Stock Recommendations for Individual Investors: A Temporal Graph Network Approach with Mean-Variance Efficient Sampling

Youngbin Lee*
young@unist.ac.kr
Ulsan National Institute of Science and Technology (UNIST)
Republic of Korea

Yejin Kim*
kimyejin99@unist.ac.kr
Ulsan National Institute of Science and Technology (UNIST)
Republic of Korea

Javier Sanz-Cruzado
javier.sanz-cruzadopuig@glasgow.ac.uk
University of Glasgow
United Kingdom

Richard McCreadie†
richard.mccreadie@glasgow.ac.uk
University of Glasgow
United Kingdom

Yongjae Lee†
yongjaelee@unist.ac.kr
Ulsan National Institute of Science and Technology (UNIST)
Republic of Korea

ABSTRACT

Recommender systems can be helpful for individuals to make well-informed decisions in complex financial markets. While many studies have focused on predicting stock prices, even advanced models fall short of accurately forecasting them. Additionally, previous studies indicate that individual investors often disregard established investment theories, favoring their personal preferences instead. This presents a challenge for stock recommendation systems, which must not only provide strong investment performance but also respect these individual preferences. To create effective stock recommender systems, three critical elements must be incorporated: 1) individual preferences, 2) portfolio diversification, and 3) the temporal dynamics of the first two. In response, we propose a new model, Portfolio Temporal Graph Network Recommender `PfoTGNRec`, which can handle time-varying collaborative signals and incorporates diversification-enhancing sampling. On real-world individual trading data, our approach demonstrates superior performance compared to state-of-the-art baselines, including cutting-edge dynamic embedding models and existing stock recommendation models. Indeed, we show that `PfoTGNRec` is an effective solution that can balance customer preferences with the need to suggest portfolios with high Return-on-Investment. The source code and data are available at https://github.com/youngandbin/PfoTGNRec.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

recommender systems, stock recommendation, temporal graph network, modern portfolio theory

ACM Reference Format:
Youngbin Lee, Yejin Kim, Javier Sanz-Cruzado, Richard McCreadie, and Yongjae Lee. 2024. Stock Recommendations for Individual Investors: A Temporal Graph Network Approach with Mean-Variance Efficient Sampling. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698662

1 INTRODUCTION

In recent years, there has been a significant increase in the number of individual investors participating in the stock market. According to [2], about 58% of U.S. households owned stocks in 2022, up from 53% in 2019, marking the highest growth trend in recent history. This surge in participation highlights the growing interest in stock market investment among individual investors.

Despite this increasing engagement, individual investors often exhibit irrational investment behaviors that negatively impact their returns. Common behaviors include overconfidence, the disposition effect, lottery preference, and herding [25]. These tendencies result in investment returns that are generally lower than the market average, with the average investor significantly underperforming the S&P 500 over time [5].

There are many established methods for enhancing portfolio performance, one of the most notable being Modern Portfolio Theory (MPT) [23]. MPT posits that an investor can achieve higher returns for a given level of risk, or reduce risk for a given level of expected return, by selecting a mix of assets. This is accomplished via the diversification effect, which combines assets with low or negative correlations. Such diversification is effective in reducing overall portfolio risk, and it can be further enhanced by using machine learning techniques [22]. While MPT has been the foundation of investment management of most institutions [16], individual investors typically do not follow these sophisticated methods [17, 18]. Instead, their investment decisions are often driven by personal preferences, which are influenced by various factors such as psychological biases, news, and peers.

[*] These authors contributed equally.
[†] Co-corresponding authors.

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11.
https://doi.org/10.1145/3677052.3698662

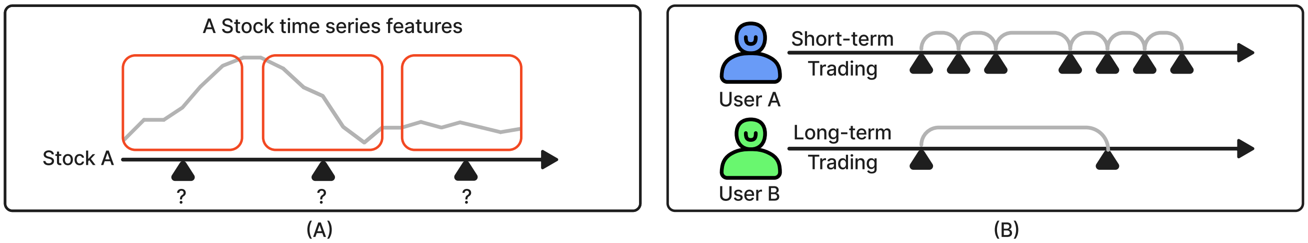

Figure 1: The importance of temporal aspects in stock recommender systems. (A) Various features of stocks would be different based on the timing of recommendations. (B) Contrasting behaviors between user A, who engages in short-term trading, and user B, who holds stocks for a long period.

Given these varied influences and the tendency for irrational investment behaviors, there is a clear need for a stock recommendation system. Such a system can guide individual investors, helping them make more disciplined and informed investment decisions. Indeed, in theory, by leveraging advanced recommendation models, it should be possible to capture both user preferences and market dynamics more effectively and concurrently, ultimately improving the investment outcomes for individual investors.

Creating an effective stock recommendation system involves several key considerations.

The first is **individual preference**. In essence, individual investment behaviors are highly heterogeneous [12–14]. Individual investors often navigate their unique paths, like interpreting and assessing information obtained from media and peers, choosing a few stocks instead of adhering to well-diversified portfolios [15]. For instance, a study analyzing Robinhood investors [39] revealed the phenomenon of "experience holding," where investors find pleasure in simply holding certain stocks that are chosen based on not purely cash-flow-based perspectives. Bhattacharya et al. [1] found that most retail investors do not follow unbiased financial advice from experts. They quoted the famous saying, *"You can lead a horse to water, but you can't make it drink."* That is, even though we can build a model that exhibits better investment performance, most retail investors would not take it if they do not like it.

However, most existing studies on investment recommendation only consider aspects related to the prices of financial assets [24, 26, 35]. There are two problems with price-based recommendations. First, it is almost impossible to provide accurate predictions of financial asset prices. Even the most sophisticated models exhibit accuracy around 52 to 57%, which is not enough to generate positive returns after transaction fees [42]. Second, they do not consider individual preferences. As noted before, many individuals are unlikely to follow recommendations that do not align with their tastes.

The second is **investment performances**, specifically, the diversification effect. No matter how well a model aligns with individual preferences, it is of no use if investment performance is poor. According to the modern portfolio theory originated from [23], diversification involves including stocks with low correlations in a portfolio to reduce risk and achieve stable returns. The diversification is crucial in investment management because the price prediction of financial assets would naturally include substantial error, and it has been the key success factor of most institutional investors [16].

However, the tricky point in stock recommendation is that the first two key aspects, individual preference, and investment performance, have a trade-off relationship. In experiments on 12 financial asset recommender (FAR) systems [30], it was concluded that transaction-based and profitability-based metrics are not interchangeable. FAR systems that learned from past pricing history showed high performance in return but performed poorly in individual preference, i.e. near zero. Conversely, FAR systems that learned from past transactions demonstrated good performance in individual preference but showed a downward trend in return. This shows that 'customers are not always right' in stock recommendations. Therefore, *it is inevitable that a trade-off between preference and profitability will need to be made if we are to achieve better investment performance for stock recommendation.*

Lastly, the **temporal nature** of stock features and user preferences is important. Figure 1 illustrates why the temporal aspect should be considered in stock recommendation. Figure 1 (A) shows that even the same stock can have very different characteristics depending on the timing of recommendations. If the recommendation is happening at a time point around the first red box, Stock A would seem like a good option. However, it would be better not to recommend Stock A during the second and third red boxes. In Figure 1 (B), there are two contrasting investment behaviors: user A engages in short-term trading, while user B holds stocks for an extended period. Thus, it is essential to consider the temporal dynamics of user behaviors.

In this paper, we propose a stock recommender system called **P**ort**fo**lio **T**emporal **G**raph **N**etwork **Rec**ommender (`PfoTGNRec`). The proposed model is based on a temporal graph network, developed by [28], to extract time-varying collaborative signals (key aspects 1 and 3: individual preference and temporal nature). Further, we incorporate MVECF [4] method in sampling contrastive pairs to enhance the diversification effect (key aspect 2: investment performance). Through experiments, we demonstrate that our model is the most effective in improving investment performance while capturing user preferences, achieving a 3.45% improvement in a comprehensive combined metric compared to the best model among various baselines, including recently developed dynamic graph embedding models and existing stock recommendation models.

2 RELATED WORKS

2.1 Stock Recommendations

Collaborative filtering (CF), which leverages historical user-item interactions, is one of the fundamental and most successful techniques in recommender systems. Methodologies include matrix factorization [20] that decomposes user-item interaction matrix to capture latent relationships between users and items, and Bayesian personalized ranking (BPR) [27] that operates by determining personalized ranking of items based on user preferences.

For stock recommendations, [31] propose a CF-based model that takes into account both individual preferences and portfolio diversification. However, CF and portfolio optimization are performed in distinct steps, and such a heuristic approach would lead to suboptimal results. [4] were the first to develop a holistic model that can effectively handle the trade-off between individual preferences and investment performance. They incorporated modern portfolio theory into a matrix factorization model, as well as developed an associated ranking loss function which can be applied to more advanced models (e.g., GNN-based models). However, all these models do not consider the temporal dynamics of stock features and user preferences.

On the other hand, there have been several attempts to adapt temporal models for stock recommender systems [7, 8, 37]. However, they focus predominantly on price prediction without consideration of the individual preferences of users. Meanwhile, [9, 32] address the dynamic characteristics of financial markets and user preference, but they fail to systemically address the diversification effect. In contrast, our model aims to simultaneously consider both user preferences and the diversification effect, optimizing stock recommendations holistically.

2.2 Dynamic Graph Learning

In recommender systems, many methodologies utilize graph convolutional networks (GCNs) [19]. This is because the user-item interactions form a graph structure, allowing effective representation learning from such graphs. For example, NGCF [38] leverages collaborative signals in high-order connectivities. LightGCN [10] is specifically designed to enhance scalability, resulting in accelerated training and inference times.

Unlike the typical graph neural networks (GNNs) that learn node embeddings in static graphs, learning embeddings in dynamic graphs where connections change over time requires considering the temporal aspect. For instance, TGAT [41] introduces a time encoding technique upon GAT [36], which is a graph attention mechanism applied to static graphs. In addition, TGN [28] proposes a more general framework that can incorporate node-wise temporal features. This is an encoder that generates node embeddings at each time step. While this framework has been utilized in various graph tasks, there has been no research applying it to recommender systems thus far. In this study, we aim to leverage this framework for recommendation purposes. Comparatively, there have been few works that attempt *dynamic* graph embedding in recommender systems. For example, TGSRec [6] introduces a temporal collaborative Transformer to explicitly model the temporal effects of interactions. Meanwhile, DGEL [33] refines embeddings based on previous time related embeddings. However, they rely on time encoding without the inclusion of an explicit memory updater, limiting their ability to effectively capture the node history. In contrast, our model utilizes the TGN framework for recommendation task to leverage its capabilities for explicitly embedding node memories with its strong embedding performance.

3 PRELIMINARIES

Problem Definition: Let us define the task associated with stock recommendations. We denote the set of users as $U = \{u_1, u_2, ...u_{|U|}\}$, the set of items (stocks) as $V = \{v_1, v_2, ..., v_{|V|}\}$, and the set of time points as $T = \{t_1, t_2, ...t_{|T|}\}$. Then, a user-item interaction can be represented as $y^t_{u,v}$. If user u purchases the item v at time t, then $y^t_{u,v} = 1$; otherwise $y^t_{u,v} = 0$. Our primary goal is to predict the value of $y^t_{u,v}$. Ultimately, for each user and time, the model aims to recommend the top-k items, leading to a personalized and time-sensitive set of stock recommendations that can improve the portfolio's investment performance.

Continuous Time Dynamic Graph: We construct a dynamic graph with user-item interactions, changing its structure over time. We define our continuous-time bipartite graph as $\mathcal{G}(T) = (\mathcal{V}, \mathcal{E}_T)$. Here, $\mathcal{V}$ represents the set of user and item nodes. $\mathcal{E}_T$ denotes the temporal set of edges. Each edge in $\mathcal{E}_T$ is characterized by a tuple $e = (u, v, t, \mathbf{e}_{uv})$, consisting of a user node u, an item node v, a timestamp t, and an edge feature $\mathbf{e}_{uv}$. If a user interacts with the same item multiple times, each interaction is represented as a distinct edge in the graph. This approach enables the construction of a dynamic graph that accurately captures the evolving relationships between nodes over time.

4 METHOD

We present the PfoTGNRec model, which consists of three integral components: (1) Dynamic embedding learning, where we utilize TGN encoder to effectively learn the evolving characteristics of user-item interactions, (2) Mean-variance efficient sampling, which involves strategic item sampling and designing contrastive pairs to enhance the user portfolio, and (3) Optimization, where the model is trained with Bayesian Personalized Ranking (BPR) loss.

4.1 Dynamic Embedding Learning

First, we learn node embeddings from our dynamic graph constructed from user-item interactions, which are later used when calculating recommendation scores.

4.1.1 *Memory embedding.*

We generate memory embeddings for each node to capture the dynamic nature. The process begins with the extraction of information from each node, termed as "message". In the case of an interaction between source node (i) and destination node (j) at time t, two messages are computed:

$$m_i(t) = s_i(t^-) \| s_j(t^-) \| \Delta t \| \mathbf{e}_{ij} \tag{1}$$

and

$$m_j(t) = s_j(t^-) \| s_i(t^-) \| \Delta t \| \mathbf{e}_{ji} \tag{2}$$

Here, $\|$ is a concatenation operator, $s_i(t^-)$ and $s_j(t^-)$ represent the memory at the previous time step for the source and destination nodes, respectively, Δt is the time interval $t - t^-$, and $\mathbf{e}_{ij}$ is the edge feature.

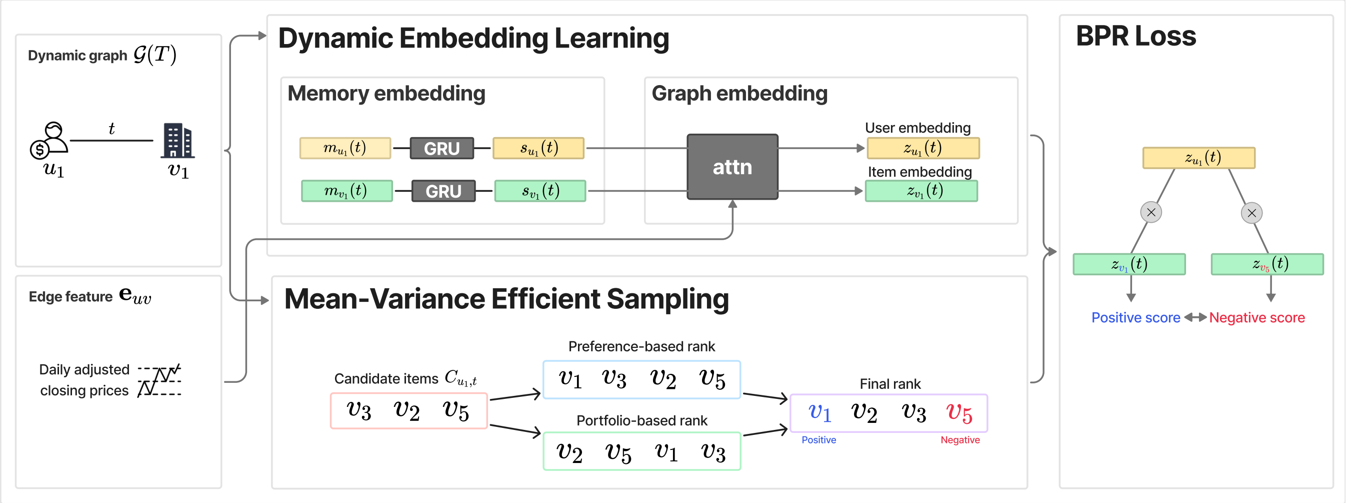

Figure 2: Model architecture of our proposed PfoTGNRec.

In the memory update process, a recurrent neural network approach is employed to update the memory of a node following each interaction that involves the node itself. Specifically, GRU [3] is utilized in our model, and the memory is updated as follows:

$$s_i(t) = GRU(m_i(t), s_i(t^-)) \tag{3}$$

4.1.2 Graph embedding. In this module, temporal embeddings for a dynamic graph are generated. In specific, embeddings are created for each node at time step t. Graph attention is utilized to effectively learn the connectivity between nodes. A node embedding can be represented as:

$$\mathbf{z}_i(t) = \sum_{j \in n_i^k(t)} attn\left(s_i(t), s_j(t), \mathbf{e}_{ij}\right) \tag{4}$$

where $attn$ refers to the graph attention mechanism as described in [28], $s_i(t)$ and $s_j(t)$ represent the memory, $\mathbf{e}_{ij}$ is the edge feature, and the neighborhood set of node i, denoted as $n_i^k(t)$, refers to the k-hop temporal neighbors connected at time t.

4.2 Mean-Variance Efficient Sampling

Unlike conventional recommender systems that sample contrastive pairs based on user purchase history or user-item similarity, we take into account the portfolio diversification effect. In other words, while performing positive and negative sampling based on user-item interactions, we are motivated by MVECF [4] to reflect rankings according to mean-variance when sampling items. Consider user u bought item v at time t. At this interaction point, a user's current portfolio $PO_{u,t}$ consists of items that the user holds at t, representing a collection of various stocks. Then, we randomly sample a set of candidate items $C_{u,t}$ that do not belong to the user portfolio $PO_{u,t}$, from the item set V.

$$C_{u,t} \leftarrow \text{sample}(V - PO_{u,t}) \tag{5}$$

Now, we create two ranked lists using the candidate items: (1) a preference-based list and (2) a portfolio-based list. In (1) preference-based list, the item that the user has actually purchased at that time is ranked first, and candidate items are ranked randomly in the remaining positions. (2) The portfolio-based list ranks the items based on their profitability and volatility, regardless of the user's preferences. This is done by calculating the mean-variance score for each item and ranking them in descending order of their scores.

The mean-variance score is designed to consider the effectiveness of adding an item to the existing portfolio in enhancing diversification effects. The modified target rating in the MVECF, y_{ui}^{MV}, is calculated as follows:

$$y_{ui}^{MV} = \frac{\frac{\mu_i}{\gamma} - \frac{1}{2}\sum_{j:j\neq i}\frac{1}{|y_u|}\sigma_{ij}}{\sigma_i^2} \tag{6}$$

Here, γ is a hyperparameter for risk-aversion level and $|y_u|$ represents the number of holdings of user u. We calculated the mean return and variance of items, denoted by μ and σ respectively, based on the prices over the next 30 days from the point of calculating the MV score. As the formula indicates, y_{ui}^{MV} assigns higher values to items that increase returns while decreasing risk, when added to the user's current portfolio.

To get the final rank of items, we combine the preference-based list and portfolio-based list by calculating a weighted sum of the rankings from them. Here, the weight λ_{MV} ranges from 0 to 1. Finally, we choose positive and negative items from the final rank. The items ranked at the top are sampled as positive items $P_{u,t}$, while the items ranked at the bottom are sampled as negative items $N_{u,t}$. In this study, we selected one top-ranked item as positive and three bottom-ranked items as negative.

$$P_{u,t} = \text{top-ranked items from the final rank} \tag{7}$$

$$N_{u,t} = \text{bottom-ranked items from the final rank} \tag{8}$$

4.3 BPR Loss

Following the typical recommender systems, we employ the BPR loss to train the model. At the time when the interaction takes place, we sample positive and negative items with mean-variance efficient sampling. Then, BPR loss is applied to calculate scores for pairs of positive and negative items.

$$\mathcal{L}_{BPR} = \sum_{(u,p,n,t) \in D} - \log \sigma \left(\mathbf{z}_u(t)^T \mathbf{z}_p(t) - \mathbf{z}_u(t)^T \mathbf{z}_n(t) \right) \quad (9)$$

In this equation, D denotes the edge set, which is derived from $\mathcal{E}_T$, u represents user, p represents the positive item selected from $P_{u,t}$, and n is the negative item selected from $N_{u,t}$

5 EXPERIMENT

In this section, we explain how we assessed the performance of our proposed model using a Greece trading dataset collected from real customer investment transactions. We formulated our experimental questions based on two pivotal aspects that ought to be considered in PfoTGNRec: recommendation and portfolio performance. We aim to answer the following research questions:

- **RQ1:** Can PfoTGNRec provide a better trade-off between recommendation and investment performance than past stock recommendation algorithms?
- **RQ2:** How effective is PfoTGNRec in comparison to past stock recommendation algorithms on predicting individual customer investments (recommendation performance)?
- **RQ3:** How profitable are the recommendations provided by PfoTGNRec in comparison to past stock recommendation algorithms (portfolio or investment performance)?
- **RQ4:** How do PfoTGNRec hyperparameters affect its investment and recommendation performance?

5.1 Experimental Settings

5.1.1 Dataset. We conduct experiments using individual investor transaction dataset, provided by National Bank of Greece [29]. This dataset includes real transaction data of users and represents a snapshot of the Greek market. The data spans from January 2018 to November 2022 comprising user buy orders during this period. To exclude abnormal transactions, we remove stocks with highly unstable price movements. We use daily adjusted closing prices for the temporal features of stocks, retrieved from an open source Python package *yfinance*.

For the sake of conducting stable experiments, we perform some filtering on items. We use stock price data from Yahoo Finance, and stocks and dates that do not exist in Yahoo Finance are excluded from the data. Additionally, to eliminate stocks that have been halted in trading, we remove stocks with no price changes for 30 consecutive days. As a result, the average number of interactions per user is 18.24, with a median of 5. For the number of interactions per item, the average is 1,653.09, with a median of 393.

To obtain real-time user portfolios of users during the data period, we utilize buy and sell orders along with the quantities of stocks ordered. Portfolios represent users' stock holdings for each interaction as the set of stocks held up to the day before. The average number of stocks in user portfolios is 6.26, with a median of 5. The minimum number of stocks is 0, and the maximum is 47. Most users hold fewer than 10 stocks in their portfolios.

For the edge features that change over time, we use daily adjusted closing prices of the most recent 30 trading days before each interaction.

For the data split, we utilize a chronological approach based on interaction timestamps to partition the dataset into training, validation, and testing sets. This division follows a ratio of 8:1:1, which preserves the temporal order of interactions. Ultimately, we use data consisting of 8,337 users, 92 stocks, and 152,084 interactions.

5.1.2 Baseline. We have selected the baseline models based on the following three categories.

Recommender models: We compare our model with competitive transaction-based algorithms, both static and dynamic. The *static* methods include Pop, BPR [27], WMF [11], LightGCN [10] which is a static graph learning method, and SGL [40] which leverages a self-supervised learning approach. For *dynamic* methods, we consider state-of-the-art dynamic graph learning models including DyRep [34], Jodie [21], TGAT [41], and TGN [28]. While most dynamic methods have not been utilized for recommendation tasks, we adapt their original architectures to recommendation task by incorporating negative sampling during training and applying BPR loss.

Price-based models: We include risk-return approaches that focus solely on prices rather than transactions. Return and Sharpe model refer to non-personalized models recommending stocks that had the best return and Sharpe ratio over the 30 days before the start of the testing period, respectively. Even the most sophisticated stock price forecasting models (e.g., [42]) show an accuracy around 55%, these simple models can serve as good proxies of such models.

Stock recommendation models: We consider the two most advanced stock recommendation models, which are the two-step method [31] and MVECF [4]. Both can be regarded as *static* methods.

5.1.3 Evaluation (Recommendation). For the evaluation of recommendation performance, we employ the Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG). All models follow an interaction-based ranking strategy, consistent with the settings in [21]. In other words, for each testing interaction (u, v, t), a list of recommended items was generated. For static models that cannot provide different recommendations for each test interaction, the same item set ranked within the train period is used throughout all test periods. To evaluate the performance, we utilize items from the entire item set, excluding those that are in the user's portfolio at each time point.

5.1.4 Evaluation (Investment). To evaluate investment performance, we utilize return and Sharpe ratio. For all models, we compare the user's original portfolio with the portfolio after the recommendation for each testing interaction (u, v, t). We constructed the recommended portfolio by adding the top K stocks with the highest recommendation scores to the original portfolio.

In specific, we measure the improvement of investment performance in two ways. First, difference. The difference in the Sharpe ratio and return is denoted as $\triangle SR = SR - SR_{\text{init}}$ and $\triangle R = R - R_{\text{init}}$, respectively. Here, terms appended with "init" represent the original portfolio (before recommendation), whereas those without the suffix refer to the recommended portfolio (after recommendation). These values are calculated for all users and then they are averaged. Second, the percentage of users whose investment performance becomes better after the recommendation. The improvement percentage in the Sharpe ratio and return are expressed as $P(SR) = P(SR > SR_{\text{init}})$ and $P(R) = P(R > R_{\text{init}})$. To evaluate the actual portfolio performance in the stock market when a stock is recommended, we employ out-of-sample assessments. That is, at the testing point, the investment performance is calculated based on the returns over the next 30 days.

5.1.5 Configuration. We train all models for 20 epochs and the reported results are based on the test data with the best performing model selected within the validation set. For model selection, we use NDCG@5 for recommender, price-based models, and a holistic approach for stock recommendation models and our model, considering both recommendation and investment performance. This is achieved by employing the validation data to independently rank the models based on their performance in NDCG@5 and $P(SR)$@5. Then, the averages of these rankings are used to determine the final model. For a fair comparison, we conduct hyperparameter tuning for all models.

5.2 Combined Recommendation and Portfolio Performance (RQ1)

For a comprehensive evaluation of user preferences and portfolio performance, we select two representative metrics: NDCG@5 and P(SR)@5. NDCG@5 is a recommendation performance metric that measures how highly the items actually purchased by the user are ranked on the list of recommended items. P(SR)@5 is an investment performance metric that measures the proportion of users who experienced an improvement in their portfolio Sharpe ratio. These metrics are visualized in Figure 3. Since higher values for both metrics indicate better performance, models positioned at the outermost points in the graph exhibit the most balanced performance. As shown in Figure 3, our model demonstrates the best performance.

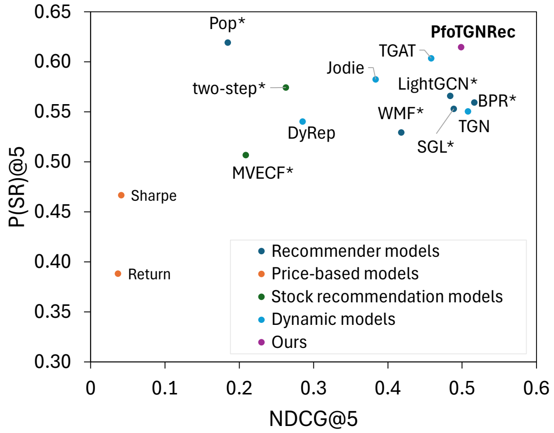

Figure 3: Visualization of Comparison of Both Recommendation and Portfolio Performance

Model	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$
Pop	0.3149	0.3584	0.4019	0.4454	0.4889
WMF	0.4516	0.4627	0.4738	0.4850	0.4961
BPR	0.5294	0.5337	0.5380	0.5423	0.5466
LightGCN	0.5087	0.5169	0.5250	0.5332	0.5414
SGL	0.5081	0.5145	0.5210	0.5274	0.5338
Return	0.1422	0.1774	0.2126	0.2477	0.2828
Sharpe	0.1688	0.2113	0.2539	0.2965	0.3390
MVECF	0.2981	0.3279	0.3578	0.3876	0.4174
two-step	0.3563	0.3875	0.4186	0.4497	0.4809
DyRep	0.3617	0.3872	0.4128	0.4383	0.4638
Jodie	0.4434	0.4632	0.4831	0.5030	0.5228
TGAT	0.5021	0.5166	0.5311	0.5456	0.5601
TGN	0.5207	0.5250	0.5292	0.5335	0.5378
PfoTGNRec	**0.5334**	**0.5450**	**0.5566**	**0.5683**	**0.5799**

Note: The best and the second best performing models are highlighted in bold and underline, respectively.

Table 1: Comparison of Models Based on Weighted Metric of Recommendation and Portfolio Performance

To provide a precise numerical comparison, we use a combined metric of NDCG@5 and P(SR)@5. We calculate the weighted average of these two metrics using the weight α. Specifically, we compute

$$m@5(\alpha) = NDCG@5 \times (1 - \alpha) + P(SR)@5 \times \alpha \qquad (10)$$

and vary α from 0.3 to 0.7 to observe performance across different balances. Table 1 displays the overall performance compared to the baseline for various values of α. Our model consistently outperforms all baseline models across both recommendation and investment metrics. Therefore, it is evident that our model offers the most balanced approach, enhancing investment performance while reflecting individual preferences.

5.3 Recommendation Performance (RQ2)

As shown in Table 2, recommender models consistently outperform price-based and stock recommendation models, demonstrating their effectiveness in capturing user preferences. Interestingly, despite the expectation that dynamic models would surpass static models in performance, both types exhibited similar performance. To investigate this, we further analyze the results based on testing interactions where items not purchased during the training period were subsequently purchased. This analysis reveals a significant drop in the performance of static recommendation models, while dynamic recommendation models perform markedly better. For example, the best-performing static model, BPR, achieves an NDCG@5 of 0.0416. In contrast, the best-performing dynamic model, TGN, achieves an NDCG@5 of 0.4535. This difference underscores the limitations of static recommendation models, which tend to recommend items that users have already purchased. In the context of stock recommendation, users may indeed repurchase previously bought items. However, recommending only previously purchased items does not contribute to portfolio diversification. Consequently, dynamic models clearly have an advantage in offering more diverse recommendations.

	Recommendation effectiveness				Portfolio performance							
Model	HR@3	HR@5	NDCG@3	NDCG@5	P(R)@3	P(R)@5	P(SR)@3	P(SR)@5	ΔR@3	ΔR@5	ΔSR@3	ΔSR@5
Pop*	0.1586	0.2787	0.1355	0.1845	0.5174	**0.5479**	0.5670	**0.6193**	-0.003	0.0106	0.1860	0.3533
WMF*	0.4654	0.5588	0.3797	0.4183	0.4561	0.4417	0.5228	0.5294	-0.0212	-0.0379	0.0374	0.0408
BPR*	0.5635	0.6538	**0.4794**	**0.5166**	0.5234	0.4970	0.5595	0.5594	0.0064	-0.0079	0.1499	0.1555
LightGCN*	0.5378	0.6399	0.4419	0.4841	0.5333	0.5041	0.5712	0.5660	0.0083	-0.0055	0.1664	0.1663
SGL*	0.5297	0.6054	0.4578	0.4888	0.5071	0.4912	0.5558	0.5531	-0.0003	-0.0223	0.1325	0.0908
Return	0.0389	0.0621	0.0274	0.0368	0.3065	0.3438	0.3403	0.3883	-0.1747	-0.1819	-0.5236	-0.4699
Sharpe	0.0453	0.0665	0.0324	0.0411	0.4137	0.4174	0.4743	0.4667	-0.0832	-0.1011	-0.1269	-0.1362
two-step*	0.2767	0.3834	0.2193	0.2629	0.4479	0.4425	0.5526	0.5743	-0.0227	-0.0335	0.1457	0.1849
MVECF*	0.2170	0.2321	0.2025	0.2087	0.4286	0.4149	0.5081	0.5068	-0.0426	-0.0644	-0.0281	-0.0482
DyRep	0.3047	0.4533	0.2243	0.2852	0.4581	0.4499	0.5383	0.5403	-0.0235	-0.034	0.0769	0.0919
Jodie	0.4324	0.5757	0.3247	0.3838	0.5156	0.4924	0.5757	0.5824	0.0074	-0.0022	0.2186	0.2617
TGAT	0.5138	0.6318	0.4100	0.4585	**0.5826**	0.5423	**0.6129**	0.6037	**0.0460**	0.0343	**0.3178**	0.3452
TGN	**0.5673**	**0.6809**	0.4611	0.5079	0.5405	0.5107	0.5612	0.5506	0.0260	0.0075	0.1959	0.1899
PfoTGNRec	0.5572	0.6674	0.4532	0.4986	0.5652	0.5434	0.6125	0.6147	0.0407	**0.0349**	0.3053	**0.3649**

Note: Models with * exclude cold start user results. The best and second best performing models are highlighted in bold and underline, respectively.

Table 2: Comparison of Models Based on Various Metrics

Compared to dynamic recommender models, our model outperforms other models but falls slightly short of TGN. This is because we intentionally sacrificed a certain level of recommendation performance through mean-variance efficient sampling to enhance the diversification effect. As expected, price-based models and existing stock recommendation models show lower recommendation performance. These models do not effectively capture individual preferences.

5.4 Portfolio Performance (RQ3)

In terms of investment performance, the results indicate that our model generally recorded superior performance across most metrics, despite a few exceptions. In particular, the price-based methods, the Return model, and the Sharpe model rank near the bottom in terms of investment performance, demonstrating the difficulty of predicting future prices based on past prices. This highlights the fundamental challenge of stock price prediction. Surprisingly, the Pop model shows high investment performance in some metrics, which appears to be due to the presence of a popularity bias in the data regarding investment performance. However, this model is not suitable as a stock recommendation model because its recommendation performance is very poor. Interestingly, the stock recommendation models, Two-Step and MVECF, have failed to demonstrate competitive investment performance. This is likely due to their inability to effectively manage the dynamic nature of stock features and user behaviors.

5.5 Hyperparameter Study (RQ4)

To thoroughly investigate the impact of various hyperparameters on our model's performance, we conduct an extensive hyperparameter study with six key hyperparameters: batch size, node dimension, number of candidate items, number of negative items, γ, and λ_{MV}. For evaluation metrics, we select NDCG@5 to represent recommendation performance and P(SR)@5 to represent portfolio performance. By analyzing these metrics, we aim to derive insights into the trade-offs and interactions between different hyperparameters, thereby guiding the optimization of our model for both recommendation and investment tasks. The results are shown in Figure 4.

- **Batch Size**: The analysis reveals that NDCG@5 exhibits a slight increase with larger batch sizes, while P(SR)@5 demonstrates a decreasing trend. This indicates that although larger batches may enhance recommendation quality, they can negatively impact investment performance more than they improve recommendation effectiveness.
- **Node Dimension**: NDCG@5 remains relatively stable across varying node dimensions. However, P(SR)@5 displays notable fluctuations, indicating that the choice of node dimension is critical. Additionally, larger node dimensions can store more information, potentially enhancing performance.
- **Number of Candidate Items**: The number of candidate items does not appear to significantly influence NDCG@5. However, P(SR)@5 shows some variability with changes in the number of candidate items. As the number of candidate items increases, the complexity of mean-variance efficient sampling also rises, highlighting the importance of determining an optimal number of candidate items.
- **Number of Negative Items**: Both NDCG@5 and P(SR)@5 show an increasing trend with the number of negative items. This positive correlation suggests that incorporating a higher number of negative samples enhances both recommendation and investment performance. However, increasing the number of negative items also may lead to longer computation times, making it crucial to determine an optimal number of negative items.
- $\boldsymbol{\gamma}$: It exhibits a peak in both NDCG@5 and P(SR)@5 at the value of 1. When performing mean-variance efficient sampling, the weight given to the volatility relative to the return of the stocks

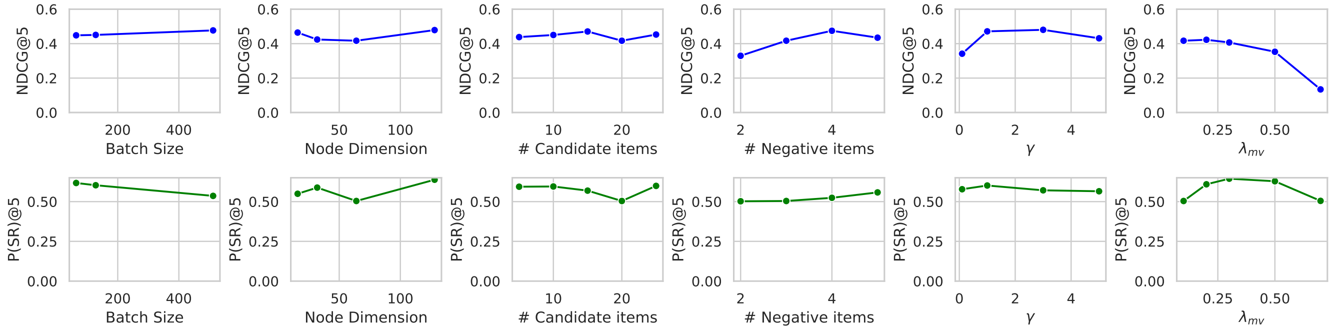

Figure 4: Hyperparameter study of the PfoTGNRec

changes according to the value of γ. Therefore, optimizing this hyperparameter is crucial.

- **λ_{MV}**: Our findings reveal that λ_{MV} has the most significant impact on both metrics. NDCG@5 decreases sharply with increasing λ_{MV}, indicating a negative impact on recommendation performance. In contrast, P(SR)@5 shows a peak at a specific λ_{MV} value. Although λ_{MV} is a hyperparameter that balances recommendation and portfolio performance, the inherent uncertainty in predicting future prices leads to inconsistent impact on investment performance.

6 CONCLUSION

In this paper, we present PfoTGNRec, a novel framework tailored for stock recommender systems, focusing on two key aspects: capturing the temporal dynamics of the stock market and user preference and integrating portfolio diversification into recommendations. Using temporal graph networks, PfoTGNRec effectively models user preferences that change over time, with a novel training approach specifically designed for portfolio diversification, balancing user preferences with investment risk management. Experiments have demonstrated PfoTGNRec's effectiveness, showing competitive recommendation accuracy and improved portfolio performance.

For future work, we propose incorporating static features of users and items as node features to enhance the model's ability to capture inherent characteristics that influence user behavior and stock performance. Additionally, expanding the recommendation system to account for various user behaviors, such as selling, holding, and ordering will provide a more comprehensive understanding of users and improve the relevance of recommendations. By addressing these areas, we aim to further enhance the robustness and applicability of PfoTGNRec in the real world.

ACKNOWLEDGMENTS

This work was supported by National Research Foundation of Korea (NRF) grant (No. NRF-2022R1I1A4069163) and Institute of Information & Communications Technology Planning & Evaluation (IITP), South Korea grant (No. RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST)) funded by the Korea government (MSIT).

REFERENCES

[1] Utpal Bhattacharya, Andreas Hackethal, Simon Kaesler, Benjamin Loos, and Steffen Meyer. 2012. Is Unbiased Financial Advice To Retail Investors Sufficient? Answers From a Large Field Study. *The Review of Financial Studies* 25, 4 (2012), 975–1032. https://doi.org/10.1093/rfs/hhr127

[2] Andrew C. Chang, Aditya Aladangady, Jesse Bricker, Sarena Goodman, Jacob Krimmel, Kevin B. Moore, Sarah Reber, Alice Henriques Volz, and Richard Windle. 2023. *Changes in US Family Finances from 2019 to 2022.* Technical Report. Board of Governors of the Federal Reserve System. https://doi.org/10.17016/8799

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).* ACL, Doha, Qatar, 1724–1734. https://doi.org/10.3115/v1/D14-1179

[4] Munki Chung, Yongjae Lee, and Woo Chang Kim. 2023. Mean-Variance Efficient Collaborative Filtering for Stock Recommendation. arXiv:2306.06590

[5] DALBAR.com. 2023. DALBAR Products and Services: QAIB. https://www.dalbar.com/ProductsAndServices/QAIB Accessed: 2023-06-18.

[6] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S Yu. 2021. Continuous-time Sequential Recommendation with Temporal Graph Collaborative Transformer. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021).* ACM, Virtual Event, 433–442. https://doi.org/10.1145/3459637.3482242

[7] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal Relational Ranking for Stock Prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–30. https://doi.org/10.1145/3309547

[8] Jianliang Gao, Xiaoting Ying, Cong Xu, Jianxin Wang, Shichao Zhang, and Zhao Li. 2021. Graph-Based Stock Recommendation by Time-Aware Relational Attention Network. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 1 (2021), 1–21. https://doi.org/10.1145/3451397

[9] Ashraf Ghiye, Baptiste Barreau, Laurent Carlier, and Michalis Vazirgiannis. 2023. Adaptive Collaborative Filtering with Personalized Time Decay Functions for Financial Product Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems (RecSys 2023).* ACM, Singapore, 798–804. https://doi.org/10.1145/3604915.3608832

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020).* ACM, Virtual Event, 639–648. https://doi.org/10.1145/3397271.3401063

[11] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008).* IEEE, Pisa, Italy, 263–272. https://doi.org/10.1109/ICDM.2008.22

[12] Yoontae Hwang, Yongjae Lee, and Frank J Fabozzi. 2023. Identifying household finance heterogeneity via deep clustering. *Annals of Operations Research* 325, 2 (2023), 1255–1289. https://doi.org/10.1007/s10479-022-04900-3

[13] Yoontae Hwang, Junpyo Park, Jang Ho Kim, Yongjae Lee, and Frank J. Fabozzi. 2024. Heterogeneous trading behaviors of individual investors: A deep clustering approach. *Finance Research Letters* 65 (2024), 105481. https://doi.org/10.1016/j.frl.2024.105481

[14] MirZat Ullah Khan. 2017. Impact of Availability Bias and Loss Aversion Bias on Investment Decision Making, Moderating Role of Risk Perception. *IMPACT: Journal of Modern Developments in General Management & Administration (IMPACT: JMDGMA)* 1, 1 (2017), 17–28.

[15] Jang Ho Kim, Yongjae Lee, Jaekyu Bae, and Woo Chang Kim. 2021. Recent Trends and Perspectives on the Korean Asset Management Industry. *Journal of Portfolio Management* 47, 7 (2021), 248. https://doi.org/10.3905/jpm.2021.1.248

[16] Jang Ho Kim, Yongjae Lee, Woo Chang Kim, and Frank J. Fabozzi. 2021. Mean-variance Optimization for Asset Allocation. *Journal of Portfolio Management* 47, 5 (2021), 24–40. https://doi.org/10.3905/jpm.2021.1.219

[17] Jang Ho Kim, Yongjae Lee, Woo Chang Kim, and Frank J. Fabozzi. 2022. Goal-based investing based on multi-stage robust portfolio optimization. *Annals of Operations Research* 313, 2 (2022), 1141–1158. https://doi.org/10.1007/s10479-021-04473-7

[18] Woo Chang Kim, Do-Gyun Kwon, Yongjae Lee, Jang Ho Kim, and Changle Lin. 2020. Personalized goal-based investing via multi-stage stochastic goal programming. *Quantitative Finance* 20, 3 (2020), 515–526. https://doi.org/10.1080/14697688.2019.1662079

[19] Thomas N. Kipf and Max Welling. 2017. Semi-supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*. Toulon, France.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

[21] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2018. Learning Dynamic Embeddings From Temporal Interactions. arXiv:1812:02289

[22] Yongjae Lee, John R.J. Thompson, Jang Ho Kim, Woo Chang Kim, and Francesco A. Fabozzi. 2023. An Overview of Machine Learning for Asset Management. *Journal of Portfolio Management* 49, 9 (2023), 31–63. https://doi.org/10.3905/jpm.2023.1.526

[23] Harry Markowitz. 1952. Portfolio Selection. *The Journal of Finance* 7, 1 (1952), 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

[24] Binoy B. Nair, V.P. Mohandas, Nikhil Nayanar, E.S.R. Teja, S. Vigneshwari, and K.V.N.S. Teja. 2015. A Stock Trading Recommender System Based on Temporal Association Rule Mining. *SAGE Open* 5, 2 (2015), 2158244015579941. https://doi.org/10.1177/21582440155799

[25] Luu Thi Bich Ngoc. 2014. Behavior Pattern of Individual Investors in Stock Market. *International Journal of Business and Management* 9, 1 (2014), 1–16. https://doi.org/10.5539/ijbm.v9n1p1

[26] Preeti Paranjape-Voditel and Umesh Deshpande. 2013. A stock market portfolio recommender system based on association rule mining. *Applied Soft Computing* 13, 2 (2013), 1055–1063. https://doi.org/10.1016/j.asoc.2012.09.012

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009)*. AUAI Press, Montreal, Canada.

[28] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. In *Proceedings of the ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*. Virtual Event.

[29] Javier Sanz-Cruzado, Nikolaos Droukas, and Richard McCreadie. 2024. FAR-Trans: An Investment Dataset for Financial Asset Recommendation. In *Proceedings of the IJCAI 2024 Workshop on Recommender Systems in Finance (Fin-RecSys 2024)*. Jeju, South Korea.

[30] Javier Sanz-Cruzado, Richard McCreadie, Nikolaos Droukas, Craig Macdonald, and Iadh Ounis. 2022. On Transaction-Based Metrics as a Proxy for Profitability of Financial Asset Recommendations. In *Proceedings of the 3rd International Workshop on Personalization & Recommender Systems (FinRec 2022)*. Seattle, WA, USA.

[31] Robin M.E. Swezey and Bruno Charron. 2018. Large-scale Recommendation for Portfolio Optimization. In *Proceedings of the 12th ACM Conference on Recommender Systems (RecSys 2018)*. ACM, Vancouver, BC, Canada, 382–386. https://doi.org/10.1145/3240323.3240386

[32] Takehiro Takayanagi, Chung-Chi Chen, and Kiyoshi Izumi. 2023. Personalized Dynamic Recommender System for Investors. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023)*. ACM, Taipei, Taiwan, 2246–2250. https://doi.org/10.1145/3539618.3592035

[33] Haoran Tang, Shiqing Wu, Guandong Xu, and Qing Li. 2023. Dynamic Graph Evolution Learning for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023)*. ACM, Taipei, Taiwan, 1589–1598. https://doi.org/10.1145/3539618.359167

[34] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. 2019. DyRep: Learning Representations over Dynamic Graphs. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*. New Orleans, LA, USA.

[35] Wenting Tu, David W Cheung, Nikos Mamoulis, Min Yang, and Ziyu Lu. 2016. Investment Recommendation using Investor Opinions in Social Media. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2016)*. ACM, Pisa, Italy, 881–884. https://doi.org/10.1145/2911451.2914699

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*. Vancouver, BC, Canada.

[37] Changhai Wang, Hui Liang, Bo Wang, Xiaoxu Cui, and Yuwei Xu. 2022. MG-Conv: A spatiotemporal multi-graph convolutional neural network for stock market index trend prediction. *Computers and Electrical Engineering* 103 (2022), 108285. https://doi.org/10.1016/j.compeleceng.2022.108285

[38] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR 2019)*. ACM, Paris, France, 165–174. https://doi.org/10.1145/3331184.3331267

[39] Ivo Welch. 2022. The Wisdom of the Robinhood Crowd. *The Journal of Finance* 77, 3 (2022), 1489–1527. https://doi.org/10.1111/jofi.13128

[40] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-Supervised Graph Learning for Recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval (SIGIR 2021)*. ACM, Virtual Event, 726–735. https://doi.org/10.1145/3404835.3462862

[41] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020)*. Virtual Event.

[42] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. 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 (KDD 2021)*. ACM, Virtual Event, 2037–2045. https://doi.org/10.1145/3447548.3467297