

Article

STGAT: Spatial–Temporal Graph Attention Neural Network for Stock Prediction

Ruizhe Feng ¹, Shanshan Jiang ^{2,*}, Xingyu Liang ³  and Min Xia ³ 

¹ School of Future Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202283250003@nuist.edu.cn

² School of Management Science and Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China

³ Jiangsu Key Laboratory of Big Data Analysis Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; 202312490016@nuist.edu.cn (X.L.); xiamin@nuist.edu.cn (M.X.)

* Correspondence: jss@nuist.edu.cn

Abstract: Stock price prediction and portfolio optimization are critical research areas in financial markets, as they directly impact investment strategies and risk management. Traditional statistical methods and machine learning approaches have been widely applied to these tasks, but they often fail to fully capture the complex dynamics of financial markets. Traditional statistical methods typically rely on unrealistic assumptions or oversimplified models, neglecting the nonlinear and high-dimensional characteristics of market data. Additionally, deep learning methods, especially temporal convolution networks and graph attention networks, have been introduced in this area and have achieved significant improvements in both stock price prediction and portfolio optimization. Therefore, this study proposes a Spatial–Temporal Graph Attention Network (STGAT) that integrates STL decomposition components and graph structures to model both temporal patterns and asset correlations. By combining graph attention mechanisms with temporal convolutional modules, STGAT effectively processes spatiotemporal data, enhancing the accuracy of stock price predictions. Empirical experiments on the CSI 500 and S&P 500 datasets demonstrate that STGAT outperforms other deep learning models in both prediction accuracy and portfolio performance. The investment portfolios constructed based on STGAT’s predictions achieve higher returns in real market scenarios, which validates the feasibility of spatiotemporal feature fusion for stock price prediction and highlights the advantages of graph attention networks in capturing complex market characteristics. This study not only provides a robust tool for portfolio optimization but also offers valuable insights for future research in intelligent financial systems.



Received: 22 February 2025

Revised: 4 April 2025

Accepted: 10 April 2025

Published: 14 April 2025

Citation: Feng, R.; Jiang, S.; Liang, X.; Xia, M. STGAT: Spatial–Temporal Graph Attention Neural Network for Stock Prediction. *Appl. Sci.* **2025**, *15*, 4315. <https://doi.org/10.3390/app15084315>

Copyright: © 2025 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: portfolio management; stock price prediction; graph attention network; time series analysis

1. Introduction

Stock price prediction plays a crucial role in the field of financial data analysis. As stock prediction and stock portfolio management possess high profits but high risks, investors urgently seek more efficient and effective models to balance returns and risks and benefit from better income growth in uncertain market environments.

Therefore, a large number of models have been developed in optimizing stock investment portfolios. The earliest study of stock price prediction can be traced back to the presentation of the Mean–Variance Model [1] and Black–Litterman Model [2]. Thereafter,

statistical-based forecasting models were introduced for stock price prediction, such as Autoregressive Integrated Models (ARIMA), Vector Autoregression (VAR), the Autoregressive Conditional Heteroskedasticity Model (ARCH), and the Generative Autoregressive Conditional Heteroskedasticity Model (GARCH) [3–5]. However, these models perform forecasting through static patterns of statistical historical series and therefore fail to capture the nonlinear characteristics of stock price series [6].

With the development of machine learning algorithms, researchers started to utilize Support Vector Regression (SVR), random forest (RF), and Principal Component Analysis (PCA) to analyze more complex patterns of stock data [7–10]. With the continuous development of machine learning techniques, deep learning models, particularly neural networks, have shown great potential in analyzing financial data. Lea et al. [11] employed dilated and causal convolutions in temporal neural networks (TCNs) to effectively capture long-term sequence features while enabling the parallelization of the neural network. Machine learning models and deep learning models enhance predictive ability by focusing on sequential feature extraction. However, they fail to capture the potential correlations among different stocks.

Recently, numerous studies revealed that a stock's future price is impacted by the performance of other stocks [12,13], which is both intuitive and insightful. Researchers have begun utilizing graph structures to represent stocks and model the correlations among stocks, providing a structured approach to capturing inter-stock relationships. Jafari and Haratizadeh [14] used a graph convolutional network (GCN) to organize stocks into an influence network that evaluates relationships among stocks based on historical and technical indicators. This approach leverages a semi-supervised framework and plausible label discovery to enhance prediction accuracy by capturing latent inter-stock dependencies effectively.

It is also worth noting that temporal prediction methods without deep learning methods are also being developed at the same time. For instance, hybrid frameworks like ARIMA combined with echo state networks (ESNs) leverage linear models to capture global trends in crude oil price volatility, using the ESNs to refine nonlinear residuals [15]. Seasonal-trend decomposition using Loess (STL) has been applied to isolate price components, enabling feature reconstruction that amplifies latent patterns under efficiency constraints [16]. Meanwhile, state models in a binary-temporal representation explicitly characterize crude oil price dynamics as state transitions governed by binary-temporal patterns, with optimized trend duration and range thresholds to capture transient inefficiencies in high-frequency regimes [17]. These approaches also address challenges in time series analysis, where interpretability is critical.

In this work, we propose a Spatial-Temporal Graph Attention Network (STGAT) that combines a multi-head graph attention network (GAT) with a temporal convolutional network (TCN) to efficiently extract both spatial and temporal features of stock closing prices. The residual attention mechanisms set in both the GAT and TCN enable the network to focus on more relevant features of the stock market by continuously updating node features. By integrating spatial and temporal features, STGAT achieves higher consistency between the predicted outputs and actual data based on the input stock prices. Stock portfolios constructed using the prediction results at different market periods also yield higher returns compared to other neural network models. The results demonstrate that the time series graph attention network proposed in this paper can effectively extract market information and apply it to stock price prediction. The main contributions of this paper are as follows:

1. By employing the Pearson correlation coefficient as a metric for stock correlation, the flattened stock market data are converted into graph data that encapsulate rich

structural information. The graph attention module introduced in this work is used to capture spatial features and identify correlation patterns within the stock market.

2. By applying time series decomposition, the stock price data are separated into trend, seasonal, and residual components. Different parts of the stock information are then used as inputs to distinct modules, leading to improved performance and accuracy in this work.
3. A pioneering graph attention network and temporal convolutional network with a residual attention mechanism are proposed to effectively capture the dynamic spatial features of the stock market. These networks uncover deeper patterns both within and among stocks, providing critical features for stock price prediction.
4. The integration of graph attention network and temporal convolutional network effectively combines spatial and temporal features, resulting in improved performance in stock price prediction and portfolio optimization. This approach offers a new intelligent model for financial portfolio management by striking a reasonable balance between risk and return.

The remaining sections of this paper are organized as follows. Section 2 provides background information and outlines research methods for portfolio optimization across different periods. Section 3 describes the algorithms used in this model and offers an overview of the framework designed in this study, ensuring the overall network's feasibility. Section 4 presents the model's performance compared to other approaches and concludes the paper by summarizing its limitations and suggesting directions for future research.

2. Related Works

The field of stock price prediction holds significant importance in the financial world as it enables investors, analysts, and institutions to make informed decisions. In accurately forecasting stock movements, this domain helps mitigate financial risks, optimize portfolio management, and enhance returns. In recent years, scholars from various disciplines have employed interdisciplinary approaches and diverse information sources to predict stock prices, achieving remarkable results. This section categorizes and presents some representative theories and studies based on the types of techniques used.

2.1. Time Series Analysis Research on Stock Price Prediction

- Statistics-based forecasting models.

The methods based on statistical analysis and machine learning models mainly treat asset indicators as time series data. For example, time series models based on statistical analysis include the Autoregressive Integrated Moving Average model (ARIMA) [3,18,19], Vector Autoregressive model (VAR) [4,20], Autoregressive Conditional Heteroskedasticity Model (ARCH), Generative Autoregressive Conditional Heteroskedasticity Model (GARCH) [5,21,22], etc. Even though these traditional methods are simple and convenient for use, they still face some restrictions when applied to financial market analysis. ARIMA and VAR rely on the assumption of linearity, which prevents them from capturing nonlinear and complex dynamics inherent in stock market data. ARCH and GARCH address some of the limitations of linear assumptions by introducing a dynamic structure for conditional variance, but their applicability remains limited in capturing higher-dimensional and complex nonlinear interactions in stock market data. Moreover, these models also impose strict stationarity requirements, necessitating transformations that can result in the loss of critical information.

- Machine learning models.

Machine learning-based methods, such as Support Vector Machines (SVMs) [7,23,24], Decision Trees, random forests (RFs) [9], and artificial neural networks (ANNs) [25], are highly effective in capturing nonlinear patterns and have gained widespread use in financial market analysis. Vijh and Chandola [9] employed artificial neural networks (ANNs) and random forests (RFs) to predict the next-day closing prices of five companies from various sectors, achieving satisfactory performance as measured using the RMSE and MAPE. Aydin and Cavdar [25] compared the performance of VAR and an ANN in predicting the exchange rate of USD/TRY, gold prices, and BIST 100 index, while the results indicated that the ANN approach has superior performance in prediction capability than the VAR method. However, these models are also prone to overfitting, particularly due to the inherent randomness and uncertainty of market conditions, which can compromise their predictive performance. Additionally, traditional machine learning methods often find it difficult to automatically discover implicit features and rely on manual feature engineering to extract them [26].

- Deep Learning Models.

In recent years, deep learning algorithms became a promising solution replacing mathematical models, especially convolutional neural networks (CNNs) [27–29], Recurrent Neural Networks (RNNs) [30,31], Long Short-Term Memory (LSTM) networks [32], temporal convolution networks (TCNs) [11], and Gated Recurrent Units (GRUs) [31]. Deep neural networks can be considered nonlinear function approximators that are capable of mapping nonlinear functions [33]. Hoseinzade and Haratizadeh [27] proposed the CNNpred framework based on a convolutional neural network (CNN) to improve the accuracy of stock market prediction. Lu and Xu [34] proposed an effective Time Series Recurrent Neural Network (TRNN) for stock price prediction. Selvin [33] compared the predict results of a CNN, an RNN, and LSTM, and the CNN outperformed the RNN and LSTM on stock price data of three representative companies. Liu et al. [32] proposed an optimized ensemble model that combines an LSTM-based attention mechanism and a cyclic multidimensional gray model, utilizing multi-source heterogeneous data, yielding a smaller mean absolute error, Mean Absolute Percentage Error, and root mean squared error than other models. Saud and Shakya [31] utilized an RNN, LSTM, and GRU for predicting the stock prices of the two most popular and strongest commercial banks listed on the Nepal Stock Exchange (NEPSE). Chen et al. [35] proposed iTCN with a multi-kernel parallel convolution structure within a residual layout at the core of a temporal convolution module, to address the low efficiency of the traditional TCN's single kernel convolution in extracting temporal features from input sequences at different time scales.

Even though time series analysis methods nowadays can overcome the defects of traditional statistical models in time series prediction due to their rapid development, they can be easily affected by noise in some complex and dynamic financial systems and ignore the correlations among stocks, making it difficult to mine the hidden features of time series, resulting in poor learning ability and limited prediction accuracy.

2.2. Time Series Decomposition and Stock Price Prediction

A trend denotes a long-run tendency in economic time series, for example, an upward inclination reflecting real growth or cost (price) inflation [36]. In recent years, numerous studies have focused on how to better decompose the time series to keep their formal trend, seasonal features, and bias. Wei et al. [37] developed an Adaptive Network-Based Fuzzy Inference System (ANFIS) based on Empirical Mode Decomposition (EMD). Xu et al. [38] proposed a new causal decomposition method and further applied it to the study of

information flow on different time scales between two financial time series. Tao et al. [39] developed series decomposition layers in the Series Decomposition Transformer with Period Correlation (SDTP) model to further discover relations between historical series and learn the changing trends in the stock market for high forecasting accuracy and generalizability. In this paper, we analyze numerical stock signals using seasonal–trend decomposition to isolate trend, seasonal, and residual components in price dynamics. While the Efficient Market Hypothesis (EMH) posits that such patterns cannot be systematically exploited for prediction, empirical evidence suggests markets occasionally exhibit localized inefficiencies. For instance, Basu found instances of inefficiency and the presence of market anomalies that provide investors with opportunities to earn abnormal returns [40]. Therefore, by decomposing price series, we aim to identify transient structures that may arise from these inefficiencies, thereby offering a framework to explore deviations from strict market efficiency.

2.3. Intrinsic Correlation Research in Stock Price Prediction

Reliable estimates of correlations are absolutely necessary to build a portfolio [41]. To better acquire the stock correlation feature, considerable research has been proposed to better understand the nature of stocks. Pollet et al. [42] showed that the average correlation between daily stock returns predicts subsequent quarterly stock market excess returns, and changes in stock market risk that hold the average correlation constant can be interpreted as changes in the average variance of individual stocks. Song et al. [43] revealed that the average values of the nondiagonal elements of the correlation matrix, correlation-based graphs, and spectral properties of the largest eigenvalues and eigenvectors of the correlation matrix are carrying information about the fast and slow dynamics of the correlation of market indices. Based on the above studies, the correlation between stocks is crucial for stock price prediction. By capturing both the fast and slow dynamics of these correlations, investors can better anticipate market trends and optimize their investment strategies, leading to more informed decision making. From the above perspective, in this work, we utilized the Pearson correlation coefficient as a metric to assess the relationships between stocks. The resulting correlation matrix was then used as an input feature for the subsequent graph attention (GAT) module.

2.4. Graph Neural Network in Stock Price Prediction

To address the limitation that most linear learning methods can only capture temporal features of stock price sequences, the graph neural network (GNN), which is capable of processing non-Euclidean network data and extracting latent interactions in complex systems, was introduced into the field of stock price prediction to extract spatial features of stock relationships [44–46]. Long et al. [47] utilized the knowledge graph and graph embedding techniques to select the relevant stocks of the target for constructing the market and trading information. Xiang et al. [13] proposed a temporal and heterogeneous graph neural network-based (THGNN) approach to learn the dynamic relations among price movements in financial time series.

In previous studies, the GNN was normally used to generate graph data from multi-source financial information available on social media, news articles, and blogs. However, unfortunately, the quality, trustworthiness, and comprehensiveness of online content related to the stock market vary drastically, and a large portion of market information consists of low-quality news, comments, or even rumors [48]. This makes it challenging to rely solely on such data for accurate predictions. Hence, in this work, we use graph structures based on stock prices to describe dynamic stock correlations.

2.5. Graph Attention Mechanism and Stock Price Prediction

With the introduction of the graph attention mechanism, nodes can now attend to the features of their neighbors by stacking layers that assign different weights to each node in a neighborhood. This is achieved without the need for expensive matrix operations (like inversion) or prior knowledge of the graph structure [49]. Building on the ability of GAT to dynamically adjust attention weights, many scholars have applied it to the field of stock price prediction, leveraging the model's capacity to capture complex dependencies and temporal relationships between different stock features. Gao et al. [12] combined a Time-aware Relational Attention Network (TRAN) that integrates historical sequences, company description documents, and a stock relation graph to rank stocks by return ratios, achieving superior investment return ratios. Su et al. [6] proposed an attention-based adaptive spatial–temporal hypergraph convolutional network to dynamically filter out invalid associations from traditional spatial attention. Lei et al. [50] proposed DR-GAT to track the evolution of intercorporate relationships over time for stock recommendation. Inspired by the above studies, our work designed a unique GAT module that excels in effectively capturing correlations between different stock features while also leveraging residual connections to capture additional, more detailed features.

3. System Framework

This study proposes a novel portfolio decision-making framework utilizing a Spatial–Temporal Graph Attention Network (STGAT). As illustrated in Figure 1, the architecture employs a seasonal–trend decomposition approach (see details in Section 3.1) to dissect raw stock temporal data into distinct components encapsulating diverse market information. Subsequently, modules, including a graph attention module (see details in Section 3.3.1), a temporal convolution module (see details in Section 3.3.2), and a multi-layer perceptron (MLP), are applied to process these decomposed components. The integration of these processed features yields a comprehensive spatial–temporal representation that simultaneously captures inter-stock correlations and temporal dynamics. The model culminates with a secondary feature refinement through an additional temporal convolution module and an output module, ultimately generating precise stock price predictions.

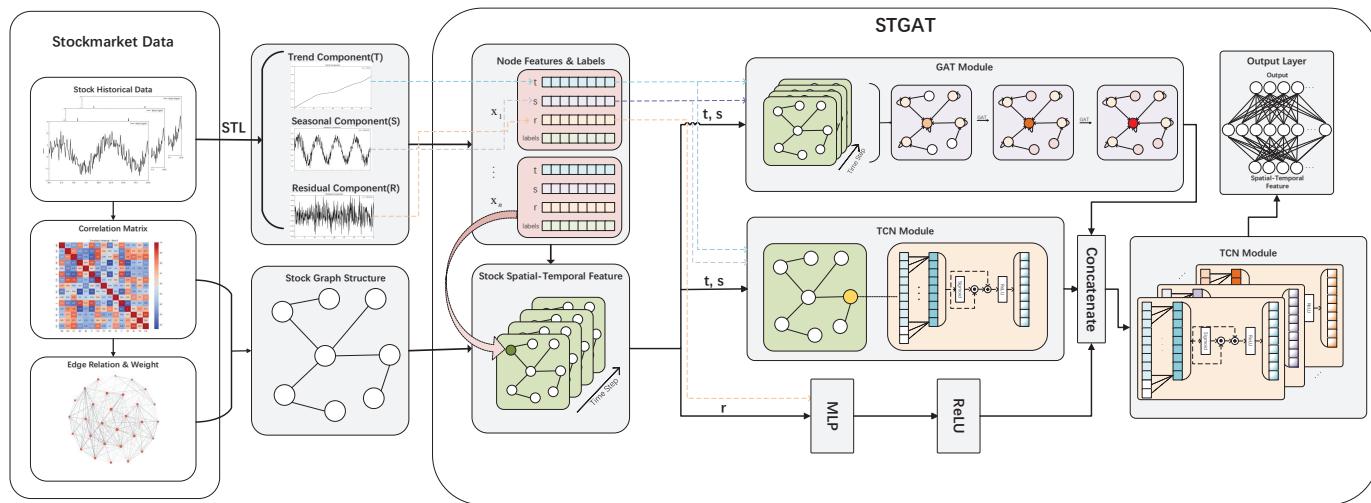


Figure 1. Architecture of STGAT.

Additionally, the prediction result and stock volatility analysis result are further combined as a convincing indicator with which to evaluate a stock's potential and select low-risk stocks as portfolios to simulate the trading process in a test dataset. Judging by

the results, STGAT effectively selects those stocks that balance high return potential with minimized investment risk and achieves high cumulative returns, offering a sophisticated approach to portfolio optimization.

3.1. Seasonal–Trend Decomposition Based on Loess

The stock market is influenced by a multitude of factors that are both diverse and complex in nature, encompassing long-term trends, cyclical patterns, and irregular fluctuations. These factors vary significantly in their characteristics, making it challenging to analyze and model stock price movements comprehensively.

1. Long-term Trends: The trend component reflects the long-term directional movement of stock prices, capturing factors such as company fundamentals, market sentiment, and broader economic cycles. It represents the underlying value changes of a stock over an extended period, smoothing out short-term fluctuations to reveal the overall trajectory—whether upward, downward, or sideways.
2. Cyclical patterns: The seasonal component captures the periodic fluctuations in stock prices, highlighting patterns that repeat at regular intervals. These fluctuations are often driven by time-related factors such as quarterly earnings reports, annual holidays, or monthly economic data releases. By isolating these cyclical patterns, the seasonal component helps identify predictable market behaviors tied to specific timeframes.
3. Irregular fluctuations: The residual component represents the random noise and unpredictable events in the stock market, including sudden market shocks, unexpected news, or shifts in investor sentiment. While it is an important feature of the market, its inherent unpredictability makes it challenging to model or forecast. After separating the trend and seasonal components, the residual must be decomposed and retained to account for the impact of these irregular and often significant market movements, ensuring a more comprehensive understanding of stock price dynamics.

To address this complexity, it is essential to decompose the stock price data into distinct components that capture different aspects of its behavior. In this study, we employed seasonal and trend decomposition using the Loess method to decompose the original stock price data into trend, seasonal, and residual components. This decomposition enables a more detailed analysis of the underlying forces driving stock price dynamics, facilitating a deeper understanding of long-term trends, periodic patterns, and irregular noise. By isolating these components and using the predictable trend and seasonal elements for training in the graph attention and temporal convolution modules, while retaining the residual features through an MLP, STGAT's predictive accuracy and interpretability are enhanced.

3.1.1. Loess

Loess, as a nonparametric smoothing method, employs iterative reweighting to robustly fit local regression surfaces, effectively capturing trends and seasonal variations in data. $\hat{g}(x)$, as the Loess regression curve used for smoothing, is computed in the following way.

Given an independent time series $T = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, each x_i means time and y_i stands for the corresponding value. First, choose a positive integer q as the number of values that are closest to x , and then calculate the neighborhood weight W for x_i according to the distance between x_i and x when $q \leq n$:

$$W(u) = \begin{cases} (1 - u^3)^3 & \text{for } 0 \leq u < 1 \\ 0 & \text{for } u \geq 1. \end{cases} \quad (1)$$

In Equation (1), u represents the distance measure between point x_i and x , and $v_i(x)$ is the neighborhood weight of x_i , which is calculated as follows:

$$v_i(x) = W\left(\frac{|x_i - x|}{\lambda_q(x)}\right). \quad (2)$$

In Equation (2), $\lambda_q(x)$ stands for the distance between the center and edge point of the sliding window. Therefore, the longer $\lambda_q(x)$ is, the smaller the weight value $v_i(x)$ is.

3.1.2. Cycling Steps

STL consists of two main procedures: an inner loop and an outer loop. The inner loop calculates and smooths the trend and seasonal components, while the outer loop regulate robustness in the next internal cycle. In taking the k th iteration of the inner loop as an example, the inner loop is computed as follows:

1. Detrending: Obtain the detrended time series D :

$$D = Y_v - T_v^k, \quad (3)$$

where Y_v represents the original value, and T_v^k is the trend component obtained in the k -th iteration.

2. Cycle Subseries Smoothing: Divide D from Step 1 into cycle subseries and regress them using LOESS. The result is denoted as C_v^{k+1} .
3. Low-Pass Filtering: Apply a low-pass filter consisting of three steps:
 - A moving average of length n ;
 - A moving average of length 3;
 - A Loess regression with $d = 1$ and $q = n$.
 The result is denoted as L_v^{k+1} .
4. Obtaining Seasonal Components: Compute the seasonal series S_v^{k+1} by subtracting low-pass components:

$$S_v^{k+1} = C_v^{k+1} - L_v^{k+1} \quad \text{for } v = 1 \text{ to } n. \quad (4)$$

5. Deseasonalizing: Obtain the deseasonalized series by subtracting the seasonal component S from the original series Y .
6. Trend Smoothing: Apply Loess regression to the result of Step 5 to obtain the trend series T_v^{k+1} .

The inner loop terminates if it converges, and the random term is obtained. Subsequently, the outer loop updates the robustness weights through a well-defined weighting function; this adjustment mechanism effectively mitigates the influence of outliers, thereby ensuring the robustness of the fitting process before initiating the next internal cycle:

1. Execute the inner loop.
2. Calculate residual component R_v :

$$R_v = Y_v - T_v - S_v \quad (5)$$

3. Calculate robustness weight ρ_v :

$$\rho_v = B\left(\frac{|R_v|}{h}\right), \quad \text{where } h = 6 \times \text{median}(|R_v|). \quad (6)$$

The bisquare weight function $B(u)$ is defined as

$$B(u) = \begin{cases} (1 - u^2)^2 & \text{for } 0 \leq u < 1, \\ 0 & \text{for } u \geq 1. \end{cases} \quad (7)$$

3.2. Asset Correlation

In the dynamic and volatile stock market, each stock exhibits unique fluctuations influenced factors such as macroeconomic conditions, industry trends, and company fundamentals. The most intuitive way to reveal the dynamic changes in overall market performance is using the stock's relation graph. The returns of various assets in the stock market are intricately interrelated, forming a complex and highly refined dependency structure. Treating stocks as individual nodes and evaluating the relationships between stocks as edges using specific metrics allows us to intuitively capture the complex interdependencies between stocks and leverage a graph attention network to deeply explore hidden spatial-temporal features in the data, thereby achieving accurate analysis and prediction of market dynamics. Therefore, we chose Pearson's Coefficient (PCC) as the metric to measure the correlation between stocks:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}, \quad (8)$$

where r_{xy} is the PCC between stocks x and y , \bar{x} and \bar{y} mean the average prices of stocks x and y . PCC measures the strength and direction of the linear relationship between two stocks' price movements, which ranges between -1 and 1 . To construct graph structures suitable for deployment in a graph attention network, non-negative weights are typically required. To address this, the original correlation coefficients are scaled using the Sigmoid function, mapping them into the interval $[0, 1]$. Shown as heatmaps in Figure 2, the closer the value of r_{xy} is to 1 , the stronger the correlation between the two stocks; conversely, the closer it is to 0 , the weaker the correlation.

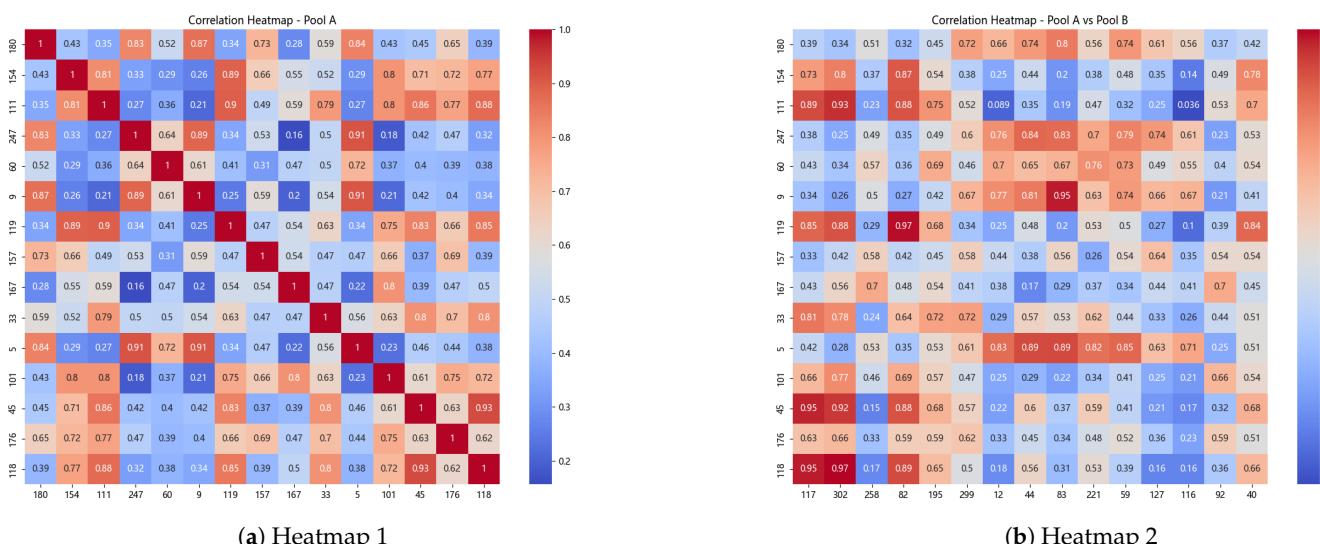


Figure 2. Partial correlation among stocks.

Based on the scaled value of the correlation coefficient, this article analyzes the correlation matrix of the stock market and categorizes the correlations between stocks into strong and weak relations. A threshold value, denoted as $\alpha = 0.4$, is applied to differentiate these relations. This threshold—which aligns with the conventional classification of

Pearson's correlation coefficients in economics and social sciences, where values below 0.4 typically indicate weak associations—distinguishes weak linkages from economically meaningful interactions [51,52], ensuring robustness against random noise inherent in efficient markets. If the scaled value of the correlation coefficient between two stocks exceeds α , the relationship is considered strongly correlated. Strong correlations typically indicate that stocks are influenced by shared macroeconomic factors, industry trends, or market sentiment. In contrast, when the correlation coefficient is below α , the stocks are deemed weakly correlated. Hence, in the graph representation, edges are added between pairs of strongly correlated stocks, and the original edge weights are set equal to their corresponding correlation coefficients. The adjacency matrix A' of the graph is derived from the asset correlations below:

$$A'_{ij} = \frac{1}{1 + e^{-A_{ij}}}, \quad A_{ij} = r_{ij} \quad \text{if } r_{ij} > \alpha. \quad (9)$$

3.3. Spatial–Temporal Graph Attention Neural Network

This article proposed a Spatial–Temporal Graph Attention Neural Network (STGAT) to extract features from decomposed temporal data and graph features preprocessed, respectively, in Sections 3.1 and 3.2. Through fusing spatial dependency information and temporal dynamic price fluctuation extracted using a graph attention module (GAT module) and temporal convolution module (TCN module), the framework can simultaneously capture the spatial and temporal features in the stock price prediction task.

Due to the long time span of the dataset used in this experiment, 20 consecutive trading days were treated as a time step to facilitate the construction of graph data. STGAT takes decomposed temporal components from one time step as temporal data, including the trend feature $T_i = (t_1^i, t_2^i, \dots, t_{20}^i)$, $i = 1, 2, \dots, m$, seasonal feature $S_i = (s_1^i, s_2^i, \dots, s_{20}^i)$, $i = 1, 2, \dots, m$, and residual feature $R_i = (r_1^i, r_2^i, \dots, r_{20}^i)$, $i = 1, 2, \dots, m$. The temporal data above serve as input features for the i_{th} stock, which functions as a node within a single graph. Subsequently, the trend and seasonal features of stocks, embedded within the graph structure, are separately fed into a graph attention module and a temporal convolution module. This dual-pathway approach enables the model to further capture the intrinsic spatiotemporal features inherent in the stock data.

Notably, due to the inherent unpredictability of the residual feature, which often contains noise or random fluctuations that cannot be explained by trend or seasonal patterns, it is filtered out and processed separately by a multi-layer perceptron (MLP). This approach aims to mitigate the potential negative impact of noise while preserving the useful features in residuals.

3.3.1. GAT Module Based on Residual Mechanism

The graph attention module utilizes attention mechanisms [53,54] to assign varying levels of importance to nodes within the same neighborhood, significantly enhancing model capacity while improving interpretability [49]. In this work, we adopted an enhanced graph attention network (GAT) to optimize information propagation beyond the capabilities of the original GAT framework. Specifically, we implemented a multi-layer GAT architecture augmented with residual connections, which effectively aggregates stock price feature information from strongly correlated nodes. As illustrated in Figure 3, the model stacks two GAT blocks to extract preliminary graph relationship features, while an additional GAT layer serves as a residual component. The residual connection preserves the initial stock features, preventing feature degradation caused by the stacking of multiple graph attention layers. This design ensures that both high-level relational patterns and low-level intrinsic features are retained, enhancing the model's ability to capture complex market dynamics.

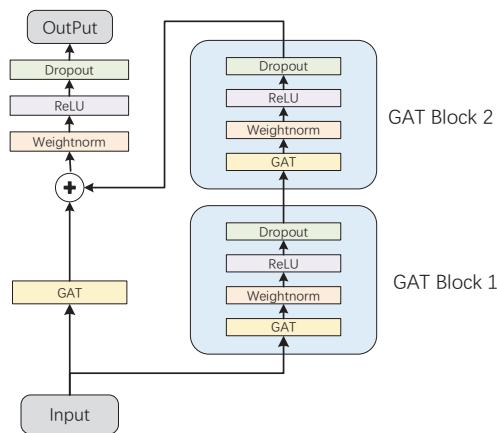


Figure 3. GAT module's framework.

In each GAT layer, $h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}, \vec{h}_i \in R^F$, represents a set of node features, N represents the number of nodes, and F is the number of features in each node. The layer's output is a set of new features $h' = \{\vec{h}'_1, \vec{h}'_2, \dots, \vec{h}'_N\}, \vec{h}'_i \in R^{F'}$; thus the GAT layer needs to first convert h into features with dimension $R^{F'}$ by applying linear transformation with a weight matrix $W \in R^{F \times F'}$ to enhance its expressive capacity. In Equation (10), \parallel represents concatenation, and α denotes normalized attention coefficients, as shown in Figure 4. The equation represents the importance of node j 's features to node i :

$$e_{ij} = \alpha(W\vec{h}_i \parallel W\vec{h}_j), \quad (10)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i \parallel W\vec{h}_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(\vec{a}^T [W\vec{h}_i \parallel W\vec{h}_k]))}. \quad (11)$$

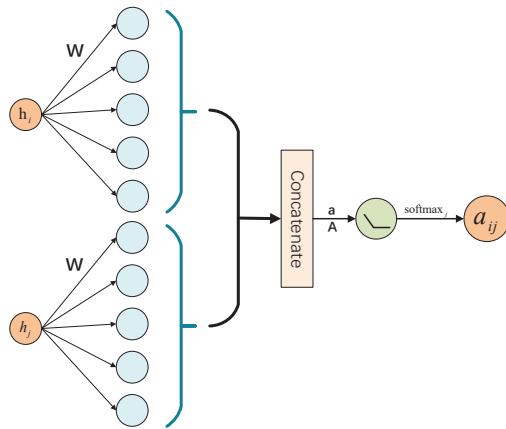


Figure 4. Graph attention mechanism.

As shown in Figure 5, to better enhance the model's ability to capture diverse patterns in the data, the multi-head mechanism [55,56] extends the standard attention framework by employing multiple independent attention heads, each with its own set of parameters. In doing so, the model can simultaneously focus on different aspects of the stock data, leveraging distinct subspaces within the multi-head matrix to represent information from various perspectives.

Specifically, each of the K -independent attention heads performs a transformation as defined in Equation (12). These features generated in each head are then concatenated to form a comprehensive output, which integrates the diverse attention information captured by each head. Combined with the normalized attention coefficients, this mechanism

effectively avoids information redundancy, enabling the model to efficiently learn the relationships among critical nodes in the graph.

$$\vec{h}'_i = \left\|_{k=1}^K \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \vec{h}_j \right) \right\|.$$
 (12)

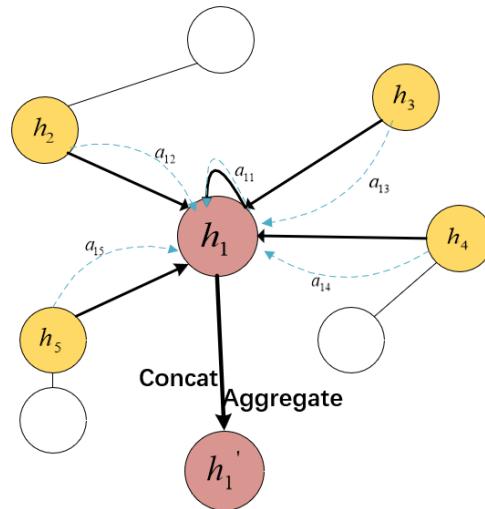


Figure 5. Multi-head attention mechanism.

Through the multi-head attention mechanism, node i can effectively aggregate the features of its neighboring nodes, enriching the local feature representation and capturing the overall graph structure more comprehensively. This approach not only enhances the model's ability to learn complex graph structures but also improves its robustness and generalization performance. The output feature generated by the GAT module includes both the high-level relationships between stocks and the subtle variations in stock correlations, providing a more refined understanding of the market's dynamics.

3.3.2. TCN Module Based on Residual Mechanism

The TCN module proposed in this work mainly uses one-dimensional convolutional layers to process time series data. Based on time series data x_1, x_2, \dots, x_t , the TCN predicts y_t by sliding on the time dimension to capture the features and the time dependency relationship in the input sequence. Notably, the y_t denotes univariate outputs that depend solely on the causal constraints of x_1, x_2, \dots, x_t . Therefore, the value of time t only convolves with values before t , which ensures every output in the time series only relates to previous input values and current time. This mechanism avoids data leakage, and the output value obtained from causal convolution at x_t is:

$$(F * X)_{x_t} = \sum_{k=0}^K f_k x_{t-k},$$
 (13)

where K represents the filter length, which determines the number of past time steps considered in calculating the current output. The weights of the filter $F = (f_1, f_2, \dots, f_K)$ specify the influence of the input at each previous time step, x_{t-k} , on the current output. As shown in Figure 6, each TCN module consists of three convolutional layers to transmit features in parallel, utilizing a residual mechanism to extract temporal information and weaken redundant information sufficiently.

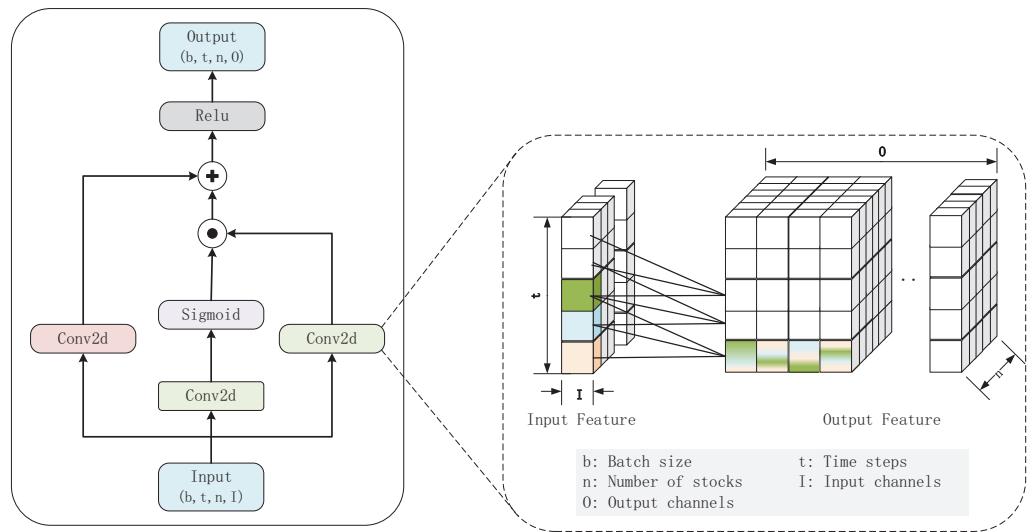


Figure 6. TCN module's mechanism.

In STGAT, two TCN modules are applied for different functions. The first module processes the time series price data of all stocks from 20 trading days. In the initial layer, temporal features are extracted from the raw stock price data, resulting in a preliminary representation of these temporal patterns. The second and third layers work together as residual blocks, focusing on capturing finer details of the stock price fluctuations. The output from the second layer provides the early-stage residual features, while the feature map from the third layer is processed through a sigmoid function. This enables the calculation of attention weights, which are then used to generate a corresponding weight map that highlights the most influential features for further processing. Multiplying the residual features generated by the second layer with the feature map generated by the third layer results in a weighted feature map, which functions as residual information, significantly weakening the impact of redundant information. The final output feature map includes both the features of the temporal features and weighted feature map, enabling the model to capture deeper temporal features hidden in input data and exclude redundant information. The process of the first TCN module can be modeled as

$$Y = TCN_1(X) + TCN_2(X) \odot \sigma(TCN_3(X)), \quad (14)$$

where Y is the output feature, and $\sigma(\cdot)$ is the sigmoid function.

The second module fuses spatial and temporal features, respectively, generated by the GAT module and first TCN module. This operation of combining features enhances the model's robustness and predictive ability in dealing with highly uncertain market environments. The second temporal convolution module can be expressed as

$$Y_{ST} = TCN_4(X_{fused}) + TCN_5(X_{fused}) \cdot \sigma(TCN_6(X_{fused})), \quad (15)$$

where Y_{ST} is the spatial-temporal feature extracted with the GAT and TCN, and $X_{fused} = concat(A, Y, dim = time)$ functions as a preliminary fused feature. After extracting features from the stock price data, STGAT uses linear layers to map the fused data to the final dimensions required for prediction, producing the predicted prices for all stocks in the pool, which can then be used for trading decisions.

3.4. Portfolio Establishment

To evaluate the practicability of STGAT, based on the prediction results generated by the model, this work establishes portfolios by constructing the investment value coefficient

shown in Equation (16), a metric designed to comprehensively evaluate the ability of stocks in the portfolio to balance risk avoidance and profit generation.

$$\begin{cases} ER_i = \frac{Y_{t+1}^i - X_t^i}{X_t^i} \\ V_i = \frac{ER_i}{Std(X^i)}, \end{cases} \quad (16)$$

where ER_i denotes the expected return of the i -th stock, and t represents the current time step. The term Y_{t+1} is the prediction result of STGAT, while X_t represents the known stock price at the current time step, and $Std(X^i)$ is the standard deviation of the i -th stock. A higher coefficient value indicates greater investment potential in the stock, characterized by higher returns and lower risk.

When constructing the investment portfolio and determining the corresponding investment allocations, the top ten stocks with the highest value coefficients are selected as the portfolio components, provided that all trading regulations are adhered to. The investment allocations are then calculated based on the weight distribution of these coefficients as specified in Equation (17):

$$\begin{cases} \zeta = \max_j V_1, V_2, \dots, V_n, j = 10 \\ \omega_j = \frac{V_j}{\sum_{j=1}^{10} V_j}, \quad j \in \zeta, \quad \sum_{j=1}^{10} \omega_j = 1, \end{cases} \quad (17)$$

where ζ is the index of the top 10 stocks with the highest investment value coefficient, and ω_j is the corresponding investment shares of each stock.

4. Experiments

4.1. Datasets and Experimental Setup

In this study, we used the stocks of the CSI 500 Index and the S&P 500 Index sourced from the Choice Financial Terminal as selected stock samples. The CSI 500 Index comprises 500 small and medium companies with capitalization from China's A-share market, covering a wide array of industries. These companies are characterized by substantial growth potential, though they tend to exhibit higher volatility. In contrast, the S&P 500 Index consists of 500 large-cap, highly liquid companies listed on the New York Stock Exchange and NASDAQ, representing key sectors of the U.S. economy. The constituent companies are typically blue-chip stocks, renowned for their stability and strong market presence, with broad representation across industries such as technology, finance, and consumer goods. Daily market data for the stocks in the CSI 500 and S&P 500 indices covered the period from January 2015 to April 2024. This timeframe was selected to capture multiple market regimes, including pre-COVID stability (2015–2019), pandemic-induced volatility (2020–2021), and post-pandemic recovery with monetary policy shifts (2022–2024). Additionally, it aligns with the availability of high-frequency granular data in both A-share and US stock markets after 2015, enabling robust decomposition of trend–seasonal–residual components. The datasets include 2170 and 2330 trading days, respectively.

Neural networks are highly sensitive to data relationships and outliers, and they require complete input data with no missing values. Therefore, it is critical to perform outlier detection and processing on time series data to prevent individual stocks from distorting the overall dataset. Missing values in stock data can stem from issues such as delayed listings, IPO delays, or errors in data reporting. To address this, we exclude ST stocks and those with considerable missing data. After filtering, the dataset for A-shares includes 304 stocks, while the dataset for U.S. stocks contains 462 stocks, as shown in Table 1.

Table 1. Dataset description.

Market Index	Stock Count	Trading Days	Train Dataset	Test Dataset
CSI500	304	2170	1953	217
S&P500	462	2330	2097	233

The experiments in this work were conducted using Python 3.11.8, running on a system with an Intel Core(TM) i5-12400F processor, an NVIDIA GeForce RTX 4060 Ti, and 12 GB of memory. The initial learning rate was set to 1×10^{-3} , and the batch size was configured to 16.

4.2. Dataset Settings

This work used the sliding window method to split time series data and construct graph data. The sliding window had a size of 20 trading days and a step size of 5 trading days, chosen to balance short-term market dynamics and computational efficiency. The 20-day window aligns with the typical monthly trading cycle to capture recurring patterns, while the 5-day step ensures sufficient overlap to smooth transient noise without oversampling redundant signals. After dividing the historical stock price data into 20-day periodic segments, STL was applied to extract the trend, seasonal, and residual components of stock data. Subsequently, all stocks' corresponding periodic data and components were combined with the edge weight matrix generated in Section 3.2 to form a periodic graph structure. As a result, the graph dataset containing primary stock relationship was generated with its decomposition features for supplement.

In this study, we propose two methods for dividing the dataset. These methods were designed to evaluate the performance of the investment portfolio under different market conditions, taking into account the variability and unpredictability inherent in financial markets. The detailed splitting methods are outlined as follows:

- Method I: Dividing the original dataset into a 9:1 ratio based on the time range. The test set is defined as the continuous trading days between September 2023 and April 2024. The performance of the portfolio during this fixed time period is compared with the real index and actual performance of each investment portfolio.
- Method II: This method uses a simulation environment where 10% of non-continuous trading days are randomly selected from the overall dataset's time range as the testing interval. This approach simulates the unpredictable nature of financial markets and helps evaluate the portfolio in different market conditions, ensuring that the model can handle diverse trends in the stock market.

To better evaluate the model's adaptability and risk management capabilities in both the A-share and US stock markets, these methods are applied to simulate the inherent uncertainty of the stock market. Furthermore, this study employed the K-fold time series validation method to better evaluate and enhance the predictive performance of models applied to stock price datasets. Unlike standard cross-validation, K-fold time series validation preserves the temporal order of observations by sequentially partitioning the time series into chronologically ordered folds. This approach ensures the model is trained on historical data and validated on subsequent periods, rigorously simulating real market forecasting scenarios. Therefore, it mitigates overfitting risks inherent in random cross-validation approaches, which could leak future information into training phases when applied to time series data.

Figures 7 and 8 show the loss curves under two cross-validation methods. To be noticed, the loss curve of five-fold time series validation exhibits periodic oscillations at every 100-epoch interval (aligned with fold transitions), which can be attributed to

our strategy of resetting the optimizer state at the beginning of each fold. Resetting the optimizer strictly isolates training dynamics between folds, aligning with the independence principle of cross-validation and avoiding overfitting performance.

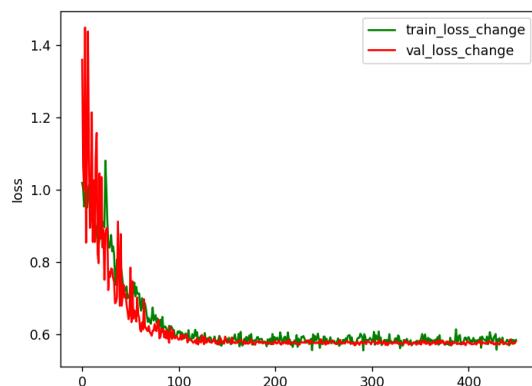


Figure 7. Loss curve generated by cross-validation.

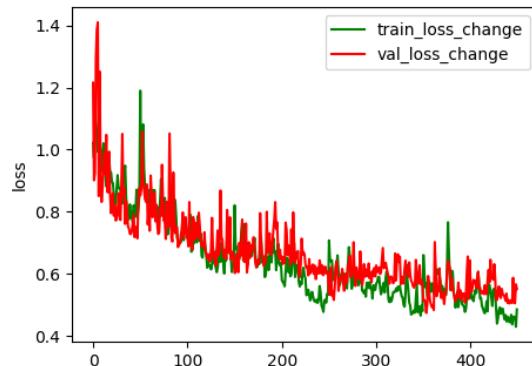


Figure 8. Loss curve generated by K-fold time series validation.

In order to achieve better returns, a dynamic investment strategy is applied to construct portfolios with the highest expected investment values. This strategy aims to optimize capital allocation and ensure that investment decisions are responsive to market dynamics. Based on the STGAT model, predictions generated are used to formulate predictive returns. The model then selects the 10 stocks with the highest predictive returns to form the portfolio for the corresponding day. Additionally, the portfolio construction incorporates a weight allocation mechanism where each stock's predicted return is divided by the total predicted returns of all selected stocks and then normalized to obtain the weight for each stock. After the portfolio is constructed, it is held for five trading days, after which it is re-evaluated and adjusted to pursue sustained maximum returns.

4.3. Experimental Results

4.3.1. Predictive Performance of STGAT

This study introduces various learning models and neural networks to perform a comprehensive comparison of their performance. To assess the predictive capabilities of these models, a diverse set of performance metrics is employed, including mean absolute error (MAE), directional accuracy (DA), relative error (RE), root mean squared error (RMSE), coefficient of determination (R^2), and accuracy in predicting stock price fluctuations. These indicators provide a well-rounded evaluation of the models' effectiveness.

As summarized in Table 2, where y_i is the true stock price and \hat{y}_i is the predicted value, these metrics holistically assess STGAT's capabilities in three dimensions: The MAE, RE, and RMSE collectively measure regression accuracy through distinct error quantification methods—MAE by averaging absolute deviations, RE by normalizing errors relative to

actual prices, and RMSE by emphasizing large errors through squaring. Simultaneously, DA specifically evaluates the alignment between predicted and actual price change directions (Δy_i), a capability essential for generating reliable trading signals. Meanwhile, R^2 quantifies the model's ability to explain stock price variance, serving as a direct indicator of how effectively it captures complex market patterns. The integrated evaluation approach here ensures models are assessed not only on numerical precision but also on the financial decision relevance and interpretability of learned relationships.

Table 2. Indicators of prediction results.

Indicators	Formula	Meaning
MAE	$\frac{1}{n} \sum_{i=1}^n y_i - \hat{y}_i $	Average of absolute errors between predicted and actual values
DA	$\frac{1}{n} \sum_{i=1}^n \begin{cases} 1 & \text{if } \Delta y_i \cdot \Delta \hat{y}_i > 0 \\ 0 & \text{otherwise} \end{cases}$	Proportion of correct directional predictions where $\Delta y_i = y_i - y_{i-1}$ and $\Delta \hat{y}_i = \hat{y}_i - \hat{y}_{i-1}$
RE	$\frac{ y_i - \hat{y}_i }{ y_i }$	Ratio of absolute error to the actual value
RMSE	$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$	Square root of MSE, in the same units as the target variable
R^2	$1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	Proportion of variance explained by the model

In Table 3, a performance comparison between different models based on the A-share dataset is shown. As shown in the table, STGAT demonstrates significant advantages over other baseline models across all evaluated metrics. STGAT achieves consistently lower prediction errors and improved prediction accuracy compared to the other models. As for the other models, time series models like the GRU and TCN unveil impressive prediction accuracy for their ability to effectively capture sequential dependencies in time series data. However, by integrating temporal features with graph structural features, as seen in STGAT, the model not only preserves the temporal relationships but also leverages the spatial dependencies and interactions between different stocks in the graph. This fusion of temporal and graph-based characteristics allows STGAT to capture more complex patterns and relationships within the data, resulting in its superior performance across all metrics. The combination of these features provides a more holistic understanding of the underlying dynamics, which is especially crucial for predicting intricate systems like stock price fluctuations.

Table 4 presents the models' predictive performance using a dataset where training and testing sets were carefully kept entirely independent. Based on this premise, trading days were randomly selected to form the test set, which simulates the randomness and volatility commonly observed in the stock market. From the table, it can be observed that, under this experimental setup, most models show an overall improvement in performance on key indicators, with the exception of GCN. This can be interpreted as an effect of the random data split creating diverse and more independent price trends within the test set, which places a greater demand on the models' ability to reason about temporal features. In this context, models relying solely on graph structure for prediction, such as the GCN, tend to perform poorly because they lack the capacity to effectively capture or infer sequential

dynamics present in the stock price movements. On the other hand, time series models, like the GRU and RNN, benefit from this setup as they are inherently designed to learn and predict based on temporal dependencies, leading to a notable increase in prediction accuracy. STGAT, with its ability to integrate both temporal features and graph-based relationships, demonstrates significant advantages in this scenario. By combining temporal and spatial features, STGAT can identify and utilize the fusion of evolving stock price trends and entity interactions, showcasing its capability to handle the complexity and randomness of stock market data more effectively than models focused on a single aspect.

Table 3. Comparison of predictive performance of different models on A-share dataset (divided using method I). The bold indicates the best performance among all models.

Model	MAE	DA %	RE	RMSE	R2	Accuracy
GCN	3.6217	51.42%	0.2701	5.6336	0.7765	0.5406
GRU	2.0720	53.12%	0.1290	3.9610	0.8895	0.6034
LSTM	2.8745	51.11%	0.1707	5.2883	0.8030	0.5768
MLP	2.3909	51.47%	0.1662	4.1556	0.8784	0.5702
RNN	2.2594	52.70%	0.1362	4.2614	0.8721	0.5941
TCN	2.4904	51.13%	0.1737	4.3004	0.8697	0.6156
Transformer	3.0561	53.21%	0.1878	5.2780	0.8018	0.5845
STGAT	1.5064	53.85%	0.0946	2.6312	0.9460	0.6445

Table 4. Comparison of predictive performance of different models on A-share dataset (divided using method II). The bold indicates the best performance among all models.

Model	MAE	DA	RE	RMSE	R2	Accuracy
GCN	2.9273	54.20%	0.2405	4.9487	0.7544	0.5760
GRU	0.8597	61.71%	0.0659	1.4769	0.9796	0.7367
LSTM	1.0651	59.62%	0.0791	1.9539	0.9643	0.7146
MLP	1.1231	56.11%	0.0878	1.8872	0.9643	0.7143
RNN	0.8785	62.80%	0.0679	1.5475	0.9776	0.7331
TCN	0.8820	60.62%	0.0727	1.4076	0.9801	0.7075
Transformer	1.7136	57.20%	0.1338	2.8327	0.9195	0.6233
STGAT	0.8440	62.77%	0.0682	1.3349	0.9821	0.7443

Tables 5 and 6 show the performance of the models in the US stock market based on the two dataset-partitioning methods mentioned above. Because the US stock market follows the T+0 trading rule and we selected 462 stocks from the S&P 500 index, the stock volatility and dataset complexity in the US market are much more remarkable than in the A-share market. The T+0 trading rule allows for rapid intraday trading, which intensifies short-term price fluctuations as investors execute high-frequency and speculative strategies, leading to heightened volatility. Additionally, the inclusion of 462 S&P 500 stocks forms a large and highly interconnected graph network, increasing the learning difficulty for models as they must capture both intricate temporal patterns and complex stock relationships. These combined factors amplify the challenges in prediction, leading to an overall increase in error indicators when models are applied to the US stock market.

In Table 5, even though the prediction capability of all models declines due to the increased volatility and complexity of the US stock market, STGAT consistently demonstrates the highest robustness and adaptability. Despite the overall performance drop in key metrics, STGAT remains the best-performing model with the lowest errors and the highest R^2 and accuracy values. This indicates that the integration of spatiotemporal features enables STGAT to perform better in prediction with higher robustness and adaptability, which are crucial in handling more volatile and complex datasets. From Table 6, it is evident

that STGAT continues to maintain its superiority under the second dataset partitioning method, achieving the best performance across all metrics. This further validates the importance of spatial–temporal feature integration, as STGAT effectively captures both temporal fluctuations and inter-stock dependencies in a large, interconnected network.

In Figures 9 and 10, it can be seen that STGAT can effectively capture the dynamic trend of each stock, although the prediction may differ from actual prices. This performance highlights the model’s proficiency in extracting spatiotemporal features from financial markets, where it successfully integrates both temporal patterns and cross-asset correlations.

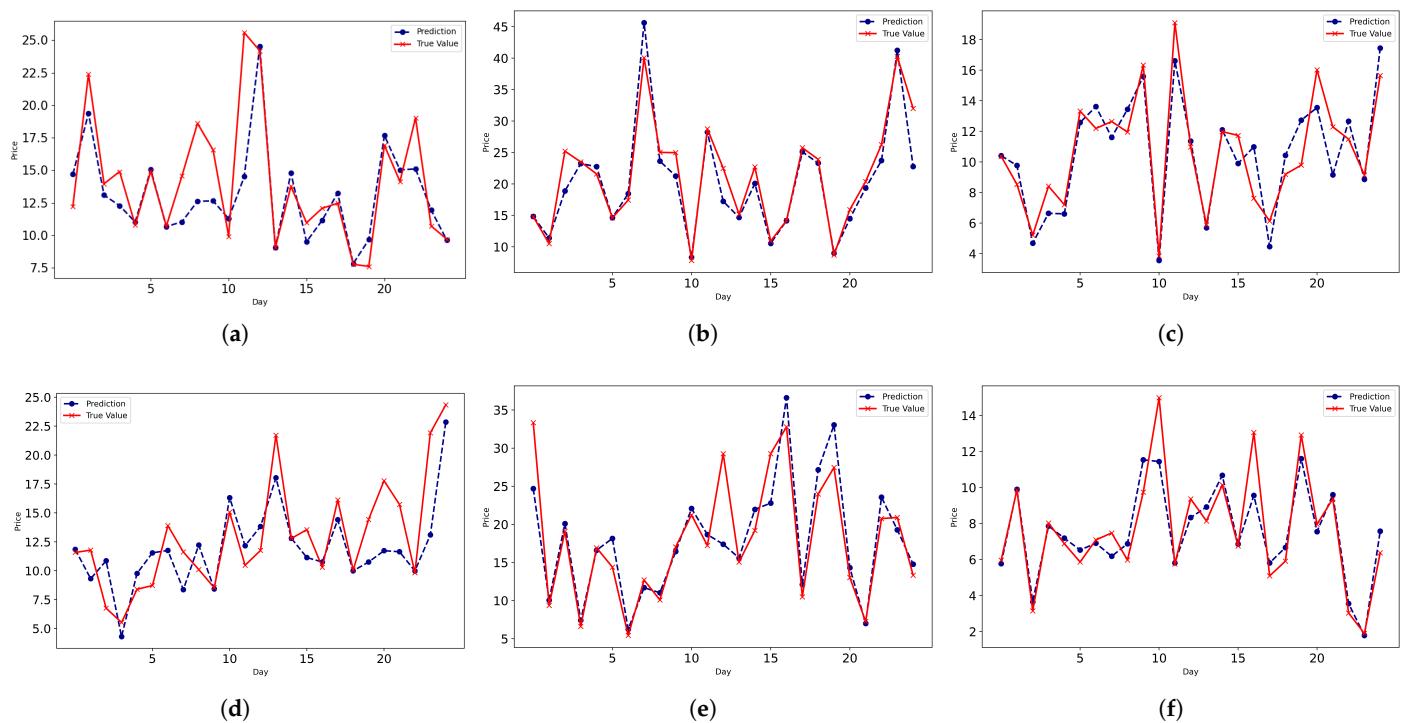


Figure 9. Partial prediction performance in A-share market. (a) Test 1; (b) Test 2; (c) Test 3; (d) Test 4; (e) Test 5; (f) Test 6.

Table 5. Comparison of predictive performance of different models on US stock dataset (divided using method I). The bold indicates the best performance among all models.

Model	MAE	DA	RE	RMSE	R2	Accuracy
GCN	17.4906	50.18%	0.2260	34.4920	0.5802	0.5536
GRU	12.6075	49.19%	0.1385	30.8219	0.6645	0.5639
LSTM	15.5489	48.66%	0.1812	33.4528	0.6051	0.5604
MLP	13.8276	49.75%	0.1760	28.7233	0.7089	0.5595
RNN	16.7846	49.34%	0.2108	34.7371	0.5743	0.5548
TCN	13.9017	49.26%	0.1933	28.7784	0.7078	0.5580
Transformer	19.0230	47.77%	0.2185	36.7984	0.5222	0.5158
STGAT	10.5988	49.33%	0.1079	29.3184	0.7555	0.5734

Table 6. Comparison of predictive performance of different models on US stock dataset (divided using method II). The bold indicates the best performance among all models.

Model	MAE	DA	RE	RMSE	R2	Accuracy
GCN	8.6079	52.13%	0.1604	16.5780	0.8476	0.5989
GRU	4.3827	59.82%	0.0763	7.4141	0.9655	0.7179
LSTM	5.3500	54.65%	0.0870	11.0945	0.9317	0.6802
MLP	3.9639	54.35%	0.0670	6.1679	0.9761	0.7269
RNN	4.6009	54.46%	0.0798	8.0679	0.9591	0.7065
TCN	3.7263	56.08%	0.0719	5.4170	0.9815	0.7352
Transformer	10.3493	54.42%	0.1807	19.6812	0.7852	0.6128
STGAT	2.9266	61.78%	0.0492	4.5396	0.9885	0.7714

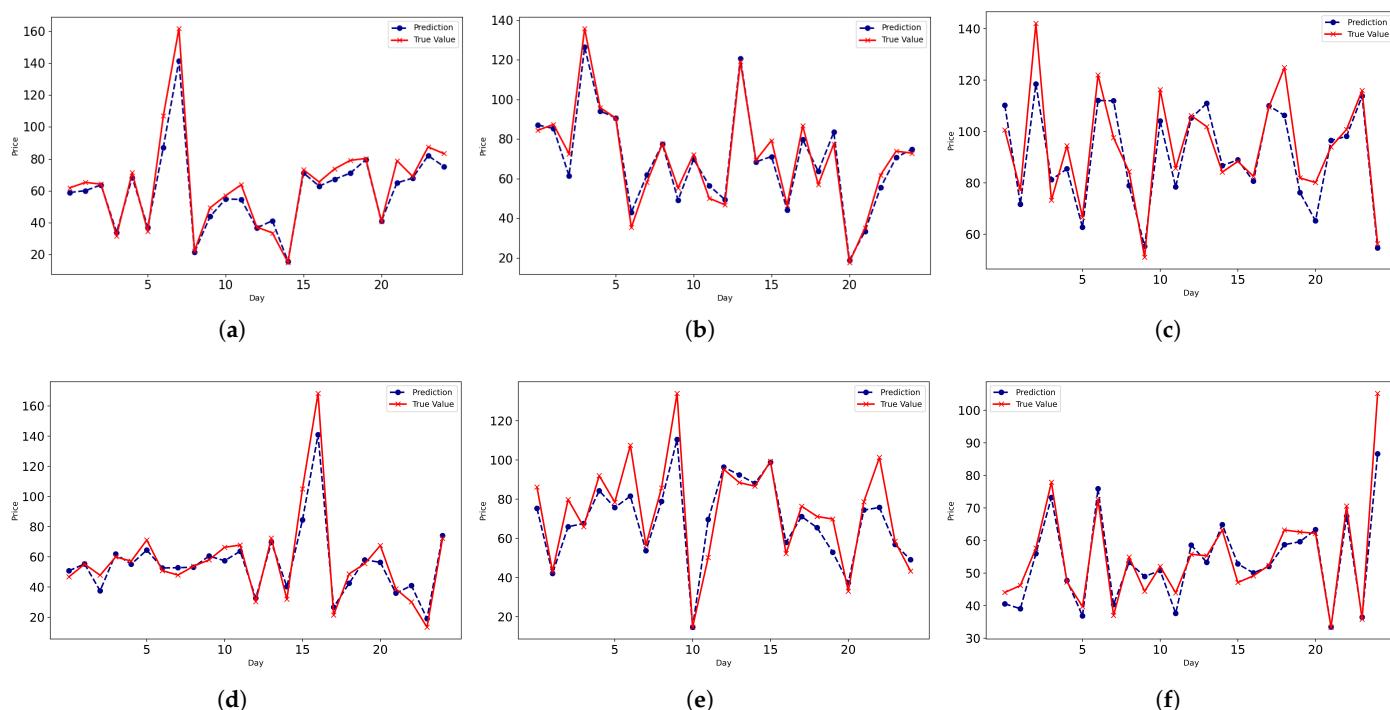


Figure 10. Partial prediction performance in US stock market. (a) Test 1; (b) Test 2; (c) Test 3; (d) Test 4; (e) Test 5; (f) Test 6.

4.3.2. Performance of Portfolio Optimization Results

To rigorously assess the effectiveness of STGAT in portfolio management, this study uses several indicators to evaluate both the risk and return of the portfolio. The indicators outlined in Table 7 aim to determine whether the portfolio's returns commensurate with the risks assumed. In Table 7, (R_1, R_2, \dots, R_n) denotes the return rates from period 1 to period n, while $E(R_p)$ represents the expected value of the portfolio's return. The volatility of the portfolio return is indicated by V , and R_f signifies the risk-free interest rate, which, in this context, is represented by the semi-annual interest rates on Chinese and US treasury bonds. Furthermore, τ is the time index, \hat{P}_t represents the asset value at time t, MDD stands for the portfolio's maximum drawdown, R_b is the benchmark return rate, σ_{p-b} indicates the standard deviation of the difference between portfolio returns and benchmark returns, and β is the systematic risk coefficient.

As shown in Tables 8 and 9, STGAT shows better performance compared to the other models in terms of cumulative returns and Sharpe ratios, indicating its effectiveness in identifying market trends, assessing individual stock potential, and managing risk. These results underscore the practical value and effectiveness of STGAT in portfolio management.

From the results presented in Tables 8 and 9, STGAT demonstrates a remarkable cumulative return, significantly outperforming other deep learning models with 28.21% in the A-share market and an even more impressive 36.87% in the US market. The volatility of all portfolio models in both markets remained within a stable range of approximately 0.1 to 0.3, reflecting consistent stability in both the A-share and US stock markets during the testing period. Such stability is particularly valuable for investors seeking reliable performance in inherently volatile environments. STGAT also excels in terms of the Sharpe ratio, achieving 2.93 in the A-share market and 4.15 in the US market. This further highlights its superior risk–return trade-off compared to the other models, indicating that STGAT not only generates higher returns but does so with relatively lower risk. Moreover, STGAT’s maximum drawdown, at 6.57% in the A-share market, showcases its robust risk management capabilities. By effectively limiting potential losses, STGAT offers a safer investment alternative for risk-averse investors. The Calmar Ratio, which measures cumulative return per unit of maximum drawdown, was notably high for STGAT, standing at 7.60 in the A-share market and 14.76 in the US market. This metric reflects STGAT’s ability to generate substantial returns while maintaining relatively lower risk levels.

Table 7. Quantitative indicators of portfolio performance.

Indicators	Formula	Meaning
Cumulative Return	$\frac{(1 + R_1) \times (1 + R_2) \times \cdots \times (1 + R_n)}{n} - 1$	Total return of the portfolio
Volatility (V)	$\sqrt{\frac{\sum_{i=1}^n (R_i - \bar{R})^2}{n-1}}$	Fluctuation range of stock prices
Sharpe Ratio	$\frac{E(R_p) - R_f}{V}$	Additional return per unit of deviation
Maximum Drawdown (MDD)	$\max_{t:\tau} \left(\frac{P_t}{P_\tau} - 1 \right)$	Maximum loss from peak to trough before new peak
Calmar Ratio	$\frac{E(R_p)}{MDD}$	Ratio of annualized return to maximum drawdown
Information Ratio	$\frac{E(R_p) - R_b}{\sigma_{p-b}}$	Excess return per unit of tracking error
Treynor Ratio	$\frac{E(R_p) - R_f}{\beta}$	Risk premium per unit of systematic risk

Table 8. Performance of investment portfolios in fixed time interval on A-share test sets.

Model	Cumulative Return	Volatility	Sharpe Ratio	Max Drawdown	Calmar Ratio	Information Ratio	Treynor Ratio
CSI500	-5.76%	0.1552	-0.7878	13.68%	-0.8069	-	-0.1222
GCN	11.52%	0.2150	1.3166	10.99%	2.6832	0.4645	0.2363
GRU	7.28%	0.2198	0.8178	13.49%	1.4208	0.3365	0.1497
LSTM	11.51%	0.2230	1.2774	13.43%	2.2095	0.4328	0.2319
MLP	8.00%	0.1979	0.9689	11.02%	1.8476	0.4659	0.1698
RNN	18.78%	0.2096	2.2460	8.62%	5.5977	0.6108	0.4161
TCN	2.65%	0.2138	0.3266	14.35%	0.5695	0.2819	0.0556
Transformer	7.48%	0.2170	0.8471	11.94%	1.6391	0.4629	0.1399
STGAT	28.21%	0.2582	2.9279	10.10%	7.6034	0.5752	0.5771

Additionally, STGAT’s Information Ratio, at 0.22 in the A-share market and 0.58 in the US market, indicates its capacity to provide significant excess returns relative to the benchmark. This suggests that STGAT is highly effective at identifying and exploiting

market inefficiencies. The Treynor Ratio, which reflects the maximum excess return a model can achieve under the same market risk, demonstrates that the majority of models, including STGAT, efficiently manage systematic risk. This makes STGAT a robust choice for investors aiming to balance risk and return. These indicators collectively reveal that the STGAT model excels at identifying stocks whose expected returns exceed their risk-taking capacity while avoiding those highly sensitive to market volatility and offering limited contribution to returns. This strategic approach allows for better control of downside risks while ensuring consistent returns, enabling the model to achieve more excess returns with lower volatility. Overall, STGAT's performance highlights its effectiveness in navigating complex market dynamics and delivering superior risk-adjusted returns.

Table 9. Performance of investment portfolios in fixed time interval on US stock market.

Model	Cumulative Return	Volatility	Sharpe Ratio	Max Drawdown	Calmar Ratio	Information Ratio	Treynor Ratio
S&P500	20.16%	0.1179	3.9280	4.73%	10.0456	-	0.4631
GCN	26.08%	0.2673	2.4776	7.68%	8.7771	0.0849	0.4403
GRU	23.98%	0.2381	2.4938	9.42%	6.4303	0.0590	0.5473
LSTM	16.77%	0.2686	1.5486	10.00%	4.2790	-0.0212	0.2874
MLP	32.42%	0.2244	3.6718	7.28%	11.4829	0.1745	0.6852
RNN	19.86%	0.2539	1.9307	10.31%	4.8712	0.0121	0.3990
TCN	16.12%	0.1784	2.0939	7.00%	5.5078	-0.0476	0.8754
Transformer	25.71%	0.2602	2.4944	7.68%	8.6104	0.0786	0.4885
STGAT	36.87%	0.2306	4.1513	6.57%	14.7553	0.2197	0.8031

To visually illustrate the returns of portfolios constructed by different neural networks, Figures 11 and 12 depict the cumulative returns over a fixed interval in the A-share and US stock markets, respectively. These figures present the cumulative return curves of the portfolios generated during the test period, demonstrating their performance in real market conditions. Notably, the STGAT portfolio exhibits divergent temporal patterns between the two markets: in the A-share market, its outperformance emerges earlier (during mid-2023), whereas in the US market, significant gains materialize later (during early-mid 2024). These temporal variations may arise from structural differences in market characteristics and cross-market spillover effects, such as the policy-driven retail dominance in A-shares versus the institutional liquidity dynamics in US equities or sentiment transmission between stock markets. However, apparently STGAT achieves superior cumulative returns in both markets during periods of relatively stable market conditions. The results reveal that STGAT outperforms both the market index and other deep learning methods, reflecting the feasibility of integrating spatial-temporal features and the advantages of incorporating graph attention mechanisms into temporal analysis.

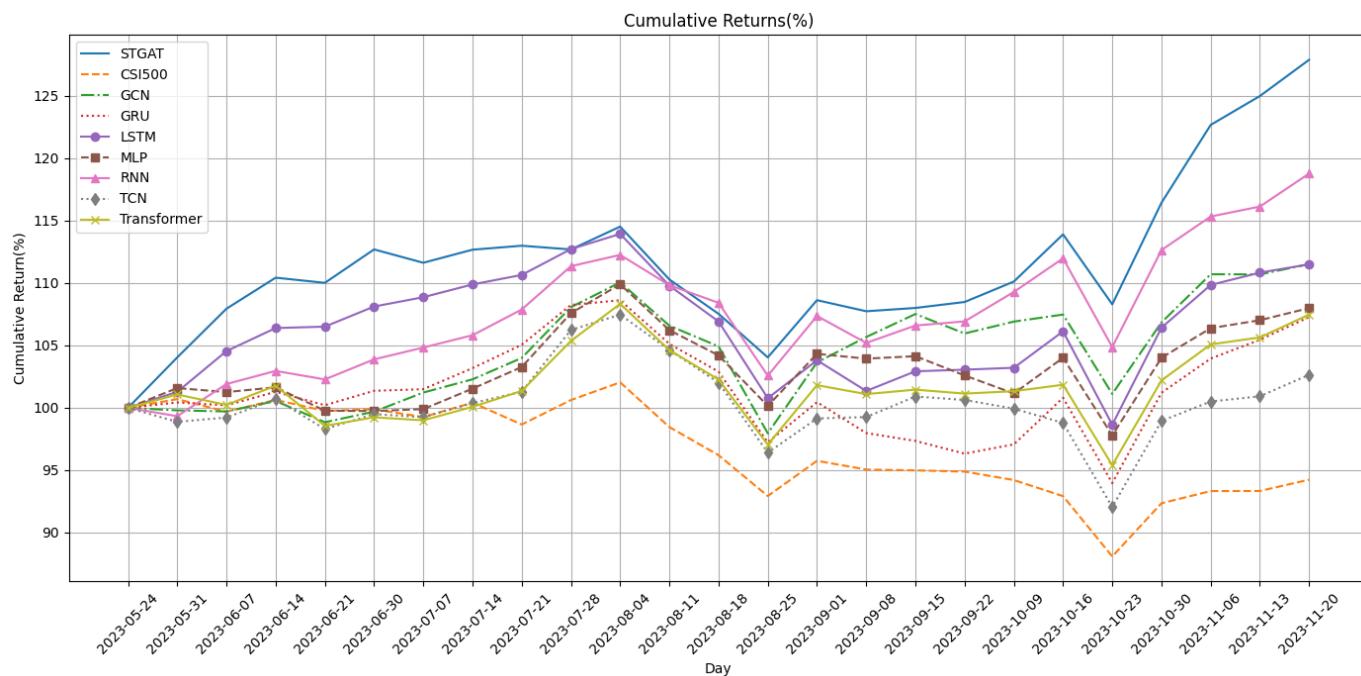


Figure 11. Cumulative return performance on A-share market.

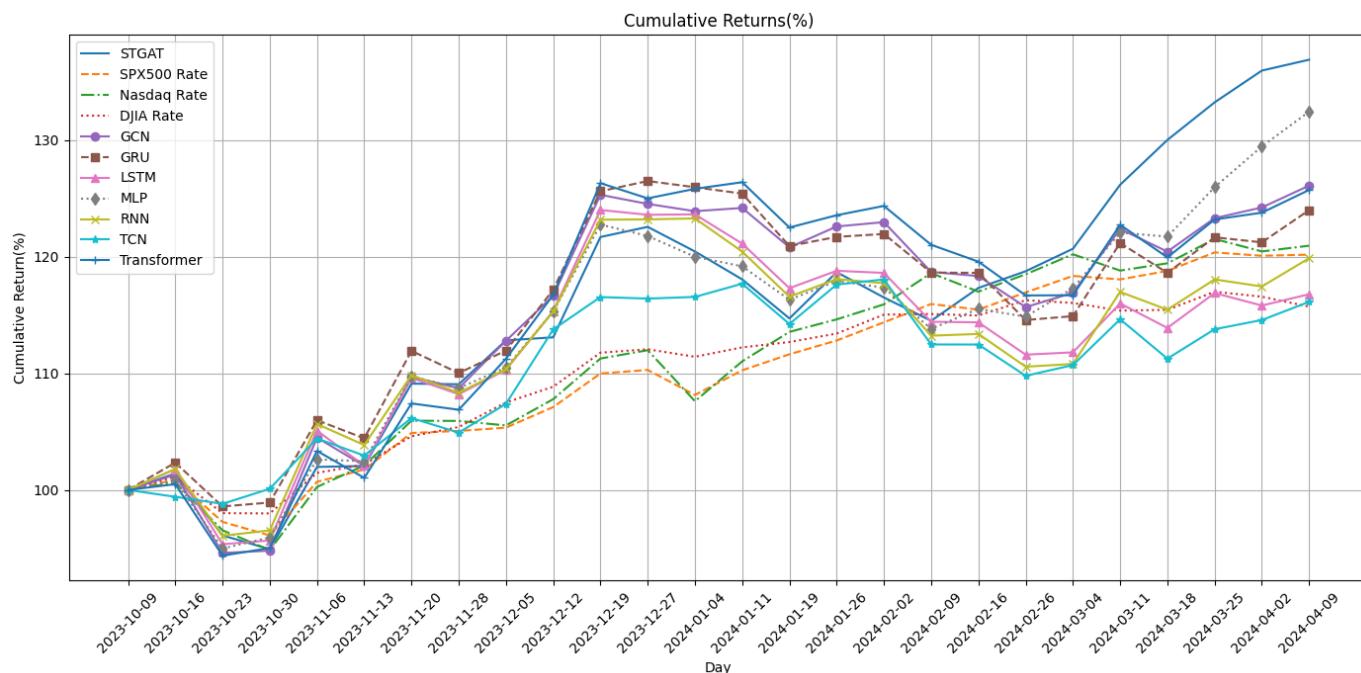


Figure 12. Cumulative return performance on US stock market.

5. Conclusions

This article proposes a Spatial–Temporal Graph Attention Neural Network (STGAT) model that leverages STL for time series decomposition and constructs a graph structure to represent the relationships between stocks. By integrating graph attention mechanisms with temporal convolutional modules, STGAT effectively captures both spatial dependencies among stocks and temporal patterns in their price movements. Compared to other deep learning methods such as a GCN, GRU, LSTM, MLP, RNN, TCN, and Transformer, STGAT achieves better accuracy in stock price prediction. Furthermore, using the predicted stock prices from the test set, we construct investment portfolios to simulate real-market

performance. The results demonstrate that STGAT-based portfolios can deliver higher returns, even in scenarios where the major market index experiences declines.

The dual approach of fusing spatial and temporal features together allows STGAT to outperform traditional machine learning and deep learning models in both prediction accuracy and portfolio performance. As a preliminary attempt to apply the Spatial–Temporal Graph Attention Network to stock market analysis, this study highlights the potential of STGAT in addressing complex financial challenges. However, given the dynamic and ever-changing nature of financial markets, future research could explore incorporating additional data sources, such as macroeconomic indicators, news sentiment, and social media trends, to further enhance the model’s predictive capabilities.

Author Contributions: Conceptualization, R.F. and S.J.; methodology, R.F., S.J. and X.L.; software, R.F. and X.L.; validation, R.F.; formal analysis, R.F. and X.L.; writing—original draft preparation, R.F.; writing—review and editing, M.X. and X.L.; visualization, R.F.; supervision, S.J.; project administration, M.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded by the Ministry of Education, Humanities and social science projects (21YJC790054) and National Natural Science Foundation of PR China (72101121).

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: The data and code used in this work are available at <https://github.com/RuizheF/STGAT>, accessed on 22 February 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Steinbach, M.C. Markowitz revisited: Mean-variance models in financial portfolio analysis. *SIAM Rev.* **2001**, *43*, 31–85. [[CrossRef](#)]
2. Black, F.; Litterman, R. Asset allocation: Combining investor views with market equilibrium. *Goldman Sachs Fixed Income Res.* **1990**, *115*, 7–18. [[CrossRef](#)]
3. Ariyo, A.A.; Adewumi, A.O.; Ayo, C.K. Stock Price Prediction Using the ARIMA Model. In Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, 26–28 March 2014; pp. 106–112. [[CrossRef](#)]
4. Vector Autoregressive Models for Multivariate Time Series. In *Modeling Financial Time Series with S-PLUS®*; Zivot, E., Wang, J., Eds.; Springer: New York, NY, USA, 2006; pp. 385–429. [[CrossRef](#)]
5. Asai, M.; McAleer, M. A Portfolio Index GARCH model. *Int. J. Forecast.* **2008**, *24*, 449–461. [[CrossRef](#)]
6. Su, H.; Wang, X.; Qin, Y.; Chen, Q. Attention based adaptive spatial-temporal hypergraph convolutional networks for stock price trend prediction. *Expert Syst. Appl.* **2024**, *238*, 121899. [[CrossRef](#)]
7. Ince, H.; Trafalis, T.B. Kernel principal component analysis and support vector machines for stock price prediction. *IIE Trans.* **2007**, *39*, 629–637. [[CrossRef](#)]
8. Basak, S.; Kar, S.; Saha, S.; Khadem, L.; Dey, S.R. Predicting the direction of stock market prices using tree-based classifiers. *N. Am. J. Econ. Financ.* **2019**, *47*, 552–567. [[CrossRef](#)]
9. Vrijh, M.; Chandola, D.; Tikkial, V.A.; Kumar, A. Stock closing price prediction using machine learning techniques. *Procedia Comput. Sci.* **2020**, *167*, 599–606. [[CrossRef](#)]
10. 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), Hong Kong, 15–18 December 2017; pp. 599–602. [[CrossRef](#)]
11. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal Convolutional Networks for Action Segmentation and Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1003–1012. [[CrossRef](#)]
12. Gao, J.; Ying, X.; Xu, C.; Wang, J.; Zhang, S.; Li, Z. Graph-Based Stock Recommendation by Time-Aware Relational Attention Network. *ACM Trans. Knowl. Discov. Data* **2021**, *16*, 1–21. [[CrossRef](#)]

13. Xiang, S.; Cheng, D.; Shang, C.; Zhang, Y.; Liang, Y. Temporal and heterogeneous graph neural network for financial time series prediction. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, 17–21 October 2022; pp. 3584–3593.
14. Jafari, A.; Haratizadeh, S. GCNET: Graph-based prediction of stock price movement using graph convolutional network. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105452. [[CrossRef](#)]
15. Zhang, J.; Liu, Z. Interval prediction of crude oil spot price volatility: An improved hybrid model integrating decomposition strategy, IESN and ARIMA. *Expert Syst. Appl.* **2024**, *252*, 124195. [[CrossRef](#)]
16. Gong, J.; Qu, Z.; Zhu, Z.; Xu, H. Parallel TimesNet-BiLSTM model for ultra-short-term photovoltaic power forecasting using STL decomposition and auto-tuning. *Energy* **2025**, *320*, 135286. [[CrossRef](#)]
17. Stasiak, M.D.; Staszak, Z.; Siwek, J.; Wojcieszak, D. Application of State Models in a Binary–Temporal Representation for the Prediction and Modelling of Crude Oil Prices. *Energies* **2025**, *18*, 691. [[CrossRef](#)]
18. Mondal, P.; Shit, L.; Goswami, S. Study of Effectiveness of Time Series Modeling (Arima) in Forecasting Stock Prices. *Int. J. Comput. Sci. Eng. Appl.* **2014**, *4*, 13–29. [[CrossRef](#)]
19. Khanderwal, S.; Mohanty, D. Stock price prediction using ARIMA model. *Int. J. Mark. Hum. Resour. Res.* **2021**, *2*, 98–107. [[CrossRef](#)]
20. Suharsono, A.; Aziza, A.; Pramesti, W. Comparison of vector autoregressive (VAR) and vector error correction models (VECM) for index of ASEAN stock price. In Proceedings of the AIP Conference Proceedings, Provo, UT, USA, 16–21 July 2017; AIP Publishing: Melville, NY, USA, 2017; Volume 1913.
21. Gabriel, A.M.; Ugochukwu, W.M. Volatility estimation and stock price prediction in the Nigerian stock market. *Int. J. Financ. Res.* **2012**, *3*, 2. [[CrossRef](#)]
22. Alam, M.Z.; Siddikee, M.N.; Masukujaman, M. Forecasting volatility of stock indices with ARCH model. *Int. J. Financ. Res.* **2013**, *4*, 126. [[CrossRef](#)]
23. Fenghua, W.; Jihong, X.; Zhifang, H.; Xu, G. Stock price prediction based on SSA and SVM. *Procedia Comput. Sci.* **2014**, *31*, 625–631. [[CrossRef](#)]
24. Heo, J.; Yang, J.Y. Stock price prediction based on financial statements using SVM. *Int. J. Hybrid Inf. Technol.* **2016**, *9*, 57–66. [[CrossRef](#)]
25. Aydin, A.D.; Cavdar, S.C. Comparison of Prediction Performances of Artificial Neural Network (ANN) and Vector Autoregressive (VAR) Models by Using the Macroeconomic Variables of Gold Prices, Borsa Istanbul (BIST) 100 Index and US Dollar-Turkish Lira (USD/TRY) Exchange Rates. *Procedia Econ. Financ.* **2015**, *30*, 3–14. [[CrossRef](#)]
26. Rezaei, H.; Faaljou, H.; Mansourfar, G. Stock price prediction using deep learning and frequency decomposition. *Expert Syst. Appl.* **2021**, *169*, 114332. [[CrossRef](#)]
27. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [[CrossRef](#)]
28. Cheng, P.; Xia, M.; Wang, D.; Lin, H.; Zhao, Z. Transformer Self-Attention Change Detection Network with Frozen Parameters. *Appl. Sci.* **2025**, *15*, 3349. [[CrossRef](#)]
29. Xu, Z.; Wang, Y.; Feng, X.; Wang, Y.; Li, Y.; Lin, H. Quantum-enhanced forecasting: Leveraging quantum gramian angular field and CNNs for stock return predictions. *Financ. Res. Lett.* **2024**, *67*, 105840. [[CrossRef](#)]
30. Pawar, K.; Jalem, R.S.; Tiwari, V. Stock market price prediction using LSTM RNN. In *Proceedings of the Emerging Trends in Expert Applications and Security: Proceedings of ICETEAS 2018*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 493–503.
31. Saud, A.S.; Shakya, S. Analysis of look back period for stock price prediction with RNN variants: A case study on banking sector of NEPSE. *Procedia Comput. Sci.* **2020**, *167*, 788–798. [[CrossRef](#)]
32. Liu, Q.; Hu, Y.; Liu, H. Enhanced stock price prediction with optimized ensemble modeling using multi-source heterogeneous data: Integrating LSTM attention mechanism and multidimensional gray model. *J. Ind. Inf. Integr.* **2024**, *42*, 100711. [[CrossRef](#)]
33. Selvin, S.; Vinayakumar, R.; Gopalakrishnan, E.A.; Menon, V.K.; Soman, K.P. Stock price prediction using LSTM, RNN and CNN-sliding window model. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1643–1647. [[CrossRef](#)]
34. Lu, M.; Xu, X. TRNN: An efficient time-series recurrent neural network for stock price prediction. *Inf. Sci.* **2024**, *657*, 119951. [[CrossRef](#)]
35. Chen, W.; An, N.; Jiang, M.; Jia, L. An improved deep temporal convolutional network for new energy stock index prediction. *Inf. Sci.* **2024**, *682*, 121244. [[CrossRef](#)]
36. Zarnowitz, V.; Ozyildirim, A. Time series decomposition and measurement of business cycles, trends and growth cycles. *J. Monet. Econ.* **2006**, *53*, 1717–1739. [[CrossRef](#)]
37. Wei, L.Y. A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Appl. Soft Comput.* **2016**, *42*, 368–376. [[CrossRef](#)]

38. Xu, C.; Zhao, X.; Wang, Y. Causal decomposition on multiple time scales: Evidence from stock price-volume time series. *Chaos Solitons Fractals* **2022**, *159*, 112137. [[CrossRef](#)]
39. Tao, Z.; Wu, W.; Wang, J. Series decomposition Transformer with period-correlation for stock market index prediction. *Expert Syst. Appl.* **2024**, *237*, 121424. [[CrossRef](#)]
40. Singh, S.; Singh, M.; Attri, S. Mapping stock market dynamics: A tripartite neural network approach using modified grid search for stock market prediction. *Expert Syst. Appl.* **2025**, *278*, 127243. [[CrossRef](#)]
41. Preis, T.; Kenett, D.Y.; Stanley, H.E.; Helbing, D.; Ben-Jacob, E. Quantifying the behavior of stock correlations under market stress. *Sci. Rep.* **2012**, *2*, 752. [[CrossRef](#)] [[PubMed](#)]
42. Pollet, J.M.; Wilson, M. Average correlation and stock market returns. *J. Financ. Econ.* **2010**, *96*, 364–380. [[CrossRef](#)]
43. Song, D.M.; Tumminello, M.; Zhou, W.X.; Mantegna, R.N. Evolution of worldwide stock markets, correlation structure, and correlation-based graphs. *Phys. Rev. E* **2011**, *84*, 026108. [[CrossRef](#)]
44. Patel, M.; Jariwala, K.; Chattopadhyay, C. A Systematic Review on Graph Neural Network-based Methods for Stock Market Forecasting. *ACM Comput. Surv.* **2024**, *57*, 1–38. [[CrossRef](#)]
45. Yin, T.; Liu, C.; Ding, F.; Feng, Z.; Yuan, B.; Zhang, N. Graph-based stock correlation and prediction for high-frequency trading systems. *Pattern Recognit.* **2022**, *122*, 108209. [[CrossRef](#)]
46. Huang, W.C.; Chen, C.T.; Lee, C.; Kuo, F.H.; Huang, S.H. Attentive gated graph sequence neural network-based time-series information fusion for financial trading. *Inf. Fusion* **2023**, *91*, 261–276. [[CrossRef](#)]
47. Long, J.; Chen, Z.; He, W.; Wu, T.; Ren, J. An integrated framework of deep learning and knowledge graph for prediction of stock price trend: An application in Chinese stock exchange market. *Appl. Soft Comput.* **2020**, *91*, 106205. [[CrossRef](#)]
48. Hu, Z.; Liu, W.; Bian, J.; Liu, X.; Liu, T.Y. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Angeles, CA, USA, 5–9 February 2018; pp. 261–269.
49. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. 2018. Available online: <http://arxiv.org/abs/1710.10903> (accessed on 16 February 2018).
50. Lei, Z.; Zhang, C.; Xu, Y.; Li, X. DR-GAT: Dynamic routing graph attention network for stock recommendation. *Inf. Sci.* **2024**, *654*, 119833. [[CrossRef](#)]
51. Li, G.; Zhang, A.; Zhang, Q.; Wu, D.; Zhan, C. Pearson Correlation Coefficient-Based Performance Enhancement of Broad Learning System for Stock Price Prediction. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 2413–2417. [[CrossRef](#)]
52. Rungskunroch, P.; Shen, Z.J.; Kaewunruen, S. Benchmarking Socio-Economic Impacts of High-Speed Rail Networks Using K-Nearest Neighbour and Pearson’s Correlation Coefficient Techniques through Computational Model-Based Analysis. *Appl. Sci.* **2022**, *12*, 1520. [[CrossRef](#)]
53. Zhan, Z.; Ren, H.; Xia, M.; Lin, H.; Wang, X.; Li, X. AMFNet: Attention-Guided Multi-Scale Fusion Network for Bi-Temporal Change Detection in Remote Sensing Images. *Remote. Sens.* **2024**, *16*, 1765. [[CrossRef](#)]
54. Wang, Z.; Gu, G.; Xia, M.; Weng, L.; Hu, K. Bitemporal Attention Sharing Network for Remote Sensing Image Change Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2024**, *17*, 10368–10379. [[CrossRef](#)]
55. Zhu, T.; Zhao, Z.; Xia, M.; Huang, J.; Weng, L.; Hu, K.; Lin, H.; Zhao, W. FTA-Net: Frequency-Temporal-Aware Network for Remote Sensing Change Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2025**, *18*, 3448–3460. [[CrossRef](#)]
56. Jiang, S.; Lin, H.; Ren, H.; Hu, Z.; Weng, L.; Xia, M. MDANet: A High-Resolution City Change Detection Network Based on Difference and Attention Mechanisms under Multi-Scale Feature Fusion. *Remote. Sens.* **2024**, *16*, 1387. [[CrossRef](#)]

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.