Preprint – IJCAI 2025: This is the accepted version made available for conference attendees.
Do not cite. The final version will appear in the IJCAI 2025 proceedings.

Dynamic Higher-Order Relations and Event-Driven Temporal Modeling for Stock Price Forecasting

Kijeong Park[1*] , **Sungchul Hong**[2] , **Jong-June Jeon**[3†]
[1]Gwanak Lab Co., Ltd., South Korea
[2]Changwon National University, South Korea
[3]University of Seoul, South Korea
krypton99@uos.ac.kr, shong@changwon.ac.kr, jj.jeon@uos.ac.kr

Abstract

In stock price forecasting, modeling the probabilistic dependence between stock prices within a time-series framework has remained a persistent and highly challenging area of research. We propose a novel model to explain the extreme co-movement in multivariate data with time-series dependencies. Our model incorporates a Hawkes process layer to capture abrupt co-movements, thereby enhancing the temporal representation of market dynamics. We introduce dynamic hypergraphs into our model adapting to higher-order (groupwise rather than pairwise) relationships within the stock market. Extensive experiments on real-world benchmarks demonstrate the robustness of our approach in predictive performance and portfolio stability.

1 Introduction

With a total market capitalization reaching approximately US$111 trillion in 2023[1], the stock market has experienced rapid growth, accompanied by increasingly complex dependencies among stock prices and their time-series characteristics. Built on years of breakthroughs in AI, many quantitative studies have adopted deep learning methods to model nonlinear relationships between stocks [Chen *et al.*, 2018; Xu *et al.*, 2021]. By employing sophisticated architectures, stock price forecasting enters a new phase in a complex pattern between stocks, modeling higher-order relationships.

The term 'higher-order relationship' extends the concept of stochastic dependence in two directions: it encompasses group-wise dependencies among stock prices and extreme dependencies represented by higher-order moments. For instance, stocks of companies within the same industry often exhibit similar reactions or trends in response to financial conditions. Furthermore, the synchronous movement of stock prices on average sometimes leads to an extreme co-movement of stocks within the same groups, such as abrupt crush, which is a crucial factor for risk management [Li *et al.*, 2021].

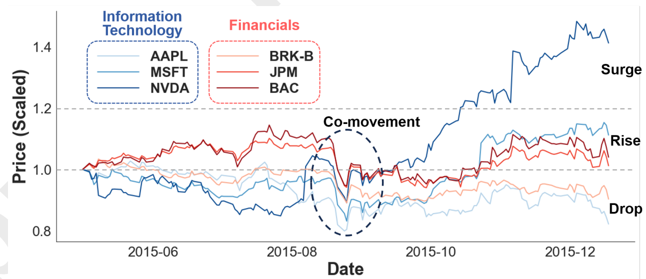

Figure 1: Illustration of complex higher-order relationships among stock prices.

To tackle these issues, existing deep learning methods typically incorporate hypergraph learning and the Hawkes process [Hawkes, 1971] in the predictive model. The hypergraph is a general concept of graphs to allow edges are able to join any number of nodes. Through the characteristics of a hypergraph, the nodes (in our context, stocks) can be connected with some hyperedge. Combining with graph neural networks (GNNs), hypergraph learning is applied to model higher-order relationships in stock price prediction [Sawhney *et al.*, 2021; Xia *et al.*, 2024]. The Hawkes process dynamically constructs the intensity function, effectively capturing the tail distribution of the time series. Abrupt temporal dependence in financial events such as transactions (buy and sell) are modeled with the Hawkes process [Zuo *et al.*, 2020; Sawhney *et al.*, 2021; Huynh *et al.*, 2023]. More details of the Hawkes process are presented in Section 3.

Despite these advancements, two critical challenges remain unaddressed. First, surges or plunges in stock prices often exhibit co-movement characteristics. For example, Figure 1 shows the scaled prices of assets across multiple sectors, normalized to 1 at the initial time point. It highlights a sharp decline across multiple stocks between August and September 2015, as indicated by a dashed circle in the figure. However, existing approaches frequently overlook simultaneous modeling for extremal and group-wise dependence across the stock prices, resulting in their inability to effectively explain the extreme co-movement. Second, the relationships among stocks are exceedingly complex, often beyond the scope of

*This work was conducted while the author was at the University of Seoul.
†Corresponding author.
[1]wikipedia.org/wiki/Market_capitalization

prior domain knowledge, and can dynamically evolve over time. As shown in Figure 1, stocks within the Information Technology sector exhibited noticeably different patterns during the same period. Furthermore, prior to October, stocks within the same sector demonstrated relatively similar movements, but significant divergences emerged afterward. At certain points, stocks from different sectors also exhibited similar movements, suggesting the possibility of temporary groupings across sectors.

In this paper, we propose a novel stock price forecasting model that effectively addresses both challenges. Our approach demonstrates significant enhancements by integrating temporal learning and hypergraph modeling in an end-to-end manner. We fully leverage the use of self-attention mechanisms to enhance temporal learning efficiency. Additionally, we incorporate a Hawkes process layer to capture abrupt changes and co-movement patterns. Finally, we combine domain knowledge-based and dynamic hypergraphs to model evolving relationships over time. These approaches collectively enable our model to achieve superior predictive performance. Our main contributions are summarized as follows:

- A novel neural network layer is proposed to model temporal point processes, with shared movement patterns among stocks effectively captured, temporal representations enriched, and portfolio stability ensured.
- A novel hypergraph learning framework is introduced, where dynamic and predefined hypergraphs are integrated to enable the modeling of dynamic higher-order relationships over time.
- The effectiveness of the proposed approach is demonstrated on real-world datasets, with superior performance shown compared to state-of-the-art baselines, and the significance of the proposed components substantiated.

The remainder of this paper is organized as follows: Section 2 reviews recent studies, and Section 3 provides background on our research. Section 4 describes task definition and our proposed model. Section 5 presents the experimental results and additional analyses. Finally, concluding remarks and directions for future work are presented in Section 6.

2 Related Works

2.1 Transformer-based Model

The transformer [Vaswani *et al.*, 2017] represents a groundbreaking advancement in deep learning, utilizing a self-attention mechanism to effectively model long-term dependencies. Its success in natural language processing led to its adaptation in various domains, including time series forecasting [Zhou *et al.*, 2021; Wu *et al.*, 2021]. In the context of stock price forecasting, unique challenges arise, requiring models that not only capture temporal patterns of individual stocks but also model complex inter-stock relationships. Several models tailored to address these challenges have been developed. HMG-TF [Ding *et al.*, 2020], a transformer-based model, is specifically designed to account for the unique characteristics of financial markets, including local dependencies and hierarchical structures. DTML [Yoo *et al.*, 2021] employs attention mechanisms along both the time and data axes to learn asymmetric and dynamic stock correlations. MASTER [Li *et al.*, 2024], building upon a framework akin to DTML, adopts a sequential attention mechanism across time, stock, and time dimensions to model cross-time correlations effectively.

2.2 Graph-based Model

Graph Neural Networks (GNNs) [Scarselli *et al.*, 2008] are designed to learn patterns from graph-structured data and have been widely applied in various domains, including social networks, traffic flow, and chemical structure. In stock price forecasting, GNNs have been utilized to model intricate dependencies and enhance predictive performance. RSR [Feng *et al.*, 2019a] models stock relationships using industry and metadata information sourced from the web, allowing the strength of these relationships to evolve over time. RT-GCN [Zheng *et al.*, 2023] enhances relational modeling by constructing three-dimensional graphs that incorporate both temporal and relational information from sources like industry. Recently, the limitations of conventional graph models, which can only handle pairwise relationships, have motivated the development of hypergraph-based stock price forecasting models. Details on hypergraphs are discussed further in Section 3.2. STHAN-SR [Sawhney *et al.*, 2021], similar to RSR, uses industry and Wikidata sources to construct hypergraphs and models higher-order relationships through hypergraph convolution and an attention mechanism. ESTIMATE [Huynh *et al.*, 2023] applies correlation augmentation to industry-based hypergraphs and employs wavelet hypergraph convolution to capture locality and ensure computational efficiency.

3 Preliminaries

3.1 Hawkes Process

A Temporal Point Process (TPP) is a stochastic process that models discrete events occurring over a continuous time axis [Cox and Isham, 1980]. TPPs are particularly useful for describing the temporal distribution of events or analyzing specific patterns and interactions, with applications in fields such as finance.

The Hawkes process [Hawkes, 1971] is a type of TPPs characterized by its "*self-exciting*" nature, meaning that the occurrence of an event increases the probability of subsequent events. The intensity function of a Hawkes process is defined as:

$$\lambda(t) = \mu + \sum_{i:t_i<t} \phi(t - t_i),$$

where $\lambda(t)$ is the conditional intensity function representing the expected event rate at time t, $\mu \geq 0$ is the base intensity, representing the exogenous occurrence of events, and $\phi(t) > 0$ is a prespecified decaying function, i.e., $\phi(t) = \exp(-t)$. Here, t_i denotes the occurrence time of the i-th event. By introducing the regression predictor the Hawkes process can describe how past events influence future occurrences dynamically.

The Hawkes process has been applied in the stock market to model extreme price movements and analyze correlations between events [Bacry *et al.*, 2015; Embrechts *et al.*, 2011]. Its self-exciting nature effectively captures scenarios where a sharp decline in one stock triggers movements in others or extreme volatility events. These applications offer valuable insights into financial risk management and investment strategies.

3.2 Hypergraph Learning

Traditional graphs are limited to modeling pairwise relationships, where edges connect only two nodes. Hypergraphs address this limitation by allowing a single edge to connect multiple nodes, enabling the modeling of higher-order relationships among data. This capability effectively extends connectivity between nodes to capture complex relationships involving multiple nodes.

Hypergraph learning was first introduced in [Zhou *et al.*, 2006], as a propagation process on hypergraph structures. Following the existing works [Bai *et al.*, 2021; Feng *et al.*, 2019b], an undirected hypergraph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of hyperedges, and $\boldsymbol{W}$ is a diagonal matrix representing the weights assigned to each hyperedge. The structure of a hypergraph can be represented using the $|\mathcal{V}| \times |\mathcal{E}|$ incidence matrix $\mathcal{H}$, where the entries are defined as:

$$h(v, e) = \begin{cases} 1, & \text{if } v \in e, \\ 0, & \text{if } v \notin e. \end{cases}$$

This incidence matrix provides a flexible representation for capturing complex higher-order relationships that cannot be modeled using traditional pairwise connections.

4 Methodology

4.1 Problem Formulation

In this paper, bold uppercase letters denote matrices or tensors (e.g., $\boldsymbol{X}, \boldsymbol{H}$), bold lowercase letters represent vectors (e.g., $\boldsymbol{x_t^s}, \boldsymbol{h_t^s}$), and scalars are represented by non-bold lowercase letters (e.g., c_t^s, r_t^s). We utilize normalized OHLCV data [Huynh *et al.*, 2023] along with technical indicators (moving average, MACD, RSI) to predict the change in stock price rather than the absolute price. OHLCV data includes open, high, low, close, and volume values, and the relative price change of these values is referred to as the OHLCV ratio. The historical data consists of both the OHLCV ratio and technical indicators. The return ratio, which is the relative close price change in τ days, is defined as $r_{t+\tau}^s = \frac{c_{t+\tau}^s - c_t^s}{c_t^s}$, where c_t^s is the closing price of stock s at time step t, and τ represents the prespecified lookahead window. The return ratio allows for the normalization of price variations between different stocks, as compared to price changes. Based on the data constructed as described above, we define our task as follows:

Definition 1. *Given the historical data $\boldsymbol{X} \in \mathbb{R}^{S \times T \times F}$, where each element $\boldsymbol{x}_t^s \in \mathbb{R}^F$ is a feature vector for stock $s \in \{1, \ldots, S\}$ at time $t \in \{1, \ldots, T\}$, the task of Stock Price Forecasting is to predict a return ratio $\boldsymbol{r} \in \mathbb{R}^S$ using a model parameterized by θ. Formally, the goal is to learn a function f such that:*

$$\hat{\boldsymbol{r}}_{T+\tau} = f(\boldsymbol{X}; \theta) \in \mathbb{R}^S,$$

where θ denotes the parameters of the model.

The optimal parameters $\hat{\theta}$ are obtained by minimizing a risk function $\mathcal{L}$:

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\boldsymbol{r}_{T+\tau}, f(\boldsymbol{X}; \theta)).$$

We present an overview of our framework in Figure 2.

4.2 Temporal Dynamics Representation Learning

Feature Embedding. Embedding input features into a multi-dimensional latent space is known to improve the model's ability to capture complex temporal patterns in time series data [Lim *et al.*, 2021; Hong *et al.*, 2024]. To achieve this, we use a simple fully connected layer, which efficiently transforms the rescaled feature vectors into a latent space. Since we use self-attention mechanisms in subsequent stages, it is crucial to preserve the temporal order during model training. Thus, we apply sinusoidal positional encoding, as proposed in [Vaswani *et al.*, 2017] at the initial embedding stage to ensure that the model maintains temporal information throughout the prediction window.

$$\boldsymbol{z}_t^s = (\boldsymbol{W}_e \boldsymbol{x}_t^s + \boldsymbol{b}_e) + \mathrm{PE},$$

where $\boldsymbol{W}_e \in \mathbb{R}^{d_e \times F}$ and $\boldsymbol{b}_e \in \mathbb{R}^{d_e}$ are learnable weights and biases, respectively, and PE is sinusoidal positional encoding.

Intra-Stock Attention. RNN-based layers were frequently employed to model the temporal dynamics of individual stocks. However, RNNs are prone to long-term dependency issues, where essential historical information may degrade over time, leading to suboptimal predictions.

To address this, we leverage self-attention mechanisms [Vaswani *et al.*, 2017] to efficiently capture temporal dependencies. To preserve the sequential nature of the data, we apply a temporal mask. The temporal self-attention output for a stock s is given by

$$\tilde{\boldsymbol{H}}^s = \mathrm{Attention}(\boldsymbol{Q}^s, \boldsymbol{K}^s, \boldsymbol{V}^s) = \mathrm{Softmax}(\frac{\boldsymbol{Q}^s \boldsymbol{K}^{s\top}}{\sqrt{d_k}})\boldsymbol{V}^s,$$

where $\boldsymbol{Q}^s$, $\boldsymbol{K}^s$, and $\boldsymbol{V}^s$ are the query, key, and value matrices obtained through linear transformations of $\boldsymbol{Z}^s = (\boldsymbol{z}_t^s : t = 1, \ldots T) \in \mathbb{R}^{T \times d_e}$, respectively. Given the complex time series, we further enhance the model's representations by incorporating multi-head attention, allowing the model to focus on different aspects of temporal features. The outputs of these multiple attention heads are concatenated and then linearly transformed to integrate the information from each head. Additionally, a residual connection is applied to preserve the important stock characteristics. After the multi-head attention, the outputs $\tilde{\boldsymbol{H}}^s \in \mathbb{R}^{T \times d_e}$ are passed through two fully connected layers with a ReLU activation and dropout, followed by a residual connection.

$$\boldsymbol{H}^s = (\mathrm{ReLU}(\tilde{\boldsymbol{H}}^s \boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2) + \tilde{\boldsymbol{H}}^s,$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d_e \times d_e}$ and $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^{d_e}$ are learnable weights and biases, respectively, and $\boldsymbol{H} = (\boldsymbol{H}^s : s = 1, \ldots, S) \in \mathbb{R}^{S \times T \times d_e}$ is historical context.

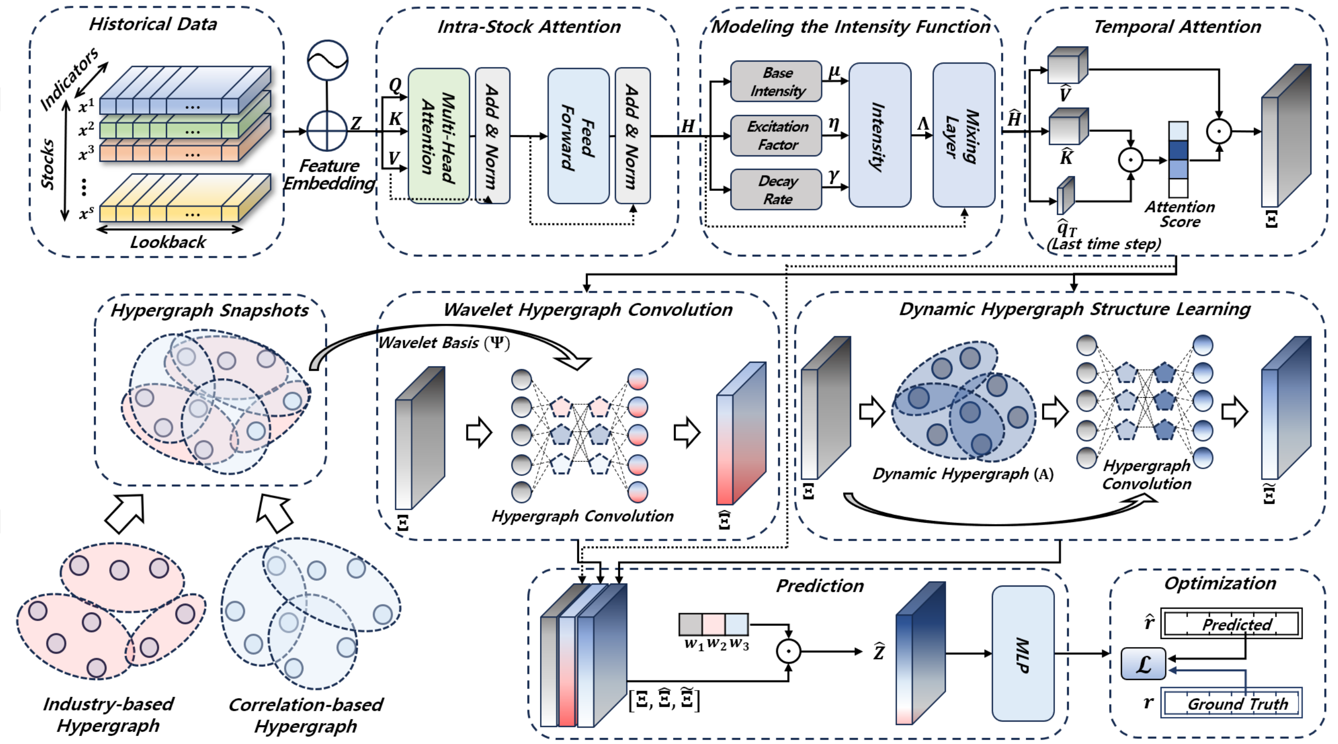

Figure 2: Overview of our framework.

4.3 Enhancing and Aggregating Temporal Representation

Modeling the Intensity Function of Hawkes Process. Existing works [Sawhney *et al.*, 2021; Huynh *et al.*, 2023] have utilized Hawkes process-based temporal attention mechanisms to address the non-stationary nature of stock markets and improve the ability of temporal attention mechanisms to capture temporal dynamics. However, the use of static parameterization alone is insufficient to accurately model the dynamic interactions between events.

To leverage the dynamic and self-exciting properties of the Hawkes process, we introduce a layer that directly models its intensity function. We utilize a slice of the historical context $\boldsymbol{H}$ as a predictor to learn the base intensity ($\boldsymbol{\mu}_t^s$), excitation factor ($\boldsymbol{\eta}_t^s$), and decay rate(γ_t^s) [Zhang *et al.*, 2020]. These quantities are parametrized by

$$\begin{aligned} &\boldsymbol{\mu}_t^s = \mathrm{GeLU}(\boldsymbol{W}_\mu \boldsymbol{h}_t^s),\ \boldsymbol{\eta}_t^s = \mathrm{GeLU}(\boldsymbol{W}_\eta \boldsymbol{h}_t^s), \\ &\gamma_t^s = \mathrm{Softplus}(\boldsymbol{W}_\gamma \boldsymbol{h}_t^s), \end{aligned} \tag{1}$$

where $\boldsymbol{h}_t^s$ is a slice of $\boldsymbol{H}$ for stock s at time t, $\boldsymbol{W}_\mu, \boldsymbol{W}_\eta, \boldsymbol{W}_\gamma \in \mathbb{R}^{d_e \times d_e}$ are learnable parameters. To ensure that γ_t^s is always positive, we applied the softplus activation function as shown in Eq. (1). To model the co-movement of stocks during extreme market conditions, we construct three parameters using shared weights across all stocks. This approach allows the model to capture the synchronized behavior of stocks that emerge during periods of high market volatility (refer to Section 5.3).

Finally, we express the intensity function of Hawkes process as follows:

$$\boldsymbol{\lambda}_t^s = \mathrm{Softplus}\Big(\boldsymbol{\mu}_t^s + (\boldsymbol{\eta}_t^s - \boldsymbol{\mu}_t^s) \sum_{j=1}^{t} \exp\left(-\gamma_j^s (t-j)\right)\Big).$$

Similarly, to ensure that the intensity function $\boldsymbol{\lambda}_t^s$ always remains positive, we applied the softplus activation function. The Hawkes process, originally designed for modeling influences in continuous time, is adapted in this study for discrete daily stock data by refining the intensity function to accumulate past information.

To construct richer temporal representations, the historical context $\boldsymbol{H}^s \in \mathbb{R}^{T \times d_e}$ for stock s and the intensity function output $\boldsymbol{\Lambda}^s = (\boldsymbol{\lambda}_t^s : t = 1, \dots T) \in \mathbb{R}^{T \times d_e}$ from the Hawkes process are combined and transformed, allowing the model to effectively integrate both temporal dynamics and event intensity.

$$\hat{\boldsymbol{H}}^s = \mathrm{ReLU}([\boldsymbol{H}^s : \boldsymbol{\Lambda}^s]\boldsymbol{W}_3 + \boldsymbol{b}_3),$$

where $[\cdot : \cdot]$ denotes the tensor concatenation operation, $[\boldsymbol{H}^s : \boldsymbol{\Lambda}^s] \in \mathbb{R}^{T \times 2d_e}$ $\boldsymbol{W}_3 \in \mathbb{R}^{2d_e \times d_e}, \boldsymbol{b}_3 \in \mathbb{R}^{d_e}$ are learnable parameters and $\hat{\boldsymbol{H}}^s \in \mathbb{R}^{T \times d_e}$ are enhanced temporal representations for stock s.

Aggregation for Temporal Information. To effectively aggregate temporal information across all time steps and generate final embeddings for each stock, we utilized temporal attention. In contrast to general self-attention, temporal attention employs the context vector $\hat{\boldsymbol{q}}_T^s \in \mathbb{R}^{d_e}$ from the last time step T as the query, enabling attention across sequences of different lengths and aggregating information with a focus on the last time step. This approach was adopted because the preceding layers were designed during training to ensure that the most critical information is concentrated near the last time step, which is closest to the prediction timestamp, thereby making it natural to base the aggregation on the query vector from the last time step.

$$\hat{\boldsymbol{q}}_T^s = \hat{\boldsymbol{W}}_q \hat{\boldsymbol{h}}_T^s, \qquad \hat{\boldsymbol{k}}_t^s = \hat{\boldsymbol{W}}_k \hat{\boldsymbol{h}}_t^s, \qquad \hat{\boldsymbol{v}}_t^s = \hat{\boldsymbol{W}}_v \hat{\boldsymbol{h}}_t^s,$$

$$a_t^s = \frac{\exp(\hat{\boldsymbol{q}}_T^{s\top} \hat{\boldsymbol{k}}_t^s / \sqrt{d_e})}{\sum_{t=1}^{T} \exp(\hat{\boldsymbol{q}}_T^{s\top} \hat{\boldsymbol{k}}_t^s / \sqrt{d_e})}, \quad \boldsymbol{\xi}^s = \sum_{t=1}^{T} a_t^s \hat{\boldsymbol{v}}_t^s,$$

where a_t^s represents the temporal attention score, and $\boldsymbol{\xi}^s \in \mathbb{R}^{d_e}$ is the representation that aggregates the temporal information for stock s. Finally, the representations for all stocks are concatenated into $\boldsymbol{\Xi} = (\boldsymbol{\xi}^s : s = 1, \dots S) \in \mathbb{R}^{S \times d_e}$, referred to as the stock representations.

4.4 Higher-Order Relational Representation Learning via Hypergraph

Constructing the Predefined Hypergraph. As introduced in Section 3.2, we construct a predefined hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ to model the intricate relationships in the stock market. Here, $\mathcal{V}$ denotes the set of nodes (individual stocks), $\mathcal{E}$ represents the hyperedges (groups of related stocks), and $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ is a diagonal matrix of hyperedge weights, assigned equally for simplicity. The hyperedge set $\mathcal{E}$ combines two components: $\mathcal{E}_{Ind}$, connecting stocks within the same industry [King, 1966], and $\mathcal{E}_{Corr}$, derived from historical price correlations [Bennett *et al.*, 2022]. The final hyperedge set is expressed as $\mathcal{E} = \mathcal{E}_{Ind} \cup \mathcal{E}_{Corr}$. For details on the hypergraph construction methodology, we refer the readers to our supplementary materials.

Wavelet Hypergraph Convolution. To model domain knowledge-based correlations among stocks, we use hypergraph convolution. We adopt the wavelet hypergraph convolution proposed in [Sun *et al.*, 2021; Huynh *et al.*, 2023], which balances efficiency and expressiveness by leveraging a sparse wavelet basis. The hypergraph Laplacian $\boldsymbol{\Delta}$ represents structural relationships in the constructed hypergraph $\mathcal{G}$ and is defined as $\boldsymbol{\Delta} = \boldsymbol{I} - \boldsymbol{D}_v^{-\frac{1}{2}} \boldsymbol{\mathcal{H}} \boldsymbol{W} \boldsymbol{D}_e^{-1} \boldsymbol{\mathcal{H}}^\top \boldsymbol{D}_v^{-\frac{1}{2}}$, where $\boldsymbol{\mathcal{H}}$ is the incidence matrix, $\boldsymbol{D}_v$ and $\boldsymbol{D}_e$ are diagonal degree matrices of nodes and hyperedges. By diagonalizing $\boldsymbol{\Delta}$ as $\boldsymbol{\Delta} = \boldsymbol{U} \boldsymbol{\Lambda}_f \boldsymbol{U}^\top$, we obtain the the Fourier basis $\boldsymbol{U}$, where $\boldsymbol{\Lambda}_f$ is the diagonal matrix of eigenvalues. Using the Fourier basis, the wavelet basis $\boldsymbol{\Psi}$ is computed as $\boldsymbol{\Psi} = \boldsymbol{U} \boldsymbol{\Lambda}_s \boldsymbol{U}^\top$, with $\boldsymbol{\Lambda} s = \mathrm{diag}(e^{-\lambda_1 s}, \dots, e^{-\lambda_n s})$, where s controls the scale of localization. Finally, the wavelet hypergraph convolution is expressed as:

$$\hat{\boldsymbol{\Xi}} = \mathrm{LeakyReLU}\left(\boldsymbol{\Psi} \boldsymbol{\Lambda}_\beta \boldsymbol{\Psi}^{-1} \boldsymbol{\Xi} \boldsymbol{W}_H\right),$$

where $\boldsymbol{\Lambda}_\beta = \mathrm{diag}(\beta_1, \dots, \beta_S)$ and $\boldsymbol{W}_H \in \mathbb{R}^{d_e \times d_e}$ are learnable parameters. $\hat{\boldsymbol{\Xi}} \in \mathbb{R}^{S \times d_e}$ contains temporal dynamics combined with prior relational knowledge.

Dynamic Hypergraph Structure Learning. In this study, we leverage the low-rank approaches proposed in [Zhao *et al.*, 2023; Ju *et al.*, 2024] to construct and learn dynamic hypergraphs, achieving a balance between computational efficiency and modeling capability. The incidence matrix $\boldsymbol{A} \in \mathbb{R}^{S \times k}$ of the dynamic hypergraph is generated as shown in Eq. (2). To construct the dynamic hypergraph, it is essential to incorporate information that varies across time steps. Utilizing stock representations $\boldsymbol{\Xi}$, derived from preceding layers that effectively capture temporal dynamics, encapsulates such time-varying information.

$$\boldsymbol{A} = \mathrm{Softmax}(\boldsymbol{\Xi} \boldsymbol{W}_I), \tag{2}$$

where $\boldsymbol{W}_I \in \mathbb{R}^{d_e \times k}$ is a learnable weight matrix, and k denotes the number of dynamic hyperedges.

After constructing a dynamic hypergraph, we utilize hypergraph convolution to obtain high-level dynamic stock representations. First, the information from neighboring nodes is aggregated based on the dynamic hypergraph structure to compute the dynamic hyperedge embedding matrix $\boldsymbol{E}$. Subsequently, the learned dynamic hyperedge embedding matrix $\boldsymbol{E}$ is used to update the node representations, resulting in the dynamic representation $\tilde{\boldsymbol{\Xi}} \in \mathbb{R}^{S \times d_e}$. The process can be formulated as follows:

$$\boldsymbol{E} = \mathrm{ReLU}(\boldsymbol{R} \boldsymbol{A}^\top \boldsymbol{\Xi}) + \boldsymbol{A}^\top \boldsymbol{\Xi}, \quad \tilde{\boldsymbol{\Xi}} = \boldsymbol{A} \boldsymbol{E},$$

where $\boldsymbol{R} \in \mathbb{R}^{k \times k}$ are learnable parameters, k denotes the number of dynamic hyperedges.

4.5 Model output

We combine the stock representations $\boldsymbol{\Xi}$, predefined representations $\hat{\boldsymbol{\Xi}}$, and dynamic representations $\tilde{\boldsymbol{\Xi}}$ using a weighted sum:

$$\hat{\boldsymbol{Z}} = w_1 \boldsymbol{\Xi} + w_2 \hat{\boldsymbol{\Xi}} + w_3 \tilde{\boldsymbol{\Xi}},$$

where $w_1, w_2, w_3 \in \mathbb{R}$ are learnable weights. The final representations $\hat{\boldsymbol{Z}} \in \mathbb{R}^{S \times d_e}$ are used to predict the return ratio for each stock via 2-layer MLP with ReLU activation. The loss function is defined as the RMSE between the ground truth return ratio $\boldsymbol{r}$ and the predictions $\hat{\boldsymbol{r}}$.

$$\hat{\boldsymbol{r}} = \mathrm{MLP}(\hat{\boldsymbol{Z}}), \quad \boldsymbol{\mathcal{L}} = \mathrm{RMSE}(\boldsymbol{r}, \hat{\boldsymbol{r}}).$$

The effectiveness of this methodology will be demonstrated and validated in the subsequent experimental sections.

5 Experiments

In this section, we conduct experiments to answer the following three research questions:

RQ1. How does the performance of our proposed approach compare with state-of-the-art methods?

RQ2. Does our approach effectively capture tail dependencies?

RQ3. Is dynamic hypergraph learning effectively integrated and implemented in our approach?

Additional details and results (including ablation study) from our numerical studies are provided in the supplementary materials and are also available in our online repository at https://github.com/kijeong22/ijcai2025-spf.

5.1 Experimental Setup

Dataset. To evaluate our model, we used real-world S&P500 market data, adapting the data collection and construction methods from [Huynh *et al.*, 2023]. Adjustments were made to the data period, stock count, and phase structure to account for differences in historical data. Specifically, stocks not listed in earlier years were excluded, reducing the dataset to 463 stocks. The dataset was divided into 10 phases, each comprising a 24-month training period, a 4-month validation period, and an 8-month test period.

Implementation Details. Our model was implemented in PyTorch, and all experiments were conducted on an NVIDIA RTX A6000 GPU. To identify the optimal hyperparameters, we employed grid search and the best hyperparameters were selected based on the validation-averaged IC across all phases, resulting in model size $d_e = 32$, the number of dynamic hyperedges $k = 64$, and learning rate $= 0.0001$. The lookback window was set to 20 days, the lookforward window to 5 days, and the number of heads in the multi-head attention mechanism to 4. For the trading simulation, we adopt a top-K daily buy-hold-sell trading strategy used in [Feng *et al.*, 2019a], setting $K = 20$.

Metrics. The evaluation metrics are categorized into predictive performance and portfolio performance. For predictive performance, we use Information Coefficient (IC), Rank Information Coefficient (RankIC), and Precision@K (Prec@K) to evaluate the alignment, ranking accuracy, and top-K precision of predictions, respectively. Portfolio performance metrics include Return, Sharpe Ratio (SR), and Maximum Drawdown (MDD), which assess profitability, risk-adjusted return, and maximum loss during the investment period.

Baselines. We compare our model with state-of-the-art models from three categories: Transformer-based models, Graph-based models, and Hypergraph-based models. These include DTML [Yoo *et al.*, 2021] and MASTER [Li *et al.*, 2024] for Transformer-based models, RSR [Feng *et al.*, 2019a] and RT-GCN [Zheng *et al.*, 2023] for Graph-based models, and STHAN-SR [Sawhney *et al.*, 2021] and ESTIMATE [Huynh *et al.*, 2023] for Hypergraph-based models.

5.2 RQ1 - Overall Performance

Table 1 presents the predictive performance of all comparison models across phases and their overall average. Most baselines were evaluated using their original configurations and the same loss function for fairness. However, DTML, designed for the classification of stock movements, employed RMSE as its loss function to align with the evaluation metrics. Our proposed model demonstrates superior performance in most phases compared to other baselines, and its overall average scores for IC, RankIC, and Prec@10 also outperform all other models. For ESTIMATE, the IC and RankIC scores are the highest in phases 2 and 4, and its average values across all metrics are the best among the baselines. This can be attributed to its architecture, which not only shares structural similarities with our model but also utilizes a predefined hypergraph for learning higher-order relationships. Table 2 presents the average portfolio performance of all comparison

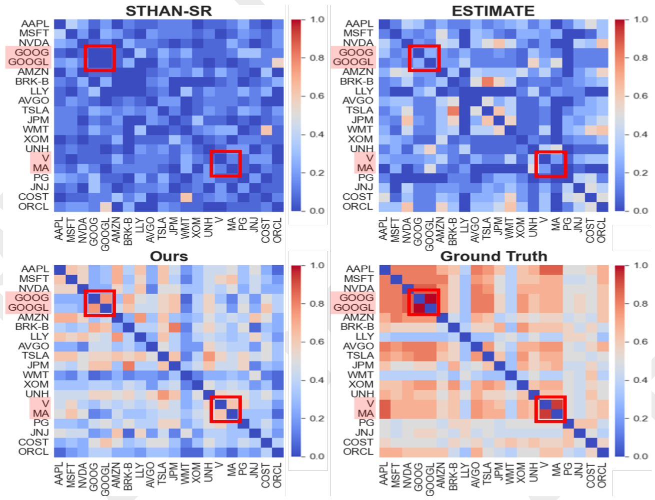

Figure 3: Visualization of the LTDC matrix for the Top 20 Stocks by Market Capitalization in phase 8.

models across phases. Our proposed model consistently outperforms others in Return, SR, and MDD. Notably, SR and MDD, which reflect portfolio risk, highlight the capability of our model to capture abrupt stock movements and construct a stable portfolio throughout the trading period.

5.3 RQ2 - Lower Tail Dependence Coefficient

We evaluate the Lower Tail Dependence Coefficient (LTDC), which measures the tendency of simultaneous extreme downward movements, a critical factor influencing investor behavior and portfolio performance [Sibuya and others, 1960; Ang *et al.*, 2006]. LTDC [Joe, 1997] is defined as follows:

$$\lambda_{i,j}^{\ell} = \lim_{q \to 0^+} \mathbb{P}\left(X_j \leq F_j^{-1}(q) \mid X_i \leq F_i^{-1}(q)\right),$$

where i and j denote stock indices, and $F^{-1}(q)$ is the inverse cumulative distribution function. We compute LTDC for all stock pairs to construct a $|\mathcal{S}| \times |\mathcal{S}|$ stock LTDC matrix. To calculate LTDC, we use the empirical cumulative distribution function due to its simplicity and effectiveness in handling sample data without assuming a specific distribution.

Figure 3 shows the LTDC matrix for phase 8, visualizing the top 20 companies by market capitalization. Our model relatively closely matches the ground truth, successfully capturing relationships, such as the near-identical movements of GOOG and GOOGL (same company) and the highly correlated stocks V and MA (with an almost identical business model). In contrast, the other two models fail to capture these relationships, demonstrating the effectiveness of designing the intensity function of the Hawkes process as a separate layer in accurately modeling LTDC.

5.4 RQ3 - Analysis of the Dynamic Hypergraph Structure

We analyze the dynamic hypergraph structure in our model, as shown in Figure 4, focusing on the top 20 companies by market capitalization and 10 hyperedges. To enhance visualization, values are scaled within the corresponding period,

		Model	Phase #1	Phase #2	Phase #3	Phase #4	Phase #5	Phase #6	Phase #7	Phase #8	Phase #9	Phase #10	Mean
IC	Transformer	DTML	0.003	0.013	0.018	0.025	-0.003	-0.010	0.012	0.028	0.010	-0.021	0.008
		MASTER	0.029	0.005	0.018	0.006	-0.005	0.038	0.038	0.010	-0.032	0.022	0.013
	Graph	RSR-E	0.003	0.013	0.008	0.022	0.012	0.040	0.005	0.031	0.004	-0.004	0.013
		RSR-I	0.025	0.014	**0.025**	0.020	0.025	**0.041**	0.031	0.004	0.008	**0.028**	0.022
		RT-GCN	0.030	0.021	0.016	-0.014	0.019	0.004	0.024	**0.054**	-0.005	-0.013	0.014
	Hypergraph	STHAN-SR	-0.003	-0.002	0.019	-0.001	0.009	0.011	0.015	0.021	0.023	0.022	0.011
		ESTIMATE	0.040	0.072	-0.030	**0.053**	-0.009	0.032	0.051	-0.002	-0.003	0.025	0.023
		Ours	**0.057**	**0.087**	0.001	0.037	**0.033**	0.035	**0.102**	-0.005	**0.052**	-0.028	**0.037**
RankIC	Transformer	DTML	0.014	0.022	0.028	0.034	-0.003	-0.021	0.020	**0.030**	0.021	-0.017	0.013
		MASTER	0.031	0.020	0.025	0.001	-0.003	**0.037**	0.036	0.007	-0.038	0.027	0.014
	Graph	RSR-E	0.016	0.016	0.024	-0.004	0.008	0.033	0.013	0.015	0.016	-0.01	0.013
		RSR-I	0.029	0.021	**0.038**	0.027	0.030	0.030	0.025	0.001	0.013	0.031	0.024
		RT-GCN	0.027	-0.005	0.001	0.005	0.033	-0.018	0.025	0.005	-0.039	0.025	0.006
	Hypergraph	STHAN-SR	0.003	-0.007	0.008	-0.009	0.026	0.016	0.018	0.027	0.029	-0.010	0.010
		ESTIMATE	0.044	**0.079**	-0.014	**0.045**	0.002	0.028	0.070	-0.015	0.003	**0.035**	0.028
		Ours	**0.045**	0.077	-0.001	0.041	**0.055**	0.030	**0.114**	-0.005	**0.060**	-0.027	**0.039**
Prec@K	Transformer	DTML	0.533	0.562	0.491	0.582	0.521	0.602	0.620	0.549	**0.569**	0.605	0.563
		MASTER	0.537	0.528	0.588	0.527	0.547	0.590	0.643	0.544	0.548	**0.610**	0.568
	Graph	RSR-E	**0.577**	0.564	0.494	0.480	0.574	0.569	0.588	0.571	0.559	0.554	0.553
		RSR-I	0.568	0.568	0.550	**0.604**	0.545	0.514	0.565	0.501	0.544	0.518	0.545
		RT-GCN	0.529	0.529	**0.592**	0.551	0.600	0.548	0.660	0.576	0.534	0.571	0.569
	Hypergraph	STHAN-SR	0.572	0.525	0.504	0.499	0.510	0.517	0.566	0.562	0.562	0.496	0.531
		ESTIMATE	0.565	0.537	0.543	0.595	0.630	0.589	0.685	0.587	0.548	0.484	0.576
		Ours	0.555	**0.582**	0.581	**0.604**	**0.643**	**0.614**	**0.720**	**0.616**	0.562	0.544	**0.602**

Table 1: Predictive performance results. Bold and underlines show the best and second best results, respectively.

Model	Return↑	SR↑	MDD↓
DTML	0.794	3.167	0.571
MASTER	0.726	2.695	0.594
RSR-E	0.573	2.633	0.537
RSR-I	0.771	3.006	0.639
RT-GCN	0.796	3.121	0.492
STHAN-SR	0.648	2.951	0.478
ESTIMATE	0.723	2.905	0.613
Ours	**0.798**	**3.391**	**0.474**

Table 2: Portfolio performance results. Bold and underlines show the best and second best results, respectively.

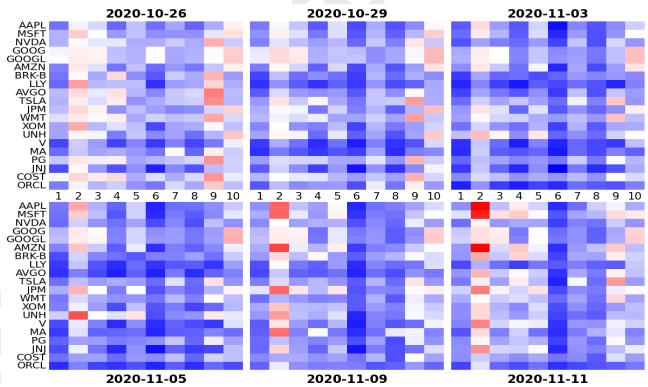

Figure 4: Visualization of submatrices of the dynamic hypergraph incidence matrix $\boldsymbol{A}$.

with blue indicating weaker connections and red stronger connections. Note that these colors represent relative edge strength, not full connectivity or disconnection.

We focus our analysis on AAPL and MSFT, both of which are global leaders driving market trends. In October 2020, both were relatively strongly connected to Hyperedge 2 and 10, but after the U.S. presidential election on November 3, 2020, their connections to other hyperedges weakened, while their connection to Hyperedge 2 strengthened significantly. Similar patterns were observed for other major companies such as AMZN, BRK-B, UNH, and JPM. These observations demonstrate the dynamic hypergraph's ability to effectively capture evolving inter-stock relationships across different time steps. Additionally, companies like GOOG-GOOGL and V-MA remain consistently connected to the same hyperedge across multiple time steps, showing that dynamic hypergraph captures temporal dynamics while still preserving domain knowledge by similar companies.

6 Conclusions

In this paper, we proposed a novel model for accurate stock price forecasting by capturing the dynamic and higher-order relationships. The model incorporates a Hawkes process layer to identify shared movement patterns and enhance temporal representations. Furthermore, by both predefined and dynamic hypergraphs, the model effectively captures higher-order relationships among stocks. Extensive experiments conducted on real-world US market data demonstrated the model's superiority over state-of-the-art approaches in terms of predictive performance and portfolio stability. Additionally, the effectiveness of the proposed components was validated through detailed analyses. In future works, we will focus on enhancing the robustness of the model, particularly during periods of unstable performance, to achieve consistent reliability across diverse market conditions.

Acknowledgments

This work was supported by the National Research Foundation of Korea grant [NRF-2022R1A4A3033874, NRF-2022R1F1A1074758, RS-2022-NR068754]. This work is also supported by a grant from the Korean Ministry of Environment through Environmental Health R&D Program [2021003310005]. The authors acknowledge the Urban Big data and AI Institute of the University of Seoul supercomputing resources (http://ubai.uos.ac.kr) made available for conducting the research reported in this paper.

References

[Ang *et al.*, 2006] Andrew Ang, Joseph Chen, and Yuhang Xing. Downside risk. *The review of financial studies*, 19(4):1191–1239, 2006.

[Bacry *et al.*, 2015] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

[Bai *et al.*, 2021] Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

[Bennett *et al.*, 2022] Stefanos Bennett, Mihai Cucuringu, and Gesine Reinert. Lead–lag detection and network clustering for multivariate time series with an application to the us equity market. *Machine Learning*, 111(12):4497–4538, 2022.

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

[Cox and Isham, 1980] David Roxbee Cox and Valerie Isham. *Point processes*, volume 12. CRC Press, 1980.

[Ding *et al.*, 2020] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. Hierarchical multi-scale gaussian transformer for stock movement prediction. In *IJCAI*, pages 4640–4646, 2020.

[Embrechts *et al.*, 2011] Paul Embrechts, Thomas Liniger, and Lu Lin. Multivariate hawkes processes: an application to financial data. *Journal of Applied Probability*, 48(A):367–378, 2011.

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

[Feng *et al.*, 2019b] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

[Hawkes, 1971] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[Hong *et al.*, 2024] Sungchul Hong, Yunjin Choi, and Jong-June Jeon. Interpretable water level forecaster with spatiotemporal causal attention mechanisms. *International Journal of Forecasting*, 2024.

[Huynh *et al.*, 2023] Thanh Trung Huynh, Minh Hieu Nguyen, Thanh Tam Nguyen, Phi Le Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 850–858, 2023.

[Joe, 1997] Harry Joe. *Multivariate models and multivariate dependence concepts*. CRC press, 1997.

[Ju *et al.*, 2024] Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Yifan Wang, Xiao Luo, and Ming Zhang. Hypergraph-enhanced dual semi-supervised graph classification. In *Proceedings of the 41st International Conference on Machine Learning*, pages 22594–22604. PMLR, 2024.

[King, 1966] Benjamin F King. Market and industry factors in stock price behavior. *the Journal of Business*, 39(1):139–190, 1966.

[Li *et al.*, 2021] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. Modeling the stock relation with graph network for overnight stock movement prediction. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pages 4541–4547, 2021.

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

[Lim *et al.*, 2021] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

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

[Scarselli *et al.*, 2008] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[Sibuya and others, 1960] Masaaki Sibuya et al. Bivariate extreme statistics. *Annals of the Institute of Statistical Mathematics*, 11(2):195–210, 1960.

[Sun *et al.*, 2021] Xiangguo Sun, Hongzhi Yin, Bo Liu, Hongxu Chen, Jiuxin Cao, Yingxia Shao, and Nguyen Quoc Viet Hung. Heterogeneous hypergraph embedding for graph classification. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 725–733, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Neural Information Processing Systems*, 2017.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

[Xia *et al.*, 2024] Hongjie Xia, Huijie Ao, Long Li, Yu Liu, Sen Liu, Guangnan Ye, and Hongfeng Chai. Ci-sthpan: Pre-trained attention network for stock selection with channel-independent spatio-temporal hypergraph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9187–9195, 2024.

[Xu *et al.*, 2021] Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. Hist: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. *arXiv preprint arXiv:2110.13716*, 2021.

[Yoo *et al.*, 2021] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2037–2045, 2021.

[Zhang *et al.*, 2020] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020.

[Zhao *et al.*, 2023] Yusheng Zhao, Xiao Luo, Wei Ju, Chong Chen, Xian-Sheng Hua, and Ming Zhang. Dynamic hypergraph structure learning for traffic flow forecasting. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2303–2316. IEEE, 2023.

[Zheng *et al.*, 2023] Zetao Zheng, Jie Shao, Jia Zhu, and Heng Tao Shen. Relational temporal graph convolutional networks for ranking-based stock prediction. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 123–136. IEEE, 2023.

[Zhou *et al.*, 2006] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *Advances in neural information processing systems*, 19, 2006.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[Zuo *et al.*, 2020] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.