# Finformer: A Static-dynamic Spatiotemporal Framework for Stock Trend Prediction

Yi Zu[†], Jiacong Mi[†], Lingning Song[‡], Shan Lu[§] and Jieyue He[†*]
[†]School of Computer Science and Engineering,
Key Lab of Computer Network and Information Integration, MOE, Southeast University, Nanjing, China
[‡]School of Software Engineering, Southeast University, Nanjing, China
[§]Nanjing Fenghuo Tiandi Communication Technology Co., Ltd, Nanjing, China
{yizu, mijiacong, lingning_s}@seu.edu.cn, bfcat.cn@gmail.com, jieyuehe@seu.edu.cn

***Abstract*—The core of quantitative investment lies in predicting future trends in stock prices. The future trend of a stock is closely related to the industry it belongs to and its relationship with other stocks. Although some research has focused on stock trend prediction in recent years, most studies have only considered the stock's own time series feature, neglecting the spatial features between stocks. Some research has incorporated spatial information, but typically only considered predefined static relationships. At the same time, capturing dynamic spatial information in the market has been a long-standing challenge. Thus, we propose a spatio-temporal model, Finformer, in order to go beyond traditional time series models. We designed a sparse static-dynamic transformer to capture dynamic market spatial information as it changes over time and combined predefined relationships to extract highly correlated spatial features in the stock market. To effectively integrate spatial and temporal features, we introduced an adaptive spatio-temporal fusion module that dynamically fuses spatio-temporal features based on market conditions at different periods. Experiments on two real-world stock market datasets show that our proposed model outperforms the state-of-the-art baselines in the signal-based and portfolio-based metrics, which are widely concerned in the financial field. Ablation study and hyper-parameter study further reveal the effectiveness of each module in the model and the impact of hyper-parameters. The code will be made publicly available.[1]**

***Index Terms*—computational finance, stock trend prediction, transformer, deep learning**

## I. Introduction

Quantitative investment has gradually become the mainstream investment approach today, whose primary aim is to identify a group of potentially profitable stocks to create a portfolio, allocate capital based on predicted stock trends dynamically, and balance the trade-off between profit and risk. In this context, the prediction of stock trends, a vital facet of quantitative investment, has been garnering significant interest in recent years.

The Efficient Market Hypothesis (EMH) [1] postulates that in an efficient market, all available information is already reflected in stock prices. Consequently, the excess returns cannot be achieved merely by analyzing historical data or publicly accessible information. Contrary to this, empirical studies [2] have evidenced that under specific circumstances, markets are not entirely efficient, and a certain degree of predictability exists in stock trends. This opens up possibilities for more accurate stock trend predictions.

In the domain of stock trend prediction, prior works have concentrated on the historical time series data of stocks, using historical performance of each stock to predict its future trend. Although the analysis of time series data [3]–[6] has been proven effective in predicting stock trends, this approach treats each stock as a standalone entity, potentially overlooking the interconnectedness within the stock market and thus constraining the accuracy of stock trend predictions.

The spatial information in the stock market encompasses the interrelationships among different stocks. Depending on the degree of temporal influence, these relationships can be categorized into static relationships and dynamic relationships. Static relationships refer to the objective connections that exist in the real world among stocks (e.g. industry associations and competitive relationships). Dynamic relationships, on the other hand, pertain to interactions that gradually evolve over time, and they are more influenced by short-term market fluctuations and investor sentiment. This includes factors such as short-term market trends, trading activity, and market sentiment. Both of these types of relationships have a comprehensive impact on the stock market, making it crucial to incorporate the spatial feature of stocks for improving the accuracy of stock trend forecasting [7], [8].

Some contemporary studies incorporate the spatial relationships within the market into models, but they often fall short by either neglecting static relationships [9], [10] or dynamic relationships [8], [11], [12], thus resulting in the inability to fully capture the characteristics of relationships between stocks. Moreover, some models that use static relationships [8], [12] need to satisfy the assumption that the number of stocks traded in the daily market is fixed, which makes it difficult to adapt to real market conditions, and some models that used dynamic relations [10], [13] often use statistical methods such as Pearson correlation coefficient or similarity to measure strength of relationships, but cannot dynamically select important information based on the sequence's context.

Therefore, to more effectively integrate spatial information into stock trend prediction models, our study seeks to address

*Corresponding author. This work was supported by National Key R&D Program of China (2019YFC1711000) and Collaborative Innovation Center of Novel Software Technology and Industrialization.

[1]https://github.com/yizu14/Finformer

979-8-3503-2445-7/23/$31.00 ©2023 IEEE

three central questions: 1) How can we capture the dynamic relationships within the stock market that evolve over time effectively and automatically? 2) How can we ensure the sparsity of these dynamic relationships, i.e., maintain the significant relationships while eliminating the redundant ones? 3) How can we effectively fuse temporal features with newly introduced spatial features?

In this paper, we confront the aforementioned challenges by proposing a spatiotemporal framework that integrates both static and dynamic spatial relationships within the stock market to augment the precision of future stock trend predictions. We model the spatial relationships in the stock market by mining sparse dynamic relationships and integrating with existing static relationships. To optimize the integration of these two types of features, we introduce the Gated Fusion mechanism [14], which diverges from simple feature concatenation by adjusting to fluctuations in the influence of temporal and spatial features on stock trends during different periods and dynamically moderating the fusion process.

To validate the efficacy of our model, we designed multiple experiments on two datasets: CSI300 and CSI100, the most representative datasets of China's stock market, spanning from January 1, 2010, to December 31, 2022. The experimental results demonstrate that our model surpasses the state-of-the-art baseline models in performance. Moreover, in ablation study, we explored the performance contributions of each module. In hyper-parameter sensitivity study, we explored the impact of significant hyper-parameters.

In summary, our contribution can be concluded as

- We proposed a Transformer-based spatiotemporal model for predicting stock trends, integrating temporal and spatial features adaptively for more accurate forecasts.
- We designed a Static-Dynamic Attention mechanism that mines the sparse potential dynamic relationships between stocks by analyzing historical time series data, and forms spatial embedding of stocks by integrating predefined static industry relationships.
- Compared to the current state-of-the-art baseline models, our proposed Finformer model demonstrates significant improvements in performance in the real stock market.

## II. Related Works

The related work in the field of stock prediction includes using historical time series data of stocks to predict future trends by analyzing their historical performance, and using relationship information between stocks to predict future trends by measuring the mutual influence between stocks.

### A. Time Series Forecasting

Previous works use the time series price data of stocks to depict their historical performance, indirectly inferring future stock trends. Classical stock trend prediction methods in technical analysis includes [15] such as ARIMA [16] and machine learning methods SVR [17]. Although these linear methods exhibit some efficacy in trend prediction, the nonlinear nature of stock prices constrains their performance.

Deep learning, with its remarkable capability for nonlinear modeling and learning effective embeddings from massive historical data, has been increasingly employed in stock trend prediction tasks in recent years. During the early adoption phase of deep learning in stock trend prediction, variations based on RNN [18], such as LSTM [19] and GRU [20], were commonly employed. And subsequent work [3], [21] introduced improvements. For instance, [3] designed an Adv-ALSTM for predicting stock trends, integrating adversarial neural networks based on [4].

In addition, historical price data from multiple stocks can reveal current market trading patterns. [22] used discrete Fourier transform to decompose the hidden state and capture multi-frequency trading patterns, and [6], [23] developed lightweight modules to predict trends based on market trading patterns.

Recently, the Transformer [24], [25] has been demonstrated to outperform in capturing long-term temporal features. However, models solely considering temporal data often overlook the spatial relationships within the stock market, resulting in a deficiency of spatial features in stocks and subsequently constraining the performance of the model.

### B. Correlation-Based Stock Forecasting

Furthermore, research has revealed that the relationships within the stock market play a significant role in stock trend prediction. As a result, the Graph Neural Network (GNN) [26] has been introduced to this domain due to its ability in relation mining. Works such as [27], [28] model the time series relationships of stock prices and predict future trends by identifying patterns of price change. For instance, [8] explores the interaction between inter-sector and intra-sector relationships, basing predictions on the sector to which the stock belongs. Meanwhile, [29], [30] construct a hypergraph based on a variety of relationships to capture cross-domain interactions between stocks. Other works, such as [9], [10], attempt to uncover dynamic relationships over time in the market in order to discern dynamic interactions between stocks. However, relying solely on static or dynamic relationships may not fully depict the complex relationships within the stock market, leading to a lack of crucial relationships.

To better integrate static and dynamic relationships, [31] discovers that a single predefined relationship could not effectively characterize the spatial relationships within the stock market. Consequently, it introduced hidden relationships to explore the dynamic interplay amongst stocks more profoundly. However, when it comes to determining the number of dynamic relationships, it relies on manual settings, which may result in imprecise dynamic relationships.

Moreover, some studies try to incorporate news [32]–[34], social media comments [35]–[37], public statements [38], and other textual information to discern stock relationships. These methods capture market sentiment and investment trends through real-world events, assessing the degree to which such events influence stock trends. However, they may be vulnerable to the irregularities and sparsity inherent in event data.

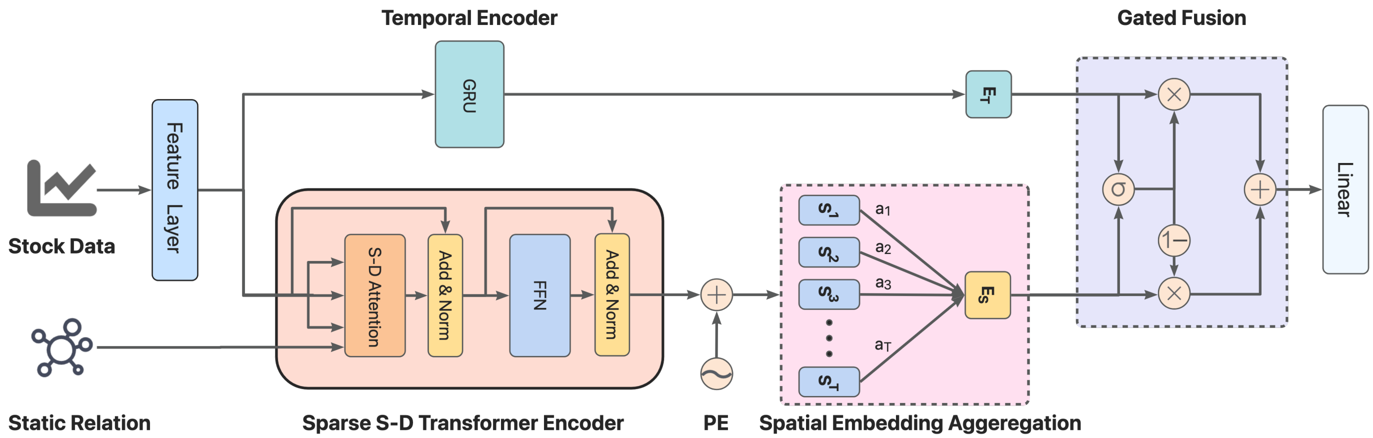



Fig. 1. Finformer architecture. Composed by Temporal Encoder, Sparse S-D Transformer Encoder for spatial embedding module, and Gated Fusion module for features integration.

## III. Problem Formulation

We formulate the stock trend prediction problem as a regression task. Denote $x_i^d = [stock_i^1, stock_i^2, ..., stock_i^T] \in \mathbb{R}^{1\times T\times F}$ represent the historical data of $i$-th stock in the market of $d$-th day in the training set, where $T$ is the size of look-back window and $F$ is feature dimension of the stock.

Here we use cross-sectional data from the stock market as input, assuming there are $N$ stocks being traded on market on day $d$, the cross-sectional data can be denoted as:

$$X^d = [x_1^d, ..., x_N^d] \in \mathbb{R}^{N\times T\times F} \tag{1}$$

Since the return rate is more stable compared to future prices, we use return rate as label. Due to the restrictions in Chinese market, when we acquire the closing price of stocks on day $d$, the stocks can be purchased on $d+1$ and sold on $d+2$. Therefore, we set labels as the return rate on $d+2$.

$$y_i^d = \frac{Price_i^{d+2} - Price_i^{d+1}}{Price_i^{d+1}} \tag{2}$$

where $Price_i^d$ is the close price of $stock_i$ at date $d$.

## IV. Approach

The overall architecture of Finformer model is depicted in Figure 1, which is segmented into three parts: the Temporal Encoder, the Sparse Static-Dynamic Transformer Encoder, and the Gated Fusion module. Specifically, the Temporal Encoder uses time series data of each stock to extract temporal features, thereby learning how historical performance influences future trends from a temporal perspective. The Sparse Static-Dynamic Transformer Encoder constructs dynamic relationships between stocks and extracts spatial features by combining them with predefined static market relationships. Since the impact of temporal and spatial features on stock trends is not always identical, the Gated Fusion module learns to adaptively adjust the fusion of temporal and spatial features .

### A. Temporal Encoder

The future price trend of a stock is closely linked to its historical performance, which can provide valuable insights into future stock prices. To obtain the temporal embedding of stocks, we input historical time series data $X^d$ and use Gated Recurrent Units (GRU) [20] to generate the temporal embedding of the stocks.

$$z_i^t = \sigma\left(W_z x_i^t + U_z h_i^{t-1} + b_z\right) \tag{3}$$

$$r_i^t = \sigma\left(W_r x_i^t + U_r h_i^{t-1} + b_r\right) \tag{4}$$

$$h_i^{t\prime} = tanh\left(W_h x_i^t + U_h\left(r_i^t \odot h_i^{t-1}\right) + b_h\right) \tag{5}$$

$$h_i^t = \left(1 - z_i^t\right) \odot h_i^{t-1} + z_i^t \odot h_i^{t\prime} \tag{6}$$

$$E_T = [h_1^T, h_2^T, ..., h_N^T] \tag{7}$$

where $x_i^t$ represents the time step $t$ of $stock_i$ in $X^d$, $W_{z,r,h}$, $U_{z,r,h}$, and $b_{z,r,h}$ are learnable parameters. $\sigma$ is the activation function, here we use Sigmoid. $z_i^t$ is the activation of the update gate at time step $t$, $r_i^t$ is activation of the reset gate at time step $t$, and $h_i^{t\prime}$ is the computed candidate hidden state, $\odot$ is the Hadamard product. We concatenate the last time step of the outputs from the GRU to form the temporal embedding $E_T \in \mathbb{R}^{N\times F}$ for $X^d$.

### B. Sparse Static-Dynamic Transformer

We designed a Sparse Static-Dynamic Transformer Encoder for obtaining stock spatial embeddings. Given the capability of the multi-head attention mechanism to discover multiple relationships across different subspaces [39], we aim to explore various types of spatial relationships by using multiple heads, with each one assigned to handle a specific type of relationship.

To obtain the daily spatial embedding, denoted as $S^T \in \mathbb{R}^{N\times T\times F}$, for each daily data $X^d$ we first process through the S-D Attention mechanism to obtain the updated static-dynamic

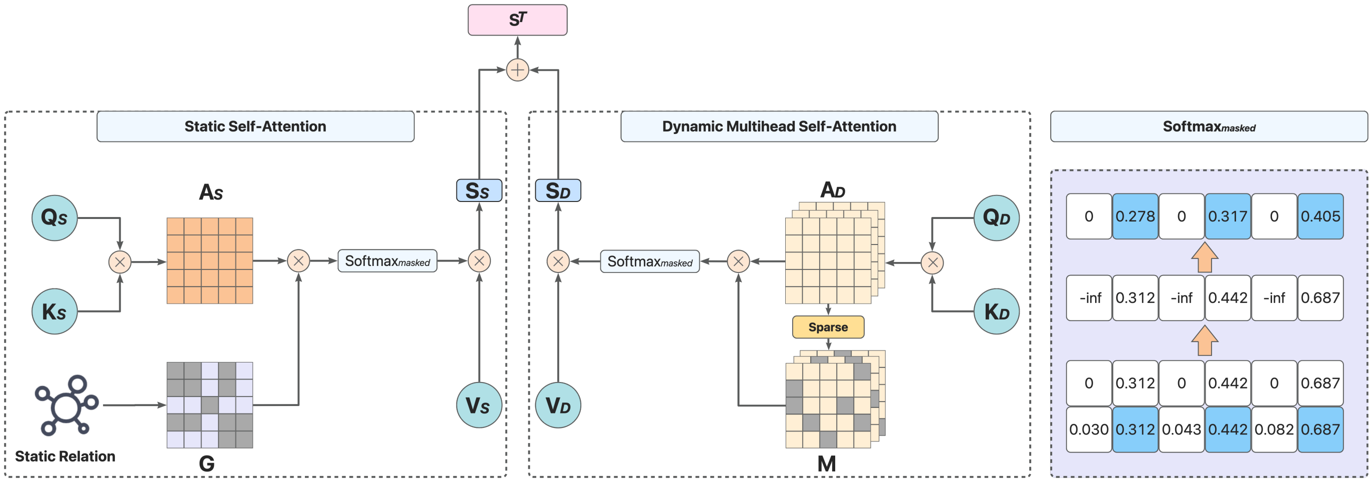



Fig. 2. S-D Attention mechanism. S-SA generates the static spatial embedding, and D-MHSA captures the sparse dynamic relationships between stocks and generate dynamic spatial embedding.

integrated spatial embedding, denoted as $S_{S-D}$, based on multiple static-dynamic relationships.

$$S_{S-D} = S - DAttention\left(X^d\right) \tag{8}$$

where $S_{S-D} \in \mathbb{R}^{N\times T\times F}$ is the static-dynamic integrated spatial embedding. Next, the daily spatial embedding $S^T$ is obtained as follows.

$$S'_{S-D} = LN\left(S_{S-D} + X^d\right) \tag{9}$$

$$S^T = LN\left(FFN\left(S'_{S-D}\right) + S'_{S-D}\right) \tag{10}$$

where $LN$ is $LayerNorm$, and $FFN$ represents feed forward network which is a 2-layers MLP.

Finally, we aggregate the daily spatial embedding $S^T$ by applying the attention mechanism along the time dimension, forming a final spatial embedding $E_S \in \mathbb{R}^{N\times F}$ for historical period.

Our primary enhancements revolve around the S-D Attention mechanism, which is composed of Static Self-Attention (S-SA) and Dynamic Multi-head Self-Attention (D-MHSA), as well as the Spatial Embedding Aggregation module.We select the most fundamental and readily accessible static spatial relationships as the input for S-SA, and capture dynamic spatial relationships through D-MHSA. In the subsequent sections, we will provide a comprehensive explanation of our design.

*1) Static Self-Attention:* The future trends of stocks are substantially influenced by the industries they belong to, particularly for stocks within the same industry, which can significantly impact one another. However, even among stocks in the same industry, the degree of interaction may differ. To incorporate industry relationship information into our proposed model, we utilize a single attention head to capture the static spatial embedding, denoted as $S_S$. As depicted in the left section of Figure 2, we employ a self-attention mechanism to calculate the degree of mutual influence between stocks based on predefined industry relationships.

Specifically, we first construct a relationship adjacency matrix $G \in \mathbb{R}^{N\times N}$ between stocks based on the input of predefined industry relationship information. If $stock_i$ and $stock_j$ are in the same industry, the $G_{i,j}$ will set to 1, otherwise it will be set to 0.

$$G_{ij} = \begin{cases} 1, stock_i \ and \ stock_j \ are \ related \\ 0, \qquad\qquad\qquad\quad otherwise \end{cases} \tag{11}$$

Following the standard self-attention mechanism, we compute the attention scores between stocks based on their historical time series data. To incorporate industry relationships, we perform a point-wise multiplication of the adjacency matrix $G$ with the static relationship attention score matrix $A_S$.

$$A_S = \frac{Q_S K_S^T}{\sqrt{e}} \cdot G \tag{12}$$

where $Q_s$ and $K_s$ are Query and Key of static attention head, and $e$ is the dimension of the key vectors. The updated attention score matrix will only have non-zero values at positions where a relationship exists between two stocks, serving as static relationship coefficients between stocks.

To address the normalization issue during the attention score calculation, we propose the use of a modified softmax function called $softmax_{masked}$, as illustrated in the right part of Figure 2. It assigns a value of $-inf$ to positions previously set to zero, thus ensuring they remain zeros during the normalization process. This approach effectively performs normalization only between stocks that have existing relationships, resulting in a sparse static relationship attention score matrix. Finally, the updated attention scores are multiplied by $value$ to obtain the stock's static relationship embedding.

$$S_S = S - SA\left(X_d, G\right) = softmax_{masked}\left(A_S\right)V_S \tag{13}$$

where $V_s$ is the Value of static attention head.

*2) Dynamic Multi-head Self-Attention:* The relationships within the stock market are complex and dynamic. Predefined static relationship falls short in providing a comprehensive representation of the ever-changing stock market relationships. Moreover, the incorporation of multiple predefined stock relationships into deep learning models presents certain limitations: 1) The static relationships are not all intuitive and easy to obtain, many types of static relationships are difficult to fully integrate and construct, some relationships may require much prior knowledge or difficult to obtain, such as company's profit model relationships, and the quality of information cannot be guaranteed. Practice has shown that the predictive performance of models that solely utilize static relationships largely depends on the quality of information, rather than the ability of the predictive model [9]. 2) The stock market is not static and varies across different periods. Predefined relationships are unable to effectively capture the dynamic interactions between stocks.

To capture and provide a more comprehensive representation of dynamic relationships, we introduce the Dynamic Multihead Self-Attention (D-MHSA) module, as depicted in the right portion of Figure 2. The traditional multi-head attention mechanism unveils various latent information by extracting relationship from disparate subspaces. Our goal is to design a sparse multi-head attention mechanism to uncover pivotal multiple types of time-varying relationships between stocks while concurrently discarding redundant relationships.

Specifically, we feed $X_d$ into the D-MHSA module. Initially, we compute attention scores between stocks with the objective of measuring their correlation across different relationships, utilizing $H$ heads for this purpose.

$$Q_D^h = X_d \cdot W_{Q_D^h}, K_D^h = X_d \cdot W_{K_D^h}, V_D^h = X_d \cdot W_{V_D^h} \tag{14}$$

$$A_D^h = \frac{Q_D^h {K_D^h}^T}{\sqrt{e_h}} \tag{15}$$

where $Q_D^h$, $K_D^h$, $V_D^h$ are Query, Key and Value, $W_{Q_D^h, K_D^h, V_D^h}$ are learnable parameters, $A_D^h$ is the dynamic attention score matrix and $e_h$ is the dimensionality of key vectors for head $h$.

Subsequently, we assign relationships for each stock. Given the interconnected nature of the stock market, no stock can operate in isolation. To maintain the sparsity of stock relationships while identifying the most important ones, we calculate an average attention score for each row in $A_D$ as a threshold which serves to retain important relationships for each stock and eliminate surplus ones, thus yielding a sparse relationship matrix.

$$M_{i,j}^h = \begin{cases} 1, M_{i,j}^h \geq mean(M_i^h) \\ 0, \qquad otherwise \end{cases} \tag{16}$$

where $M_{i,j}^h$ represents the mask status between stock $stock_i$ and $stock_j$ in attention head $h$. Then, we use the $softmax_{masked}$ mentioned ahead to normalize the matrix, obtaining an updated sparse attention score matrix.

$$A_D^h = softmax_{masked}\left(A_D^h \cdot M^h\right) \tag{17}$$

Finally, we multiply the multi-head attention score matrix with $value$ and concatenate to obtain the attention dynamic relationship representation.

$$O_h = A_D^h \cdot V_D^h \tag{18}$$

$$S_D = D - MHSA\left(X_d\right) = [O_1 ... O_H] W_H \tag{19}$$

where $S_D$ is the dynamic spatial embedding. $V_D^h$ is the Value, and $O_h$ is the output of dynamic attention head $h$ and $W_H$ is a learnable matrix to integrate $H$ heads.

So far, we have obtained the static spatial embedding $S_S$ and the dynamic spatial embedding $S_D$. Next we combine embeddings to generate a comprehensive spatial representation that captures both the static and dynamic relationships between stocks. We add the static and dynamic spatial embeddings together to ensures that both the static and dynamic aspects of the relationships between stocks are captured.

$$S_{S-D} = S_S + S_D \tag{20}$$

Then we pass $S_{S-D}$ through next parts of Sparse S-D Transformer Encoder, following the processes outlined in Equation 9 and Equation 10, producing the daily spatial embedding $S^T$.

*3) Spatial Embedding Aggregation:* Considering the impact of different date embedding on prediction results, to aggregate daily spatial embeddings $S^T$, for each stock, we use temporal attention mechanism to aggregate the spatial embeddings along time dimension based on the impact of different dates on the next stock trend. We add positional encoding to integrate positional information of temporal sequence into the aggregation to overcome the position insensitivity of the attention mechanism. The temporal attention mechanism assigns attention weights to each time step, and thus, we take the outputs of the last time step as the final stock spatial embedding $E_S$.

$$E_S^i = TemporalAttention\left(S_i^T + PE\left(S_i^T\right)\right) \tag{21}$$

$$E_S = [E_S^1, E_S^2, ..., E_S^N] \tag{22}$$

where $i$ represents the $i$-th stock in date $d$.

### C. Gated Fusion

Stocks possess both temporal and spatial attributes, but their impact on future trends is not equal at all time. Instead of simply concatenating these temporal and spatial features, we use the Gated Fusion module to combine two features adaptively. Specifically, it learns a gate parameter to establish the respective contributions of temporal and spatial embeddings in the final representation.

As illustrated in the right segment of Figure 1, the Gated Fusion module takes temporal embedding $E_T$ and spatial embedding $E_S$ as input, and are aggregated as follows.

$$g = \sigma\left(W_g\left[E_S^i; E_T^i\right] + b_g\right) \tag{23}$$

$$Z_i = g \odot E_S^i + (1-g) \odot E_T^i \tag{24}$$

where $Z_i$ is the spatiotemporal embedding of $stock_i$ that used for predict the future trends of stock, $g$ is the gate and $W_g$,$b_g$ are learnable parameters used to adjust the fusion of spatial and temporal embeddings. $\sigma$ represents the activation function, and here we use LeakyReLU.

### D. Learning Objective

We use the Concordance Correlation Coefficient ($CCC$) [40] as the loss function, with the goal of optimizing our model through the minimization of the negative $CCC$ value. The $CCC$ loss is significant as it takes into account not just the deviation of predicted values, but also their correlation with ground truth values. It enhances the robustness of training process by simultaneously considering the means and variances of both predicted and ground truth values.

$$\mathcal{L} = -\sum_{d \in D} \frac{2\rho_{\hat{y}^d y^d} \sigma_{\hat{y}^d} \sigma_{y^d}}{\sigma_{\hat{y}^d}^2 + \sigma_{y^d}^2 + \left(\mu_{\hat{y}^d} - \mu_{y^d}\right)^2} \tag{25}$$

where $D$ represents the dates in the training set, $\hat{y}^d$ represents the predicted return of all stocks on a day, $y^d$ represents the ground-truth return of all stocks on date $d$. $\rho$ represents the Pearson correlation coefficient, $\sigma$ represents the standard deviation, and $\mu$ represents the mean value.

## V. Experiments

In this section, we conduct a series of experiment to address the following research questions:

**RQ1**: Can our proposed model surpass other baseline models on key performance metrics in the quantitative domain using real market datasets? Is our proposed method effective for both large-scale and small-scale datasets?

**RQ2**: How do different modules in our model impact performance, specifically, is our custom-designed sparse attention mechanism more effective than other global attention mechanisms?

**RQ3**: How sensitive is our proposed model to changes in hyper-parameters?

**RQ4**: Is our model practically effective in real market trading? Can it generate significant returns in backtesting?

### A. Implementation Details

*1) Dataset:* We built experimental datasets from two real-world stock markets. We collected data on stocks from CSI300 and CSI100, including $opening\ price$, $closing\ price$, $highest\ price$, $lowest\ price$, $trading\ volume$ (OCHLV) and $volume\ weighted\ average\ price$ of the CSI300 and CSI100 from JoinQuant[2] for the period of 2010-01-01 to 2022-12-31.

TABLE I
DATA SEGMENTATION

| | CSI100 | CSI300 |
|---|---|---|
| Train | 2010/01/01-2015/12/31 | 2010/01/01-2015/12/31 |
| Valid | 2016/01/01-2017/12/31 | 2016/01/01-2017/12/31 |
| Test | 2018/01/01-2022/12/31 | 2018/01/01-2022/12/31 |

The CSI300 comprises 300 highly representative securities characterized by larger market capitalization and superior liquidity within China's A-share market which reflects the overall status of the Chinese stock market significantly. The CSI100 mirrors the overall situation of a group of large-capitalization companies wielding the most market influence in the Chinese stock market. We utilized the Alpha360 factor library provided by Qlib[3], which includes daily stock trading data with six features and a 60-day lookback window. The stock industry classification was derived from the Shenwan Secondary Industry Classification[4], which is widely used in the Chinese stock market. We divided the dataset as Table I.

*2) Baselines:* We selected several state of the art models in the field of stock trend prediction as a comparison. To compare performance with multiple types of models, we chose temporal models and correlation based models.

- **MLP**: A three-layer multi-layer perceptron. The unit of each layer is set to 512.
- **TCN** [41]: A Time Convolution Network, which is improved based on CNN and applied to time series analysis.
- **ALSTM** [3]: A variant of LSTM, which applies the attention mechanism to the time step to adaptively extract features at each time step.
- **TRA+ALSTM** [6]: TRA enhances ALSTM model by recognizing the trading pattern. It forecasts the stock trend according to the trading pattern.
- **SFM** [4]: Discrete Fourier transform is used to decompose hidden state and capture multi-frequency trading patterns.
- **Transformer** [39]: Enhanced Transformer designed for stock forecasting task.
- **Crossformer** [42]: A Transformer based multivariate time series analysis model. A two-stage attention mechanism is designed for both time and variable dimensions, and an efficient routing attention mechanism is adopted.
- **GAT** [43]: Each stock is regarded as a node, and the GRU is used to extract the temporal embeddings of the stock as node embedding.
- **DTML** [9]: Extracting dynamic relationships between stocks by using Transformers without prior knowledge.
- **THGNN** [10]: A GNN-based approach, using dynamic heterogeneous graph to forecast stock trend.
- **HIST** [31]: Predict stock trends based on predefined conceptual relationships and mining hidden conceptual relationships in the stock market using historical data.

*3) Implementations:* We use PyTorch library to implement the models and use Adam optimizer [44] for training. The training process is accelerated by a single NVIDIA GeForce RTX 3090 GPU. For fairly comparison, we conduct grid search for all models. We explore hidden size of {32, 64, 128, 256}, dynamic spatial attention heads of {1, 2, 4, 8}, learning rates of {1e-3, 1e-4, 2e-4} for our proposed model, we add an additional searching of batch size of {512, 1024, 2048}, and layer numbers of {1, 2, 3} for baselines. To mitigate randomness, we conduct five rounds of experiments.

*4) Metrics:* We employ several critical indicators in quantitative finance to evaluate the performance of models. We

[2]see joinquant.com

[3]https://github.com/microsoft/qlib

[4]https://www.swsresearch.com/institute_sw/allIndex/releasedIndex

TABLE II

MAIN RESULT(WITH STANDARD DEVIATION). PERFORMANCE EVALUATION OF STOCK TREND PREDICTION COMPARISON MODELS ON CSI100 AND CSI300. IC AND RANKIC MEASURES THE CORRELATION AND ACCURACY BETWEEN PREDICTED RESULTS AND ACTUAL RETURNS, THE HIGHER THE BETTER. ICIR AND RANKICIR MEASURES THE STABILITY OF MODEL PREDICTIONS AT DIFFERENT PERIODS, AND THE HIGHER THE BETTER.

| Method | | CSI100 | | | | CSI300 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **IC** | **ICIR** | **RankIC** | **RankICIR** | **IC** | **ICIR** | **RankIC** | **RankICIR** |
| **Time Series** | MLP | 0.021 (0.002) | 0.126 (0.009) | 0.020 (0.002) | 0.123 (0.011) | 0.024 (0.003) | 0.186 (0.027) | 0.022 (0.003) | 0.177 (0.028) |
| | TCN | 0.036 (0.003) | 0.199 (0.027) | 0.035 (0.003) | 0.195 (0.029) | 0.039 (0.001) | 0.289 (0.020) | 0.035 (0.001) | 0.265 (0.018) |
| | ALSTM | 0.042 (0.003) | 0.235 (0.019) | 0.042 (0.003) | 0.233 (0.019) | 0.045 (0.001) | 0.339 (0.010) | 0.043 (0.001) | 0.330 (0.011) |
| | TRA+ALSTM | 0.037 (0.006) | 0.215 (0.038) | 0.035 (0.007) | 0.208 (0.045) | 0.043 (0.003) | 0.338 (0.030) | 0.042 (0.003) | 0.332 (0.028) |
| | SFM | 0.035 (0.002) | 0.192 (0.017) | 0.032 (0.002) | 0.178 (0.012) | 0.037 (0.003) | 0.285 (0.023) | 0.033 (0.004) | 0.251 (0.030) |
| | Transformer | 0.032 (0.004) | 0.168 (0.021) | 0.032 (0.004) | 0.173 (0.024) | 0.023 (0.002) | 0.145 (0.015) | 0.024 (0.002) | 0.154 (0.019) |
| | Crossformer | 0.040 (0.001) | 0.221 (0.015) | 0.038 (0.002) | 0.217 (0.018) | 0.040 (0.002) | 0.306 (0.026) | 0.039 (0.002) | 0.301 (0.027) |
| **Correlation Based** | GAT | 0.030 (0.004) | 0.147 (0.024) | 0.026 (0.004) | 0.129 (0.030) | 0.038 (0.003) | 0.250 (0.036) | 0.036 (0.004) | 0.234 (0.042) |
| | DTML | 0.043 (0.003) | 0.205 (0.019) | 0.042 (0.003) | 0.205 (0.020) | 0.042 (0.002) | 0.259 (0.015) | 0.041 (0.002) | 0.257 (0.014) |
| | THGNN | 0.041 (0.001) | 0.230 (0.007) | 0.040 (0.002) | 0.226 (0.012) | 0.043 (0.001) | 0.317 (0.014) | 0.041 (0.001) | 0.310 (0.015) |
| | HIST | 0.044 (0.003) | 0.216 (0.014) | 0.043 (0.003) | 0.212 (0.015) | 0.049 (0.004) | 0.293 (0.029) | 0.047 (0.004) | 0.279 (0.028) |
| | **Finformer** | **0.050 (0.001)** | **0.266 (0.007)** | **0.049 (0.002)** | **0.264 (0.009)** | **0.052 (0.002)** | **0.359 (0.012)** | **0.050 (0.002)** | **0.345 (0.013)** |

choose IC and RankIC to evaluate the accuracy of the model's predictions. To assess the stability of the model's predictions across various periods, we also include ICIR and RankICIR.

- **IC** Information Coefficient. Calculated by Pearson correlation coefficient [45] between the predicted return and the ground-truth return of stocks.

$$IC = \frac{1}{N}\frac{(\hat{y} - mean(\hat{y}))^T(y - mean(y))}{std(\hat{y}) * std(y)} \tag{26}$$

  where $y$ is the ground-truth value of return rate, $\hat{y}$ is the predicted value.
- **RankIC** Rank Information Coefficient. Calculated by Spearman correlation coefficient [46] between ranking of predicted returns and ranking of ground-truth returns.

$$RankIC = corr(\hat{y}, y) \tag{27}$$

  where $corr$ represents Spearman correlation.
- **ICIR** Combined IC and IR(Information Ratio) [47]. Evaluate the stability in different period

$$ICIR = \frac{mean(IC)}{std(IC)} \tag{28}$$

- **RankICIR** Combined RankIC and IR. Evaluate the stability of the model's return ranking in different period.

$$RankICIR = \frac{mean(RankIC)}{std(RankIC)} \tag{29}$$

### B. Main Results

To address RQ1, we designed an experiment aimed at predicting stock returns, focusing on the evaluating accuracy. Table II depicts the results of experiments. From Table II, the following observations can be made:

1) Compared with other models that integrated spatial information, our proposed model significantly enhances key performance indicators. Specifically, we observed a 13% and 6% increase on IC and a 14% and 6% increase on RankIC compared to the state-of-the-art model, respectively. Results suggest that our model improves the accuracy of stock trend prediction considerably. Concurrently, the improvement in ICIR and RankICIR indicates the higher stability of our model.

2) Relative to time series models, the approaches which integrate static or dynamic relationships demonstrate superior competitiveness. This suggests that the inclusion of spatial relationships can contribute substantially to the enhancement of trend prediction performance.

3) Compared to the GAT model, which uses a fully connected adjacency matrix, our model, which focuses on significant spatial relationships while excluding redundant ones, yields better prediction performance.

### C. Ablation Study

To address RQ2, we executed ablation studies to investigate the influence of each module on the results of stock trend predictions. We progressively eliminated specific modules to examine the effect of predefined static relations, dynamic relations, sparse attention, and gated fusion on the precision and consistency of the model's predictions. We designed the ablation models as follows.

- **w/o S**: S-SA module is removed, the model concentrates solely on dynamic relations in spatial embedding.
- **w/o D**: D-MHSA module is removed, the model focuses exclusively on static relations in spatial embedding.
- **w/o SD**: D-MHSA and S-SA are all removed, degrading the Sparse S-D Transformer Encoder to vanilla Transformer Encoder.
- **w/o GF**: Gated Fusion module is removed, and the method of aggregating spatial-temporal features is replaced with simple concatenation.

Table III presents the results of ablation experiments. These results are aimed at examining the influence of each module in the model on stock trend prediction. The following conclusions can be drawn from the results:

1) The $w/o$ $D$ model, which only utilizes predefined static relationships, decreases the model's predictive accuracy. It implies that a single predefined static relationship is insufficient in fully describing the complex, time-evolving relationships within the stock market.

2) The $w/o$ $S$ model, which only employs dynamic relationships, improves prediction accuracy compared to the $w/o$ $D$ model, yet still falls short of optimal results. It indicates that

TABLE III
ABLATION STUDY RESULT. MAINLY SHOWCASING COMPARISON BETWEEN SPARSE ATTENTION AND GLOBAL ATTENTION, AS WELL AS INTEGRATED STATIC-DYNAMIC RELATIONSHIPS AND SINGLE SPATIAL RELATIONSHIPS.

| Method | CSI100 | | | | CSI300 | | | |
|---|---|---|---|---|---|---|---|---|
| | **IC** | **ICIR** | **RankIC** | **RankICIR** | **IC** | **ICIR** | **RankIC** | **RankICIR** |
| w/o S | 0.042 (0.003) | 0.214 (0.010) | 0.042 (0.004) | 0.213 (0.016) | 0.049 (0.002) | 0.339 (0.015) | 0.048 (0.002) | 0.329 (0.013) |
| w/o D | 0.040 (0.004) | 0.220 (0.037) | 0.039 (0.006) | 0.213 (0.044) | 0.045 (0.001) | 0.354 (0.022) | 0.043 (0.001) | 0.340 (0.018) |
| w/o SD | 0.044 (0.003) | 0.236 (0.021) | 0.043 (0.002) | 0.228 (0.018) | 0.050 (0.001) | 0.342 (0.006) | 0.047 (0.001) | 0.324 (0.004) |
| w/o GF | 0.045 (0.001) | 0.224 (0.007) | 0.044 (0.002) | 0.222 (0.008) | 0.050 (0.003) | 0.327 (0.026) | 0.048 (0.003) | 0.310 (0.029) |
| **Finformer** | **0.050 (0.001)** | **0.266 (0.007)** | **0.049 (0.002)** | **0.264 (0.009)** | **0.052 (0.002)** | **0.359 (0.012)** | **0.050 (0.002)** | **0.345 (0.013)** |

models relying solely on dynamic relationships are unable to fully capture the robust, intrinsic relationships exist between stocks in the real world.

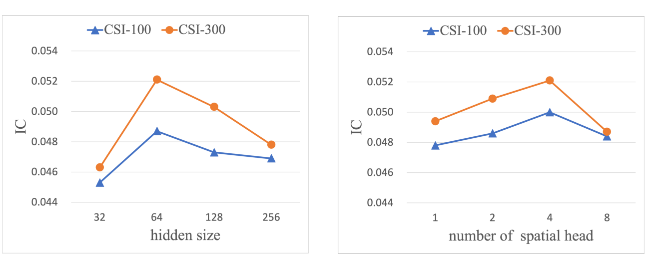
3) The $w/o\ SD$ model, which applies global spatial attention, outperforms both the $w/o\ S$ and $w/o\ D$ that only consider a single type of relationship. Nonetheless, the full model, which employs sparse attention to focus on essential relationships, outperforms the $w/o\ SD$ model that uses global attention. It demonstrates that considering global relationship can prevent information loss and enhance predictive accuracy, but including massive redundant relationships may lead to attention distraction. And the full model can maintain the significant relationships while eliminating the redundant ones.

4) The results of the $w/o\ GF$ model suggest that adaptive spatial-temporal fusion can more effectively balance the trade-off between two types of features compared to simple concatenation.

Overall, after considering the temporal and spatial relationships separately, the performance of the model has been improved, which means that the addition of spatial relationships has a gain effect on the model's performance. Meanwhile, both solely consideration of static relationships or dynamic relationships have a negative impact on performance. Compared to global relationships, the application of sparse relationships avoids the reduction in performance caused by the introduction of a large number of redundant relationships.

### D. Hyper-parameter Sensitivity

To address RQ3, we conducted hyper-parameter study to explore the influence of hyper-parameters on the performance of our proposed model. The hyper-parameters include the hidden size and the number of spatial attention heads.




(a) Impact of hidden size (b) Impact of s-head number

Fig. 3. Hyper-parameter study results

As illustrated in Figure 3(a), when the hidden size of the model is increased to 64, the model's performance peaks. However, a further increase in hidden size leads to a significant drop in the model's performance which suggests that increasing the hidden layer size allows the model to capture more intricate patterns and additional information, enhancing its ability to adapt to training data and boosting prediction accuracy. However, too many parameters may lead to overfitting to noise in training data, thereby impairing its ability to generalize to test data.

Furthermore, with regards to the number of dynamic spatial attention heads, as shown in Figure 3(b), we found that a gradual increase in the spatial head number initially improved the model's prediction accuracy. The model achieved optimal performance when the value reached 4, but any subsequent increase led to a decline in prediction accuracy which suggests that moderately increasing the number of dynamic relationship types can be beneficial for modeling dynamic relationships in the stock market. However, too many types of relationship may make it difficult to effectively identify the primary ones, resulting in a decrease in prediction accuracy.

### E. Portfolio Evaluation

To address RQ4, we conducted portfolio experiments on the datasets to verify the effectiveness of our model in real-world stock markets investment applications.

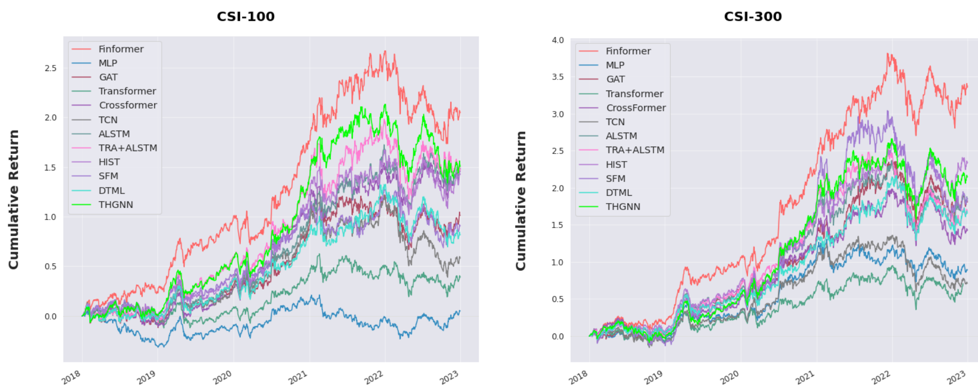
Given the discrepancies in the size of the securities across different datasets, we chose the top 10% of stocks to construct the investment portfolio. Following the stock selection, we allocated the remaining funds. To further demonstrate the impact of predictive accuracy on backtesting outcomes, we established differentiated weights based on predicted returns. Specifically, we normalized the predicted returns of all stocks in the current portfolio to form capital allocation weights for each stock, in contrast to the even allocation strategy. Accordingly, we allocated all remaining funds as the obtained weight vector, constituting the current investment plan. In backtesting, we conducted daily rebalancing, selling the currently held stocks to gain profit and then repurchasing stocks as the aforementioned steps. We set the initial funds at $10^8$.

We selected multiple indicators which are widely used [6], [48] in the field of portfolio to evaluate the performance of the model. The **Annualized Return** (AR) measures the expected profits of the portfolio that generated based on the model predictions on a yearly basis, **Annualized Volatility** (AV) measures the value fluctuation of portfolio, **Sharpe Ratio** (SR) measures risk-adjusted return, where $SR = \frac{AR}{AV}$ and

TABLE IV

PORTFOLIOS RESULTS. AR MEASURES THE RETURN RATES ON PORTFOLIOS, THE HIGHER THE BETTER. AV AND MDD MEASURE THE RISK OF THE PORTFOLIO, THE LOWER ABSOLUTE VALUE THE BETTER. SR MEASURES THE PROFIT UNDER A UNIT OF RISK, AND THE HIGHER THE BETTER.

| Method | CSI100 | | | | CSI300 | | | |
|---|---|---|---|---|---|---|---|---|
| | **AR(%)** | **AV(%)** | **SR** | **MDD(%)** | **AR(%)** | **AV(%)** | **SR** | **MDD(%)** |
| MLP | 1.00 | 23.06 | 0.04 | 36.65 | 13.67 | 23.04 | 0.59 | 24.92 |
| TCN | 9.61 | 24.19 | 0.39 | 37.43 | 11.36 | 23.28 | 0.48 | 32.12 |
| ALSTM | 20.00 | 23.86 | 0.83 | 25.20 | 23.40 | 24.23 | 0.96 | 31.67 |
| TRA+ALSTM | 19.82 | 24.78 | 0.79 | 27.02 | 21.87 | 24.08 | 0.90 | 34.26 |
| SFM | 13.83 | 23.52 | 0.58 | 25.99 | 23.06 | 27.49 | 0.83 | 35.90 |
| Transformer | 6.88 | 23.03 | 0.29 | 26.69 | 11.44 | 24.83 | 0.46 | 30.44 |
| Crossformer | 19.68 | 22.48 | 0.94 | **21.11** | 19.50 | 23.96 | 0.89 | 25.71 |
| GAT | 15.02 | 24.50 | 0.61 | 25.68 | 22.95 | 24.23 | 0.94 | 27.51 |
| DTML | 12.71 | 23.88 | 0.53 | 27.09 | 21.61 | 23.49 | 0.91 | 27.57 |
| THGNN | 20.16 | 22.66 | 0.88 | 25.28 | 25.82 | 23.44 | 1.10 | 29.64 |
| HIST | 19.44 | 23.32 | 0.83 | 24.39 | 27.36 | 22.89 | 1.19 | 26.07 |
| **Finformer** | **25.04** | **22.47** | **1.11** | 25.47 | **34.39** | **22.58** | **1.52** | **23.11** |




(a) Cumulative return on CSI100 (b) Cumulative return on CSI300

Fig. 4. The cumulative return of models from 2018 to the end of 2022 on CSI100 and CSI300 datasets.

**Maximum Drawdown** (MDD) measures the maximum value decline of portfolio during the backtesting period.

Table IV shows the backtesting results, demonstrating that our model outperforms other baselines in achieving a favorable balance between risk and return. Our model yields an annualized return of 25% on CSI100 and 34% on CSI300, showcasing a significant enhancement compared to the baselines. Alongside the high returns, the MDD remains relatively low, indicating that the investment portfolio generated by our model strikes an impressive balance between risk and return.

As illustrated in Figure 4, the cumulative returns of the model during the five-year backtesting on both datasets substantially exceeded the baselines for the majority of the time. This is particularly notable during the period from mid-2018 to early-2019, when several models underwent a significant drawdown in cumulative returns due to market downturns. This demonstrates the superior resilience to risk of the investment portfolio designed by our proposed model. Given the impact of COVID-19, all models experienced significant volatility in 2022 unavoidably. However, our proposed model had a shorter recovery period compared to the other baseline models.

## VI. CONCLUSION

Stock trend prediction is a vital component of quantitative investment, impacting decision-making and efficiency. In this paper, we propose a Sparse Static-Dynamic Transformer Encoder that captures the potential sparse dynamic relationships between stocks using historical time series data, and combines them with predefined relationships to further describe the relationships in stock market and avoid redundancy. Additionally, we employ a Gated Fusion module to adaptively integrate spatial-temporal features. Experimental results demonstrate that our Finformer model outperforms current models in signal and portfolio indicators. In future research, we aim to further refine dynamic relationship modeling in the stock market by incorporating real-time events, enhancing prediction accuracy.

## REFERENCES

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

[2] N. Barberis and R. Thaler, “A survey of behavioral finance,” *Handbook of the Economics of Finance*, vol. 1, pp. 1053–1128, 2003.

[3] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua, “Enhancing stock movement prediction with adversarial training,” *IJCAI*, 2019.

[4] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. W. Cottrell, “A dual-stage attention-based recurrent neural network for time series prediction,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 2627–2633.

[5] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long-short term memory,” *PloS one*, vol. 12, no. 7, p. e0180944, 2017.

[6] H. Lin, D. Zhou, W. Liu, and J. Bian, "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*, 2021, pp. 1017–1026.

[7] R. Cheng and Q. Li, "Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 1, 2021, pp. 55–62.

[8] Y.-L. Hsu, Y.-C. Tsai, and C.-T. Li, "Fingat: Financial graph attention networks for recommending top-k profitable stocks," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[9] J. Yoo, Y. Soun, Y.-c. Park, and U. Kang, "Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2037–2045.

[10] S. Xiang, D. Cheng, C. Shang, Y. Zhang, and Y. Liang, "Temporal and heterogeneous graph neural network for financial time series prediction," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 3584–3593.

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

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

[13] L. Zhao, W. Li, R. Bao, K. Harimoto, X. Sun *et al.*, "Long-term, short-term and sudden event: trading volume movement prediction with graph-based multi-view modeling," *arXiv preprint arXiv:2108.11318*, 2021.

[14] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[15] R. D. Edwards, W. Bassetti, and J. Magee, *Technical analysis of stock trends*. CRC press, 2012.

[16] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the arima model," *International Conference on Computer Modelling and Simulation*, 2014.

[17] L.-J. Cao and F. E. H. Tay, "Support vector machine with adaptive parameters in financial time series forecasting," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1506–1518, 2003.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

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

[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[21] Y. Gu, D. Yan, S. Yan, and Z. Jiang, "Price forecast with high-frequency finance data: An autoregressive recurrent neural network model with technical indicators," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 2485–2492.

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

[23] H. Huang, J. Gao, C. Xu, and X. Ying, "Pattern adaptive specialist network for learning trading patterns in stock market," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 4044–4048.

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

[25] S. Gao, Y. Wang, and X. Yang, "Stockformer: Learning hybrid trading machines with predictive coding," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, 2023, pp. 4766–4774.

[26] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[27] J. Wu, K. Xu, X. Chen, S. Li, and J. Zhao, "Price graphs: Utilizing the structural information of financial time series for stock prediction," *Information Sciences*, vol. 588, pp. 405–424, 2022.

[28] S. Li, J. Wu, X. Jiang, and K. Xu, "Chart gcn: Learning chart information with a graph convolutional network for stock movement prediction," *Knowledge-Based Systems*, vol. 248, p. 108842, 2022.

[29] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 497–504.

[30] X. Li, C. Cui, D. Cao, J. Du, and C. Zhang, "Hypergraph-based reinforcement learning for stock portfolio selection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4028–4032.

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

[32] D. Cheng, F. Yang, X. Wang, Y. Zhang, and L. Zhang, "Knowledge graph-based event embedding framework for financial quantitative investments," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 2221–2230.

[33] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," in *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, 2021, pp. 4541–4547.

[34] Y. Yang, Z. Wei, Q. Chen, and L. Wu, "Using external knowledge for financial event prediction based on graph neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2161–2164.

[35] Y. Xu and S. B. Cohen, "Stock movement prediction from tweets and historical prices," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1970–1979.

[36] R. Sawhney, S. Agarwal, A. Wadhwa, and R. Shah, "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)*, 2020, pp. 8415–8426.

[37] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, and H. Chen, "Knowledge-driven stock trend prediction and explanation via temporal convolutional network," in *Companion Proceedings of The 2019 World Wide Web Conference*, 2019, pp. 678–685.

[38] W. Xu, W. Liu, C. Xu, J. Bian, J. Yin, and T.-Y. Liu, "Rest: Relational event-driven stock trend forecasting," in *Proceedings of the Web Conference 2021*, 2021, pp. 1–10.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[40] I. Lawrence and K. Lin, "A concordance correlation coefficient to evaluate reproducibility," *Biometrics*, pp. 255–268, 1989.

[41] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[42] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The Eleventh International Conference on Learning Representations*, 2023.

[43] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

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

[45] I. Cohen, Y. Huang, J. Chen, J. Benesty, J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise reduction in speech processing*, pp. 1–4, 2009.

[46] L. Myers and M. J. Sirois, "Spearman correlation coefficients, differences between," *Encyclopedia of statistical sciences*, vol. 12, 2004.

[47] R. C. Grinold, "The fundamental law of active management," *Streetwise, The Best of the Journal of Portfolio Management*, pp. 161–8, 1989.

[48] H. Niu, S. Li, and J. Li, "Metatrader: An reinforcement learning approach integrating diverse policies for portfolio optimization," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1573–1583.