Non-local Graph Aggregation for Diversified Stock Recommendation

Zhihan Yue[1] and Ying Tan[1,2](✉)

[1] Key Laboratory of Machine Perception (MOE), School of Artificial Intelligence, Institute for Artificial Intelligence, Peking University, Beijing 100871, China
{zhihan.yue,ytan}@pku.edu.cn

[2] Nanjing Kangbo Intelligent Health Academy, Nanjing 211100, China

Abstract. Stock prediction plays a key role in stock investments. Despite the promising achievements of existing solutions, there are still limitations. First, most methods focus on mining the local features from node neighbors, while ignoring non-local features in the stock market. Second, most existing works form the portfolio with the stocks with the highest predicted return, exposed to some risk factors that cause common price movements. To reduce the risk exposure, it is crucial to learn a diversified portfolio. To address the shortage of existing methods, this paper proposes a novel stock recommendation framework that enables both local and non-local feature learning for stock data. Different from the existing methods, the stocks are selected locally according to the ranks within each independent group. This strategy diversifies the recommended stocks effectively. Experimental results on multiple datasets from the U.S. and Chinese stock markets demonstrate the superiority of the proposed method over existing state-of-the-art methods.

Keywords: Stock prediction · Non-local aggregation · Graph neural networks

1 Introduction

Stock prediction, aiming to predict the future movements of stock prices, plays a key role in active stock investments. It helps investors to select the stocks with the best profitability. Sequential models based on recurrent neural networks (RNNs) [8,14,23] and convolutional neural networks (CNNs) [1,3,4] have been widely applied to stock prediction tasks. These methods forecast each stock time series independently, without incorporating the correlations between stocks. Recent studies have focused on modeling the stock relationships with graph neural networks (GNNs) [9,18]. GNN-based methods represent the stock relationships as a graph and enable features learning from relevant stocks with local aggregation. However, there are still notable limitations in these methods.

First, few of the existing methods mine the non-local representations of the market to enhance the prediction of individual stocks. Most existing works leverage local operators to aggregate features from node neighbors. However, the price

Y. Tan and Y. Shi (Eds.): DMBD 2022, CCIS 1745, pp. 147–159, 2022.
https://doi.org/10.1007/978-981-19-8991-9_12

movement correlations over different stocks are both local and non-local. Empirical studies have proven that capitalization, industry, liquidity, and many other non-local factors have a significant impact on the price movements of individual stocks. [7] Therefore, learning non-local features is crucial for making accurate stock predictions.

Second, most existing works form the portfolio with the stocks with the highest predicted returns. The resulting portfolio may be exposed to some risk factors that cause common price movements on different stocks, which contributes to the overall risk of the portfolio. To reduce the risk exposures, it is crucial to learn a diversified portfolio.

To address these issues, this paper proposes a novel stock recommendation framework, which enables both local and non-local feature learning for stock data. It assigns the stocks into diversified portfolios and learns non-local states for each portfolio. These non-local states are attended by the stock embeddings with the attention mechanism, which injects global features into node-level representations. Different from the existing methods, the stocks are selected locally according to the ranks within each independent group. This strategy diversifies the recommended stocks effectively. Extensive experiments on multiple datasets demonstrate the superiority of our method.

The major contributions of this paper are summarized as follows:

- This paper proposes a novel framework that utilizes both local and non-local features for diversified stock recommendation. To the best of our knowledge, it is one of the first few studies exploring the role of non-local features in stock recommendation.
- To achieve the above goal, two novel designs are leveraged in the framework. First, the non-local aggregation module is proposed to capture the non-local market states and inject the non-local states into the stock embeddings. Second, a diversity loss is introduced to learn independent groups for diversified stock recommendation.
- The proposed method outperforms existing state-of-the-art baselines on three real-world datasets. For example, our method improves the Sharpe Ratio by 61.1% on ACL18, 9.7% on KDD17, and 2.6% on CH compared to the best baseline methods.

2 Related Work

2.1 Stock Prediction

There is a rich history of studies about stock prediction. Most classical methods are based on time series analysis models, such as Auto-Regressive Moving Average (ARMA) [2], Vector Auto-Regression (VAR) [16], and Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) [6]. However, these methods are based on specific linear assumptions about the stochastic processes, facing difficulties when dealing with complex time series tasks such as stock prediction.

To address the limitations of classical methods, there have been many efforts bringing in deep learning to predict the stock trends, which outperform classical models in precision. For some concrete examples, [23] decomposes the hidden states of memory cells with discrete Fourier transform and captures multi-frequency trading patterns from market data to predict stock prices. [8] leverages adversarial training to improve the generalization of neural networks for stock prediction. [5] proposes multi-scale Gaussian prior to enhance the locality of vanilla transformer and applies fixed temporal windows to learn hierarchical features of market data. In addition, there have been a few attempts utilizing graph neural networks to model the cross-sectional relationship. [9] extracts cross-sectional features by graph convolution on the predefined industry and company graphs. [18] leverages multi-graph interaction to learn stock correlations dynamically, showing competitive performance on various datasets. Despite they have achieved promising results, few of them explored non-local feature learning for the stock graph, which is crucial to stock prediction tasks.

2.2 Graph Neural Networks

In recent years, a wide variety of graph neural networks has been proposed. Most of these models adopt the framework of "message passing" [11], in which the GNN aggregates features from neighbors and updates the node representation. For example, GCN [19] aggregates the linearly transformed features from each node's neighbors to update its representation; GAT [17] performs masked attention to adaptively aggregate features from neighbors; GraphSAGE [12] introduces a sampling strategy for local aggregation to scale to large graphs. Pooling methods have also been proposed for graphs to coarsen features of a group of nodes. [21] proposes a differentiable graph pooling module that enables hierarchical representation learning of graphs in an end-to-end fashion. [15] proposes a sparse pooling method that captures coarse information hierarchically with better edge connectivity. [10] introduces not only a pooling operator (gPool) but also its inverse operation (gUnpool) to inject coarse features into the original graph. Unlike these previous approaches, this paper focuses on risk-aware coarsening of the stock relation graph, which is underexplored.

3 Method

3.1 Problem Statement

Given a set of historical stock time series $\mathcal{X} = \{X_1, X_2, \cdots, X_N\}$ from N correlated stocks, the target is to learn a mapping function $f_\theta(\mathcal{X})$ that predicts the future price movements of these stocks based on observed historical features. Each input time series X_i is a multivariate time series that has dimension $T \times F$, where T is the time series length and F is the feature dimension. To enable the cross-sectional feature interaction between latent stock embeddings, this paper further introduces a stock graph $\mathcal{G}$. The graph $\mathcal{G}$ can be represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ is the set of stock nodes, $\mathcal{E}$ is the set of edges, and $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the adjacent matrix representing the stock correlations.

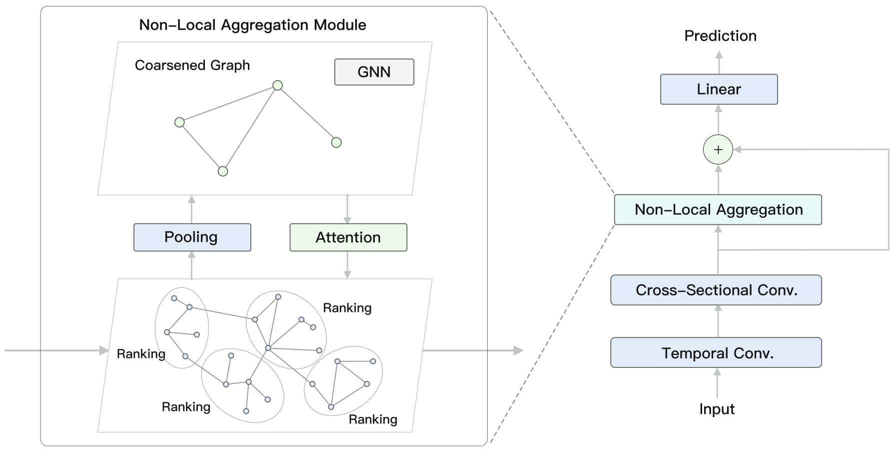

Fig. 1. The overall framework of the proposed method.

3.2 Architecture

Figure 1 shows the overall framework of the proposed method. The input time series $\mathcal{X}$ are firstly fed into a temporal convolution module to encode temporal dynamics for individual stocks. Then, a cross-sectional convolution module is applied to enable cross-sectional feature learning that aggregates features from local neighbors. Finally, a non-local aggregation module is developed, which has the capability of capturing the global correlations between stocks. Therefore, both local and non-local dependency learning are incorporated in the framework. The final predictions are produced with a linear mapping layer. The details of the framework are described in the following subsections.

3.3 Spatial-Temporal Embedding

To capture both non-linear temporal dependencies and cross-sectional correlations in stock time series, this paper introduces the temporal and cross-sectional convolution modules, respectively.

Temporal Convolution Module. The temporal convolution module consists of several dilated CNN blocks to extract high-level temporal features for individual stocks. The l-th block contains two 1-D convolution layers with a dilation parameter of 2^l. The dilated convolutions enable a large receptive field for temporal encoding [1]. This module contains 3 hidden building blocks. Each block has the structure of "GatedGELU $\rightarrow$ DilatedConv $\rightarrow$ GELU $\rightarrow$ DilatedConv" with skip connections between adjacent blocks. The kernel size is set to 5 in order to encode the original daily observations into weekly features. The input gate (GatedGELU) is designed to filter the noise of the financial data, which can be formulated as:

$$Z' = \mathrm{SIGM}(w \cdot Z + b) \odot \mathrm{GELU}(Z), \tag{1}$$

where $Z, Z' \in \mathbb{R}^{N \times T \times H}$ is the input and output respectively, $w, b \in \mathbb{R}^H$ is the learnable weights, SIGM is the sigmoid activation function, and GELU is the Gaussian Error Linear Unit activation function.

Cross-Sectional Convolution Module. The cross-sectional convolution module is applied after the temporal convolution module to aggregate time series embeddings from local neighbors. The SAGE convolution [12] is selected as the building block in the module, for its capability of encoding large graphs. This module is comprised of three SAGE convolutional layers. To exploit contextual information from multiple hops, the concatenation of these three SAGE layers is passed to a fully connected layer to produce spatial embedding. The encoding process of the cross-sectional convolution module is as follows:

$$E_i = \mathrm{ReLU}(\mathrm{SAGEConv}_i(E_{i-1}, \mathcal{A})), \tag{2}$$
$$E = \tanh(\mathrm{FC}([E_1, E_2, E_3])), \tag{3}$$

where $E_i \in \mathbb{R}^{N \times H}$ represents the embedding from i-th SAGE convolutional layer, H is the hidden dimension, E_0 is the output of temporal convolution module at the last time point, and E is the final spatial embedding.

3.4 Non-local Graph Aggregation

This subsection proposes the non-local aggregation module that injects global market states into stock nodes. The assignment matrix is required to coarsen the local embeddings to global market states. This paper adopts an adaptive way for learning assignment matrix due to the dynamic nature of the market states. For example, although the industry of a stock is relatively fixed, varied themes still emerge over time. Inspired by this, the assignment matrix is defined as the sum of a dynamic assignment matrix, that changes over time, and a learnable static assignment matrix:

$$S_{\mathcal{G}} = \mathrm{softmax}(\tilde{S} + f_{assign}(E, \mathcal{A})), \tag{4}$$

where $\tilde{S} \in \mathbb{R}^{N \times G}$ is the learnable static assignment matrix that mines intrinsic properties of the stocks, f_{assign} is a cross-sectional convolution module with an output dimension of G, G is the number of the pooled nodes, and $S_{\mathcal{G}} \in \mathbb{R}^{N \times G}$ is the final assignment matrix for graph $\mathcal{G}$.

Following [21], the hidden features of the stock nodes are mapped into the coarse high-level graph with:

$$E_c = S_{\mathcal{G}}^T E, \tag{5}$$
$$\mathcal{A}_c = S_{\mathcal{G}}^T A S_{\mathcal{G}}, \tag{6}$$

where $E_c \in \mathbb{R}^{G \times H}, \mathcal{A}_c \in \mathbb{R}^{G \times G}$ are the node features and the adjacent matrix of the coarse graph respectively.

Then, a GNN module is applied to aggregate global features in the coarse graph:

$$E'_c = f_c(E_c, \mathcal{A}_c), \tag{7}$$

where f_c is a cross-sectional convolution module applied in the coarse graph with an output dimension of H.

To unpool the coarse features into the original graph, a scaled dot-product attention layer is introduced, which queries the aggregated node features in the coarse graph E'_c by the original graph embeddings E:

$$Q = EW_q, \quad K = E'_c W_k, \quad V = E'_c W_v, \tag{8}$$

$$E' = \text{softmax}\left(\frac{QK^T}{\sqrt{H}}\right) V, \tag{9}$$

where $W_q, W_k, W_v \in \mathbb{R}^{H \times H}$ are learnable weights for linear transformation.

3.5 Diversified Stock Recommendation

Stock ranking [24] serves as an effective way for stock recommendation. It aims to predict the relative rank of investment revenues in a cross-section. It is required to learn a function f_θ that gives better rank predictions. To achieve this, the Information Coefficient metric is used as a training objective. The corresponding loss function can be formulated as:

$$L_{IC}^{(t)} = -\frac{\sum_i (\hat{r}_{i,t} - \mu_{\hat{r}_{:,t}})(r_{i,t} - \mu_{r_{:,t}})}{(N-1)\sigma_{\hat{r}_{:,t}}\sigma_{r_{:,t}}}, \tag{10}$$

where $r_{i,t}$ denotes the i-th stock return at the time point t, $\hat{r}_{i,t}$ denotes the predicted ranking score of i-th stock at the time point t, and μ, σ represents the sample mean and standard deviation respectively.

After getting the rank predictions, one may select the stocks with the highest ranking scores as the portfolio. However, the portfolio produced by this rule may be risky due to the common price movements of different stocks. Hence this paper proposes a risk parity strategy aiming to select stocks from several independent groups. This strategy is based on a loss function that enforces the non-local aggregation module to learn a diversified assignment matrix. Note that the nodes in the coarse graph can be seen as portfolios weighted by the assignment matrix, and their returns can be calculated by:

$$R = S_{\mathcal{G}}^T r, \tag{11}$$

where $r \in \mathbb{R}^{N \times T_f}$ represents the return matrix of the future T_f time points for all stocks.

The goal is to minimize the covariance between these portfolios. Because of $\text{Var}\left(\frac{1}{G}\sum_{i=1}^{G} R_i\right) = \frac{1}{G^2}\sum_i \sum_j \text{Cov}(R_i, R_j)$, it is feasible to optimize the variance of the mean of these portfolio returns. Therefore, we propose the diversity loss as follows:

$$\mathcal{L}_D = \text{Std}\left(\frac{1}{G}\sum_{i=1}^{G} R_i\right), \tag{12}$$

where Std represents the sample standard deviation function.

The final loss is the weighted sum of these two losses balanced by a factor α:

$$\mathcal{L} = \alpha\mathcal{L}_D + \frac{1}{T_{train}}\sum_t \mathcal{L}_{IC}^{(t)}. \tag{13}$$

To select diversified stocks, the stock nodes are divided into groups using the assignment matrix. For each stock, the group it belongs to is defined as the group with the largest assignment weight for this stock. The stock that ranks in the top k_1 in its group will be selected as a candidate. The stocks with top k_2 ranking scores in all candidate stocks form the final portfolio.

4 Experiments

4.1 Settings

Datasets. The proposed method is evaluated on three datasets: ACL18 [20], KDD17 [23], and CH. KDD17 and ACL18 are widely used public datasets for stock prediction, and CH is a real-world dataset from Chinese markets.

- ACL18 collects the historical time series of 88 stocks in NASDAQ and NYSE markets, which ranges from 2012-09 to 2017-09. The data from 2012-09 to 2016-02 are used for training, 2016-03 to 2016-08 for validation, and 2016-09 to 2017-09 for testing.
- KDD17 contains 50 stocks in U.S. markets ranging from 2007-01 to 2016-12. The data from 2007-01 to 2014-12 are used for training, 2015-01 to 2015-12 for validation, and 2016-01 to 2016-12 for testing.
- CH includes the constituent stocks of the SH50 index of the Chinese market. It contains 49 stocks ranging from 2013-01 to 2020-12. The data from 2013-01 to 2018-12 are used for training, 2019-01 to 2019-12 for validation, and 2020-01 to 2020-12 for testing.

Features. To evaluate the end-to-end performance without feature engineering, only the price and volume features are used for the prediction model. To avoid drifting, the unit roots are removed from the original k-line data by:

$$x_t^{open} = \text{open}_t/\text{close}_t - 1, \tag{14}$$

$$x_t^{high} = \text{high}_t/\text{close}_t - 1, \tag{15}$$

$$x_t^{low} = \text{low}_t/\text{close}_t - 1, \tag{16}$$

$$x_t^{close} = \text{close}_t/\text{close}_{t-1} - 1, \tag{17}$$

$$x_t^{vol} = \text{volume}_t/\text{Mean}(\text{volume}_{t-41:t}) - 1, \tag{18}$$

where $\text{open}_t, \text{high}_t, \text{low}_t, \text{close}_t$ and volume_t denote the open price, the highest price, the lowest price, the close price and the trading volume at day t respectively, and $\text{Mean}(\text{volume}_{t-41:t})$ is the average volume of the data from last 42 trading days (about two months).

Graph Construction. The common movements of the stock prices often reflect the inherent characteristics of stocks. For example, the prices of stocks belonging to the same industry often rise and fall at the same time. Hence this paper adopts a predefined graph in which the edge weights are defined as the Pearson correlation coefficients between stocks:

$$\mathcal{A}_{i,j} = \frac{\sum_t (r_{i,t} - \mu_{r_i})(r_{j,t} - \mu_{r_j})}{(T_{train} - 1)\sigma_{r_i}\sigma_{r_j}}, \tag{19}$$

where $r_{i,t}$ denotes the i-th stock return at the time point t, and μ, σ represents the sample mean and standard deviation respectively. The adjacent matrix $\mathcal{A}$ is calculated using the training set with T_{train} trading days.

Evaluation Metrics. Multiple metrics are used to evaluate the performance of all methods, including Information Coefficient (IC), Annual Return, Sharpe Ratio, and Calmar Ratio [13]. IC is the Pearson correlation between the predicted signals and the ground-truth returns. The Annual Return is the annualized return of the backtesting result for trading strategies. The Sharpe Ratio and Calmar Ratio are the expected return per unit of risk measured by the standard deviation and the maximum drawdown respectively.

Hyperparameters. The number of training epochs is set to 5 empirically. α is set to 1.0. The model inputs time series data of the previous 21 trading days (about one month). T_f is set to 63. H is set to 64. G is set to 30. k_1 is set to 1 for ACL18 and CH, and 2 for KDD17. k_2 is set to 10 for ACL18 and KDD17, and 5 for CH. All experiments are conducted on a NVIDIA GeForce RTX 3090.

4.2 Performance Comparison

Extensive experiments on stock recommendation are conducted to evaluate our method, compared with other state-of-the-art methods, including SFM [23], ALSTM [14], Adv-ALSTM [8], GCN [19], TGC [9], G-Transformer [5], and DTML [22]. The evaluation results are presented in Table 1, 2 and 3. Overall, the proposed method achieves substantial improvement over existing baselines on ACL18, KDD17, and CH datasets. Specifically, our method improves the Sharpe Ratio by 61.1% on ACL18, 9.7% on KDD17, and 2.6% on CH compared to the best result of baseline methods. We also note that our method achieves a similar annual return as GCN on the KDD17 dataset. However, our method substantially improves the Sharpe and Calmar ratios over GCN. This implies the proposed method effectively reduces the risk while ensuring the return.

Table 1. Performance comparison results on ACL18 dataset.

Method	IC	Annual return	Sharpe	Calmar
SFM	−0.22%	−8.29%	−0.734	−0.473
ALSTM	−0.31%	0.24%	0.019	0.012
Adv-ALSTM	0.00%	−1.21%	−0.111	−0.156
GCN	1.61%	21.87%	1.730	2.143
TGC	1.69%	28.51%	1.734	2.880
G-Transformer	0.26%	5.82%	0.463	0.396
DTML	−0.44%	−8.49%	−0.723	−0.400
Ours	**2.49%**	**37.6%**	**2.795**	**5.142**

Table 2. Performance comparison results on KDD17 dataset.

Method	IC	Annual return	Sharpe	Calmar
SFM	−0.44%	12.56%	1.231	1.358
ALSTM	0.47%	17.96%	1.582	2.722
Adv-ALSTM	1.58%	24.21%	1.858	2.226
GCN	1.35%	25.81%	1.552	2.964
TGC	−0.01%	16.16%	1.366	2.477
G-Transformer	0.95%	21.35%	1.699	2.708
DTML	2.14%	12.72%	0.868	1.609
Ours	**2.63%**	**26.24%**	**2.039**	**5.199**

Table 3. Performance comparison results on CH dataset.

Method	IC	Annual return	Sharpe	Calmar
SFM	1.48%	10.73%	0.423	0.367
ALSTM	1.63%	−2.60%	−0.095	−0.063
Adv-ALSTM	0.77%	10.93%	0.404	0.295
GCN	**4.32%**	35.71%	1.334	2.663
TGC	3.81%	34.28%	1.261	2.501
G-Transformer	4.18%	46.01%	1.880	2.770
DTML	2.81%	37.91%	1.524	1.313
Ours	**4.32%**	**70.84%**	**1.930**	**5.841**

4.3 Ablation Study

Table 4. Ablation results on ACL18 dataset.

	IC	Annual return	Sharpe	Calmar
Ours	**2.49%**	**37.6%**	**2.795**	**5.142**
w/o ST Embedding	1.73%	7.99%	0.598	0.607
w/o Non-Local Aggregation	1.50%	24.80%	1.993	4.550
w/o Diversity Loss	1.67%	27.90%	1.896	4.839

To verify the effectiveness of the proposed components in our method, the ablation study is conducted. The proposed method is compared with its three variants on ACL18 dataset, where (1) **w/o ST Embedding** removes the temporal and cross-sectional convolution modules, (2) **w/o Non-local Aggregation** removes the non-local aggregation module, and (3) **w/o Diversity Loss** sets α to 0. As shown in Table 4, all the above components are indispensable.

4.4 Signal Analysis

The objectives of diversified stock recommendation are twofold. First, the predicted signals of the model should be capable of distinguishing the levels of stock returns. This provides a basis for recommending the top-ranked stocks. Second, the recommended stocks should be as diversified as possible, which reduces the portfolio risk and improves investment performance. These two aspects are analyzed respectively below.

Discrimination. For each trading day, the stocks are divided into 5 levels according to the quantile of their predicted scores. For example, top 0%–20% means always recommending the top-20% ranked stocks for each trading day. The backtesting results over different signal quantiles on the ACL18 dataset for our method are shown in Fig. 2. Among these quantiles, the top quantile (0%–20%) achieves the best cumulative return, and the bottom quantile (80%–100%) achieves the worst cumulative return. This shows that the proposed model is able to distinguish the stocks with different returns clearly.

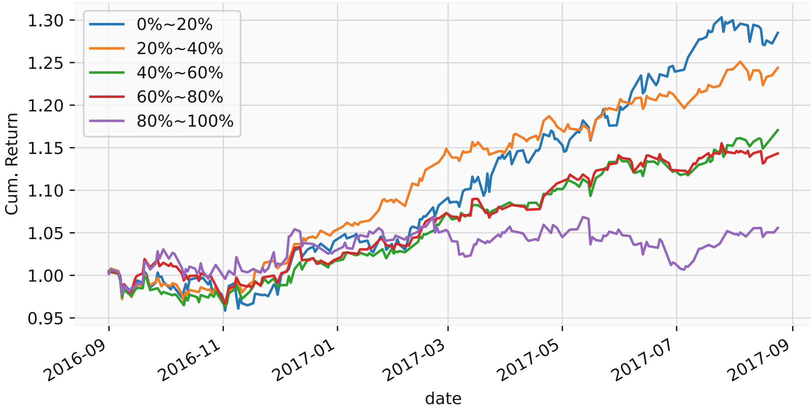

Fig. 2. Backtesting results over different signal quantiles on ACL18 dataset.

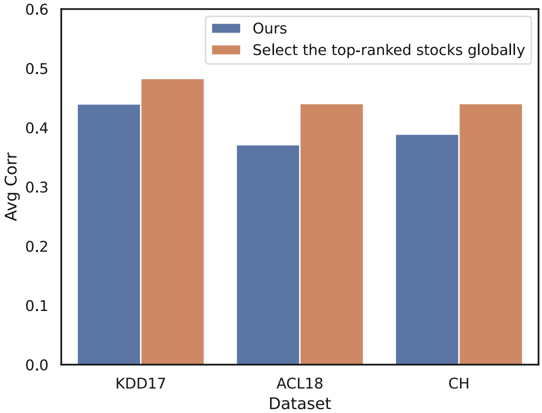

Fig. 3. The average correlation of the daily selected stocks with different strategies.

Diversification. To assess the portfolio diversity, we calculate the average correlation coefficient between the return series of daily recommended stocks. The proposed strategy in Sect. 3.5 is compared to the vanilla strategy that selects the top-ranked stocks globally. Figure 3 shows that the proposed model is able to improve the diversity of recommended stocks on all the datasets, thus achieving better performance.

5 Conclusion

This paper proposes a novel framework for diversified stock recommendation. Inside the framework, the spatial-temporal embedding is leveraged for capturing both temporal and cross-sectional dependencies, the non-local graph aggregation module is proposed for learning global market states, and the IC and diversity losses are introduced for stock ranking and diversification respectively. Extensive experiments on ACL18, KDD17, and CH datasets demonstrate the superiority of the proposed method over the existing state-of-the-art methods. Furthermore, the ablation study proves the effectiveness of the proposed components. Signal analysis validates the capability of the proposed method to distinguish the profitability of stocks and recommend a diversified portfolio. The proposed framework is generic and has the potential to be applied for other spatial-temporal prediction tasks.

Acknowledgements. This work is supported by Science and Technology Innovation 2030 - 'New Generation Artificial Intelligence' Major Project (Grant Nos.: 2018AAA0100302) and partially supported by the National Natural Science Foundation of China (Grant No. 62076010 and 62276008).

References

1. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271 (2018)
2. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. John Wiley & Sons, Hoboken (2015)
3. Chen, W., Jiang, M., Zhang, W.G., Chen, Z.: A novel graph convolutional feature based convolutional neural network for stock trend prediction. Inf. Sci. **556**, 67–94 (2021)
4. Deng, S., Zhang, N., Zhang, W., Chen, J., Pan, J.Z., Chen, H.: Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In: Companion Proceedings of The 2019 World Wide Web Conference, pp. 678–685 (2019)
5. Ding, Q., Wu, S., Sun, H., Guo, J., Guo, J.: Hierarchical multi-scale gaussian transformer for stock movement prediction. In: IJCAI, pp. 4640–4646 (2020)
6. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica: J. Econ. Soc., 987–1007 (1982)
7. Fama, E.F., French, K.R.: A five-factor asset pricing model. J. Finan. Econ. **116**(1), 1–22 (2015)
8. Feng, F., Chen, H., He, X., Ding, J., Sun, M., Chua, T.S.: Enhancing stock movement prediction with adversarial training. IJCAI (2019)
9. Feng, F., He, X., Wang, X., Luo, C., Liu, Y., Chua, T.S.: Temporal relational ranking for stock prediction. ACM Trans. Inf. Syst. (TOIS) **37**(2), 1–30 (2019)
10. Gao, H., Ji, S.: Graph u-nets. In: International Conference on Machine Learning, pp. 2083–2092. PMLR (2019)
11. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International Conference on Machine Learning, pp. 1263–1272. PMLR (2017)
12. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Adv. Neural Inf. Process. Syst. **30** (2017)
13. Ma, T., Tan, Y.: Stock ranking with multi-task learning. Expert Syst. Appl. **199**, 116886 (2022)

14. Qin, Y., Song, D., Cheng, H., Cheng, W., Jiang, G., Cottrell, G.W.: A dual-stage attention-based recurrent neural network for time series prediction. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp. 2627–2633 (2017)
15. Ranjan, E., Sanyal, S., Talukdar, P.: Asap: Adaptive structure aware pooling for learning hierarchical graph representations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 5470–5477 (2020)
16. Sims, C.A.: Macroeconomics and reality. Econometrica: J. Econ. Soc., 1–48 (1980)
17. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
18. Wang, H., Li, S., Wang, T., Zheng, J.: Hierarchical adaptive temporal-relational modeling for stock trend prediction. In: IJCAI, pp. 3691–3698 (2021)
19. Welling, M., Kipf, T.N.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR 2017) (2016)
20. Xu, Y., Cohen, S.B.: Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 1970–1979 (2018)
21. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. Adv. Neural Inf. Process. Syst. **31** (2018)
22. Yoo, J., Soun, Y., Park, Y.c., Kang, U.: 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, pp. 2037–2045 (2021)
23. Zhang, L., Aggarwal, C., Qi, G.J.: Stock price prediction via discovering multi-frequency trading patterns. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2141–2149 (2017)
24. Zhang, X., Tan, Y.: Deep stock ranker: a lstm neural network model for stock selection. In: International Conference on Data Mining and Big Data, pp. 614–623. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-319-93803-5_58