

DART: Domain Aware Relational Transformer for Stock Prediction

Ruoyu Huang

Fujian Agriculture and Forestry University

Xiaoyu Lin

`xiaoyulin@fafu.edu.cn`

Fujian Agriculture and Forestry University

Yibao Mao

Fujian Agriculture and Forestry University

Jiqiang Yan

Fujian Agriculture and Forestry University

Research Article

Keywords: Stock Forecasting, Deep Learning, Sparse Attention, Multi-Scale Modeling, Quantitative Investment

Posted Date: August 6th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9891943/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Stock return prediction with Transformer-style architectures faces a practical trade-off between expressive relational modeling and scalability on large stock universes. To address this issue, this paper proposes Domain-Aware Relational Transformer (DART), an efficient framework that improves both predictive performance and computational cost for large-scale stock prediction. DART contains two core components. First, a Multi-Scale Temporal Fusion (MSTF) module extracts short-, medium-, and longer-horizon signals through parallel convolutions and aggregates them with a Progressive Temporal Fusion (PTF) network to capture cumulative temporal dynamics. Second, a Domain-Knowledge-Guided Sparse Attention (DSA) mechanism uses industry classification priors to constrain cross-stock attention to economically relevant neighborhoods, reducing unnecessary interactions while preserving informative co-movement structure. Experiments on three real-world U.S. market datasets, NASDAQ, NYSE, and S&P 500, compare DART with classical baselines and recent strong models, including STHAN-SR, CausalStock, and MambaStock, under a unified setting. The results show that DART achieves state-of-the-art performance across the evaluated benchmarks, delivering the strongest overall results in Information Coefficient (IC), Rank Information Coefficient (RIC), Precision@10, and Sharpe Ratio, with especially clear advantages in risk-adjusted returns. In addition, floating-point-operation analysis shows that the sparse attention design reduces the core computational load by about 83%-84% relative to full attention. These findings indicate that DART provides a practical, scalable, and state-of-the-art solution for large-scale stock prediction and portfolio-oriented ranking tasks.

1. Introduction

1.1 Research Background and Motivation

Price fluctuations in the stock market are a complex composite of macroeconomic conditions, industry trends, and investor sentiment, making the accurate prediction of its future trajectory one of the most compelling yet challenging core tasks in finance. Effective stock return forecasting not only provides decision support for investors to construct alpha-generating investment strategies (Grinold and Kahn 2000), but it also holds crucial theoretical and practical significance for understanding market dynamics, conducting risk management, and maintaining financial system stability (Engle 2001).

For a long time, the academic discourse was dominated by the Efficient Market Hypothesis (EMH), proposed by Eugene Fama, which posits that asset prices fully reflect all available information, thereby rendering market prediction futile (Fama 1970). However, with the rise of behavioral finance, a substantial body of research has unveiled market anomalies stemming from irrational investor behaviors, such as overconfidence and herding effects, indicating that the market is not perfectly efficient and thus providing a theoretical foundation for predictability (Shiller 2003). In recent years, the rapid development of big data and artificial intelligence (AI) technologies has made it feasible to process and analyze massive, high-dimensional, and multi-modal financial data. Leveraging its powerful capabilities for non-linear feature extraction and pattern recognition, Deep Learning has emerged as the

predominant paradigm for unearthing latent predictive signals in the market (Heaton et al. 2017; Gu et al. 2020). This, coupled with continued breakthroughs in the broader field of asset pricing (Kong et al. 2024), has collectively spurred a profound transformation from traditional statistical arbitrage to modern quantitative investment (Prado 2018). Recent surveys and empirical studies further show that large-language-model- and machine-learning-based forecasting are increasingly influential in stock prediction and portfolio construction (Darwish et al. 2025; Ślusarczyk and Ślepaczuk 2025).

Despite the immense potential of deep learning, the inherent characteristics of financial time series—namely, high noise, non-stationarity, and multi-scale properties (Bollerslev 1986; Mandelbrot 1963)—place stringent demands on the robustness and generalization capabilities of the models. The motivation for this study is therefore to design a novel deep learning architecture better adapted to the unique properties of financial data. We position our work as a deep learning paradigm that, under realistic financial constraints (e.g., large scale, high noise), incorporates domain-specific prior knowledge into its structural design to address the dual challenges that existing methods face in capturing complex temporal patterns and modeling large-scale inter-stock relationships.

1.2 Challenges of Existing Methods

Although extensive research has been dedicated to stock prediction using deep learning, two core challenges are commonly encountered.

The first, in the realm of time series modeling, is how to effectively capture the multi-level dynamic patterns inherent in financial markets. Early models primarily relied on Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs), to process sequential data (Hochreiter and Schmidhuber 1997). Numerous studies have demonstrated the effectiveness of LSTMs in capturing temporal dependencies (Fischer and Krauss 2018; Nelson et al. 2017). However, the inherent sequential processing mechanism of RNN-based models makes it difficult to capture very long-term dependencies and exposes them to the problems of vanishing or exploding gradients (Pascanu et al. 2013). On the other hand, Convolutional Neural Networks (CNNs) have also been employed to extract local patterns from time series (LeCun et al. 1998), for instance, by treating stock price charts as images for analysis (Chen et al. 2016). Temporal Convolutional Networks (TCNs) expand the receptive field through causal and dilated convolutions, outperforming RNNs on certain tasks (Bai et al. 2018). Nevertheless, a key limitation of both CNNs and TCNs is their fixed kernel size, which struggles to adaptively capture the co-existing, multi-frequency fluctuation patterns found in financial markets, such as high-frequency noise at the minute level, daily trading patterns, and weekly trends.

Second, concerning the modeling of inter-stock relationships, is the challenge of efficiently processing the mutual influences within a large-scale stock pool. The stock market is a highly interconnected complex system where the price fluctuation of an individual stock is influenced not only by its own historical data but also significantly by other related stocks, such as industry competitors or companies upstream and downstream in the supply chain (Plerou et al. 1999; Mantegna 1999). To model these interrelationships, researchers have introduced Graph Neural Networks (GNNs) (Scarselli et al. 2009; Kipf

and Welling 2017). By constructing a static graph based on industry classifications, supply chain relationships, or correlation matrices, GNNs can aggregate information from neighboring nodes to learn inter-stock correlations (Feng et al. 2019; Chen et al. 2021). However, the primary limitation of the GNN-based approach is that its performance is highly dependent on a predefined graph structure, and such a static graph may fail to capture the dynamically changing economic linkages in the market.

To overcome the reliance of GNNs on static graphs, researchers have turned to the more flexible attention mechanism, particularly the self-attention mechanism that forms the core of the Transformer model (Vaswani et al. 2017). The Transformer and its variants can dynamically learn the relational weights between all pairs of stocks without needing a predefined graph structure, thus demonstrating immense potential (Ding et al. 2020). However, the core bottleneck of the standard self-attention mechanism is that its computational and memory complexities are both quadratic with respect to the number of stocks (N), i.e., $O(N^2)$ (Vaswani et al. 2017). When applied to real-world markets containing thousands of stocks, this quadratic complexity renders the training and inference costs of the model prohibitively expensive, severely limiting its scalability and practicality in live quantitative trading scenarios (Tay et al. 2020). Therefore, reducing the complexity of the attention mechanism has become a critical challenge for its application in large-scale stock prediction. Although the academic community has proposed several general-purpose sparse attention methods, such as Longformer or BigBird (Beltagy et al. 2020; Zaheer 2020), these methods are mostly designed for generic sequences and lack financial domain-specific prior knowledge, meaning they may not be the optimal solution. Recent stock-market studies also report that LLM- and transformer-enhanced pipelines can improve both prediction and trading performance (Chen 2025; Wang 2026).

1.3 Our Method and Contributions

To address the aforementioned challenges, this paper proposes a novel and efficient deep learning framework for stock return prediction. This framework tackles the difficulties of capturing complex temporal patterns and modeling large-scale inter-stock relationships through two core innovative components.

First, to more comprehensively capture financial time-series features, we have designed a Multi-Scale Temporal Fusion (MSTF) module. This module utilizes a parallel, multi-branch convolutional structure to simultaneously extract market signals at different temporal scales and innovatively introduces a Progressive Temporal Fusion (PTF) network to incrementally aggregate the fused multi-scale features, thereby learning the cumulative effects within the time series.

Second, to resolve the $O(N^2)$ complexity bottleneck of the standard attention mechanism, we propose a more general methodological framework: the Domain-Knowledge-Guided Sparse Attention (DSA) mechanism. This mechanism leverages the industry classifications of stocks as financial domain-specific prior knowledge to construct a sparse attention mask, thereby confining the attention computation to within more relevant industry sectors. This approach not only dramatically reduces

computational complexity, enabling it to efficiently handle thousands of stocks, but its design is also grounded in solid economic intuition, which enhances the model's interpretability.

The main contributions of this paper can be summarized as follows:

Methodological Innovation: We propose a novel deep learning architecture centered around the MSTF and DSA mechanisms. The former is designed to capture multi-level temporal dynamics, while the latter provides an efficient and financially-informed solution for modeling inter-stock relationships.

Superior Performance: Extensive experiments on three large-scale, real-world stock market datasets (NASDAQ, NYSE, and S&P 500) demonstrate that, compared to various state-of-the-art baseline models, our proposed model achieves significant and consistent performance gains on key quantitative investment metrics, including the Information Coefficient (IC) and the Sharpe Ratio.

Efficiency Optimization: We successfully reduce the complexity of modeling inter-stock relationships from $O(N^2)$ to a near-linear level. A theoretical analysis of computational cost (FLOPs) proves that the core computational overhead is reduced by over 84% compared to standard full attention, laying a solid efficiency foundation for the model's application in large-scale scenarios.

1.4 Organization of This Paper

The remainder of this paper is organized as follows: Section 2 reviews the related work in the field of stock prediction. Section 3 details our proposed model architecture and its various components. Section 4 presents our experimental setup, evaluation metrics, and a detailed comparison of the results against baseline models. Finally, Section 5 concludes the paper with a summary of our work and discusses future research directions.

2. Related Work

This chapter aims to review and discuss the body of research related to stock return prediction. Our review is structured around three key dimensions: time series modeling methods for individual stocks, methods for modeling the relational connections among multiple stocks, and the sparse attention mechanisms developed to address large-scale computational challenges.

2.1 Time Series-Based Stock Prediction Models

Modeling the time series of stock prices is the most fundamental and core research direction in this field. Early research was primarily focused on statistical and econometric models. The Autoregressive Integrated Moving Average (ARIMA) model and its derivatives have been widely used to capture linear temporal dependencies (Box et al. 2015). The ARCH and GARCH models, on the other hand, focus on modeling the "volatility clustering" phenomenon commonly found in financial time series (Engle 2001; Bollerslev 1986). However, these linear models struggle to depict the complex non-linear dynamics of financial markets.

With the advancement of machine learning techniques, models such as Support Vector Machines (SVM) and Gradient Boosting Decision Trees (GBDT) have proven to be more advantageous in handling non-linear problems (Krauss et al. 2017; Huang et al. 2005). These methods typically frame the prediction task as a regression or classification problem based on historical technical indicators. Although they have achieved some success, their feature engineering process is highly dependent on expert knowledge, and their ability to capture the intrinsic dynamic dependencies of time series data is limited.

In recent years, deep learning has become the mainstream paradigm in this field due to its automatic feature extraction and powerful non-linear modeling capabilities. Recurrent Neural Networks (RNNs) and their advanced variants, like LSTM and GRU, are widely applied to stock prediction due to their architecture being naturally suited for time series data (Hochreiter and Schmidhuber 1997; Fischer and Krauss 2018; Nelson et al. 2017). For example, Akita et al. utilized LSTM combined with corporate textual reports and price data for prediction (Akita et al. 2016). Although RNN-based models can theoretically capture long-term dependencies, in practice, they are still limited by gradient propagation issues and the inefficiency of sequential computation (Pascanu et al. 2013).

Another line of technical development parallel to RNNs is the Convolutional Neural Network (CNN). Researchers have found that one-dimensional CNNs can effectively extract local price patterns or shapes from time series, such as "head and shoulders tops" and "double bottoms" (Chen et al. 2016; Ding et al. 2020). The Temporal Convolutional Network (TCN), by combining causal convolutions, dilated convolutions, and residual connections, constructs a non-recurrent structure that strictly adheres to temporal order and possesses a vast receptive field, showing performance superior to RNNs on many sequence modeling tasks (Bai et al. 2018). However, a key limitation of both CNNs and TCNs is their fixed kernel size, which struggles to adaptively capture the co-existing, multi-frequency fluctuation patterns in financial markets. Some studies have attempted to mitigate this issue through parallel multi-branch CNN structures (Ding et al. 2020), while more recent research has begun to explore more flexible time-series hybrid architectures. For example, LightCTS, proposed by Nie et al., decouples and predicts the periodic and trend components in time series via Fourier transforms and subspace learning (Nie et al. 2023). Similarly, the MTS-Mixers model also adopts a strategy of mixing across time and channel dimensions to capture complex dependencies (Zhang et al. 2023). These works have inspired our research, demonstrating the limitations of a single model architecture and motivating our design of the MSTF to more comprehensively capture market dynamics.

2.2 Modeling Inter-Stock Relationships

The stock market is an interconnected, complex network, and modeling the spillover effects among stocks is crucial for improving predictive accuracy.

Graph Neural Networks (GNNs) provide a powerful tool for explicitly modeling stock relationships (Scarselli et al. 2009; Kipf and Welling 2017). Researchers typically construct a stock graph based on certain prior knowledge, where nodes represent stocks and edges represent their relationships. Common graph construction methods include industry relationship graphs based on the Global Industry

Classification Standard (GICS) (Chen et al. 2016), supply chain or competitive relationship graphs mined from corporate financial reports or news text (Yao et al. 2021), or correlation matrices calculated from historical price series (Ying et al. 2021). On this graph, GNNs aggregate information from neighboring nodes via a message-passing mechanism to learn the embedding representation of each stock within the network. For instance, GraphSAGE has been used to aggregate information from stocks with similar attributes (Hamilton et al. 2017). More recent studies, such as KG-GCN, have even attempted to integrate richer entity relationships from knowledge graphs (e.g., 'subsidiary,' 'competitor') into the GNN to build a more information-dense stock correlation graph, achieving promising results (Zhao et al. 2023). However, the primary drawback of the GNN approach is that its performance is highly dependent on the quality and validity of the predefined graph. A static or inaccurate graph structure may fail to reflect the true and dynamic economic linkages in the market and can even introduce noise that misleads the model's learning process.

To overcome the reliance on static graphs, attention-based models, particularly the Transformer, have been introduced to this field (Vaswani et al. 2017). The core of the Transformer is the self-attention mechanism, which allows the model to dynamically compute the relationship weights between any two stocks without a predefined structure. This end-to-end learning approach endows the model with tremendous flexibility. For example, the Hierarchical-GAT model proposed by Ding et al. utilizes both intra-stock and inter-stock attention mechanisms (Ding et al. 2015). However, this flexibility comes at a great computational cost. The computational complexity and memory consumption of standard self-attention are both quadratic with respect to the number of stocks (n), i.e., $O(n^2)$ (Vaswani et al. 2017). This makes it impractical to apply Transformers to real-world investment portfolios containing thousands of stocks, as the prohibitive cost severely hinders their scalability and deployment in industrial applications.

2.3 Sparse Attention Mechanisms

To address the complexity bottleneck of the standard Transformer, the academic community has proposed various "Efficient Transformer" or sparse attention solutions (Tay et al. 2020). The core idea behind these methods is to approximate the full attention matrix with a sparse one, thereby reducing the complexity to a near-linear or sub-quadratic level.

Mainstream sparse attention methods can be broadly categorized into two types: position-based and content-based sparse patterns. Position-based methods predefine fixed sparse connection patterns, such as focusing only on an adjacent local window (Sliding Window Attention), attending with a certain interval (Dilated Attention) (Beltagy et al. 2020), or attending to a few pre-selected global nodes (Global Attention) (Beltagy et al. 2020; Zaheer 2020). These methods have achieved great success in fields like Natural Language Processing. However, the rationale of directly applying these general-purpose, content-agnostic sparse patterns to financial markets is questionable. The correlation between stocks is not determined by their "position" in a list but by their intrinsic economic connections.

The other category is content-based sparse methods, which attempt to dynamically select the most important keys for attention computation. However, this usually requires additional clustering or hashing

algorithms, which increases the model's complexity.

Unlike the aforementioned general-purpose methods, the DSA proposed in this paper is a sparsity strategy based on domain knowledge. We leverage a simple yet effective financial prior: stocks within the same industry typically exhibit stronger co-movement than stocks across different industries (Plerou et al. 1999; Moskowitz and Grinblatt 1999). By focusing the attention computation within industries, we not only substantially reduce the complexity from $O(N^2)$ to $O(N \cdot I)$ (where I is the average industry size and is far smaller than N), but also make the model more aligned with economic intuition, enhancing its interpretability, and potentially improving predictive performance by reducing noisy correlations. This approach of integrating domain knowledge to guide model architecture design offers an effective new path for tackling large-scale modeling problems in the financial domain.

3. Methodology

This chapter elaborates on the overall architecture and core components of our proposed DART model. Through a meticulously designed hierarchical structure, the model first performs deep temporal feature extraction for each stock independently. It then utilizes a unique dual-path prediction mechanism to generate forecasts from two distinct dimensions—pure time-series and cross-sectional interactions—and finally integrates both outputs to achieve a more robust and comprehensive prediction.

3.1 Model Architecture

Our proposed model follows a hierarchical processing paradigm of "temporal-first (intra-stock), then cross-sectional (inter-stock)". As illustrated in Fig. 1, the overall data flow of the model is as follows:

Input Layer: The model's input is a three-dimensional tensor comprising N stocks, T time steps, and F features.

Temporal Feature Extraction: The time series data for each stock is independently and parallelly fed into the MSTF module to capture and fuse dynamic patterns at different temporal scales.

Channel Compression & Dual-Path Prediction: After temporal feature extraction and channel compression, the resulting temporal feature vectors enter two parallel prediction paths. This design constitutes an intrinsic form of Ensemble Learning. One path performs direct temporal aggregation to generate the time-series prediction component, aimed at capturing the independent trend of each stock (i.e., the 'idiosyncratic return' component). The other path conducts cross-sectional interaction among stocks via the DSA module to generate the cross-sectional prediction component, aimed at capturing systemic market influences (such as industry co-movement).

Output & Ensemble: The prediction components from the two paths are finally summed by an ensemble module to produce the final integrated prediction result.

The entire model is trainable end-to-end and is jointly optimized through a composite loss function.

3.2 Input Representation

For a stock pool containing N stocks, we use the historical data of each stock over the past T time steps as input. The data for each time step includes F features, such as the open, high, low, and close prices, and trading volume. Therefore, the raw input to the model is a three-dimensional tensor, X , with the following dimensions:

$$X \in \mathbb{R}^{N \times T \times F}$$

3.3 Multi-Scale Temporal Fusion (MSTF)

Financial time series contain complex patterns at different frequencies and periods. For example, short-term high-frequency trading behaviors, medium-term trend momentum, and long-term mean-reversion effects may co-exist simultaneously (Cont 2001). Traditional CNN or RNN models, which use a single receptive field or a fixed temporal dependency structure, struggle to capture these multi-level dynamic signals at the same time.

To address this issue, we designed the MSTF module. The core idea of this module is inspired by the Inception architecture from computer vision (Szegedy et al. 2015); it uses a parallel multi-branch structure to extract features at different temporal scales and then effectively fuses these features.

3.3.1 Multi-Scale Convolutional Branches

As shown in Fig. 1, the MSTF module contains multiple parallel processing branches. For the input time series data of a single stock, $X_i \in \mathbb{R}^{T \times F}$, the module's processing flow is as follows:

Identity Path: One branch is an identity mapping path, which preserves the original time series information to prevent the loss of important raw signals during feature extraction.

Convolutional Paths: The remaining branches consist of one-dimensional convolutional layers (Conv1D) with different configurations. Each convolutional branch uses a different kernel size. For instance, one branch might use a small kernel size (e.g., 2) to capture short-term local patterns, while another uses a larger kernel size (e.g., 4) to capture longer-term macroscopic trends. This design allows the model to simultaneously "see" market dynamics at different granularities of time.

Feature Concatenation (Multi-Scale Fusion Layer): The outputs from all branches are concatenated along the temporal dimension. The resulting concatenated feature tensor, $H_{concat} \in \mathbb{R}^{T_{cat} \times F}$, contains rich information extracted from multiple time scales, where T_{cat} is the total length of the concatenated temporal dimension.

3.3.2 Progressive Temporal Fusion (PTF) Network

After obtaining the multi-scale fused features H_{concat} , we introduce the novel PTF network to further model the temporal relationships within them. The core of PTF lies in utilizing a learnable lower triangular matrix to transform the time series, thereby modeling the cumulative and causal effects of the sequence. Figure 2 illustrates the core idea of this causal operation, where the key is to strictly adhere to the temporal order, ensuring the model's causality. That is, predictions for the future can only depend on past and present information, without "seeing" into the future. Specifically, when calculating the output for the $\backslash \text{varvect}$ -th time step, the model can only use input information from the current moment $\backslash \text{varvect}$ and past moments $(\backslash \text{varvect} - 1, \dots, 1)$. Any information from future moments (e.g., $\backslash \text{varvect} + 1$) is strictly prohibited to prevent information leakage. This structure ensures that the model's decision-making process is rational and logically sound when processing time series data, which is crucial for scenarios requiring rigorous causal inference, such as financial forecasting.

The PTF network contains a learnable weight matrix $W_{tril} \in \mathbb{R}^{T_{cat} \times T_{cat}}$. During the network's forward pass, we first convert the matrix into a strictly lower triangular matrix (i.e., the main diagonal and all elements above it are zero). Then, this lower triangular matrix is multiplied with the input temporal features. This operation can be formalized as:

$$H_{out} = H_{concat} \cdot tril(W_{tril})$$

where $tril(\bullet)$ denotes the operation of taking the lower triangular part of a matrix. Due to the structural properties of a strictly lower triangular matrix, the feature for the $\backslash \text{varvect}$ -th time step in the output sequence, H_{out} , is a weighted sum of the features from time step 1 up to time step t in the input sequence, H_{concat} . This forward-looking cumulative effect, up to step t , ensures that the output at each time point incorporates all its historical information, strictly adhering to the principle of causality. Unlike traditional RNNs, which capture temporal dependencies recursively, our method accomplishes this using parallel matrix operations, offering a novel perspective for efficient, non-recurrent sequence feature extraction.

3.4 Domain-Knowledge-Guided Sparse Attention (DSA)

After modeling the time series for each individual stock, we need to model the relationships among the stocks. As previously discussed, the complexity of the standard self-attention mechanism is the main obstacle to its application in large-scale stock pools. To address this bottleneck, we propose the DSA mechanism. The design of this method is based on a widely accepted prior in the financial domain: the co-movement effect of stock prices is typically much stronger within the same industry than across different industries (Moskowitz and Grinblatt 1999). Therefore, by constraining the scope of attention computation to within industries, we can safely ignore the majority of weakly correlated or uncorrelated stock pairs, thereby substantially improving computational efficiency while preserving model performance.

After obtaining the temporal feature vectors, the DSA module is responsible for efficiently modeling the mutual influences among these stocks. At its core is the Intra-Industry Attention layer. This layer utilizes

a pre-generated industry adjacency matrix as a mask, calculating attention weights and aggregating information only among stocks within the same industry. This approach drastically reduces computational complexity while efficiently capturing key market correlations. After a series of transformations including the attention calculation, a residual connection, and a Feed-Forward Network (FFN), the module ultimately outputs the cross-sectional prediction component.

3.4.1 Sparse Attention Mask Generation

Before model training begins, we first generate a static binary adjacency matrix, $A \in \{0, 1\}^{N \times N}$, for the stock pool based on an external file that provides the industry classification for each stock. The value of an element A_{ij} in this matrix is determined as follows:

$$A_{ij} = \begin{cases} 1, & \text{stock } i \text{ and stock } j \text{ are in the same industry} \\ 0, & \text{otherwise} \end{cases}$$

This adjacency matrix A serves as the basis for the attention mask and remains constant throughout the training and inference processes.

3.4.2 Sparse Attention Calculation

During the self-attention calculation, we first follow the standard Transformer procedure, mapping the input $H \in R^{N \times D_{model}}$ to Query (Q), Key (K), and Value (V) matrices through linear transformations. Then, when calculating the attention score matrix, we introduce an attention bias matrix M , which is generated from the adjacency matrix A :

$$AttentionScore = \frac{QK^T}{\sqrt{d_k}} + M$$

where d_k is the dimension of the key vectors. The value of the bias matrix $M \in R^{N \times N}$ is determined as follows:

$$M_{ij} = \begin{cases} 0, & \text{if } A_{ij} = 1 \text{ (stocks are in the same industry)} \\ -\infty, & \text{if } A_{ij} = 0 \text{ (stocks are in different industry)} \end{cases}$$

By adding such a bias matrix, the output of the *softmax* function will cause the attention weights between pairs of stocks in different industries to approach zero. The final attention output is:

$$Attention(Q, K, V) = Softmax(AttentionScore) V$$

In this way, when aggregating information, each stock only attends to information from other stocks within the same industry, thereby achieving attention sparsity.

3.4.3 Complexity Analysis

The computational bottleneck of standard self-attention lies in the matrix multiplication of QK^T , which has both computational and memory complexities of $O(N^2)$. After applying our DSA, since the attention computation is confined within individual industries, the total computational complexity is decomposed into the sum of complexities within each industry. Assuming there are m industries in total, and the size (number of stocks) of the m -th industry is k_m , the total complexity becomes:

$$O\left(\sum_{i=1}^m k_i^2\right)$$

Given that the size of an industry is typically much smaller than the total number of stocks, N , this complexity is far lower than $O(N^2)$. For example, in a market with 2000 stocks distributed across 10 industries (averaging 200 stocks per industry), the theoretical computational load is only about 1/10th of that of full attention. In the general case of a relatively balanced industry distribution, where the average industry size $k_{avg} \ll N$, the overall computational complexity approximates $O(Nk_{avg})$, achieving a leap from quadratic to near-linear complexity.

3.5 Loss Function

In quantitative investment, we are not only concerned with the absolute accuracy of predicted returns but, more importantly, with the correctness of their predicted rankings, as this directly determines the quality of long-short portfolios. To address both aspects, we designed a composite loss function that includes a regression component and a ranking component.

The first component is a **regression loss**, which utilizes the standard Mean Squared Error (MSE) to penalize the absolute deviation between predicted (y'_{it}) and true (y_{it}) values. For a set V of all valid stock-time pairs, the MSE loss is calculated as:

$$L_{mse} = \frac{1}{|V|} \sum_{(i,t) \in V} (y'_{it} - y_{it})^2$$

The second component is a **ranking loss**, designed to directly optimize the model's ranking capability. We employ the negative correlation coefficient between the predicted returns (y') and the true returns (y) for this purpose. This metric is equivalent to the Information Coefficient (IC) in finance, a key measure of predictive skill. The ranking loss is therefore defined as:

$$L_{rank} = -corr(y', y)$$

Finally, the **overall loss function** L is a weighted sum of these two components, governed by a hyperparameter α that balances the importance of regression accuracy and ranking capability. The final loss is expressed as:

$$L = L_{mse} - \alpha \cdot L_{rank}$$

By minimizing L , we expect the model to optimize the IC while ensuring its predicted values remain within a reasonable range.

4. Experiments

To comprehensively evaluate the effectiveness and efficiency of our proposed DART model, we conducted a series of experiments on three large-scale, public, real-world stock market datasets. This chapter will detail the datasets, evaluation metrics, baseline models, implementation details, and a thorough analysis of the experimental results.

4.1 Datasets

For our experiments, we utilize data from three major U.S. stock indices: the NASDAQ, S&P 500, and NYSE. These datasets provide a diverse representation of stocks across various industries and market capitalizations, forming a robust benchmark for evaluating the model's generalization capabilities. For each stock, the data comprises five core features from daily trading records: Open, High, Low, Close, and Volume (OHLCV). All datasets cover the time span from 2020 to 2024.

To simulate a realistic forecasting scenario, we adopt a chronological rolling-window approach for splitting the data. The entire time series is partitioned into a training set (60%), a validation set (20%), and a test set (20%). This strict temporal separation ensures that no future information leaks into the training or validation processes. A detailed summary of each dataset's statistics, including the precise number of stocks and trading days, is presented in Table 1.

Table 1
Dataset Information

Dataset	Number	Trading Days	Time Range	Feature	Train/Val/Test
NASDAQ	561	1148	2020–2024	5	60% / 20% / 20%
S&P 500	464	1148	2020–2024	5	60% / 20% / 20%
NYSE	1007	1148	2020–2024	5	60% / 20% / 20%

4.2 Evaluation Metrics

To comprehensively evaluate the model's performance, we employ a suite of metrics that assess its predictive accuracy, ranking capability, and practical investment value.

First, we assess the accuracy of the predicted return values. The **Mean Squared Error (MSE)** measures the absolute deviation between predicted returns (y'_{it}) and true returns (y_{it}), where a lower value indicates a stronger regression fit.

$$MSE = \frac{1}{|V|} \sum_{(i,t) \in V} (y'_{it} - y_{it})^2$$

In parallel, the **Information Coefficient (IC)**, defined as the Pearson correlation coefficient between predicted (y') and true (y) values, measures the directional correctness of the predictions. The IC ranges from -1 to 1 , with a higher absolute value indicating superior predictive power.

$$IC = \text{corr}(y', y)$$

Furthermore, to specifically evaluate the model's ability to correctly order stocks, we use the **Rank Information Coefficient (RIC)**, which is the Spearman's rank correlation between the predicted and true returns. This is crucial for portfolio construction, which relies on relative rankings.

Beyond statistical accuracy, we evaluate the model's practical utility. We calculate **Precision@10 (Prec@10)**, which is the proportion of stocks with positive true returns among the 10 stocks with the highest model-predicted returns, reflecting the model's "hit rate" in stock selection. Finally, we assess risk-adjusted performance by simulating a simple long-only strategy on the top k predicted stocks. We then compute the annualized **Sharpe Ratio** from the resulting daily return series. A higher Sharpe Ratio indicates that the strategy yields greater excess return for each unit of risk undertaken.

4.3 Experimental Setup

4.3.1 Baselines

To validate the superiority of our proposed model, we benchmark its performance against six representative baseline models, ranging from a simple linear method to recent advanced deep learning architectures.

The first baseline is a Linear Model, utilizing simple linear regression to serve as a fundamental performance benchmark. Next, we include the Long Short-Term Memory (LSTM) network, a classic recurrent neural network widely used for time series forecasting. In our implementation, the LSTM processes each stock's data independently, thus it cannot model inter-stock relationships. To incorporate such relationships, we employ a Graph Convolutional Network (GCN), which leverages an adjacency matrix based on industry classifications to model static inter-stock correlations. We also compare against STHAN-SR (Spatio-Temporal Hierarchical Attention Network for Stock Recommendation), an advanced baseline that simultaneously captures temporal dynamics and inter-stock influences through a hierarchical spatio-temporal attention network. In addition, two recent strong baselines are included under the same experimental setting: CausalStock, a causal-discovery-based stock prediction framework, and MambaStock, a selective state-space model tailored for stock forecasting (Li et al. 2024; Shi 2024).

4.3.2 Ablation Models

To verify the necessity and contribution of each innovative component in our model, we conducted a series of ablation studies using three variant models.

The **DART w/o Sparse** variant validates the effectiveness and efficiency of our sparsity strategy by replacing the industry sparse attention mechanism with standard full attention. To assess the importance of our multi-scale temporal modeling, the **DART w/o Fusion** variant removes the entire Multi-scale Temporal Fusion (MSTF) module and substitutes it with a simple linear layer. A more granular test, **DART w/o Prog**, isolates the impact of the progressive fusion architecture by removing only the Progressive Temporal Fusion (PTF) network from within the MSTF module, allowing for a focused evaluation of this specific structure.

4.3.3 Training Details and Hyperparameters

All models were implemented using the PyTorch framework and were trained and tested on a single NVIDIA GeForce RTX 4060 GPU. To ensure a fair comparison, the evaluation for all models on the same dataset followed identical procedures and data splits. For each model-dataset combination, we conducted independent hyperparameter tuning on the validation set to find the optimal performance configuration. The main hyperparameters and their search ranges are as follows:

- • Optimizer: Adam
- • Learning Rate: Selected from {0.0001, 0.00005, 0.00002}
- • Batch Size: Selected from {32, 64}
- • Epochs: 100
- • alpha in Loss Function: Selected from {0.5, 0.75, 0.8}
- • Random Seed: 42, to ensure the reproducibility of the experiments.

4.4 Comparative Experiments and Results Analysis

4.4.1 Overall Performance Comparison

Tables 2, 3, and 4 present the overall performance comparison between our model and all baseline models, including the recent CausalStock and MambaStock baselines, on the three datasets.

Table 2
Performance comparison on the NASDAQ dataset.

Model	IC	RIC	Prec@10	Sharpe
LinearModel	0.009	0.070	0.490	1.007
GCN	0.003	0.031	0.486	1.158
STHAN-SR	0.012	0.127	0.476	1.186
LSTM	0.011	0.116	0.503	1.827
CausalStock	0.010	0.066	0.488	1.379
MambaStock	0.019	0.191	0.530	0.617
DART	0.016	0.143	0.506	2.432

Table 3
Performance comparison on the S&P 500 dataset.

Model	IC	RIC	Prec@10	Sharpe
LinearModel	0.001	0.010	0.508	0.255
GCN	0.012	0.082	0.504	0.339
STHAN-SR	0.011	0.120	0.506	0.469
LSTM	0.010	0.110	0.521	0.711
CausalStock	0.032	0.206	0.512	1.142
MambaStock	0.027	0.189	0.513	1.866
DART	0.025	0.209	0.526	2.616

Table 4
Performance comparison on the NYSE dataset.

Model	IC	RIC	Prec@10	Sharpe
LinearModel	0.002	0.018	0.500	0.002
GCN	0.002	0.033	0.516	0.002
STHAN-SR	0.004	0.030	0.577	0.004
LSTM	0.005	0.065	0.536	0.005
CausalStock	0.018	0.112	0.511	1.770
MambaStock	0.013	0.101	0.516	2.358
DART	0.006	0.069	0.587	0.006

The results presented in these tables lead to several key observations. Across the three datasets, DART delivers the strongest overall risk-adjusted performance, achieving the highest Sharpe Ratios on NASDAQ (2.432) and S&P 500 (2.616) while maintaining competitive IC, RIC, and Prec@10 scores against all baselines. This indicates that our design is particularly effective at converting predictive signals into stable portfolio returns.

At the metric level, the newer baselines exhibit complementary strengths rather than uniform dominance. On NASDAQ, MambaStock attains slightly higher IC and Prec@10, but its Sharpe Ratio remains far below DART, whereas CausalStock is weaker overall. On S&P 500, CausalStock and MambaStock are competitive in IC and RIC, yet DART still preserves the best overall balance and the highest Sharpe Ratio. On NYSE, CausalStock and MambaStock achieve higher IC, RIC, and Sharpe Ratio values, while DART records the best Prec@10, indicating stronger discrimination among the top-ranked opportunities. Overall, these results suggest that the synergy of the MSTF module and DSA mechanism enables DART to learn more robust market signals, especially when performance is evaluated from the perspective of portfolio stability and top-stock selection.

4.4.2 Ablation Study

To investigate the contribution of each component of the model, we conducted an ablation study on the NASDAQ dataset. The results are shown in Table 5.

Table 5
Ablation study results on the NASDAQ dataset.

Model	IC	RIC	Prec@10	Sharpe
DART (Full Model)	0.016	0.143	0.506	2.432
DART w/o Sparse	0.001	0.010	0.475	1.371
DART w/o Fusion	0.009	0.079	0.494	1.122
DART w/o Prog	0.007	0.063	0.487	1.356

The performance of DART w/o Sparse drops sharply, which provides strong evidence that unconstrained, full attention captures a large number of spurious correlations in a high-noise financial environment, leading to model failure. Our domain-knowledge-guided mechanism essentially acts as a strong, domain-knowledge-based regularizer.

The performance degradation of DART w/o Fusion demonstrates that capturing multi-scale temporal patterns is crucial for understanding complex market dynamics.

The performance drop of DART w/o Prog validates the effectiveness of our cumulative effect modeling mechanism.

4.4.3 Efficiency Analysis

In real-world quantitative investment applications, the computational efficiency of a model is a key determinant of its scalability and practicality. To fairly measure the true algorithmic efficiency gains of our proposed sparse attention mechanism, we adopt Floating Point Operations (FLOPs), a recognized academic standard, as the core metric to avoid interference from hardware and underlying library optimizations that can affect timing results.

We focus on the matrix multiplication step within the attention module, which is the most computationally expensive part, and conduct a theoretical computational load analysis for full attention versus our sparse attention in a scenario with 1007 stocks (based on the NYSE dataset) and 56-dimensional embedding vectors.

The results are shown in Table 6. Traditional full attention requires 0.23 GFLOPs to complete the core computation for a single forward pass. In sharp contrast, our proposed DSA, by only calculating intra-industry correlations, drastically reduces this overhead to just 0.04 GFLOPs. This order-of-magnitude improvement in efficiency makes it possible to scale Transformer-based models from academic experiments involving a few hundred stocks to real-world trading scenarios covering thousands of stocks across the entire market, thus solving a critical scalability bottleneck in the field.

Table 6
Computational efficiency comparison (NYSE dataset).

Attention Type	Core Computation (GFLOPs)	Computation Reduction
Full Attention	0.23	-
DSA	0.04	84.27%

This substantial theoretical computation reduction of 84.27% stems directly from the fundamental advantages of our algorithmic design. It successfully reduces the complexity of attention calculation from a quadratic relationship with the total number of stocks, $O\left(N^2\right)$, to a near-linear level that is only dependent on the number of active intra-industry connections. This paves the way for scaling the model to much larger applications covering the entire market, which is of profound practical significance.

5. Conclusion

5.1 Work Summary

This paper addressed the core challenge of stock return prediction in financial markets by proposing a novel and efficient deep learning model, DART Traditional prediction models often fall short in both capturing complex time-series patterns and efficiently handling correlations among large-scale stocks, especially due to the $O\left(N^2\right)$ computational complexity issue of standard attention mechanisms, which severely limits their application in real-world investment scenarios.

To overcome these challenges, we designed a hierarchical model architecture. First, we proposed the MSTF module, which captures market dynamics at different frequencies through parallel multi-scale convolutional branches and innovatively uses the PTF network to progressively aggregate the fused temporal features. Second, we designed the DSA mechanism, which leverages the financial domain prior of industry classification to constrain attention computations to within more relevant industry sectors, thereby significantly reducing computational complexity from a quadratic level while maintaining predictive accuracy.

5.2 Experimental Achievements and Significance

In this work, we proposed and validated DART, a novel model for large-scale stock return prediction. Comprehensive experiments on three real-world datasets—NASDAQ, NYSE, and S&P 500—show that DART remains highly competitive against a broad range of baseline models, from simple linear methods to recent advanced architectures. Its particularly strong Sharpe Ratio performance on NASDAQ and S&P 500, together with its leading Prec@10 result on NYSE, underscores its potential for building robust risk-adjusted investment strategies.

The model's success is attributable to its innovative design; ablation studies confirmed that both the Multi-scale Temporal Fusion (MSTF) and Dynamic Sparse Attention (DSA) modules make indispensable contributions to its final performance. Beyond its accuracy, DART is highly efficient. A theoretical analysis of Floating Point Operations (FLOPs) shows that the DSA mechanism reduces the core computational load by over 84%, ensuring the model's scalability and feasibility for large-scale applications.

In summary, this research contributes a solution for stock return prediction that is both highly accurate and highly efficient, offering significant reference value for academic and industry practitioners in quantitative finance.

5.3 Limitations and Future Outlook

Although the DART model has achieved promising results, several directions can be explored further.

One key direction is to enhance the modeling of dynamic inter-stock relationships. The current industry sparse attention relies on static classifications, whereas real-world correlations often transcend these boundaries and evolve over time. Future work could thus explore dynamic graph structures or leverage natural language processing to extract more granular supply chain and competitive links from news and financial reports. Another promising avenue lies in the fusion of multi-modal data. While this paper focused on price and volume data, incorporating other modalities—such as macroeconomic indicators, market sentiment factors, and analyst ratings—could provide a more comprehensive market understanding and yield more robust predictions. Recent work on LLM-based sentiment prediction and forecast-driven portfolio construction further supports this direction (Darwish et al. 2025; Chen 2025; Ślusarczyk and Ślepaczuk 2025).

Furthermore, enhancing the model's interpretability is a valuable research direction. Visualizing the decision-making process, for instance by analyzing the weight distributions of the Dynamic Sparse Attention (DSA) mechanism and the patterns captured by the Multi-scale Temporal Fusion (MSTF) module, could transform the model from a 'black box' into a tool that provides fund managers with more insightful and trustworthy decision support.

Declarations

Funding

This work was supported by the Natural Science Foundation of Fujian Province (2025J011222).

Competing Interests

The authors have no relevant financial or non-financial interests to disclose.

Author Contribution

Ruoyu Huang: Conceptualization, Methodology, Software, Data Curation, Formal Analysis, Writing, Original Draft. Designed the overall framework of the DART model, implemented the relational transformer algorithm and experimental procedures, collected and preprocessed stock market datasets, conducted quantitative experiments and result analysis, and drafted the initial manuscript. Xiaoyu Lin: Conceptualization, Supervision, Writing, Review & Editing, Validation. Proposed the core research ideas and innovation points, guided the model design and experimental scheme, revised the manuscript critically, supervised the whole research process, and approved the final version of the paper. Yibao Mao: Formal Analysis, Visualization. Participated in experimental verification, analyzed the experimental ablation results, completed result visualization and comparative analysis, and sorted out the experimental discussion content. Jiqiang Yan: Resources, Validation, Writing, Review & Editing. Provided experimental resource support, verified the rationality and reproducibility of the experimental results, participated in manuscript revision and grammar polishing, and checked the integrity of the research content.

Acknowledgement

This work was supported by the Natural Science Foundation of Fujian Province (2025J011222).

Data Availability

The data that support the findings of this study were derived from Polygon.io market data and related reference information, including daily OHLCV records and industry classification metadata. Restrictions

apply to the availability of these data, which were used under license for this study and therefore cannot be publicly redistributed by the authors. Processed data products and code required to reproduce the reported experiments are available from the corresponding author upon reasonable request, subject to the data access terms of Polygon.io.

References

1. Akita R, Yoshii, Akira, Ono T, Uehara K (2016) Parannmt-1.0: A parallel corpus of news articles and their headlines. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), pp. 3173–3178
2. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271
3. Barberis N, Thaler R (2003) A survey of behavioral finance. In: Constantinides GM, Harris M, Stulz RM (eds) Handbook of the Economics of Finance vol, vol 1. Elsevier, Amsterdam, pp 1053–1128
4. Beltagy I, Peters ME, Cohan A (2020) Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150
5. Bollerslev T (1986) Generalized autoregressive conditional heteroskedasticity. J Econ 31(3):307–327
6. Box GEP, Jenkins GM, Reinsel GC, Ljung GM (2015) Time Series Analysis: Forecasting and Control, 5th edn. Wiley, Hoboken
7. Chen L, Chen Z, Zhai G (2016) Stock price movement prediction with candlestick chart and deep learning. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3912–3916
8. Chen Y, Wei Z, Huang X, Dai P (2021) Stock-tool-net: A graph neural network for stock selection with a semi-supervised learning approach. Inf Sci 551:106–120
9. Chen W, Liu W, Zheng J et al (2025) Leveraging large language model as news sentiment predictor in stock markets: a knowledge-enhanced strategy. Discover Comput 28:74
10. Cont R (2001) Empirical properties of asset returns: stylized facts and statistical issues. Quant Finance 1(2):223–236
11. Darwish M, Hassanien EE, Eissa AHB (2025) Stock Market Forecasting: From Traditional Predictive Models to Large Language Models. Computational Economics
12. Ding X, Zhang Y, Liu T, Duan J (2015) Deep learning for event-driven stock prediction. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp. 2327–2333
13. Ding H, Liu L, Zhang Z, Liu J (2020) Hierarchical multi-scale gaussian transformer for stock movement prediction. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 2548–2554
14. Engle RF (2001) The use of ARCH/GARCH models in applied econometrics. J Economic Perspect 15(4):157–168

15. Fama EF (1970) Efficient capital markets: A review of theory and empirical work. *J Finance* 25(2):383–417
16. Feng F, He, Xiangnan, Wang X, Luo C, Liu Y, Chua TS (2019) Temporal relational ranking for stock prediction. *ACM Trans Inform Syst (TOIS)* 37(2):1–30
17. Fischer T, Krauss C (2018) Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res* 270(2):654–669
18. Grinold RC, Kahn RN (2000) *Active Portfolio Management: a Quantitative Approach for Producing Superior Returns and Controlling Risk*. McGraw-Hill, New York
19. Gu S, Kelly B, Xiu D (2020) Empirical asset pricing via machine learning. *Rev Financial Stud* 33(5):2223–2273
20. Hamilton WL, Ying R, Leskovec J (2017) Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1025–1035
21. Heaton JB, Polson NG, Witte JH (2017) Deep learning for finance: deep portfolios. *Appl Stoch Models Bus Ind* 33(1):3–12
22. Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
23. Huang W, Nakamori Y, Wang SY (2005) Forecasting stock market movement direction with support vector machine. *Comput Oper Res* 32(10):2513–2522
24. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations (ICLR)*
25. Kong W, Tian G, Guevara MA, Zhou F (2024) Deep learning in asset pricing: A review and new evidence. *J Economic Dynamics Control* 159:104791
26. Krauss C, Do XA, Küchlin N (2017) Deep neural networks, gradient-boosted trees, and random forests: Statistical arbitrage on the S&P 500. *Eur J Oper Res* 259(2):689–702
27. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11), 2278–2324
28. Li S, Sun Y, Lin Y, Gao X, Shang S, Yan R (2024) CausalStock: Deep end-to-end causal discovery for news-driven multi-stock movement prediction. In: *Advances in Neural Information Processing Systems (NeurIPS)*
29. Mandelbrot B (1963) The variation of certain speculative prices. *J Bus* 36(4):394–419
30. Mantegna RN (1999) Hierarchical structure in financial markets. *Eur Phys J B-Condensed Matter Complex Syst* 11(1):193–197
31. Moskowitz TJ, Grinblatt M (1999) Do industries explain momentum? *J Finance* 54(4):1249–1290
32. Nelson DM, Pereira ACM, Oliveira RA (2017) Stock market’s price movement prediction with LSTM neural networks. In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8
33. Nie Y, Nguyen H, Xu K, Subedi DKC (2023) LightCTS: A lightweight framework for correlated time series forecasting. In: *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3959–3967

34. Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning (ICML), pp. 1310–1318
35. Plerou V, Gopikrishnan P, Rosenow B, Amaral LAN, Stanley HE (1999) Universal and nonuniversal properties of cross correlations in financial time series. *Phys Rev Lett* 83(7):1471
36. Prado M (2018) *Advances in Financial Machine Learning*. Wiley, Hoboken
37. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. *IEEE Trans Neural Networks* 20(1):61–80
38. Shi Z (2024) MambaStock: Selective state space model for stock prediction. *arXiv preprint arXiv:2402.18959*
39. Shiller RJ (2003) From efficient markets theory to behavioral finance. *J Economic Perspect* 17(1):83–104
40. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9
41. Tay Y, Dehghani M, Bahri D, Metzler D (2020) Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*
42. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008
43. Wang H, Xie D, Duan Y, Xiong W et al (2026) Optimizing stock market prediction and stock trading strategies with deep learning models enhanced by nonlinear feature identification and robust prediction evaluation. *Financial Innov* 12:117
44. Yao L, Li C, Wang S, Wang Z (2021) Ms-gcn: Multi-stream graph convolutional networks for stock movement prediction. *IEEE Access* 9:110539–110549
45. Ying C, Cai TB, Luo SC, Zheng SJ, Ke G, He D (2021) Do gnn-based stock prediction models really work? In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2269–2278
46. Zaheer M, Guruganesh G, Dubey KA, Ainslie J, Alberty C, Ontanon S, Pham P, Ravula A, Wang Q, Yang L et al (2020) Big bird: Transformers for longer sequences. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 16738–16751
47. Zhang C, Zhang Q, Sun L, Sun J (2023) MTS-Mixers: A simple and effective time series forecasting model. In: *Proceedings of the AAAI Conference on Artificial Intelligence* vol. 37, pp. 11215–11223
48. Zhao L, Li J, Wang H, Srivastava G (2023) KR-GCN: Knowledge-aware relational graph convolutional network for stock recommendation. *Expert Syst Appl* 213:119020
49. Ślusarczyk D, Ślepaczuk R (2025) Optimal Markowitz portfolio using returns forecasted with time series and machine learning models. *J Big Data* 12(1):127

Figures

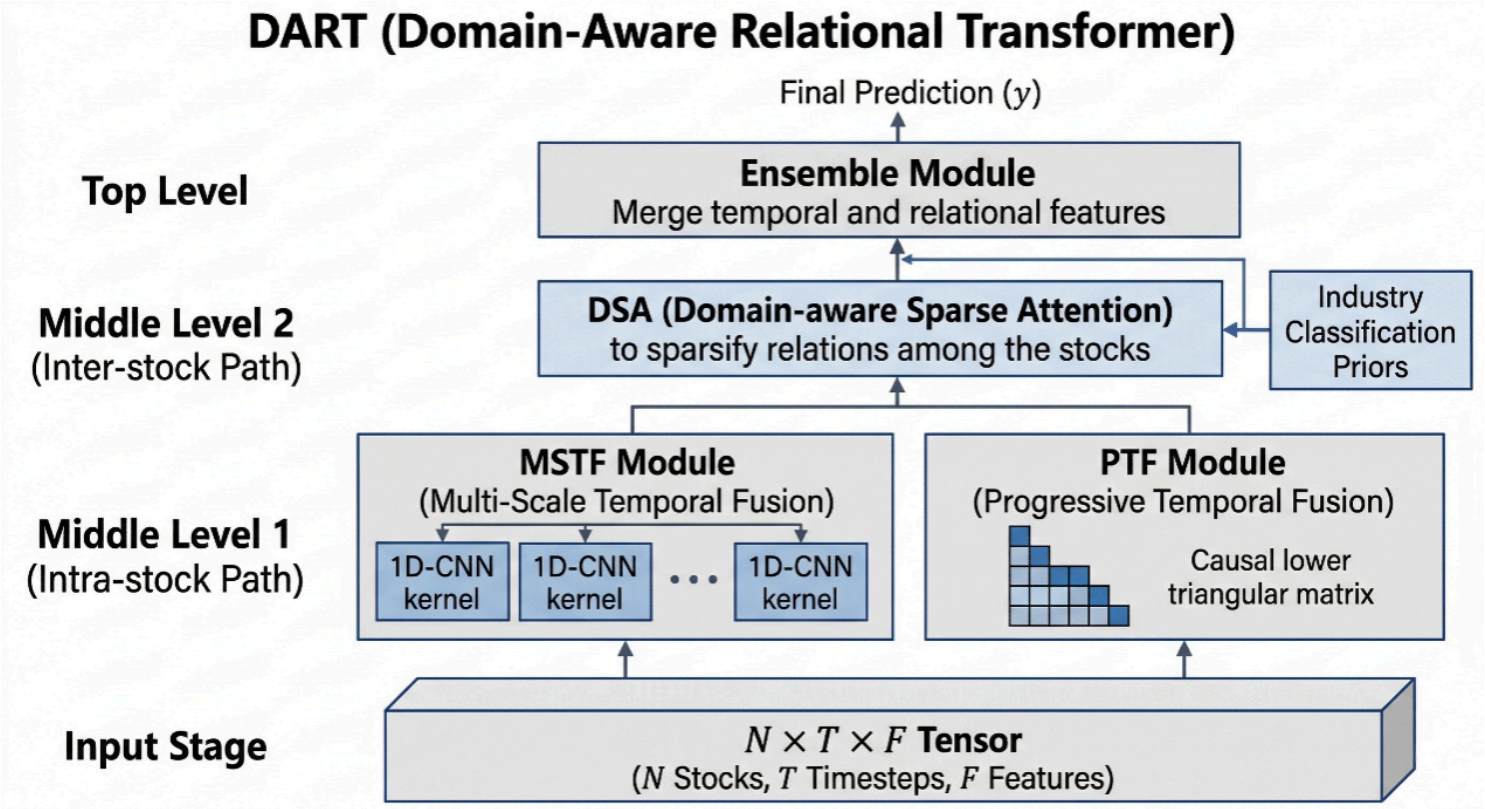


Figure 1

The proposed model architecture.

Causal Weight Matrix and Progressive Temporal Fusion

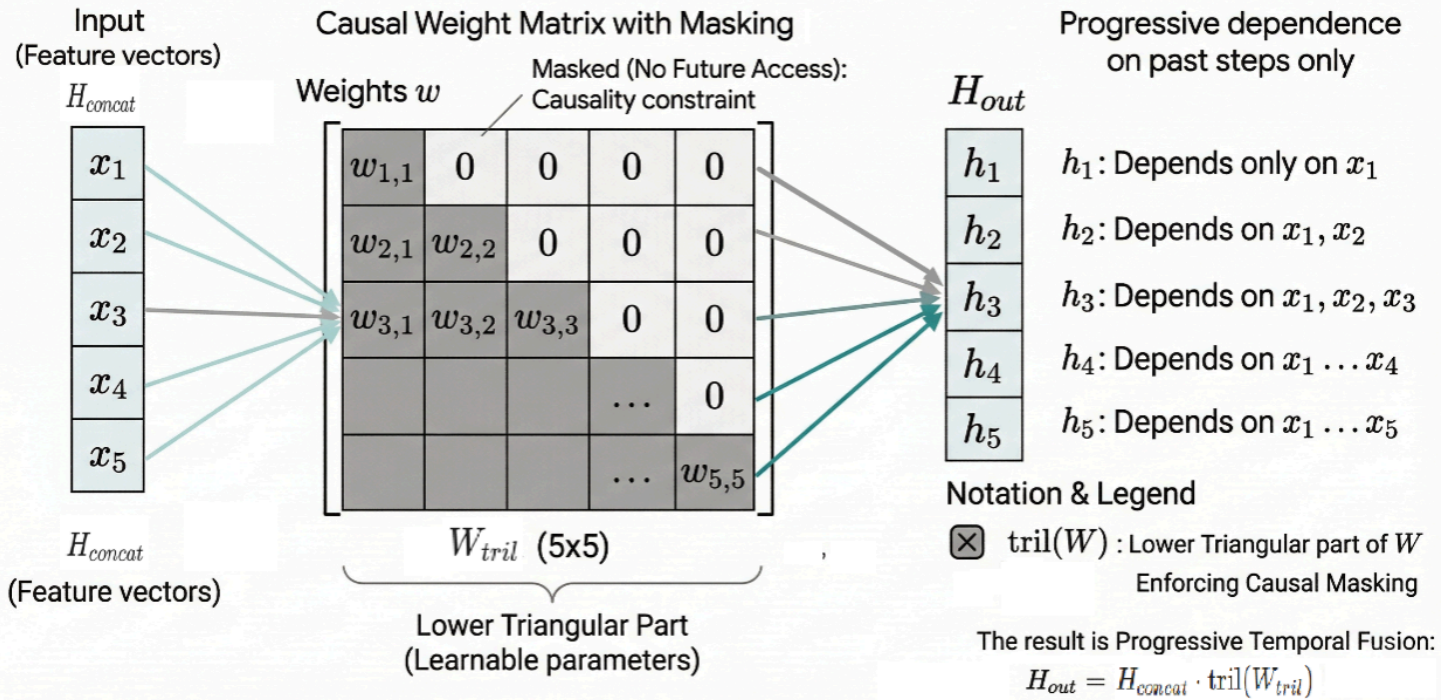


Figure 2

The Progressive Temporal Fusion (PTF) Network.