

MASTER: Market-Guided Stock Transformer for Stock Price Forecasting

Tong Li^{1*}, Zhaoyang Liu², Yanyan Shen^{1†}, Xue Wang², Haokun Chen², Sen Huang²

¹ Shanghai Jiao Tong University, ² Alibaba Group
 {2017lt, shenyy}@sjtu.edu.cn, {jingmu.lzy, xue.w, hankel.chk, huangsen.huang}@alibaba-inc.com

Abstract

Stock price forecasting has remained an extremely challenging problem for many decades due to the high volatility of the stock market. Recent efforts have been devoted to modeling complex stock correlations toward joint stock price forecasting. Existing works share a common neural architecture that learns temporal patterns from individual stock series and then **mixes up temporal representations to establish stock correlations**. However, they only consider time-aligned stock correlations stemming from all the input stock features, which suffer from two limitations. First, stock correlations often occur **momentarily and in a cross-time manner**. Second, the feature effectiveness is **dynamic with market variation**, which affects both the stock sequential patterns and their correlations. To address the limitations, this paper introduces MASTER, a **MArket-Guided Stock TransformER**, which models the momentary and cross-time stock correlation and leverages market information for **automatic feature selection**. MASTER elegantly tackles the complex stock correlation by alternatively engaging **in intra-stock and inter-stock** information aggregation. Experiments show the superiority of MASTER compared with previous works and visualize the captured realistic stock correlation to provide valuable insights.

Introduction

Stock price forecasting, which utilizes historical data collected from the stock market to predict future trends, is a vital technique for profitable stock investment. Unlike stationary time series that often exhibit regular patterns such as periodicity and steady trends, the dynamics in the stock price series are intricate because stock prices fluctuate subject to multiple factors, including **macroeconomic factors, capital flows, investor sentiments, and events**. The mixing of factors interweaves the stock market as a correlated network, making it difficult to **precisely predict the individual behavior of stocks without taking other stocks into account**.

Most previous works (Feng et al. 2019; Xu et al. 2021; Wang et al. 2021, 2022; Wang, Qu, and Chen 2022) in the field of stock correlation have relied on predefined concepts, relationships, or rules and established a **static correlation graph**, e.g., stocks in the same industry are connected to each

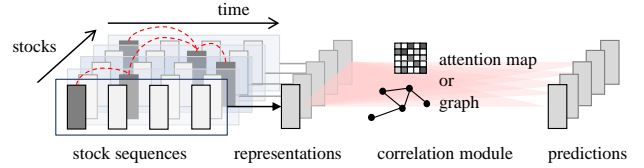


Figure 1: The framework of existing works. The dashed lines represent the underlying momentary and cross-time stock correlations, which reside between some $(stock_1, time_1)$, $(stock_2, time_2)$ pairs.

other. While these methods provide insights into the relations between stocks, they do not account for the real-time correlation of stocks. For example, different stocks within the same industry can experience opposite price movements on a particular day. Additionally, the pre-defined relationships may not be generalizable to new stocks in an evolving market where events such as company listing, delisting, or changes in the main business happen normally. Another line of research (Yoo et al. 2021) follows the Transformer architecture (Vaswani et al. 2017), and use the **self-attention mechanism to compute dynamic stock correlations**. This data-driven manner is more flexible and applicable to the time-varying stock sets in the market. Despite different schemes for establishing stock correlations, the existing methods generally follow a common two-step computation flow. As depicted in Figure 1, the first step is using a sequential encoder to summarize the historical sequence of stock features, and obtain stock representation, and the second step is to refine each stock representation by aggregating information from correlated stocks using graph encoders or attention mechanism. However, such a flow suffers from two limitations.

First, existing works distill an overall stock representation and blur the time-specific details of stock sequence, leading to weakness in modeling the de-facto stock correlations, which often occurs *momentarily* and in a *cross-time* manner (Bennett, Cucuringu, and Reinert 2022). To be specific, the stock correlation is highly dynamic and may reside in misaligned time steps rather than holding true through the whole lookback period. This is because the dominating factors of stock prices constantly change, and different stocks **may react to the same factors with different delays**. For in-

*This work was done during her internship at Alibaba Group.

†Corresponding author.

stance, upstream companies' stock prices may react faster to a shortage of raw materials than those of downstream companies, and individual stocks exhibit a lot of catch-up and fall-behind behaviors.

Since the stock correlation may underlie between every stock pair and time pair, a straightforward way to simulate the momentary and cross-time correlation is to gather the $\tau \times |\mathcal{S}|$ feature vectors for pair-wise attention computation, where τ is the lookback window length and \mathcal{S} is the stock set. However, in addition to the increased computational complexity, this approach faces practical difficulties because the stock forecasting task is in intense data hunger. Intuitively, there are only around 250 trading days per year, producing limited observations on stocks. When the model adopts such a large attention field with insufficient training samples, it often struggles to optimize and may even fall into suboptimal solutions. Although clustering approaches like local sensitive hashing (Kitaev, Kaiser, and Levskaya 2020) have been proposed to reduce the size of the attention field, they are sensitive to initialization, which is a fatal issue in a data-hungry domain like stock forecasting. To address these challenges, we propose a novel stock transformer architecture specifically designed for stock price forecasting. Rather than directly modeling the $\tau \times |\mathcal{S}|$ attention field or using clustering-based approximation methods, our model aggregates information from different time steps and different stocks alternately to model realistic stock correlation and facilitate model learning.

Another limitation of existing works is that they ignore the impact of varying market status. In long-term practice with the market variation, one essential observation by investors is that the features come into effect and expire dynamically. The effectiveness of features has an influence on both the intra-stock sequential pattern and the stock correlation. For instance, in a bull market, the correlations among stocks are more significant due to the investors' optimism. Traditional investors repeatedly conduct statistical examination on to select effective feature, which is exhaustive and face a gap when integrated with learning-based methods. To save the human efforts, we are motivated to equip our stock transformer with a novel gating mechanism, which incorporates the market information to perform automatically feature selection. We name the proposed method MASTER, standing for **MA**rk**ET** Guided **ST**ock **TR**ansform**ER**. To summarize, our main contributions are as follows.

- We propose a novel stock transformer for stock price forecasting to effectively capture the stock correlation. To the best of our knowledge, we are the first to mine the momentary and cross-time stock correlation with learning-based methods.
- We introduce a novel gating mechanism that integrates market information to automatically select relevant features and adapt to varying market scenarios.
- We conducted experiments to validate the designs of our proposed method and demonstrated its superiority compared to baselines. The visualization results provided valuable insights into the real-time dynamics of stock correlations.

Methodology

Problem Formulation

The indicators of each stock $u \in \mathcal{S}$ are collected at every time step $t \in [1, \tau]$ to form the feature vector $x_{u,t} \in \mathbf{R}^F$. Following existing works on stock market analysis (Feng et al. 2018; Sawhney et al. 2020; Huynh et al. 2023), we focus on the prediction of the change in stock price rather than the absolute value. The return ratio, which is the relative close price change in d days, is $\tilde{r}_u = (c_{u,\tau+d} - c_{u,\tau+1})/c_{u,\tau+1}$, where $c_{u,t}$ is the closing price of stock u at time step t , and d represents the predetermined prediction interval. The return ratio normalizes the market price variety between different stocks in comparison to the absolute price change. Since stock investment is to rank and select the most profitable stocks, we perform daily Z-score normalization of return ratio to encode the label with the rankings, $r_u = \text{Norm}_{\mathcal{S}}(\tilde{r}_u)$, as in previous work (Yang et al. 2020).

Definition 1 (Stock Price Forecasting) Given stock features $\{x_{u,t}\}_{u \in \mathcal{S}, t \in [1, \tau]}$, the stock price forecasting is to jointly predict the future normalized return ratio $\{r_u\}_{u \in \mathcal{S}}$.

Overview

Figure 2 depicts the architecture of our proposed method MASTER, which consists of five steps. (1) **Market-Guided Gating**. We construct a vector representing the current market status m_τ and leverage it to rescale feature vectors by a gating mechanism, achieving market-guided feature selection. (2) **Intra-Stock Aggregation**. Within the sequence of each stock, at each time step, we aggregate information from other time steps to generate a local embedding that preserves the temporal local details of the stock while collecting all important signals along the time axis. The local embedding $h_{u,t}$ will serve as relays and transport the collected signals to other stocks in subsequent modules. (3) **Inter-Stock Aggregation**. At each time step, we compute stock correlation with attention mechanism, and each stock further aggregates the local embedding of other stocks. The aggregated information $z_{u,t}$, which we refer to as temporal embedding, contains not only the information of the momentarily correlated stocks at t , but also preserves the personal information of u . (4) **Temporal Aggregation**. For each stock, the last temporal embedding queries from all historical temporal embedding and produce a comprehensive stock embedding e_u . (5) **Prediction**. The comprehensive stock embedding is sent to prediction layers for label prediction. We elaborate on the details of MASTER step by step in the following subsections.

Market-Guided Gating

Market Status Representation First, we propose to combine information from two aspects into a vector m_τ to give an abundant description of the current market status. (1) **Market index price**. The market index price is a weighted average of the prices of a group of stocks \mathcal{S}' by their share of market capitalization. \mathcal{S}' is typically composed of top companies with the most market capitalization, representing a particular market or sector, and may differ from user-

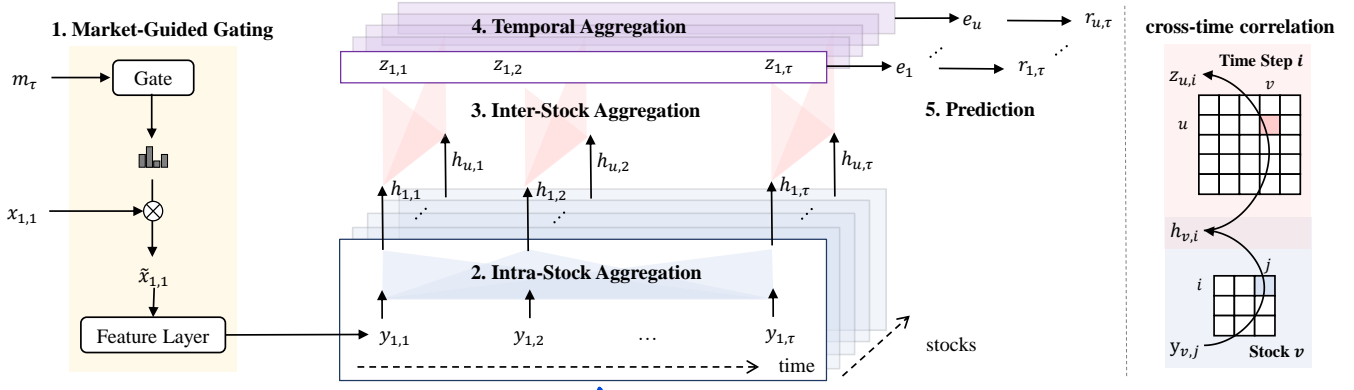


Figure 2: Overview of the MASTER framework.

interested stocks in investing \mathcal{S} . We include both the current market index price at τ and the historical market index prices, which is described by the average and standard deviation in the past d' days to reveal the price fluctuations. Here, d' specifies the referable interval length to introduce historical market information in applications. (2) Market index trading volume. The trading volumes of \mathcal{S}' reveals the investors involvement, reflecting the activity of the market. We include the average and standard deviation of market index trading volume in the past d' days, to reveal the actual size of the market. \mathcal{S}' and d' are identical to the aforementioned definitions. Now we present the market-guided stock price forecasting task.

Definition 2 (Market-Guided Stock Price Forecasting)

Given $\{x_{u,t}\}_{u \in \mathcal{S}, t \in [1, \tau]}$ and the constructed market status vector m_τ , market-guided stock price forecasting is to jointly predict the future normalized return ratio $\{r_u\}_{u \in \mathcal{S}}$.

Gating Mechanism The gating mechanism generates one scaling coefficient for each feature dimension to enlarge or shrink the magnitude of the feature, thereby emphasizing or diminishing the amount of information from the feature flowing to the subsequent modules. The gating mechanism is learned by the model training, and the coefficient is optimized by how much the feature contributes to improve forecasting performance, thus reflect the feature effectiveness.

Given the market status representation $m_\tau, |m_\tau| = F'$, we first use a single linear layer to transform m_τ into the feature dimension $F' = |x_{u,t}|$. Then, we perform Softmax along the feature dimension to obtain a distribution.

$$\alpha(m_\tau) = F \cdot \text{softmax}_\beta(W_\alpha m_\tau + b_\alpha),$$

where W_α, b_α are learnable matrix and bias, β is the temperature hyperparameter controlling the sharpness of the output distribution. Softmax compels a competition among features to distinguish the effective ones and ineffective ones. Here, a smaller temperature β encourages the distribution to focus on certain dimension and the gating effect is stronger while a larger β makes the distribution incline to even and the gating effect is weaker. Note that we enlarge the value at each dimension by F times as the scaling coefficient. This operation

compare the generated distribution with a uniform distribution where each dimension is $1/F$, to determine whether to enlarge or shrink the value. The intuition to generate coefficients from m_τ is that the effectiveness of features are influenced by market status. For example, if the model learns moving average (MA) factor is useful during volatile market periods, it will emphasize MA when the market becomes volatile again. Under the same m_τ , α are shared for $\{x_{u,t}\}$, $u \in \mathcal{S}, t \in [1, \tau]$, in that we incorporate the most recent market status to perform unified feature selection. The rescaled feature vectors are $\tilde{x}_{u,t} = \alpha(m_\tau) \circ x_{u,t}$, where \circ is the Hadamard product.

Intra-Stock Aggregation

In MASTER, we use intra-stock aggregation followed by inter-stock aggregation to break down the large and complex attention field. Although the entire market is complicated with diverse behaviours of individual stocks, the patterns of a specific stock tend to be relatively continuous. Therefore, we perform intra-stock aggregation first due to its smaller attention field and simpler distribution. In our proposed intra-stock aggregation, the feature at each time step aggregate information from other time steps and form a local embedding. Compared with existing works which initially mix the feature sequence into one representation (Yoo et al. 2021), we maintain a sequence of local embedding which are advised with the important signals in sequence through intra-stock aggregation while reserve the local details.

We first send the rescaled feature vectors to a feature encoder and transform them into the embedding space, $y_{u,t} = f(\tilde{x}_{u,t}), |y_{u,t}| = D$. We simply use a single linear layer as $f(\cdot)$. Then, we apply a bi-directional sequential encoder to obtain the local output at each time step t . Inspired by the success of transformer-based models in modeling sequential patterns, we instantiate the sequential encoder with a single-layer transformer encoder (Vaswani et al. 2017). Each feature vector at a particular time step is treated as a token, and we add a fixed D -dimensional sinusoidal positional encoding p_t to mark the chronically order in the look back window.

$$Y_u = \parallel_{t \in [1, \tau]} \text{LN}(f(\tilde{x}_{u,t}) + p_t),$$

where \parallel denotes the concatenation of vectors and LN the

layer normalization. Then, the feature embedding at each time step queries from all time steps in the stock sequence. We introduce multi-head attention mechanisms, denoted as $\text{MHA}(\cdot)$, with N_1 heads to perform different aggregations in parallel. We also utilize feed forward layers, $\text{FFN}(\cdot)$, to fuse the information obtained from the multi-head attention.

$$Q_u^1 = W_Q^1 Y_u, \quad K_u^1 = W_K^1 Y_u, \quad V_u^1 = W_V^1 Y_u,$$

$$H_u^1 = \parallel_{t \in [1, \tau]} h_{u,t} = \text{FFN}^1(\text{MHA}^1(Q_u^1, K_u^1, V_u^1) + Y_u),$$

where FFN is a two-layer MLP with ReLU activation and residual connection. As a result, the local embedding $h_{u,t}$ both reserve the local details and encode indicative signals from other time steps.

Inter-Stock Aggregation

Then, we consider aggregating information from correlated stocks. Compared with existing works that distill an overall stock correlation, we establish a series of momentary stock correlation corresponding to every time step. Instead of using pre-defined relationships that face a mismatch with the proximity of real-time stock movements, we propose to mine the asymmetric and dynamic inter-stock correlation via attention-mechanism. The quality of the correlation will be measured by its contribution to improving the forecasting performance, and automatically optimized by the model training process.

Specifically, at each time step, we gather the local embedding of all stocks $H_t^2 = \parallel_{u \in \mathcal{S}} h_{u,t}$ and perform multi-head attention mechanism with N_2 heads.

$$Q_t^2 = W_Q^2 H_t^2, \quad K_t^2 = W_K^2 H_t^2, \quad V_t^2 = W_V^2 H_t^2,$$

$$Z_t = \parallel_{u \in \mathcal{S}} z_{u,t} = \text{FFN}^2(\text{MHA}^2(Q_t^2, K_t^2, V_t^2) + H_t^2).$$

With the residual connection of FFN , the temporal embedding $z_{u,t}$ is encoded with both the information from momentarily correlated stocks and the personal information of stock u itself. Our stock transformer is able to model the cross-time correlation of stocks, as shown in Figure 2 (Right). The local details of $y_{v,j}$ can first be conveyed to $h_{v,i}$ by the intra-stock aggregation of stock v , and then transmitted to $z_{u,i}$ by inter-stock aggregation at time step i , hence modeling the correlation from any (v, j) to (u, i) . We further visualize and explain the captured cross-time correlation in the experiments section.

Temporal Aggregation

In contrast with existing works which obtain one embedding for each stock after modeling stock correlation (Feng et al. 2019), our approach involves producing a series of temporal embedding $z_{u,t}, t \in [1, \tau]$. Each $z_{u,t}$ is encoded with information from stocks that are momentarily correlated with (u, t) . To summarize the obtained temporal embeddings and obtain a comprehensive stock embedding e_u , we employ a temporal attention layer along the time axis. We use the latest temporal embedding $z_{u,\tau}$ as the query vector, and compute the attention score $\lambda_{u,t}$ in a hidden space with transformation matrix W_λ ,

$$\lambda_{u,t} = \frac{\exp(z_{u,\tau}^T W_\lambda z_{u,t})}{\sum_{i \in [1, \tau]} \exp(z_{u,\tau}^T W_\lambda z_{u,i})}, \quad e_u = \sum_{t \in [1, \tau]} \lambda_{u,t} z_{u,t}.$$

Prediction and Training

Finally, the stock embedding e_u is fed into a predictor $g(\cdot)$ for label regression. We use a single linear layer as the predictor, and the forecasting quality is measured by the MSE loss. In each batch, MASTER is jointly optimized for all $u \in \mathcal{S}$ on a particular prediction date. And a training epoch is composed of multiple batches correspond to different prediction dates in the training set.

$$\hat{r}_u = g(e_u), \quad L = \sum_{u \in \mathcal{S}} \text{MSE}(r_u, \hat{r}_u).$$

Discussions

Relationships with Existing Works Modeling stock correlations has long been an indispensable research direction for stock price prediction. Today, many researchers and quantitative analysts, still opt for linear models, support vector machines, and tree-based methods for stock price forecasting (Nugroho, Adji, and Fauziati 2014; Chen and Guestrin 2016; Kamble 2017; Xie et al. 2013; Li et al. 2015; Piccolo 1990). The aggregation of correlation information within and between stocks is often achieved through feature engineering, which relies heavily on manual expertise and constantly faces the risk of factor decay. Inspired by the success of neural sequential data analysis, researchers are driven to take into account the stock feature sequences and learn the temporal correlation automatically. They design various sequential models, such as RNN-based (Feng et al. 2019; Sawhney et al. 2021; Yoo et al. 2021; Huynh et al. 2023), CNN-based (Wang et al. 2021), and attention-based models (Liu et al. 2019; Ding et al. 2020), to mine the internal temporal dynamics of a stock. Recent research focus on the modeling of stock correlation, which add a correlation module in posterior to the sequential model as illustrated in Figure 1. They propose to use graph-based (Feng et al. 2019; Xu et al. 2021; Wang et al. 2021, 2022), hypergraph-based (Sawhney et al. 2021; Huynh et al. 2023) and attention-based (Yoo et al. 2021; Xiang et al. 2022) modules to build the overall stock correlation and perform joint prediction. Our MASTER is dedicated to momentary and cross-time stock correlation mining. To do so, we develop a novel model architecture as in Figure 2 that is genuinely different from all existing methods. Furthermore, MASTER is specialized for stock price forecasting, which is distinct in data form and task properties from existing transformer-based models in spatial-temporal data (Bulat et al. 2021; Cong et al. 2021; Xu et al. 2020; Li et al. 2023) or multivariate time series domains (Zhang and Yan 2022; Nie et al. 2022).

Complexity Analysis We now analyze the computation complexity of our proposed method. Let $M = |\mathcal{S}|$, the market-guided gating rescale $M \times \tau$ feature vectors of dimension F . In intra-stock aggregation, the calculation amount of pair-wise attention is τ^2 for each stock at each attention head. In inter-stock aggregation, the calculation amount is M^2 at each time step and each attention head. In temporal aggregation, we compute τ attention scores for each stock. The overall computation complexity is $O(FM\tau + N_1 M \tau^2 D^2 + N_2 M^2 \tau D^2 + M \tau D^2)$,

where $M \gg \tau$. Therefore, MASTER is of $O(N_2 M^2 \tau D^2)$ time complexity. Compared with directly operating on the $M \times \tau$ attention field with N attention heads, which is in $O(N M^2 \tau^2 D^2)$, we reduce the computation cost by about τ times and achieve modeling cross-time correlations between stocks more efficiently. The overall parameters to be trained in MASTER are transformation matrices $W_Q^1, W_K^1, W_V^1, W_Q^2, W_K^2, W_V^2, W_\lambda$, which is in shape $D \times D$, and parameters in MLP layers $\alpha, f, \text{FFN}^1, \text{FFN}^2$ and g .

Experiments

In this section, we conduct experiments to answer the following four research questions:

- **RQ1** How is the overall performance of MASTER compared with state-of-the-art methods?
- **RQ2** Is the proposed stock transformer architecture effective for stock price forecasting?
- **RQ3** How do hyper-parameter configurations affect the performance of MASTER?
- **RQ4** What insights on the stock correlation can we get through visualizing the attention map?

Datasets We evaluate our framework on the Chinese stock market with CSI300 and CSI800 stock sets. CSI300 and CSI800 are two stock sets containing 300 and 800 stocks with the highest capital value on the Shanghai Stock Exchange and the Shenzhen Stock Exchange. The dataset contains daily information ranging from 2008 to 2022 of CSI300 and CSI800. We use the data from Q1 2008 to Q1 2020 as the training set, data in Q2 2020 as the validation set, and the last ten quarters, i.e., Q3 2020 to Q4 2022, are reserved as the test set. We apply the public Alpha158 indicators (Yang et al. 2020) to extract stock features from the collected data. The lookback window length τ and prediction interval d are set as 8 and 5 respectively. For market representation, we constructed 63 features with CSI300, CSI500 and CSI800 market indices, and refereable interval $d' = 5, 10, 20, 30, 60$.

Baselines We compare the performance of MASTER with several stock price forecasting baselines from different categories. • XGBoost (Chen and Guestrin 2016): A decision-tree based method. According to the leaderboard of Qlib platform (Yang et al. 2020), it is one of the strongest baselines. • LSTM (Graves and Graves 2012), GRU (Cho et al. 2014), TCN (Bai, Kolter, and Koltun 2018), and Transformer (Vaswani et al. 2017): Sequential baselines that leverage vanilla LSTM/GRU/Temporal convolutional network/-Transformer along the time axis for stock price forecasting. • GAT (Veličković et al. 2017): A graph-based baseline, which first use sequential encoder to gain stock presentation and then aggregate information by graph attention networks¹. • DTML (Yoo et al. 2021): A state-of-the-art stock correlation mining method, which follows the framework in Figure 1. DTML adopts the attention-mechanism to mine

the dynamic correlation among stocks and also incorporates the market information into the modeling.

Evaluation We adopt both ranking metrics and portfolio-based metrics to give a thorough evaluation of the model performance. Four ranking metrics, Information Coefficient (IC), Rank Information Coefficient (RankIC), Information Ratio based IC (ICIR) and Information Ratio based RankIC (RankICIR) are considered. IC and RankIC are the Pearson coefficient and Spearman coefficient averaged at a daily frequency. ICIR and RankICIR are normalized metrics of IC and RankIC by dividing the standard deviation. Those metrics are commonly used in literature (e.g., Xu et al. 2021 and Yang et al. 2020) to describe the performance of the forecasting results from the value and rank perspectives. Furthermore, we employ two portfolio-based metrics to compare the investment profit and risk of each method. We simulate daily trading using a simple strategy that selects the top 30 stocks with the highest return ratio and reports the Excess Annualized Return (AR) and Information Ratio (IR) metrics. AR measures the annual expected excess return generated by the investment, while IR measures the risk-adjusted performance of an investment.

Implementation We implemented MASTER² with PyTorch and build our methods based on the open-source quantitative investment platform Qlib (Yang et al. 2020). For DTML, we implement it based on the original paper since there is no official implementation publicly. For other baselines, we use their Qlib implementations. For hyperparameters of each baseline method, the layer number and model size are tuned from $\{1, 2, 3\}$ and $\{128, 256, 512\}$ respectively. The learning rate lr is tuned among $\{10^{-i}\}_{i \in \{3, 4, 5, 6\}}$, and we selected the best hyperparameters based on the IC performance in the validation stage. For hyperparameters of MASTER, we tune the model size D and learning rate lr among the same range as the baselines, and the final selection is $D=256, lr=10^{-5}$ for all datasets; we set $N_1=4, N_2=2$ for all datasets and $\beta=5$ and $\beta=2$ for CSI300 and CSI800 respectively. More implementation details of baseline methods are summarized in the supplementary materials. Each model is trained for at most 40 epochs with early stopping. All the experiments are conducted on a server equipped with Intel(R) Xeon(R) Platinum 8163 CPU, 128GB Memory, and a Tesla V100-SXM2 GPU (16GB Memory). Each experiment was repeated 5 times with random initialization and the average performance was reported.

Overall Performance (RQ1)

The overall performance are reported in Table 1 MASTER achieves the best results on 6/8 of the ranking metrics, and consistently outperforms all benchmarks in the portfolio-based metrics. In particular, MASTER achieve 13% improvements in ranking metrics and 47% improvements in portfolio-based metrics compared to the second-best results on the average sense. Note that ranking metrics are computed with the whole set and portfolio-based metrics

¹More discussion is provided in the supplementary materials.

²Code and supplementary materials are at <https://github.com/SJTU-Quant/MASTER>

Dataset	Model	IC	ICIR	RankIC	RankICIR	AR	IR
CSI300	XGBoost	0.051 \pm 0.001	0.37 \pm 0.01	0.050 \pm 0.001	0.36 \pm 0.01	<u>0.23 \pm 0.03</u>	1.9 \pm 0.3
	LSTM	0.049 \pm 0.001	<u>0.41 \pm 0.01</u>	0.051 \pm 0.002	0.41 \pm 0.03	<u>0.20 \pm 0.04</u>	<u>2.0 \pm 0.4</u>
	GRU	0.052 \pm 0.004	<u>0.35 \pm 0.04</u>	0.052 \pm 0.005	0.34 \pm 0.04	0.19 \pm 0.04	1.5 \pm 0.3
	TCN	0.050 \pm 0.002	0.33 \pm 0.04	0.049 \pm 0.002	0.31 \pm 0.04	0.18 \pm 0.05	1.4 \pm 0.5
	Transformer	0.047 \pm 0.007	0.39 \pm 0.04	0.051 \pm 0.002	<u>0.42 \pm 0.04</u>	0.22 \pm 0.06	2.0 \pm 0.4
	GAT	<u>0.054 \pm 0.002</u>	0.36 \pm 0.02	0.041 \pm 0.002	<u>0.25 \pm 0.02</u>	0.19 \pm 0.03	1.3 \pm 0.3
	DTML	0.049 \pm 0.006	0.33 \pm 0.04	<u>0.052 \pm 0.005</u>	0.33 \pm 0.04	0.21 \pm 0.03	1.7 \pm 0.3
	MASTER	0.064* \pm 0.006	0.42 \pm 0.04	0.076* \pm 0.005	0.49 \pm 0.04	0.27 \pm 0.05	2.4 \pm 0.4
CSI800	XGBoost	0.040 \pm 0.000	0.37 \pm 0.01	0.047 \pm 0.000	0.42 \pm 0.01	0.08 \pm 0.02	0.6 \pm 0.2
	LSTM	0.028 \pm 0.002	0.32 \pm 0.02	0.039 \pm 0.002	0.41 \pm 0.03	0.09 \pm 0.02	0.9 \pm 0.2
	GRU	0.039 \pm 0.002	0.36 \pm 0.05	0.044 \pm 0.003	0.39 \pm 0.07	0.07 \pm 0.04	0.6 \pm 0.3
	TCN	0.038 \pm 0.002	0.33 \pm 0.04	0.045 \pm 0.002	0.38 \pm 0.05	0.05 \pm 0.04	0.4 \pm 0.3
	Transformer	0.040 \pm 0.003	0.43 \pm 0.03	0.048 \pm 0.003	0.51 \pm 0.05	0.13 \pm 0.04	1.1 \pm 0.3
	GAT	<u>0.043 \pm 0.002</u>	0.39 \pm 0.02	0.042 \pm 0.002	0.35 \pm 0.02	0.10 \pm 0.04	0.7 \pm 0.3
	DTML	0.039 \pm 0.004	0.29 \pm 0.03	<u>0.053 \pm 0.008</u>	0.37 \pm 0.06	<u>0.16 \pm 0.03</u>	<u>1.3 \pm 0.2</u>
	MASTER	0.052* \pm 0.006	<u>0.40 \pm 0.06</u>	0.066 \pm 0.007	<u>0.48 \pm 0.06</u>	0.28* \pm 0.02	2.3* \pm 0.3

Table 1: Overall performance comparison. The best results are in bold and the second-best results are underlined. And * denotes statistically significant improvement (measured by t-test with p-value < 0.01) over all baselines.

Model	IC	ICIR	RankIC	RankICIR	AR	IR
(MA)STER	0.064 \pm 0.003	0.43 \pm 0.02	0.074 \pm 0.004	0.48 \pm 0.04	0.25 \pm 0.03	2.1 \pm 0.3
(MA)STER-Bi	<u>0.058 \pm 0.005</u>	0.38 \pm 0.04	<u>0.066 \pm 0.008</u>	<u>0.41 \pm 0.05</u>	0.19 \pm 0.03	1.6 \pm 0.2
Naive	0.041 \pm 0.008	0.30 \pm 0.05	0.046 \pm 0.007	0.32 \pm 0.04	0.18 \pm 0.05	1.6 \pm 0.6
Clustering	0.044 \pm 0.003	0.36 \pm 0.02	0.049 \pm 0.005	0.39 \pm 0.04	0.18 \pm 0.04	<u>1.7 \pm 0.3</u>

Table 2: Experiments on CSI300 to validate the effectiveness of proposed stock transformer architecture. The best results are in bold and the second-best results are underlined.

mostly consider the 30 top-performed stocks. The achievements in both types of metrics imply that MASTER is of good predicting ability on the whole stock set without sacrificing the accuracy of the important stocks. The significant improvements cast light on the importance of stock correlation modeling, so each stock can also benefit from the historical signals of other momentarily correlated stocks. We also observe all methods gain better performance on CSI300 over CSI800. We believe it is because CSI300 consists of companies with larger capitalization whose stock prices are more predictable. When compared to the existing stock correlation method (i.e., DTML), MASTER outperforms in all 6 metrics, which tells our proposed Market-Guided Gating and aggregation techniques are more efficient in mining cross-stock information than existing literature.

Stock Transformer Architecture (RQ2)

We validate the effectiveness of our specialized stock transformer architecture by experiments on four settings. (1) (MA)STER, which is our stock transformer without the gating. (2) (MA)STER-Bi, in which we substitute the single-layer transformer encoder with a bi-directional LSTM to evince that the effectiveness of our proposed architecture is not coupled with strong sequential encoders. (3) Naive, which directly performs information aggregation among $\tau \times |S|$ tokens. (4) Clustering, in which we adapt the Local Sensitive Hashing (Kitaev, Kaiser, and Levskaya 2020) to allo-

cate all tokens into 10 buckets by similarity and perform aggregation within each group, which is a classic task-agnostic technique to reduce the scale of the attention field. For a fair comparison, in (3) and (4), we first use the same transformer encoder to extract token embedding and then use the same multi-head attention mechanism as in our stock transformer, so the only difference is the attention field. Due to resource limits, we only conduct experiment on CSI300 dataset. The results in Table 2 illustrate the efficacy of our tailored stock transformer architecture, which performs intra-stock aggregation and inter-stock aggregation alternatively.

Ablation Study (RQ3)

First, we conduct ablation study on (N_1, N_2) combination. The results of CSI300 are shown in Figure 3 and the results on CSI800 are similar. The difference among head combinations is not significant compared with the inherent variance under each setting. In the studied range, most settings consistently performed better than the baselines.

Second, we study the influence of temperature β in the gating mechanism. As explained before, a smaller β forces a stronger feature selection while a larger β turns off the gating effect. Figure 4 shows the performance with varying β . The CSI300 is a relatively easier dataset where most features are quite effective, so the temperature is expected to be larger to relax the feature selection, while more powerful feature selection intervention is needed for the sophisticated

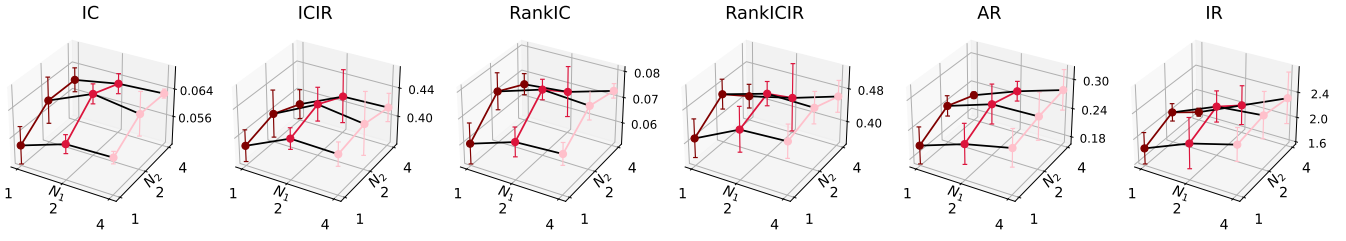


Figure 3: The average and standard deviation of metrics with different (N_1, N_2) combinations on CSI300.

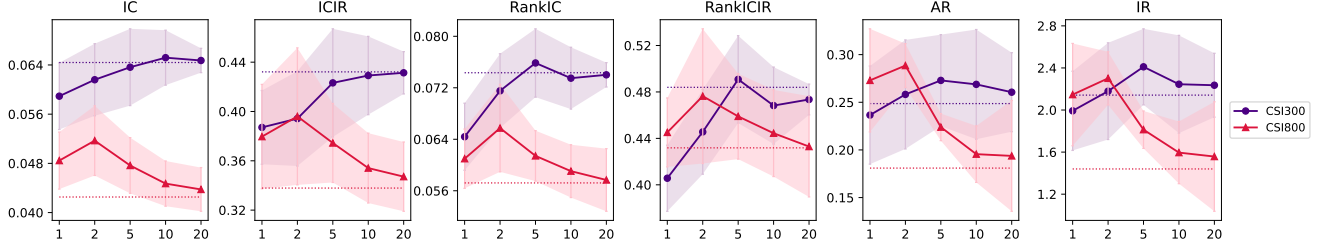


Figure 4: MASTER performance with varying β . The horizontal dash lines are performance without market-guided gating.

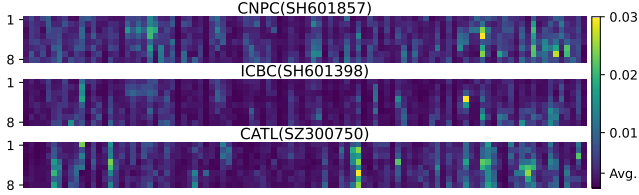


Figure 5: The correlation towards three target stocks on Aug 19th, 2022. The y-axis is time steps in the lookback window and the x-axis is source stocks. Avg. denotes the evenly distributed value.

CSI800 dataset whose β of the best performance is smaller.

Visualization of Attention Maps (RQ4)

We show how MASTER captures the momentary and cross-time stock correlation that previous methods are not expressive enough to model. Figure 5 shows the inter-stock attention map at different time steps in the lookback window. We choose three representative stocks as the target and sample 100 random stocks as sources for visualization. The highlighted part is scattered instead of exhibiting neat strips, implying that the correlation is momentary rather than long-standing. Also, the inter-stock correlation is sparse, with only a few stocks having strong correlations toward the target stocks. Figure 6 displays the correlation between stock pairs to show how the correlation resides in time. From source stock v to target stock u , we compute $I_{u \leftarrow v}[i, j] = S_v^1[i, j]S_u^2[u, v]$ as the $\tau \times \tau$ correlation map, while S_v^1 and S_u^2 are the intra-stock and inter-stock attention map. First, the highlighted blocks are not centered on the diagonal, because the stock correlation is usually cross-time rather than temporally aligned. Second, the left two figures are totally different, illustrating that correlation is

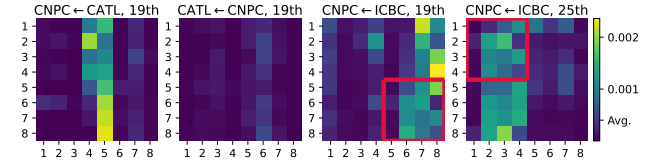


Figure 6: Cross-time correlation of stock pairs on Aug 19th and 25th, 2022. The x-axis is the source time steps and the y-axis is the target time steps.

highly asymmetric between $u \leftarrow v$ and $v \leftarrow u$. Third, the importance of mined correlation changes slowly when the lookback window slides to forecast on different dates. For example, blocked regions in the right two figures correspond to the same absolute time scope of different prediction dates, whose patterns are to a certain degree similar.

Conclusion

We introduce a novel method MASTER for stock price forecasting, which models the realistic stock correlation and guide feature selection with market information. MASTER consists of five steps, market-guided gating, intra-stock aggregation, inter-stock aggregation, temporal aggregation, and prediction. Experiments on the Chinese market with 2 stock universe shows that MASTER achieves averagely 13% improvements on ranking metrics and 47% on portfolio-based metrics compared with all baselines. Visualization of attention maps reveals the de-facto momentary and cross-time stock correlation. In conclusion, we provide a more granular perspective for studying stock correlation, while also indicating an effective application of market information. Future work can explore to mine stock correlations of higher quality and study other uses of market information.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful reviews. This work is supported by the National Key Research and Development Program of China (2022YFE0200500), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and SJTU Global Strategic Partnership Fund (2021SJTHKUST).

References

- Bai, S.; Kolter, J. Z.; and Koltun, V. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.
- Bennett, S.; Cucuringu, M.; and Reinert, G. 2022. Lead-lag detection and network clustering for multivariate time series with an application to the US equity market. *Machine Learning*, 111(12): 4497–4538.
- Bulat, A.; Perez Rua, J. M.; Sudhakaran, S.; Martinez, B.; and Tzimiropoulos, G. 2021. Space-time mixing attention for video transformer. *Advances in neural information processing systems*, 34: 19594–19607.
- Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.
- Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Cong, Y.; Liao, W.; Ackermann, H.; Rosenhahn, B.; and Yang, M. Y. 2021. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16372–16382.
- Ding, Q.; Wu, S.; Sun, H.; Guo, J.; and Guo, J. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction. In *IJCAI*, 4640–4646.
- Feng, F.; Chen, H.; He, X.; Ding, J.; Sun, M.; and Chua, T.-S. 2018. Enhancing stock movement prediction with adversarial training. *arXiv preprint arXiv:1810.09936*.
- 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.
- Graves, A.; and Graves, A. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.
- 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*, 850–858.
- Kamble, R. A. 2017. Short and long term stock trend prediction using decision tree. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, 1371–1375. IEEE.
- Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.
- Li, L.; Duan, L.; Wang, J.; He, C.; Chen, Z.; Xie, G.; Deng, S.; and Luo, Z. 2023. Memory-Enhanced Transformer for Representation Learning on Temporal Heterogeneous Graphs. *Data Science and Engineering*, 8(2): 98–111.
- Li, Q.; Jiang, L.; Li, P.; and Chen, H. 2015. Tensor-based learning for predicting stock movements. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.
- Liu, J.; Lin, H.; Liu, X.; Xu, B.; Ren, Y.; Diao, Y.; and Yang, L. 2019. Transformer-based capsule network for stock movement prediction. In *Proceedings of the first workshop on financial technology and natural language processing*, 66–73.
- Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2022. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.
- Nugroho, F. S. D.; Adji, T. B.; and Fauziati, S. 2014. Decision support system for stock trading using multiple indicators decision tree. In *2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering*, 291–296. IEEE.
- Piccolo, D. 1990. A distance measure for classifying ARIMA models. *Journal of time series analysis*, 11(2): 153–164.
- Sawhney, R.; Agarwal, S.; Wadhwa, A.; Derr, T.; and Shah, R. R. 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 497–504.
- Sawhney, R.; Agarwal, S.; Wadhwa, A.; and Shah, R. R. 2020. Spatiotemporal hypergraph convolution network for stock movement forecasting. In *2020 IEEE International Conference on Data Mining (ICDM)*, 482–491. IEEE.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Wang, H.; Li, S.; Wang, T.; and Zheng, J. 2021. Hierarchical Adaptive Temporal-Relational Modeling for Stock Trend Prediction. In *IJCAI*, 3691–3698.
- Wang, H.; Wang, T.; Li, S.; Zheng, J.; Guan, S.; and Chen, W. 2022. Adaptive long-short pattern transformer for stock investment selection. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 3970–3977.
- Wang, Y.; Qu, Y.; and Chen, Z. 2022. Review of graph construction and graph learning in stock price prediction. *Procedia Computer Science*, 214: 771–778.
- Xiang, S.; Cheng, D.; Shang, C.; Zhang, Y.; and Liang, Y. 2022. Temporal and Heterogeneous Graph Neural Network for Financial Time Series Prediction. In *Proceedings of*

the 31st ACM International Conference on Information & Knowledge Management, 3584–3593.

Xie, B.; Passonneau, R.; Wu, L.; and Creamer, G. G. 2013. Semantic frames to predict stock price movement. In *Proceedings of the 51st annual meeting of the association for computational linguistics*, 873–883.

Xu, M.; Dai, W.; Liu, C.; Gao, X.; Lin, W.; Qi, G.-J.; and Xiong, H. 2020. Spatial-temporal transformer networks for traffic flow forecasting. *arXiv preprint arXiv:2001.02908*.

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*.

Yang, X.; Liu, W.; Zhou, D.; Bian, J.; and Liu, T.-Y. 2020. Qlib: An ai-oriented quantitative investment platform. *arXiv preprint arXiv:2009.11189*.

Yoo, J.; Soun, Y.; Park, Y.-c.; and Kang, U. 2021. 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*, 2037–2045.

Zhang, Y.; and Yan, J. 2022. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*.