Automatic De-Biased Temporal-Relational Modeling for Stock Investment Recommendation

Weijun Chen[1,2,3], **Shun Li**[4*], **Xipu Yu**[5], **Heyuan Wang**[1,2,3], **Wei Chen**[1,2,3] and **Tengjiao Wang**[1,2,3]

[1]Key Lab of High Confidence Software Technologies (MOE), School of Computer Science, Peking University, Beijing, China

[2]Research Center for Computational Social Science, Peking University

[3]Institute of Computational Social Science, Peking University (Qingdao)

[4]University of International Relations

[5]New York University

oncecwj@stu.pku.edu.cn, lishunmail@foxmail.com, xy2253@nyu.edu, {wangheyuan,pekingchenwei,tjwang}@pku.edu.cn

Abstract

Stock investment recommendation is crucial for guiding investment decisions and managing portfolios. Recent studies have demonstrated the potential of temporal-relational models (*TRM*) to yield excess investment returns. However, in the complicated finance ecosystem, the current *TRM* suffer from both the intrinsic temporal bias from the low signal-to-noise ratio (*SNR*) and the relational bias caused by utilizing inappropriate relational topologies and propagation mechanisms. Moreover, the distribution shifts behind macro-market scenarios invalidate the underlying *i.i.d.* assumption and limit the generalization ability of *TRM*. In this paper, we pioneer the impact of the above issues on the effective learning of temporal-relational patterns and propose an Automatic De-Biased Temporal-Relational Model (*ADB-TRM*) for stock recommendation. Specifically, *ADB-TRM* consists of three main components, i.e., (i) *a meta-learned architecture* forms a dual-stage training process, with the inner part ameliorating temporal-relational bias and the outer meta-learner counteracting distribution shifts, (ii) *automatic adversarial sample generation* guides the model adaptively to alleviate bias and enhance its profiling ability through adversarial training, and (iii) *global-local interaction* helps seek relative invariant stock embeddings from local and global distribution perspectives to mitigate distribution shifts. Experiments on three datasets from distinct stock markets show that *ADB-TRM* excels state-of-the-arts over 28.41% and 9.53% in terms of cumulative and risk-adjusted returns.

1 Introduction

The stock market is an indispensable component of the financial ecosystem, which enables a large body of transactions between businesses and investors. Exploring methods for stock prediction carries significant economic implications and academic research value. With the rapid development of deep learning, deep neural networks have become a promising avenue for stock prediction. Most existing deep learning solutions formulate stock prediction as a classification (to predict stock trends) or a regression problem (to predict stock prices), which are not optimized toward the investment target, i.e., selecting the stocks that yield the highest profit. Recent breakthroughs in *TRM* target the disparity between optimizing optimal stock selection for maximizing profit [Sawhney *et al.*, 2021a; Wang *et al.*, 2022c; Feng *et al.*, 2019b; Sawhney *et al.*, 2021b; Wang *et al.*, 2022b; He *et al.*, 2022]. They jointly model temporal states and relational interactions to capture fine granular evolving patterns of the stock market. Despite previous successes, there are two significant challenges that have yet to be thoroughly explored.

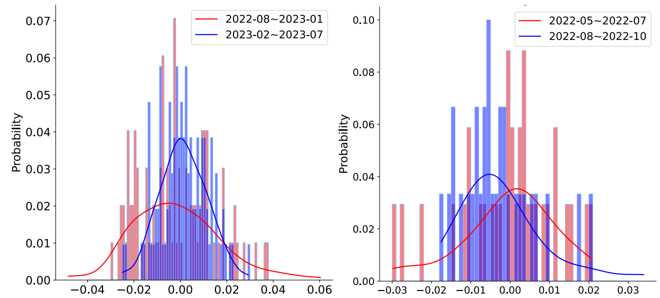

Figure 1: The left figure shows distribution shifts of stock market volatility in the NASDAQ Market, while the right figure shows distribution shifts in the Tokyo Stock Exchange Market. The horizontal axis represents the change of stock market volatility, while the vertical axis represents the probability of different volatility values.

The first challenge is the temporal-relational bias of individual and grouped stock objects. Unlike general time-series tasks, the stock market is characterized by a low *SNR* nature. The mixture of multi-source noise can obscure the discovery of reliable prediction patterns and genuinely effective influ-

*Corresponding Author.

encing factors, which leads to unstable gradient-based training and ineffective volatility patterns learning. To identify sound market signals, many stock prediction methods adopt a multi-level frequency decomposition approach [Zhang *et al.*, 2017; Rezaei *et al.*, 2021]. However, they may destroy the original statistical indicators, which are critical task-specific information. Some other models apply complex attention mechanisms or hyper-networks to capture volatility patterns [Wang *et al.*, 2022c; Zhao *et al.*, 2022; Huynh *et al.*, 2023], while the complex designs will exacerbate the overfitting in the low *SNR* stock data. On the other hand, relational bias can arise from the modeling of momentum spillover effects. In the stock market, the fluctuation event of a stock object not only determines its future movement but may also spread to associated companies [Cheng and Li, 2021; Zhao *et al.*, 2022; Hsu *et al.*, 2021; Wang *et al.*, 2021; Wang *et al.*, 2022a]. Existing methods exploiting stock relations generally rely on predefined adjacency graphs, which manifest the scale-free characteristics with a long-tailed distribution of node degrees. Hence, during the propagation of the observed relational signals through graph neural networks (GNN), a small number of hub stocks tend to receive the majority of momentum spillovers. This inevitably induces a notable bias between the attention and returns of various stock objects, as well as unfairness in investment recommendations. To alleviate such issues, the *HyperStockGAT* [Sawhney *et al.*, 2021b] conducts relational modeling on the Riemannian Manifolds, and *ALSP-TF* [Wang *et al.*, 2022c] applies multi-dimensional dynamic time warping to reconstruct the inter-stock dependencies from historical price curves. However, the evolution of stock correlations over time are overlooked. Furthermore, as the *learning-to-rank* loss function prioritizes top-ordered stocks and the influence of blended noise in the stock market, the optimization process is still prone to learning highly biased and noisy graphs.

Apart from the micro-level bias of stock objects, the macro-market scenarios also suffer from distribution shifts due to non-stationary properties and economic cycles [Lin *et al.*, 2021; Næs *et al.*, 2011]. Figure 1 gives the illustrative examples. Even between adjacent time periods, the volatility of the stock market may undergo substantial changes. The distribution shifts actually invalidate the strict *i.i.d.* assumption in many stock prediction methods including current *TRM*. Neglecting this distribution shift phenomenon will have a detrimental impact on the model's prediction accuracy and stability. However, the common practices for handling distribution shifts including causal inference and domain generalization [Granger, 1969; Wang *et al.*, 2022d] are inapplicable in this task, as it is hard to explore causality and prior domain identifiers for the evolution of financial time series.

In this paper, we propose *ADB-TRM*, an **A**utomatic **D**e-**B**iased **T**emporal-**R**elational **M**odel to counter the above challenges in a unified meta-learning framework. *ADB-TRM* facilitates the extraction of invariant information from both global and local volatility distributions, which has two optimization stages. One is the inner meta-layer stage, which focuses on alleviating the temporal-relational bias through adversarial training in a local view, and the other is the outer meta-learner that interacts with the inner part to discover relative invariant information for alleviating distribution shifts in a global perspective. Specifically, we improve the robustness of extracting temporal dynamics in a noise-aware adversarial manner without requiring the decomposition of the original time series or attaching excessive model complexity. To leverage unbiased structures, we devise an adaptive dynamic graph learning approach for extracting inter-stock dependencies and automatically generate adversarial graph samples to steer the learned graphs away from highly biased conditions, which facilitates the robustness and fairness of our model's encoder and decoder. Our major contributions are as follows:

- We propose *ADB-TRM* with the aim of alleviating the bias from diverse stock objects and market scenarios in investment recommendations. The research plays a pioneering role in making accurate investment decisions by de-biasing the intrinsic stock market biases.
- The proposed de-biasing paradigm improves the model's adaptability to dynamic market conditions and enhances its resilience against malicious attacks, a capability that current *TRM* lack. Detailed experiments also demonstrate the superior training convergence of the model.
- Through extensive experiments on three long-term datasets from NYSE, NASDAQ, and TSE exchange markets, we demonstrate the exceptional performance of *ADB-TRM* compared to state-of-the-art approaches.

2 Related Work

2.1 Stock Investment Recommendation

Stock investment recommendation redefines stock prediction as a *learning-to-rank* task. This approach necessitates choosing stocks with the highest returns from a vast stock pool, making the inter-stock correlations crucial additional market signals, which can significantly enhance returns and minimize losses. Recent work enhances the temporal extraction module and emphasizes the correlation information between stocks [Sawhney *et al.*, 2021a; Wang *et al.*, 2022c; Huynh *et al.*, 2023; Zheng *et al.*, 2023; Sawhney *et al.*, 2021b; Hsu *et al.*, 2021]. The above models are temporal-relational and have realized state-of-the-art profitability.

2.2 Adversial Training

Adversarial training [Goodfellow *et al.*, 2014] can reveal the defects of models and improve their robustness. Most adversarial examples are generated by adding small perturbations to clean samples. We focus on the application of adversarial training in stock prediction. [Feng *et al.*, 2019a] explores the potential of adversarial training in stock movement prediction, small perturbations are added to the hidden representation to improve the generalization of the model. Adversarial training is exploited in STLAT [Li *et al.*, 2022] to improve the generalization against the stochasticity of stock. [Liang *et al.*, 2018] proposes to combine deep reinforcement learning with adversarial training, which adds random noise to the market stock prices, improving the training efficiency and promoting average daily return and sharp ratio in the backtest. [Khuwaja *et al.*, 2021] proposes to use two adversarial networks to increase the efficacy of stock prediction. The two adversarial networks are heterogeneous data fusion representing market crash Q-learning and confrontational Q-learning network.

3 Proposed Framework

In this section, we describe our framework of *ADB-TRM* for stock ranking in detail. The framework is shown in Figure 2. We first present the stock recommendation task in our paper.

3.1 Stock Investment Recommendation

We conceptualize the task of stock prediction as a *learning-to-rank problem*, with a focus on developing it for practical investment recommendations. Let c_i^t be the closing price of stock i at time t, $y_i^t = \frac{c_i^t - c_i^{t-1}}{c_i^{t-1}}$ is the associated 1-day return ratio. Given the stock relational graph $\mathcal{G}_s \in \mathbb{R}^{N\times N}$ and stock prices $\mathcal{X}^t \in \mathbb{R}^{N\times D}$, where N represents the number of all stocks, and D represents the feature dimension of each stock. Our objective is to develop a model $\mathcal{F}_\theta(\cdot)$, that accurately predicts the ranking of all stocks at a given time t. This ranking, $\mathcal{P}^t = \mathcal{F}_\theta(\mathcal{G}_s, \mathcal{X}^t) = \{p_1^t \geq p_2^t \geq \ldots \geq p_N^t\}$, is arranged such that stocks expected to generate higher revenues are ranked higher. The optimal outcome is to ensure that for any pair of stocks $s_i, s_j \in \mathcal{S}$, the stock s_i is ranked higher than s_j (i.e., $p_i^t \geq p_j^t$) if and only if $y_i^t \geq y_j^t$.

3.2 Rank Loss

The objective of the rank loss is to assist in the identification of investment opportunities, specifically targeting the selection of top-ranked stocks [Feng *et al.*, 2019b]. In ranking optimization, we obtain predicted return ratios for stocks on day t. Subsequently, we perform concurrent calculations for point-wise regression and pair-wise ranking losses, introducing a weighting coefficient denoted as λ. This approach is designed to minimize the disparity between the predicted values $p^t[1:N]$ and the actual values $y^t[1:N]$, while simultaneously preserving the relative order of the stocks:

$$\mathcal{L}_{\mathcal{P}} = \sum_{i=1}^{N} \|p_i^t - y_i^t\|^2 + \lambda \sum_{i=1}^{N}\sum_{j=1}^{N} max(0, -(p_i^t - p_j^t)(y_i^t - y_j^t)). \tag{1}$$

3.3 De-biased Temporal-Relational Model

Following the *bilevel* optimization view [Hospedales *et al.*, 2021], we first dive into the inner meta-layer stage.

Adaptive Graph Learning

First of all, we intend to infer static relational graph $\mathcal{G}_{st}$ to reveal the stable correlations (e.g., sector and industry). Using the low-rank approximation method, we randomly initialize two learnable parameters $E_1 \in \mathbb{R}^{N\times F}, E_2 \in \mathbb{R}^{F\times N}$ as two stock embedding dictionaries, where F is the hidden dimension. We propose the static relational graph $\mathcal{G}_{st}$ as:

$$\mathcal{G}_{st} = E_1 E_2. \tag{2}$$

We name E_1 as the source stock embedding and E_2 as the target stock embedding. By multiplying E_1 and E_2, we derive the relational weights between the sources and the targets.

Later, we intend to infer a time-specific adjacency matrix $\mathcal{G}_{dy}^t$, which aims at capturing dynamic correlations between stocks. Specifically, $\mathcal{G}_{dy}^t[\boldsymbol{i},\boldsymbol{j}]$ captures how much stock $\boldsymbol{i}$ is affected by stock $\boldsymbol{j}$ at a specific timestamp t. We use the normalized embedded Gaussian function to capture the correlation between $\mathcal{X}_{\boldsymbol{i}}^t$ of stock $\boldsymbol{i}$ and $\mathcal{X}_{\boldsymbol{j}}^t$ of stock $\boldsymbol{j}$ at t:

$$\mathcal{G}_{dy}^t[\boldsymbol{i},\boldsymbol{j}] = \frac{e^{\rho(\mathcal{X}_i^t)^T \phi(\mathcal{X}_j^t)}}{\sum_{j=1}^{N} e^{\rho(\mathcal{X}_i^t)^T \phi(\mathcal{X}_j^t)}}. \tag{3}$$

Eq.3 resembles the attention mechanism. We first use two embedding functions ρ and ϕ, to perform a linear transformation on the data between stock entities $\boldsymbol{i}$ and $\boldsymbol{j}$. Specifically, an embedding function multiplies input data by a learnable weight matrix to generate linearly transformed data. We use two embedding functions because we want to distinguish source and target stocks, which enables us to capture asymmetric correlations between two stocks.

Besides, due to the smooth evolving nature of the graph structures [Ye *et al.*, 2022], we assume that the $\mathcal{G}_{dy}^t$ remains unchanged in a time interval while having evolutionary relationships between adjacent time intervals. The final stock relation graph $\mathcal{G}_{in}^t$ within the input time-span at start time point t_{in} can be calculated as follows:

$$\mathcal{G}_{in}^t = \sigma(\mathcal{G}_{dy}^{t_{in}} + \mathcal{G}_{st}), \tag{4}$$

where σ represents the Sigmoid activation function to normalize the value of the graph matrix to be $(\mathcal{G}_{in}^t)_{ij} \in [0,1]$.

Temporal-Relational Fusion

After obtaining $\mathcal{G}_{in}^t$, we combine the GRU [Chung *et al.*, 2014] with graph convolution [Kipf and Welling, 2017] as *Graph Convolutional Gated Recurrent Unit* (GCRU) to incorporate the momentum spillover signals among stocks into the temporal representation and infer the ranking results:

$$\begin{aligned} R_t &= \sigma(W_{\mathcal{G}}^R \star (\mathcal{X}^t || H_{t-1} || \mathcal{I}_{nv}) + b_R), \\ C_t &= tanh(W_{\mathcal{G}}^C \star (\mathcal{X}^t || R_t \odot H_{t-1} || \mathcal{I}_{nv}) + b_C), \\ U_t &= \sigma(W_{\mathcal{G}}^U \star (\mathcal{X}^t || H_{t-1} || \mathcal{I}_{nv}) + b_U), \\ H_t &= U_t \odot H_{t-1} + (1 - U_t) \odot C_t, \end{aligned} \tag{5}$$

where H_t denotes the output of at time t, $\mathcal{X}^t$ is the input series at time t, R_t, U_t are reset gate and update gate at time t respectively responsible for catching irrelevant information to forget and the part of past state to move forward. $||$ is concatenation along the feature dimension and $\odot$ represents the element-wise product. b_R, b_C, b_U are model parameters, $\mathcal{I}_{nv}$ is the relative invariant information learned by outer meta-learner, and $\star$ is the two-step graph convolutional operation:

$$W_{\mathcal{G}}^Q \star \mathbf{X}_{in} = \sum_{k=0}^{K=2} ((\mathcal{G}_{in}^t)^k \mathbf{X}_{in} W_k^Q), \tag{6}$$

where $W_k^Q, (k = 0,1,2)$ are trainable parameters, I is an identity matrix, and $\mathbf{X}_{in}$ denotes all of the input sequences. *ADB-TRM* adopts the architecture of encoder and decoder, both of which can be represented as Eq.5. The difference is that the encoder encodes historical information while the decoder decodes future stock sequences based on

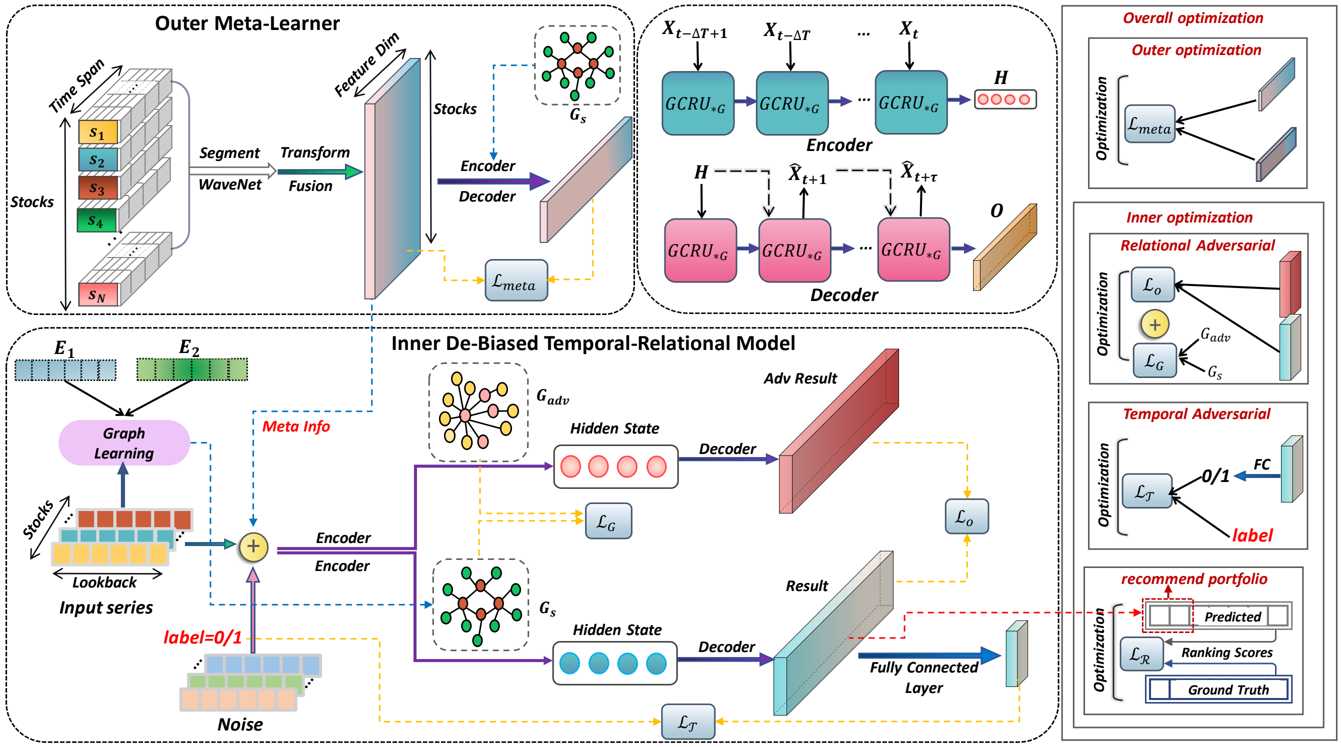

Figure 2: The framework of *ADB-TRM*. *ADB-TRM* is mainly divided into two parts: the outer meta-learner and the inner de-biased *TRM*. The inner part combines GRU with the graph convolutional operation to jointly learn the temporal-relational patterns and applies temporal-relational adversarial training to enhance the model's robustness against temporal-relational biases. The outer meta-learner is responsible for interacting with the inner part and inferring relative invariant information from the global perspective to alleviate the distribution shifts.

historical encoding embeddings. To simplify the notation, in subsequent representations, we use ***Enc(.)*** to represent the encoder, ***Dec(.)*** to represent the decoder of our proposed method, and $\mathcal{P}^t \in \mathbb{R}^N$ as the stock predicted return ratios.

Temporal Adversarial Training

We propose temporal adversarial training (*TAT*) to enhance the model's predictability to counter the temporal bias. Specifically, we incorporate perturbations to mimic the inherent stochastic nature of stocks, training the model to perform effectively even under deliberate perturbations:

$$\mathcal{X}^t = \mathcal{X}^t + a\frac{\mathcal{X}^{t_p}}{\tau\|\mathcal{X}^{t_p}\|}, \quad (7)$$

where τ is a temperature hyper-parameter and $t_p \neq t$. However, when the added noise is highly conspicuous, the process of identifying noise may not effectively enhance the temporal robustness of the model. Besides, considering the inherently low *SNR* in stock market data, introducing minimal noise levels can impede effective learning convergence. Consequently, we introduce a "warm-up" strategy for injecting noise into the input training data. As the number of training batches increases, the τ gradually escalates, resulting in a reduction in the amplitude of noise introduced:

$$\tau = \tau_o * min(S, \delta), \quad (8)$$

where τ_o denotes the initialized temperature, S denotes the training batches, and δ is a hypermeter for restricting minimum amplitude. We decide whether to add noise to the input with equal probability, which means a becomes 0/1 with equal probability. Our training model judges whether noise is added to the $\mathcal{X}^t$ from the final output $\mathcal{P}^t$ to transform the temporal adversarial training into a binary classification task:

$$\mathcal{L}_{\mathcal{T}} = \mathbf{BCE}(softmax(FC_{\mathcal{T}}(\mathcal{P}^t)), label = a), \quad (9)$$

where $FC_{\mathcal{T}}$ is one fully connected layer and **BCE** is the Binary CrossEntropy Loss. Unlike any previous method, we use stock sequences from different time periods as noise. These noise sources used for adversarial training can be obtained simply by sampling at different periods and reflect the evolution dynamics of stock variables themselves. We apply a binary classification task to make the model judge whether the noise is added to the input, which can effectively distinguish different temporal dynamics of stocks to enhance the capability to capture dynamic temporal patterns. Moreover, since noise itself implies the temporal dynamics of stocks, we only need to make the model aware of the presence of noise without noise removal. By minimizing $\mathcal{L}_{\mathcal{T}}$, the model is encouraged to correctly classify both clean and adversarial samples, which helps perceive the inherent data distribution.

Relational Adversarial Training

As mentioned earlier, we adopt relational adversarial training (*RAT*) to boost the generalization capacity and modeling ability of the relational learning part. Specifically, we automatically generate adversarial samples for regularizing the learned dynamic graphs. In *RAT*, we consider the following three adversarial graph samples: **(i)** The degree ratio between nodes in the adversarial graph is high ($\text{degree}_{max}/\text{degree}_{min} > \mathcal{U}_1$). **(ii)** Extreme sparsity ($\text{Sparsity} > \mathcal{U}_2$) in adversarial graph samples. **(iii)** Excessive density ($\text{Sparsity} < \mathcal{U}_3$) in adversarial graph samples. Sparsity refers to the proportion of zero elements in a graph matrix, and $\mathcal{U}_k, (k = 1, 2, 3)$ are three hyperparameters.

To simulate the graph generation task in the real world, we use the Barabasi Albert (BA) graph model [Barabási and Albert, 1999] to generate the first type of adversarial graph. The hyperparameters of the BA graph model are designed to make the degree ratio between nodes in the generated adversarial graph fairly large. We further generate the second and third adversarial graph samples based on the first type to maintain the scale-free characteristics to some extent. Specifically, we generate overly sparse adversarial graph samples by randomly generating a mask matrix to mask the edges in the first generated adversarial graph. To generate overly dense adversarial graph samples, we add a noise matrix to the first generated adversarial graph. The probability setting of adopting the above three adversarial graph samples for adversarial training is 1:1:1. We mark the adversarial graph sample as $\mathcal{G}_{adv}$, and we push the latent representation of $\mathcal{G}_{in}^t$ away from the latent representation of $\mathcal{G}_{adv}$:

$$\mathcal{L}_{\mathcal{G}} = \mathbf{MI}(FC_{\mathcal{G}}(\mathcal{G}_{in}^t), FC_{\mathcal{G}}(\mathcal{G}_{adv})), \tag{10}$$

where $FC_{\mathcal{G}}$ is one fully connected layer and **MI** is the metric used to measure the proximity of mutual semantic information. In addition to obtaining a well-represented graph, we hope that the encoder and decoder of *ADB-TRM* have robustness against erroneous relational interactions. Therefore, we express as follows:

$$\begin{aligned}\mathcal{P}_{adv} &= \textbf{\textit{Dec}}(\textbf{\textit{Enc}}(\mathcal{G}_{adv}; \mathbf{X}_{in}); \mathcal{G}_{adv}),\\ \mathcal{P}^t &= \textbf{\textit{Dec}}(\textbf{\textit{Enc}}(\mathcal{G}_{in}^t; \mathbf{X}_{in}); \mathcal{G}_{in}^t).\end{aligned} \tag{11}$$

Intuitively speaking, we use the same encoder and decoder parameters with different stock relational graphs to generate different ranking results. We increase mutual semantic information between the two ranking results to enhance the robustness of the model to the erroneous relational graph:

$$\mathcal{L}_{\mathcal{O}} = -\mathbf{MI}(\mathcal{P}^t, \mathcal{P}_{adv}). \tag{12}$$

which means the model can make accurate inferences even when it receives erroneous relational interactions. The combination of $\mathcal{L}_{\mathcal{O}}$ and $\mathcal{L}_{\mathcal{G}}$ forms adversarial training because the simple way to increase the **MI** between $\mathcal{P}^t$ and $\mathcal{P}_{adv}$ is to make $\mathcal{G}_{in}^t$ as close as possible to $\mathcal{G}_{adv}$, which counters $\mathcal{L}_{\mathcal{G}}$.

3.4 Outer Meta-Learner

We introduce the outer meta-learner to address the inherent vulnerability of the inner meta component in handling distribution shifts. The role of the outer meta-learner is to effectively capture relative invariant information $\mathcal{I}_{nv}$ pertaining to stocks, in contrast to the inner meta-layer. To operationalize this concept, it is imperative to endow the outer meta-learner with a comprehensive global perspective, enabling it to analyze and understand the temporal dynamics of stocks across diverse time spans. We input the whole training stock sequence $\mathbf{X}_{all}$ into the outer meta-learner to provide a global distribution perspective. From a long-term perspective, the patterns in different time intervals and across time intervals can be captured in parallel by splitting the stock series into segments. Specifically, we set a hyper-parameter period P to segment $\mathbf{X}_{all}$ into $S = \lfloor T_{\text{train}}/P \rfloor$ segments, each containing time-series $\hat{\mathbf{X}}_i, i = 1, 2, ..., S$.. After acquiring these time-series segments, they are concatenated to form a four-dimensional tensor: $\mathcal{Q} = [\hat{\mathbf{X}}_\mathbf{1} \| \hat{\mathbf{X}}_\mathbf{2} \| ... \| \hat{\mathbf{X}}_\mathbf{S}] \in \mathbb{R}^{N \times P \times S \times D}$.

To further emphasize the uniqueness of different time segments and the particular dynamic characteristics of stocks, we apply the WaveNet [Oord *et al.*, 2016] across the dimension S and P to have different time perspectives and obtain two four-dimensional tensors:

$$\begin{aligned}\mathcal{Z}^{(1)} &= \mathbf{WaveNet}(\mathcal{Q}) \in \mathbb{R}^{N \times P \times S \times D},\\ \mathcal{Z}^{(2)} &= \mathbf{WaveNet}(\mathcal{Q}^T) \in \mathbb{R}^{N \times S \times P \times D}.\end{aligned} \tag{13}$$

WaveNet employs a stacked expansion convolution layer, which requires only a few layers to obtain a large receptive field while retaining the input resolution and computing efficiency. We then use three fully connected layers and one 1D convolution as feature extractors to perform dimensional transformation and information extraction on $\mathcal{Z}^{(1)}$ and $\mathcal{Z}^{(2)}$:

$$\mathcal{I}^{(i)} = \sigma(g_1^{(i)}(\delta(g_2^{(i)}(\delta(g_3^{(i)}(\delta(Conv^{(i)}(\mathcal{Z}^{(i)}))))), \tag{14}$$

where $g_j^{(i)}, (j = 1, 2, 3, i = 1, 2)$ are six fully connected layers, $\mathcal{I}^{(i)} \in \mathbb{R}^{N \times D}$. $\mathcal{I}^{(1)}$ and $\mathcal{I}^{(2)}$ harvest temporal information from different time perspectives (S and P). We fuse $\mathcal{I}^{(1)}$ and $\mathcal{I}^{(2)}$ in the form of weighted sum to obtain $\mathcal{I}_{nv}$:

$$\mathcal{I}_{nv} = FC\left[\mathcal{I}^{(1)} || \mathcal{I}^{(2)}\right] \in \mathbb{R}^{N \times D}. \tag{15}$$

$\mathcal{I}_{nv}$ contains the semantic information of the stock volatility patterns from the global perspective. To further train the outer meta-learner and discover robust time-invariant information, we leverage the interaction between the outer layer's global perspective and the inner layer's local view, akin to a latent projection. Specifically, we input $\mathcal{I}_{nv}$ as the daily sequence information, i.e., $\mathcal{X}^t = \mathcal{I}_{nv}$ and replace the original $\mathcal{I}_{nv}$ with the same size zero tensor into the inner meta part, and obtain output $\mathcal{I}_{evo}$. Intuitively, $\mathcal{I}_{evo}$ is the evolving result of $\mathcal{I}_{nv}$ over a short time horizon. To ensure that $\mathcal{I}_{nv}$ is relatively invariant stock embeddings, we aim to maximize the mutual semantic information between $\mathcal{I}_{evo}$ and $\mathcal{I}_{nv}$:

$$\mathcal{L}_{meta} = -\mathbf{MI}(\mathcal{I}_{nv}, \mathcal{I}_{evo}). \tag{16}$$

We feed $\mathcal{I}_{nv}$ as the time-invariant info into the Temporal-Relational Fusion module to mitigate distribution shifts.

4 Experiments

4.1 Experimental Setup

Dataset

We conduct an extensive analysis of *ADB-TRM* on three stock exchange datasets. Detailed statistics are presented in Table 2. The first dataset [Feng *et al.*, 2019b] consists of 1,026 stock

		Methods	NASDAQ		NYSE		TSE	
			SR	IRR	SR	IRR	SR	IRR
CLF	ARIMA [Wang and Leu, 1996]	Auto Regressive Integrated Moving Average to fit non-stationary stock price data	0.55	0.10	0.33	0.10	0.47	0.13
	ALSTM [Feng *et al.*, 2019a]	Adversarial LSTM simulates stock stochasticity during training	0.97	0.23	0.81	0.14	1.10	0.43
	HATS [Kim *et al.*, 2019]	Hierarchical graph attention model to aggregate information from multi-graph	0.80	0.15	0.73	0.12	0.96	0.31
	HMG-TF [Ding *et al.*, 2020]	Enhanced Transformer for learning multi-scale features of Finance data	0.83	0.19	0.75	0.13	1.05	0.33
RL	DQN [Carta *et al.*, 2021]	Annotation-free ensembled RL method for maximizing return function	0.93	0.20	0.72	0.12	1.08	0.31
	iRDPG [Liu *et al.*, 2020]	An enhanced model combines deep reinforcement learning and imitation learning	1.32	0.28	0.85	0.18	1.10	0.55
	RAT [Xu *et al.*, 2021]	Relation-aware Transformer for portfolio selection with reinforcement learning	1.37	0.40	1.03	0.22	1.20	0.64
REG	SFM [Zhang *et al.*, 2017]	State Frequency Memory recurrent network for modeling multi-level time frequency	0.16	0.09	0.19	0.11	0.08	0.07
	MTGNN [Wu *et al.*, 2020]	Adaptive GNN framework with dilated inception module for time-series forecasting	0.82	0.29	0.94	0.17	1.01	0.33
	THGNN [Xiang *et al.*, 2022]	A temporal and heterogeneous GNN based on learning dynamic relations	0.88	0.31	0.78	0.13	1.15	0.41
RAN	RSR-E [Feng *et al.*, 2019b]	Temporal GCN based on similarity measure as relation weight	1.12	0.26	0.88	0.20	1.07	0.50
	RSR-I [Feng *et al.*, 2019b]	Temporal GCN based on neural network to calculate relation weight	1.34	0.39	0.95	0.21	1.08	0.53
	STHAN-SR [Sawhney *et al.*, 2021a]	A temporal-relational hypergraph attentive architecture for stock selection	1.42	0.44	1.12	0.33	1.19	0.62
	HyperStockGAT [Sawhney *et al.*, 2021b]	Hyperbolic graph attention network on the Riemannian Manifolds for stock selection	1.40	0.44	1.10	0.25	1.20	0.75
	ALSP-TF [Wang *et al.*, 2022c]	A temporal-relation adaptive transformer architecture for stock selection	1.55	0.53	1.24	0.41	1.27	0.71
	RT-GCN [Zheng *et al.*, 2023]	A relational-temporal GCN based on three relation-aware strategies	1.49	0.48	1.22	0.37	1.29	0.78
	ADB-TRM (Ours)	Automatic de-biased temporal-relational model for stock selection	**1.66**	**0.66**	**1.42**	**0.58**	**1.38**	**0.93**
	Improve	Improvements over state-of-the-art	7.10%	24.53%	14.52%	41.46%	6.98%	19.23%

Table 1: Comparison of profitability with Classification (CLF), Reinforcement Learning (RL), Regression (REG), and Ranking (RAN) baselines. The improvement is statistically significant ($p < 0.01$) under Wilcoxon's signed rank test.

Datasets	Stocks	Train Days	Valid Days	Test Days
NASDAQ	1026	01/13-12/15 (756)	01/16-12/16 (252)	01/17-12/17 (237)
NYSE	1737	01/13-12/15 (756)	01/16-12/16 (252)	01/17-12/17 (237)
TSE	95	11/15-08/18 (693)	08/18-07/19 (231)	07/19-08/20 (235)

Table 2: Dataset statistics.

shares from the relatively volatile US S&P 500 and NASDAQ Composite Indexes. The second dataset [Feng *et al.*, 2019b] encompasses 1,737 stocks listed on the NYSE, which is renowned as the world's largest stock exchange in terms of market capitalization of listed companies, and is comparatively more stable with NASDAQ. The third dataset [Li *et al.*, 2021] is centered around the widely recognized TOPIX-100 Index, which includes 95 stocks with the highest market capitalization on the Tokyo stock exchange.

Framework Training Strategy

We follow the bilevel optimization format as in [Hospedales *et al.*, 2021], where the inner focuses on addressing temporal-relational bias and the outer part tackles distribution shifts:

$$\begin{aligned} &\hat{\omega} = \arg\min_{\omega} \mathcal{L}_{meta}(\mathcal{D}^{train}_{source}; \hat{\theta}^{(i)}(\omega), \omega), \\ &s.t. \quad \hat{\theta}^{(i)}(\omega) = \arg\min_{\theta} \mathcal{L}_{task}((\mathcal{D}^{train(i)}_{source}; \theta, \omega)), \\ &\mathcal{L}_{task} = \mathcal{L}_{\mathcal{P}} + \mathcal{L}_{\mathcal{T}} + \beta\mathcal{L}_{\mathcal{G}} + \gamma\mathcal{L}_{\mathcal{O}}, \end{aligned} \tag{17}$$

where the data source $\mathcal{D}_{source}$ is the price series of all stocks. We follow the normal training process of meta-learning and first train the parameters of the inner part, then fix the inner parameters and train the parameters of the outer meta-learner. The overall training procedure is end-to-end. It's important to highlight that the **MI** metric plays a significant role in our overall optimization strategy. As such, we conduct a thorough discussion on the selection of **MI**, with detailed experiments provided in Appendix A[1]. In Table 1, we utilize Wasserstein distance [Panaretos and Zemel, 2019] as the **MI** metric.

[1]Appendix is provided in https://oncecwj.github.io/ADB-TRM-Appendix/Appendix.pdf

Implementation Details

Our model is implemented with PyTorch. We collect stock data from Wind-Financial Terminal[2], including normalized *opening-high-low-closing* prices and *trading volume* (OHLCV). For a fair comparison, we follow [Sawhney *et al.*, 2021b] and generate samples by moving a 16-day lookback window along trading days. We use grid search to find optimal hyperparameters. For the proposed framework, the period P and dimension F are searched within $\{10, 20, 30, 40, 50\}$ and finally set to 20 and 10, respectively. In temporal-relational fusion, we set the RNN hidden units H_u to 96. The dilation depth D_{ep} and stacked layers L_s in WaveNet are both set to 2. The hyperparameters $\lambda \in [1, 10]$, $\beta, \gamma \in [0.5, 5]$, and $b \in \{1e^2, 5e^2, 1e^3, 1.5e^3, 2e^3\}$ are finally set to 4, 1.2, 1.2, and $1e^3$, respectively. The initial temperature $\tau_o \in \{10^{-2}, 10^{-1}, 1, 10\}$ and $\delta \in \{10, 10^2, 10^3, 10^4\}$ are set to 1 and 10^3. The hyperparameters $\mathcal{U}_1 \in \{20, 25, 30, 35, 40\}, \mathcal{U}_2 \in \{83\%, 87\%, 91\%, 95\%, 99\%\}$, and $\mathcal{U}_3 \in \{63\%, 67\%, 71\%, 75\%, 79\%\}$ are finally set to 25, 95%, and 67%, respectively. We tune the model and ablation variants on one Nvidia GeForce RTX 3090 GPU by Adam optimizer [Kingma and Ba, 2014] for 50 epochs, the learning rate is set to 0.001, and the batch size is set to 20.

Metrics

Follow prior research [Sawhney *et al.*, 2021b; Feng *et al.*, 2019b], we employ a daily buy-hold-sell trading strategy to evaluate the profitability of *ADB-TRM* and use the Sharpe Ratio (SR) and the cumulative investment return ratio (IRR) as metrics. Specifically, the trader purchases k stocks with the highest anticipated revenues after the market closes on day t and sells these shares at the close of the following day's market session. Formally, $\mathrm{IRR}^t = \sum_{i \in \hat{\mathcal{S}}^t} \frac{c_i^{t+1} - c_i^t}{c_i^t}$, where $\hat{\mathcal{S}}^t$ is the stocks in the portfolio on day t. The Sharpe Ratio (SR) is a risk-adjusted return metric that quantifies the additional earnings an investor receives per unit of increased risk, given by $\mathrm{SR} = \frac{E[R_p] - R_f}{std[R_p]}$. We also assess the model's ranking abil-

[2]https://www.wind.com.cn/en/wft.html

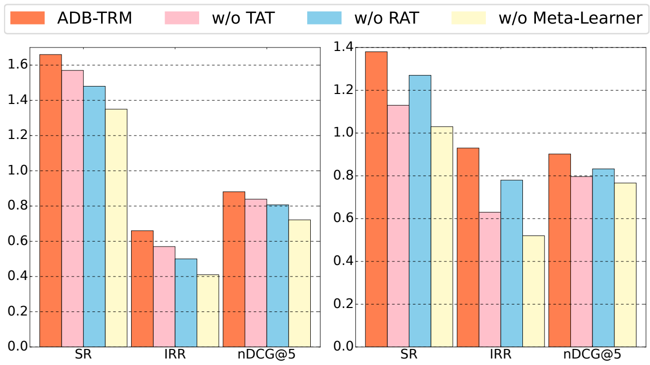

Figure 3: Ablation study over different components (*outer Meta-Learner, temporal adversarial training, and relational adversarial training*) on NASDAQ (left) and TSE (right).

ity using the widely adopted metric nDCG@k. We report the mean results obtained from ten independent runs with $k = 5$.

4.2 Overall Performance

Following the previous work [Sawhney *et al.*, 2021a; Wang *et al.*, 2022c], we consider four categories of baselines for comparison. The results are shown in Table 1, from which we have several observations: **(1)** In general, RL and ranking approaches (e.g., *iRDPG*, *RSR*) perform better in investment returns than conventional price classification and regression methods (e.g., *HATS*, *SFM*), which justifies the effectiveness of *learning-to-rank* optimization and temporal-relational modeling toward stock selection. **(2)** Through the strategic enhancement of *TRM* to counter temporal-relational bias and adapt to distribution shifts, our novel *ADB-TRM* consistently achieves superior results across all datasets. In particular, it demonstrates an average relative performance improvement of 9.53% and 28.41% in terms of risk-adjusted returns and cumulative profits when compared to the leading baseline models. Empowering the model with enhanced adjustment and error-correction capabilities in the context of a volatile and evolving stock market effectively bolsters its generalization and results in improved returns. **(3)** In contrast to the prior state-of-the-art temporal-relational models such as *RT-GCN* and *ALSP-TF*, which exhibit consistent computational demands between training and testing phases, our *ADB-TRM* primarily assigns the computational load to the adversarial training stage rather than during inference, enhancing model performance without additional model complexity. The overall model structural design of *ADB-TRM* is lightweight.

Ablation Study

In our ablation experiments, we investigate the impacts of *TAT*, *RAT*, and the outer meta-learner on the overall performance. The results are depicted in Figure 3, with similar trends observed on NASDAQ and TSE, and comparable findings anticipated for the NYSE. As illustrated, these different components collectively contribute to enhanced performance. The ablation study demonstrates incremental improvements stemming from *TAT*, *RAT*, and the outer meta-learner. The primary benefits arise from the outer meta-learner, which effectively mitigates the substantial distributional shifts en-

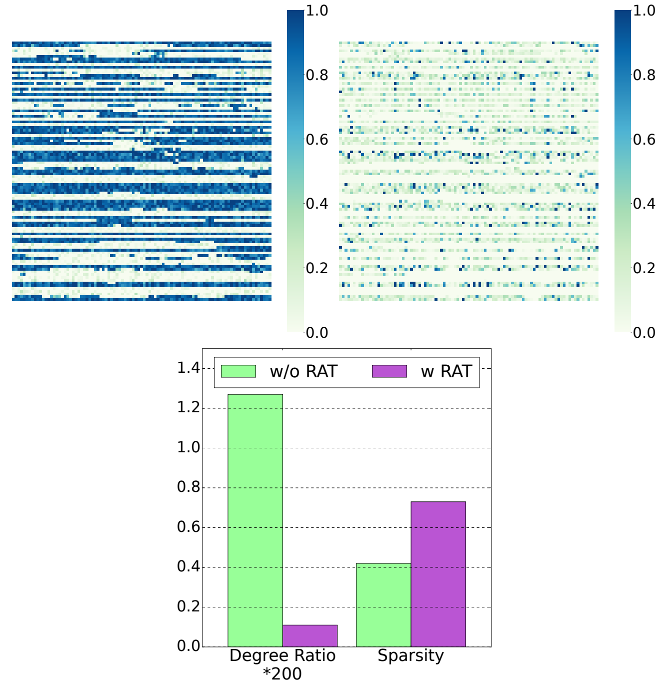

Figure 4: Visualization of learned relational graph with *RAT* (upper right) and without *RAT* (upper left). The lower figure compares the statistical properties of the learned graph with and without *RAT*.

countered in the stock market and enhances the model's generalization capabilities. Furthermore, it's noteworthy that the impact of *RAT* on the NASDAQ is significantly more pronounced than on the TSE, a trend similarly observed on the NYSE. This observation suggests that larger stock pools may exhibit a heightened susceptibility to relational bias.

Stock Graph Visualization

Figure 4 illustrates the role of *RAT* on the TSE dataset. In essence, the design of *RAT* primarily functions as a regularization mechanism, facilitating the sparsity of the learned graph and efficacious redistribution of momentum spillovers among stocks. The incorporation of *RAT* does not notably alter the intercorrelations between stocks but, instead, restructures the relational characteristics and eliminates potentially task-irrelevant connections, resulting in a more concise relational graph. Besides, *RAT* effectively uplifts the sparsity of the learned graph to a normal range and mitigates the degree polarization issue in the learned graph. Based on the results of the ablation experiment, these reformulations also contribute to an overall improvement in the model's profitability.

4.3 In-depth Analysis

Adversarial Attacks

Moreover, our findings reveal that the *ADB-TRM* is capable of effectively countering the impact of malicious activities (like stock price manipulation) within the temporal and relational domains on the overall revenue. In contrast, existing *TRM* are vulnerable to these attacks, leading to a swift decrease in overall profit. Details are provided in Appendix B.

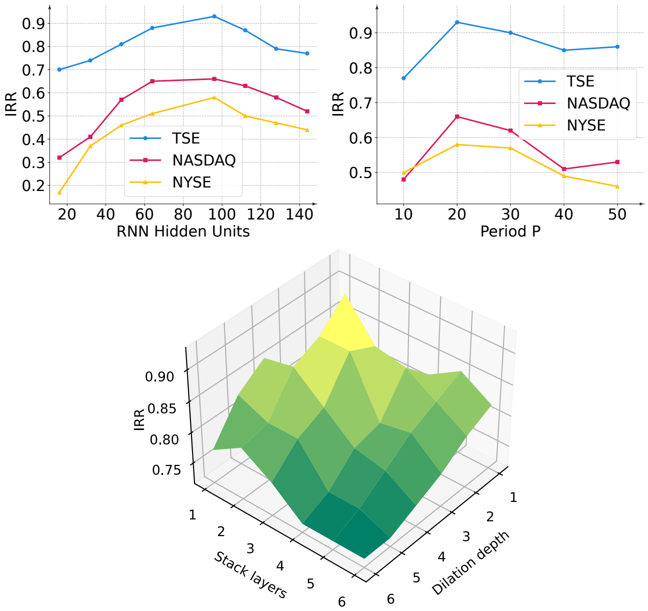

Figure 5: Influence of H_u (upper left), P (upper right) on IRR. The lower picture shows the influence of L_s and D_{ep} on IRR on the TSE dataset.

4.4 Hyper-parameter Sensitivity

We focus on the number of RNN hidden units H_u, the segmentation period P, the stack layers L_s, and the dilation depth D_{ep}. The experimental results on IRR are shown in Figure 5. Due to space limitations, we demonstrate the impact of L_s and D_{ep} on the IRR metric on the TSE dataset, similar regularities can be observed on other datasets. Note that when studying the effect of one hyperparameter, others are kept as the default values. Specifically, the *ADB-TRM* model demonstrates a significant sensitivity to the hyperparameter H_u. If H_u is set too small, the model struggles to capture effective temporal patterns because the recurrent memory module has insufficient training parameters. Conversely, setting H_u too high can lead to issues in the volatile and low *SNR* stock market environment, resulting in overfitting. The effects of hyperparameters L_s and D_{ep} on IRR exhibit a similar trend, with larger values rendering the model more vulnerable to noise within the stock market, albeit with diminishing returns. The influence of period P on the IRR exhibits a diminished effect. Specifically, when P assumes the value of $20/30$, IRR attains its peak, a phenomenon that is likely attributable to the extraction of pertinent monthly features.

5 Conclusion

This paper pioneers alleviating the intrinsic temporal-relational bias and distribution shifts in the stock market by employing the well-designed meta-learning framework and carefully calibrated temporal-relational adversarial training techniques, thereby enhancing stock investment returns. Experiments are conducted on three real-world datasets from the US and Japanese Stock Exchange markets. The experiments cover four major categories and sixteen compared methods. Results show that our model outperforms other non-rank and rank-based state-of-the-art stock investment methods. Detailed experiments also substantiate the efficacy of our approach in mitigating the biases within the stock market and fortifying the model's robustness.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No.62176005) and the Art Project of the National Social Science Fund of China (2022CC02195).

References

[Barabási and Albert, 1999] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[Carta *et al.*, 2021] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting. *Expert systems with applications*, 164:113820, 2021.

[Cheng and Li, 2021] Rui Cheng and Qing Li. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, pages 55–62, 2021.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Ding *et al.*, 2020] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. Hierarchical multi-scale gaussian transformer for stock movement prediction. In *IJCAI*, pages 4640–4646, 2020.

[Feng *et al.*, 2019a] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. Enhancing stock movement prediction with adversarial training. *IJCAI*, 2019.

[Feng *et al.*, 2019b] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–30, 2019.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Granger, 1969] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[He *et al.*, 2022] Yanshen He, Qiutong Li, Feng Wu, and Jianliang Gao. Static-dynamic graph neural network for stock recommendation. In *Proceedings of the 34th International Conference on Scientific and Statistical Database Management*, pages 1–4, 2022.

[Hospedales *et al.*, 2021] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2021.

[Hsu *et al.*, 2021] Yi-Ling Hsu, Yu-Che Tsai, and Cheng-Te Li. Fingat: Financial graph attention networks for recommending top-k k profitable stocks. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):469–481, 2021.

[Huynh *et al.*, 2023] Thanh Trung Huynh, Minh Hieu Nguyen, Thanh Tam Nguyen, Phi Le Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 850–858, 2023.

[Khuwaja *et al.*, 2021] Parus Khuwaja, Sunder Ali Khowaja, and Kapal Dev. Adversarial learning networks for fintech applications using heterogeneous data sources. *IEEE Internet of Things Journal*, 2021.

[Kim *et al.*, 2019] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999*, 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[Li *et al.*, 2021] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the twenty-ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4541–4547, 2021.

[Li *et al.*, 2022] Yang Li, Hong-Ning Dai, and Zibin Zheng. Selective transfer learning with adversarial training for stock movement prediction. *Connection Science*, 34(1):492–510, 2022.

[Liang *et al.*, 2018] Zhipeng Liang, Hao Chen, Junhao Zhu, Kangkang Jiang, and Yanran Li. Adversarial deep reinforcement learning in portfolio management. *arXiv preprint arXiv:1808.09940*, 2018.

[Lin *et al.*, 2021] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. 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*, pages 1017–1026, 2021.

[Liu *et al.*, 2020] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2128–2135, 2020.

[Næs *et al.*, 2011] Randi Næs, Johannes A Skjeltorp, and Bernt Arne Ødegaard. Stock market liquidity and the business cycle. *The Journal of Finance*, 66(1):139–176, 2011.

[Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[Panaretos and Zemel, 2019] Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6:405–431, 2019.

[Rezaei *et al.*, 2021] Hadi Rezaei, Hamidreza Faaljou, and Gholamreza Mansourfar. Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, 169:114332, 2021.

[Sawhney *et al.*, 2021a] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 497–504, 2021.

[Sawhney *et al.*, 2021b] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proceedings of the Web Conference 2021*, pages 11–22, 2021.

[Wang and Leu, 1996] Jung-Hua Wang and Jia-Yann Leu. Stock market trend prediction using arima-based neural networks. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 4, pages 2160–2165. IEEE, 1996.

[Wang *et al.*, 2021] Heyuan Wang, Shun Li, Tengjiao Wang, and Jiayi Zheng. Hierarchical adaptive temporal-relational modeling for stock trend prediction. In *IJCAI*, pages 3691–3698, 2021.

[Wang *et al.*, 2022a] Heyuan Wang, Tengjiao Wang, Shun Li, and Shijie Guan. Hatr-i: Hierarchical adaptive temporal relational interaction for stock trend prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Wang *et al.*, 2022b] Heyuan Wang, Tengjiao Wang, Shun Li, Shijie Guan, Jiayi Zheng, and Wei Chen. Heterogeneous interactive snapshot network for review-enhanced stock profiling and recommendation. In *IJCAI*, pages 3962–3969, 2022.

[Wang *et al.*, 2022c] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. Adaptive long-short pattern transformer for stock investment selection. In *Proc. 31st Int. Joint Conf. Artif. Intell*, pages 3970–3977, 2022.

[Wang *et al.*, 2022d] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & data mining*, pages 753–763, 2020.

[Xiang *et al.*, 2022] Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang. Temporal and heterogeneous graph neural network for financial time series prediction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3584–3593, 2022.

[Xu *et al.*, 2021] Ke Xu, Yifan Zhang, Deheng Ye, Peilin Zhao, and Mingkui Tan. Relation-aware transformer for portfolio policy learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4647–4653, 2021.

[Ye *et al.*, 2022] Junchen Ye, Zihan Liu, Bowen Du, Leilei Sun, Weimiao Li, Yanjie Fu, and Hui Xiong. Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2296–2306, 2022.

[Zhang *et al.*, 2017] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2141–2149, 2017.

[Zhao *et al.*, 2022] Yu Zhao, Huaming Du, Ying Liu, Shaopeng Wei, Xingyan Chen, Fuzhen Zhuang, Qing Li, and Gang Kou. Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[Zheng *et al.*, 2023] Zetao Zheng, Jie Shao, Jia Zhu, and Heng Tao Shen. Relational temporal graph convolutional networks for ranking-based stock prediction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 123–136. IEEE, 2023.