entropy

MDPI

Article

Wavelet-Decoupled Spatiotemporal Network for Stock Return Prediction

Lei Liao [1], Chao Wang [2], Jun Wang [3,*], Yinchao Liao [1] and Yanjie Lai [4]

[1] School of Finance, Southwestern University of Finance and Economics, Chengdu 611130, China; 122020204056@smail.swufe.edu.cn (L.L.); yl1123@smail.swufe.edu.cn (Y.L.)
[2] School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu 611130, China; charlewang@vip.163.com
[3] School of Management Science and Engineering, Southwestern University of Finance and Economics, Chengdu 611130, China
[4] College of Economics and Management, Beijing University of Technology, Beijing 100124, China; lyj13693377646@emails.bjut.edu.cn
* Correspondence: wangjun1987@swufe.edu.cn

Abstract

Stock price prediction is a challenging problem in quantitative investment, as financial markets generate complex, noisy, and dynamic time series containing heterogeneous signals. Short-term fluctuations usually exhibit greater uncertainty and stronger local variation, whereas long-term trends convey relatively stable and persistent information shaped by market and macroeconomic conditions. However, most existing methods struggle to distinguish these two components effectively, often leading to interference between short-term fluctuations and longer-term trends. In addition, they fail to capture dynamic temporal dependencies and cross-stock information propagation while preserving the causal structure of financial time series. To tackle these issues, we propose the Wavelet-Decoupled Spatiotemporal Network (WaveDSTN). It leverages wavelet transformation to decompose stock returns into high-frequency and low-frequency information, corresponding to short-term fluctuations and long-term trends, respectively. This decomposition enables the model to learn complementary predictive patterns more effectively. Furthermore, WaveDSTN incorporates a Dual-Path Spatiotemporal Encoder to capture complex temporal dependencies and evolving cross-stock information propagation while preserving temporal order and causal consistency. Extensive experiments demonstrate that WaveDSTN achieves significant improvements over existing methods, showing that explicitly modeling trend and fluctuation components can enhance predictive accuracy and reduce uncertainty in stock return forecasting.

Keywords: stock prediction; information decomposition; spatiotemporal encoder; cross attention

Check for updates

Academic Editor: Gil Cohen

Received: 7 April 2026
Revised: 5 May 2026
Accepted: 7 May 2026
Published: 12 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Stock returns prediction is a fundamental of quantitative investment, with applications ranging from portfolio optimization to algorithmic trading. Nevertheless, accurate forecasting remains difficult because financial markets are highly dynamic and noisy, and stock returns are shaped by multiple interacting sources of information, including macroeconomic indicators, firm-level disclosures, geopolitical events, and investor sentiment [1,2]. The core challenge of stock return prediction lies in extracting useful predictive information

from mixed market signals while suppressing irrelevant disturbances. In practice, such complexity is largely reflected in two complementary components of market behavior: long-term trends and short-term fluctuations [3]. Long-term trends aggregate relatively stable information associated with broader market conditions and macroeconomic forces, and therefore evolve more smoothly over extended horizons [4]. In contrast, short-term fluctuations are often driven by localized shocks, rapid information arrivals, trading activity, and sentiment shifts, which introduce stronger uncertainty and richer local variation. Since these two components carry different types of predictive information, effective stock return prediction depends on the ability to distinguish, preserve, and jointly model them [5].

Traditional methods, including decision trees and support vector machines, have been widely employed to predict stock returns [6–9]. While these approaches leverage statistical relationships, they heavily rely on manually engineered features and often struggle to capture the nonlinear and dynamic relationships inherent in stock market data. Moreover, they cannot effectively distinguish between long-term trends and short-term fluctuations.

With the advent of deep learning, researchers have made significant progress in stock price prediction by leveraging the powerful representation capabilities of neural networks [10,11]. Two predominant directions have emerged: (1) Time correlation. Stock price movements are caused by continuous changes in supply and demand, and time trend changes have obvious dependencies. Representative models include recurrent and convolutional neural networks, which learn stock-price dynamics from individual time-series features. [12–14]. (2) Stock correlation. Different stocks in the market form complex and dynamic dependencies due to various factors, such as division of labor and industry status, and stock-price fluctuations affect one another [15,16]. The fusion of stock spatiotemporal features through graphs or self-attention mechanisms can effectively enhance prediction capabilities [2,17]. However, these methods face two limitations.

First, existing studies often fail to adequately exploit the distinct predictive information contained in trend and fluctuation components. Trend information reflects relatively stable long-horizon movement and captures the overall direction of market evolution, whereas fluctuation information corresponds to short-term deviations around the trend and contains rich local variation associated with transient market changes. When these two components are directly mixed or only crudely decomposed, useful predictive patterns may be weakened by information interference, causing complementary signals to be diluted or obscured [15,18,19]. As a result, previous methods often cannot fully utilize the complementary information contained in trend and fluctuation components.

In financial markets, information does not remain isolated within individual stocks, but can propagate across stocks over time through shared fundamentals, market sentiment, industry linkages, and common external shocks [20,21]. Cross-stock relationships are therefore time-varying rather than fixed, and features at different time steps encode distinct temporal states. Moreover, information exchange among stock representations in a forecasting setting must respect causal order: earlier information can influence later states, but not the reverse. Nevertheless, many correlation-mining pipelines either enforce time alignment across series [22], which compresses stock-specific temporal heterogeneity, or treat dependencies as static and omit step-specific representations [23], thereby obscuring causal structure and weakening the model's ability to capture time-varying information propagation across stocks.

To address these challenges, we propose the Wavelet-Decoupled Spatiotemporal Network (WaveDSTN), a framework for stock return prediction that separately models long-term trends and short-term fluctuations. Specifically, WaveDSTN employs a wavelet transform to decompose stock returns into high-frequency and low-frequency components, corresponding to fluctuation and trend information, respectively. The use of wavelet

decomposition is motivated by the representation learning difficulty caused by mixed-frequency return signals. The original return series contains both relatively persistent low-frequency movements and high-frequency variations. If the raw return sequence is directly encoded as a single input, the model has to learn heterogeneous information with different temporal patterns and noise levels within the same representation space. This may weaken the extraction of stable predictive information and increase the interference between trend-related and fluctuation-related signals. Wavelet decomposition provides a multi-resolution representation before neural encoding, allowing the model to learn trend-related and fluctuation-related representations more explicitly. Therefore, in WaveDSTN, wavelet decomposition is not treated merely as a preprocessing step, but as a representation learning strategy for structuring heterogeneous market information. On this basis, WaveDSTN further incorporates a Dual-Path Spatiotemporal Encoder, which combines the temporal representation capability of RWKV with the relational modeling strength of attention mechanisms. This design not only captures evolving dependencies among stocks but also facilitates the modeling of information propagation across stocks and over time while preserving temporal order under causal constraints. Our main contributions are summarized as follows:

(1) We propose a wavelet-decoupled stock return prediction framework that explicitly separates stock returns into trend and fluctuation components. Unlike existing methods that usually model the original return series as a single mixed signal, the proposed framework decomposes heterogeneous market information into low-frequency and high-frequency components. This design helps reduce the interference between relatively stable long-term information and volatile short-term variations, thereby providing cleaner and more informative inputs for return prediction.

(2) We introduce a Dual-Path Spatiotemporal Encoder to jointly model temporal information evolution and cross-stock information propagation. The temporal path adopts RWKV-based causal sequence modeling to preserve the chronological order of financial time series and avoid future information leakage. The spatial path uses an adaptive attention-based relation learning mechanism to capture dynamic cross-stock dependencies. Compared with static graph-based models or purely temporal models, this design allows WaveDSTN to better capture time-varying market relationships under causal constraints.

(3) We design a cross-attention mechanism to model the interaction between trend and fluctuation components. Instead of treating the two decomposed components as independent signals, the proposed mechanism allows the model to adaptively learn how long-term trends and short-term fluctuations influence each other under different market conditions. This improves the model's ability to utilize complementary information from different frequency components.

(4) Through extensive experiments, we demonstrate the superiority of WaveDSTN over state-of-the-art methods, achieving substantial improvements in both ranking metrics and portfolio-based indicators. The results show that separating and jointly modeling trend and fluctuation information can improve predictive performance and enhance the utilization of complementary market information.

2. Related Work

Stock price prediction is a central problem in finance due to its practical importance and complexity. Existing approaches can be broadly categorized into traditional machine learning methods and deep learning-based techniques. This section reviews the key developments in these areas, highlighting their strengths, limitations, and relevance to our proposed framework.

2.1. Machine Learning Methods

Traditional machine learning methods—such as decision trees, support vector machines (SVMs), and ensemble approaches like random forests and gradient boosting machines—have been used in stock return prediction for their interpretability, ease of implementation, and computational efficiency [6–8,24]. These models typically transform historical market and fundamental data into manually engineered features, including technical indicators (e.g., moving averages, RSI) and fundamental ratios (e.g., P/E, debt-to-equity). While ensemble learners, such as XGBoost, have improved non-linear fitting by combining multiple weak learners, and SVMs can capture non-linear decision boundaries via kernel methods [25], the representation capacity of such models is still constrained by the scope and quality of handcrafted inputs. Unsupervised learning methods, such as k-means clustering and principal component analysis (PCA), have also been employed to preprocess financial data by grouping similar stocks or reducing high-dimensional feature sets [26,27]. Additionally, Bayesian models offer a probabilistic framework for capturing dependencies and dynamically updating predictions based on new data [28,29]. Despite these advancements, traditional methods often struggle to adapt to the complex and dynamic nature of financial markets due to their reliance on manually engineered features, limited ability to capture temporal dependencies, and scalability issues when dealing with large datasets or rapidly changing market conditions. These limitations underscore the need for more advanced techniques that can better model the intricate, spatiotemporal interdependencies inherent in stock markets.

2.2. Deep Learning for Stock Prediction

With the rise of deep learning, researchers have explored the representation capabilities of neural networks to model complex behaviors in stock prices. Existing methods primarily focus on two aspects: temporal correlation modeling and stock correlation modeling. Temporal correlation modeling aims to capture sequential dependencies in stock price movements. Recurrent neural networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), have been widely used to process time-series data [12,13]. These models excel at capturing temporal patterns but often suffer from vanishing gradients when dealing with long-term dependencies. Convolutional neural networks (CNNs) have also been applied to stock prediction, leveraging their ability to detect local patterns in time-series data [14]. However, CNNs are inherently limited in modeling long-range temporal dependencies. Stock correlation modeling focuses on understanding the interactions and dependencies among stocks. Graph-based approaches and self-attention mechanisms have emerged as powerful tools in this domain. Graph neural networks (GNNs) represent stocks as nodes and their relationships as edges, enabling the modeling of inter-stock dependencies [2,17]. Self-attention mechanisms, such as those employed in the Transformer architecture, further enhance modeling capabilities by capturing both local and global dependencies [23]. Despite their success, these methods often fail to incorporate dynamic temporal features and causal relationships across time steps, limiting their ability to fully capture the complex interplay of temporal and spatial features in stock data.

While these methods have advanced the field, they face several critical limitations. Traditional methods are constrained by their reliance on manual feature engineering and their inability to model complex spatiotemporal interactions. Deep learning-based approaches, though more effective, often treat stock price movements as a whole, failing to distinguish between long-term trends and short-term fluctuations. This oversight leads to mutual interference between the two components, thereby reducing the prediction accu-

racy. Furthermore, many existing methods oversimplify the temporal dependencies among stocks, neglecting the causal relationships inherent in financial time series.

3. Preliminaries

Based on existing works on stock market analysis [30], we adopt a commonly used framework, panel data analysis. Under this setting, historical returns, along with 158 factor-based features of multiple stocks, are utilized to forecast their future returns. Formally, let $x_t^i \in \mathbb{R}^{158}$ denote the 158-dimensional feature vector of stock i at time step t, where $i \in \{1, \ldots, N\}$, and N is the number of stocks. Stacking all stocks features at time step t gives $X_t = \{x_t^1, .., x_t^2, \ldots x_t^N\} \in \mathbb{R}^{N \times 158}$ Given a historical dataset: $\mathcal{X} = \{X_1, \ldots, X_{T_1}\} \in \mathbb{R}^{N \times T_1 \times 158}$ of length T_1, the objective is to predict the next-period cross-sectional returns $\hat{Y}_{T_1+1} \in \mathbb{R}^N$.

4. Methodology

4.1. Overall Architecture

As illustrated in Figure 1, the proposed architecture consists of three key modules: the Wavelet Decomposition Module, the Dual-Path Spatiotemporal Encoder, and the Stock Embedding Aggregation Decoder. The Wavelet Decomposition Module processes historical stock return series using discrete wavelet transforms (DWT) to separate low- and high-frequency components, corresponding to stable long-term trends and volatile short-term fluctuations, respectively. By decomposing the original return series into these two components, the model is able to separate mixed information with different dynamic characteristics and reduce interference between them. The decomposed components are then reconstructed through inverse filtering, concatenated with multifactor features, and projected into high-dimensional representations for downstream modeling. The Dual-Path Spatiotemporal Encoder serves as the core of the framework, comprising a temporal feature learning layer and a spatial feature learning layer. The temporal layer utilizes the RWKV model, which maintains temporal causality through its RNN-like structure while leveraging attention mechanisms for efficiency, allowing the model to capture sequential dependencies within individual stocks. The spatial layer adopts a Graph Attention Network (GAT) with an adaptive adjacency matrix to dynamically model cross-stock dependencies and information propagation in the market. In addition, a cross-attention mechanism is introduced to capture the dynamic exchange between trend and fluctuation information, allowing the model to adaptively emphasize the more informative component under different market conditions. Finally, the Stock Embedding Aggregation Decoder consolidates temporal and spatial representations through a temporal attention layer, aggregating relevant historical information into a unified stock embedding. This embedding is then fed into a linear predictor to generate predictions for stock returns.

4.2. Wavelet Decomposition Module

Separating long-term and short-term dynamics requires an appropriate decomposition method. Traditional techniques such as moving averages [31] or the Hodrick–Prescott (HP) filter [32] rely on fixed smoothing rules and often fail to capture local shocks in financial data. In contrast, the Discrete Wavelet Transform (DWT) provides a flexible decomposition framework in both the time and frequency domains, allowing the model to separate mixed information with different dynamic characteristics and to retain informative local patterns more effectively. This property makes wavelets particularly suitable for financial markets, where non-stationarity, volatility clustering, and abrupt regime shifts are common [3,5,33]. Based on this motivation, we employ DWT to disentangle stock returns into trend and fluctuation components, so that relatively stable information and rapidly

varying information can be represented separately, thereby reducing interference between them and providing cleaner inputs for subsequent modeling.

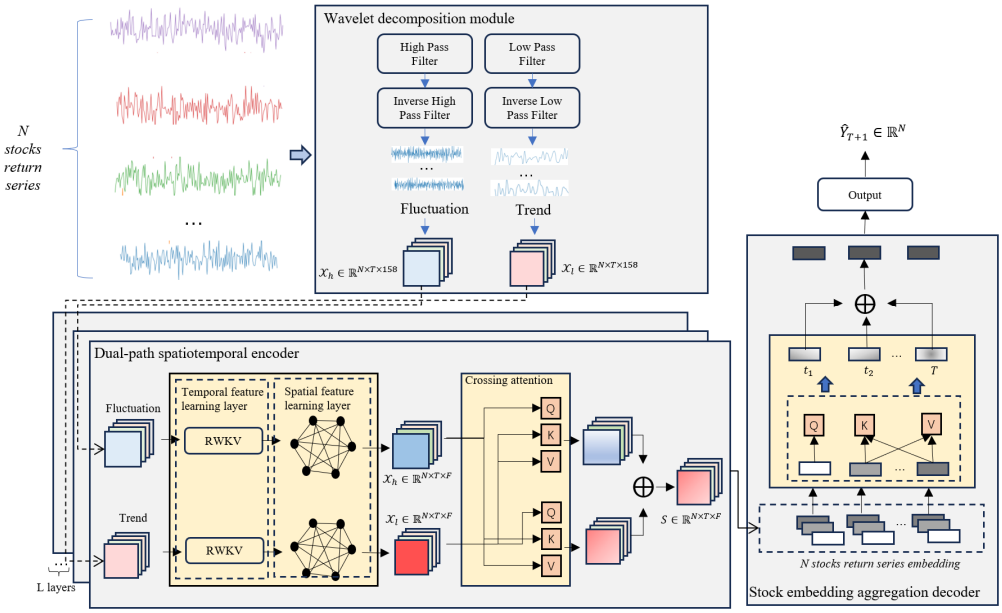


Figure 1. The overview of the WaveDSTN framework, which primarily consists of three parts: the Wavelet decomposition module, the Dual-path spatiotemporal encoder, and the Stock embedding aggregation decoder.

To extract the long-term and short-term temporal features from historical stock return data, we apply the Discrete Wavelet Transform (DWT) to decompose the input sequence $\mathcal{X}$ into two distinct components: low-frequency and high-frequency parts, which correspond to long-term and short-term market patterns, respectively [34]. The low-frequency part captures smoother, stable trends over longer horizons, while the high-frequency part reflects short-term, volatile fluctuations driven by noise or shocks. The transformation of $\mathcal{X}$ is given in Equation (1).

$$\overline{\mathcal{X}}_l = \boldsymbol{g}\mathcal{X}, \overline{\mathcal{X}}_h = \boldsymbol{h}\mathcal{X}, \tag{1}$$

where g and h represent the low-pass and high-pass filters applied in the DWT process, following this, both sub-series $\overline{\mathcal{X}}_l$ and $\overline{\mathcal{X}}_h$ are downsampled to half their original temporal resolution. To recover the original length, we apply the inverse low-pass and high-pass filters $\boldsymbol{g}^T$ and $\boldsymbol{h}^T$ for upsampling. The resulting sequences are then integrated with the 158-dimensional Alpha158 factor features through concatenation. To prepare these fused features for downstream learning, we project them into a unified latent space using fully connected layers, which produce the transformed representations $\mathcal{X}_l$ and $\mathcal{X}_h$. The computation is formalized as:

$$\mathcal{X}_l = W^g \boldsymbol{g}^T \overline{\mathcal{X}}_l + b^g, \mathcal{X}_h = W^h \boldsymbol{h}^T \overline{\mathcal{X}}_h + b^h, \tag{2}$$

where W^g, W^h and b^g, b^h denote the learnable weights and biases associated with the projection layers.

4.3. Dual-Path Spatiotemporal Encoder

4.3.1. Temporal Feature Learning Layer

We model temporal correlations for each stock by capturing interactions across time steps t in its feature sequence X_i. Unlike prior work [23] that uses self-attention over the entire lookback window, we adopt the RNN-based attention architecture RWKV [35]

as an efficient alternative to the Transformer [36]. RWKV offers three advantages for financial time series. First, its recurrent formulation preserves temporal order and avoids the need for positional encodings, allowing sequential information to evolve naturally over time. Second, its causal attention [37] restricts step t to attend only to the current and past steps, which prevents future information leakage and ensures that information flow remains consistent with market causality. Third, its time-recursive score update with a time-dependent softmax achieves linear complexity in time and memory, enabling efficient modeling of temporal information over long horizons. In our framework, the same RWKV block is used for both the trend and fluctuation paths to maintain a consistent dual-path information modeling scheme.

As shown in Figure 2, the RWKV architecture comprises two main parts, namely a time-mixing block and a channel-mixing block. Each sub-block jointly processes features from the current step and its preceding steps, employing a token-shift mechanism to integrate temporal context in a concise and interpretable manner.

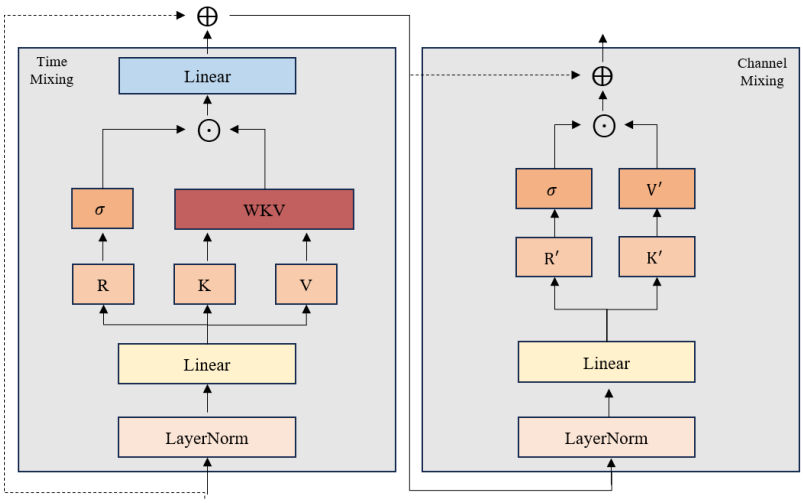


Figure 2. The RWKV structure.

In the time-mixing block, the input at time step t is combined with the previous step using the learnable parameters $\boldsymbol{\mu_r}$, $\boldsymbol{\mu_k}$, $\boldsymbol{\mu_v}$. These combined inputs are then linearly mapped to obtain the receptance vector $\boldsymbol{r}$, the keyword vector $\boldsymbol{k}$, and the value vector $\boldsymbol{v}$:

$$\boldsymbol{r_t} = W_r \cdot (\boldsymbol{\mu_r} \odot X_{i,t} + (1 - \boldsymbol{\mu_r}) \odot X_{i,t-1}), \tag{3}$$

$$\boldsymbol{k_t} = W_k \cdot (\boldsymbol{\mu_k} \odot X_{i,t} + (1 - \boldsymbol{\mu_k}) \odot X_{i,t-1}), \tag{4}$$

$$\boldsymbol{v_t} = W_v \cdot (\boldsymbol{\mu_v} \odot X_{i,t} + (1 - \boldsymbol{\mu_v}) \odot X_{i,t-1}). \tag{5}$$

In RWKV, the WKV operator plays a role analogous to self-attention while achieving linear time and space complexity. It incorporates a time-dependent mechanism for updating attention scores, which can be expressed as:

$$\boldsymbol{wkv_t} = \frac{\sum_{i=1}^{t-1} \boldsymbol{e}^{-(t-1-i)w+k_i} \odot \boldsymbol{v_i} + \boldsymbol{e}^{u+k_t} \odot \boldsymbol{v_t}}{\sum_{i=1}^{t-1} \boldsymbol{e}^{-(t-1-i)w+k_i} + \boldsymbol{e}^{u+k_t}}. \tag{6}$$

Here, w and u are trainable parameters, $t \in [1, T]$ *is* the time stamp index, $\odot$ denotes element-wise product. The vector u introduces a preference at first occurrences, encouraging higher weights when a token is seen for the first time. w is a per-head, channel-wise time-decay parameter that controls how attention diminishes over time across channels.

A multi-head variant is used as $\boldsymbol{wkv_t} = concat\left\{\boldsymbol{wkv}_t^1, ..., \boldsymbol{wkv}_t^N\right\}$, where N is the number of heads. The output vector o_t at time t obtained by the WKV operator is defined as:

$$\boldsymbol{o_t} = W_o \cdot (\sigma(\boldsymbol{r_t}) \odot \boldsymbol{wkv_t}), \tag{7}$$

where σ denotes the squared ReLU activation function [38].

The channel-mixing block performs a nonlinear combination across feature channels, similar to the time-mixing block. The variables k' and r' follow the computation in Equation (4). The output $\boldsymbol{o'_t}$ differs from the time-mixing case and is formulated as:

$$\boldsymbol{o'_t} = \sigma(\boldsymbol{r'_t}) \odot \left(W'_v \cdot Max\left(\boldsymbol{k'_t}, \boldsymbol{0}\right)^2\right), \tag{8}$$

Finally, we obtain the local embedding $\boldsymbol{h}_{i,t}, t \in [1, T]$ that retains information at step t and aggregates dependencies across different time steps, enabling the model to learn temporal correlations for stock i.

4.3.2. Spatial Feature Learning Layer

In financial markets, stock movements are not isolated but are connected through a complex network of dependencies. From an information transmission perspective, shocks, sentiment changes, and market signals arising in one stock can propagate to related stocks through shared economic fundamentals, sectoral linkages, and investor behavior. One prominent manifestation of this process is momentum spillover, in which price movements or volatility in one stock transmit to others and may further amplify their responses. To accurately model such dynamics, we incorporate a Graph Attention Network (GAT) equipped with an adaptive adjacency matrix. This design captures the evolving relationships among stocks and explicitly integrates momentum spillover effects into the modeling process [39].

To capture the dynamic nature of momentum spillover, we design an adaptive adjacency matrix. Unlike static adjacency matrices that rely on predefined relationships (e.g., industry classification or fundamental features), the adaptive adjacency matrix A_t dynamically evolves based on the feature representations of stocks. At each time step t, the relation strength is computed as:

$$R_{i,j}^t = LeakyReLU\left(\boldsymbol{a_r}^\top \mathcal{W}_r \left[\boldsymbol{h}_i^t, \boldsymbol{h}_j^t\right]\right), \tag{9}$$

where $\boldsymbol{h}_i^t$, $\boldsymbol{h}_j^t$ are the firm embeddings after extracting the temporal features. We concatenate $\boldsymbol{h}_i^t$ and $\boldsymbol{h}_j^t$ and project the result to $\mathbb{R}^{F'}$ using a linear map $\mathrm{W}_r \in \mathbb{R}^{F' \times 2F}$. The projected vector is then transformed to a scalar score with a vector $\boldsymbol{a}_r \in \mathbb{R}^{F'}$, which measures the connection strength between the two firms. To make scores $R_{i,j}^t$ comparable across all firms, we apply a Softmax normalization:

$$\tilde{A}{}_{i,j}^t = Softmax_j\left(R_{i,j}^t\right) = \frac{exp\left(R_{i,j}^t\right)}{\sum_{k \in N, k \neq i} exp\left(R_{i,k}^t\right)}, \tag{10}$$

where $\tilde{A}{}_{i,j}^t$ is the normalized connection weight from stock j to i at time t. It automatically adjusts to changes in market conditions, such as shifts in sector correlations or the emergence of new market trends. This adaptability is critical in financial markets, where static assumptions about stock relationships may lead to outdated or inaccurate predictions.

After we have obtained the inferred adjacency matrix, we can use the GAT model to model the spatial relationships between firms. In a traditional graph attention network, the momentum spillover from firm j to firm i can be calculated as:

$$\boldsymbol{s}_i^t = \sigma\left(\sum_{j,j\neq i}^{N} \underbrace{A_{i,j}\mathcal{W}_s\boldsymbol{h}_j^t}_{\text{spillovers from j to i}}\right), \tag{11}$$

where $\mathcal{W}_s \in \mathbb{R}^{F'\times F}$ is a shared weight matrix that linearly maps neighbor attributes to higher-level features. $A_{i,j}$ denotes the normalized relation between firms i and j, The function $\sigma(\cdot)$ is the sigmoid. We replace the original relation matrix A with the learned adaptive matrix $\widetilde{A}$.

$$\boldsymbol{s}_i^t = \sigma\left(\sum_{j,j\neq i}^{N} \underbrace{\widetilde{A}_{i,j}^{t}\mathcal{W}_s\boldsymbol{h}_j^t}_{\text{spillovers from j to i}}\right). \tag{12}$$

To capture the heterogeneity of stock relationships and spillover dynamics, we employ multi-head attention, where multiple attention heads compute independent adjacency matrices and aggregate features from diverse perspectives. The combined representation for stock i is:

$$\mathbf{s}_i^t = \|_{m=1}^{M}\sigma\left(\sum_{j,j\neq i}^{N} \widetilde{A}_{i,j}^{t,[m]}\mathcal{W}_s^{[m]}\boldsymbol{h}_j^t\right). \tag{13}$$

Finally, we obtain a representation $S_l \in \mathbb{R}^{N\times T_1\times F}$ about the trend term and a representation $S_h \in \mathbb{R}^{N\times T_1\times F}$ about the volatility term, respectively. These representations encapsulate the long-term trends S_l and short-term fluctuations S_h, respectively. However, in financial markets, trends and fluctuations are not independent; they interact dynamically, and their relative dominance varies over time. To capture these interactions and determine which component plays a dominant role, we design a Cross-Attention Mechanism.

4.3.3. Cross-Attention Layer

The cross-attention mechanism enables us to model the mutual influence between S_l and S_h by dynamically estimating how these two components affect each other. Specifically, we treat one component as the query and the other as the key and value pair, enabling a bidirectional interaction. For the trend component S_l influenced by the volatility component S_h, the updated representation $\widetilde{S}_l$ is computed as:

$$\widetilde{S}_l = Softmax\left(\frac{Q_l \cdot K_h^\top}{\sqrt{d}}\right) \cdot V_h, \tag{14}$$

where $Q_l = W_l^q S_l$, $K_h = W_h^k S_h$, $V_h = W_h^v S_h$.

Similarly, the updated representation $\widetilde{S}_h$ for the volatility component influenced by the trend component is:

$$\widetilde{S}_h = Softmax\left(\frac{Q_h \cdot K_l^\top}{\sqrt{d}}\right) \cdot V_l, \tag{15}$$

where Q_h, K_l, V_l are defined analogously.

Finally, we obtain a representation $\widetilde{S}_l$ about the trend term and a representation $\widetilde{S}_h$ about the volatility term, respectively. We add them up to get a spatial–temporal representation of the stock $Z \in \mathbb{R}^{N \times T_1 \times F}$.

$$Z = \widetilde{S}_l + \widetilde{S}_h. \tag{16}$$

4.4. Stock Embedding Aggregation Decoder

In the stock embedding aggregation decoder, a temporal attention layer pools each stock's temporal features to produce the final embedding $\widetilde{Z} \in \mathbb{R}^{N \times F}$ [19]. This step focuses on and consolidates the most informative temporal signals. For stock i with embedding $z_{i,t}$, the last-day embedding $z_{i,T}$ serves as the query over all time steps in the window. The attention weights and the aggregated representation are computed as:

$$\lambda_{u,t} = \frac{exp\left(z_{i,t}^T W_\lambda z_{i,T}\right)}{\sum_{t=1}^{T} exp\left(z_{i,t}^T W_\lambda z_{i,T}\right)}, \tag{17}$$

$$e_u = \sum_{t \in [1,T]} \lambda_{i,t} z_{i,t}. \tag{18}$$

Finally, the stock-level embedding e_u is passed to the predictor $g(\cdot)$ to generate the forecast. We use a linear head for g and train with mean-squared error (MSE).

$$\hat{r}_u = g(e_u), \text{ Loss} = \sum_{i \in \mathcal{S}} MSE(r_i, \hat{r}_i). \tag{19}$$

5. Experiment

5.1. Experiment Setup

5.1.1. Dataset

We evaluate our framework using the Chinese and US stock markets, specifically the CSI300 and S&P500 stock datasets. The selection of these datasets is consistent with recent stock prediction literature, which provides a comparable empirical setting with prior studies and allows us to test the cross-market applicability of the proposed framework [23,31,40]. The time range is from 1 January 2010 to 31 December 2020. We utilize the Alpha158 dataset from the Qlib platform [41], which comprises 158 technical features extracted from price–volume data. The lookback window length T is set to 20. We split the data into training (from 1 January 2010 to 31 December 2017), validation (from 1 January 2018 to 31 December 2018), and test (from 1 January 2019 to 31 December 2020) datasets. We summarize the statistics of these datasets in Table 1. To ensure the quality and consistency of the input data for training and evaluation, we performed several preprocessing steps, including standardization and handling of missing values.

Table 1. Dataset details.

Market	CSI300	S&P500
Stocks	463	507
Training period	1 January 2010–31 December 2017	
Validation period	1 January 2018–31 December 2018	
Test period	1 January 2019–31 December 2020	
Features	158	

5.1.2. Evaluation Metrics

To evaluate the performance of the compared methods, we use the rank information coefficient (RankIC) and RankICIR (Rank Information Coefficient Information Ratio) as metrics, which is the dominant ranking metric in finance. RankIC is the Spearman rank correlation coefficient between the ranking of predicted stock returns and the ranking of realized stock returns. [42,43]. It is computed as the Spearman rank correlation coefficient between the two rankings: $RankIC = Spearman\left(R_{pred}, R_{true}\right)$. Therefore, its value ranges from -1 to 1. A RankIC close to 1 indicates that the predicted ranking is highly consistent with the realized ranking, while a RankIC close to -1 indicates an inverse ranking relationship. A higher RankIC indicates better ranking accuracy, but not necessarily lower numerical prediction error. RankICIR is the ratio of the mean RankIC to its standard deviation, providing a measure of the model's stability and consistency in predictive performance over time. A higher positive RankICIR suggests that the model not only achieves better ranking performance on average but also maintains this performance more consistently across different periods. It can be calculated as: $RankICIR = \frac{mean(RankIC)}{std(RankIC)}$.

5.1.3. Baselines

We evaluate WaveDSTN against a set of representative baselines covering tree ensembles, recurrent and convolutional sequence models, and graph-based methods that are widely used for stock prediction. The baselines were introduced in Related Work and are implemented with comparable training setups. In total, we compare with seven models:

- LightGBM [44]: A gradient-boosted decision-tree method noted for speed, efficiency, and suitability for large datasets.
- LSTM [45]: A gated recurrent architecture designed to retain long-range temporal information, widely adopted for forecasting time series and for NLP tasks.
- GRU [13]: The gated recurrent unit, a lighter alternative to LSTM with fewer gates, usually achieves comparable accuracy with lower computational cost.
- ALSTM [18]: An LSTM variant equipped with an attention module so the model can place higher weight on informative time steps.
- TCN [14]: A temporal convolutional network using causal (often dilated) 1-D convolutions to obtain long receptive fields for sequence modeling.
- ADGATs [39]: A homogeneous graph attention approach that integrates attribute-based aggregation to capture momentum spillovers among related firms.
- STGCN [46]: It captures spatial–temporal correlations by combining GCN with causal convolutions.

5.1.4. Parameter Settings

The proposed WaveDSTN model is implemented using the PyTorch version 1.10.1 and optimized with the Adam optimizer over 100 training epochs, with a batch size of 8. In the model architecture, the multi-head attention mechanism consists of 8 heads, and the base feature dimension d is set to 256. The spatiotemporal encoder comprises 5 layers. The training process starts with an initial learning rate of 0.0001, which is adjusted using a decay scheduler with a rate of 0.1. To reduce overfitting, dropout regularization is applied with a rate of 0.2. To ensure the robustness of our evaluation, each method included in the experiments was trained five times with different random initializations. The average performance across these runs during the testing phase is reported to minimize fluctuations caused by initialization variance. Each training session took approximately 6 h on a single NVIDIA A100 GPU.

5.2. Comparative Analysis of Predictive Performance

Table 2 reports the cross-sectional stock-ranking performance of all compared methods measured by RankIC and RankICIR. Overall, WaveDSTN achieves the best performance on both the CSI300 and S&P500 datasets in terms of RankIC and RankICIR. On the CSI300 dataset, LightGBM achieves a RankIC of 0.012 and a RankICIR of 0.272. Although LightGBM can capture nonlinear feature relationships, its performance is limited because it does not explicitly model temporal dependencies or cross-stock information transmission. Temporal models, including LSTM, GRU, ALSTM, and TCN, achieve higher RankIC values ranging from 0.027 to 0.039, suggesting that modeling historical sequential patterns improves cross-sectional return prediction. However, these models mainly focus on the time-series evolution of individual stocks and do not sufficiently capture dynamic relationships among stocks. Graph-based models perform better, with ADGATs and STGCN achieving RankIC values of 0.046 and 0.042, respectively, which indicates that incorporating cross-stock dependencies and information propagation is useful for return prediction. WaveDSTN further improves upon these baselines, achieving a RankIC of 0.057 and a RankICIR of 0.568 on the CSI300 dataset. Compared with the strongest baseline, ADGATs, WaveDSTN improves RankIC by 0.011 and RankICIR by 0.154, corresponding to relative improvements of 23.91% and 37.19%, respectively. These improvements suggest that WaveDSTN produces predicted returns that are more consistent with realized cross-sectional return rankings and that its predictive signal is more stable across periods. This advantage can be attributed to its decomposition-based spatiotemporal design: instead of modeling stock returns as a single mixed signal, WaveDSTN separates trend and fluctuation components and then models their temporal evolution and cross-stock propagation. A similar pattern is observed on the S&P500 dataset. LightGBM obtains a RankIC of 0.012 and a RankICIR of 0.211, while temporal models achieve RankIC values between 0.017 and 0.029. Graph-based models again show stronger performance, with ADGATs reaching a RankIC of 0.042 and a RankICIR of 0.387. WaveDSTN achieves the highest RankIC and RankICIR, with values of 0.054 and 0.582, respectively. These results show that the advantage of WaveDSTN is consistent across both the Chinese and US markets. Overall, the comparison across model categories indicates that each additional modeling dimension contributes to better performance: temporal models improve upon traditional machine learning by capturing sequential dependence, graph-based models further benefit from cross-stock relational information, and WaveDSTN achieves the best results by jointly modeling multi-scale return information, temporal dependence, and cross-stock interactions. Therefore, the empirical results support the effectiveness of the proposed framework rather than merely showing numerical superiority over individual baselines.

5.3. Robustness Check

To further evaluate the robustness of WaveDSTN under more recent market conditions, we conduct an additional out-of-time test using data from 2021 to 2025. This period differs from the original test period because it includes post-COVID-19 market adjustments, inflationary pressure, monetary policy changes, and heightened geopolitical uncertainty. These factors make stock return prediction more challenging and provide a stricter test of model robustness. Table 3 reports the post-2020 robustness results on the CSI300 and S&P500 datasets. Compared with the original 2019–2020 test period, the overall RankIC and RankICIR values decline for most models. This is consistent with the more challenging market environment after 2020. Although the absolute performance decreases, WaveDSTN still achieves the highest RankIC and RankICIR on both datasets. On the CSI300 dataset, WaveDSTN obtains a RankIC of 0.046 and a RankICIR of 0.471, outperforming the strongest baseline, ADGATs, which achieves a RankIC of 0.037 and a RankICIR of 0.356. On the

S&P500 dataset, WaveDSTN achieves a RankIC of 0.047 and a RankICIR of 0.483, also higher than ADGATs, which obtains 0.036 and 0.265, respectively. These results suggest that WaveDSTN maintains a relative advantage in cross-sectional ranking performance during the post-2020 period, although the prediction task becomes more difficult under more volatile market conditions. Therefore, the robustness test supports the generalizability of the proposed framework, while also showing that market regime changes can reduce the absolute level of predictive performance.

Table 2. Cross-sectional returns prediction performance of the compared methods.

Datasets	Model	RankIC	RankICIR
CSI300	LightGBM	0.021 ± 0.0018	0.272 ± 0.018
	LSTM	0.027 ± 0.0021	0.322 ± 0.021
	GRU	0.029 ± 0.0020	0.331 ± 0.020
	ALSTM	0.034 ± 0.0024	0.340 ± 0.024
	TCN	0.039 ± 0.0019	0.334 ± 0.022
	ADGATs	0.046 ± 0.0018	0.414 ± 0.026
	STGCN	0.042 ± 0.0028	0.375 ± 0.024
	WaveDSTN	0.057 ± 0.0023	0.568 ± 0.022
S&P500	LightGBM	0.012 ± 0.0023	0.211 ± 0.017
	LSTM	0.018 ± 0.0021	0.252 ± 0.019
	GRU	0.017 ± 0.0019	0.261 ± 0.018
	ALSTM	0.026 ± 0.0022	0.318 ± 0.021
	TCN	0.029 ± 0.0024	0.306 ± 0.020
	ADGATs	0.042 ± 0.0017	0.387 ± 0.025
	STGCN	0.034 ± 0.0026	0.325 ± 0.022
	WaveDSTN	0.054 ± 0.0022	0.582 ± 0.021

Each model is trained five times with different random seeds. We report the mean and standard deviation of RankIC and RankICIR across these independent runs. Statistical significance is evaluated by paired *t*-tests between WaveDSTN and the strongest baseline on each dataset. The *p*-values for the *t*-tests were all less than the critical confidence value (0.05), indicating that the superior performance of the proposed approach is statistically significant.

Table 3. Post-2020 robustness results for cross-sectional stock return prediction.

Datasets	Model	RankIC	RankICIR
CSI300	LightGBM	0.012 ± 0.0016	0.118 ± 0.018
	LSTM	0.019 ± 0.0021	0.214 ± 0.023
	GRU	0.020 ± 0.0020	0.226 ± 0.022
	ALSTM	0.025 ± 0.0025	0.271 ± 0.021
	TCN	0.031 ± 0.0020	0.289 ± 0.024
	ADGATs	0.037 ± 0.0019	0.356 ± 0.019
	STGCN	0.034 ± 0.0028	0.331 ± 0.025
	WaveDSTN	0.046 ± 0.0024	0.471 ± 0.023
S&P500	LightGBM	0.010 ± 0.0023	0.154 ± 0.018
	LSTM	0.015 ± 0.0022	0.205 ± 0.021
	GRU	0.016 ± 0.0020	0.218 ± 0.019
	ALSTM	0.022 ± 0.0018	0.286 ± 0.022
	TCN	0.026 ± 0.0023	0.302 ± 0.020
	ADGATs	0.036 ± 0.0022	0.365 ± 0.024
	STGCN	0.032 ± 0.0027	0.334 ± 0.026
	WaveDSTN	0.047 ± 0.0021	0.483 ± 0.022

Each model is trained five times with different random seeds. We report the mean and standard deviation of RankIC and RankICIR across these independent runs. Statistical significance is evaluated by paired *t*-tests between WaveDSTN and the strongest baseline on each dataset. The *p*-values for the *t*-tests were all less than the critical confidence value (0.05), indicating that the superior performance of the proposed approach is statistically significant.

5.4. Ablation Study

To investigate the effects of different components within the WaveDSTN model, this study compares it against five distinct variants. Each variant removes or modifies a specific component to analyze its contribution to the overall performance. The ablation experiments were conducted on the CSI300 and S&P500 datasets using RankIC and RankICIR as evaluation metrics. The five variants are described as follows:

- "w/o" decomposition: The wavelet decomposition module is removed, and the raw stock return series is used as input. Without separating stock returns into trend (low-frequency) and fluctuation (high-frequency) components, the model processes all features as a unified series.
- "w/o" trend: This variant excludes the trend component (low-frequency) from the input. Only the fluctuation component (high-frequency) is processed by the model.
- "w/o" fluctuation: In contrast to the second variant, this version removes the fluctuation component (high-frequency) from the input, retaining only the trend component (low-frequency).
- "w/o" cross-attention: This variant removes the cross-attention mechanism, preventing the model from capturing the interaction between trend and fluctuation components. The trend and fluctuation are modeled independently without considering their mutual influence.
- "w/o" aggregation: The final temporal aggregation step, which consolidates temporal features into a single embedding for each stock, is removed. Instead, the feature of the last time step is directly used for prediction.

As shown in Table 4, the ablation results provide more direct evidence for the contribution of each component in WaveDSTN. Removing the wavelet decomposition module leads to a clear decline in performance. On the CSI300 dataset, RankIC decreases from 0.057 to 0.041, and RankICIR decreases from 0.568 to 0.415. On the S&P500 dataset, RankIC decreases from 0.054 to 0.038, and RankICIR decreases from 0.582 to 0.382. This result indicates that separating stock returns into trend and fluctuation components helps the model learn more useful return representations than directly modeling the original mixed return series. Removing the trend component causes the largest performance decline on both datasets, with RankIC decreasing to 0.024 on CSI300 and 0.022 on S&P500. This suggests that low-frequency trend information plays an important role in the prediction task. Removing the fluctuation component also reduces performance, with RankIC decreasing to 0.025 on both datasets, indicating that high-frequency fluctuation information provides complementary predictive signals. The removal of the cross-attention mechanism also weakens model performance, with RankIC decreasing from 0.057 to 0.049 on CSI300 and from 0.054 to 0.044 on S&P500. This suggests that modeling the interaction between trend and fluctuation components is useful, rather than treating the two components as fully independent inputs. Finally, removing the temporal aggregation module leads to a smaller but consistent decline, indicating that aggregating historical temporal representations contributes to the final stock embedding. Overall, the ablation results show that the performance of WaveDSTN is not driven by a single component alone. Instead, the results support the combined design of wavelet-based decomposition, trend–fluctuation interaction, and temporal aggregation for stock return prediction.

5.5. Hyperparameter Sensitivity Analysis

To assess the impact of hyperparameter choices on the performance of the WaveDSTN model, we conducted a sensitivity analysis focusing on four critical hyperparameters within the Dual-Path Spatiotemporal Encoder and overall architecture.

Table 4. Ablation study.

Datasets	Model	RankIC	RankICIR
CSI300	"w/o" decomposition	0.041 ± 0.0024	0.415± 0.021
	"w/o" trend	0.024 ± 0.0025	0.281± 0.023
	"w/o" fluctuation	0.025 ± 0.0023	0.325± 0.022
	"w/o" cross-attention	0.049 ± 0.0024	0.478± 0.024
	"w/o" aggregation	0.052 ± 0.0022	0.512± 0.021
	WaveDSTN	0.057 ± 0.0023	0.568 ± 0.022
S&P500	"w/o" decomposition	0.038 ± 0.0025	0.382± 0.022
	"w/o" trend	0.022 ± 0.0024	0.211± 0.024
	"w/o" fluctuation	0.025 ± 0.0023	0.272± 0.021
	"w/o" cross-attention	0.044 ± 0.0022	0.452± 0.023
	"w/o" aggregation	0.045 ± 0.0023	0.484± 0.022
	WaveDSTN	0.054 ± 0.0022	0.582 ± 0.021

Each model is trained five times with different random seeds. We report the mean and standard deviation of RankIC and RankICIR across these independent runs. Statistical significance is evaluated by paired *t*-tests between WaveDSTN and the strongest baseline on each dataset. The *p*-values for the *t*-tests were all less than the critical confidence value (0.05), indicating that the superior performance of the proposed approach is statistically significant.

The hyperparameter sensitivity analysis examines the effects of four key hyperparameters on the performance of WaveDSTN: the number of encoder layers (L), hidden dimension size (d), lookback window size (T), and the number of attention heads (num_heads). In the experiments, one hyperparameter is varied at a time while the others are kept at their default values. Specifically, L is tested over {3, 4, 5, 6, 7}, d over {64, 128, 256, 512}, T over {10, 20, 30, 40}, and num_heads over {2, 4, 8, 16}. As shown in Figure 3, the model is sensitive to these hyperparameter choices. For the number of encoder layers, the best performance is obtained at L = 5, while further increasing the depth does not lead to additional improvement and may introduce redundant model complexity. For the hidden dimension size, d = 256 achieves the best performance. Smaller hidden dimensions may limit the representation capacity of the model, whereas larger dimensions increase computational cost without clear performance gains. For the lookback window size, T = 20 provides the best result, suggesting that this setting captures sufficient historical information while avoiding excessive noise from a longer input window. For the number of attention heads, num_heads = 8 performs best, indicating a suitable balance between relational representation capacity and model complexity. Overall, the sensitivity analysis shows that WaveDSTN performs best under a moderate model size and input window. Excessively small settings may lead to insufficient representation capacity, while overly large settings may introduce unnecessary complexity and noise.

5.6. Investment Simulation

To assess the real-world applicability of WaveDSTN, we construct portfolios based on the predictions of different models and compare their performance through backtesting. We adopt the TopK-Drop strategy to maintain a portfolio on each trading day. Specifically, on trading day t, TopK-Drop constructs an equal-weight portfolio P_t consisting of k stocks selected according to the ranking of predicted returns, subject to the turnover constraint $| P_t \cap P_{t-1} | \geq k - n$. Here, k denotes the number of stocks held in the portfolio, while n controls the maximum number of stocks that can be replaced between two consecutive trading days. In our experiment, we set $k = 50$ and $n = 5$, meaning that at least 45 stocks are retained from the previous portfolio and at most 5 stocks are replaced at each rebalancing period.

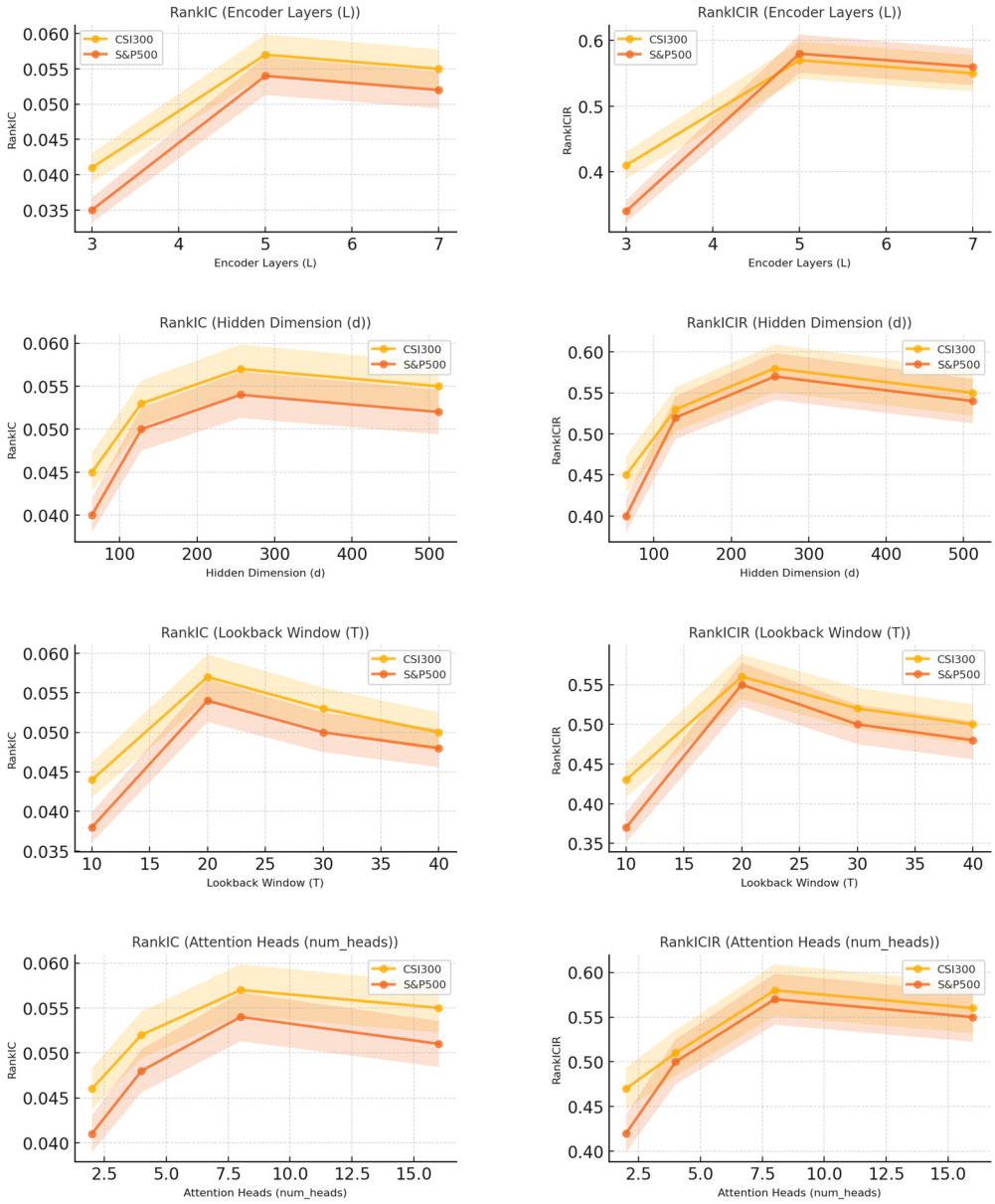


Figure 3. The results of hyperparameter sensitivity.

As shown in Figure 4a,b, the portfolios constructed based on WaveDSTN predictions achieve the highest cumulative returns on both the CSI300 and S&P500 datasets. On the CSI300 dataset, WaveDSTN reaches a cumulative return of more than 60% by the end of 2020, clearly outperforming the baseline models. Graph-based methods such as STGCN and ADGATs achieve relatively stronger performance than traditional sequence models, but their cumulative returns remain lower than those of WaveDSTN. A similar pattern is observed on the S&P500 dataset, where WaveDSTN achieves a cumulative return of more than 80%, while the strongest baseline models remain at a lower level. After adding the CSI300 and S&P500 index benchmarks, the results also show that the portfolio performance of WaveDSTN is not solely driven by the overall market trend, but is related to the predictive signals generated by the model.

Table 5 further reports the annualized return (AR) and Sharpe ratio (SR) of different portfolio strategies. On the CSI300 dataset, WaveDSTN achieves the highest AR of 28% and SR of 2.3, compared with 26% and 2.1 for the strongest baseline, ADGATs. On the S&P500 dataset, WaveDSTN also performs best, with an AR of 35% and an SR of 3.1, while ADGATs achieve an AR of 27% and an SR of 2.4. These results suggest that WaveDSTN not only

improves cumulative portfolio returns, but also provides better risk-adjusted performance under the adopted TopK-Drop backtesting setting.

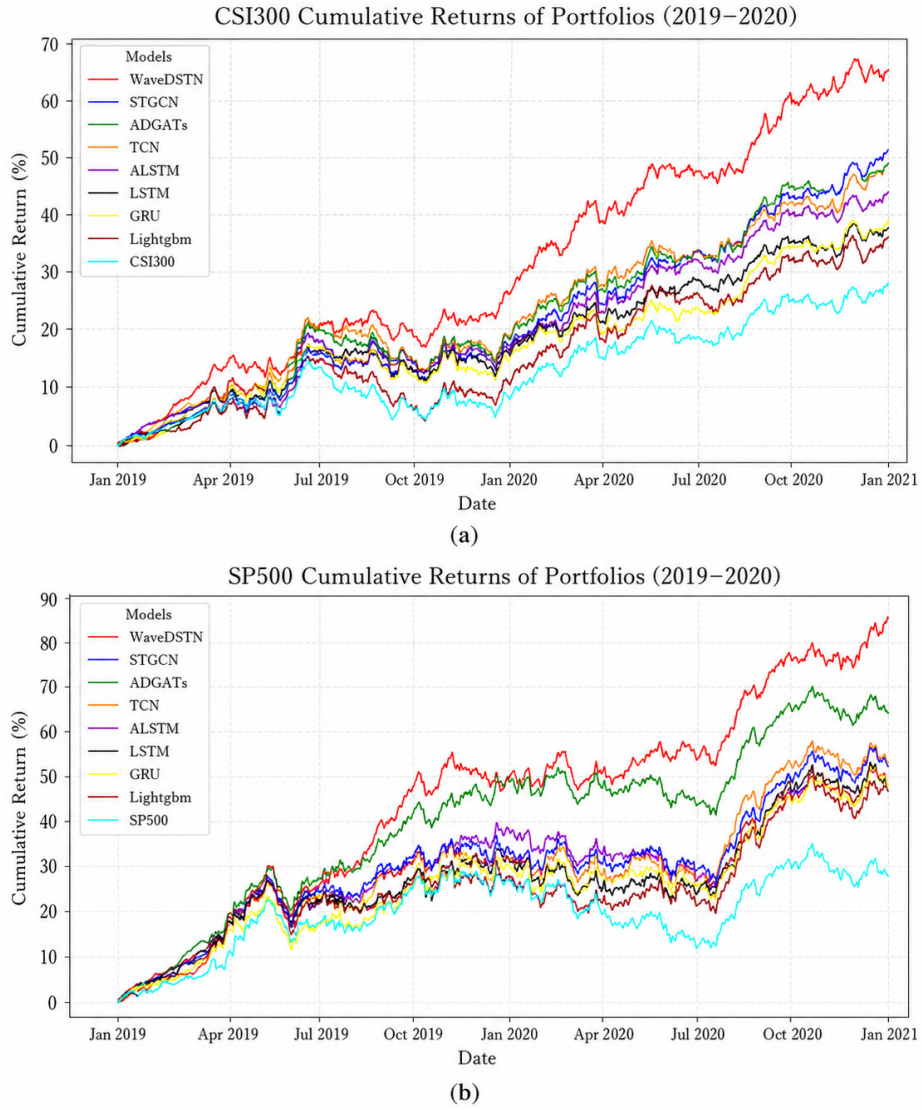


Figure 4. Cumulative return comparison. (**a**) Cumulative return comparison in CSI300; (**b**) Cumulative return comparison in S&P500.

Table 5. The portfolio performance of AR and SR.

Datasets	Model	AR	SR
CSI300	LightGBM	0.10	0.9
	LSTM	0.14	1.1
	GRU	0.15	1.2
	ALSTM	0.17	1.3
	TCN	0.19	1.6
	ADGATs	0.26	2.1
	STGCN	0.23	1.9
	WaveDSTN	0.28	2.3
S&P500	LightGBM	0.12	0.8
	LSTM	0.15	1.0
	GRU	0.16	1.1
	ALSTM	0.18	1.4
	TCN	0.20	1.5

Table 5. *Cont.*

Datasets	Model	AR	SR
S&P500	LightGBM	0.12	0.8
	ADGATs	0.27	2.0
	STGCN	0.25	1.8
	WaveDSTN	0.35	2.5

Overall, the investment simulation indicates that the stronger ranking performance of WaveDSTN can be translated into portfolio-level gains. However, these results should be interpreted within the limits of the experimental setting. The backtest does not fully account for transaction costs, liquidity constraints, market impact, and other implementation frictions. Therefore, the results demonstrate the potential practical value of WaveDSTN for portfolio construction, rather than guaranteeing the same level of performance in real trading environments.

6. Discussion

In this section, we further discuss the effectiveness and limitations of WaveDSTN. The main idea behind WaveDSTN is that stock returns should not be treated as a single homogeneous signal. From a representation learning perspective, directly feeding the original return series into a prediction model may force the model to learn different types of market information within a single mixed representation. However, trend and fluctuation components have different frequency characteristics and noise levels. Low-frequency trend information is usually smoother and more persistent, while high-frequency fluctuation information is more irregular and sensitive to short-term shocks. When these components are not separated, useful low-frequency signals may be disturbed by high-frequency variations, and informative short-term patterns may also be weakened in the learning process. Wavelet decomposition provides a multi-resolution representation of financial time series by separating the original return sequence into low-frequency and high-frequency components. This is useful for predictive learning because it allows the model to construct more specialized representations for different types of market information: the low-frequency component can support the learning of trend-related representations, while the high-frequency component can support the learning of fluctuation-related representations. In this sense, wavelet decomposition is not only a preprocessing step, but also a representation learning strategy that reduces mixed-frequency interference and provides more structured inputs for subsequent spatiotemporal modeling.

Regarding the model's limitations, this study has several limitations that should be acknowledged. First, although our model leverages wavelet decomposition to separate market information, the current framework relies on predefined decomposition scales to construct the trend and fluctuation components. Financial markets contain information at different time scales. Future work could extend the current trend–fluctuation decomposition to a richer multi-scale structure, so that the model can capture short-, medium-, and long-term return patterns more effectively. Second, our current empirical evaluation focuses primarily on predictive accuracy (e.g., MSE, IC) and idealized portfolio returns. In real-world trading environments, factors such as transaction costs, bid-ask spreads, and market slippage could discount the actual profitability of the generated signals. Future studies could integrate reinforcement learning algorithms to directly optimize portfolio construction while explicitly accounting for realistic trading frictions. Third, WaveDSTN relies exclusively on structured market data (i.e., historical price and volume). It does not incorporate unstructured alternative data, such as financial news, corporate filings, or social media sentiment, which are known to be significant drivers of short-term market

fluctuations. Future work could extend the framework into a multi-modal architecture by fusing these unstructured textual signals to further enhance predictive performance.

7. Conclusions

This study aims to improve stock return prediction by addressing two key challenges in financial time-series modeling: the mixed nature of heterogeneous market information and the dynamic propagation of information across stocks. To this end, we propose WaveDSTN, a Wavelet-Decoupled Spatiotemporal Network that decomposes stock returns into trend and fluctuation components and models their temporal evolution, cross-stock propagation, and interaction within a unified framework. The empirical results on the CSI300 and S&P500 datasets show that WaveDSTN achieves better performance than traditional machine learning models, temporal sequence models, and graph-based spatiotemporal baselines in terms of RankIC and RankICIR. The results indicate that WaveDSTN generates predicted return signals with stronger cross-sectional information content and greater temporal stability. The ablation study further confirms the contribution of the main components of the framework. In particular, wavelet decomposition helps separate heterogeneous market information, while the spatiotemporal encoder and cross-attention mechanism improve the modeling of temporal dynamics, cross-stock relationships, and trend–fluctuation interactions. The investment simulation also shows that portfolios constructed based on WaveDSTN predictions achieve stronger cumulative return and risk-adjusted performance under the adopted TopK-Drop strategy.

The significance of this study is twofold. From a methodological perspective, the results show that combining frequency-domain information decomposition with spatiotemporal representation learning can improve stock return prediction. From a practical perspective, the findings suggest that predicted return signals generated by WaveDSTN can be useful for portfolio construction and quantitative investment analysis.

Despite its strengths, WaveDSTN has several limitations. It is currently evaluated on two large-cap equity universes and mainly relies on price–volume-based Alpha158 features, which may limit its applicability to broader markets and information sources. The investment simulation also uses a simplified backtesting setting without fully considering transaction costs, liquidity constraints, short-selling limits, or market impact. Future research could extend the model to broader asset universes, incorporate alternative data, and evaluate its performance under more realistic trading conditions.

Author Contributions: Conceptualization, L.L.; methodology, L.L.; software, C.W.; formal analysis, C.W.; investigation, J.W.; writing—original draft preparation, J.W.; data curation, Y.L. (Yinchao Liao) and Y.L. (Yanjie Lai); visualization, Y.L. (Yinchao Liao); writing—review and editing, Y.L. (Yinchao Liao). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (NSFC) (72471197).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this study were obtained from the open-source Qlib platform. Publicly archived datasets (e.g., Alpha158) are available through the Qlib repository and documentation at: https://github.com/microsoft/qlib and https://qlib.readthedocs.io/, accessed on 8 October 2025. In addition, for reproducibility purposes, the data used in this study have been archived on Zenodo and are available at: https://doi.org/10.5281/zenodo.19150237.

Conflicts of Interest: The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

1. Fan, J.; Shen, Y. StockMixer: A Simple yet Strong MLP-Based Architecture for Stock Price Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 8389–8397. [CrossRef]
2. Qian, H.; Zhou, H.; Zhao, Q.; Chen, H.; Yao, H.; Wang, J.; Liu, Z.; Yu, F.; Zhang, Z.; Zhou, J. Mdgnn: Multi-Relational Dynamic Graph Neural Network for Comprehensive and Dynamic Stock Investment Prediction. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 14642–14650. [CrossRef]
3. Ramsey, J.B. The Contribution of Wavelets to the Analysis of Economic and Financial Data. *Philos. Trans. R. Soc. London Ser. A Math. Phys. Eng. Sci.* **1999**, *357*, 2593–2606. [CrossRef]
4. Sadique, S.; Silvapulle, P. Long-term Memory in Stock Market Returns: International Evidence. *Int. J. Financ. Econ.* **2001**, *6*, 59–67. [CrossRef]
5. Rua, A.; Nunes, L.C. International Comovement of Stock Market Returns: A Wavelet Analysis. *J. Empir. Financ.* **2009**, *16*, 632–639. [CrossRef]
6. Nugroho, F.S.D.; Adji, T.B.; Fauziati, S. Decision Support System for Stock Trading Using Multiple Indicators Decision Tree. In *Proceedings of the 2014 The 1st International Conference on Information Technology, Computer, and Electrical Engineering, Semarang, Indonesia, 8 November 2014*; IEEE: New York, NY, USA, 2014; pp. 291–296.
7. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: San Francisco, CA, USA, 2016; pp. 785–794.
8. Kamble, R.A. Short and Long Term Stock Trend Prediction Using Decision Tree. In *Proceedings of the 2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*; IEEE: New York, NY, USA, 2017; pp. 1371–1375.
9. Xie, B.; Passonneau, R.; Wu, L.; Creamer, G.G. Semantic Frames to Predict Stock Price Movement. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 4–9 August 2013; pp. 873–883.
10. Chen, Y.; Shen, W.; Liu, H.; Cao, X. EMAT: Enhanced Multi-Aspect Attention Transformer for Financial Time Series Forecasting. *Entropy* **2025**, *27*, 1029. [CrossRef] [PubMed]
11. Dong, J.; Liang, S. Hybrid CNN-LSTM-GNN Neural Network for A-Share Stock Prediction. *Entropy* **2025**, *27*, 881. [CrossRef]
12. Nelson, D.M.; Pereira, A.C.; De Oliveira, R.A. Stock Market's Price Movement Prediction with LSTM Neural Networks. In *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2017; pp. 1419–1426.
13. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078. [CrossRef]
14. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271. [CrossRef]
15. Hsu, Y.-L.; Tsai, Y.-C.; Li, C.-T. FinGAT: Financial Graph Attention Networks for Recommending Top-k Profitable Stocks. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 469–481. [CrossRef]
16. Tan, J.; Li, Q.; Wang, J.; Chen, J. FinHGNN: A Conditional Heterogeneous Graph Learning to Address Relational Attributes for Stock Predictions. *Inf. Sci.* **2022**, *618*, 317–335. [CrossRef]
17. Xia, H.; Ao, H.; Li, L.; Liu, Y.; Liu, S.; Ye, G.; Chai, H. Ci-Sthpan: Pre-Trained Attention Network for Stock Selection with Channel-Independent Spatio-Temporal Hypergraph. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 9187–9195. [CrossRef]
18. Wang, Q.; Hao, Y. ALSTM: An Attention-Based Long Short-Term Memory Framework for Knowledge Base Reasoning. *Neurocomputing* **2020**, *399*, 342–351. [CrossRef]
19. Feng, F.; He, X.; Wang, X.; Luo, C.; Liu, Y.; Chua, T.-S. Temporal Relational Ranking for Stock Prediction. *ACM Trans. Inf. Syst.* **2019**, *37*, 1–30. [CrossRef]
20. Zou, Y.; Chen, Q.; Han, J.; Xiao, M. Measuring the Risk Spillover Effect of RCEP Stock Markets: Evidence from the TVP-VAR Model and Transfer Entropy. *Entropy* **2025**, *27*, 81. [CrossRef]
21. David, J.J.; Sabhahit, N.G.; Stramaglia, S.; Matteo, T.D.; Boccaletti, S.; Jalan, S. Functional Hypergraphs of Stock Markets. *Entropy* **2024**, *26*, 848. [CrossRef]
22. Yoo, J.; Soun, Y.; Park, Y.; Kang, U. Accurate Multivariate Stock Movement Prediction via Data-Axis Transformer with Multi-Level Contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual, 14 August 2021*; ACM: Singapore, 2021; pp. 2037–2045.
23. Li, T.; Liu, Z.; Shen, Y.; Wang, X.; Chen, H.; Huang, S. MASTER: Market-Guided Stock Transformer for Stock Price Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2024**, *38*, 162–170. [CrossRef]
24. Lin, Y.; Guo, H.; Hu, J. An SVM-Based Approach for Stock Market Trend Prediction. In *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN)*; IEEE: New York, NY, USA, 2013; pp. 1–7.
25. Huang, W.; Nakamori, Y.; Wang, S.-Y. Forecasting Stock Market Movement Direction with Support Vector Machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [CrossRef]
26. Desokey, E.N.; Badr, A.; Hegazy, A.F. Enhancing Stock Prediction Clustering Using K-Means with Genetic Algorithm. In *Proceedings of the 2017 13th International Computer Engineering Conference (ICENCO)*; IEEE: New York, NY, USA, 2017; pp. 256–261.

27. Waqar, M.; Dawood, H.; Guo, P.; Shahnawaz, M.B.; Ghazanfar, M.A. Prediction of Stock Market by Principal Component Analysis. In *Proceedings of the 2017 13th International Conference on Computational Intelligence and Security (CIS)*; IEEE: New York, NY, USA, 2017; pp. 599–602.
28. Tumminello, M.; Lillo, F.; Mantegna, R.N. Correlation, Hierarchies, and Networks in Financial Markets. *J. Econ. Behav. Organ.* **2010**, *75*, 40–58. [CrossRef]
29. Plerou, V.; Gopikrishnan, P.; Rosenow, B.; Amaral, L.A.N.; Guhr, T.; Stanley, H.E. Random Matrix Approach to Cross Correlations in Financial Data. *Phys. Rev. E* **2002**, *65*, 066126. [CrossRef] [PubMed]
30. Muhammad, S.; Ali, G. The Relationship between Fundamental Analysis and Stock Returns Based on the Panel Data Analysis; Evidence from Karachi Stock Exchange (Kse). *Res. J. Financ. Account.* **2018**, *9*, 84–96.
31. Dong, C.; Cao, Z.; Zhou, S.K.; Liu, J. DFT: A Dual-Branch Framework of Fluctuation and Trend for Stock Price Prediction. *arXiv* **2024**, arXiv:2411.06065. [CrossRef]
32. De Jong, R.M.; Sakarya, N. The Econometrics of the Hodrick-Prescott Filter. *Rev. Econ. Stat.* **2016**, *98*, 310–317. [CrossRef]
33. Gençay, R.; Selçuk, F.; Whitcher, B. *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*; Elsevier: Amsterdam, The Netherlands, 2002.
34. Wu, D.; Wang, X.; Wu, S. A Hybrid Framework Based on Extreme Learning Machine, Discrete Wavelet Transform, and Autoencoder with Feature Penalty for Stock Prediction. *Expert Syst. Appl.* **2022**, *207*, 118006. [CrossRef]
35. Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Biderman, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; et al. *RWKV: Reinventing RNNs for the Transformer Era*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023.
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems*; NeurIIPS Proceedings: San Diego, CA, USA, 2017; p. 30.
37. Duan, Y.; Wang, W.; Chen, Z.; Zhu, X.; Lu, L.; Lu, T.; Qiao, Y.; Li, H.; Dai, J.; Wang, W. Vision-RWKV: Efficient and Scalable Visual Perception with RWKV-Like Architectures. *arXiv* **2025**, arXiv:2403.02308.
38. So, D.; Mańke, W.; Liu, H.; Dai, Z.; Shazeer, N.; Le, Q.V. Searching for Efficient Transformers for Language Modeling. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 6010–6022.
39. Cheng, R.; Li, Q. Modeling the Momentum Spillover Effect for Stock Prediction via Attribute-Driven Graph Attention Networks. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 55–62. [CrossRef]
40. Duan, Y.; Wang, W.; Li, J. FactorGCL: A Hypergraph-Based Factor Model with Temporal Residual Contrastive Learning for Stock Returns Prediction. *Proc. AAAI Conf. Artif. Intell.* **2025**, *39*, 173–181. [CrossRef]
41. Yang, X.; Liu, W.; Zhou, D.; Bian, J.; Liu, T.-Y. Qlib: An AI-Oriented Quantitative Investment Platform. *arXiv* **2020**, arXiv:2009.11189. [CrossRef]
42. Cao, Z.; Xu, J.; Dong, C.; Yu, P.; Bai, T. MATCC: A Novel Approach for Robust Stock Price Prediction Incorporating Market Trends and Cross-Time Correlations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, Boise ID USA, 21 October 2024*; ACM: Boise, ID, USA, 2024; pp. 187–196.
43. Duan, Y.; Wang, L.; Zhang, Q.; Li, J. FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder for Predicting Cross-Sectional Stock Returns. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 4468–4476. [CrossRef]
44. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems*; NeurIIPS Proceedings: San Diego, CA, USA, 2017; p. 30.
45. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
46. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *arXiv* **2017**, arXiv:1709.04875.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.