Pre-training Time Series Models with Stock Data Customization

Mengyu Wang
University of Edinburgh
School of Informatics
Edinburgh, United Kingdom
mengyu.wang@ed.ac.uk

Tiejun Ma
University of Edinburgh
School of Informatics
Edinburgh, United Kingdom
tiejun.ma@ed.ac.uk

Shay B. Cohen
University of Edinburgh
School of Informatics
Edinburgh, United Kingdom
scohen@inf.ed.ac.uk

arXiv:2506.16746v1 [cs.CE] 20 Jun 2025

Abstract

Stock selection, which aims to predict stock prices and identify the most profitable ones, is a crucial task in finance. While existing methods primarily focus on developing model structures and building graphs for improved selection, pre-training strategies remain underexplored in this domain. Current stock series pre-training follows methods from other areas without adapting to the unique characteristics of financial data, particularly overlooking stock-specific contextual information and the non-stationary nature of stock prices. Consequently, the latent statistical features inherent in stock data are underutilized. In this paper, we propose three novel pre-training tasks tailored to stock data characteristics: stock code classification, stock sector classification, and moving average prediction. We develop the Stock Specialized Pre-trained Transformer (SSPT) based on a two-layer transformer architecture. Extensive experimental results validate the effectiveness of our pre-training methods and provide detailed guidance on their application. Evaluations on five stock datasets, including four markets and two time periods, demonstrate that SSPT consistently outperforms the market and existing methods in terms of both cumulative investment return ratio and Sharpe ratio. Additionally, our experiments on simulated data investigate the underlying mechanisms of our methods, providing insights into understanding price series. Our code is publicly available at: https://github.com/astudentuser/Pre-training-Time-Series-Models-with-Stock-Data-Customization.

CCS Concepts

• **Applied computing** → **Economics**; **Forecasting**.

Keywords

Stock Prediction, Time Series Pre-training, Representation Learning

ACM Reference Format:
Mengyu Wang, Tiejun Ma, and Shay B. Cohen. 2025. Pre-training Time Series Models with Stock Data Customization. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737005

1 Introduction

Stock selection is a critical aspect of investment decision-making in the vast stock market [15, 43]. Predicting stock trends to identify the most profitable investment opportunities has become a popular research topic [18, 41, 62]. Although financial time series are volatile, they are not entirely random, persistent inefficiencies in markets lead to exploitable patterns [23, 33, 50]. Stock movements are driven by a range of economic and behavioral factors, including volatility dynamics, momentum effects, and sector-specific trends [12, 38]. These factors introduce subtle yet recurring signals in the data, which cannot be adequately captured through supervised learning alone. Many advanced methods in financial area are fundamentally focused on uncovering such latent structures to enhance predictive performance [18, 58]. Thus, the belief that financial markets contain learnable, non-random signals is a foundational assumption shared across both academic and applied finance communities.

Traditional approaches often rely on time-series analysis models to evaluate stock price data [1, 56]. With the advent of deep learning, neural networks have shown promising capabilities in analyzing historical price series [7, 11, 26].

However, stock selection poses greater challenges compared to general time-series tasks. Stock prices exhibit high volatility due to numerous influencing factors in the real market, introducing additional complexities for prediction [9, 14]. Therefore, recent studies have focused on extracting more robust profit-related information from stock prices by building connections among different stocks [18], mining spatio-temporal information [43, 54], and studying multi-scale patterns [48].

While existing works have highlighted the importance of considering market interactions, pre-training, a widely used representation learning technique in other fields like natural language processing (NLP) and computer vision (CV), has been less explored for stock data. Current pre-training on stock data primarily focuses on two directions: contrastive learning [20] and masked value prediction [54]. However, both approaches have limitations in adapting to stock data. Contrastive learning methods typically have more stringent data requirements, such as minute-level or second-level prices, which are impractical in many situations. The masked value prediction methods, following general time-series pre-training approaches [60], do not adapt well to stock data, due to the non-stationary nature of stock prices and the complex market dependencies [46]. Therefore, there exists a gap in exploring pre-training methods for stock data to fully extract predictive power from price series.

Designing customized pre-training methods for stock data has the potential to improve stock selection because the unique characteristics of stock data, such as volatility and many other statistical features, are likely underutilized. Since profit is the primary objective of stock prediction tasks, most training objectives are directly related to price changes. However, pre-training research [10, 67] and exploration of multi-task learning [65] in other areas have

This work is licensed under a Creative Commons Attribution 4.0 International License.
KDD '25, Toronto, ON, Canada
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737005

shown that a specific task can benefit from the features learned from other tasks, which cannot be fully used by the task itself. Therefore, exploring other training objectives for stock data is promising to extract more information from stock prices.

In this paper, we propose three specialized pre-training tasks, stock code classification, stock sector classification, and moving average prediction. These tasks require only easily accessible daily price data and basic company information. We implement these tasks using a simple two-layer transformer architecture, which we call the Stock Specialized Pre-trained Transformer (SSPT). Experiments demonstrate that these pre-training tasks effectively capture stock price characteristics and improve stock selection performance, enabling SSPT to outperform existing methods.

The first two tasks, stock code classification and stock sector classification, are designed to identify the stock or sector to which a given price series slice belongs (as illustrated in Figure 1 with an example of stock code classification). While these tasks focus on objectives not directly related to profits, they help capture unique characteristics of price series. Experiments demonstrate that these two tasks effectively capture distinctive price patterns and benefit stock selection, enabling SSPT to outperform existing methods.

The third task, moving average prediction, adapts the widely used masked value prediction idea to stock data. The non-stationary nature of stock prices indicates the presence of random content in price changes, making it impossible to accurately predict each price [49]. Traders use technical indicators like moving average values to obtain relatively stable price features. Inspired by this, we propose predicting the average values from a period after masking some prices, rather than predicting specific masked values.

We evaluate our approach on data from NASDAQ, NYSE, and TOPIX-100 markets over a five-year period, following recent stock selection studies [18, 48, 54]. We also test SSPT on FTSE-100 and recent NASDAQ data, to demonstrate our methods can generalize across markets and time periods. Results show that our pre-training methods enhance cumulative investment return ratio (IRR) and Sharpe ratio (SR), outperforming both market benchmarks and existing methods. We provide detailed analyses of pre-training settings and fine-tuning strategies. Additionally, we conduct experiments on simulated data to explore the reasons for the effectiveness of our methods. In summary, our main contributions are:

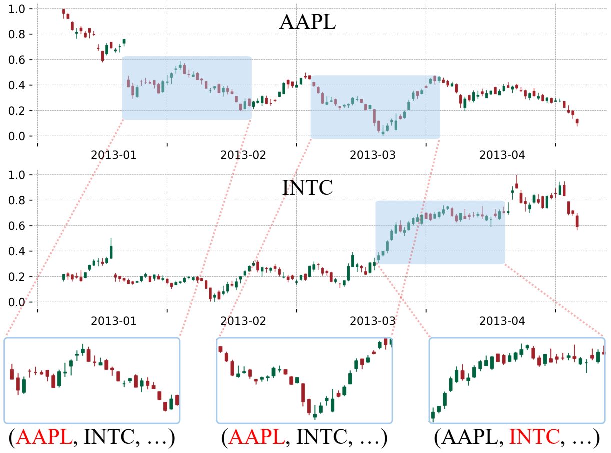

Figure 1: An example of stock code classification. This task explores whether price series slices from different stocks contain distinguishing features. All prices are normalized to the range of 0 to 1.

(1) We propose three customized pre-training tasks for stock data: stock code classification, stock sector classification, and moving average prediction. We demonstrate their effectiveness in enhancing stock selection and provide guidelines for their use.
(2) We introduce the Stock Specialized Pre-trained Transformer (SSPT), a simple yet effective model based on our pre-training methods. SSPT outperforms market benchmarks and existing methods on IRR and SR across five datasets spanning different markets and time periods.
(3) Through ablation studies and simulations, we investigate why our pre-training tasks are effective, highlighting underutilized information in stock prices and providing insights into price series analysis.

2 Related Work

2.1 Stock Prediction and Selection

Stock prediction has been a longstanding area of research. Early works developed statistical models, such as the ARIMA model, for technical analysis and price prediction [49]. In addition, machine learning models like Hidden Markov Models (HMM) and Support Vector Machines (SVM) have been explored for this purpose [24, 40]. With the rise of deep learning, neural networks, particularly recurrent neural networks (RNNs) and their variants, have achieved great success in stock forecasting [2, 42, 61]. Later, the transformer structure revolutionized the field of deep learning, and these models have been widely adopted for stock prediction tasks [11, 30, 45].

Recently, researchers have recognized the importance of considering market relationships in identifying profitable opportunities, moving beyond treating stocks independently [7]. This shift has led to the formulation of the stock selection task, where the goal is to predict multiple stocks collectively and select the most profitable ones. A benchmark dataset was created to facilitate this task [18]. Graph Neural Networks (GNNs) have been applied from various angles to analyze the spatio-temporal dynamics of stock markets. For instance, some studies have explored hypergraph-based models [43], while others focused on adaptive price patterns [48]. These methods have demonstrated effectiveness on large stock markets such as NASDAQ and NYSE.

However, while existing approaches have extensively explored model architectures to enhance stock analysis, the training objectives and stock feature representations have received limited attention. Most models are trained to minimize movement prediction errors, potentially leading to the underutilization of stock data information. Although features closely related to profitability are well-considered, other features that may seem less directly relevant to profits are likely to be overlooked. These overlooked features may hold potential to improve the understanding of price series and enhance prediction performance.

2.2 Neural Networks Pre-training

Numerous studies have demonstrated the effectiveness of pre-training across various domains, from NLP, CV to Vision-and-Language (VL)

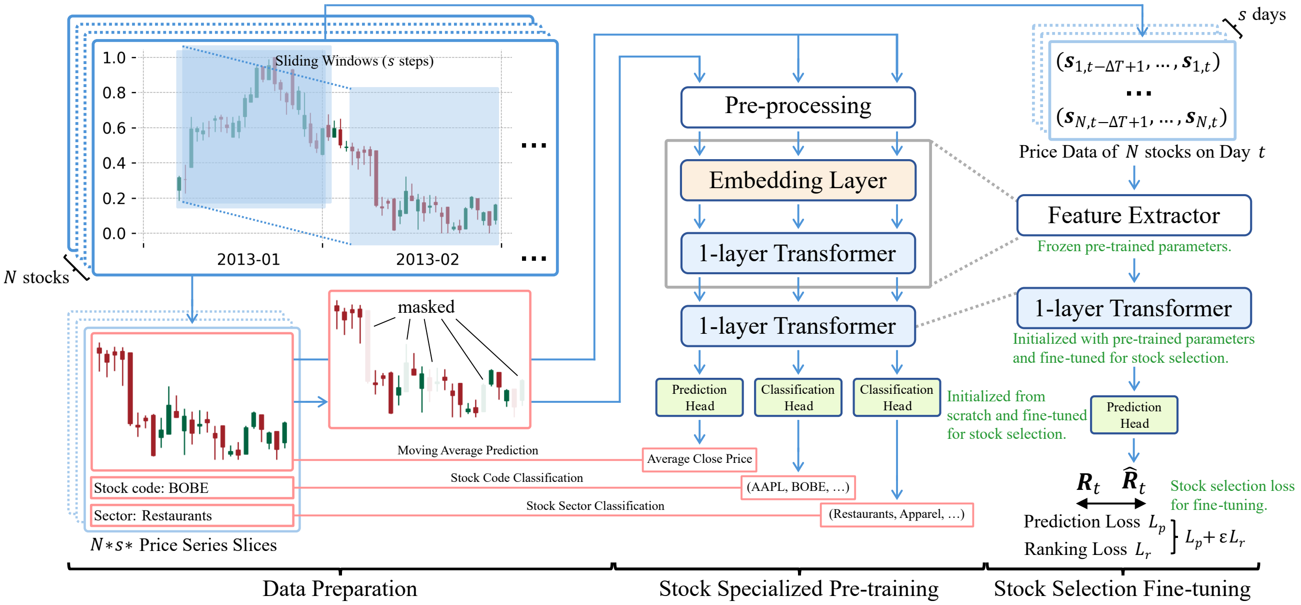
Sliding Windows (s steps)
1.0
0.8
0.6
0.4
0.2
0.0
2013-01
2013-02
...
N stocks
masked
Stock code: BOBE
Sector: Restaurants
$N*s*$ Price Series Slices
Moving Average Prediction
Stock Code Classification
Stock Sector Classification
Pre-processing
Embedding Layer
1-layer Transformer
1-layer Transformer
Prediction Head
Classification Head
Classification Head
Average Close Price
(AAPL, BOBE, ...)
(Restaurants, Apparel, ...)
s days
$(s_{1,t-\Delta T+1}, \dots, s_{1,t})$
$(s_{N,t-\Delta T+1}, \dots, s_{N,t})$
Price Data of N stocks on Day t
Feature Extractor
Frozen pre-trained parameters.
1-layer Transformer
Initialized with pre-trained parameters and fine-tuned for stock selection.
Initialized from scratch and fine-tuned for stock selection.
Prediction Head
R_t $\hat{R}_t$
Stock selection loss for fine-tuning.
Prediction Loss L_p
Ranking Loss L_r
$L_p + \varepsilon L_r$
Data Preparation
Stock Specialized Pre-training
Stock Selection Fine-tuning

Figure 2: Overview of the pre-training and fine-tuning procedures for our SSPT model. The frozen parameters can be adjusted, and we conduct experiments to explore the impact of freezing different model components in Section 6.1.

tasks [5, 10, 67]. Pre-training has been shown to benefit downstream tasks by facilitating the learning of high-quality features [21, 32, 52]. Inspired by successes in NLP and CV, pre-trained models for time series analysis have also gained attention in recent research [60, 63, 66]. These works adapt pre-training methods, such as masked value prediction, which are commonly used in NLP and CV, to time series data for enhanced feature extraction.

As a specialized type of time series data, stock prices have also been explored using pre-training methods, primarily in two directions: contrastive learning [20] and masked value prediction [54]. The former aims to improve feature representation by analyzing multi-granularity data, while the latter simulates the masked value prediction strategies from other fields to learn contextual information. However, both approaches are largely transplanted from other domains without adequately considering the unique characteristics of stock data. Contrastive learning methods impose stricter requirements on data and continue to focus solely on time series analysis. Meanwhile, masked value prediction methods struggle to adapt effectively to stock data due to the non-stationary nature of stock prices, which makes precise prediction of specific values impractical [48]. Therefore, there remains a need for stock data-specific pre-training tasks to improve price feature learning.

3 Problem Formulation

To ensure a fair comparison with advanced methods, we follow the stock selection formulation presented in previous works [18, 43, 48, 54]. Let $\mathcal{S} = \{s_1, s_2, ..., s_N\}$ denotes a set of N stocks. For a given trading day t, each stock $\mathbf{s}_i$ is associated with M price features over the past ΔT days, denoted as $\mathbf{X}_{i,t} = \{\mathbf{x}_{i,t-\Delta T+1}, ..., \mathbf{x}_{i,t}\} \in \mathbb{R}^{M\times\Delta T}$. Among the M features of $\mathbf{x}_{i,t}$, there exists a closing price $p_{i,t}$, from which we can calculate the 1-day return ratio as $r_{i,t} = \dfrac{p_{i,t+1} - p_{i,t}}{p_{i,t}}$. The return ratio $r_{i,t}$ serves as our prediction target, and the predicted return ratios for all stocks on day t are denoted by $\hat{\mathbf{R}}_t = (\hat{r}_{1,t}, ...\hat{r}_{N,t})$. Based on $\hat{\mathbf{R}}_t$, we rank the stocks in descending order of return ratios and select the top-ranked stocks for investment on trading day t. Our goal is to build a model $f(\cdot\,; w_f)$ with parameters w_f that predicts the return ratios $\hat{r}_{i,t} = f(\mathbf{X}_{i,t}; w_f)$ and select the most profitable stocks for investment.

4 Methodology

To acquire high-quality stock price representations, we propose three customized pre-training tasks that leverage the unique characteristics of financial market dynamics. Specifically, we design two classification tasks, stock code classification and stock sector classification, which use stock-specific contextual information. Additionally, we introduce a moving average prediction task that considers the non-stationary nature of stock prices. We use a standard two-layer transformer architecture and pre-train the model using these three tasks. Following pre-training, we freeze a subset of the model parameters to retain the acquired knowledge, and fine-tune the remaining parameters for the stock selection task. Figure 2 provides an overview of our pre-training tasks and the stock selection framework.

4.1 Stock Code/Sector Classification (SCC/SSC)

Each stock is unique, corresponding to individual companies with distinct backgrounds, environments, business models, and other characteristics that lead to different reactions to market events. While the Efficient Market Hypothesis (EMH) [13] suggests that a completely efficient market would immediately reflect all such information and leave no room for prediction, practical markets are rarely fully efficient. Numerous studies have demonstrated that stock markets can be predictable to some extent [8, 34, 64], indicating the existence of exploitable patterns and market inefficiencies that are not yet fully incorporated into prices.

Based on this observation, we hypothesize that specific patterns distinguishing different stocks must exist. Although these patterns may not directly correlate with profits, they could reveal hidden features in price series beneficial for stock prediction. To explore this hypothesis, we design a stock code classification task. As illustrated in Figure 2, we segment price series into equal-length slices (matching the look-back period used for subsequent stock selection) and mix slices from different stocks. The model is then trained to identify the source stock of each slice. Using the notation from Section 3 and denoting the additional classification head parameters as w_{scc}, we train the model f with the cross-entropy loss for stock code classification, $\mathcal{L}_{scc}$, calculated as:

$$\mathcal{L}_{scc} = -\log(f(\mathbf{X}_{i,t}; w_f, w_{scc})_i). \tag{1}$$

Note that the value of i is only contained in our annotation $\mathbf{X}_{i,t}$, not in the input value itself.

While stocks are unique, they are also interconnected. Market events can affect multiple related stocks simultaneously, and stocks can influence each other. Companies within the same sector often exhibit similar behavior during major market events due to their related businesses. Therefore, we hypothesize that there exist distinct patterns characterizing different sectors and design a stock sector classification task. Similar to the stock code classification, we mix price series slices and train a model to identify the sector of origin for each slice. With $sec(i)$ denoting the sector that stock s_i belongs to and w_{ssc} representing the added classification parameters, we train the model with the loss of stock sector classification $\mathcal{L}_{ssc}$ as:

$$\mathcal{L}_{ssc} = -\log(f(\mathbf{X}_{i,t}; w_f, w_{ssc})_{sec(i)}). \tag{2}$$

Both pre-training tasks use stock-specific information (stock code and sector) to identify distinguishing patterns in price series. While price series segments often exhibit similarity and complexity that challenge effective classification, our results support the existence of discernible patterns that can differentiate stock codes and sectors. Figure 3 presents the classification accuracy for both tasks on NASDAQ and NYSE datasets, comparing random guesses, classifications based solely on close prices, and classifications based on all daily price values. Although the highest classification accuracies only reach approximately 0.2, indicating the difficulty of accurately predicting stock source information, these post-training accuracies significantly surpass random guessing. Similar patterns emerge in the other three datasets, as shown in Appendix Section B. This outcome demonstrates the presence of identifiable features within stock price series, which we expect will contribute to improved stock selection performance.

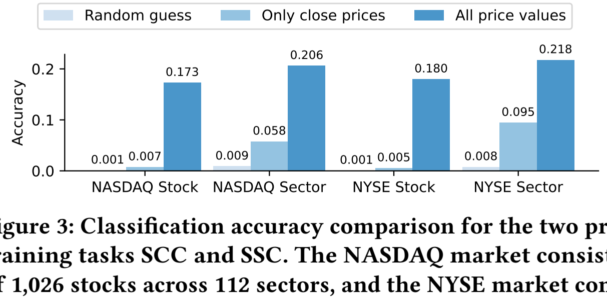

Figure 3: Classification accuracy comparison for the two pre-training tasks SCC and SSC. The NASDAQ market consists of 1,026 stocks across 112 sectors, and the NYSE market comprises 1,737 stocks in 129 sectors. All results are rounded to three decimal places.

4.2 Moving Average Prediction (MAP)

Stock prices exhibit high volatility and contain significant random noise [49]. Moreover, stock price series are non-stationary, meaning their statistical properties change over time. These characteristics make precise price prediction extremely challenging. When a model achieves very low prediction errors on training and validation sets, it likely indicates overfitting to random fluctuations rather than capturing meaningful patterns.

Despite these challenges, predicting masked prices remains the primary pre-training technique in existing stock prediction methods [54]. Masked value prediction, a widely used pre-training strategy across various fields, has proven effective in predicting masked words in sentences [10] and masked image patches [19]. Consequently, it has been adopted in stock prediction. However, as previously noted, the unique characteristics of price series limit the effectiveness of direct masked value prediction in the stock market context.

To address these limitations, we propose an adaptation of masked value prediction for stock data, inspired by Moving Average (MA) indicators. MA is a common technical indicator in financial analysis that smooths short-term price fluctuations [4, 39]. By calculating the average price over a specified period, MA provides a more stable signal compared to raw prices and is widely used by traders for trend analysis. Therefore, since predicting individual prices is unreliable, we propose an alternative approach: predicting the moving average values of a period containing masked values.

As illustrated in Figure 2, we calculate the average closing prices within sliding windows. We then mask a portion of the prices within each window and train the model to predict these average prices from the partially masked data. Let w_{map} represents the additional parameters of the prediction head. The model is trained with the loss $\mathcal{L}_{map}$, defined as:

$$\mathcal{L}_{map} = \left(f(\mathbf{X}_{i,t}; w_f, w_{map}) - \frac{1}{\Delta T} \sum_{j=1}^{\Delta T} p_{i,t-j+1} \right)^2. \tag{3}$$

This approach mitigates the impact of price volatility and non-stationarity, providing a more robust pre-training task that better aligns with the inherent characteristics of stock data.

4.3 Pre-training

We implement our pre-training methods and subsequent stock selection task using a standard two-layer transformer architecture. Apart from the classification and prediction heads necessary for different tasks, we do not add any extra specialized structures. This general model structure, combined with our pre-training methods, forms our Stock Specialized Pre-trained Transformer (SSPT), which proves effective in stock selection.

The basic approach to using these three tasks involves training a model on each task individually, followed by fine-tuning for the stock selection task. However, since the three pre-training tasks are designed to capture different aspects of stock information, it is likely that the model acquires diverse knowledge from each task. Therefore, combining these three tasks during pre-training has the potential to further enhance stock selection performance.

To maximize the benefits of our customized stock pre-training tasks, we explore pre-training the model on multiple tasks simultaneously. Although various combination methods exist, such as feature fusion or sequential pre-training through continual learning, we focus on multi-task pre-training in this paper, following popular approaches in the NLP field [10]. This involves incorporating multiple task heads into the model and optimizing their corresponding losses simultaneously.

Formally, let $\mathcal{L}_{pt}$ denote the total loss during pre-training, calculated as follows:

$$\mathcal{L}_{pt} = \alpha\mathcal{L}_{scc} + \beta\mathcal{L}_{ssc} + \gamma\mathcal{L}_{map}. \quad (4)$$

Here, α, β, and γ are coefficients that control the relative influence of each corresponding task's loss during pre-training. We conduct experiments with various combinations of these coefficients to explore the balance among the three tasks, as detailed in Section 6.2.

4.4 Fine-tuning

Following pre-training, we adapt the model for stock selection by replacing task-specific output layers with a profit ratio prediction head. In line with established approaches [18, 43, 54], we incorporate both profit ratio regression loss and profit ranking loss during fine-tuning. The stock selection loss $\mathcal{L}_{ft}$ is calculated as:

$$\mathcal{L}_{ft} = \sum_{i=1}^{N}(\hat{r}_{i,t} - r_{i,t})^2 + \epsilon\sum_{i=1}^{N}\sum_{j=1}^{N}\max(0, -(\hat{r}_{i,t} - \hat{r}_{j,t})(r_{i,t} - r_{j,t})). \quad (5)$$

where ϵ is a hyper-parameter balancing the two loss components.

During fine-tuning, the model's parameters are categorized into three groups. The parameters of the initial layers are kept frozen to preserve the pre-trained knowledge, following common practices in pre-training methods [10, 19]. The intermediate layers, between the frozen layers and the prediction head, inherit their values from the pre-trained model but are fine-tuned for the stock selection task. Finally, the prediction head, which was not involved in the pre-training tasks, is initialized from scratch and trained specifically for stock selection.

This approach, shown in Figure 2, enables flexible fine-tuning strategies. As the categorization of parameters can significantly impact performance [37], we explore various parameter arrangements to optimize stock selection effectiveness, as detailed in Section 6.1.

5 Experiment Settings

5.1 Data

We evaluate our approach using historical stock price data from five markets: NASDAQ (2013-2017), NYSE (2013-2017), FTSE-100 (2013-2017), TOPIX-100 (2016-2020), and NASDAQ-recent (2018-2022). The NASDAQ, NYSE, and TOPIX-100 datasets serve as established benchmarks in stock selection research [18], and numerous studies in this field have used them to ensure comparable results [43, 48, 54]. We additionally include FTSE-100 and NASDAQ-recent datasets to evaluate our methods across different markets and time periods. Detailed information of datasets are provided in Appendix Section A. In line with previous studies, we chronologically partition the data into training (3 years), validation (1 year), and testing (1 year) sets. The historical data comprises five daily values: open price, high price, low price, close price, and trading volume. Following previous works, we augment these values with 5, 10, 20, and 30-day moving averages. All features undergo min-max normalization based solely on training data statistics.

5.2 Evaluation Metrics

We assess model performance using a daily buy-hold-sell trading strategy, measuring the cumulative Investment Return Ratio (IRR) and Sharpe ratio (SR), consistent with standard practice in related literature [18, 43, 48, 54]. The strategy involves buying the top k stocks based on model predictions and selling them at the next day's close.

Formally, the IRR on day t is calculated as $\text{IRR}_t = \sum_{i\in\hat{\mathcal{S}}_t} r_{i,t}$, where $\hat{\mathcal{S}}_t$ represents the selected stock set on day t. The SR measures the risk-adjusted return of a portfolio and is calculated as $\text{SR} = \dfrac{E(R_p) - R_f}{\text{std}(R_p)}$, where R_p is the profit over the tested period, $E(R_p)$ and $\text{std}(R_p)$ denote the expectation and standard deviation of the profits, and R_f means the risk-free profit. These two metrics are widely used in financial prediction studies [51, 57, 59]. Previous research [18] has explored the impact of varying the value of k and identified $k = 5$ as the most representative for performance assessment. Most subsequent works report results based on this $k = 5$ setting, and we also adhere to it.

5.3 Baseline Methods

We conduct a comprehensive comparisons with various baseline methods, including recent SOTA methods. The baselines are categorized into four groups: classification (CLF) models, regression (REG) models, reinforcement learning (RL) methods, and ranking (RAN) approaches. Table 2 provides brief descriptions of these methods.

5.4 Implementation Details

Our SSPT model uses a standard two-layer transformer architecture. The embedding layer consists of a feature embedding layer and a positional embedding layer. Both the task heads for pre-training and the stock selection prediction head are implemented as fully connected layers with corresponding output dimensions. We set the hidden size of attention vectors to 32 and use 4 attention heads. For other intermediate features, we use a hidden size of 128. Following previous studies, we select the look-back length from the set $\{16, 32\}$. The model is trained using the Adam optimizer, with the learning rate chosen from $\{10^{-3}, 10^{-4}, 10^{-5}\}$. The hyper-parameter ϵ, which balances the prediction and ranking losses, is selected from $\{1, 5, 10\}$. To facilitate ranking loss computation, the batch size is set equal to the number of stocks N. Both pre-training and fine-tuning phases are limited to 100 epochs, with the optimal epoch and hyper-parameter combination determined based on validation set performance. The model training process, including both pre-training and fine-tuning, is conducted using the training and validation sets, while the reported results reflect the stock selection performance on the testing split.

6 Results and Discussion

Our analysis aims to provide comprehensive insights and practical guidelines for effectively implementing our customized stock pre-training tasks, rather than merely demonstrating performance improvements. Therefore, we structure our investigation around three research questions (RQs):

RQ1 How can we maximize the benefits of stock pre-training for the subsequent stock selection task?

RQ2 How does our stock selection framework SSPT compare to existing approaches?

RQ3 Why do these pre-training tasks improve stock selection performance?

Section 6.1 and Section 6.2 address **RQ1** from two perspectives: optimizing the performance of individual pre-training tasks and effectively combining multiple pre-training tasks. Section 6.3 tackles **RQ2** through comprehensive baseline comparisons. Section 6.4 answers **RQ3** through experiments on controlled simulated data.

While we present results across five datasets for baseline comparisons in Section 6.3, our detailed analysis in other sections primarily focuses on the NASDAQ dataset. We present SR results when analyzing pre-training tasks, as the analysis based on SR or IRR can reflect similar conclusions. However, SR is considered the more important metric as it incorporates not only returns but also risks, which are crucial factors in market investment [51]. We also include the corresponding IRR results and analysis in Appendix Section D.

6.1 Analysis of Individual Pre-training Tasks

Our analysis of individual pre-training tasks reveals two critical hyper-parameters: input features and learning rate. As presented in Figure 3 (Section 4.1), incorporating all daily price values as input features substantially improves the accuracy for the two classification tasks, compared to using only close prices. The MAP task exhibits a similar trend, with the optimized mean squared error (MSE) decreasing from 5.5×10^{-6} when using only closing prices to 2.3×10^{-6} when using all price values. This consistent finding across tasks highlights the importance of using comprehensive price information during pre-training.

However, the optimal learning rates vary across tasks. As shown in Figure 4, the pre-training tasks demonstrate sensitivity to the learning rate, with the classification tasks (SCC and SSC) performing best with a higer learning rate (10^{-3}), while the value prediction task (MAP) requires a lower rate (10^{-4}).

After optimizing the pre-trained models, we explore effective fine-tuning strategies for stock selection. A crucial aspect during fine-tuning is the management of model parameters. While the added task-specific prediction head is a confirmed component, the frozen and fine-tuned parameters can be flexibly adjusted. Although Figure 2 shows an example of freezing the entire feature extractor, this may not be optimal, as research in NLP has shown that parameter freezing strategies significantly impact performance [37].

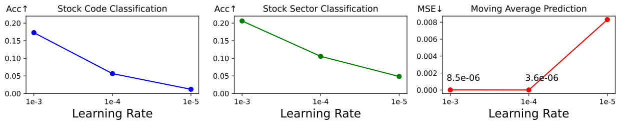

Figure 4: Evaluation results of the three pre-training tasks at varying learning rates. Note that the metrics differ among tasks: higher accuracy is better, whereas lower MSE is better.

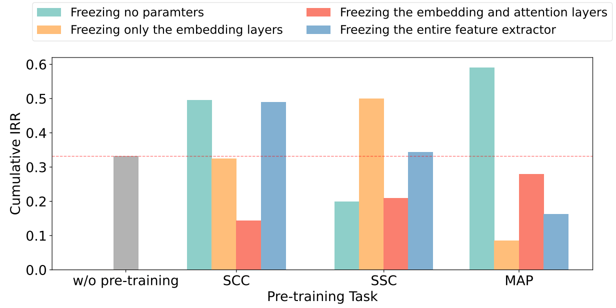

Figure 5: The Sharpe ratio results of the stock selection task with different fine-tuning strategies. The group 'w/o pre-training' presents the result of directly training the model on the stock selection task without any pre-training.

We evaluate four fine-tuning strategies: (1) freezing no parameters, (2) freezing only the embedding layers, (3) freezing the embedding and attention layers, and (4) freezing the entire feature extractor. Note that freezing no parameters is equivalent to using the pre-trained parameters to initialize the model for stock selection. Additionally, the third strategy differs from the fourth due to an additional feed-forward layer after the attention layer not shown in Figure 2.

The results of these strategies are presented in Figure 5, where we include results of direct training without pre-training as a control group. The three pre-training tasks show distinct responses to fine-tuning strategies. SCC and SSC generally improve stock selection performance across most strategies, with SCC performing best when no parameters are frozen, and SSC favoring freezing only the embedding layer. However, the MAP task shows high sensitivity to the fine-tuning strategy. It can significantly enhance performance when no parameters are frozen but harm the subsequent task when using other strategies. This suggests that MAP provides effective model initialization for stock selection but cannot directly transfer its learned knowledge. This phenomenon can be attributed to the structural similarity between MAP and stock selection tasks. Since both tasks involve price-based value regression, they likely train the model to extract similar features for different targets. Consequently, MAP can establish a favorable initial model for stock selection, but its learned features cannot be directly transferred to the target task.

Furthermore, the comparison with the control group serves as an ablation study, confirming that our three pre-training tasks are indeed beneficial for the subsequent stock selection task.

In summary, through the analysis of individual pre-training tasks, we identify the most influential factors: pre-training feature selection, learning rate, and fine-tuning strategy. Additionally, an ablation study comparing traditional masked value prediction with our proposed MAP approach, along with an analysis of different mask rates, is presented in Appendix Section C. Based on these results, we recommend the following guidelines: (1) include the complete set of price features during pre-training, (2) use a learning rate of 10^{-3} for SCC and SSC but 10^{-4} for MAP during pre-training, and (3) during fine-tuning, do not freeze any parameters for SCC and MAP but freeze the embedding layer for SSC.

Table 1: Evaluation results of pre-training with task combinations. The first row presents the results of pre-training with individual tasks. The following four rows show the results when the loss coefficients of the three tasks are equal, meaning $\alpha = \beta = \gamma = 1$ in Equation 4.

Pre-training Tasks	Acc of SCC	Acc of SSC	MSE of MAP
SCC / SSC / MAP	0.173	0.206	8.5e-6
SCC + SSC	0.129	0.124	\
SCC + MAP	0.088	\	2.5e-3
SSC + MAP	\	0.097	7.3e-4
SCC + SSC + MAP	0.069	0.069	3.1e-3

6.2 Analysis of Combined Pre-training Tasks

After analyzing the three individual pre-training tasks, we explore potential performance improvements through task combinations. During combined pre-training, in addition to the input features and learning rate, the coefficients of the loss terms (α, β, γ) in Equation 4 also impact performance. Table 1 presents the results for balanced multi-task pre-training ($\alpha = \beta = \gamma = 1$). Notably, each task's performance degrades when trained simultaneously with others, indicating conflicts between different training objectives.

Through extensive experimentation, we observe several key patterns. The conclusion regarding input features follows the previous section: using all price values consistently yields better results. The learning rate is confirmed to be 10^{-3} for all combinations, as when combined with other tasks, MAP's MSE results are not sensitive to the learning rate, remaining around the values in Table 1. Regarding loss coefficients, increasing α to 5 marginally improves SCC accuracy without compromising SSC performance. Increasing γ reduces the MSE of MAP but significantly harms the classification tasks. Given the uncertain relationship between pre-training balance and stock selection performance, we explore various coefficient combinations and select the best model based on validation set results for further investigation into how combined pre-training influences subsequent stock selection.

Figure 6 presents the Sharpe ratio results for several representative pre-training combinations. Given MAP's significant performance degradation (higher MSE) in combined settings, we prioritize SCC and SSC in determining the fine-tuning strategy, freezing only the embedding layers. We observe that the combination of SCC and SSC maintains good performance. However, the combinations including the MAP task are different. Although combined training presents improved SR results compared with using MAP individually, two out of the three combinations including MAP are worse than not using pre-training, suggesting that MAP is not an optimal component for combined pre-training. These findings are consistent with our previous conclusions, as MAP favors different settings from the other two tasks, and the MAP task is negatively affected significantly when combined with other tasks in pre-training.

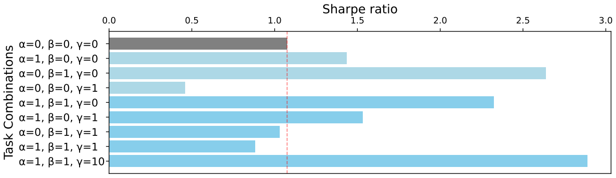

Figure 6: Sharpe ratio results on the NASDAQ market for the stock selection task using pre-trained models with different combinations of pre-training tasks. The values of α, β, γ represent the loss coefficients for the SCC, SSC, and MAP tasks in the combined pre-training loss (Equation 4). When a coefficient is 0, the corresponding task is excluded from pre-training. The first row shows the baseline result without pre-training. The next three rows display results when pre-training on individual tasks (using the fine-tuning strategy of freezing only the embedding layers). Subsequent rows present results from combinations of pre-training tasks.

However, the last line in Figure 6 presents an interesting case: combining all three tasks with increased MAP coefficient ($\gamma = 10$), yields a Sharpe ratio exceeding the SCC+SSC combination. This indicates that MAP includes knowledge that the other two tasks cannot capture. However, as this coefficient combination highly relies on validation selection, and other coefficients with higher γ values, like $\gamma = 5$, do not consistently present the same trend, including MAP in the combined pre-training tasks still indicates potential robustness issues.

In summary, we find that the combination of SCC and SSC pre-training can reliably improve the stock selection results. However, the inclusion of MAP is difficult to determine as its performance is sensitive to the loss coefficient settings. Therefore, a relatively robust recipe for pre-training task combination is to combine SCC and SSC with balanced coefficients ($\alpha = \beta = 1$).

6.3 Comparison with Existing Methods

After analyzing the effectiveness and usage of our pre-training tasks, we now demonstrate SSPT's superior performance compared to existing methods and market benchmarks. In addition to various baselines for stock selection, we include a baseline strategy of selecting all available stocks daily to establish basic market performance. Trading strategies that outperform this market baseline are considered as effectively beating the market.

Note that while previous sections' results were based on models selected from specific hyper-parameter setting groups to facilitate detailed analysis, our comparative evaluation uses models selected from a broader range of settings based on validation results, ensuring fair comparison. Consequently, some results here may differ from those previously reported.

Table 2 compares our methods with various baselines across three markets: NASDAQ, NYSE, and TOPIX-100. These widely-studied datasets enable comprehensive performance comparison. We evaluate our method in two ways: using only individual pre-training tasks (SSPT-ind) and using combined pre-training tasks (SSPT-comb). Both our models consistently outperform the market and achieve the best SR performance, with competitive IRR results across all three datasets.

Table 3 extends our analysis to FTSE-100 and NASDAQ-recent datasets. Due to implementation constraints, such as unavailable codes or extra required information for other methods, we compare these results only against market performance. Having demonstrated SSPT's superiority against various methods in Table 2, we focus on showing consistent market outperformance across more exchanges and time periods in Table 3. Notably, SSPT maintains

Table 2: Comparison of stock selection performance on NASDAQ, NYSE, and TOPIX-100 markets. Our SSPT models using individual pre-training tasks (SSPT-ind) and combined pre-training tasks (SSPT-comb) are compared with various existing methods. Bold and underline values indicate the best and second-best results, respectively.

	Model	Description	NASDAQ		NYSE		TOPIX-100	
			IRR	SR	IRR	SR	IRR	SR
	Market	Select all stocks, showing basic market performance.	0.15	1.53	0.10	1.49	0.02	0.19
CLF	ARIMA [49] (1996)	RNN with features from ARIMA analysis.	0.10	0.55	0.10	0.33	0.13	0.47
	Adv-ALSTM [17] (2019)	Adversarial training for better prediction generalization.	0.23	0.97	0.14	0.81	0.43	1.10
	HGCluster [35] (2014)	Use a hyper-graph model to predict the stock trends.	0.10	0.06	0.11	0.10	0.10	0.20
	HATS [25] (2019)	A hierarchical attention network using relational data.	0.15	0.80	0.12	0.73	0.31	0.96
	HMG-TF [11] (2020)	Use a Multi-Scale Gaussian Prior to improve model.	0.19	0.83	0.13	0.75	0.33	1.05
	LSTM-RGCN [28] (2021)	Model stock connections by their correlation matrix.	0.13	0.75	0.10	0.70	0.28	0.90
	DTML [58] (2021)	Learn the correlations between stocks for prediction.	0.41	1.35	0.45	1.17	0.35	1.07
	HATR [47] (2022)	Grasp multi-scale transition regularities of stocks.	0.31	0.92	0.14	0.76	0.36	0.98
REG	SFM [61] (2017)	A State Frequency Memory model.	0.09	0.16	0.11	0.19	0.07	0.08
	DA-RNN [42] (2017)	A dual-stage attention-based model.	0.14	0.71	0.13	0.66	0.25	0.86
	TimeMixer [53] (2024)	Analyze temporal variations by multiscale-mixing.	0.42	1.64	0.23	1.23	0.30	0.93
	StockMixer [16] (2024)	Use influences between stock and market.	0.20	1.40	0.54	1.57	0.33	1.12
	Master [27] (2024)	Use market information for automatic feature selection.	0.24	1.20	0.23	1.27	0.25	0.95
RL	DQN [6] (2021)	An ensemble of deep Q-learning agents.	0.20	0.93	0.12	0.72	0.31	1.08
	iRDPG [31] (2020)	Deep reinforcement learning and imitation learning.	0.28	1.32	0.18	0.85	0.55	1.10
	RAT [55] (2021)	A relation-aware Transformer with RL.	0.40	1.37	0.22	1.03	$\underline{0.64}$	1.20
RAN	RSR-I [18] (2019)	A temporal GCN capturing stock relations.	0.39	1.34	0.21	0.95	0.53	1.08
	STHAN-SR [43] (2021)	A neural hyper-graph architecture for stock selection.	0.44	1.42	0.33	1.12	0.62	1.19
	MTSR [36] (2022)	Stock ranking with multi-task learning.	0.30	1.58	0.57	1.36	0.33	1.03
	ALSP-TF [48] (2022)	An adaptive long-short patter transformer.	0.53	1.55	0.41	1.24	**0.71**	$\underline{1.27}$
	TSPRank [29] (2024)	A hybrid pairwise-listwise ranking method.	0.29	1.43	0.28	1.74	0.35	1.11
	CI-STHPAN [54] (2024)	A spatio-temporal hyper-graph model with pre-training.	0.66	2.01	**0.79**	$\underline{2.14}$	0.28	0.91
	SSPT-ind (ours)	Models based on our individual stock pre-training tasks.	$\underline{0.74}$	**2.32**	0.41	2.11	0.51	**1.33**
	SSPT-comb (ours)	Models based on our combined stock pre-training tasks.	**0.82**	$\underline{2.25}$	$\underline{0.60}$	**2.35**	0.43	1.21

Table 3: Comparison of market performance and our SSPT method on FTSE-100 and recent NASDAQ markets.

Model	FTSE-100		NASDAQ-recent	
	IRR	SR	IRR	SR
Market	0.17	2.08	0.16	0.97
SSPT-ind	0.38	2.51	0.59	1.23
SSPT-comb	0.33	2.30	0.48	1.33

strong performance even on the NASDAQ-recent dataset, which includes the challenging COVID-19 period.

These comprehensive comparisons demonstrate that SSPT's price series pre-training tasks extract more beneficial knowledge for stock selection than the spatial or temporal information captured by other methods. Using a standard transformer structure, which is widely used as the backbone for the listed baselines, SSPT's SOTA results highlight the effectiveness of our customized stock pre-training approach.

6.4 Task Analysis on Simulated Data

Having demonstrated our pre-training tasks can benefit stock selection, we investigate the underlying mechanisms of their effectiveness. We hypothesize that these tasks enhance stock selection by extracting distinguishing features from price series data. To investigate this hypothesis, we analyze the classification tasks through controlled simulation experiments.

We use the Wiener Process, a continuous-time stochastic process widely used in finance to model the random behavior of asset prices, to generate simulated stock price series [3, 22, 44]. The simulation process is formulated as:

$$S(t + \Delta t) = S(t) * \exp((\mu - \sigma^2/2)\Delta t + \sigma\sqrt{\Delta t}Z_t). \qquad (6)$$

Here, Δt represents a small time increment, Z_t is a random variable sampled from a standard normal distribution, μ and σ are the simulation parameters representing the expected return and volatility of

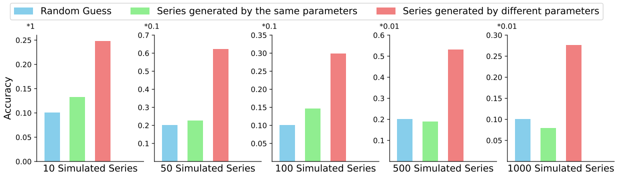

Figure 7: Classification results on 10, 50, 100, 500, and 1000 simulated series. Each accuracy value is averaged over 5 simulation runs. The y-axis scales are independent for each figure, with the scale factors indicated.

the stock. To generate a price series, we need to specify the initial value $S(0)$, the time horizon t, and the parameters μ and σ.

Following our SCC and SSC task designs, we simulate N price series, segment them into equal-length slices, and train a classification model to identify source series for given slices. We maintain constant $S(0)$ and t while comparing two scenarios: (1) simulating N series with identical random μ and σ, and (2) simulating N series with different random μ and σ. Figure 7 shows classification accuracy of the two scenarios across different values of N.

We observe that the classification results for series generated with identical parameters are very close to random guessing, while the accuracy for series generated with different parameters is much higher. This indicates that classification models primarily rely on the different statistical properties of the original series to differentiate series slices. This conclusion supports our hypothesis that the SCC and SSC tasks can extract information related to the statistical characteristics that distinguish various stocks, thereby providing more informative features for the stock selection task.

While practical stock price series are much more complex and contain more latent statistical features than our simulated data, our results indicate that the pre-training tasks function by extracting distinguishing information. Additional simulation results and analysis in Appendix Section E further demonstrate the pre-training tasks' sensitivity to price series' statistical features, strengthening our hypothesis.

7 Conclusions

This paper introduces three novel pre-training tasks specifically designed for stock price data. Through extensive experiments, we demonstrate the effectiveness of these customized pre-training tasks in enhancing stock selection. Our Stock Specialized Pre-trained Transformer (SSPT) framework, built upon these pre-training methods and a standard transformer architecture, outperforms the market and existing methods on five datasets. We conduct a comprehensive analysis to provide practical guidance on optimally using our pre-training tasks. This includes identifying the most influential factors and the best fine-tuning strategies for individual tasks as well as their combinations. Furthermore, through experiments on simulated data with controlled parameters, we explore the underlying reasons for the effectiveness of our pre-training methods. In summary, this work introduces an effective stock selection framework, provides practical implementation guidelines, and offers insights into stock price series analysis. The demonstrated success of our specialized pre-training approach suggests promising directions for future research in financial market analysis and prediction.

Acknowledgments

This work was supported by the UKRI Centre for Doctoral Training (CDT) in Natural Language Processing through UKRI grant EP/S022481/1. We would like to thank Chang Luo, Waylon Li, and the anonymous reviewers for their valuable feedback.

References

[1] Ayodele Ariyo Adebiyi, Aderemi Oluyinka Adewumi, and Charles Korede Ayo. 2014. Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics* 2014, 1 (2014), 614342.

[2] Wei Bao, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one* 12, 7 (2017), e0180944.

[3] Tomas Björk. 2009. *Arbitrage theory in continuous time.* Oxford university press.

[4] William Brock, Josef Lakonishok, and Blake LeBaron. 1992. Simple technical trading rules and the stochastic properties of stock returns. *The Journal of finance* 47, 5 (1992), 1731–1764.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. [n. d.]. Language models are few-shot learners. *Advances in neural information processing systems* 33 ([n. d.]), 1877–1901.

[6] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. 2021. Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting. *Expert systems with applications* 164 (2021), 113820.

[7] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management.* 1655–1658.

[8] Bent Jesper Christensen and Morten Ørregaard Nielsen. 2007. The effect of long memory in volatility on stock market fluctuations. *The Review of Economics and Statistics* 89, 4 (2007), 684–700.

[9] David M Cutler, James M Poterba, and Lawrence H Summers. 1988. *What moves stock prices?* Vol. 487. National Bureau of Economic Research Cambridge, Massachusetts.

[10] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[11] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. 2020. Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction.. In *IJCAI.* 4640–4646.

[12] Fama Eugene and Kenneth French. 1992. The cross-section of expected stock returns. *Journal of finance* 47, 2 (1992), 427–465.

[13] Eugene F Fama. 1970. Efficient capital markets: A review of theory and empirical work. *The journal of Finance* 25, 2 (1970), 383–417.

[14] Eugene F Fama. 1995. Random walks in stock market prices. *Financial analysts journal* 51, 1 (1995), 75–80.

[15] Alan Fan and Marimuthu Palaniswami. 2001. Stock selection using support vector machines. In *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*, Vol. 3. IEEE, 1793–1798.

[16] Jinyong Fan and Yanyan Shen. 2024. StockMixer: a simple yet strong MLP-based architecture for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8389–8397.

[17] Fuli Feng, Huimin Chen, Xiangnan He, Jie Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training.. In *IJCAI*, Vol. 19. 5843–5849.

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

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 16000–16009.

[20] 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 & Knowledge Management.* 700–709.

[21] Wenpeng Hu, Mengyu Wang, Bing Liu, Feng Ji, Haiqing Chen, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Transformation of dense and sparse text representations. *arXiv preprint arXiv:1911.02914* (2019).

[22] John C Hull and Sankarshan Basu. 2016. *Options, futures, and other derivatives.* Pearson Education India.

[23] 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.

[24] G Kavitha, A Udhayakumar, and D Nagarajan. 2013. *Stock Market Trend Analysis Using Hidden Markov Models.* Technical Report. arXiv. org.

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

[26] Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2024. Learning to generate explainable stock predictions using self-reflective large language models. In *Proceedings of the ACM on Web Conference 2024.* 4304–4315.

[27] Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. 2024. Master: Market-guided stock transformer for stock price forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 162–170.

[28] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2021. 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.* 4541–4547.

[29] Weixian Waylon Li, Yftah Ziser, Yifei Xie, Shay B Cohen, and Tiejun Ma. 2024. TSPRank: Bridging Pairwise and Listwise Methods with a Bilinear Travelling Salesman Model. *arXiv preprint arXiv:2411.12064* (2024).

[30] Jintao Liu, Hongfei Lin, Xikai Liu, Bo Xu, Yuqi Ren, Yufeng Diao, and Liang Yang. 2019. Transformer-based capsule network for stock movement prediction. In *Proceedings of the first workshop on financial technology and natural language processing*. 66–73.

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

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[33] Andrew W Lo and A Craig MacKinlay. 1990. When are contrarian profits due to stock market overreaction? *The review of financial studies* 3, 2 (1990), 175–205.

[34] Di Luo, Weiheng Liao, Shuqi Li, Xin Cheng, and Rui Yan. 2023. Causality-Guided Multi-Memory Interaction Network for Multivariate Stock Price Movement Prediction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 12164–12176.

[35] Yongen Luo, Jicheng Hu, Xiaofeng Wei, Dongjian Fang, and Heng Shao. 2014. Stock trends prediction based on hypergraph modeling clustering algorithm. In *2014 IEEE International Conference on Progress in Informatics and Computing*. IEEE, 27–31.

[36] Tao Ma and Ying Tan. 2022. Stock ranking with multi-task learning. *Expert Systems with Applications* 199 (2022), 116886.

[37] Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? *arXiv preprint arXiv:2004.14448* (2020).

[38] Tobias J Moskowitz and Mark Grinblatt. 1999. Do industries explain momentum? *The Journal of finance* 54, 4 (1999), 1249–1290.

[39] John J Murphy. 1999. *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Penguin.

[40] Rudra Kalyan Nayak, Debahuti Mishra, and Amiya Kumar Rath. 2015. A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices. *Applied Soft Computing* 35 (2015), 670–680.

[41] G Preethi and B Santhi. 2012. STOCK MARKET FORECASTING TECHNIQUES: A SURVEY. *Journal of Theoretical & Applied Information Technology* 46, 1 (2012).

[42] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971* (2017).

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

[44] Steven Shreve. 2005. *Stochastic calculus for finance I: the binomial asset pricing model*. Springer Science & Business Media.

[45] Chaojie Wang, Yuanyuan Chen, Shuqi Zhang, and Qiuhui Zhang. 2022. Stock market index prediction using deep Transformer model. *Expert Systems with Applications* 208 (2022), 118128.

[46] Heyuan Wang, Shun Li, Tengjiao Wang, and Jiayi Zheng. 2021. Hierarchical Adaptive Temporal-Relational Modeling for Stock Trend Prediction.. In *IJCAI*. 3691–3698.

[47] Heyuan Wang, Tengjiao Wang, Shun Li, and Shijie Guan. 2022. HATR-I: Hierarchical adaptive temporal relational interaction for stock trend prediction. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2022), 6988–7002.

[48] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. 2022. Adaptive Long-Short Pattern Transformer for Stock Investment Selection.. In *IJCAI*. 3970–3977.

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

[50] Mengyu Wang, Shay B Cohen, and Tiejun Ma. 2024. Modeling News Interactions and Influence for Financial Market Prediction. *arXiv preprint arXiv:2410.10614* (2024).

[51] Mengyu Wang and Tiejun Ma. 2024. MANA-Net: Mitigating Aggregated Sentiment Homogenization with News Weighting for Enhanced Market Prediction. *arXiv preprint arXiv:2409.05698* (2024).

[52] Mengyu Wang, Yijia Shao, Haowei Lin, Wenpeng Hu, and Bing Liu. 2022. Cmg: A class-mixed generation approach to out-of-distribution detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 502–518.

[53] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616* (2024).

[54] Hongjie Xia, Huijie Ao, Long Li, Yu Liu, Sen Liu, Guangnan Ye, and Hongfeng Chai. 2024. CI-STHPAN: Pre-trained Attention Network for Stock Selection with Channel-Independent Spatio-Temporal Hypergraph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9187–9195.

[55] Ke Xu, Yifan Zhang, Deheng Ye, Peilin Zhao, and Mingkui Tan. 2021. Relation-aware transformer for portfolio policy learning. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*. 4647–4653.

[56] Xu Yan and Zhang Guosheng. 2015. Application of kalman filter in the prediction of stock price. In *5th international symposium on knowledge acquisition and modeling (KAM 2015)*. Atlantis press, 197–198.

[57] Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. 2020. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1112–1119.

[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 Conference on Knowledge Discovery & Data Mining*. 2037–2045.

[59] Xu Yuemei, Wang Zihou, and Wu Zixin. 2021. Predicting Stock Trends with CNN-BiLSTM Based Multi-Feature Integration Model. *Data Analysis and Knowledge Discovery* 5, 7 (2021), 126–138.

[60] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2114–2124.

[61] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2141–2149.

[62] Wentao Zhang, Yilei Zhao, Shuo Sun, Jie Ying, Yonggang Xie, Zitao Song, Xinrun Wang, and Bo An. 2024. Reinforcement Learning with Maskable Stock Representation for Portfolio Management in Customizable Stock Pools. In *Proceedings of the ACM on Web Conference 2024*. 187–198.

[63] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems* 35 (2022), 3988–4003.

[64] Yongjie Zhang, Weixin Song, Dehua Shen, and Wei Zhang. 2016. Market reaction to internet news: Information diffusion and price pressure. *Economic Modelling* 56 (2016), 43–49.

[65] Yu Zhang and Qiang Yang. 2021. A survey on multi-task learning. *IEEE transactions on knowledge and data engineering* 34, 12 (2021), 5586–5609.

[66] Tian Zhou, Peisong Niu, Liang Sun, Rong Jin, et al. 2023. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems* 36 (2023), 43322–43355.

[67] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020).

A Dataset Introduction

Our experiments use five datasets spanning major stock markets across the US, UK, and Japan over different time periods: NASDAQ (2013-2017), NYSE (2013-2017), FTSE-100 (2013-2017), TOPIX-100 (2016-2020), and NASDAQ-recent (2018-2022). NASDAQ, NYSE, and TOPIX-100 serve as established benchmark datasets in stock selection research [18, 48], while FTSE-100 and NASDAQ-recent were collected from Yahoo Finance (https://finance.yahoo.com). Sectors are assumed to be constant in each dataset. They can be updated periodically, as we do with NASDAQ and NASDAQ-recent. Additionally, sector shifts are rare and minor for large companies, thus affecting pre-training marginally.

NASDAQ [18] is a highly volatile US exchange, comprising 1,026 stocks from 112 sectors, drawn from S&P 500 and NASDAQ Composite Indexes.

NYSE [18] is the world's largest stock exchange by market capitalization, offers relatively stable market conditions compared to NASDAQ. The dataset includes 1,737 stocks across 129 sectors.

TOPIX-100 [28] is a smaller market contrasting with US markets. It represents Japan's major market index, featuring 95 stocks from 10 sectors, focusing on the largest market capitalizations in the Tokyo Stock Exchange.

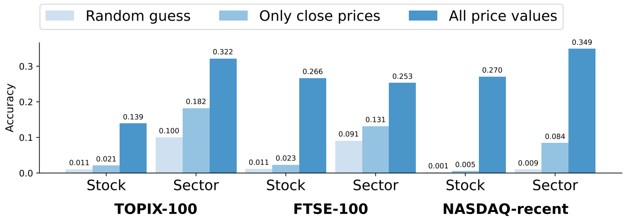

Figure 8: Classification accuracy comparison for the two pre-training tasks SCC and SSC on TOPIX-100, FSTE-100, and NASDAQ-recent datasets. All results are rounded to three decimal places.

FTSE-100 is the UK's best-known stock market index, representing the largest market capitalizations on the London Stock Exchange. This dataset includes 87 stocks from 11 sectors.
NASDAQ2 tracks the same companies as the NASDAQ datast but during 2018-2022, a recent period including the challenging COVID time. This dataset includes 718 stocks across 106 sectors.

These datasets provide comprehensive coverage of diverse market conditions, including both established American markets (widely considered most efficient) and representative markets from other regions. The datasets span both growth periods and volatile years, ensuring robust and generalizable evaluation of our methods.

B More Results of Classification Tasks

In Section 4.1, we presented classification accuracy results for SCC and SSC on the two largest datasets, NASDAQ and NYSE. Figure 8 extends this analysis to the TOPIX-100, FTSE-100, and NASDAQ-recent datasets. The results demonstrate consistent patterns across all datasets: classification accuracy significantly exceeds random guessing, and using comprehensive price values as inputs consistently yields optimal results. These findings reinforce our conclusions from Section 4.1: price series contain distinctive information that enables effective classification, and using all price values maximizes feature quality.

C Further Analysis of the MAP Task

Since our MAP task is motivated by the traditional masked value prediction (MVP) approach and introduces additional hyper-parameters such as the mask rate, we conduct further experimental analysis to better understand its behavior.

First, we compare our MAP method with the traditional MVP approach to validate the effectiveness of our design tailored for stock data. Both pre-training tasks are conducted on the NASDAQ dataset, followed by a downstream stock selection task. We report MSE during pre-training and IRR and SR during the stock selection phase. The results are summarized in Table 4.

Table 4: Comparison between traditional masked value prediction (MVP) and our proposed MAP method on pre-training and downstream stock selection performance.

Method	Pre-training			Stock Selection	
	MSE (lr=1e-3)	MSE (lr=1e-4)	MSE (lr=1e-5)	IRR	SR
				(NASDAQ)	
MVP	7.6e-5	9.5e-6	1.5e-4	0.53	1.54
MAP	8.5e-6	3.6e-6	8.3e-3	0.59	1.97

The lowest MSE of MVP is much higher than that of MAP, suggesting that predicting masked moving averages is more feasible than predicting specific masked values in the context of volatile financial data. Moreover, the stock selection performance (IRR and SR) achieved using MAP-based pre-training also surpasses that of MVP-based pre-training under the same experimental settings. Notably, our baseline method [54] is itself designed around masked value prediction and includes specialized structures to improve its performance. Despite this, our SSPT framework outperforms it, further validating the advantages of our MAP design over traditional MVP approaches.

Next, we investigate how varying the mask rate affects MAP performance. We vary the mask rate from 0.05 to 0.5 in increments of 0.05 and report the corresponding pre-training MSE and stock selection SR in Figure 9.

As shown, the MSE increases from 8.0×10^{-7} to 1.9×10^{-6} as the mask rate increases. In terms of SR, stock selection performance is 1.08 at a mask rate of 0.05, even underperforming the no-pretraining baseline. MAP begins to show benefits at a mask rate of 0.15, with the highest SR observed around a mask rate of 0.3.

D IRR Results for Pre-training Analysis

Both the cumulative investment return ratio (IRR) and Sharpe ratio (SR) are widely used metrics to evaluate stock selection performance. While IRR focuses solely on returns, SR considers both returns and risks. Although these two metrics are not linearly related, their overall trends are generally similar. In the main text, we presented SR results in Figures 5 and 6 to analyze the pre-training tasks. Here, we present the corresponding IRR results in Figures 10 and 11.

If disregarding the specific values, these two figures closely resemble the previous figures for the SR metric. The main difference lies in the fact that the SCC task with frozen embedding layer fine-tuning and the SSC task with frozen embedding and attention layers fine-tuning fail to surpass the performance of direct training without pre-training. However, the most optimized results remain consistent. All conclusions regarding the hyper-parameter and training strategies from the SR metric remain unchanged. This consistency across different metrics indicates the robustness of our methods, demonstrating reliable performance improvements.

E More Analysis on Simulated Data

To further investigate the ability of our classification tasks to capture distinguishing statistical features from time series data, we

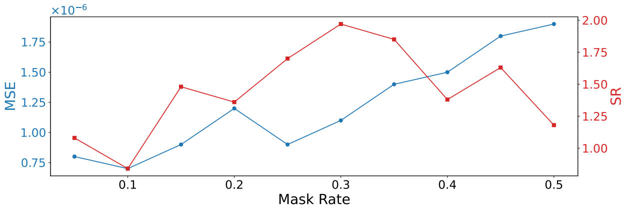

Figure 9: Pre-training MSE and stock selection SR results using MAP under different mask rate settings. The blue line represents MSE (left y-axis), while the red line indicates SR (right y-axis).

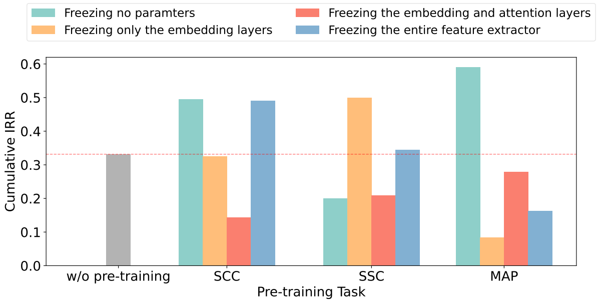

Figure 10: The cumulative IRR results of the stock selection task with different training strategies. The group 'w/o pre-training' presents the result of directly training the model on the stock selection task without any pre-training.

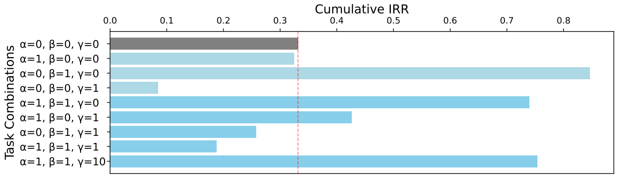

Figure 11: Cumulative IRR results on the NASDAQ market for the stock selection task using pre-trained models with different combinations of pre-training tasks. The values of α, β, γ represent the loss coefficients for the SCC, SSC, and MAP tasks in the combined pre-training loss (Equation 4). When a coefficient is 0, the corresponding task is excluded from pre-training. The first row shows the baseline result without pre-training. The next three rows display results when pre-training on individual tasks (using the fine-tuning strategy of freezing only the embedding layers). Subsequent rows present results from combinations of pre-training tasks.

analyze the sensitivity of the classification accuracy to the degree of difference in the statistical parameters used to generate the simulated series.

Focusing on the classification of 10 simulated series, we control the values of $S(0)$, the time horizon t, and the expected return μ to be the same across all series, while varying the volatility σ used to generate each series. Specifically, we change the random range from which the σ values are drawn to control the difference in volatilities between the series. For example, σ values randomly drawn from $(0.1, 0.2)$, with a range of 0.1, have a smaller difference compared to values drawn from $(0.1, 0.3)$, with a range of 0.2. Figure 12 presents the classification results for different random ranges of σ.

We observe a clear upward trend in accuracy as the range of σ increases, indicating that the classification task becomes more effective when the statistical features of the time series differ more significantly. This further substantiates our hypothesis that the stock code classification (SCC) and stock sector classification (SSC) tasks can learn to distinguish the unique statistical characteristics inherent to different stocks or sectors, thereby extracting informative features that benefit the subsequent stock selection task.

While practical stock price series exhibit far greater complexity than our simulated data and likely contain additional hidden statistical features, these controlled experiments provide insights into the underlying mechanisms through which our pre-training tasks enhance the stock selection performance. By capturing distinguishing statistical patterns within price series data, the SCC and SSC tasks acquire knowledge that may not directly relate to profitability but can improve the overall understanding and prediction of stock price movements.

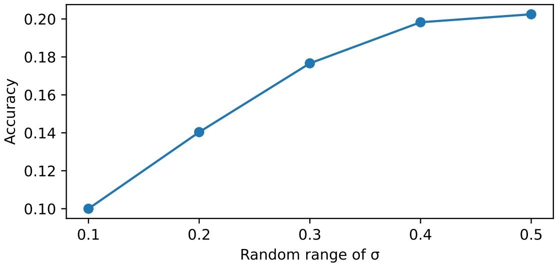

Figure 12: Classification accuracy on 10 simulated stock price series, where the volatility parameter σ is randomly drawn from different ranges. The accuracy is averaged over 5 simulation runs with identical settings for each range of σ.