time-series forecasting problem. For all stocks, a set of stock historical sequence data at day t is represented by $\mathcal{X} \in \mathbb{R}^{N \times T \times D}$, where D is the dimension of features of one stock, the OHLCV. With a given lookback window of length δ, the model can make predictions for the next day's $(t+1)$ movements of the stocks, using the data of $\mathcal{X}_{t-\delta:t}$. The movement is calculated by $p_c^{t+1} - p_c^t$, in which p_c^t means the *close* price of the stock at day t, resulting two prediction outcomes. A positive result means the price will go up $(label = 1)$, while a negative result means it will go down $(label = 0)$, making this a binary classification task.

4 METHODOLOGY

In this section, we elucidate the framework of the proposed DGRCL as shown in Fig.1. Given OHLCV of the stocks in a certain interval $[t - \delta + 1, t]$ and the company relations, we aim to predict the next day's stock price movement. First, we construct the stock graph with envolving edges for different time steps and generate initial embeddings for each node. Second, we extract the dynamic node representations for each time step. Then, a constraint data augmentation method is encoded using the company relations. After that, we feed the constrained dynamic node representations to an RNN-based model to get the final representation of the nodes. Finally, we predict the stock movement based on the generated node representation from RNNs.

4.1 Embedding Enhancement

4.1.1 Graph Construction

The future performance of a stock is influenced by its past behavior and its connections to related stocks. Current research often depends heavily on established knowledge, such as manual industry classifications (Huynh et al., 2023; Feng et al., 2019; Xu et al., 2021). However, this approach can be expensive, relies heavily on empirical experience, and may not generalize well to new forecasting tasks. In this regard, we follow the data-driven manner in (Wang et al., 2022) to calculate a proximity function by applying Dynamic

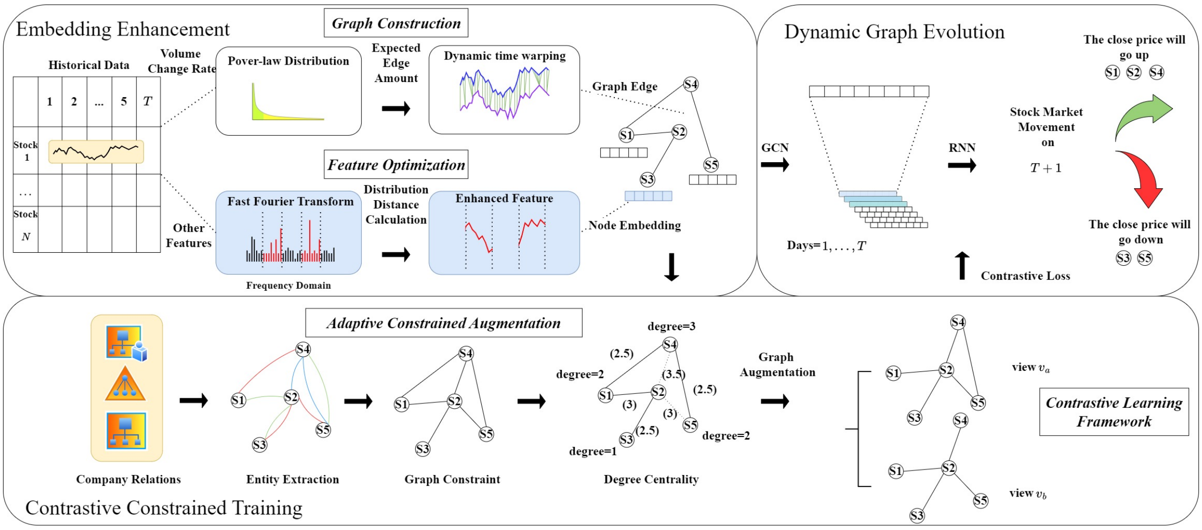

Figure 1: Schematic illustration of DGRCL framework. Top-left: In the process of embedding enhancement, dynamic stock graphs with enhanced features are generated. Bottom: Then, an encoder is trained with the company relations in a contrastive manner inside the CCT module, to improve the generation of an initial embedding matrix for the subsequent step. Top-right: Finally, the layer weights are learned through general RNNs (e.g., GRU, LSTM) to yield the final predictions.

Time Warping (DTW) (Jeong et al., 2011) on the input signals $\mathbf{x}_t^i$, which will be further introduced in the experiment part. This process involves utilizing dynamic programming on a path matrix to determine the minimum alignment cost for each pair of stock sequences $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ with a lookback window δ:

$$cost(\mathbf{x}_t^i, \mathbf{x}_t^j) \Longleftarrow DTW(\mathbf{x}_{t-\delta:t}^i, \mathbf{x}_{t-\delta:t}^j) \quad (1)$$

In DGRCL, we set the time step as 1. Through computations for each pair of stock sequences within the same time step, a corresponding cost matrix $\dot{\boldsymbol{A}}_t$ is generated for that particular time step t, and the total number of the cost matrices is $T-\delta$.

Traditional methods typically employ a hard threshold to partition the cost matrix $\dot{\boldsymbol{A}}_t$ and form the adjacency matrix $\boldsymbol{A}_t$ of the graph (Wang et al., 2022). However, such approaches are often subjective. The adjacency matrix, which represents the edges in a graph, heavily relies on the structural information of its edges for information transmission. Hence, we introduce an adaptive method to estimate the number of significant edges that a graph should possess based on the number of nodes it contains.

According to (Sawhney et al., 2021b), the scale-free structure of stock graphs is characterized by a power-law distribution in node degrees, at least in an asymptotic sense. Specifically, the fraction $P(n)$ of nodes in the graph having n connections to other nodes follows the distribution (Barabási and Albert, 1999), where s is the exponent.

$$P(n) \sim n^{-s} \quad (2)$$

This observation can be explained by Zipf's law (Newman, 2005), an empirical principle that tends to hold approximately when a list of measured values is sorted in descending order. The probability mass function of the Zipf distribution is a discrete probability distribution as,

$$P(X=n) = \frac{n^{-s}}{\sum_{i=1}^{N}(i^{-s})} \quad (3)$$

where N is the total number of possible outcomes, n is a specific outcome. Then we can give the expectation of a random variable X with a Zipf distribution using the formula:

$$E(X) = \sum_{n=1}^{N} n \cdot P(X=n) \quad (4)$$

It should be noted that when applied, n will represent the count of known nodes, signifying the number of stocks in the stock graph. This approach allows us to provide a rough estimate of the expected number of edges $E(n)$ in a stock network based on a given number of nodes n. Then, by employing a binary classification method, we manage the number of values exceeding the threshold to remain below the specified limit $E(n)$. This enables us to dynamically construct edges to build a stock graph $\mathcal{G}$. Furthermore, we obtain the dynamic graph for each time step as $\mathcal{G}_1, \mathcal{G}_2 ... \mathcal{G}_{T-\delta}$.

4.1.2 Feature Optimization

Statistical properties such as mean and variance in time series data often change over time, leading to distribution shift problems (Du et al., 2021; Kim et al., 2021). This change in time distribution poses a significant challenge in stock series to achieving accurate time series forecasting. Therefore, we need to enhance the input stock series.

To identify the distribution characteristics of a time series, it needs to be transformed into another domain, such as the frequency domain. The Fourier transform (FT) (Cochran et al., 1967; Bloomfield, 2004) is an integral transform that presents a function into a representation that emphasizes the original function, which has wide applications in physics, mathematics, and engineering.

Here, we apply the Fast Fourier Transformation (FFT) (Cochran et al., 1967) to one stock series, and follow the temporal distribution characterization in (Du et al., 2021), with the principle of maximum entropy (Jaynes, 1982) to find the most dissimilar periods, whose objective can be formulated as:

$$\max_{0<k\leq k_0} \max_{n_1,\ldots,n_k} \frac{1}{k} \sum_{1<i\neq j\leq k} d(FFT(\mathbf{x}^i), FFT(\mathbf{x}^j)) \quad (5)$$

$$s.t. \forall i, \triangle_1 < \left|\mathbf{x}^i\right| < \triangle_2; \sum_i \left|\mathbf{x}^i\right| = \boldsymbol{X}^i \in \mathbb{R}^{D\times T} \quad (6)$$

where $d(\cdot)$ is a distance metric, k_0 is the hyperparameter to avoid over-splitting, and $\triangle_1, \triangle_2$ are predefined parameters to avoid trivial solutions (e.g., extremely small or large values might not effectively capture the distribution information). T is the length of time-series and D is the dimension of features. The metric $d(\cdot)$ can be any distribution distance calculations such as Euclidean, JS-divergence, and KL-divergence. By searching k, the corresponding periods, and then applying this approach to each feature dimension d, we can find the most dissimilar period pairs in the feature domain for the stock. By performing similar operations on each stock, we obtain the enhanced stock graph node embedding matrix $\boldsymbol{X}'$.

It's important to notice that in one stock series, the sequences derived from FFT transformation in both rising and falling markets exhibit similarity, details please refer the experiments. The distinguishing factor lies in the fluctuating market. Therefore, we will generate three sets of results (rising, falling, and fluctuating) to serve as enhanced representations $\mathbf{x}'^i$ of one feature for stock i.

4.2 Contrastive Constrained Training

In previous stock prediction methods, company relations are mainly used to define the edge structure of stock graphs for information propagation (Feng et al., 2019; Kim et al., 2019). Consideration of these relations as a constraint has been largely ignored during the training process. Therefore, we propose a Contrastive Constrained Training (CCT) module that integrates company relations to learn a constrained GNN encoder through contrastive learning. We will begin by introducing the adaptive constrained augmentation, followed by the contrastive learning framework, as the two components of the CCT module.

4.2.1 Adaptive Constrained Augmentation

Contrastive learning methods, which aim to maximize agreement between views, strive to learn representations consistent despite perturbations introduced by augmentation techniques. Within the CCT module, we propose an adaptive enhancement scheme based on company relation constraints. The aim is designed to disrupt less important links and features while protecting the important constraint structure and properties.

In this step, contrary to previous methods, we incorporate company relations as constraints during training, resulting in a reevaluation of the former stock graph. The probability of the edge between node i, j remains in the augmentation step $P\{(i,j) \in \mathcal{E}\}$ is calculated as:

$$1 - min\left(\frac{\log w_{max} - \log w}{\log w_{max} - \log w_{\mu}} \mathrm{e} \cdot p_e, \quad p_\tau\right) \quad (7)$$

where w is the average of the degree centrality (Newman, 2018) between node i and j. We first calculate w for all edges and then make a max-mean normalization using the max and mean (μ). To balance the probabilities, we then take the logarithm of the result. p_e and p_τ are hyperparameters to control the overall probability of removing edges and the cut-off threshold.

The key approach presented in CCT is that the degree centrality calculation method is based on the company relation of the stocks, rather than the edges within the stock graph. That is, given the stock relation encodings $\boldsymbol{A}$ and each kind of relation $\boldsymbol{A}_{ij}^{relation} \in \mathcal{A}$ for node pair (i,j) we must adhere to the constraint that there must be at least one type of relation that exists $\boldsymbol{A}_{ij}^{\text{relation}} > 0$. With this constraint, we can calculate w using the following method.

$$\frac{1}{2}\left(\sum_{(i,j)\in\mathcal{E}} \boldsymbol{A}_{ji}^{\text{relation}} + \sum_{(i,j)\in\mathcal{E}} \boldsymbol{A}_{ij}^{\text{relation}}\right) \quad (8)$$

After calculating the edge remaining probability, we generate two views for the input graph $\mathcal{G}$ to perform contrastive learning.

4.2.2 Contrastive Learning Framework

The proposed CCT module follows the common graph contrastive learning paradigm, in which the model aims to maximize the alignment of representations across various perspectives (Hassani and Khasahmadi, 2020; Zhu et al., 2021). We initially create two different views of the graph $\mathcal{G}$, by applying adaptive constrained augmentation to the input. Subsequently, we utilize a contrastive objective function that ensures the encoded embeddings of each node in these two distinct views align with each other, while also being distinguishable from embeddings of other nodes.

Specifically, during each training epoch, the CCT module begins by randomly selecting two data augmentation functions from $q(\cdot)$, which is a common operation to obtain two views $v_a \sim q(G)$, $v_b \sim q(G)$, then, node representations z_a and z_b are obtained from a GNN encoder $g(\cdot)$. For a given node i, its embedding z_a^i generated in view v_a serves as the anchor, while the embedding of it generated in the other view z_b^i forms the positive sample, and the remaining embeddings in the two views are considered as negative samples. Following (Zhu et al., 2021), we define the pairwise objective $\ell(z_a^i, z_b^i)$ for each positive pair (z_a^i, z_b^i) as :

$$\log \frac{e^{\theta(z_a^i, z_b^i)/\tau}}{e^{\theta(z_a^i, z_b^i)/\tau} + \sum e^{\theta(z_a^i, z_b^j)/\tau} + \sum e^{\theta(z_a^i, z_a^j)/\tau}} (i \neq j) \quad (9)$$

where τ is a temperature parameter, and $e^{\theta(z_a^i, z_b^i)/\tau}$ represent positive pair. $\sum e^{\theta(z_a^i, z_b^j)/\tau}, \sum e^{\theta(z_a^i, z_a^j)/\tau}$ mean inter-view negative pairs and intra-view negative pairs. $\theta(z_a, z_b) = s(\phi(z_a), \phi(z_b))$, in which, $s(\cdot)$ refers to the cosine similarity and $\phi(\cdot)$ is a non-linear projection, implemented as a two-layer perception model. Moreover, the overall objective $\mathcal{L}_{cl}$ for the CCT module to be maximized uses the average for all positive pairs, since the two views are symmetric.

$$\mathcal{L}_{cl} = \frac{1}{2N} \sum_{i=1}^{N} (\ell(z_a^i, z_b^i) + \ell(z_b^i, z_a^i)) \quad (10)$$

In this way, we use the number of relationships as a basis for measuring the possibility of the existence of a certain edge, thereby achieving adaptive data augmentation with constraints, which is then trained together with the dynamic graph evolution.

4.3 Dynamic Graph Evolution

Using dynamic graphs and node embedding matrices, we will now explain how to acquire the dynamic latent representations of nodes. This process consists of two components: the graph convolutional network and the weight evolution.

Based on GCN (Kipf and Welling, 2017), we propose the following propagation rules:

$$\boldsymbol{H}_t^{(l+1)} = \sigma(\widehat{\boldsymbol{A}}_t \boldsymbol{H}_t^{(l)} \boldsymbol{W}_t^{(l)}) \quad (11)$$

Here, at time t, the l-th layer takes the adjacency matrix $\boldsymbol{A}_t$ and the node embedding matrix $\boldsymbol{H}_t^{(l)}$ as input, and outputs $\boldsymbol{H}_t^{(l+1)}$ using a layer-specific trainable weight matrix $\boldsymbol{W}_t^{(l)}$ to update. Specifically, $\sigma(\cdot)$ denotes an activation function.

$$\widehat{\boldsymbol{A}} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}, \widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}, \widetilde{\boldsymbol{D}} = diag\left(\sum_j \widetilde{\boldsymbol{A}}_{ij}\right) \quad (12)$$

To acquire the dynamic latent representations, the trainable weight matrix $\boldsymbol{W}_t^{(l)}$, we follow the methodology outlined in (Pareja et al., 2020). $\boldsymbol{W}_t^{(l)}$ is treated as the hidden state of the dynamical system updated by any RNN-based model $f(\cdot)$, such as LSTM and GRU, with a prediction loss $\mathcal{L}_{pred}$.

$$\boldsymbol{W}_t^{(l)} = f(\boldsymbol{H}_t^{(l)}, \boldsymbol{W}_{t-1}^{(l)}) \quad (13)$$

The final prediction result will be derived from the output of $\boldsymbol{W}$ at the final time step. This result is binarized and then mapped through a $softmax$ function to predict each stock. The final form of the loss function is shown below, where λ is a trade-off parameter.

$$\mathcal{L} = \mathcal{L}_{pred} + \lambda \mathcal{L}_{cl} \quad (14)$$

5 EXPERIMENT

In this section, we describe the experimental setup used to evaluate the effectiveness of our proposed method. Experimental results demonstrate the effectiveness of our DGRCL framework in comparison with different backbones and datasets. We specifically aim to answer the following questions: **(RQ1)**: How effective is the proposed DGRCL framework for the graph classification task? **(RQ2)**: How effective are the individual modules proposed in DGRCL? **(RQ3)**: How does sensitivity to hyperparameters affect the performance of the proposed model?

5.1 Dataset

We closely follow the experimental setup outlined in Feng et al. (2019), using two major US stock markets.

Table 1: Data statistics of NASDAQ and NYSE.

Item	NASDAQ dataset			NYSE dataset		
	Train	Valid	Test	Train	Valid	Test
Time Span	01/02/2013-12/08/2017			01/02/2013-12/08/2017		
Valid Trading Days	611	94	236	611	94	236
Data Points	5 (OHLCV)			5 (OHLCV)		
Nodes	1026			1737		
Edges	164			255		
Relation Types	112			130		

Many studies have utilized this same dataset to evaluate their stock forecasting models (Kim et al., 2019; Sawhney et al., 2021a; Hsu et al., 2021; You et al., 2024a,b), and to ensure easier comparison with the existing literature, we also adopt this dataset in our experiments.

Dataset statistics are summarized in Table 1. We follow the setup of (Pareja et al., 2020) and divide the two datasets into training, validation, and testing sets in the proportions of 0.65, 0.1, and 0.25, respectively. This results in 611, 94, and 236 effective trading days for each set. OHLCV are daily values for each stock. Edges number is calculated by (4) with a certain number of nodes. Recognizing that stocks within the same industry tend to be similarly affected by industry-wide prospects, we gather data on the sector-industry relationships among stocks. For two datasets, we apply the sector-industry relations as Relation Types in Table 1, for more details, please refer to (Feng et al., 2019). The dataset and source code are available in github.[1]

Each stock market dataset spans four years, exceeding one thousand time points, with more than one thousand traded stocks in each market. Compared to the datasets used in the baseline models, the datasets employed in DGRCL are characterized by their long duration, multiple nodes, and dynamic temporal graph changes. This makes financial datasets particularly well-suited for studying long-term information propagation in GNNs, as compared to other datasets.

5.2 Model Setting

All experiments are performed on a Nvidia GeForce RTX 3090 graphic card, CUDA version 11.5. The lookback window δ applied in $DTW(\cdot)$ is 20. In general, each month consists of approximately 22.5 working days. The input signals $\mathbf{x}_t^i$ for DTW is the volume volatile, by calculating vol^t/vol^{t-1}. The RNN-based model $f(\cdot)$ is a two-layer LSTM, following the methodology outlined in Pareja et al. (2020). Three hyperparameters in the CCT module are τ, p_e, and p_τ. Following the work of Zhu et al. (2021), we set τ as 0.4. For the other two, we train the model using p_e and p_τ from 0.1 to 0.9 with increments of 0.2. The activation function $\sigma(\cdot)$ is set to $ReLU(\cdot)$. Based on the method outlined in Liu et al. (2022b), we control the trade-off parameter λ to bring $\mathcal{L}_{pred}$ and $\mathcal{L}_{cl}$ to the same order of magnitude. The empirical value of λ obtained from multiple experiments is 0.1.

5.3 Baseline

We consider state-of-the-art dynamic graph node classification models as baseline methods to learn the representations of nodes in a graph, including Graph WaveNet (Wu et al., 2019), MTGODE (Jin et al., 2022), STGCL (Liu et al., 2022b), and Evolvegcn (Pareja et al., 2020).

- Graph WaveNet (Wu et al., 2019): A spatiotemporal graph modeling approach that captures spatial-temporal dependencies across multiple time series by integrating graph convolution with dilated causal convolution. Two datasets are used, the METR-LA dataset spans four months of traffic speed data from 207 sensors, while the PEMS-BAY dataset covers six months from 325 sensors, both aggregated into 5-minute intervals.
- MTGODE (Jin et al., 2022): A continuous model that forecasts multivariate time series by integrating dynamic graph neural Ordinary Differential Equations to unify spatial-temporal message passing and enhance latent spatial-temporal dynamics. The evaluation involves five benchmark datasets: three conventional time series datasets (Electricity, Solar-Energy, Traffic) and two traffic datasets (Metr-La, Pems-Bay) with predefined graph structures and specific time spans.
- STGCL (Liu et al., 2022b): A study that integrates contrastive learning into spatio-temporal graph (STG) forecasting, leveraging novel node and graph-level contrastive tasks to mitigate data scarcity, enhancing performance through joint learning schemes and strategic data augmentations. The model is tested using two traffic benchmarks, PEMS-04 and PEMS-08, with aggregated 5-minute traffic flow, average speed, and occupancy data, applying Z-score normalization and Gaussian kernel adjacency matrices.
- EvolveGCN (Pareja et al., 2020): An approach to dynamic graph representation learning that adapts graph convolutional networks over time by evolving GCN parameters with recurrent neural networks, without relying on node embeddings. The benchmark datasets include the Stochastic Block Model (SBM) for simulating community struc-

[1] https://github.com/PEIYUNHUA/DGRCL

Table 2: Performance of proposed DGRCL and other baselines on next trading day stock trend classification over the test period. Bold denotes the best result.

Methods	NASDAQ dataset			NYSE dataset		
	Accuracy(%)	F1 score	MCC($\times 10^{-3}$)	Accuracy(%)	F1 score	MCC($\times 10^{-3}$)
Graph WaveNet (Wu et al., 2019)	50.94±0.15	60.51±1.10	5.50±1.00	50.86±0.13	57.28±1.86	11.12±2.21
MTGODE (Jin et al., 2022)	51.41±0.15	62.41±4.24	5.92±2.61	52.35±0.17	62.73±1.72	20.62±5.76
STGCL (Liu et al., 2022b)	50.56±0.08	57.81±1.19	**9.71±1.15**	51.21±0.05	67.47±0.07	22.33±1.53
EvolveGCN (Pareja et al., 2020)	50.19±0.02	56.94±0.21	4.41±0.76	51.14±0.15	66.85±0.37	14.41±0.45
DGRCL (Ours)	**53.06±0.18**	**66.53±2.74**	9.32±1.53	**54.07±0.20**	**67.53±0.19**	**27.51±5.52**

Table 3: Ablation study on the two datasets. We replace the Embedding Enhancement module and Contrastive Constrained Training module in DGRCL with their discrete implementations, denoted w/o EE and w/o CCT. We further remove all proposed modules except dynamic graphs for training as w/o EE & CCT.

Methods	NASDAQ dataset			NYSE dataset		
	Accuracy(%)	F1 score	MCC($\times 10^{-3}$)	Accuracy(%)	F1 score	MCC($\times 10^{-3}$)
DGRCL	53.06±0.18	66.53±2.74	9.32±1.53	54.07±0.20	67.53±0.19	27.51±5.52
w/o EE	51.45±0.03	62.48±1.19	4.69±0.02	52.77±0.05	62.95±1.40	21.88±2.35
w/o CCT	52.69±0.32	65.61±0.22	5.78±1.07	53.87±0.11	67.34±0.18	23.12±1.77
w/o EE & CCT	51.08±0.12	57.43±1.22	2.91±0.26	51.65±0.21	64.12±0.19	7.61±2.11

tures, Bitcoin OTC (BC-OTC) and Bitcoin Alpha (BC-Alpha) for rating predictions, UC Irvine messages (UCI) for link prediction, and several other networks for various predictive tasks.

5.4 Evaluation Metric

Following the approaches taken in previous studies (You et al., 2024a; Kim et al., 2019; Deng et al., 2019; Sawhney et al., 2020), we assess our result using Classification Accuracy, F1 score, and Matthews Correlation Coefficient (MCC) to evaluate model performance. For all three metrics, higher values indicate better model performance. We record the time taken for each training epoch of the models in seconds.

5.5 Evaluation Result

5.5.1 RQ1. Performance Comparison

The experimental results in Table 2 show that our model outperforms all other baseline models in terms of accuracy and F1 score, and is also comparable to the baseline models in terms of MCC. For the NASDAQ dataset, DGRCL achieves the highest accuracy at 53.06%, surpassing the second-best MTGODE by 1.65%. Additionally, DGRCL attains the top F1 score of 66.53% and a strong MCC value of 9.32, nearly matching the best MCC of 9.71. On the NYSE dataset, DGRCL similarly outperforms other methods, with an accuracy of 54.07%, an F1 score of 67.53%, and a substantial MCC of 27.51, marking the highest recorded metrics across all models. The notable discrepancy where MTGODE exhibits a high F1 score but a lower MCC indicates the model may handle accuracy and recall well for the majority class but struggles with the minority class, affecting its overall prediction quality. A similar finding is also shown in the performance of EvolveGCN on the NYSE dataset.

Moreover, we observe that the accuracy of all baseline methods is not very high, being only slightly better than 50%, which deviates significantly from the experimental results described in their respective papers. A reasonable explanation for this inconsistency is that the nodes and the temporal span of their datasets are relatively small. As a result, their models may be unable to effectively learn the representations of large-scale graph models over long time series. For instance, in the datasets used by Graph WaveNet and STGCL, the number of nodes is 207 and 325, respectively, which is far smaller than the 1026 and 1737 nodes in our dataset. Similarly, in the dataset used by MTGODE, the largest sample size is 52,560, which is much smaller than the smallest sample size in our study 626,886 (1026×611).

5.5.2 RQ2. Ablation Study

To verify the effectiveness of each component of DGRCL, we conducted ablation studies on the two datasets (see Table 3). The complete DGRCL model incorporates EE and CCT, exhibiting the highest performance in all metrics. When both EE and CCT are removed (w/o EE & CCT), DGRCL has the most sig-

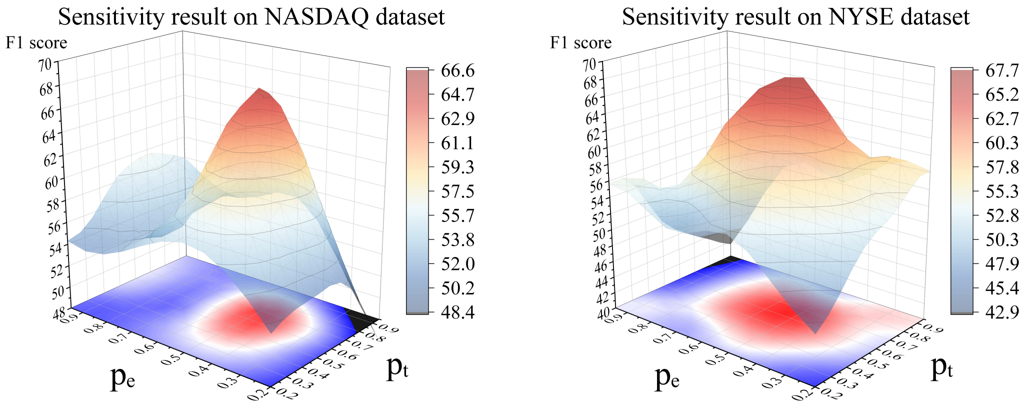

Figure 2: Sensitivity study of the overall probability of removing edges p_e and the cut-off threshold p_τ via F1 score.

Table 4: Run-time complexity of the experiments.

Model	Dataset	Epoch Training Time (s)	Model	Dataset	Epoch Training Time (s)
Graph WaveNet	NASDAQ	1.63	DGRCL (Ours) w/o EE	NASDAQ	406.53
	NYSE	3.14		NYSE	485.38
MTGODE	NASDAQ	13.22	DGRCL (Ours) w/o CCT	NASDAQ	42.58
	NYSE	29.96		NYSE	96.43
STGCL	NASDAQ	2.12	DGRCL (Ours) w/o EE & CCT	NASDAQ	41.79
	NYSE	3.17		NYSE	80.06
EvolveGCN	NASDAQ	24.63	DGRCL (Ours)	NASDAQ	414.05
	NYSE	38.25		NYSE	505.54

nificant performance drop, with accuracy falling to 51.08% and 51.65%, the F1 score to 57.43 and 64.12, and the MCC to 2.91 and 7.61 in the NASDAQ and NYSE datasets respectively.

From the result we can find that, by incorporating the EE module, DGRCL experiences a significant improvement, nearly doubling the MCC score on both datasets. This indicates that the EE module enables the model to learn more effective latent features of the time series, thereby enhancing its predictive performance. Another observation is that the CCT module alone does not result in a significant improvement in model performance and is less effective compared to adding the EE module alone. Nonetheless, when both modules are combined, they greatly enhance the overall performance of the model. This suggests that the model achieves optimal predictive results by learning the dynamic temporal relationships and the static relationships in the graph data.

5.5.3 RQ3. Hyperparameter Sensitivity

We explore the sensitivity of two important hyperparameters: the overall probability of removing edges p_e and the cutoff threshold p_τ. These two hyperparameters directly affect the performance of the CCT module and further impact the overall performance of the DGRCL. Here, we use the global evaluation metric F1 score to reflect the overall performance of different combinations of these hyperparameters. Results are presented in Fig. 2. We make the following observations.

When both p_e and p_τ are set at their maximum or minimum values simultaneously, overall performance is poor, with the F1 score in both datasets falling below 50. This indicates that finding a balance between these two hyperparameters is crucial to improving the model's predictive performance. The second finding is that p_e is more important than p_τ, as it determines the dynamic lower bound in Eq. (7). If p_τ is set very low, $P\{(i,j)\in\mathcal{E}\}$ becomes a fixed value equal to $1-p_\tau$. This indicates that all edges have the same existence probability, resulting in random remaining. Another observation is that, Compared to the NASDAQ dataset, the NYSE dataset has a larger optimal hyperparameter space, making it easier to find local optima during model training. Previous financial papers (Lian et al., 2022; Jiang et al., 2011) also support this finding, noting that the NASDAQ market is more volatile, leading to a smaller optimal hyperparameter space.

5.5.4 Complexity

Table 4 compares the training times per epoch (in seconds) of various models on the NASDAQ and NYSE datasets. Our model, DGRCL, generally has significantly higher training times compared to other models. For example, Graph WaveNet and STGCL require only 1.63s and 2.12s per epoch on NASDAQ, respectively, while DGRCL without embedding enhancement (EE) and contrastive constrained training (CCT) takes 41.79s. The full DGRCL model, which includes these enhancements, requires 414.05s on NASDAQ and 505.54s on NYSE.

Notably, adding the CCT unit significantly increases the training time, accounting for around 90% of the total run-time, which indicates that the graph contrastive learning component is the main source of computational overhead. However, as shown in the comparison between Table 2 and Table 3, even without the CCT module, DGRCL still outperforms the baselines in terms of accuracy. This demonstrates that our framework performs well without relying on the computationally expensive CCT module. While DGRCL offers improved prediction performance, this high computation cost, particularly from the CCT module, suggests a need for further optimization to balance performance and efficiency, especially in large-scale applications.

6 CONCLUSION

In this paper, we propose the Dynamic Graph Representation with Contrastive Learning (DGRCL) framework, which unifies dynamic temporal graph learning and static relational graph learning through a contrastive learning paradigm. Within DGRCL, both the Embedding Enhancement (EE) and Contrastive Constrained Training (CCT) modules complement each other: the EE module dynamically captures evolving stock market trends, while the CCT module leverages static inter-stock relationships to refine feature representations through contrastive learning. This dual approach enhances the model's ability to predict stock movements with higher accuracy. Additionally, we utilize objective laws to construct dynamic graph structures. Meanwhile, we integrate a Fourier transform-based technique into the EE module to effectively address the challenges posed by distribution shifts in time series data. Ultimately, we validate the effectiveness of DGRCL through extensive experiments on two real-world datasets, demonstrating that DGRCL consistently outperforms four baseline models.

For future work, we will also refine the CCT module to better distinguish between static and dynamic graph features. We will perform comparisons with additional baselines on financial market datasets from various countries (such as the Shanghai Stock Exchange in China) to further demonstrate the superiority of the DGRCL framework as a long-term time-series forecasting model for financial predictions. In addition, to make our framework more efficient, future work will focus on optimizing the graph contrastive learning part to reduce its computational overhead and enhance the overall performance of the DGRCL framework.

ACKNOWLEDGEMENTS

This work was supported by UKRI EPSRC Grant No. EP/Y028392/1: AI for Collective Intelligence (AI4CI), and Innovate UK Project No. 10094067: Stratlib.AI - A Trusted Machine Learning Platform for Asset and Credit managers. The authors have no conflicts of interest to declare.

REFERENCES

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Bloomfield, P. (2004). *Fourier analysis of time series: An introduction*. John Wiley & Sons.

Chen, S. and Ge, L. (2019). Exploring the attention mechanism in LSTM-based Hong Kong stock price movement prediction. *Quantitative Finance*, 19(9):1507–1515.

Cochran, W. T., Cooley, J. W., Favin, D. L., Helms, H. D., Kaenel, R. A., Lang, W. W., Maling, G. C., Nelson, D. E., Rader, C. M., and Welch, P. D. (1967). What is the fast Fourier transform? *Proceedings of the IEEE*, 55(10):1664–1674.

Deng, S., Zhang, N., Zhang, W., Chen, J., Pan, J. Z., and Chen, H. (2019). Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 678–685.

Du, Y., Wang, J., Feng, W., Pan, S., Qin, T., Xu, R., and Wang, C. (2021). AdaRNN: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 402–411.

Feng, F., He, X., Wang, X., Luo, C., Liu, Y., and Chua, T.-S. (2019). Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–30.

Gao, J., Ying, X., Xu, C., Wang, J., Zhang, S., and Li, Z. (2021). Graph-based stock recommendation by time-aware relational attention network. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1):1–21.

Hassani, K. and Khasahmadi, A. H. (2020). Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126.

Hsu, Y.-L., Tsai, Y.-C., and Li, C.-T. (2021). FinGAT: Financial graph attention networks for recommending top-k profitable stocks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):469–481.

Huynh, T. T., Nguyen, M. H., Nguyen, T. T., Nguyen, P. L., Weidlich, M., Nguyen, Q. V. H., and Aberer, K. (2023). Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 850–858.

Jaynes, E. T. (1982). On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70(9):939–952.

Jeong, Y.-S., Jeong, M. K., and Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240.

Jiang, C. X., Kim, J.-C., and Wood, R. A. (2011). A comparison of volatility and bid–ask spread for NASDAQ and NYSE after decimalization. *Applied Economics*, 43(10):1227–1239.

Jin, M., Zheng, Y., Li, Y.-F., Chen, S., Yang, B., and Pan, S. (2022). Multivariate time series forecasting with dynamic graph neural ODEs. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180.

Kim, R., So, C. H., Jeong, M., Lee, S., Kim, J., and Kang, J. (2019). HATS: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999*.

Kim, T., Kim, J., Tae, Y., Park, C., Choi, J.-H., and Choo, J. (2021). Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Li, C., Song, D., and Tao, D. (2019). Multi-task recurrent neural networks and higher-order Markov random fields for stock price movement prediction: Multi-task RNN and higher-order MRFs for stock price classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1141–1151.

Lian, Y.-M., Jhong, Y.-J., Wang, P.-H., and Chen, W.-M. (2022). An empirical examination of VIX market fluctuations. *Advances in Management and Applied Economics*, 12(4):1–6.

Liu, C., Yan, J., Guo, F., and Guo, M. (2022a). Forecasting the market with machine learning algorithms: an application of NMC-BERT-LSTM-DQN-X algorithm in quantitative trading. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4):1–22.

Liu, X., Liang, Y., Huang, C., Zheng, Y., Hooi, B., and Zimmermann, R. (2022b). When do contrastive learning signals help spatio-temporal graph forecasting? In *Proceedings of the 30th international conference on advances in geographic information systems*, pages 1–12.

Liu, Y., Jin, M., Pan, S., Zhou, C., Zheng, Y., Xia, F., and Philip, S. Y. (2022c). Graph self-supervised learning: A survey. *IEEE transactions on knowledge and data engineering*, 35(6):5879–5900.

Matsunaga, D., Suzumura, T., and Takahashi, T. (2019). Exploring graph neural networks for stock market predictions with rolling window analysis. In *Proceedings of the NeurIPS Workshop on Machine Learning for Finance (NeurIPS 2019)*. Published in NeurIPS 2019 Workshop on Machine Learning for Finance.

Newman, M. (2018). *Networks*. Oxford University Press.

Newman, M. E. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary physics*, 46(5):323–351.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370.

Sawhney, R., Agarwal, S., Wadhwa, A., Derr, T., and Shah, R. R. (2021a). Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 497–504.

Sawhney, R., Agarwal, S., Wadhwa, A., and Shah, R. (2020). Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8415–8426.

Sawhney, R., Agarwal, S., Wadhwa, A., and Shah, R. (2021b). Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proceedings of the Web Conference 2021*, pages 11–22.

Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. (2019). Deep graph infomax. *ICLR (Poster)*, 2(3):4.

Wang, H., Wang, T., Li, S., Zheng, J., Guan, S., and Chen, W. (2022). Adaptive long-short pattern transformer for stock investment selection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3970–3977.

Wang, J., Sun, T., Liu, B., Cao, Y., and Zhu, H. (2019). CLVSA: A convolutional LSTM based variational sequence-to-sequence model with attention for predicting trends of financial markets. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3705–3711. International Joint Conferences on Artificial Intelligence Organization.

Wu, Z., Pan, S., Long, G., Jiang, J., and Zhang, C. (2019). Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1907–1913.

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

Yin, T., Liu, C., Ding, F., Feng, Z., Yuan, B., and Zhang, N. (2022). Graph-based stock correlation and prediction for high-frequency trading systems. *Pattern Recognition*, 122:108209.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. (2020). Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823.

You, Z., Shi, Z., Bo, H., Cartlidge, J., Zhang, L., and Ge, Y. (2024a). DGDNN: Decoupled graph diffusion neural network for stock movement prediction. In *Proceedings of the 49th IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pages 6545–6549.

You, Z., Zhang, P., Zheng, J., and Cartlidge, J. (2024b). Multi-relational graph diffusion neural network with parallel retention for stock trends classification. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6545–6549. IEEE.

Zhang, H., Lin, S., Liu, W., Zhou, P., Tang, J., Liang, X., and Xing, E. P. (2023). Iterative graph self-distillation. *IEEE Transactions on Knowledge and Data Engineering*, 36:1161–1169.

Zhao, Y., Du, H., Liu, Y., Wei, S., Chen, X., Zhuang, F., Li, Q., and Kou, G. (2022). Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, F., Zhou, H.-m., Yang, Z., and Yang, L. (2019). Emd2fnn: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. *Expert Systems with Applications*, 115:136–151.

Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080.