

Learning Universal Multi-level Market Irrationality Factors to Improve Stock Return Forecasting

Chen Yang

School of Computer Science and Engineering
Beihang University
Beijing, China
stx221166@buaa.edu.cn

Xiaohan Jiang

School of Computer Science and Engineering
Beihang University
Beijing, China
jxh199@buaa.edu.cn

Jingyuan Wang*

School of Computer Science and Engineering
School of Economics and Management
Beihang University
Beijing, China
jywang@buaa.edu.cn

Junjie Wu

MIIT Key Laboratory of Data Intelligence and
Management, School of Economics and Management
Beihang University
Beijing, China
wujj@buaa.edu.cn

Abstract

Recent years have witnessed the perfect encounter of deep learning and quantitative trading has achieved great success in stock investment. Numerous deep learning-based models have been developed for forecasting stock returns, leveraging the powerful representation capabilities of neural networks to identify patterns and factors influencing stock prices. These models can effectively capture general patterns in the market, such as stock price trends, volume-price relationships, and time variations. However, the impact of special irrationality factors – such as market sentiment, speculative behavior, market manipulation, and psychological biases – has not been fully considered in existing deep stock forecasting models due to their relative abstraction as well as lack of explicit labels and data description. To fill this gap, we propose *UMI*, a *Universal multi-level Market Irrationality* factor model to enhance stock return forecasting. The *UMI* model learns factors that can reflect irrational behaviors in market from both individual stock and overall market levels. For the stock-level, *UMI* construct an estimated rational price for each stock, which is cointegrated with the stock's actual price. The discrepancy between the actual and the rational prices serves as a factor to indicate stock-level irrational events. Additionally, we define market-level irrational behaviors as anomalous synchronous fluctuations of stocks within a market. Using two self-supervised representation learning tasks, *i.e.*, sub-market comparative learning and market synchronism prediction, the *UMI* model incorporates market-level irrationalities into a market representation vector, which is then used as the market-level irrationality factor. We also

developed a forecasting model that captures both temporal and relational dependencies among stocks, accommodating the *UMI* factors. Extensive experiments on U.S. and Chinese stock markets with competitive baselines demonstrate our model's effectiveness and the universality of our factors in improving various forecasting models. We provide our code at <https://github.com/lcIII/UMI>.

CCS Concepts

- Applied computing → Economics;
- Information systems → Data mining;
- Computing methodologies → Neural networks.

Keywords

Stock Return Forecasting, Market Irrationality, Deep Learning, Self-supervised Learning

ACM Reference Format:

Chen Yang, Jingyuan Wang, Xiaohan Jiang, and Junjie Wu. 2025. Learning Universal Multi-level Market Irrationality Factors to Improve Stock Return Forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD '25), August 3–7, 2025, Toronto, ON, Canada*. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3690624.3709328>

1 Introduction

Quantitative trading (QT) strategies have seen extensive adoption by financial institutions and investment funds, leading to significant successes in the securities investment markets worldwide. According to Morgan Stanley, quant-based investing has contributed to \$1.5 trillion in retail and hedge funds since 2017, and has been particularly favored by the macroeconomic conditions from 2023 to 2024¹. The stock return forecasting is a crucial technology in QT strategies. It utilizes historical stock features to predict future price trends. In the literature, classic QT strategies adopt specific financial phenomenon to forecast stock returns. For instance, the *momentum* strategy uses the market's momentum to predict stock price trends [17], while the *mean reversion* strategy relies on the principle that asset prices will eventually revert to their average [36].

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '25, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-1245-6/25/08

<https://doi.org/10.1145/3690624.3709328>

¹<https://www.morganstanley.com/ideas/quantitative-investing-outperformance-2023>

Recent years have witnessed the revival of deep learning methods [10, 12, 18–22, 33, 37, 38, 46–49, 60], which shed light on developing more effective stock forecasting models. Deep learning models have powerful representation learning capabilities, making them particularly well-suited for handling intricate data relation in real financial markets. Early methods consider stock price or return series as ordinary time series and employ regular sequential deep learning models [1] such as RNN [26], LSTM [50] and Transformer [52]. Subsequently, specially designed neural networks are proposed to handle the characteristics of financial data, such as distribution shifts [61], stochasticity [24], and multiple trading patterns [30]. In addition, many studies have explored incorporating information beyond prices into model inputs to achieve excess returns, such as stock comments [27], experience of trading experts [42], etc.

The core of deep learning-based models lies in using neural networks to extract predictive correlation patterns for stock prices from market behavior data. For existing deep learning stock forecasting models, the correlations they learn typically reflect the most general patterns in the market, such as stock price trend [23], volume-price relationships [25], time variation [9]. However, beyond these general patterns, stock prices are also influenced by many special irrationality factors, such as market sentiment (emotional fluctuations of investors, i.e., panic selling or greedy buying) [43], market manipulation (individual investors or institutions manipulating market prices through improper means) [16], and psychological biases (investor psychological biases such as overconfidence, anchoring effects, loss aversion, etc.) [41]. Compared with the general patterns that can be directly described by explicit labels and side information, the irrationality factors are more abstract and thus challenging to explicitly incorporate into the model. Most existing deep learning forecasting models lack consideration for these irrationality factors, resulting in suboptimal performance.

To bridge this gap, in this paper, we propose a *Universal multi-level Market Irrationality* factor mining model, abbreviated as *UMI* (You-Me) model. In the *UMI* model, we categorize irrational behavior in markets into two types: stock-level irrational events and market-level irrational events. A stock-level irrational event is defined as a temporary deviation of a stock's price from its fundamental value (rational price). To exploit these events, we propose a cointegration attention mechanism with stationary regularization to construct an estimated rational price for each stock. This rational price is cointegrated with the actual price (the difference between the actual and rational prices forms a stationary series) but is more stable because it is derived by combining the prices of multiple stocks. The discrepancy between the actual price and the estimated rational price is used as a factor to indicate stock-level irrationalities. On the other hand, we define a market-level irrational event as an anomalous synchronous fluctuation of all stocks in a market, as such broad synchrony in stock fluctuations is atypical in an efficient and rational market. To exploit this type of irrationality, we extract dynamic representations for each stock and combine them into an overall market representation. We then use two self-supervised tasks, i.e., sub-market comparative learning and market synchronism prediction, to incorporate market-level irrationalities into the market representation as market-level irrationality factors. Finally, both stock- and market-level irrationality

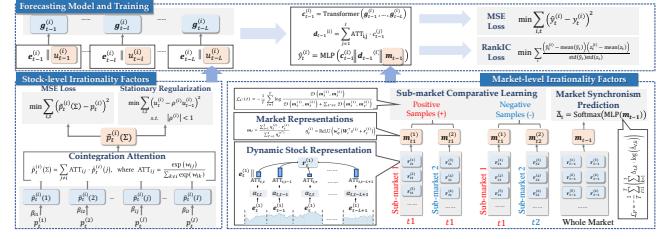


Figure 1: The overview framework of the *UMI* model.

factors are used as inputs of a deep learning model for stock return forecasting. To fully leverage the irrationality factor, we propose a Transformer and graph-attention mixed model with a RankIC loss as the forecasting model. This model captures correlations in historical stock features and inter-stock relationships, rendering it highly suitable for investment strategies.

We have conducted extensive experiments on the stock markets of the U.S. and China, the two largest economies. The results demonstrate the effectiveness of our proposed *UMI* model in terms of both forecasting accuracy and investment benefits. Notably, the irrationality factors exhibit strong universality in improving various forecasting models. To the best of our knowledge, our work is the first to incorporate market irrational behaviors into deep learning-based stock forecasting.

2 Preliminaries

Basic Notations. We define a *stock market* as a set of tradable stocks, denoted as $S = \{s_1, \dots, s_i, \dots, s_T\}$, where s_i denotes the i -th stock. We divide the tradable time of a market as a sequential *periods*, denoted as $\{1, \dots, t, \dots, T\}$. Usually, a period is a trading day. For each period, we describe a stock using its features.

DEFINITION 1 (STOCK FEATURE). *The feature of a stock is a set of numerical attributes that describe the state of a stock, such as the opening price, closing price, trading volume, etc. The feature of the stock s_i in the t -th period is defined as a vector $e_t^{(i)} \in \mathbb{R}^{D_e}$, where D_e is the dimensionality. The feature of s_i during T successive periods is defined as the vector sequence $E^{(i)} = (e_1^{(i)}, \dots, e_t^{(i)}, \dots, e_T^{(i)})$.*

DEFINITION 2 (STOCK PRICE). *Given T successive periods, the price of the stock s_i is defined as the sequence $p^{(i)} = (p_1^{(i)}, \dots, p_t^{(i)}, \dots, p_T^{(i)})$, where $p_t^{(i)}$ is the price of s_i at the end of the period t . If t is a trading day, $p_t^{(i)}$ is the closing price of that day.*

DEFINITION 3 (STOCK RETURN). *The return of the stock s_i in the period t is defined as*

$$y_t^{(i)} = \left(p_t^{(i)} - p_{t-1}^{(i)} \right) / p_{t-1}^{(i)} \times 100\%. \quad (1)$$

Problem Formulation. The *stock return forecasting* problem is using the historical features and prices of stocks in a market to predict the future price of a stock. Our model adopts a two-step approach to implement a stock return forecasting model. Given the historical features $E_{t-1}^{(i)} = (e_1^{(i)}, \dots, e_{t-1}^{(i)})$ for the stock s_i in the period $t-1$, we first adopt a factor learning function to extract indicative factors that can reflect the market irrational behaviors as

$$\Phi_{t-1}^{(i)} = \mathcal{G}(E_{t-1}^{(i)}), \quad (2)$$

where $\mathcal{E}_{t-1} = \{\mathbf{E}_{t-1}^{(i)}\}_{i=1}^I$. $\Phi_t^{(i)}$ are the factors to indicate the influence of market irrationality, namely *irrationality factor Representations*. Next, we employ a stock return forecasting function that takes both $\Phi_{t-1}^{(i)}$ and the historical stock features \mathcal{E}_{t-1} as inputs to predict the future stock return as

$$\hat{y}_t^{(i)} = \mathcal{F}\left(\Phi_{t-1}^{(i)}, \mathcal{E}_{t-1}\right). \quad (3)$$

In the following sections, we provide detailed implementations of the irrationality factor extraction function $\mathcal{G}(\cdot)$ (in Sec. 3 and Sec. 4) and the stock return forecasting function $\mathcal{F}(\cdot)$ (in Sec. 5) for the *UMI* model. The overall framework is illustrated in Fig. 1.

3 Stock-level Irrationality Factors

In this section, we propose the method to learn factors that can indicate the market irrational behaviors for individual stocks.

3.1 Irrationality of Individual Stocks

Stock-level Irrational Events. The stock-level irrational events refer the irrational phenomenon that only creates an investment opportunity for a certain stock. Specifically, we define a stock-level irrational event as the following.

DEFINITION 4 (STOCK-LEVEL IRRATIONAL EVENTS). Given the real price $p_t^{(i)}$ for the stock s_i , we assume there is a rational price $\tilde{p}_t^{(i)}$, if the price divergence between the real price and the rational price is larger than a threshold H_s , i.e.,

$$\left| p_t^{(i)} - \tilde{p}_t^{(i)} \right| > H_s, \quad (4)$$

there is a stock-level irrational event at the stock s_i in period t .

A temporary deviation of the actual price from the rational price, i.e., an irrational event, creates an investment opportunity. We can profit by buying stocks priced below the rational price or selling stocks priced above it, expecting reversion to the rational price.

Cointegration Relations. The challenge in exploiting stock-level irrational events lies in determining the rational price $\tilde{p}_t^{(i)}$. Obviously, $\tilde{p}_t^{(i)}$ is a hidden factor and cannot be directly observed from the historical stock features and prices. Most of the time, we cannot even explicitly calculate rational prices. In classic quantitative trading strategies, one method of addressing this challenge is using the price of a stock closely related to the target stock as a “proxy” for the rational price of the target stock, such prices of the same company’s stocks in different markets. The price of a proxy stock is expected to have a *cointegration relations* with the target stock. The definition of cointegration relationships is given in the following [7].

DEFINITION 5 (COINTEGRATION). Given price series of two stocks, denoted as $\mathbf{p}^{(i)} = (p_1^{(i)}, \dots, p_T^{(i)})$ and $\mathbf{p}^{(j)} = (p_1^{(j)}, \dots, p_T^{(j)})$, we define the combination of the two series as a new time series, i.e.,

$$u_t = p_t^{(j)} - \beta p_t^{(i)}, \quad (5)$$

where β is a combination coefficient. If $\mathbf{u} = (u_1, \dots, u_t, \dots, u_T)$ is a stationary time series, we say $\mathbf{p}^{(i)}$ and $\mathbf{p}^{(j)}$ are cointegrated.

According to Definition 5, the stationary time series u_t is expected to exhibit fluctuations around its mean value. For a pair of stocks with cointegration relation, if the corresponding u_t deviates

significantly from its mean, it indicates the occurrence of an stock-level irrational event. This presents an investment opportunity for us to capitalize on by buying undervalued stocks and selling overvalued ones, i.e., *pair trading*. When u_t reverts to its mean, i.e., the undervalued or overvalued stock return its rational prices, the pair trading strategy yields returns.

Cointegration Test. The cointegration relations between two stocks can be identified using the *Dickey–Fuller (DF) Test*. The DF test assumes that the series \mathbf{u} follows an autoregressive process as

$$u_t = \rho u_{t-1} + \epsilon_t, \quad (6)$$

where ρ is a regression parameter and ϵ_t is a random error. The value of ρ determines whether the series \mathbf{u} is stationary. Specifically, we can expand the iterative process of Eq. (6) as

$$\begin{aligned} u_t &= \rho u_{t-1} + \epsilon_t = \rho(\rho u_{t-2} + \epsilon_{t-1}) + \epsilon_t \\ &= \rho(\rho(u_{t-3} + \epsilon_{t-2}) + \epsilon_{t-1}) + \epsilon_t = \dots \\ &= \rho^{t-1} u_1 + \rho^{t-2} \epsilon_1 + \rho^{t-3} \epsilon_2 + \dots + \epsilon_t. \end{aligned} \quad (7)$$

As we can see, if $|\rho| >= 1$, the influence of error $\{\epsilon_t\}_{t=1}^n$ to u_t cannot decrease with increasing t , leading to non-stationarity. To the contrary, if $|\rho| < 1$, the impact of past errors $\{\epsilon_q\}_{q=1}^n$ on u_t diminishes as t increases. This leads to stationarity. The DF test verifies the stationarity of u_t by testing whether $|\rho|$ is less than 1.

3.2 Stock-level Irrationality Factor Learning

Although utilizing cointegration to construct a pair trading strategy can capitalize on stock-level market irrationality, its effectiveness remains limited. Natural cointegration relations in the stock market are very sparse. Traditional pair trading strategies require screening all potential stock pairs to identify those that are cointegrated, a method that proves to be highly inefficient. Furthermore, the strategy can only invest in a limited number of cointegrated stocks, resulting in higher investment risk and lower strategy stability. To overcome this limitation, we propose an attention-based method to construct a cointegrated series for each stock as an estimated rational price. This estimated rational price is then used as an indicative factor to enhance the performance of stock return forecasting.

Cointegration Attention. Given the price series $p_t^{(i)}$ of stock i , we use the price $p_t^{(j)}$ of stock $j \neq i$ to generate a candidate cointegrated price at period t as

$$\tilde{p}_t^{(i)}(j) = \beta_{ij} \cdot p_t^{(j)}, \quad (8)$$

where β_{ij} is a learnable parameter. Rather than identifying cointegration relationships between stock i and the candidate cointegrated prices via rigorous testing, we incorporate an attention mechanism to combine all candidate prices into a *virtual rational price* for each $p_t^{(i)}$ as

$$\tilde{p}_t^{(i)}(\Sigma) = \sum_{j \neq i} \text{ATT}_{ij} \cdot \tilde{p}_t^{(i)}(j), \text{ where } \text{ATT}_{ij} = \frac{\exp(w_{ij})}{\sum_{k \neq i} \exp(w_{ik})}. \quad (9)$$

w_{ij} are learnable parameters to calculate the attentions ATT_{ij} .

The virtual rational price $\tilde{p}_t^{(i)}(\Sigma)$ is expected to be cointegrated with the real price series of the stock i , i.e.,

$$u_t^{(i)} = \tilde{p}_t^{(i)}(\Sigma) - p_t^{(i)} \quad (10)$$

should be a stationary time series. To ensure this condition, we introduce a cointegration objective function to train the parameters β_{ij} and w_{ij} of the cointegration attention.

Cointegration Objective Functions. The cointegration objective function contains a *Regression Loss* and a *Stationary Regularization*. The regression loss is based on Eq. (10). If the time series of $u_t^{(i)}$ in Eq. (10) is a stationary series, the expectation of $\mathbb{E}[u_t^{(i)}]$ should be a constant with the time t . Then, Eq. (10) can be considered as using the attention result $\hat{p}_t^{(i)}(\Sigma)$ to regress stock price $p_t^{(i)}$, where $u_t^{(i)}$ is a regression error, i.e., $p_t^{(i)} = \hat{p}_t^{(i)}(\Sigma) - u_t^{(i)}$.

Thus, we use a mean square error (MSE) loss to minimize the regression error as

$$\mathcal{L}_\beta = \frac{1}{T \times I} \sum_{t=1}^T \sum_{i=1}^I \left(p_t^{(i)} - \hat{p}_t^{(i)}(\Sigma) \right)^2 \quad (11)$$

The stationary regularization is based on the DF test given in Eq. (6). According to Eq. (6), the series of $u_t^{(i)}$ can be considered as an autoregressive process of

$$u_t^{(i)} = \rho^{(i)} u_{t-1}^{(i)} + \varepsilon_t^{(i)}, \quad (12)$$

where $\rho^{(i)}$ is a regression parameter to be estimated. According to DF test, when $|\rho^{(i)}| < 1$, the series $u_t^{(i)}$ is stationary. Therefore, we introduce a constraint $|\rho^{(i)}| < 1$ in the parameter optimization and obtain the stationary regularization as

$$\begin{aligned} \mathcal{L}_\rho &= \frac{1}{T \times I} \sum_{t=1}^T \sum_{i=1}^I \left(u_t^{(i)} - \rho^{(i)} u_{t-1}^{(i)} \right)^2. \\ \text{s.t. } |\rho^{(i)}| &< 1, \forall i \in \{1, \dots, I\}, \end{aligned} \quad (13)$$

Through adjusting the learnable parameters w_{ij} and β_{ij} , the stationary regularization can make the constructed virtual rational price series $\hat{p}_t^{(i)}(\Sigma)$ as cointegrated as possible with the real price series $p_t^{(i)}$. Compared to the traditional pair trading strategy, which identifies sparse natural cointegration relationships between actual stock prices to detect stock-level irrational events, our method constructs a “soft” cointegrated series that serves as an estimated rational price for each stock. This approach is more efficient and universally applicable.

We combine the loss function in Eq.(11) with the stationary regularization in Eq.(13) to form the objective function for the estimated irrational price parameters learning as

$$\mathcal{L}_S = \mathcal{L}_\beta(B, W_C) + \lambda_1 \mathcal{L}_\rho(\rho), \quad \text{s.t. } |\rho^{(i)}| < 1, \quad (14)$$

where λ_1 is a hyper-parameter to balance the regression and $i \in \{1, \dots, I\}$. B and W_C are parameter matrices constituting of β_{ij} and w_{ij} , and ρ is the parameter vector constituting of $\rho^{(i)}$, which is constrained as absolute values less than 1.

Finally, we use $u_t^{(i)} = \hat{p}_t^{(i)}(\Sigma) - p_t^{(i)}$ as the *stock-level irrationality factor* to indicate local irrational events for the stock s_i in the period t . A high value of $u_t^{(i)}$ suggests that the stock s_i is currently undervalued, while a low value of $u_t^{(i)}$ indicates the opposite. We include the stock-level irrationality factor as part of $\Phi_t^{(i)}$ in Eq. (2) and input it into the stock return forecasting function in Eq. (3).

4 Market-level Irrationality Factors

4.1 Market-level Irrational Events

In a perfectly rational market, the price of an individual stock should be determined solely by its fundamental value and the price trend of rises and falls should not be highly synchronized either. However, in real-world stock markets, the prices of individual stocks are always impacted by overall market sentiment, resulting in the synchronicity of price behaviors [2]. A bullish market sentiment can lead to upward price movement in most stocks, while a bearish one does the opposite. For instance, between Oct. 1973 and Oct. 1974, US stock prices plummeted by 44.1%, whereas from Apr. 1994 to Apr. 1999, they witnessed a remarkable surge of 184.8%. The 2013 Nobel laureate in Economics, Robert J. Shiller posits that such market behaviors are linked to irrationality factors within populations [40]. To quantitatively characterize the irrational behavior of the entire market, we provide the following definition of *Market-level Irrational Events*.

DEFINITION 6 (MARKET-LEVEL IRRATIONAL EVENTS). Given the stocks $\{s_1, \dots, s_I\}$ in a market, we denote the increase and decrease of the stocks’ returns during period t as $\{\delta_t^{(1)}, \dots, \delta_t^{(I)}\}$, where $\delta_t^{(i)} = 1$ if the return of stock s_i increases more than a threshold, $\delta_t^{(i)} = -1$ if that decreases more than a threshold, otherwise $\delta_t^{(i)} = 0$. We define that there is a market-level irrational event when the ratio of stocks with $\delta_t^{(i)} = 1$ or with $\delta_t^{(i)} = -1$ are higher than a threshold H_m , i.e.,

$$\left| \sum_{i=1}^I \delta_t^{(i)} \right| > H_m. \quad (15)$$

In an efficient and rational market, stock fluctuations typically do not occur in broad synchrony. Therefore, Definition 6 denotes a market-level irrational event as an anomalous synchronous fluctuation of all stocks in a market.

4.2 Market-level Irrationality Factor Learning

As Robert J. Shiller emphasizes, the reasons for irrational market behaviors are interconnected with many different factors, such as market structure, culture, and even investor psychology [40]. Simply including market-level irrational events defined in Eq. (15) as factors in a stock return forecasting model cannot adequately capture these underlying factors. Therefore, we adopt a representation learning method to learn comprehensive representation of market-level irrationality. The representation learning comprises two modules: *i*) a market-level representation extraction module for constructing representations that can accurately reflect the behavior of the entire market; and *ii*) two self-supervised tasks aimed at incorporating market-level irrationality into the market representations.

4.2.1 Market Representation Extraction. The market representation extraction module first extracts a dynamic behavior representation for each stock, and then combines these stock representations as an overall market representation.

Dynamic Stock Representation Extraction. We first extract representation that captures the dynamic behavior of each stock based on its historical features. Specifically given the short-term historical features in a sliding window for a stock, denoted as

$(\mathbf{e}_{t-L+1}, \dots, \mathbf{e}_{t-1}, \mathbf{e}_t)$ ², where L is a historical slice window. We employ a self-attention mechanism to compute the interdependences between the stock features in period t and those in historical periods $\tau \in \{t-L+1, \dots, t\}$ as

$$\alpha_{t,\tau} = (\mathbf{W}_S \mathbf{e}_t + \mathbf{b})^\top (\mathbf{W}_S \mathbf{e}_\tau + \mathbf{b}) / \sqrt{D_e}, \quad (16)$$

where $\mathbf{W}_S \in \mathbb{R}^{D_e \times D_e}$ and $\mathbf{b} \in \mathbb{R}^{D_e}$ are learnable parameters, and D_e is the dimensionality of the feature \mathbf{e}_t . Next, we use $\alpha_{t,\tau}$ to calculate the attention weight between t and τ as $\text{ATT}_{t,\tau}$ and employ the weights to combine \mathbf{e}_τ as

$$\mathbf{r}_t = \mathbf{e}_t \| \sum_{\tau=t-L+1}^t \text{ATT}_{t,\tau} \cdot \mathbf{e}_\tau, \text{ where } \text{ATT}_{t,\tau} = \frac{\exp(\alpha_{t,\tau})}{\sum_{k=t-L+1}^t \exp(\alpha_{t,k})}. \quad (17)$$

Here, $\|$ is the concatenation operator and \mathbf{r}_t is the final representation of the stock in the period t . The representation \mathbf{r}_t not only captures the current stock features \mathbf{e}_t , but also incorporates temporal dynamic patterns from historical features \mathbf{e}_τ , such as trends or volatility. Therefore, we refer to it as a *dynamic stock representation*.

Market Representation Generation. In the dynamic stock representations \mathbf{r}_t , a stock is described exclusively by the dynamic behaviors of its features, implying that two distinct stocks with identical short-term features may yield the same representation. However, in practice, even with the same short-term features, different stocks can influence market sentiment differently due to factors such as being an index component, industry sector, and investor perception. For example, if two companies of the same type have similar short-term historical features, but one is a component of a major index while the other is not, the index component stock will certainly have a greater impact on market sentiment. To capture this phenomenon, we propose a stock ID-involved stock-dependent weight, which is used to combine dynamic stock representations as a market representation.

Specifically, given the i -th stock, we utilize one-hot encoding to represent its ID as a binary vector of $\mathbf{i}^{(i)} \in \{0, 1\}^I$, where the i -th element in $\mathbf{i}^{(i)}$ is 1 and all others are 0. Next, we construct an ID representation matrix as $\mathbf{W}_i = (\mathbf{w}_i^{(1)}, \dots, \mathbf{w}_i^{(i)}, \dots, \mathbf{w}_i^{(I)}) \in \mathbb{R}^{2D_e \times I}$, where $\mathbf{w}_i^{(i)} \in \mathbb{R}^{2D_e}$ is the column vector of the matrix. Using \mathbf{W}_i as learnable parameters, we calculate the *ID-involved stock-dependent weight* for the stock s_i as

$$\eta_t^{(i)} = \text{ReLU}(\mathbf{w}_\eta^\top (\mathbf{W}_i^\top \mathbf{i}^{(i)} + \mathbf{r}_t^{(i)})), \quad (18)$$

where $\text{ReLU}(\cdot)$ is a Rectified Linear Unit activation function. $\mathbf{w}_\eta \in \mathbb{R}^{2D_e}$ is a learnable parameter.

Given the I stocks in the market, we use the stock-dependent weight $\eta_t^{(i)}$ to combine the dynamic stock representations as a market representation for the period t :

$$\mathbf{m}_t = \frac{\sum_{i=1}^I \eta_t^{(i)} \cdot \mathbf{r}_t^{(i)}}{\sum_{i'=1}^I \eta_t^{(i')}}. \quad (19)$$

Here, we use $\sum_{i'=1}^I \eta_t^{(i')}$ to normalize the market representation. Thus, the magnitude of \mathbf{m}_t is not affected by the number I of stocks.

²We omit the superscript i of stock features where it causes no ambiguity.

4.2.2 Self-supervised Tasks for Market-level Irrationality Factors Learning. This module uses two self-supervised tasks, i.e., *sub-market comparative learning* and *market synchronism prediction*, to incorporate market-level irrationality into the market representation.

Sub-market Comparative Learning. According to Definition 6, market-level irrational events are defined as anomalous synchronous fluctuations of stocks in a market. This means that during a market-level irrational event, different parts of the stock market should exhibit similar behaviors. To capture this characteristic in the market representations, we employ a market comparative learning approach to encourage similar representations among different sub-markets during the same time periods.

Specifically, for each time period, we randomly divide the stocks in a market into two sub-markets, labeled as sub-market 1 and sub-market 2. Using Eq. (19), we calculate corresponding market representations for these two sub-markets, denoted as $\mathbf{m}_t^{(1)}$ and $\mathbf{m}_t^{(2)}$. In the sub-market comparative learning, we consider $\mathbf{m}_t^{(1)}$ and $\mathbf{m}_{t'}^{(2)}$ as a positive sample if $t = t'$, and as a negative sample if $t \neq t'$. We adopt the InfoNCE loss [35] to ensure that the positive sample pair $\mathbf{m}_t^{(1)}$ and $\mathbf{m}_{t'}^{(2)}$ is consistent for all periods as

$$\mathcal{L}_C = -\frac{1}{T} \sum_{t=1}^T \log \frac{\mathcal{D}(\mathbf{m}_t^{(1)}, \mathbf{m}_t^{(2)})}{\mathcal{D}(\mathbf{m}_t^{(1)}, \mathbf{m}_t^{(2)}) + \sum_{t' \neq t} \mathcal{D}(\mathbf{m}_t^{(1)}, \mathbf{m}_{t'}^{(2)})}. \quad (20)$$

Here, $\mathcal{D}(\cdot, \cdot)$ is a criterion function defined as

$$\mathcal{D}(\mathbf{m}_t^{(1)}, \mathbf{m}_{t'}^{(2)}) = \exp\left(\frac{1}{|t-t'|+1} (\mathbf{w}_M^\top (\mathbf{m}_t^{(1)} \| \mathbf{m}_{t'}^{(2)}) + b_M)\right), \quad (21)$$

where \mathbf{w}_M and b_M are learnable parameters. We weight each sample pair using $\frac{1}{|t-t'|+1}$ to ensure that the negative example with a larger time difference $|t-t'|$ contributes to a smaller similarity in the loss function.

Market Synchronism Prediction. This task directly uses market representations to predict the synchronism of market defined in Eq. (15). Specifically, we quantify $\sum_i \delta_t^{(i)}$ as a one-hot label Δ_t with three classes: two classes correspond to market-level irrational events, i.e., $\Delta_t = (1, 0, 0)$ if $\sum_i \delta_t^{(i)} > H_m$, and $\Delta_t = (0, 1, 0)$ if $\sum_i \delta_t^{(i)} < -H_m$; one class for other situations, i.e., $\Delta_t = (0, 0, 1)$ if $H_m \geq \sum_i \delta_t^{(i)} \geq -H_m$. Then, we use the market representation \mathbf{m}_{t-1} with a Multilayer Perceptron (MLP) neural network to predict the label Δ_t as

$$\hat{\Delta}_t = \text{Softmax}(\text{MLP}(\mathbf{m}_{t-1})), \quad (22)$$

where $\hat{\Delta}_t$ is the predicted label. We use a cross entropy loss to minimize the error between Δ_t and $\hat{\Delta}_t$ as

$$\mathcal{L}_P = -\frac{1}{T} \sum_{t=1}^T \left(\sum_{k=1}^3 \Delta_{t,k} \cdot \log(\hat{\Delta}_{t,k}) \right), \quad (23)$$

where $\Delta_{t,k}$ is the k -th element of the vector Δ_t . Finally, the joint loss function for the market-level irrationality factor learning is

$$\mathcal{L}_M = \mathcal{L}_C + \lambda_2 \mathcal{L}_P. \quad (24)$$

Trained by the two self-supervised tasks, the market representation \mathbf{m}_t is incorporated with the information of market-level irrationality. We consider \mathbf{m}_t as a *Market-level Irrationality Factor*, which is used as a part of $\Phi_t^{(i)}$ in Eq. (2).

5 Forecasting Model and Training

The stock-level irrationality factor $u_{t-1}^{(i)}$ and the market-level irrationality factors \mathbf{m}_t can be used as inputs to enhance the performance of any stock return forecasting model. Therefore, we name them as universal indicative factors. To fully harness the potential of the universal indicative factors, we propose a hybrid model combining Transformers and graph neural networks to effectively capture the temporal and relation dependencies in stock data.

Forecasting Model. Specifically, we concatenate the stock-level factor $u_{t-1}^{(i)}$ and stock features in a historical period window of size L as $\mathbf{g}_{t-L}^{(i)} = \mathbf{e}_{t-L}^{(i)} \| u_{t-L}^{(i)}$, where $l \in \{1, \dots, L\}$. We then use a Transformer to encode the historical $\mathbf{g}_{t-L}^{(i)}$ as

$$\mathbf{c}_{t-1}^{(i)} = \text{Transformer}(\mathbf{g}_{t-L}^{(i)}, \dots, \mathbf{g}_{t-1}^{(i)}). \quad (25)$$

Next, we adopt a graph attention-based *Stock Relation Dependency* module to capture the relation among stocks, where the attention between stock s_i and s_j in period t is calculated as

$$\gamma_t^{(i,j)} = (\mathbf{W}_Y (\mathbf{W}_t^\top \mathbf{r}_t^{(i)}) + \mathbf{r}_t^{(j)})^\top (\mathbf{W}_Y (\mathbf{W}_t^\top \mathbf{r}_t^{(j)} + \mathbf{r}_t^{(i)})), \quad (26)$$

where $\mathbf{W}_Y \in \mathbb{R}^{2D_e \times 2D_e}$ is a learnable parameter matrix, the vectors $\mathbf{W}_t^\top \mathbf{r}_t^{(j)}$ and $\mathbf{r}_t^{(i)}$ are learned in Eq. (18). We utilize $\gamma_t^{(i,j)}$ to compute the attention of stock s_i towards other stocks, and leverage these attentions to combine $\mathbf{c}_{t-1}^{(i)}$ generated by Eq. (25) as

$$\mathbf{d}_{t-1}^{(i)} = \sum_{j=1}^I \text{ATT}_{i,j} \cdot \mathbf{c}_{t-1}^{(j)}, \text{ where } \text{ATT}_{i,j} = \frac{\exp(\gamma_t^{(i,j)})}{\sum_{k=1}^I \exp(\gamma_t^{(i,k)})}. \quad (27)$$

The representation $\mathbf{d}_{t-1}^{(i)}$ captures the interdependence between stock s_i and all other stocks $s_j \forall j \neq i$. Finally, the future stock return at period t is predicted using a MLP network as

$$\hat{y}_t^{(i)} = \text{MLP}(\mathbf{c}_{t-1}^{(i)} \| \mathbf{d}_{t-1}^{(i)} \| \mathbf{m}_{t-1}), \quad (28)$$

where \mathbf{m}_{t-1} is the market-level irrationality factor at period $t-1$, and $\hat{y}_t^{(i)}$ is a forecasted return for the stock i .

RankIC Incorporated Loss Function. A direct way to optimize the stock return forecasting model is using a mean square error (MSE) loss as

$$\mathcal{L}_{\text{MSE}} = \frac{1}{T \times I} \sum_{t=1}^T \sum_{i=1}^I (\hat{y}_t^{(i)} - y_t^{(i)})^2. \quad (29)$$

In addition to the MSE loss, we also incorporate a *RankIC regularization* to enhance the adaptability of our forecasting model to investment strategies. For many investment strategies, accurate ranking prediction of stock returns holds greater significance than precise numerical prediction of returns. For example, our experiment adopts a *long-short hedging strategy*³ as the backbone investment strategy for our forecasting model. For each trading period, the long-short hedging strategy first ranks stocks in the market based on their predicted returns. The strategy then takes long positions (buys) in the top- N stocks with the highest predicted returns and short positions (sells) in the bottom- N stocks with the lowest predicted returns. As long as the predicted order of

³https://en.wikipedia.org/wiki/Long/short_equity

returns is accurate, this hedging strategy can be profitable (Refer to Appendix A.1 for a detailed description of the hedging strategies).

We adopt the information correlation (IC) between the predicted stock return and the rank of the true return as a regularization term for the forecasting model training. Specifically, we denote $z_t^{(i)}$ as the rank of the stock s_i 's return at period t . The IC between $z_t^{(i)}$ and $\hat{y}_t^{(i)}$ is calculated as

$$\text{IC}_t = \frac{1}{I} \sum_{i=1}^I \frac{(\hat{y}_t^{(i)} - \text{mean}(\hat{y}_t)) (\hat{z}_t^{(i)} - \text{mean}(z_t))}{\text{std}(\hat{y}_t) \text{std}(z_t)}, \quad (30)$$

where $\text{mean}(\cdot)$ and $\text{std}(\cdot)$ are mean and standard deviation for $\hat{y}_t = (\hat{y}_t^{(1)}, \dots, \hat{y}_t^{(I)}, \dots)$ and $z_t = (z_t^{(1)}, \dots, z_t^{(I)}, \dots)$ with $i \in \{1, \dots, I\}$. The IC_t in Eq. (30) measures the normalized correlation between the predicted return and the rank of true return. For the predicted returns $\hat{y}_t^{(i)}$, even if the prediction accuracy is not very high, as long as their rank is similar to the true return, they can also have a high IC_t .

We define the RankIC regularization for all samples as $\mathcal{L}_{\text{IC}} = -\frac{1}{T} \sum_{t=1}^T \text{IC}_t$. The final objective function is defined as

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_3 \mathcal{L}_{\text{IC}}, \quad (31)$$

By incorporating RankIC regularization into the objective function, the model can better capture the relative rank relationship among stocks, which can improve the performance of our model in the hedging investment strategy.

Factor-learning + Forecasting-training Procedure. Finally, we implement our model training procedure in two steps. The first step is factor-learning, which uses the training dataset to learn stock-level and market-level irrationality factors by minimizing self-supervised losses \mathcal{L}_S (Eq. (14)) and \mathcal{L}_M (Eq. (24)), respectively. The second step is forecasting-training, which minimizes the objective function \mathcal{L} (Eq. (31)) using supervised labels $y_t^{(i)}$ and $z_t^{(i)}$.

6 Experiments

6.1 Experiment Setups

Datasets. We employ data from the stock markets of the two largest economies: United States (US) and China (CN) to evaluate the performance of our model. The US stock market data contains totally 8993 stocks in American Stock Exchange, New York Stock Exchange and National Association of Securities Dealers Automated Quotations⁴ from 03/27/2000 to 10/10/2020, with a test set from 10/10/2015 to 10/10/2020. The CN stock market data contains totally 5148 stocks in Shanghai and Shenzhen Stock Exchanges⁵ from 01/10/2006 to 03/01/2023, with a test set from 03/01/2018 to 03/01/2023. We use Alpha360 features of the open-source quantitative investment platform Qlib [56] to extract six price/volume-related features for each stock, which are opening price, closing price, highest price, lowest price, volume-weighted average price, and trading volume. We use the closing price as the stock price to calculate the return. We compare our model to the baselines under *Rolling Retraining* setup,

⁴<https://github.com/microsoft/qlib/>

⁵https://github.com/cheditc/investment_data/

where we retrain the models every 30 trading days. For each re-training, data of new trading days and initial public offerings (IPO) stocks are incorporated into the training sets.

Model Implementation & Baselines. For the comparative experiments, seven baseline methods are utilized, including three stock return forecasting methods and four time series forecasting methods.

First, we select two state-of-the-art stock return forecasting models as our baselines

- *DoubleAdapt*, (DA) [61] is a stock return forecasting approach to transform stock data into a locally stationary distribution to handle the impact of distribution shifts in financial data.

- *D-Va* [24] is a stock return forecasting approach that addresses stochasticity in both input and target sequences using a hierarchical variational autoencoder technique.

Second, since the stock RoR prediction can also be considered as a normal time series forecasting problem, we introduce two state-of-the-art time series forecasting methods as baselines. These baselines are equipped with components to handle some special characteristics of complex time series, such as frequency and non-stationarity. These characteristics are also existed in the stock data.

- *SAN* [32] is a time series forecasting approach that introduces a comprehensive normalization framework for non-stationary data.

- *FreTS* [57] is a time series forecasting method that incorporates frequency analysis for forecasting tasks. This approach considers both global perspectives and energy compaction to exploit the frequency information in time series.

Our model employs self-supervised methods to extract hidden factors from stock data. For comparison, we also adopt three state-of-the-art self-supervised methods in our baselines, including a stock return forecasting method and two time series forecasting methods.

- *Co-CPC* [44] is a contrastive learning method in stock market analysis. This approach shuffles the future states of stocks and identifies them as positive or negative samples.

- *Basisformer* [34] is a self-supervised time series forecasting approach that integrates contrastive learning and basis learning for forecasting tasks.

- *SimMTM* [6] is a self-supervised time series forecasting method that utilizes a masked pre-training framework along with the transformer architecture.

Metrics. We evaluate the performance of our model from two perspectives: forecasting accuracy and investment strategy performance. For forecasting accuracy, we use RMSE, MAE, IC, ICIR, RankIC and RankICIR as evaluation metrics. The details about the forecasting accuracy metrics are given in Appendix.

For investment strategy performance evaluation, we put the RoR prediction results of our model and baseline into the long-short portfolio hedging investment strategy defined in Eq. (33), and calculate the returns of the investment strategy for each period in the test set. We set the portfolio size N in Eq. (33) to 10% of the total number of stocks per day. To model the effect of transaction costs on the strategy, we adjust the RoR of the portfolio as $R_t = y_t^{\text{STR}} - 0.1\% \times \frac{\text{TC}}{N}$, where TC is the number of stocks that is transacted at time t . Here, the transaction cost of trading a stock is set to 0.1%.

Carrying out trading simulations day by day, we have a portfolio return sequence as $R = (R_1, \dots, R_t, \dots, R_T)$. In the experiment, the investment strategy performance with the return sequence R is evaluated by five metrics: Annualized Return, Annualized Volatility, Sharpe Ratio, Maximum DrawDown and Calmar Ratio.

- *Annualized Return (AR)*: Annualized Return is a metric that gauges the overall profit generated by investing in line with model predictions on an annual basis. This metric is determined by adjusting the average daily portfolio return to account for the total number of trading days in a year, which is calculated as $\text{AR} = \text{mean}(R) \times N_Y$, where N_Y is the total number of trading days in a year.

- *Annualized Volatility (AV)*: Annualized Volatility measures the risk of the portfolio, which is calculated as $\text{AV} = \text{std}(R) \times \sqrt{N_Y}$.

- *Sharpe Ratio (SR)*: The Sharpe Ratio is a metric to measure the risk-adjusted return of an investment. It is a measure of how much excess return is generated for a given level of volatility risk. Given the annualized return and annualized volatility, the Sharpe Ratio (SR) is calculated as $\text{SR} = \text{AR}/\text{AV}$.

- *Maximum DrawDown (MDD)*: The Maximum Drawdown quantifies the worst performance of a portfolio within specific periods. It is calculated as the maximum observed loss that a portfolio experiences during a decline from its highest point to its lowest point. The MDD during T periods is calculated as

$$\text{MDD} = \max_{\tau \in [1, T]} \left(\max_{t \in [1, \tau]} \left(\sum_{\leq t} R_t - \sum_{\leq \tau} R_\tau \right) \right) \quad (32)$$

- *Calmar Ratio (CR)* is the risk-adjusted AR based on Maximum DrawDown. It is calculated as $\text{CR} = \text{AR}/\text{MDD}$.

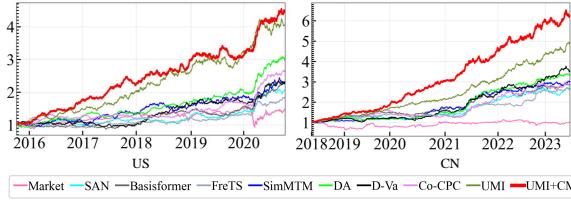
- *Cumulative Wealth (CW)*: Cumulative wealth is the total accumulation of stock returns over time, reflecting an individual's or entity's net worth resulting from investments. It is calculated as $\text{CW}_t = \prod_{t'=1}^t (1 + y_{t'}^{\text{(STR)}})$.

6.2 Forecasting and Investment Performance

Overall Performance. Table 1 presents the experimental results of the forecasting and investment performance. Each experiment was repeated five times, and the average results are reported. From the table, we have the following observation. *Firstly*, our model achieves the best performance in both US and CN markets for forecasting and investment performance, demonstrating its effectiveness. *Secondly*, models specifically designed for stock data, such as DoubleAdapt, D-Va, Co-CPC, and our model, demonstrate superior performance compared to general time series forecasting baselines. This suggests that stock series possess unique characteristics that require specialized mechanisms for effective modeling. *Thirdly*, self-supervised methods outperform purely supervised methods. Specifically, Co-CPC surpasses supervised stock forecasting methods, while SimMTM and Basisformer outperform supervised time series forecasting methods. Self-supervised methods can extract hidden information from multiple perspectives through various self-supervised tasks, leading to better performance. Our model employs a self-supervised approach to uncover market irrationality factors that previous methods have overlooked, resulting in significant performance improvements. *Finally*, our model's performance improvements in the rank accuracy (IC/ICIR/RankIC/RankICIR) is

Table 1: Comparison of model performance on US and CN markets, with the best results emphasized in bold and the second-best results underlined for clarity. Metrics with (↓) indicate that lower values are better, otherwise the higher values are better.

	Market	Metrics	DA	D-Va	Co-CPC	SAN	Basisformer	FreTS	SimMTM	<u>UMI</u>	Imp.	NR	NS	NM	ND	<u>UMI +CM</u>
Forecasting Performance	US	RMSE(↓)	.0398	.0399	.0401	.0401	.0402	.0400	.0401	.0394	1.1%	.0395	.0397	.0396	.0395	.0393
		MAE(↓)	.0234	.0235	<u>.0234</u>	.0238	.0237	.0239	.0235	.0231	1.1%	.0231	.0232	.0233	.0231	.0230
		IC(↑)	<u>.054</u>	.049	.052	.049	.050	.048	.053	.057	6%	.055	.055	.056	.056	.061
		ICIR(↑)	.624	.555	.591	.569	.581	.540	.621	.663	6%	.649	.645	.655	.637	.721
		RankIC(↑)	<u>.057</u>	.055	.055	.053	.053	.050	.055	.061	9%	.058	.056	.058	.057	.060
		RankICIR(↑)	.615	.606	.603	.588	.598	.553	.604	.666	8%	.646	.606	.635	.617	.663
	CN	RMSE(↓)	.0354	.0357	.0355	.0360	.0358	.0359	.0358	.0350	1.1%	.0351	.0353	.0352	.0351	.0349
		MAE(↓)	.0225	.0226	<u>.0227</u>	.0229	.0230	.0228	.0227	.0222	1.4%	.0223	.0223	.0225	.0224	.0221
		IC(↑)	.069	<u>.071</u>	.071	.065	.068	.062	.068	.078	10%	.072	.072	.074	.073	.079
		ICIR(↑)	.735	.747	<u>.762</u>	.702	.732	.670	.746	.831	9%	.773	.775	.785	.791	.828
		RankIC(↑)	.069	.067	<u>.070</u>	.061	.064	.062	.066	.073	5%	.070	.071	.070	.071	.074
		RankICIR(↑)	.730	.708	<u>.737</u>	.652	.693	.673	.699	.777	5%	.754	.771	.758	.776	.786
	Market	Metrics	DA	D-Va	Co-CPC	SAN	Basisformer	FreTS	SimMTM	<u>UMI</u>	Imp.	NR	NS	NM	ND	<u>UMI +CM</u>
Investment Performance	US	AR(↑)	<u>.220</u>	.177	.213	.154	.172	.142	.180	.289	31%	.257	.227	.229	.253	.307
		AV(↓)	.174	.157	.156	.160	<u>.155</u>	.156	.165	.144	7%	.153	.148	.151	.152	.141
		SR(↑)	1.260	1.134	<u>1.366</u>	.971	1.119	.901	1.100	.207	47%	1.667	1.529	1.516	1.670	2.126
		MDD(↓)	<u>.184</u>	.194	<u>.184</u>	.284	.262	.304	.254	.129	30%	.161	.164	.163	.152	.126
		CR(↑)	<u>.198</u>	.927	1.165	.538	.649	.475	.711	.236	87%	1.593	1.386	1.402	1.664	2.434
		AR(↑)	.261	<u>.279</u>	.274	.215	.219	.222	.254	.358	28%	.340	.313	.337	.336	.395
	CN	AV(↓)	.153	.147	<u>.143</u>	.157	.159	.151	.157	.133	7%	.140	.147	.134	.135	.131
		SR(↑)	1.710	1.902	<u>1.921</u>	1.377	1.380	1.477	1.623	.2680	39%	2.435	2.126	2.523	2.495	3.011
		MDD(↓)	<u>.098</u>	.117	.106	.193	.223	.185	.166	.072	26%	.074	.083	.094	.078	.072
		CR(↑)	<u>.266</u>	2.369	2.584	1.115	.981	1.210	1.534	.4958	86%	4.600	3.748	3.597	4.293	5.515

**Figure 2: The Cumulative Wealth in US and CN markets.**

much higher than value accuracy (RMSE/MAE), verifying the effectiveness of the RankIC regularization in Eq. (31). For the hedging investment strategy, the relative rank of stock returns is more crucial. Hence, our model’s investment performance improvements are significantly greater than its forecasting improvements. Moreover, Fig. 2 compares the cumulative wealth of *UMI* and the baselines. We also provide the cumulative wealth for the market return in Fig. 2. For the CN and US markets, we use CSI500 and S&P500 as their respective market returns. The figure shows that *UMI* significantly outperforms the baselines, with its advantages expanding over time. This indicates that even a small forecasting performance advantage can lead to substantial profit increases in a long-term investment.

Cross-Market Factor Learning Experiments. The *UMI* model works as a factor-learning + forecasting-training framework. In the overall performance experiments, both stages occur within the same market. In fact, we can use one market data in the self-supervised learning process to enhance the performance of the other market. In the last column of Tab. 1 (*UMI +CM*), we conducted a cross-market factor learning experiment. For US market forecasting, we learned the market-level irrationality factor in the CN market (temporarily disabling the learnable ID representation W_i) and then further tuned the entire model in the US market. For CN market forecasting, we reversed the process. The cumulative wealth of *UMI +CM* is also shown in Fig. 2. The results show that *UMI +CM* has

Table 2: The universality of the irrationality factors.

Data	Metrics	DA+F	Imp.	D-Va+F	Imp.	CPC+F	Imp.	<i>UMI</i>
US	AR(↑)	0.241	10%	0.233	32%	0.249	17%	0.289
	AV(↓)	0.153	12%	0.149	5%	0.153	2%	0.144
	SR(↑)	1.580	25%	1.570	38%	1.628	19%	2.007
	MDD(↓)	0.139	24%	0.148	23%	0.148	20%	0.129
	CR(↑)	1.742	45%	1.565	69%	1.704	46%	2.236
Data	Metrics	DA+F	Imp.	D-Va+F	Imp.	CPC+F	Imp.	<i>UMI</i>
CN	AR(↑)	0.342	31%	0.348	25%	0.334	22%	0.358
	AV(↓)	0.142	7%	0.139	5%	0.137	4%	0.133
	SR(↑)	2.411	41%	2.502	32%	2.443	27%	2.680
	MDD(↓)	0.090	8%	0.084	29%	0.091	14%	0.072
	CR(↑)	3.820	43%	4.170	76%	3.663	42%	4.958

an improved performance, demonstrating our model’s ability to leverage transferable knowledge across markets.

Ablation Study. Table 1 also shows the performance of four *UMI* model variants in the 2nd to 5th columns from the end: *i*) *NS* without the stock-level irrationality factors, *ii*) *NM* without the market-level irrationality factors, *iii*) *NR* without the RankIC regularization, and *iv*) *ND* without the Stock Relation Dependency module in Eq. (26) and Eq. (27), *i.e.*, only uses a Transformer as the forecasting function. The performance order is: *NS* < *NM* < *ND* < *NR* < *UMI*. These results highlight the importance of the modules to model performance. Additionally, stock-level factors seem more beneficial than the market-level, likely because they provide detailed, stock-specific information, while the market-level factors are uniform across all stocks.

Universality of the Irrationality Factors. The irrationality factors extracted by our model can also be used in various forecasting models. To evaluate this feature, we added these factors as inputs to three stock forecasting baselines (DoubleAdapt, D-Va, and Co-CPC). Table 2 shows the investment performance, with “+F” indicating models with additional factor inputs. The results demonstrate that

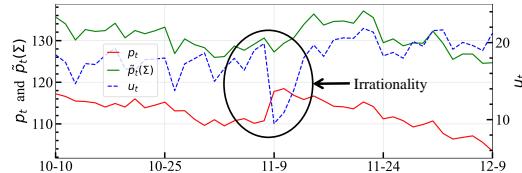


Figure 3: Visualization of u_t value, cointegrated price $\tilde{p}_t(\Sigma)$ and price p_t of stock SZ002253.

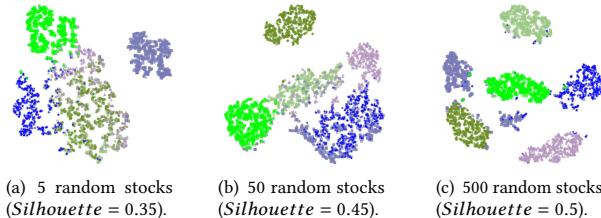


Figure 4: Market representations from six days.

all models are significantly improved by our irrationality factors, confirming their universality. Despite these enhancements, our *UMI* model still outperforms the baselines, highlighting the effectiveness of our efficient forecasting model and RankIC regularization design.

6.3 Exploratory Analysis

Here, we illustrate how the irrationality factors work via visualized exploratory analysis.

Stock-level Irrationality Factors. Fig. 3 visualizes the series of the price p_t , estimated rational price $\tilde{p}_t(\Sigma)$ and the stock-level irrationality factors u_t for the stock Wisesoft (SZ002253) in the CN market during 2016. Here, p_t and $\tilde{p}_t(\Sigma)$ exhibit a cointegration relationship, with their difference, u_t , being stationary. There is an anomalous fluctuation of u_t on Nov. 9, 2016, indicating an irrational event. This happened because the company's name is a homophone for "Trump's big win" and on that day, Donald Trump happened to be elected President of the United States, causing the stock price to hit its limit. This is a typical example of stock-level irrationality⁶, demonstrating that our model can effectively capture irrational events in the market.

Market-level Irrationality Factors. Here, we examine how the individual stock dynamic representations construct the market-level irrationality factor. Specifically, we selected two trading days with the market-level irrational event of $\Delta_t = (1, 0, 0)$, two days with the event of $\Delta_t = (0, 1, 0)$, and two days without the irrational event. For each day, we generate market-level irrationality factors using a number of (5, 50, 500) randomly selected stocks. In Fig. 4(a), each point represents an instance of the market-level factor generated from 5 randomly selected stocks, with points of the same color corresponding to the same trading day. In Fig 4(b) and Fig 4(c), we increase the randomly selected stocks to 50 and 500. We used t-SNE [13] to convert the instances into 2D for visualization. We find that when the number of stocks reaches 500, days with different Δ_t can be clearly distinguished. This suggests our approach effectively identifies market-level irrational events from ordinary trading days. Additionally, days with the same Δ_t label are distinguishable by

our factors, indicating that market-level factors provide sufficient information to capture overall market characteristics. The Silhouette score [39], which measures clustering cohesion, increases with the number of stocks, showing that a larger stock sample enhances the market representation capability of our factors.

7 Related Works

Deep Learning for Stock Prediction. In recent years, various types of deep neural network models have been introduced for stock data analysis [4, 11, 42, 45, 52, 54, 55, 59]. Some aim to forecast accurate stock prices/returns [53, 63], while others directly design trading strategies using reinforcement learning [8, 11, 31]. To better handle stock data, advanced deep learning methods proposed sophisticated network architectures [5, 15, 62]. Other methods incorporate supplementary information, such as fund manager details and stock comments, into the models [1, 27, 28]. Some self-supervised methods employ contrastive learning to exploit information in different types of positive/negative samples, such as daily-frequency and minutely-frequency [14], raw data and shuffled historical data [44], as well as correlated price movements [29].

Stock Factor Learning. For stock-level analysis, all deep learning-based methods can be considered as using representational learning to extract stock indicative factors [26, 30]. However, very few studies involve market irrationality. Moreover, in the stock-level factor analysis, we adopt a novel method to exploit cointegration relations, which is particularly innovative as this type of stock relationship is rarely considered. Most works focus on designing specialized neural networks [50, 51] to model stock relations, or incorporate additional information about stock relations, such as collective investments of funds [1, 28] and textual media about firm relevance [3]. For the market-level, traditional methods use stock market indexes to represent market overall state. A straightforward method is using the market indexes as a feature of stock forecasting models [58]. Some advanced methods use representations of some selected stocks, such as stocks with top-ranked returns, to indicate the overall state of market [51]. However, there are still few methods that focus on market irrationalities.

8 Conclusions

We introduce *UMI*, a method that exploit irrational events in stock markets from stock return forecasting. By pinpointing the gap between factual and estimated rational prices, *UMI* uncovers stock-level irrationalities. Moreover, it detects market-level irrationalities through representing anomalous synchronous of stocks. Extensive experiments in the U.S. and Chinese markets validated our model's exceptional performance and universality.

Acknowledgments

This work is supported by the National Key R&D Program of China (2023YFC3304700). Prof. Jingyuan Wang's work was partially supported by the National Natural Science Foundation of China (No. 72222022, 72171013). Prof. Junjie Wu's work was partially supported by the National Natural Science Foundation of China (No. 72242101, 72031001), and the Outstanding Young Scientist Program of Beijing Universities (JWZQ20240201002).

⁶<https://www.ft.com/content/22ffdb04-6d08-34e6-9543-73f744a500dc>

References

- [1] Chi Chen, Li Zhao, Jiang Bian, Chunxiao Xing, and Tie-Yan Liu. [n. d.]. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction.
- [2] Son-Nan Chen. 1982. An examination of risk-return relationship in bull and bear markets using time-varying betas. *Journal of Financial and Quantitative Analysis* 17, 2 (1982), 265–286.
- [3] Rui Cheng and Qing Li. 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In *AAAI'21*, Vol. 35. 55–62.
- [4] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. 2020. Hierarchical multi-scale gaussian transformer for stock movement prediction. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 4640–4646.
- [5] Yi Ding, Weiqing Liu, Jiang Bian, Daoqiang Zhang, and Tie-Yan Liu. [n. d.]. Investor-imitator: A framework for trading knowledge extraction.
- [6] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Ming-sheng Long. 2023. SimMTM: A simple pre-training framework for masked time-series modeling. In *NeurIPS'23*.
- [7] Robert F Engle and Clive WJ Granger. 1987. Co-integration and error correction: Representation, estimation, and testing. *Econometrica* 55, 2 (1987), 251–276.
- [8] Siyu Gao, Yunbo Wang, and Xiaokang Yang. 2023. Stockformer: Learning hybrid trading machines with predictive coding. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*. 4766–4774.
- [9] Alberto Giovannini and Philippe Jorion. 1989. The time variation of risk and return in the foreign exchange and stock markets. *The Journal of Finance* 44, 2 (1989), 307–325.
- [10] Suiming Guo, Qianrong Shen, Zhiqian Liu, Chao Chen, Chaoxiong Chen, Jingyuan Wang, Zhetao Li, and Ke Xu. 2023. Seeking Based on Dynamic Prices: Higher Earnings and Better Strategies in Ride-on-Demand Services. *IEEE Transactions on Intelligent Transportation Systems* (2023), 1–16.
- [11] Li Han, Nan Ding, Guoxuan Wang, Dawei Cheng, and Yuqi Liang. [n. d.]. Efficient continuous space policy optimization for high-frequency trading.
- [12] Kethmi Hirushini Hettige, Jiahao Ji, Shili Xiang, Cheng Long, Gao Cong, and Jingyuan Wang. 2024. AirPhyNet: Harnessing Physics-Guided Neural Networks for Air Quality Prediction. In *Proceedings of the Twelfth International Conference on Learning Representations*.
- [13] Geoffrey E Hinton and Sam Roweis. 2002. Stochastic neighbor embedding. In *NeurIPS'02*, Vol. 15.
- [14] Min Hou, Chang Xu, Yang Liu, Weiqing Liu, Jiang Bian, Le Wu, Zhi Li, Enhong Chen, and Tie-Yan Liu. 2021. Stock trend prediction with multi-granularity data: A contrastive learning approach with adaptive fusion. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. 700–709.
- [15] Kentaro Imajo, Kentaro Minami, Katsuya Ito, and Kei Nakagawa. 2021. Deep portfolio optimization via distributional prediction of residual factors. In *AAAI'21*, Vol. 35. 213–222.
- [16] Serkan Imisiker and Bedri Kamil Onur Tas. 2018. Wash trades as a stock market manipulation tool. *Journal of Behavioral and Experimental Finance* 20 (2018), 92–98.
- [17] Narasimhan Jegadeesh and Sheridan Titman. 1993. Returns to buying winners and selling losers: Implications for stock market efficiency. *The Journal of Finance* 48, 1 (1993), 65–91.
- [18] Jiahao Ji, Jingyuan Wang, Chao Huang, Junjie Wu, Boren Xu, Zhenhe Wu, Junbo Zhang, and Yu Zheng. 2023. Spatio-Temporal Self-Supervised Learning for Traffic Flow Prediction. In *AAAI'23*.
- [19] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jiawei Jiang, and Hu Zhang. 2022. STDEN: Towards physics-guided neural networks for traffic flow prediction. In *AAAI'22*, Vol. 36. 4048–4056.
- [20] Jiahao Ji, Jingyuan Wang, Zhe Jiang, Jingtian Ma, and Hu Zhang. 2020. Interpretable spatiotemporal deep learning model for traffic flow prediction based on potential energy fields. In *ICDM'20*. IEEE, 1076–1081.
- [21] Jiawei Jiang, Chengkai Han, Wayne Xin Zhao, and Jingyuan Wang. 2023. PFDformer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. In *AAAI'23*.
- [22] Wenjun Jiang, Wayne Xin Zhao, Jingyuan Wang, and Jiawei Jiang. 2023. Continuous Trajectory Generation Based on Two-Stage GAN. In *AAAI'23*. AAAI Press.
- [23] Charles P Jones and Robert H Litzenberger. 1970. Quarterly earnings reports and intermediate stock price trends. *The Journal of Finance* 25, 1 (1970), 143–148.
- [24] Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2023. Diffusion variational autoencoder for tackling stochasticity in multi-step regression stock price prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1087–1096.
- [25] Charles MC Lee and Bhaskaran Swaminathan. 2000. Price momentum and trading volume. *The Journal of Finance* 55, 5 (2000), 2017–2069.
- [26] Chang Li, Dongjin Song, and Dacheng Tao. 2019. Multi-task recurrent neural networks and higher-order Markov random fields for stock price movement prediction: Multi-task RNN and higher-order MRFs for stock price classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1141–1151.
- [27] Shuqi Li, Weiheng Liao, Yuhuan Chen, and Rui Yan. 2023. PEN: prediction-explanation network to forecast stock price movement with better explainability. In *AAAI'23*, Vol. 37. 5187–5194.
- [28] Zhiguo Li, Derek Yang, Li Zhao, Jiang Bian, Tao Qin, and Tie-Yan Liu. [n. d.]. Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding.
- [29] Yun-Hsuan Lien, Yuan-Kui Li, and Yu-Shuen Wang. 2023. Contrastive learning and reward smoothing for deep portfolio management. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*. 3966–3974.
- [30] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. [n. d.]. Learning multiple stock trading patterns with temporal routing adaptor and optimal transport.
- [31] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuaren Liu. 2020. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *AAAI'20*, Vol. 34. 2128–2135.
- [32] Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. 2023. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. In *NeurIPS'23*.
- [33] Zehui Liu, Zimeng Li, Jingyuan Wang, and Yue He. 2024. Full Bayesian Significance Testing for Neural Networks. In *AAAI'24*, Vol. 38. 8841–8849.
- [34] Zelin Ni, Hang Yu, Shizhan Liu, Jianguo Li, and Weiying Lin. 2023. BasisFormer: Attention-based time series forecasting with learnable and interpretable basis. In *NeurIPS'23*.
- [35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv Preprint ArXiv:1807.03748* (2018).
- [36] James M Poterba and Lawrence H Summers. 1988. Mean reversion in stock prices: Evidence and implications. *Journal of Financial Economics* 22, 1 (1988), 27–59.
- [37] Houxing Ren, Jingyuan Wang, and Wayne Xin Zhao. 2022. Generative Adversarial Networks Enhanced Pre-training for Insufficient Electronic Health Records Modeling. In *KDD'22*. 3810–3818.
- [38] Houxing Ren, Jingyuan Wang, Wayne Xin Zhao, and Ning Wu. 2021. Rapt: Pre-training of time-aware transformer for learning robust healthcare representation. In *KDD'21*. 3503–3511.
- [39] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65.
- [40] Robert J Shiller. 2015. *Irrational Exuberance*. Princeton university press.
- [41] Paul Slovic. 1972. Psychological study of human judgment: Implications for investment decision making. *The Journal of Finance* 27, 4 (1972), 779–799.
- [42] Shuo Sun, Xinrun Wang, Wanqi Xue, Xiaoxuan Lou, and Bo An. [n. d.]. Mastering stock markets with efficient mixture of diversified trading Experts.
- [43] Paul C Tetlock. 2007. Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance* 62, 3 (2007), 1139–1168.
- [44] Guifeng Wang, Longbing Cao, Hongke Zhao, Qi Liu, and Enhong Chen. 2021. Coupling macro-sector-micro financial indicators for learning stock representations with less uncertainty. In *AAAI'21*, Vol. 35. 4418–4426.
- [45] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. 2022. Adaptive long-short pattern transformer for stock investment selection. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. 3970–3977.
- [46] Jingyuan Wang, Qian Gu, Junjie Wu, Guannan Liu, and Zhang Xiong. 2016. Traffic speed prediction and congestion source exploration: A deep learning method. In *ICDM'16*. IEEE, 499–508.
- [47] Jingyuan Wang, Jiahao Ji, Zhe Jiang, and Leilei Sun. 2022. Traffic Flow Prediction Based on Spatiotemporal Potential Energy Fields. *TKDE* (2022).
- [48] Jingyuan Wang, Xiaoda Wang, Chao Li, Junjie Wu, et al. 2020. Deep Fuzzy Cognitive Maps for Interpretable Multivariate Time Series Prediction. *TFS* (2020), 1–14.
- [49] Jingyuan Wang, Ze Wang, Jianfeng Li, and Junjie Wu. 2018. Multilevel wavelet decomposition network for interpretable time series analysis. In *KDD'18*. 2437–2446.
- [50] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. [n. d.]. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks.
- [51] Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. 2021. Deep-Trader: A deep reinforcement learning approach for risk-return balanced portfolio management with market conditions Embedding. In *AAAI'21*, Vol. 35. 643–650.
- [52] Ku Xu, Yifan Zhang, Deheng Ye, Peilin Zhao, and Mingkui Tan. 2020. Relation-aware transformer for portfolio policy learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. 4647–4653.
- [53] Wentao Xu, Weiqing Liu, Chang Xu, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. Rest: Relational event-driven stock trend forecasting. In *Proceedings of The Web Conference 2021*. 1–10.
- [54] Linyi Yang, Jiazheng Li, Ruihai Dong, Yue Zhang, and Barry Smyth. 2022. NumHTML: Numeric-oriented hierarchical transformer model for multi-task financial forecasting. In *AAAI'22*, Vol. 36. 11604–11612.

- [55] Mengyuan Yang, Xiaolin Zheng, Qianqiao Liang, Bing Han, and Mengying Zhu. 2022. A smart trader for portfolio management based on normalizing flows. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*.
- [56] Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. 2020. Qlib: An ai-oriented quantitative investment platform. *ArXiv Preprint ArXiv:2009.11189* (2020).
- [57] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. 2023. Frequency-domain MLPs are more effective learners in time series forecasting. In *NeurIPS'23*.
- [58] 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 International Conference on Knowledge Discovery & Data Mining*. 2037–2045.
- [59] Shuo Yu, Hongyan Xue, Xiang Ao, Feiyang Pan, Jia He, Dandan Tu, and Qing He. [n. d.]. Generating synergistic formulaic alpha collections via reinforcement learning.
- [60] J. Wang Z. Liu, Z. Li and Y. He. 2024. Full Bayesian Significance Testing for Neural Networks in Traffic Forecasting. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*.
- [61] Lifan Zhao, Shuming Kong, and Yanyan Shen. 2023. DoubleAdapt: A meta-learning approach to incremental learning for stock trend forecasting. In *Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3492–3503.
- [62] Dawei Zhou, Lecheng Zheng, Yada Zhu, Jianbo Li, and Jingrui He. 2020. Domain adaptive multi-modality neural attention network for financial forecasting. In *Proceedings of the Web Conference 2020*. 2230–2240.
- [63] Haoren Zhu, Shih-Yang Liu, Pengfei Zhao, Yingying Chen, and Dik Lun Lee. 2022. Forecasting asset dependencies to reduce portfolio risk. In *AAAI'22*, Vol. 36. 4397–4404.

A Supplemental Materials

A.1 Hedging Investment Strategy

We adopt a long-short hedging strategy as the backbone investment strategy for our forecasting model. We first give the concept of long/short position as follows.

DEFINITION 7 (LONG/SHORT POSITION). *The long position is the trading operation that buys a stock at time t_1 first and then sells it at t_2 , while a short position is the trading operation that sells an stock at t_1 first and then buys it back at t_2 .*

By taking a long position, traders expect a stock will rise in price, and the profit is $v^{(i)}(p_{t_2}^{(i)} - p_{t_1}^{(i)})$, where $v^{(i)}$ is the buying volume of the stock s_i . On the contrary, the trader's expectation in short position is that the price will drop, and the profit is $v^{(i)}(p_{t_1}^{(i)} - p_{t_2}^{(i)})$. In practice, a short position trader in the stock market borrows stocks from a broker and sells them at t_1 , then at t_2 , the trader buys the sold stocks back and returns them to the broker.

We then define the *long-short hedging* investment strategy adopted in our study, which uses long and short positions to hedge investment risks. Given a set of M tradable stocks, the strategy first ranks the stocks in descending order by their returns in period t as $s_t^{\text{rank}} = (s^{(i_1)}, \dots, s^{(i_m)}, \dots, s^{(i_M)})$, where $s^{(i_1)}$ is the largest return and $s^{(i_M)}$ is the smallest. The strategy takes long position on the stocks with top- N returns, i.e., $s_t^{\text{long}} = (s^{(i_1)}, \dots, s^{(i_N)})$ and short position on the stocks with bottom- N returns, i.e., $s_t^{\text{short}} = (s^{(i_{M-N+1})}, \dots, s^{(i_M)})$. Then the total return of the strategy in t is simply the average of the returns of all traded stocks, which is given by

$$y_t^{\text{STR}} = \frac{1}{N} \left(\sum_{m=1}^N y_t^{(i_m)} - \sum_{m=M-N+1}^M y_t^{(i_m)} \right). \quad (33)$$

Note that without of generality we here assume all stocks are invested with the same amount.

A.2 Forecasting Accuracy Metrics

- *Root Mean Square Error (RMSE):* RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{T \times I} \sum_{t=1}^T \|\hat{\mathbf{y}}_t - \mathbf{y}_t\|_2^2}, \quad (34)$$

where $\|\cdot\|_2$ is the l_2 norm, \mathbf{y}_t is the actual Rate of Return (RoR) vector at period t , $\hat{\mathbf{y}}_t$ is the predicted RoR and I is the dimension of \mathbf{y}_t .

- *Mean Absolute Error (MAE):* MAE is defined as

$$\text{MAE} = \frac{1}{T \times I} \sum_{t=1}^T |\hat{\mathbf{y}}_t - \mathbf{y}_t|, \quad (35)$$

where $|\cdot|$ is the l_1 norm.

• *Information Correlation (IC) and IC Information Ratio (ICIR):* The Information Correlation between the actual RoR vector \mathbf{y}_t and the predicted RoR vector $\hat{\mathbf{y}}_t$ is calculated by Eq. (30). We evaluate the prediction performance of our model using the average IC for all periods in the test set. Based on the IC, we define the IC Information Ratio (ICIR), which is obtained by dividing the average IC by its standard deviation.

• *Rank Information Correlation (RankIC) and RankIC Information Ratio (RankICIR):* RankIC is computed by replacing $\hat{\mathbf{y}}$ and \mathbf{y} with RoR rank vectors, i.e., $\hat{\mathbf{z}}$ and \mathbf{z} . Similarly, the RankICIR is obtained by dividing the average RankIC by its standard deviation. Compared to IC, RankIC ignores the numerical error of RoR, which only relates to the correctness of the ranking.

A.3 Complete Training Procedure

Alg. 1 shows our complete training procedure. In *UMI*, we first define $\mathbf{p}_{t-1}^{(i)} = (p_1^{(i)}, \dots, p_{t-1}^{(i)})$ and $P_{t-1} = \{\mathbf{p}_{t-1}^{(i)}\}_{i=1}^I$. We then train the stock-level irrationality factor learning module using the cointegration objective functions \mathcal{L}_S . Subsequently, a sequence of stock-level irrationality factors is derived from a given sequence of stock prices, where

$$\mathbf{u}_{t-1}^{(i)}(\Sigma) = (u_1^{(i)}(\Sigma), \dots, u_{t-1}^{(i)}(\Sigma)), \quad (36)$$

and

$$\mathbf{U}_{t-1}(\Sigma) = (\mathbf{u}_{t-1}^{(1)}(\Sigma), \dots, \mathbf{u}_{t-1}^{(i)}(\Sigma), \dots, \mathbf{u}_{t-1}^{(I)}(\Sigma)). \quad (37)$$

Next, the market-level irrational factor learning module is trained with the joint self-supervised market learning loss $\mathcal{L}_M = \mathcal{L}_C + \lambda_2 \mathcal{L}_P$. This allows us to derive a market-level irrational factor \mathbf{m}_{t-1} from \mathcal{E}_{t-1} , which encompasses historical features of various stocks.

Lastly, the forecasting module is trained using the forecasting loss \mathcal{L} as defined in Eq. (31). Subsequently, the forecasted stock return is then generated through a combination of a Transformer encoder, a stock relation dependency module and a MLP neural network.

Algorithm 1 Factor-learning and Forecasting-training Procedure

```

Initializing parameters  $B, W_C$  for stock-level irrationality factor learning.
for episode = 1, 2, ... do
    Optimizing loss  $\mathcal{L}_S$  in training dataset.
end for
 $U_{t-1} \leftarrow \text{Cointegration}(P_{t-1})$ .
Initializing parameters  $W_S, b, W_t, w_\eta, w_M, b_M$  for market-level irrationality factor learning.
for episode = 1, 2, ... do
    Optimizing loss  $\mathcal{L}_M = \mathcal{L}_C + \lambda_2 \mathcal{L}_P$  in training dataset.
end for
 $m_{t-1} \leftarrow \text{Market}(\mathcal{E}_{t-1})$ .
for episode = 1, 2, ... do
    Optimizing loss  $\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_3 \mathcal{L}_{\text{RankIC}}$  in training dataset with stock-level irrationality factors  $u$  and market-level irrationality factors  $m$ .
end for
 $C_{t-1} \leftarrow \text{Transformer}(u_{t-1}, \mathcal{E}_{t-1})$ .
 $D_{t-1} \leftarrow \text{Dependency}(C_{t-1}, \mathcal{E}_{t-1})$ .
 $\hat{y}_t \leftarrow \text{MLP}(C_{t-1} \parallel D_{t-1} \parallel m_{t-1})$ .
return  $\hat{y}_t$ 

```

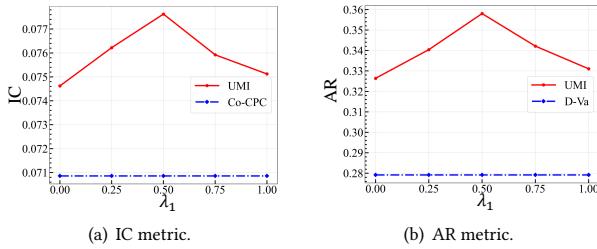


Figure 5: Parameter sensitivity of λ_1 on CN markets for two metrics.

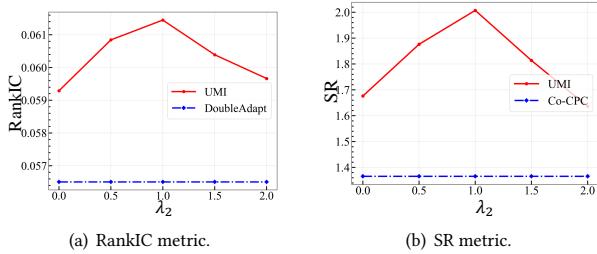


Figure 6: Parameter sensitivity of λ_2 on US markets for two metrics.

A.4 Additional Universality of Irrationality Factors

Table 3 shows additional universality experiment results with forecasting performance. Analogous to Table 2, the forecasting performance of all models undergoes notable enhancement with the

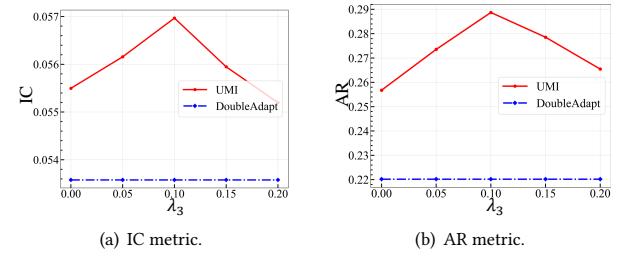


Figure 7: Parameter sensitivity of λ_3 on US markets for two metrics.

Table 3: The universality of the irrationality factors with forecasting performance.

US Market							
Metrics	DA+F	Imp.	D-Va+F	Imp.	CPC+F	Imp.	UMI
RMSE(\downarrow)	.0395	1.0%	.0395	0.8%	.0394	1.6%	.0394
MAE(\downarrow)	.0231	1.2%	.0232	1.1%	.0232	0.8%	.0231
IC(\uparrow)	0.056	4%	0.054	11%	0.054	4%	0.057
ICIR(\uparrow)	0.636	2%	0.603	9%	0.626	6%	0.663
RankIC(\uparrow)	0.057	2%	0.058	4%	0.062	14%	0.061
RankICIR(\uparrow)	0.642	1%	0.637	5%	0.697	16%	0.666

CN Market							
Metrics	DA+F	Imp.	D-Va+F	Imp.	CPC+F	Imp.	UMI
RMSE(\downarrow)	.0351	0.9%	.0352	1.2%	.0351	1.0%	.0350
MAE(\downarrow)	.0223	0.8%	.0223	1.0%	.0225	1.1%	.0222
IC(\uparrow)	0.073	6%	0.074	5%	0.074	5%	0.078
ICIR(\uparrow)	0.776	6%	0.798	7%	0.803	5%	0.831
RankIC(\uparrow)	0.072	5%	0.069	3%	0.072	3%	0.073
RankICIR(\uparrow)	0.759	4%	0.733	3%	0.762	3%	0.777

incorporation of the irrationality factors. This underscores the proficiency of our irrationality factors in harnessing valuable insights that may have been overlooked by the baselines.

A.5 Parameter Sensitivity

Fig. 5 reports the sensitivity of two evaluating metrics to the parameter λ_1 in Eq. (14) on the CN market. We also incorporate the best baseline methods on CN dataset for comparison. Here, we vary λ_1 in the set {0, 0.25, 0.5, 0.75, 1}. As shown in the figures, our model is relatively stable and consistently competitive with the best baseline methods.

Fig. 6 reports the sensitivity of two evaluating metrics to the parameter λ_2 in Eq. (24) on the US market. We also incorporate the best baseline methods on the US dataset for comparison. Here, we vary λ_2 in the set {0, 0.5, 1, 1.5, 2}.

Fig. 7 reports the sensitivity of two evaluating metrics to the parameter λ_3 in Eq. (31) on the US market. We also incorporate the best baseline methods on the US dataset for comparison. Here, we vary λ_3 in the set {0, 0.05, 0.1, 0.15, 0.2}. The sensitivities of other parameters are similar for the three parameters.