Deep Portfolio Optimization Modeling based on Conv-Transformers with Graph Attention Mechanism

1st Jifeng Sun
Tsinghua-Berkeley Shenzhen institute
Tsinghua University
Shenzhen, China
sjf19@mails.tsinghua.edu.cn

1* Wentao Fu
Tsinghua-Berkeley Shenzhen institute
Tsinghua University
Shenzhen, China
fuwentao@sz-sgd.com
* co-author

3rd Jianwu Lin
Tsinghua Shenzhen Graduate School
Tsinghua University
Shenzhen, China
lin.jianwu@sz.tsinghua.edu.cn

4th Yong Jiang
Tsinghua-Berkeley Shenzhen institute
Tsinghua University
Shenzhen, China
jiangy@sz.tsinghua.edu.cn

5* Shu-Tao Xia
Tsinghua Shenzhen Graduate School
Tsinghua University
Shenzhen, China
xiast@sz.tsinghua.edu.cn
* Corresponding author

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892317

***Abstract*—Optimizing portfolios is an important concern for all investors. Nowadays, deep learning has been applied to the study of portfolio investment. Still, the widely used deep learning methods based on asset return prediction do not guarantee to maximize the performance of a portfolio. In this paper, we design a neural network with the overall return-risk ratio of the portfolio as the optimization objective to determine the optimal allocation weights of the portfolio. We design the network architecture based on the Conv-Transformer with graph attention mechanisms(CTG) to better model the temporal dependence of assets and the correlation relationship between assets. The empirical results on the Chinese stock market show that our approach achieves the best results compared to the current SOTA model.**

***Index Terms*—Stock Feature Modeling, Graph Feature Modeling, Transformer, Chinese Market**

I. Introduction

Portfolio optimization is an important issue for all investors. With the advancement of science and technology and the participation of professional investors, investment theory has developed rapidly. Financial data is strongly time-series, and past historical data can be used to predict future trends, which means that modeling financial data is necessarily an NP-hard problem, and it is impossible to use violent traversal or algorithms to search for features automatically. Therefore, considering the similarity of scenarios, modeling historical financial data to build asset portfolios can be borrowed from the field of speech recognition or natural language processing [5], [18], [20]. Nowadays, deep learning methods have been applied to stock investment modeling. However, deep learning is not a panacea for all problems. The widely used deep learning methods based on asset return prediction are not guaranteed to maximize the performance of the portfolio. Several recent research advances [1] have shown that return forecast-based approaches are not guaranteed to maximize portfolio performance, and that minimizing forecast losses is not maximizing the overall portfolio return. According to Markowitz's work [3], widely known as Modern Portfolio Theory (MPT), constructing a multi-asset portfolio leads to a smoother overall return curve and thus to higher returns per unit of risk than individual asset trades. This idea has been confirmed (see [2]). In addition, some stocks in the portfolio show similar or highly correlated stock price trends because they are closely related to each other in the industry chain or have similar operating conditions that affect each other and are all response to systematic market risk. Paired trading or so-called statistical arbitrage strategies are widely used investment strategy solutions. The underlying assumption is that two or several stocks whose corresponding price series show similar behavior or are correlated with each other should

Fig. 1. Normalized Price series composition of some related stocks in the Chinese stock market.

persist in the future. This suggests that these securities are exposed to relevant risk factors and tend to react in the same way. Figure 1 demonstrates how these stocks are linked to each other. Thus, if there is a relative price deviation between stocks, a trading opportunity will arise from which investors can profit. Therefore, modeling the movements of these stocks with correlations is important for constructing this type of portfolio. Some recent research works [9] have mined information about the association between stocks around stock concepts or rather industries (e.g., technology, Internet retailing) to improve the prediction results of models. However, most of the existing work assumes that the linkage between stocks and equities is fixed, but the relationship between stocks is dynamic and changing in real financial markets. The existing models ignore the dynamics of the linkages and limit the prediction results. To address these limitations, we proposed a novel graph-based framework with the overall return-risk ratio of the portfolio as the optimization objective to determine the optimal allocation weights of the portfolio. The model can use historical data to optimize the next position weights for each stock in the portfolio. In addition, to better model the time dependence of assets and the correlation relationship between assets, this paper designs a network structure based on the Transformer and graph attention mechanisms. The empirical results on the Chinese stock market show that our approach achieves the best results relative to the current SOTA model.

The main contributions of this paper include:

- We proposed a novel framework with the overall return-risk ratio of the portfolio as the optimization objective.
- Our proposed framework can mine the stocks' hidden shared information. The shared information we mined can reflect the valuable indication of the stock's future trends with shared commonness.
- We conducted the experimental evaluation and investment simulation on the real-world data, and the results verified our framework's validity.

II. Related Work

This section reviews popular portfolio optimization methods and discusses how deep learning models have been applied to this field, especially for the deep learning methods for the portfolio with a strong relationship.

A. Stock temporal models

The size of stock market data has increased rapidly in recent years. Due to their excellent large-scale data processing capabilities, deep learning models have been introduced into many research scenarios in stock market forecasting. Chen et al. [11] used the Long Short Term Memory (LSTM) model to classify the rise and fall of the Chinese stock market. Applying the deep learning model to stock market prediction reflects the current research trend. In addition, deep learning models can learn effective feature representations directly from the raw financial data without many complex artificial features. For example, Ribeiro et al. [12] net used Deep Brief Network (DBN) to learn unsupervised representations of financial data for predicting future price movements of stocks. Bao et al. [41] used deep learning methods like Auto Encoder (AE) to stock price data for unsupervised representation learning. They then used the learned representation for predicting the trends of stock prices. The paper published in WWW 2019 [36] proposes the introduction of TCN, a novel time-series network model. The nodal relationship graph is constructed by the entity relationships of companies, countries, and industries, and the graph node representation is updated based on the daily news. The node Embedding and price data are obtained using type tools such as TransE for index prediction. The model can beat some classical and conventional deep learning time series models under general market conditions and is more effective in markets with severe volatility than the conventional time series models. The model beats the classical deep time series models such as LSTM, GRU, etc., on many time series tasks.

The studies mentioned above all point out that deep feature representations yield better predictions than conventional feature representations (e.g., technical indicators.) in their studies. Deep models can also capture potential connections in stock numerical data better than conventional models.

B. Stock graph embedding models

Graph neural networks build inter-sample relationships and aggregate features from neighboring nodes to the central node. Graph Neural Network (GNN) extends the usage scenario of deep learning techniques from conventional image and speech to graph-structured data. It has wide applications in fraud detection, shopping recommendation, etc. GNN is developed from graph signal theory and spectral-domain graph convolution, and the idea is to build a graph of inter-sample relationships explicitly or implicitly. Each node corresponds to one sample, and then the features of neighboring nodes are aggregated to the central node to update the node features. The quantitative investment approach based on graph neural networks was first published in [41]. It is the first work applying inter-listed company relationship graphs as input and graph convolutional neural networks for stock price prediction. The contribution of this paper is to propose a construction method for constructing inter-listed company relationship graphs: using data (obtained via Wind) from the ten major shareholders of a listed company to represent the connectivity between two companies through a common shareholding ratio. The first paper presents the boosting effect of prior knowledge of graph structure for stock prediction based on the introduction of embedding. This paper inspired subsequent work on graph neural networks for stock price prediction, such as extracting graphs of relationships between listed companies from knowledge graphs such as wiki data, Nikkei Value Search data, and brokerage research reports as inputs to the model. The work of Feng, F.et al. [29] proposes an approach to update Spatio-temporal graph convolution based on prices dynamically. The specific strategy is to design graph nodes with dynamically updated relationships over time. This implied relationship can be learned from historical data by assuming that the price influence relationship between stocks is variable. HATS [38] uses historical prices of individual 431 companies in the SP500 and stock relationships in Wikidata. This paper uses an attention mechanism to selectively choose information about different relationship types and finds that if inappropriate relationship graphs are used, they can instead hurt stock movement prediction performance. Also, this paper predicts index movements based on individual stocks at the same time. Compared to the benchmark MLP without a relational graph, CNN, LSTM, and Basic Graph Convolutional Neural network with relational graph, Temporal Graph Convolution Network, the authors' proposed methods have better prediction performance. This paper inspired subsequent research, which specified that inter-firm relationships do not necessarily enhance predictions, that useless graph information may reduce the effect. Therefore, the selection of valid firm associations deserves further research. Matsunaga D et al.'s work [44] is the latest paper to use relationship graphs between stocks to improve stock movement forecasts. It builds graph neural networks on 20-year-historical data on the Nikkei 225 for backtesting, constructs company relationships from Nikkei Value Search data to enhance the model, and compares different dependencies' gaining effect have on the return performance. The paper uses sliding window analysis better to evaluate the model's performance over time.

III. Methodology

This section presents our approach and discusses how to optimize the model's performance by optimizing the return-risk ratio and using a neural network based on the graph attention mechanism and Transformer. We describe the neural networks and the role of each module in our approach in detail. We depict our overall neural network architecture in Figure 2, and the dimensional transformation of the features during processing is shown in detail in Figure 4.

A. Overview of Workflow

Our model CTG consists of several building blocks: stock temporal feature modeling, stock graph embedding modeling and object function. The idea of the workflow is to use neural networks to extract temporal and graph features from input stock features and then predict the optimal holding weight of stocks. The overall architecture figure can be seen in Figure 2.

Features extracted from deep learning models have been suggested to perform better than conventional hand-crafted features [1]. The convolutional block has been widely used [11] for the first layer to extract features. Considering the fairness of comparison with other methods, we do not adopt the modeling of correlation features by adding industry or concept information but directly adopt the graph attention mechanism to learn the correlation features from historical stock price data.

B. Objective Function

The objective function is used to optimize the return per risk of a portfolio. According to Markowitz's investment theory [3], we can define it as:

$$L_T = \frac{\frac{1}{T}\sum_{t=1}^{T} R_{p,t}}{\sqrt{\frac{1}{T}\sum_{t=1}^{T} R_{p,t}^2 - \left(\frac{1}{T}\sum_{t=1}^{T} R_{p,t}\right)^2}} \tag{1}$$

T is a trading period, represents the length of time for each sample. The $R_{p,t}$ is realized portfolio return over n assets at time t and is denoted as:

$$R_{p,t} = \sum_{i=1}^{n} w_{i,t-1} \cdot r_{i,t} \tag{2}$$

$r_{i,t}$ is the return of asset i. According to the previous section, our target output is a long-short strategy, so we go long on some stocks and short on some stocks. The range of weights for each stock $w_{i,t}$ is between 1 and -1.

$$\begin{gathered} \sum_{i}^{n} w_{i,t} = 0 \\ \sum_{i}^{n} |w_{i,t}| = 1 \\ w_{i,t} \in [-1, 1] \end{gathered} \tag{3}$$

The neural network f with parameters θ is adopted to model $w_{i,t}$ for a long-short portfolio. We use gradient ascent to maximize the objective target L_T. The gradient of L_T with respect to parameters θ is readily calculable, with an excellent derivation presented in [43]. By using gradient ascent,we obtain $\partial L_T/\partial\theta$. We update the parameters θ every epoch.

$$\theta^{T+1} = \theta^T + \alpha\frac{\partial L_T}{\partial\theta} \tag{4}$$

where α is the learning rate.

C. Stock temporal feature modeling

Considering the powerful representation capability of cnn for local modeling and the outstanding performance of Transformer for temporal modeling, our module consists of a one-layer convolutional block structure stacked with two layers of Transformer. The output of the convolutional block is used as input to the Transformer.

The benefits of CNN over RNN are that the kernel is parallelizable and the CNN projection provides a more informative dynamic representation compared to the original time series because it captures the relative local dependencies between data points. Therefore, in financial time series modeling, CNN is generally used as the first layer of neural networks to directly process the raw financial feature data like [11], [13].

The weights are normalized using a weight normalization layer (Weight Norm) for each convolutional block, then activated by an activation function layer (Relu). The parameters are randomly lost using a Dropout layer. The significance of adding a Dropout layer is that it can effectively mitigate the

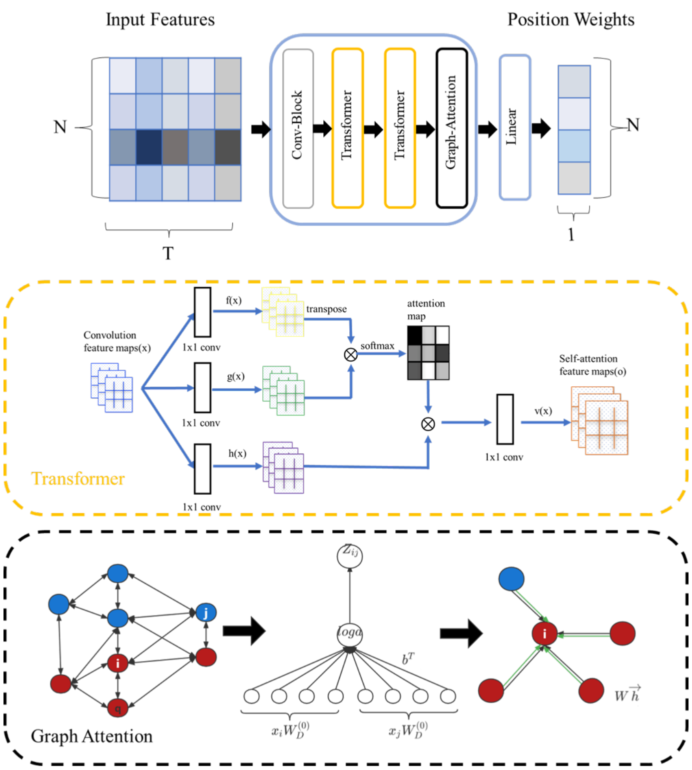

Fig. 2. Model architecture schematic. Overall, our model contains several main building blocks.GAT is used to aggregate the stock association feature. Then the spatially fused features are fed to the linear layer module to output a position of the portfolio

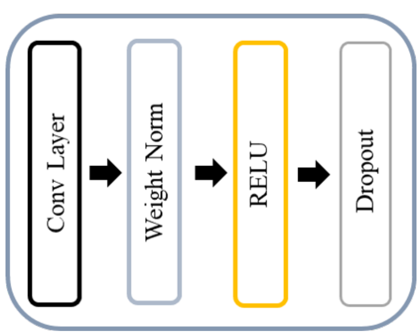

Fig. 3. The component structure of Convolutional block

occurrence of overfitting and, to some extent, achieve the effect of regularization to improve the model performance.

However, the convolutional block is only a local feature encoder for features structurally, and we need the Transformer network to detect temporal dependence patterns globally. The temporal module in our workflow is a one-layer Transformer, following the implementation of [18]. The Transformer has rapidly become the choice model for sequence modeling such as Natural Language Processing (NLP) problems. It replaces older recurrent neural network models such as the Long Short-Term Memory (LSTM) network [5]. Transformer abandons the conventional CNN and RNN design architecture, and the entire network structure is composed entirely of Attention mechanisms. More precisely, Transformer consists of and only consists of Self-Attention and Feed Forward Neural Network. The core formula of the self-attention mechanism is shown below, where $\tilde{X}$ is the outputs of the previous convolution block layer. Transformer adopts attention mechanism with Query-Key-Value (QKV) model. Given the matrix representations of queries $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$, attention used by Transformer is defined as:

$$\operatorname{Attention}(Q, K, V)=\operatorname{softmax}\left(\frac{Q K^{\top}}{\sqrt{D_k}}\right) V \tag{5}$$

$$\mathbf{Q}=\mathbf{K}=\mathbf{V}=\tilde{X} \in \mathbb{R}^{N \times D_k} \tag{6}$$

$$\begin{aligned} \operatorname{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) &= \text{ Concat } \left(\text{ head }_1, \cdots, \text{ head }_H\right) \mathbf{W}^O \\ \text{where head }_i &= \operatorname{Attention}\left(\mathbf{Q} \mathbf{W}_i^Q, \mathbf{K} \mathbf{W}_i^K, \mathbf{V} \mathbf{W}_i^V\right) \end{aligned} \tag{7}$$

$$\mathrm{z}=\operatorname{ReLU}(\operatorname{MultiHeadAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})) \tag{8}$$

D. Stock graph embedding modeling

To model the correlation among stocks, we use GAT to model the Graph embedding. GAT uses the global attention mechanism, which does not require explicit graph building for the stock market, but implicitly learns the impact of all nodes on the central node and then aggregates this information to the central node. z_i denotes the transformed time series data of the target stock, and z_j denotes the transformed time series data of the associated stock, and a^T denotes the vector of pre-defined weights, and LeckyReLU is the activation function, and e_{ij} denotes the graph attention score.

$$e_{ij}=\operatorname{LeckyReLU}\left(\vec{a}^T\left(z_i \| z_j\right)\right) \tag{9}$$

After obtaining the graph attention scores of each associated stock corresponding to the target stock, the SoftMax function is applied to each graph attention score to get the respective attention weights corresponding to each associated stock, wherein the formula may calculate attention weights.

$$\boldsymbol{a}_{ij}=\frac{\exp \left(e_{ij}\right)}{\sum_{k \in N_i} \exp \left(e_{ik}\right)} \tag{10}$$

The a_{ij} is the attention weights. At last, we feed the attention weight a into a linear layer with LeakyReLU activation function to generate the forecast position weights $\hat{y}$.

$$\hat{y}=\operatorname{LeakyReLU}(a W+b) \tag{11}$$

IV. Empirical Analysis

A. Data

Experimental data: We obtain Chinese stock data from Wind, a financial data provider, and select CSI 300 constituent stocks as the pool of stocks for strategy trading. CSI 300 is a free-float weighted index consisting of 300 Chinese stocks listed on the Shanghai or Shenzhen Stock Exchanges. Our analysis should use the most liquid stocks in order to avoid trading and market friction issues. CSI 300 is the top 300 stocks with the largest market capitalization and

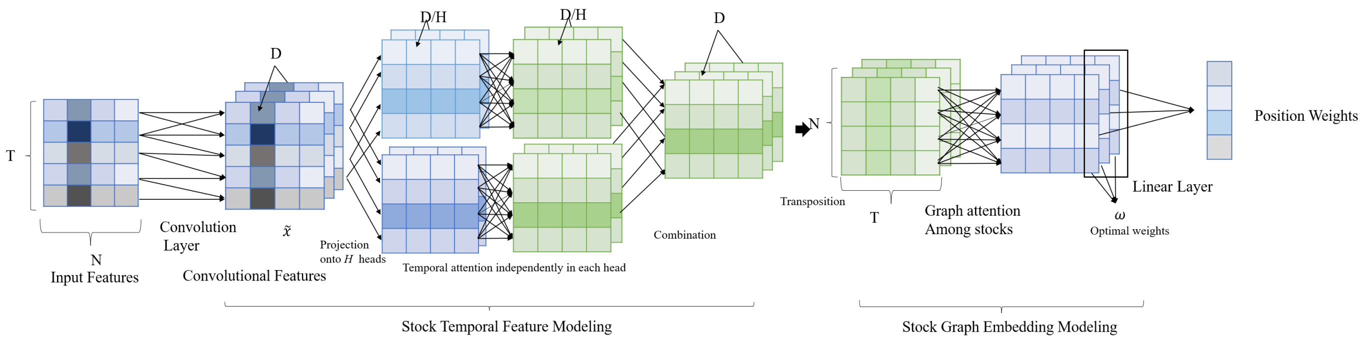

Fig. 4. This figure shows the structure of our neural network. The convolutional block takes a window of T consecutive daily features of N stocks as input, and outputs D features maps. Each of the features is a nonlinear function of the observations in the block, and captures a common pattern. The Transformer takes the matrix $\tilde{x}$ that we obtain from the convolutional block as input, which contains D features for each block. These features are projected onto H attention heads, which independently quantify the temporal relation between the blocks and aggregate them into hidden states. These hidden states are finally combined by the graph attention layer and a linear layer to predict the optimize weight for the portfolio in the next day.

the best liquidity, meeting the requirements. we select the daily frequency stock features of Alpha360 in the quantitative investment platform Qlib [14]. The Alpha360 dataset contains six stock data on each day, which are opening price, closing price, highest price, lowest price, the volume-weighted average price (VWAP), and trading volume. For each stock on date t, Alpha360 looks back 60 days to construct a 360-dimensional historical stock data as a stock feature of this stock at date t. The training set interval is from 2015-01-04 to 2017-01-01. The test set interval is from 2017-01-01 to 2022-01-01.

B. Experimental Setting

Baselines. We compare our framework with the following competitive baselines, which can be categorized into several groups. The first group consists of time series forecasting models, including MLP, Long Short-Term Memory(LSTM) [5], and Transformer [18]. The second group consists of classical graph neural network models, including GCN and GAT. The third group contains both graph and temporal information of the stock, which is ALSTM [22]. The above model is what we used to set up for the ablation experiment for exploring the strength of the model's good performance. The detailed introduction of these baselines are in Appendix A.1.

Evaluation Metrics. The metrics can be categorized into two:

- Four widely-used evaluation metrics: the Information Coefficient (IC) [45] ,Rank IC [46],Information Coefficient Information Ratio(ICIR) [45] and Rank ICIR [45].
- Portfolio-based metrics: Annualized Return, Information Ratio, Maximum Drawdown

IC is the Information Coefficient, which indicates the cross-sectional correlation coefficient between the selected stock's factor value and the stock's next return, and the IC value can determine the predictive power of the factor value on the next return, which can be calculated as

$$\mathrm{IC} = \frac{1}{N}\frac{(\hat{\mathbf{y}} - \mathrm{mean}(\hat{\mathbf{y}}))^{\mathrm{T}}(\mathbf{y} - \mathrm{mean}(\mathbf{y}))}{\mathrm{std}(\hat{\mathbf{y}}) * \mathrm{std}(\mathbf{y})} \tag{12}$$

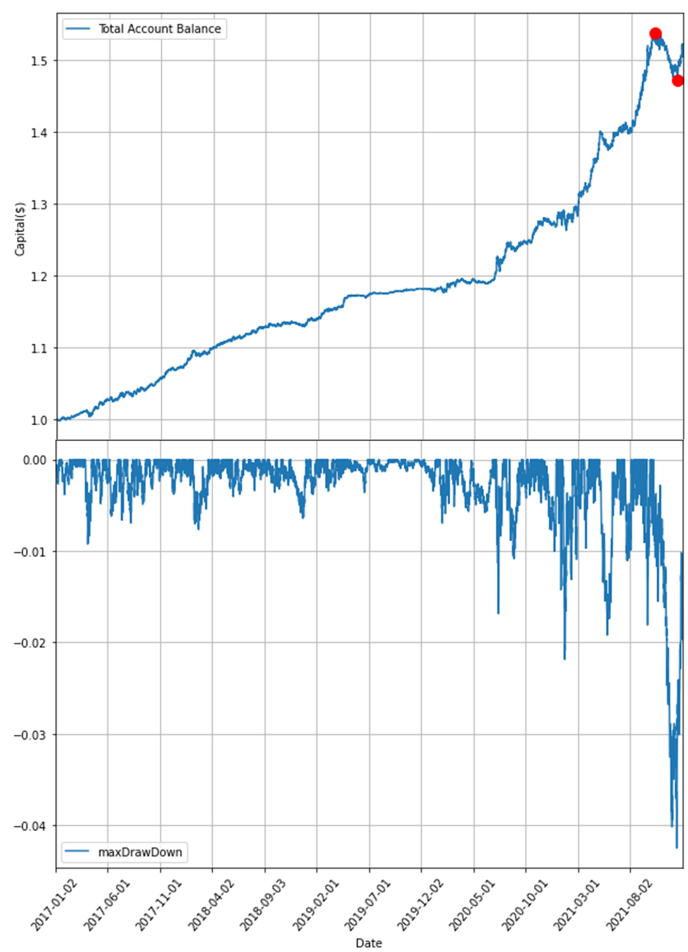

Fig. 5. Cumulative Return and Maximum Drawdown figure of our CTG method on CSI 300 from 2017 to 2021. The cumulative return curve reflects the strategy returns over time. The Maximum Drawdown (MDD) is an indicator used to assess the relative riskiness of a trading strategy because it focuses on capital preservation, which is the primary concern of most investors.

TABLE I
THE MAIN RESULTS OF METHODS ON CSI 300.

Methods	CSI 300						
	IC	Rank IC	ICIR	Rank ICIR	Annualized Return	Information Ratio	Max Drawdown
MLP	0.0273 (0.00)	0.0396 (0.00)	0.1870 (0.02)	0.2910 (0.02)	0.0029 (0.02)	0.274 (0.23)	-0.1385 (0.03)
XGBoost	0.0394 (0.00)	0.0448 (0.00)	0.2909 (0.00)	0.3679 (0.00)	0.0344 (0.00)	0.4527 (0.02)	-0.1004 (0.00)
TCN	0.0441 (0.00)	0.0519 (0.00)	0.3301 (0.02)	0.4130 (0.01)	0.0604 (0.02)	0.8295 (0.34)	-0.1018 (0.03)
LSTM	0.0448 (0.00)	0.0549 (0.00)	0.3474 (0.04)	0.4366 (0.03)	0.0647 (0.03)	0.8963 (0.39)	-0.0875 (0.02)
GRU	0.0493 (0.00)	0.0584 (0.00)	0.3772 (0.04)	0.4638 (0.03)	0.0720 (0.02)	0.9730 (0.33)	-0.0821 (0.02)
GATs	0.0476 (0.00)	0.0598 (0.02)	0.3508 (0.02)	0.4604 (0.01)	0.0824 (0.02)	1.1079 (0.26)	-0.0894 (0.03)
ALSTM	0.0497 (0.00)	0.0599 (0.00)	0.3829 (0.04)	0.4736 (0.03)	0.0626 (0.02)	0.8651 (0.31)	-0.0994 (0.03)
Transformer	0.0114 (0.00)	0.0327 (0.00)	0.0716 (0.03)	0.2248 (0.02)	0.0270 (0.03)	0.3378 (0.37)	0.1653 (0.05)
CTG	**0.0625** **(0.00)**	**0.0637** **(0.00)**	**0.4387** **(0.00)**	**0.4756** **(0.03)**	**0.0856** **(0.01)**	**1.2558** **(0.42)**	**-0.0425** **(0.00)**

where $\hat{\mathbf{y}}$ and $\mathbf{y}$ are the predicted position weight rankings and actual stock return rankings, respectively.

IR is the Information Ratio, which is the ratio of the mean to the standard deviation of excess returns and can be approximated by the IC with the following formula.

$$IR \approx \frac{\overline{IC_t}}{std\left(\overline{IC_t}\right)} \tag{13}$$

To show the stability of IC, we report the information ratio of IC, i.e., ICIR, which is calculated by dividing the average by the standard deviation of IC.

Since the values of IC are of continuous type, in order to prevent the calculation process due to the excessive disparity of factor values, the calculation of Rank IC is proposed, which differs from IC in that the specific values of factors, as well as the specific values of returns, are converted to the ranking order of the corresponding values in their cross-sections. Rank ICIR is calculated in a similar way to Rank IC.

Annualized Return(AR) measures the total profits generated by following the model predictions for investment yearly. It's calculated by scaling the daily average portfolio return to the whole calendar days in one year. We use the Annual Return to evaluate the investment simulation result.

$$\mathrm{AR} = \mathrm{mean(R)} \times 252 \times 100\% \tag{14}$$

The Maximum Drawdown(MDD) is the maximum value of the return retracement when the net value of the portfolio goes to its lowest point at any historical point backward in the selected period. Maximum Drawdown is used to describe the worst-case scenario after building a portfolio. Maximum Drawdown is a vital risk indicator. We also use the MDD to analyze the risk of investment simulation.

$$\mathrm{MDD(t)} = \frac{\text{Trough Value }_t - \text{ Peak Value }_{0-\mathrm{t}}}{\text{Peak Value }_{0-\mathrm{t}}} \tag{15}$$

Experiment Setups. We set the dropout rate to 0.2. We set the learning rate to 1e-4. The number of units on the convolutional layer is 64. The attention head number of the Transformer is set to 4. We implement our framework with the PyTorch library [42] and run all experiments on a single NVIDIA GeForce RTX™ 3080 Ti GPU. For the simulate real-world trading, we consider a transaction cost of 0.05% for buying stocks and 0.15% for selling stocks.

EXPERIMENT RESULTS

Table 1 shows the experimental results of CTG and other baselines on the stocks of CSI 300. From table 1, we can see that our proposed model, CTG, achieves the best performance on all the metrics. It indicates that our model performs well in fitting financial signals with fewer errors than some latest stock trend modeling methods like TCN, LSTM, ALSTM, Transformer. Besides, the CTG exceeds GATs, indicating the power of relationship modeling of our methods. While the dataset is updated to January 2022, CTG has the most significant benefit, meaning that our model is practicable and effective until the latest time. To further evaluate our CTG framework's effectiveness, we simulate the investment in the CSI 300's test set (from 01/01/2017 to 01/01/2022). The cumulative return curve reflects the strategy returns over time. We can see that our strategy obtained good performance results throughout the test period, with only a period of Maximum Drawdown in the third quarter of 2021. The results may be related to the market conditions at that time. This situation is also reflected in the Maximum Drawdown curve. Therefore, in real trading, we need to promptly train and update the model parameters.

CONCLUSION AND FUTURE WORK

In this paper, we introduce a novel framework to compare different deep learning approaches based on return-risk ratio of the portfolio as the optimization objective to determine the optimize allocation weights of the portfolio and the network architecture based on the Conv-Transformer and graph attention mechanisms. We conduct a comprehensive empirical out-of-sample study on Chinese stock market and demonstrate the potential of deep learning methods in stock trading. Our CTG substantially outperforms all benchmark approaches. The core idea of the temporal graph neural network is to combine stock temporal modeling with graph attention neural network to learn richer information on the temporal and relationship domains in financial data, which applies to the field of quantitative stock selection. The CTG outperforms the compared models in all evaluation metrics in the empirical analysis.

In the future, we plan to explore the self-supervised training framework for stocks based on our current study. Self-supervised learning has gradually become a hot research topic in recent several years. In financial scenarios, the effect of artificial labels is not ideal to describe the market. The labeling methods are not perfect, and the labels are often unbalanced. Self-supervised training can generate valuable labels and correct the "conceited" manual labels. We will label the stocks based on the self-supervised learning method and lay the foundation for the subsequent modeling.

References

[1] Zihao Zhang, Stefan Zohren, and Roberts Stephen. Deep reinforcement learning for trading. The Journal of Financial Data Science, 2020.

[2] Eric Zivot. Introduction to computational finance and financial econometrics. Chapman Hall Crc, 2017.

[3] Harry Markowitz. Portfolio selection. The journal of finance, 7(1):77–91, 1952.

[4] Yang G , Wang D , Wang J . Correlation analysis of price fluctuation of stock market by stochastic interacting system[J]. Beijing Jiaotong Daxue Xuebao/Journal of Beijing Jiaotong University, 2017, 41(3):120-126.

[5] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8),1735-1780.

[6] Xu Nianxing, Hong Tao, Wu Shinong, et al. Information transmission patterns, investor psychological biases, and the "same up, same down" phenomenon in stock prices[J]. Economic Research, 2011(4):135-146.

[7] Podobnik B , Wang D , Horvatic D , et al. Time-lag cross-correlations in collective phenomena[J]. Europhysics Letters, 2010, 90(6):68001 (6 pp.) — 68001 (6 pp.) 68001 (6 pp.).

[8] Liu Yichen. Analysis of the price linkage effect of each stock within the industry in China's stock market [D]. North University of Technology,2017.

[9] BEST M J,GRAUERRR.The Analytics of Sensitivity Analysis for Mean-Variance Portfolio Problems[J]. International Review of Financial Analysis,1992,1(1):17-37.

[10] Hamilton, W., Ying, Z., Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in neural information processing systems, 30.

[11] Chen, J. F., Chen, W. L., Huang, C. P., Huang, S. H., Chen, A. P. (2016, November). Financial time-series data analysis using deep convolutional neural networks. In 2016 7th International conference on cloud computing and big data (CCBD) (pp. 87-92). IEEE.

[12] Black F,Jensen M C,Scholes M.The capital asset pricing model: some empirical tests, Studies in Theory of Capital Markets.New York: Praeger ,1972:20-74.

[13] Zhang Z, Zohren S, Roberts S. Deeplob: Deep convolutional neural networks for limit order books[J]. IEEE Transactions on Signal Processing, 2019, 67(11): 3001-3012.

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

[15] Fama E. F.,French K. R.Common Risk Factors in the Returns on Stocks and Bonds[J]. Journal of Financial Economics,1993,33(1):3-56.

[16] Fama E. F., French K. R. Financing Decisions: Who Issues Stock? [J]. Journal of Financial Economics, 2005, 76(3): 549-582.

[17] Fama E. F.,French K. R. A Five-Factor Asset Pricing Model[J]. Journal of Financial Economics,2015,116(1): 1-22.

[18] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]// Advances in neural information processing systems. 2017: 5998-6008.

[19] Zhang, S., Tong, H., Xu, J., Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. Computational Social Networks, 6(1), 1-23.

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

[21] Xu Y, Cohen S B. Stock movement prediction from tweets and historical prices[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018: 1970-1979.

[22] Feng F, Chen H, He X, et al. Enhancing stock movement prediction with adversarial training [J]. arXiv preprint arXiv:1810.09936, 2018.

[23] Berthelot D, Carlini N, Cubuk E D, et al. Remixmatch: semi-supervised learning with distribution matching and augmentation anchoring [C]// International Conference on Learning Representations. 2019.

[24] Xie Q, Dai Z, Hovy E, et al. Unsupervised data augmentation for consistency training[J]. arXiv preprint arXiv:1904.12848, 2019.

[25] Deng S, Zhang N, Zhang W, et al. Knowledge-driven stock trend prediction and explanation via temporal convolutional network[C]//Companion Proceedings of The 2019 World Wide Web Conference. 2019: 678- 685.

[26] Rozantsev A, Salzmann M, Fua P. Beyond sharing weights for deep domain adaptation[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(4): 801-814.

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

[28] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: data-driven traffic forecasting[J]. arXiv preprint arXiv:1707.01926, 2017.

[29] Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.S., 2019. temporal relational ranking for stock prediction. acm transactions on information systems (TOIS) 37, 27.

[30] Deng S, Zhang N, Zhang W, Chen J, Pan JZ, Chen H. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. InCompanion Proceedings of The 2019 World Wide Web Conference 2019 May 13 (pp. 678-685). ACM.

[31] Li Q, Han Z, Wu XM. Deeper insights into graph convolutional networks for semi-supervised learning. inThirty-Second AAAI Conference on Artificial Intelligence 2018 Apr 29.

[32] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. arXiv:1609.02907, 2016.

[33] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

[34] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: data-driven traffic forecasting[J]. arXiv preprint arXiv:1707.01926, 2017. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems. 2019 Aug 22.

[35] Cho, Kyunghyun; van Merrienboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holger; Bengio, Yoshua (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". arXiv:1406.1078.

[36] Bandinelli N, Bianchini M, Scarselli F. Learning long-term dependencies using layered graph neural networks. in: The 2010 International Conference on Neural Networks (IJCNN). Barcelona: IEEE, 2010, 1-8

[37] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks. arXiv:1511.05493, 2015 Liao R, Brockschmidt M, Tarlow D, et al. Graph partition neural networks for semi-supervised classification. arXiv: 1803.06272, 2018.

[38] Chung J, Gulcehre C, Cho K, Bengio Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555. 2014 Dec 11.

[39] Jing L, Tian Y. Self-supervised visual feature learning with deep neural networks: a survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[40] Yves Grandvalet and Yoshua Bengio. semi-supervised learning by entropy minimization. in Advances in Neural Information Processing Systems, 2005.

[41] Chen Y, Wei Z, Huang X. Incorporating corporation relationship via graph convolutional neural networks for stock price predic-

tion[C]//Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 1655-1658.

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 (2019).

[43] Gabriel Molina. Stock trading with recurrent reinforcement learning (RRL). CS229, nd Web, 15, 2016.

[44] Matsunaga D, Suzumura T, Takahashi T. Exploring graph neural networks for stock market predictions with rolling window analysis[J]. arXiv preprint arXiv:1909.10660, 2019.

[45] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. 2021. Learning Multiple Stock Trading Patterns with Temporal Routing Adaptor and Optimal Transport. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining. 1017–1026.

[46] Zhige Li, Derek Yang, Li Zhao, Jiang Bian, Tao Qin, and Tie-Yan Liu. 2019. Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining. 894–902.

APPENDIX

A.1 Baselines

We compare our proposed framework with the following stock trend forecasting methods:

MLP: a 3-layers multi-layer perceptron (MLP) with the number of units on each layer is 512.

TCN:a Temporal Convolutional Network(TCN) network based stock trend forecasting method.

LSTM: a Long Short-Term Memory (LSTM) network based stock trend forecasting method.

GRU: a Gated Recurrent Unit (GRU) network based stock trend forecasting method.

XGBoost: XGBoost stands for eXtreme Gradient Boosting, is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library.

GATs: a forecasting model that utilizes graph attention networks (GATs) to aggregate stock embeddings encoded by GRU on the stock graph. We use the stocks as nodes to construct a stock graph, and two stocks have a relation when they share the same predefined concept.

ALSTM: a variant of LSTM with a temporal attentive aggregation layer to aggregate information from all hidden states in previous timestamps.

Transformer: A Transformer based stock trend forecasting model.