StockLTG: Multi-Scale Stock Trend Forecasting via Lagged Co-Trending Dynamic Graph Networks

Shiwei Pu, Chuanchang Liu*
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, 100876, China
pushiweiaa@163.com, lcc3265@bupt.edu.cn

***Abstract*—In the realm of stock forecasting, traditional relational modeling approaches are constrained by two significant limitations. First, relying solely on static attributes to describe stock relationships makes it challenging to capture the dynamic changes in these relationships over time. Second, these methods often overlook the interaction between the return rate of an individual stock and its lagged relation. To address these issues, this paper proposes a novel modeling framework based on return rate trend decomposition, which captures the dynamic changes in stock relationships and lagged interactions through multi-scale trend decomposition and lagged cross-correlation algorithms. Empirical analysis on two major datasets from Chinese and American stock markets demonstrates that the proposed model's information coefficient on Chinese stock market data is 80% higher than that of the baseline model, with significantly higher returns compared to the benchmark model. Additionally, the performance of the proposed framework on certain datasets surpasses that of current mainstream stock prediction models.**

***Index Terms*—Stock prediction, Lagged relation, Co-trending dynamic graph, Financial time series, Quantitative trading**

I. Introduction

Stock prediction is a foundational task within the realm of quantitative investment. However, stock prediction remains highly challenging due to the complex nature of financial markets and the low signal-to-noise ratio in financial data. Most of the existing neural network-based [1] stock prediction models depend on single time series or static modeling strategies [2], which are difficult to capture the complex dynamic characteristics of the stock market effectively.

Traditional static models [3] are based on industry graphs and tend to ignore the time-varying correlations between stocks when dealing with stock relationships. Direct feature modeling approaches [4]–[6] fail to capture the directional correlations among features, which leads to the inaccuracy of the prediction results. Stock correlation modeling is critical for stock prediction, since stock time series show distinct market-driven fluctuations from conventional ones, inspiring extensive related research. Hsu et al. [7] proposed MAN-SF and FinGAT respectively, leveraging Graph Attention Networks (GATs) [8] for time signal processing and stock-industry interaction analysis. To alleviate static graph limitations (lag, overfitting), Ma et al. [9] integrated dynamic graphs into stock relationship modules, and Lei et al. [10] proposed DR-GAT for stock recommendation. Other relevant works: Zhao et al. [11] explored knowledge graph spillovers with heterogeneous GNNs; Matsunaga et al. [12] combined knowledge graphs and GNNs for the Nikkei 225 market; Yang et al. [13] applied GGNNs for event prediction; Cheng et al. [14] presented the multi-modal GNN (MAGNN). The above research demonstrates the necessity of modeling the relationships between stocks. In measuring inter-stock relationships, the Pearson correlation coefficient is used to calculate stock correlations [15], yet its computational limitations render it unable to handle **non-monotonic** relationships and **lag effects**. Likewise, attention-based correlation calculation fails to reflect the **asymmetry** between stocks. Furthermore, when there is noise in the time series, the Pearson coefficient [16] is significantly distorted, and it cannot accurately reflect the similar trending patterns of stocks. To address these issues, this paper proposes a series decomposition-based cross-correlation calculation method to construct a stock lagged co-trending dynamic graph, and integrates Mamba [17] blocks to develop the **StockLTG** (**S**tock **L**agged co-**T**rending **G**raph) model, offering a new solution for stock prediction.

The StockLTG model constructs a new implicit graph relationship between stocks, effectively capturing stock trend changes and feature factor relationships. The main contributions of this paper are as follows: **1)** The paper decomposes and constructs the stock relationship matrix on three time scales: long-term trends, medium-term cycles, and short-term fluctuations. This approach overcomes single-scale analysis constraints, capturing lag effects and dynamic associations, enhancing the analysis of complex stock relationships. **2)** The study combines the dynamic trend graph with the feature factor graph to construct a heterogeneous graph of stock relationships. **3)** The correlation dynamic graph is calculated in the frequency domain, and the frequency component of the stock trend relationship is adaptively weighted, and the power spectral density weight is dynamically adjusted by the learnable linear transformation, which can suppress the interference of noise in the implicit graph.

II. Methodology

This paper focuses on mining the lagged trend implicit relationship between stocks and the feature factor graph re-

*Corresponding author

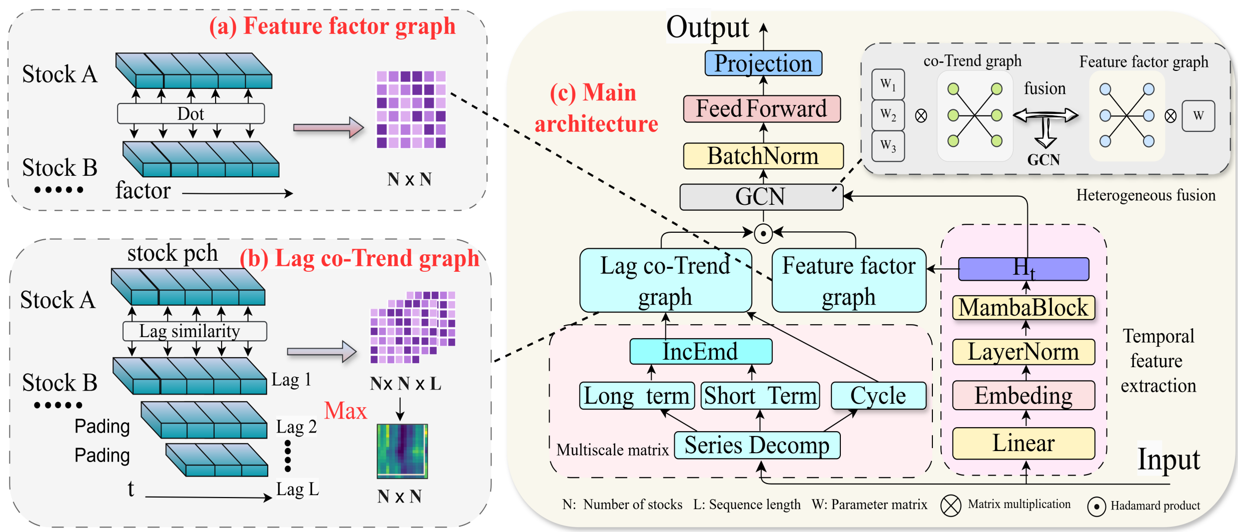

Fig. 1. The overall framework of StockLTG.

lationship, and modeling the stock dynamic graph and time series relationship to predict the stock return rate. The goal is to predict the return rate of N stocks at T+2 by taking the historical information and industry information of N stocks in the past L days as the model input for the trading day T.

Fig. 1 is the architecture diagram of StockLTG, which represents the overall model network block of StockLTG. Panel c is a simplification of the StockLTG model, including series decomposition as well as feature factor plots and lagged co-trending graph between stocks, and Mamba time series feature extraction blocks. Finally, after GCN [12] convolution and residual connection, it goes through the feedforward layer and then outputs. In the Fig.1, panels a and b show the details of the feature factor and lagged co-trending graph respectively.

A. Stock relationship construction

The construction of stock relationships involves creating a multi-scale lagged co-trending graph and a feature relationship graph, followed by graph aggregation. These two methods model inter-stock relationships through different embedding dimensions: one captures feature correlations, while the other identifies co-trends. In graph construction, the fusion of edges of different types of stocks forms a heterogeneous graph.

B. Lagged Co-Trending Dynamic Graph

Inspired by the cross-correlation of time series [18]–[20] and the theory of stochastic processes, we propose the lagged trend relationship of cross-correlation among stocks. The difference is that Autoformer [21] describes the lag relationship of its time series at different moments, while we describe the trend lag relationship between different time series.

When constructing the lagged co-trending graph, to reflect the positive correlation between the importance of stock trends and their distance from the prediction point, this paper designs an Increase Encoding (IncEmd) algorithm.

For the assumption that r is a sequence of rise/fall to be decomposed, encoding and embedding r are performed: $r = \text{Pading}(\text{IncEmd}(r))$. For the calculation of lag dot product between lag co-trend, we designed a coding based on logistic [22] time increasing, which can give more attention to the trend close to the forecast time when calculating the inner product of time series between stocks. The formula is: $\text{IncEmd} = \frac{x(t)}{1+e^{-k(t-t_0)}}$, where $x(t)$ is the sequence to be encoded, k is the growth rate, and t_0 is the midpoint of the encoding.

For each r, decompose into long-term, short-term, and medium-term sequences $r^w, w \in \{k,p,q\}$. For stock i and j the yield sequence of the $r_i^w = [r_i(1), \ldots, r_i(T)]$, $r_j^w = [r_j(1), \ldots, r_j(T)]$, the return series r is encoded with IncEmd, and the series pays more attention to the points closer to the prediction point when doing the dot product. The lag similarity $\text{LCoS}(i,j,\tau)$ at the lag order (τ) is defined as $\sum_{t=1}^{T-\tau} {r_i}^w(t) \cdot {r_j}^w(t+\tau)$. This formula assesses the lagged co-trend between two stock return series by computing their similarity at lag τ. Within a sliding window w, calculate the lagged similarity L_τ^{w} of the returns of all stocks in the market.

Here, $\text{LCoS}(r_i, r_j, \tau)$ represents the similarity of lagged trends for stocks i and j under lag τ, with computational complexity $O(\tau N^2 T)$. The lagged co-trend similarity matrix is

$$L_\tau^w = \begin{bmatrix} \text{LCoS}(r_1,r_1,\tau) & \text{LCoS}(r_1,r_2,\tau) & \cdots & \text{LCoS}(r_1,r_j,\tau) \\ \text{LCoS}(r_2,r_1,\tau) & \text{LCoS}(r_2,r_2,\tau) & \cdots & \text{LCoS}(r_2,r_j,\tau) \\ \vdots & \vdots & \ddots & \vdots \\ \text{LCoS}(r_i,r_1,\tau) & \text{LCoS}(r_i,r_2,\tau) & \cdots & \text{LCoS}(r_i,r_j,\tau) \end{bmatrix}. \quad (1)$$

This matrix represents the lagged co-trend similarity of all stock pairs at a specific time shift step. Because of $\text{LCoS}(r_i, r_j, \tau) \neq \text{LCoS}(r_j, r_i, \tau)$, the matrix L_τ^w is an asymmetric matrix. The w represents the trend of decomposition in different periods for the long-term L^k, medium-term L^q and short-term L^p, $w \in \{k,p,q\}$, for different periods of L^w

for linear weighting. The final similarity matrix is obtained by running Max (the maximum in the lag time dimension) :

$$L^w = \max_{\tau}\big(L_{\tau}^w\big), \quad (\tau = 0, 1, \ldots, T-1), \tag{2}$$

$$\alpha_{i,j}^w = \mathrm{c}_1 L^k + \mathrm{c}_2 L^q + \mathrm{c}_3 L^p. \tag{3}$$

The final similarity matrix $\alpha_{i,j}^w$ is obtained, where c_i is the parameter of the linear combination. The final calculated L matrix simultaneously includes the values of r_i lagging r_j and r_j lagging r_j, and it is asymmetric. To make it convenient, we only take the maximum value of the two, and set the other symmetrical value to 0. The original time-series cross-correlation matrix L is inherently asymmetric; the zeroing operation is designed to retain a unique unidirectional lead-lag causal relationship rather than destroy asymmetry. This treatment further strengthens the asymmetric property and temporal causal rationality of the lagged co-trend matrix.

C. Adaptively weighted lag co-trend graph in frequency domain

Stock return time series feature high-frequency non-stationarity and massive extraneous noise. Direct dynamic graph construction impairs inter-stock correlations, hence we build a lag co-trend graph in the frequency domain. Calculating the trend in the frequency domain results in a more accurate lag order because the interference of irrelevant noise is excluded. At the same time, the calculation in the frequency domain also makes it faster. So we propose to calculate the Lagged Co-Trending Dynamic Graph calculation between stocks based on frequency domain. We have proposed a frequency-domain method with adaptive spectral weighting. For the i-th stock's time series $x_i(t)$, $X_i(k)$ represents the discrete Fourier transform (DFT) [23] of $x_i(t)$. where $X_j^*(k)$ denotes the complex conjugate of $X_j(k)$. The frequency-domain cross-correlation between stocks i and j is defined as:

$$R_{ij}(k) = X_i(k) X_j^*(k), \tag{4}$$

$$\psi_{ij}(k) = \frac{1}{\alpha\sqrt{\phi_{x_i}(k)\phi_{x_j}(k)} + \beta}, \tag{5}$$

$$r_{ij}(\tau) = \frac{1}{T}\sum\nolimits_{k=0}^{T-1} \psi_{ij}(k) R_{ij}(k) e^{j\frac{2\pi}{T}k\tau}, \tag{6}$$

$$L^w = \max_{\tau} r_{ij}(\tau), \tag{7}$$

which captures the coherence across frequency components with a time complexity of $O(N^2T + NT\log(T))$, where $O(NT\log(T))$ corresponds to the FFT precomputation complexity, and $O(N^2T)$ corresponds to the complexity of frequency-domain multiplication. The calculation of time complexity indicates that the adaptive weights perform better in the frequency domain than in the time domain in terms of computational efficiency. An adaptive weighting function $\psi_{ij}(k)$ is introduced to suppress noise-dominant or weak-signal frequency bands. Where $\phi_{x_i}(k) = |X_i(k)|^2$ and $\phi_{x_j}(k) = |X_j(k)|^2$ are the power spectral densities (PSD) [24] of $x_i(t)$ and $x_j(t)$, respectively. The learnable parameter $\alpha, \beta > 0$ balances the spectral magnitude. The resulting cross-correlation tensor spans $r_{ij}(\tau) \in \mathbb{R}^{N\times N\times T}$ for N stocks. Finally, it is again necessary to linearly combine the different windows of the trend decomposition according to (3). We derive $\alpha_{i,j}^w$ to quantify the co-trend similarity between stocks i and j.

D. Feature factor Dynamic Graph

Relationships of different features between stocks are computed at a certain point in time. The features are mainly determined by the feature factors. This relationship represents the stock relationship graph related to the factor at the current time, and $\alpha_{u,v}^t$ represents the feature relationship graph of the stock at time t:

$$\alpha_{u,v}^t = \mathrm{Linear}\Big(\sum\nolimits_{d=1}^{D} u_d v_d\Big). \tag{8}$$

Among them, D represents the number of features, and u and v are the features factors of the two stocks respectively.

E. Sparsification of Implicit graphs

After calculating the similarity between the lagged co-trending graph $\alpha_{i,j}^w$ of stocks and the feature factor graph $\alpha_{u,v}^t$, the graphs are sparsified by a certain percentage to eliminate the interference of irrelevant noise. Let A be either the graph $\alpha_{i,j}^w$ or $\alpha_{u,v}^t$. The definition of graph sparsification is as follows: Let $A \in \mathbb{R}^{m\times n}$, and $p = 0.9$. Define t as the p-th quantile of A, $t = \mathrm{Quantile}(A, p)$. The sparsified matrix $B \in \mathbb{R}^{m\times n}$ is given by:

$$B(i,j) = \begin{cases} A(i,j), & \text{if } A(i,j) > t \\ 0, & \text{if } A(i,j) \le t \end{cases}, \tag{9}$$

for $i = 1, \ldots, m$ and $j = 1, \ldots, n$.

F. Trend decomposition of the time window

Trend decomposition of the stock series is used to alleviate the series' unstationarity. Decompose the time window w into long-term trends, medium-term trends, and short-term fluctuations. Let X be the original time series vector, which can be decomposed into the long-term moving average sequence $Avg(X)$. The medium-term trends decomposition sequence $\mathcal{F}^{-1}(\mathcal{F}(x - avg(x)))_{Max})$, and the short-term residual res. The magnitude of the moving average is an adjustable hyperparameter, usually 5. $\mathcal{F}$ is the Fourier transform, and $\mathcal{F}^{-1}$ is the inverse Fourier transform, which means taking the maximum period of the $\mathcal{F}$ of the X sequence and then performing the $\mathcal{F}^{-1}$. res is the original sequence X minus the moving average sequence and the Fourier decomposition sequence. Sequence decomposition forms the basis of multi-scale and also provides richer features.

G. Temporal feature embedding fusion

Before extracting features using the Mamba block, we first perform a mapping operation through a linear layer. Subsequently, we utilize an encoding module to encode the sequence, which includes positional encoding (P) [25] and temporal encoding (S) [26]. Finally, the encoded sequence is processed by a Normalization layer.

$$H_1 = \text{Linear}(X_t) + P + S, \tag{10}$$

$$u_t = \text{LayerNorm}(H_1). \tag{11}$$

The final output $u_t \in \mathbb{R}^{N\times T\times D}$, where T represents the sequence length and D represents the feature embedding dimension. Here, Linear$(\cdot)$ denotes a linear transformation, and LayerNorm$(\cdot)$ represents the layer normalization operation [27]. In the Mamba [17] S6 module, since the Mamba block effectively integrates temporal features, it can easily capture the correlations within stock sequences. Therefore, we adopt Mamba as the temporal feature extraction block for stock prediction. The output after Mamba model time series extraction is $\hat{h}_t$. Then, we perform graph aggregation on $\hat{h}_t$ based on the feature dynamic graph $\alpha^t_{u,v}$ and the co-trend graph $\alpha^w_{i,j}$. The convolution operation representing the relationships between stocks can be expressed as:

$$\hat{h}_t = \text{Mamba}(u_t), \tag{12}$$

$$Z^{t+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}\hat{h}_t W\right) + \hat{h}_t, \tag{13}$$

where $\tilde{A} = [\alpha^w_{i,j} \circ \alpha^t_{u,v}]$+I, is the self-connection matrix of the aggregated matrix. $\alpha^w_{i,j}$ represents the return rate fluctuation relationship graph of nodes i and j within the same window after the sparse transformation in (9), while $\alpha^t_{u,v}$ represents the sparse dynamic weight graph for nodes u and v under different factors at the same time point after the sparse transformation in (9). The $\circ$ denotes the Hadamard product aggregation. As a new temporal framework, Mamba is superior to Transformer in long sequence modeling and speed, and its performance is not lost to Transformer, so we choose Mamba for temporal feature extraction.

H. Output prediction

After processing by StockLTG, we can obtain the returns at the future time T+2, denoted as $Z^t \in \mathbb{R}^{N\times D}$. Here, we aim to predict the returns at time T+2. The parameter $N \times D$ represents the number of stocks N and the embedding dimension D involved in the training process. Our model employs full-batch training, meaning that all stocks are trained simultaneously in one batch. We are predicting the returns at time T+2. The return of the i-th stock on the T+2 day is defined as $y_i^{t+2} = \frac{p_i^{t+2} - p_i^{t+1}}{p_i^{t+1}}$. Then, we train and make predictions using the model:

$$\hat{y}_i^{t+2} = \text{BN}(\text{FFN}(Z^{t+1})), \tag{14}$$

$$Loss = \sum\nolimits_{i=1}^{N} \text{MSE}(y_i^{t+2}, \hat{y}_i^{t+2}). \tag{15}$$

Here, p_i^{t+1} is the closing price of the i-th stock on the T+1 day. FFN is a multi-layer perceptron (MLP) connection layer with the Leaky ReLU activation function. Z_{t+1} is the output after the GCN convolution operation. BN represents batch normalization [28], and $\hat{y}_i^{t+2} \in \mathbb{R}^{N\times 1}$ is the predicted return of the model for the T+2 th day. We adopt the mean squared error MSE as the loss function.

III. Experiment

A. Dataset

This paper's experimental training data comprises 10 features from S&P 500's basic volume-price data and 14 features from the Chinese stock market, all downloadable and calculable via the Tushare open-source framework. Based on listed sectors' timelines and trading rules, the training data is divided into CSI1000 (representing Chinese stocks) and S&P500 (representing U.S. stocks). The data is split into a training set (June 2021-2023), a validation set (6 months), and a prediction set (2024).

B. Comparison metrics

To assess the model's performance, we adopt four widely used metrics. The Information Coefficient (IC), derived from the average Pearson correlation coefficient, measures the alignment between predictions and actual outcomes. The Rank Information Coefficient (RIC), calculated using the average Spearman coefficient, ranks stocks based on their short-term profit potential. The Information Coefficient Information Ratio (ICIR) and Rank-based Information Coefficient Information Ratio (RICIR) normalize IC and RIC by dividing them by their respective standard deviations. These four metrics collectively evaluate the model's stock selection prowess and overall ranking performance. Additionally, we compute the annualized Sharpe Ratio (SR) for the top-5 stocks, the Precision (Prec) of the top-10 stocks, and the Cumulative Annualized Return (AR) of the top-10 stocks.

C. Comparison Models

The stock hybrid model GATs [29] is an attention-based convolution model combining RNN type, HGAT [30] is an attention-based hypergraph model, RTGCN [31] is a dynamic convolution graph model, ALSTM [32] is a model combining LSTM and attention, StockMixer [33] is a lightweight hybrid stock prediction model, MASTER [34] is a stock prediction model based on market orientation, and DFT [35] is a stock prediction model combining trend decomposition and stock relationships. There is also a TCN [36] convolution-based model for comparison of convolution types.

D. Main results

As shown in Table I, the best and second-best values on each panel are highlighted in bold and underlined, respectively. The $\uparrow$ symbol indicates that bigger is better. On the CSI1000 dataset, StockLTG surpasses the second-best baseline by the following margins: IC increased by 80.00%, ICIR increased by 44.35%, RIC increased by 122.73%, RICIR increased by 87.10%, SR increased by 108.29%, Prec increased by 0.79%, AR increased by 97.17%. This indicates that the model has better performance on small market capitalization, and the fluctuation range of small market capitalization is more drastic. The model has learned this rule. In addition, ICIR and SR metrics are also optimal metrics on the S&P500 dataset.

TABLE I
MODEL PERFORMANCE COMPARISON (T+2). TOP PANEL: CSI1000; BOTTOM PANEL: S&P500.

Model	IC ↑	ICIR ↑	RIC ↑	RICIR ↑	SR ↑	Prec ↑	AR ↑
GATs	0.012	0.095	0.029	0.195	0.811	0.497	0.274
HGAT	0.005	0.030	0.014	0.074	1.078	0.503	0.389
RTGCN	0.003	0.043	−0.004	−0.032	0.685	0.475	0.131
ALSTM	0.009	0.076	0.014	0.097	1.134	0.485	0.307
StockMixer	−0.001	−0.012	0.006	0.049	0.807	0.497	0.207
MASTER	0.015	0.124	0.022	0.155	1.134	0.480	0.389
DFT	0.001	0.004	0.001	0.006	−0.918	0.474	−0.188
TCN	0.012	0.098	0.023	0.173	−0.223	0.481	−0.049
StockLTG	**0.027**	**0.179**	**0.049**	**0.290**	**2.365**	**0.507**	**0.768**
GATs	0.011	0.088	−0.002	−0.016	**2.529**	0.517	0.422
HGAT	0.003	0.021	0.004	**0.026**	1.068	0.517	0.185
RTGCN	−0.011	−0.112	−0.018	−0.106	−0.134	0.504	0.059
ALSTM	0.002	0.018	−0.004	−0.036	2.294	**0.531**	**0.479**
StockMixer	−0.006	−0.043	−0.002	−0.014	2.092	0.522	0.242
MASTER	0.007	0.055	**0.004**	0.025	0.391	0.526	0.173
DFT	0.002	0.012	0.000	0.001	0.711	0.501	0.102
TCN	0.013	0.089	0.001	−0.003	1.890	0.513	0.422
StockLTG	**0.014**	**0.128**	−0.001	−0.006	1.399	0.522	0.292

E. Module ablation experiment

Ablation experiments were performed on each module of the StockLTG of the CSI1000 dataset. All ablation experiments kept the same parameters except for the ablation module.

TABLE II
ABLATION EXPERIMENTS ON CSI1000 DATASET AT TIME T+2. FEA. REFERS TO THE FEATURE FACTOR GRAPH, AND LAG. TR. REPRESENTS LAGGED CO-TRENDING DYNAMIC GRAPH. DEC. REPRESENTS TREND DECOMPOSITION. IND. STANDS FOR INDUSTRY (STATIC) GRAPH.

Model Ablation	IC	ICIR	RIC	RICIR
StockLTG (ours)	**0.027**	**0.179**	**0.049**	**0.290**
w/o Fea.	0.018	0.123	0.027	0.170
w/o Lag. Tr.	0.018	0.123	0.027	0.176
w/o Lag. Tr. + Fea.	0.019	0.126	0.028	0.188
w/o Fea. + Dec.	0.017	0.124	0.029	0.190
Mamba + Ind.	0.001	0.001	-0.010	-0.065
Mamba	0.012	0.099	0.022	0.161

Ablation results (Table II) show that removing the feature-factor or lagged co-trending graph markedly hurts performance, as together they encode rich stock-correlation information that jointly characterizes market dynamics. Dropping either the feature-factor graph (stock-time interactions) or the multi-scale module (temporal dynamics) also weakens representation capacity. Moreover, Mamba performs poorly with the static industry graph, whose noise distorts stock relationships. Our implicit relational modeling outperforms the traditional industry-graph approach by capturing latent stock dependencies more accurately.

a) Hyperparameter Analysis and Correlation Visualization: The following is an analysis of lag order and using sequence length hyperparameters. It can be seen from the left graph of Fig. 2 that fully mining the lag order of stocks is helpful to improve the value of IC. The CSI1000 stocks have better results at lag order 1 and 6, while S&P500 has better results at lag order 3, indicating that different markets may have different optimal lag orders. The influence of the backtest length on IC can be seen from the right figure, which has a better effect when the backtest length is 30 and 50. In Fig. 3, the lag relationship between stocks along with the

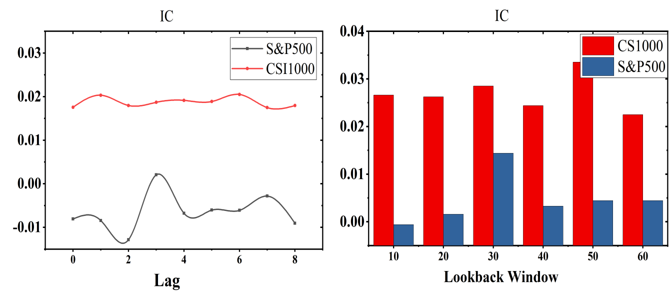

Fig. 2. The influence of lag order and lookback window (sequence length) on IC.

sliding window is analyzed. $corr(i, j)$ is the maximum lagged cross correlation between stock i and stock j. For example $corr(0, 2)$ the second stock is ahead of the 0th stock around the sliding window 40-64. It can be seen from the graph that some of the lag relationships of stocks persist, and some only have a short-term lag. It is necessary to model the lag relationship to explain the stock relationship. Sliding window and lagged

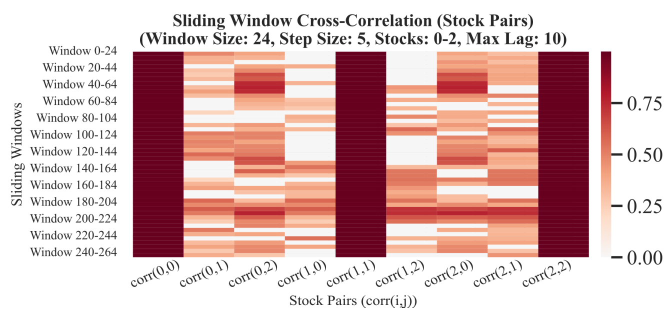

Fig. 3. Plot of lag relationship with sliding window.

correlation visualization help to understand the overall lagged trend change relationship between stocks.

F. Backtest

We backtested on the CSI1000 dataset on the Qlib platform [37]. The limit of rise or fall is 0.1, the transaction fee is 0.0005, and the minimum transaction cost is 2.5. Backtested in 2024, we adopted the Top30-Drop5 strategy to simulate daily trading. Fig. 4 presents the backtest results of the CSI1000 for 2024. It can be seen that the performance of the backtest is that in the rising period of 2024, the model is in the lead from February to March, and in the bear market from May to September, our model is directly ahead of other models in the bull period from September to November.

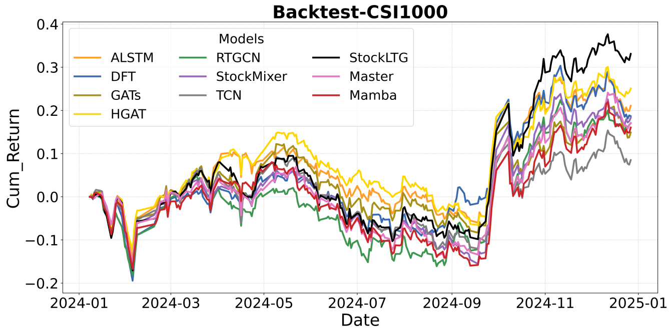

Fig. 4. The performance of StockLTG and other models in backtesting on the CSI1000.

IV. Summary

This paper proposes StockLTG, a multi-scale stock trend forecasting model based on return rate decomposition and lagged co-trending dynamic graph networks, addressing the core limitations of traditional models—overreliance on static attributes and neglect of interactions between individual stock returns and lagged relationships. The model decomposes stock return series into long-term trends, medium-term cycles, and short-term fluctuations to construct multi-scale lagged co-trending graphs, fuses these with feature factor graphs to form heterogeneous graphs, and computes correlation dynamic graphs in the frequency domain with adaptive weighting, integrating Mamba blocks and Multi-scale GCN convolution for feature extraction and aggregation.

References

[1] Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *27th ACM Int. Conf.*, 2018.

[2] B. Xie, R. Passonneau, L. Wu, and G. G. Creamer, "Semantic frames to predict stock price movement," *ERN: Other Econometric Modeling: Capital Markets - Asset Pricing (Topic)*, 2013.

[3] J. Ye, J. Zhao, K. Ye, and C. Xu, "Multi-graph convolutional network for relationship-driven stock movement prediction," in *2020 25th Int. Conf. Pattern Recognit. (ICPR)*, 2020, pp. 6702–6709.

[4] 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 *AAAI Conf. Artif. Intell.*, 2021.

[5] Z. Cao, J. Xu, C. Dong, P. Yu, and T. Bai, "MATCC: A novel approach for robust stock price prediction incorporating market trends and cross-time correlations," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manag. (CIKM 2024)*, E. Serra and F. Spezzano, Eds. ACM, 2024, pp. 187–196.

[6] Q. Xiang, Z. Chen, Q. Sun, and R. Jiang, "Rsap-dfm: Regime-shifting adaptive posterior dynamic factor model for stock returns prediction," in *Proc. 33rd Int. Joint Conf. Artif. Intell. (IJCAI-24)*, 2024.

[7] Y. L. Hsu, Y. C. Tsai, and C. T. Li, "Fingat: Financial graph attention networks for recommending top-k profitable stocks," *IEEE Trans. Knowl. Data Eng.*, 2023.

[8] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *ArXiv e-prints*, p. arXiv:1710.10903, Oct. 2017.

[9] X. Ma, X. Li, W. Feng, L. Fang, and C. Zhang, "Dynamic graph construction via motif detection for stock prediction," *Inf. Process. Manag.*, vol. 60, no. 6, p. 20, 2023.

[10] Z. Lei, C. Zhang, Y. Xu, and X. Li, "Dr-gat: Dynamic routing graph attention network for stock recommendation," *Inf. Sci.*, vol. 654, no. 0, p. 19, 2024.

[11] Y. Zhao, H. Du, Y. Liu, S. Wei, X. Chen, F. Zhuang, Q. Li, and G. Kou, "Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks," *IEEE Trans. Autom. Control.*, vol. 35, no. 8, p. 13, 2023.

[12] D. Matsunaga, T. Suzumura, and T. Takahashi, "Exploring graph neural networks for stock market predictions with rolling window analysis," *Papers*, 2019.

[13] Y. Yang, Z. Wei, Q. Chen, and L. Wu, "Using external knowledge for financial event prediction based on graph neural networks," in *Conf. Inf. Knowl. Manag.*, 2019.

[14] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognit.*, vol. 121, 2022.

[15] P. Zhu, Y. Li, Y. Hu, S. Xiang, Q. Liu, D. Cheng, and Y. Liang, "Mci-gru: Stock prediction model based on multi-head cross-attention and improved gru," *ArXiv*, vol. abs/2410.20679, 2025.

[16] V. Profillidis and G. Botzoris, "Chapter 5 - statistical methods for transport demand modeling," in *Modeling of Transport Demand*, V. Profillidis and G. Botzoris, Eds. Elsevier, 2019, pp. 163–224.

[17] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *CoRR*, vol. abs/2312.00752, 2023.

[18] M. Azaria and D. Hertz, "Time delay estimation by generalized cross correlation methods," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 2, pp. 280–285, 1984.

[19] L. Zhang and X. Wu, "On cross correlation based-discrete time delay estimation," *IEEE*, 2005.

[20] J. Lai, M. Chen, and J. Cai, "Time-lag and edge features incorporated graph attention network for stock movement prediction," in *2025 IEEE Int. Joint Conf. Neural Netw. (IJCNN), Rome, Italy, Jun. 30 - Jul. 5, 2025*. IEEE, 2025, pp. 1–8.

[21] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *NeurIPS*, 2021.

[22] A. Albert and J. A. Anderson, "On the existence of maximum likelihood estimates in logistic regression models," *Biometrika*, vol. 71, no. 1, pp. 1–10, 1984.

[23] K. Cattermole, "The fourier transform and its applications," *Electronics and Power*, vol. 11, no. 10, pp. 357–, 1965.

[24] T. T. Georgiou, "Distances between power spectral densities," *Mathematics*, 2006.

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv*, 2017.

[26] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," 2024.

[27] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[29] U. Gupta, V. Bhattacharjee, and P. S. Bishnu, "Stocknet—gru based stock index prediction," *Expert Syst. Appl.*, vol. 207, p. 117986, 2022.

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

[31] Z. Zheng, J. Shao, J. Zhu, and H. T. Shen, "Relational temporal graph convolutional networks for ranking-based stock prediction," in *2023 IEEE 39th Int. Conf. Data Eng. (ICDE)*, 2023, pp. 123–136.

[32] Q. Wang and Y. Hao, "Alstm: An attention-based long short-term memory framework for knowledge base reasoning," *Neurocomputing*, vol. 399, pp. 342–351, 2020.

[33] Fan, Jinyong, Shen, and Yanyan, "Stockmixer: A simple yet strong mlp-based architecture for stock price forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 8, 2024, pp. 8389–8397.

[34] T. Li, Z. Liu, Y. Shen, X. Wang, H. Chen, and S. Huang, "Master: Market-guided stock transformer for stock price forecasting," 2023.

[35] C. Dong, Z. Cao, S. K. Zhou, and J. Liu, "Dft: A dual-branch framework of fluctuation and trend for stock price prediction," 2024.

[36] V. K. Shaojie Bai, J. Zico Kolter, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018.

[37] X. Yang, W. Liu, D. Zhou, J. Bian, and T.-Y. Liu, "Qlib: An ai-oriented quantitative investment platform," 2020.