DySTAGE: Dynamic Graph Representation Learning for Asset Pricing via Spatio-Temporal Attention and Graph Encodings

Jingyi Gu*
New Jersey Institute of Technology
US
jg95@njit.edu

Junyi Ye*
New Jersey Institute of Technology
US
jy394@njit.edu

Ajim Uddin
New Jersey Institute of Technology
US
ajim.uddin@njit.edu

Guiling Wang†
New Jersey Institute of Technology
US
gwang@njit.edu

Abstract

Current GNN-based asset price prediction models often focus on a fixed group of assets and their static relationships within the financial network. However, this approach overlooks the reality that the composition of asset pools and their interrelationships evolves over time, necessitating the development of a flexible framework capable of adapting to this dynamism. Accordingly, we propose DySTAGE, a framework with a universal formulation that transforms asset pricing time series into dynamic graphs, accommodating asset addition, deletion, and changes in correlations. Our framework includes a graph learning model specifically designed for this purpose. In our framework, assets at various historical time steps are structured as a sequence of dynamic graphs, where connections between assets reflect their long-term correlations. DySTAGE effectively captures both topological and temporal patterns. The *Topological Module* deploys *Asset Influence Attention* to learn global interrelationships among assets, further enhanced by *Asset-wise Importance Encoding*, *Pair-wise Spatial Encoding*, and *Edge-wise Correlation Encoding*. Meanwhile, the *Temporal Module* encapsulates node representations across the temporal dimension via the attention mechanism. We validate our approach through extensive experiments using three different real-world stock pricing data, demonstrating that DySTAGE surpasses popular benchmarks in return prediction, and offers profitable investment strategies. The code is publicly available under NJIT FinTech Lab's GitHub[1].

CCS Concepts

• **Applied computing → Economics**; • **Computing methodologies → Neural networks**; • **Mathematics of computing → Graph algorithms**.

*These authors contributed equally to this research.
†Corresponding author.
[1]https://github.com/NJIT-Fintech-Lab/DySTAGE

ICAIF '24, November 14–17, 2024, Brooklyn, NY, USA

ACM ISBN 979-8-4007-1081-0/24/11
https://doi.org/10.1145/3677052.3698680

Keywords

Graph Neural Networks, Dynamic Graph, Asset Pricing, Stock Price Prediction

ACM Reference Format:
Jingyi Gu, Junyi Ye, Ajim Uddin, and Guiling Wang. 2024. DySTAGE: Dynamic Graph Representation Learning for Asset Pricing via Spatio-Temporal Attention and Graph Encodings. In *5th ACM International Conference on AI in Finance (ICAIF '24), November 14–17, 2024, Brooklyn, NY, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3677052.3698680

1 Introduction

Asset pricing models are pivotal in estimating the future returns of underlying assets such as stocks, by employing a multitude of variables to model their interactions [37]. These models are essential for market participants who seek to discern potential high-performing assets within the market universe. The assets in this universe are interdependent through various dimensions such as supply chains, industry sectors, return similarities, volatility spillovers, and market conditions. These complex interrelationships significantly impact their relative prices [26, 31]. Given these intricacies, graph networks have become a preferred tool for modeling such relationships [8, 16, 36]. Financial markets are inherently dynamic; assets and their interconnections evolve continuously due to factors such as new market entries, asset maturation, and corporate events like bankruptcies, mergers, and acquisitions. This ongoing evolution necessitates a dynamic representation in our models.

Figure 1 visually depicts the evolution of the market in terms of asset participation and their interrelationships. Conceptualized as a graph, each asset is represented as a node, and correlations are depicted as edges. Nodes dynamically appear and disappear from the asset graph, reflecting the fluid and responsive nature of the market to real-world events. For instance, Lehman Brothers was removed in October 2008 following its bankruptcy, while Tesla was added in June 2010 with its IPO. Similarly, the connections between assets—edges—also evolve, reflecting shifts in corporate strategy and market conditions. A prime example is Apple, which, upon launching Apple TV+ in 2019, pivoted from focusing solely on hardware and software to entering the competitive streaming service market, positioning itself as a rival to Netflix.

Despite numerous studies on network dynamics, there remains a substantial gap in developing frameworks that can effectively learn and represent the time-varying dynamics of financial markets.

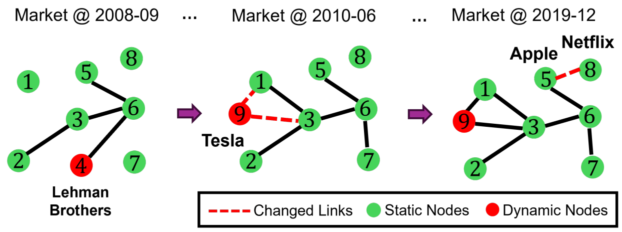

Figure 1: An example of dynamic graphs in asset pricing.

Recent advancements in graph-based asset pricing methodologies have leveraged static graph structures to model relationships and predict future asset prices [5, 30]. Meanwhile, dynamic graph representation learning, commonly used in domains such as traffic, social, and rating networks [1, 4, 24, 27], has been less frequently applied in financial networks. These financial networks are distinguished by their evolving interdependencies, which add complexity to their analysis.

To address this gap, we introduce a Dynamic graph representation learning model via Spatio-Temporal Attention and Graph Encodings (**DySTAGE**), an end-to-end dynamic graph learning framework designed to accurately reflect and predict market evolution. We propose a universal formulation for dynamic graphs tailored to asset pricing, constructing a sequence of dynamic graphs from asset time series data that accommodate node (asset) additions and removals. Additionally, asset correlation over different economically meaningful time scales are incorporated and reflected as multiple edge features between the same pair of nodes. Subsequently, we design a graph learning model that employs attention mechanisms and graph encodings enriched with financial insights to predict the future excess returns of existing nodes, capturing both topological and temporal patterns. This framework enhances predictive accuracy and supports informed investment strategies in a rapidly changing market.

The framework features a *Topological Module* and a *Temporal Module*. The *Topological Module* individually processes structural information for each graph and comprises several components: *Asset Influence Attention*, which delineates the global interrelationships between assets; *Asset-wise Importance Encoding*, emphasizing the broad market impact of individual assets; *Pair-wise Spatial Encoding*, exploring intermediary assets and uncovering hidden connections; and *Edge-wise Correlation Encoding*, revealing the evolution of relationships between highly correlated assets. Following this, the *Temporal Module* delves into the historical representations of each asset across time, offering a detailed temporal analysis.

We conducted extensive experiments on three real-world datasets: Russell-3000 monthly data, MLFI monthly data, and S&P 500 daily data. The results demonstrate DySTAGE's superior performance in return prediction and portfolio optimization. Our contributions can be summarized as follows:

(1) Introduction of DySTAGE, a novel dynamic graph representation learning framework for asset pricing, with formulation accommodating dynamic asset composition and evolving edge connections, and model designed with financial insights.
(2) Effective capture of both topological and temporal patterns, utilizing graph encodings that reflect asset importance, hidden connections, and their evolving impact within the financial network.
(3) Proven superiority of DySTAGE over conventional and popular benchmarks in predictive accuracy, illustrating its efficacy in balancing profit and risk for portfolio management.

The remainder of the paper is organized as follows: Section 2 reviews current research in asset pricing, Section 3 formulates the problem addressed by our study, Section 4 details of DySTAGE framework, Section 5 showcases the numerical results based on our research questions. Finally, Section 6 provides the conclusion.

2 Related Works

Machine learning in asset pricing has evolved with advancements in time series prediction, Graph Neural Networks for asset correlations, and dynamic graphs for temporal market dynamics.

2.1 Time Series Model

Asset pricing has evolved from traditional models like ARIMA [22] to advanced deep learning techniques [25] such as RNNs, LSTMs [13, 28], and MLP-based models [38] like N-BEATS [23]. These models, with complex structures like residual connections and Transformer-like blocks, enhance forecasting accuracy by recognizing complex temporal patterns. Multi-scale analysis techniques, including Fourier transforms, wavelets, and downsampling, enrich temporal modeling by examining stock behavior across various time scales [10, 19]. Deep neural networks [11, 18] and Transformer-based approaches [9, 12] with Multi-Scale Gaussian Priors model financial trends from intraday fluctuations to long-term shifts. However, these models often overlook interactions and interdependencies among financial time series, especially in stock price movements.

2.2 Graph Models

Graph Neural Networks (GNNs) facilitate the mapping of spatial connections among financial assets. Static graphs, with invariant structures and variable node attributes, serve as the foundation for this analysis. GNN approaches like GCN [17], GraphSAGE [14], GAT [33], and UniMP [29] implement distinct aggregation strategies to interpret complex graph structures. Studies have explored various relational frameworks through shareholding ratios, ownership structures, sectoral similarities, and stock correlations [6, 16, 30, 39]. While, they fall short in capturing dynamic market dynamism, where firms constantly emerge and dissolve. Dynamic graphs track evolving system dynamics by accommodating changes in graph structure and node characteristics. Most works, such as T-GCN [42], GCLSTM [4], and DY-GAP [30], maintain a fixed node count despite dynamic linkages and node changes. Conversely, studies like DySAT [27] and EvolveGCN [24] address scenarios with variable node counts, utilizing attention mechanisms and GCN regularization to manage node embeddings in these complex environments. Many of these models are designed for different domains and require significant adaptation for financial time series. Additionally, most dynamic graph analyses, such as DyTed [41] and DGIB [40], primarily target link prediction and struggle to extend

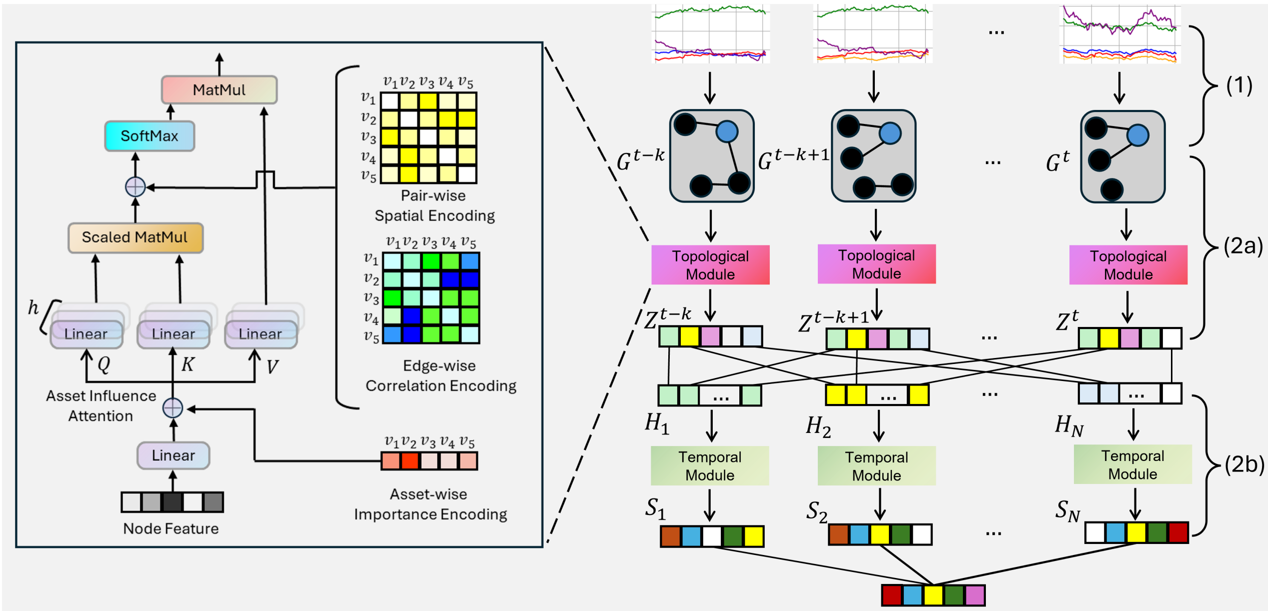

Figure 2: The framework of DySTAGE. (1) Dynamic graph construction from time series (top right). (2) Dynamic graph learning model: (a) Topological Module individually processing structural information for each graph (left), (b) Temporal Module capturing historical representations across time (bottom right).

to regression tasks. Complex asset pricing models require handling large spatio-temporal datasets with thousands of nodes and time steps, underscoring the need for innovative approaches tailored to financial markets dynamics.

3 Problem Formulation

The core challenge in asset pricing is to estimate the expected future value of an asset based on the information currently available. Research has consistently shown that the expected future value of an asset is influenced by a combination of factors including asset-specific characteristics, historical performance trends, real-time news, and broader market dynamics [15, 30]. This study aims to capture both the temporal dependencies and the spatial dynamics of the market using a dynamic spatio-temporal graph model.

Our objective is to develop a model $f(\cdot)$ that can accurately predict future return for assets $\mathbf{y}^{t+1}$ based on a sequence of historical financial interconnection networks $\mathbb{G} = \{\mathcal{G}^{t-k}, ..., \mathcal{G}^{t}\}$ and corresponding asset characteristics $\mathbf{X} = \{\mathbf{X}^{t-k}, ..., \mathbf{X}^{t}\}$:

$$\mathbf{y}^{t+1} = f(\mathcal{G}^{t-k}, \cdot\cdot\cdot, \mathcal{G}^{t}, \mathbf{X}^{t-k}, \cdot\cdot\cdot, \mathbf{X}^{t-k}, ; \theta) \tag{1}$$

where k represents the window of historical data considered. This formulation enables the model to utilize temporal data from past periods and integrate it with spatial information from the asset interconnections, capturing the dynamic interplay of market variables that affect asset pricing.

4 Methodology

Figure 2 shows the architecture of DySTAGE. It starts with a dynamic graph construction phase. Then it contains two key modules: the topological module learns the structural information of each graph independently; the temporal module generates node embeddings for each node by capturing its historical patterns along the temporal dimension.

4.1 Dynamic Graph Construction

We consider the historical dynamic graphs $\mathbb{G} = \{\mathcal{G}^{t-k}, ..., \mathcal{G}^{t}\}$ as a sequence, where $\mathcal{G}^t = (\mathcal{V}^t, \mathcal{E}^t)$ represents an undirected graph at time step t, with $n^t (= |\mathcal{V}^t|)$ nodes (i.e., total number of assets at time t) and $m^t (= |\mathcal{E}^t|)$ edges. Let $\mathbf{X}^t \in \mathbb{R}^{n^t \times d}$ denote node features, where n^t is the number of assets and d is the feature dimension. $\mathcal{A}^t \in \mathbb{R}^{n^t \times n^t}$ is the adjacency matrix of graph $\mathcal{G}^t$, representing the long-term interrelationship between assets. In this work, we utilize Pearson Correlation on the historical excess returns to quantify the graph connections:

$$\mathcal{A}^t_{u,v} = \begin{cases} \rho^t_{u,v} & |\rho^t_{u,v}| > \gamma \\ 0 & |\rho^t_{u,v}| \leq \gamma \end{cases} \tag{2}$$

For any pair of assets u and v, if their absolute correlation $\rho^t_{u,v}$ based on returns over the past w time steps is larger than the threshold $\gamma \in (0, 1)$, they are considered to have a strong long-term correlation and should be connected by an edge. Both positive and negative correlations between assets are incorporated in a single graph. $\mathcal{E}^t \in \mathbb{R}^{m^t \times p}$ denotes edge attributes containing multi-scale excess return correlations between assets, calculated from short-term to long-term perspectives, where p is the number of scales. For monthly asset data, we choose scales of 3, 6, 12, 24, and 36 to describe quarterly, semiannually, and yearly trends. For daily data, we choose scales of 5, 10, 15, and 20 for weekly patterns. The target of our problem is the excess return $\mathbf{y}^{t+1} \in \mathbb{R}^{n^{t+1}}$ of all existing assets in the future time step $t+1$. Our primary objective is to train a network that learns historical dynamic graphs $\mathbb{G}$ and predicts the future excess return $\mathbf{y}^{t+1}$.

To streamline implementation, we unify the node set across all time steps into a shared set $\mathcal{V}$, encompassing all appearing nodes, irrespective of their addition or removal over time. Therefore, the graphs, including their adjacency matrices $\mathcal{A}^t \in \mathbb{R}^n$ and node features $\mathbf{X}^t \in \mathbb{R}^{n\times d}$ at all time step, contain the same nodes. If node u does not exist at time step t, then there does not exist an edge between itself and any other nodes, and its feature $\mathbf{x}_u^t \in \mathbb{R}^d = \mathbf{0}$ is also set to a zero vector.

4.2 Topological Module

To predict future asset returns, each historical graph $\mathcal{G}^{t-k}, \ldots, \mathcal{G}^t$ is individually fed into the Topological Module. $\mathcal{G}^t$ represents the asset interconnection network for that time point. Asset Influence attention is utilized to describe interrelationship and their influences between assets based on their node features globally. To fully investigate the structural information, we incorporate three graph encodings derived from dynamic asset graphs from financial perspectives: Asset-wise Importance Encoding highlights the importance of each asset and its potential impact on the market; Pair-wise Spatial Encoding investigates the hidden connections between assets; Edge-wise Correlation Encoding provides evolving correlation between assets. The output of this module is a set of node embeddings in the graph. The details of each of these encodings are discussed below. Note that the index of time step t is omitted in the remaining discussion in this module for simplicity.

4.2.1 Asset Influence Attention. Given the asset features $\mathbf{X}$ in the input graph $\mathcal{G}$, we follow the standard multi-head attention mechanism [32] to design an asset influence attention and derive the attention matrix. Within each head, the input feature $\mathbf{X}$ is firstly mapped into $\mathbf{X}^* \in \mathbb{R}^{d\times d^*}$, where superscript $\cdot^*$ indicates the topological module, and d^* denote the dimension in this module. Then it is projected by three learnable weight matrices $\mathbf{W}_q^*, \mathbf{W}_k^*, \mathbf{W}_V^* \in \mathbb{R}^{d^*\times d_k^*}$, where d_k^* denotes the hidden dimension of each head. Then the attention matrix $\mathbf{A}_h \in \mathbb{R}^{n\times n}$ is computed, where $h \in \{1, ..., H\}$ and H represents the number of heads, with each attention element $A^*_{u,v,h}$ describing the influence of asset u on asset v:

$$\mathbf{A}_h^* = \mathbf{M_0}^* \odot softmax(\frac{(\mathbf{X}^*\mathbf{W}_q^*)(\mathbf{X}^*\mathbf{W}_k^*)^\top}{\sqrt{d_k^*}} + \mathbf{M}_\infty{}^*) \tag{3}$$

$$\mathbf{Z} = \mathbf{X}^* + [\mathbf{A}_1^*(\mathbf{X}^*\mathbf{W}_v^*), ..., \mathbf{A}_H^*(\mathbf{X}^*\mathbf{W}_v^*)]\mathbf{W}_o^* \tag{4}$$

For non-existing asset u, it should not exert influence and relationship with other assets. To exclude the current non-existing assets, we incorporate two mask matrices before and after the softmax activation function. Softmax activation operates on the row sum of attention matrix. For negative mask matrix $\mathbf{M}_\infty{}^* \in \mathbb{R}^{n\times n}$ with $M_{iu}^* = -\infty, i \in \mathcal{V}$, it ensures the column vector of asset u is masked out, yielding corresponding softmax results is 0. The zero mask matrix $\mathbf{M_0}^* \in \mathbb{R}^{n\times n}$ with $M_{uj}^* = 0, j \in \mathcal{V}$ guarantees the row vector of asset u is masked out as a zero vector, $\odot$ denotes the element-wise multiplication.

Utilizing multiple heads facilitates the aggregation of diverse embedding spaces. The embeddings from multiple attention heads are concatenated together and mapped by $W_o \in \mathbb{R}^{Hd^*\times d^*}$ to yield the output embeddings $\mathbf{Z} \in \mathbb{R}^{n\times d^*}$. Following the common practice, we apply layer normalization [35] before attention and incorporate skip connections after concatenating the heads, aiming at enhancing optimization efficiency [21, 34]. The Asset Influence Attention can be treated as one layer and extended to multiple layers.

4.2.2 Asset-wise Importance Encoding. Unlike the position encoding in the original transformer, which signifies the order of sequence, the nodes within the graph lack such inherent sequence. However, it is crucial to identify the importance of each asset and its potential impact within its sector or the broader financial market. Such insights can be learned from the topological structure of the graph. Therefore, we introduce an Importance Encoding to capture asset-wise structural information. Specifically, for each asset u, a higher node degree implies that the asset has a strong correlation with a larger number of other assets, indicating its potential market impact. Let $\mathbf{D}_u$ denote the degree encoding of asset u, which is a learnable embedding vector determined by its degree, and $\mathbf{D} \in \mathbb{R}^{n\times d}$ denote the degree encoding of all assets in the graph. This encoding serves similarly as position encodings and is added to node features before feeding into the attention mechanism. Therefore, the $\mathbf{X}^*$ in Eq.3 and Eq.4 can be replaced by $\tilde{\mathbf{X}}^* = \mathbf{X}^* + \mathbf{D}$.

4.2.3 Pair-wise Spatial Encoding. In addition to the attention mechanism that comprehensively considers every pair of assets globally, we complement the local information by extracting pair-wise structure insights from the graph, specifically targeting pairs of assets that are connected by a path. When two assets are not directly connected by an edge but a path exists between them, this indirect connection implies that although the assets do not directly affect each other, they are linked through intermediary assets, shared market factors, or underlying economic dynamics. Analyzing these paths between assets allows us to uncover hidden connections and better understand the underlying structure of the financial market network. In this study, we utilize the shortest path distance between two assets as a key metric. This distance represents the most direct route between assets, considering how far the influence can be transmitted by the intermediary assets between them. Let $\mathbf{S} \in \mathbb{R}^{n\times n}$ denote the Pair-wise Spatial Encoding of the graph, where $S_{u,v}$ is a learnable scalar based on the shortest path distance between asset u and asset v. If no path exists between them, the shortest path distance will be represented by a large value, representing they are not reachable from each other. This encoding is then incorporated as a piece of complementary information to the dot product in the attention; therefore, the attention matrix can be modified:

$$\mathbf{A}_h^* = \mathbf{M_0}^* \odot softmax(\frac{(\tilde{\mathbf{X}}^*\mathbf{W}_q^*)(\tilde{\mathbf{X}}^*\mathbf{W}_k^*)^\top}{\sqrt{d_k^*}} + \mathbf{S} + \mathbf{M}_\infty{}^*) \tag{5}$$

4.2.4 Edge-wise Correlation Encoding. Recall that the edge attributes encapsulate multi-scale excess return correlation between assets. For two assets exhibiting strong long-term correlations, leveraging their edge attributes becomes imperative to reveal how their relationship evolves over time. It provides insights into whether the correlation remains consistent or fluctuates across various time frames, spanning from short-term to long-term durations.

$$E_{u,v} = \frac{1}{p}\sum_{i}^{p}(\mathbf{c}_{u,v}\mathbf{w}_e^\top)_i, \quad \mathbf{c}_{u,v} = \begin{cases} \mathcal{E}_{u,v} & if \mathcal{A}_{u,v} \neq 0 \\ \mathbf{0} & if \mathcal{A}_{u,v} = 0 \end{cases} \tag{6}$$

$$\mathbf{A}_h^* = \mathbf{M_0}^* \odot softmax(\frac{(\tilde{\mathbf{X}}^*\mathbf{W}_q^*)(\tilde{\mathbf{X}}^*\mathbf{W}_k^*)^\top}{\sqrt{d_k^*}} + \mathbf{S} + \mathbf{E} + \mathbf{M}_\infty{}^*) \tag{7}$$

For a given pair of assets u and v, we first compute $\mathbf{c}_{u,v} \in \mathbf{R}^p$. If an edge exists between them, then $\mathbf{c}_{u,v}$ corresponds to their edge attributes $\mathcal{E}_{u,v}$; otherwise, it defaults to a zero vector. Subsequently, $\mathbf{c}_{u,v}$ is transformed by a learnable weight matrix $w_e \in \mathbb{R}^{p\times p}$ and then averaged across the scale dimension. Similarly to the Pair-wise Spatial Encoding, the Edge-wise Correlation Encoding $\mathbf{E}$ can be seamlessly integrated into the attention matrix, complementing the model's understanding of asset interactions within the topological framework.

4.3 Temporal Module

Given the sequence of graph inputs $\mathbb{G} = (\mathcal{G}^{t-k}, ..., \mathcal{G}^t)$, upon passing through the Topological Module, we obtain a sequence of outputs $(\mathbf{Z}^{t-k}, ..., \mathbf{Z}^t)$. Each output embedding $\mathbf{Z}^t$ comprises a set of node embeddings $\mathbf{z}_u^t$ corresponding to each node $u \in \mathcal{V}$, Thus, for every node u, we obtain a sequence of historical node embeddings $\mathbf{H}_u = (\mathbf{z}_u^{t-k}, ..., \mathbf{z}_u^t) \in \mathbb{R}^{k\times d^*}$, encapsulating both global and local structure information. The Temporal Module aims to investigate patterns from the historical embeddings along the temporal dimension for each node individually. The node index u is omitted in the remaining discussion within this module for simplicity.

Similar to the Asset Influence Attention, we adopt the multi-head attention mechanism to capture temporal dynamics. The input $\mathbf{H}$ is firstly mapped by a weight matrices $\mathbf{W}^\star \in \mathbb{R}^{d^*\times d^\star}$, where superscript $\cdot^\star$ indicates temporal module, and $d^\star$ denotes the hidden dimension in this module. Leveraging the time step due to the sequential nature of inputs, the position encoding is defined as $\mathbf{P} \in \mathbb{R}^{k\times d^\star}$ where each $\mathbf{P}_t \in \mathbb{R}^{d^\star}$ is a learnable embedding vector determined by its time index. It is added to $\mathbf{H}$, yielding $\tilde{\mathbf{H}} = \mathbf{H}\mathbf{W}^\star + \mathbf{P}$ which is then utilized to generate temporal attention matrix $\mathbf{A}_{h'}^\star$, where $h' \in \{1, ..., H'\}$ and H' represents the number of heads. Here, each element $A_{i,j,h'}^\star$ captures the interrelationship of node embeddings between time step i and j:

$$\mathbf{A}_{h'}^\star = softmax(\frac{(\tilde{\mathbf{H}}\mathbf{W}_q^\star)(\tilde{\mathbf{H}}\mathbf{W}_k^\star)^\top}{\sqrt{d_k^\star}} + \mathbf{M}_\infty{}^\star) \tag{8}$$

$$\mathbf{S} = \mathbf{H} + [\mathbf{A}_1^\star(\tilde{\mathbf{H}}\mathbf{W}_v^\star), ..., \mathbf{A}_H^\star(\mathbf{H}\mathbf{W}_v^\star)]\mathbf{W}_o^\star \tag{9}$$

Recognizing that embeddings from future time steps cannot be foreseen in the past, we address this by applying a mask matrix $\mathbf{M}_\infty{}^\star$ to mitigate such issues. For each element $M_{ij}^\star$, if time step i is ahead of j, then it is set to negative infinite to guarantee the softmax result is 0; otherwise it is set to 0. Additionally, We employ multiple heads in temporal attention, concatenating their outputs to obtain $\mathbf{S} \in \mathbb{R}^{k\times d^*}$, the final output embeddings from Temporal Module.

4.4 Node-level Prediction Task

Our task is to forecast the excess return for each node in the subsequent future time step. For each asset u, we derive the embeddings $\mathbf{S}_u = (\mathbf{S}_u^{t-k}, ..., \mathbf{S}_u^t)$ after undergoing temporal attention. To obtain the node-level prediction, we extract the embedding vector $\mathbf{S}_u^t \in \mathbb{R}^{d^*}$ at the final time step t which wraps all structural information from related assets and historical patterns. A multi-layer perceptron (MLP) layer and tanh activation function are then applied to generate $\tilde{y}_u$ in the range of (-1,1), representing the predicted excess return for asset u on the time step $t+1$.

To exclude non-existing nodes from consideration, we incorporate a mask $M^{(P)}$ into both target and predictions for all shared assets. This mask is assigned a value of 1 if asset u exists for both time steps $t+1$ and t. This criterion ensures that for predicting future returns, the asset must exist in the future time step and at least one historical data point immediately preceding it available; otherwise, the mask is set to 0. For optimization, we adopt Mean Square Error Loss as our objective function.

$$\hat{y}_u = M^{(P)} \cdot \tilde{y}_u, \quad \tilde{y}_u = tanh(MLP(\mathbf{S_u})) \tag{10}$$

$$\mathcal{L} = \sum_{u\in\mathcal{V}} (\hat{y}_u - M^{(P)} \cdot y_u)^2 \tag{11}$$

5 Experiment

To comprehensively evaluate our proposed model and the effectiveness of node representations learned from dynamic graphs, we raise four research questions and conduct experiments from the following perspectives.

- **RQ1**: Does DySTAGE effectively learn dynamic graphs compared with other available *dynamic* and *static* approaches regarding asset return prediction?
- **RQ2**: Can the model *generate positive portfolio return* in real-world scenarios?
- **RQ3**: Does *graph representation* learned from DySTAGE effectively reflect asset patterns?
- **RQ4**: What is the contribution of each component in DySTAGE?

5.1 Dataset

In this study, we utilize three dynamic asset graph datasets, each sampled at different frequencies — monthly and daily — to assess our model's performance with dynamic graphs.

(1) **Russell 3000 Monthly Dataset** [3] includes monthly data for all Russell 3000 index constituents. The data provides security monthly returns for all index constituents, with 166 firm-specific characteristics that previously identified as proven predictors for future return. This includes variables related to firm size, profitability, accruals, momentum, earnings surprises, intangibles, and trading frictions.
(2) **MLFI Monthly Dataset** is publicly available from Machine Learning for Factor Investing (MLFI) [7], including 1,207 U.S.-listed stocks, with 990 stocks retained after data cleaning. It

Table 1: Summary of Statistics for Dynamic Asset Graph Datasets.

Dataset	# Snapshots	# Nodes	# Edges	# Features	Lag	Horizon	Frequency	Range
Russell 3000	193	2,151	1,290K	166	12	1	Monthly	Jan 2000 - Dec 2021
MLFI	172	990	387K	93	12	1	Monthly	Jan 2000 - Mar 2019
S&P 500	2,396	460	123K	24	20	1	Daily	Jan 3, 2011 - Dec 31, 2020

Table 2: Comparison results from benchmarks and our model. MAPE results are in the form of percentages (%).

Type	Model	Russell 3000			MLFI			S&P 500		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Time Series	ARIMA	0.1798	0.1422	117.8364	0.1254	0.0970	87.0359	0.0223	0.0178	17.8370
	N-Beats	0.1327	0.1050	83.8827	0.1005	0.0793	69.7604	0.0195	0.0158	15.8619
Static GNN	GAT	0.1073	0.0874	66.3047	0.0802	0.0651	55.6068	0.0154	0.0131	13.1286
	GraphSAGE	0.1060	0.0864	65.4026	0.0811	0.0656	56.0802	0.0156	0.0131	13.1464
	ARMAConv	0.1081	0.0877	66.2636	0.0808	0.0653	55.8012	0.0158	0.0133	13.2557
	UniMP	0.1078	0.0853	64.1338	0.1078	0.0853	64.1331	0.0156	0.0134	13.3536
Dynamic GNN	DySAT	0.1039	0.0840	63.2542	0.0806	0.0652	55.7200	0.0155	0.0132	13.2118
	DY-GAP	0.1357	0.1089	87.7335	0.0806	0.0652	55.7378	0.0155	0.0131	13.0723
	T-GCN	0.1078	0.0882	67.0031	0.0813	0.0657	56.0604	0.0168	0.0145	14.5104
	EvolveGCN	0.1064	0.0845	63.5845	0.0806	0.0651	55.5625	0.0155	0.0132	13.1989
	GCLSTM	0.1033	0.0839	62.9582	0.0807	0.0650	55.5467	0.0156	0.0134	13.4025
	DyTed	0.1040	0.0844	63.3368	0.0800	0.0647	55.3379	0.0155	0.0132	13.1666
	DGIB	0.1031	0.0837	62.8114	0.0802	0.0649	55.4483	0.0154	0.0131	13.0751
	DySTAGE	**0.1026**	**0.0833**	**62.5027**	**0.0797**	**0.0644**	**54.9632**	**0.0154**	**0.0131**	**13.0602**

has 93 features on crucial company fundamentals such as financial ratios, cash flow metrics, and performance indicators. It also incorporates liquidity data and market capitalization figures.

(3) **S&P 500 Daily Dataset** covers daily market performance of 460 stocks from 2011 to 2020, providing detailed data on price movements and trading volumes. It enables the examination of short-term market dynamics and investor behaviors by tracking daily trends, patterns, and potential trading signals through fluctuations in stock prices, volumes, and volatility indicators.

Table 3: Descriptive statistics for selected asset pricing features, exret_rf denotes the excess return which is the target.

Feature	Mean	STD	Min	Max
exret_rf	0.009	0.163	-0.995	24.00
VolSD	-4.548	35.52	-6121	-0.000
VolumeTrend	-0.006	0.020	-0.065	0.056
Accruals	-0.009	0.108	-6.265	5.750
Beta	0.974	0.716	-7.747	52.64
AbnormalAccruals	-0.002	0.135	-2.218	6.909
zerotrade	1.229	3.008	0.000	19.85
Activism1	14.88	49.00	6.000	23.00
VolMkt	-0.155	1.792	-1330	0.000
VarCF	-0.887	55.77	-20078	0.000

The dynamic evolution of assets varies across all datasets: over 30% of asset change in Russell 3000, and over 20% in MLFI. While assets in S&P remain constant, the graphs still retain dynamism due to changing edge correlations. For all datasets, we use monthly (daily) excess returns after the risk-free rate as the target. The stock returns data are sourced from CRSP, while the fundamental variables come from Compustat. Each dataset offers unique variables suitable for the intended frequency and type of analysis. Detailed data descriptions are provided in Table 1. and selected feature statistics in Table 3. Datasets are split along the temporal dimension into training, validation, and test sets, in a 70%:15%:15% ratio, with testing periods of 2020/01 - 2021/12, 2017/01 - 2019/03, and 2019/08 - 2020/12, respectively. Additionally, features are normalized to the range of $(-1, 1)$.

5.2 Baselines

We compare DySTAGE against a variety of traditional and state-of-the-art time series and GNN models. Our return prediction benchmarks are categorized into three main types:

(1) **Time series models**: ARIMA [20] is a traditional financial forecasting model, and N-Beats [23] is a neural basis expansion model for interpretable time series forecasting.
(2) **Static GNN approaches**: Among static GNN models, we employ GAT [33], GraphSAGE [14], ARMAConv [2], and UniMP [29]. These models differ in their approaches to aggregating network information. For instance, GAT and GraphSAGE aggregate neighboring embeddings, ARMAConv leverages recurrent neural networks (RNNs), while UniMP utilizes a self-attention mechanism. These models handle only static graphs and cannot deal with temporal patterns. Hence, we

utilize the latest historical graph as input with all historical features as node features.

(3) **Dynamic GNN models**: Our selection includes RNN-based models such as T-GCN [42], EvolveGCN [24], and GCLSTM [4], as well as attention-based models DY-GAP [30] for asset pricing and DySAT [27] utilizing self-attention to learn node representations. We also include two state-of-the-art models, DyTed [41] in contrastive learning framework, and DGIB [40] for robust representations.

5.3 Implementation Details

DySTAGE is implemented in PyTorch. Hyperparameters are tuned on the validation set. The window size for Pearson correlation w is 36 and 61, representing 3 years and 3 months, and the historical time step T is set to 12 and 20 for monthly and daily datasets respectively. The threshold of correlation γ for graph construction and adjacency matrix is 0.3. We employ the AdamW optimizer for training, running for up to 300 epochs with early stopping after 30 epochs. The learning rate is 0.0001. Both modules use 16 heads and a concatenated hidden dimension of 128. For other baselines, we fine-tune the hyperparameters based on optimal values reported in their studies and choose the best-performing model for implementation.

5.4 Comparison of Baselines (RQ1)

We implement DySTAGE and benchmarks on all monthly and daily datasets. As per common practice, model performance is evaluated using three metrics: Root Mean Absolute Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). Table 2 presents the numerical results, and we draw the following observations. (1) DySTAGE stands out as the most reliable and accurate model across diverse datasets, consistently surpassing time series models, static GNNs, and dynamic benchmarks across all evaluation metrics, especially regarding MAPE. It achieves impressive MAPE of 62.50% and 54.96% on Russell and MLFI, and 13.06% on S&P daily assets. (2) Time series models fall short in performance as they rely solely on a historical series of excess returns, disregarding other fundamental features and failing to capture the intricate relationship between assets in the financial networks. (3) Among static GNNs, UniMP and GAT demonstrate superior performance in terms of MAPE. Nevertheless, static GNNs generally falls behind dynamic counterparts across multiple datasets, primarily due to their lack of temporal dependencies and exclusive focus on fixed topological information. (4) In dynamic GNNs, other than DySTAGE, DyTed and DGIB emerge as the most powerful models for two monthly datasets, while DY-GAP achieves the lowest errors on S&P daily data. This suggests the superiority of dynamic models in capturing inherent variations in stock market data. DySTAGE outperforms other dynamic GNNS due to its enhanced structural attention and graph encodings. (5) Notably, all models exhibit better performance on daily S&P data compared with two monthly datasets. This can be attributed to the lower volatility and more predictable patterns observed in daily prices, facilitating more accurate predictions due to smoother and more consistent trends. Therefore, the difference in performance among models on S&P data is smaller compared to monthly data, yet DySTAGE maintains its superior performance in terms of MAPE.

5.5 Portfolio Management (RQ2)

To demonstrate how DySTAGE provides insights for investors and generate profits, we utilize our prediction results to construct portfolios from the dynamic asset pool and observe the returns. We conduct portfolio management on GNNs only due to their comparable performance. At the beginning of each period, we invest long positions in the top 10% assets with highest predicted excess returns and assign them equal weight, then liquidate them at the end of the period. The portfolio performance is evaluated by the following metrics:

(1) **Cumulative Return (CR)** is the total return achieved over testing period.
(2) **Annual Return (AR)** is the percentage change of equity over 1-year period.
(3) **Sharpe Ratio (SR)** is a risk-adjusted metric to calculate the excess return relative to its risk. A higher value indicates a more favorable strategy.

The numerical results are shown in the table 4. (1) DySTAGE consistently generates the highest profits, achieving cumulative returns of 50% on Russell, 10% on MLFI, and 31% on S&P. This demonstrates DySTAGE's ability to offer lucrative investment recommendations in real-world scenarios. (2) DySTAGE achieves a Sharpe Ratio exceeding 1 on S&P. Although GraphSAGE exhibits the highest Sharpe ratios on Russell, DySTAGE maintains a competitive Sharpe ratio close to 1.2, indicating a strong balanced between profitability and risk management. (3) N-Beats is competitive in portfolio management but weak in prediction performance. This may be because it learns parameters for each asset series individually, which leads to less accurate predictions, but top-return assets can be identified and fall into the leading 10% group. (4) Portfolio performance on MLFI is generally poorer due to higher volatility compared to the other datasets. DY-GAP and GAT show robustness during volatility periods but carry higher associated risk compared with DySTAGE. Conversely, GraphSAGE and DySAT perform well during the recovery period marked by an upward trend but struggle more with volatility. It is worth noting that DySTAGE showcases stability across all test periods.

5.6 Graph Learning (RQ3)

Figure 3 illustrates DySTAGE's capability to accurately capture and represent temporal dynamics within financial markets. The top panel, which depicts the absolute return differences between two selected stocks over ten months, establishes a baseline for financial performance fluctuations between these assets. The bottom panel, a t-SNE visualization of the DySTAGE's node embeddings, shows the relative positioning of these two stocks within the market's broader context at critical timestamps. Notably, the distance between the two stocks in this embedding space is consistent with observed absolute return differences, suggesting that the spatial distribution in model's embeddings effectively mirrors the actual financial performance disparities. This consistency supports the conclusion that DySTAGE exhibits robust graph representational learning capabilities. By effectively mapping the temporal changes and maintaining fidelity to the ground truth, the model demonstrates its potential utility for predictive analytics in finance. The accuracy of these embeddings in reflecting true market behavior

Table 4: Portfolio management results on the three datasets. CR and AR are in the format of percentage (%).

Type	Model	Russell 3000			MLFI			S&P 500		
		CR(%) ↑	AR(%) ↑	SR ↑	CR(%) ↑	AR(%) ↑	SR ↑	CR(%) ↑	AR(%) ↑	SR ↑
Time Series	ARIMA	42.1388	20.1368	1.0047	0.9074	0.5435	0.1198	17.6767	12.1484	0.7275
	N-Beats	49.5191	23.3519	1.1667	9.2134	5.0084	0.4069	25.9006	18.0936	1.0282
Static GNN	GAT	42.7684	20.4142	1.1837	8.0401	4.7492	0.3549	10.5096	7.4825	0.5219
	GraphSAGE	49.8937	23.5131	**1.2077**	-0.7568	-0.4547	0.0483	24.5304	17.1641	0.9445
	ARMAConv	25.7004	12.6751	0.7184	-0.5407	-0.3247	0.0675	14.1315	10.0146	0.6152
	UniMP	40.0290	19.2031	1.0231	5.8182	3.4514	0.2748	15.6653	1.0802	0.7083
Dynamic GNN	DySAT	37.9606	18.2812	1.0156	-0.3293	-0.1977	0.0808	23.3353	16.3512	0.9530
	DY-GAP	40.8272	19.5572	1.1609	9.8075	5.7740	0.4285	19.0134	13.3927	0.8330
	T-GCN	35.7086	17.2698	0.9543	7.6595	4.5486	0.3626	6.0337	4.3211	0.3410
	EvolveGCN	29.7708	14.5642	0.8721	2.2540	1.3464	0.1611	5.1677	3.7051	0.2837
	GCLSTM	41.9357	20.4726	1.0616	-3.2744	-1.9777	-0.0330	8.0919	5.7793	0.4216
	DyTed	40.1514	19.2575	0.9667	-0.7692	-0.4622	0.0643	9.4853	6.5678	0.5208
	DGIB	29.2723	14.3344	0.9369	6.1102	3.6226	0.3047	8.5284	6.0876	0.4149
	DySTAGE	**50.3428**	**23.7152**	1.1975	**10.2829**	**6.0486**	**0.4614**	**31.5506**	**21.8969**	**1.2945**

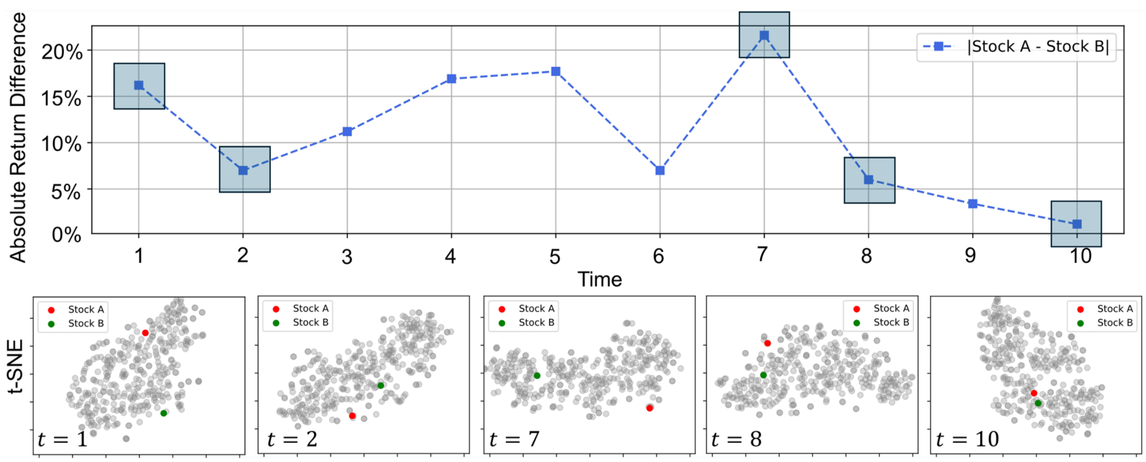

Figure 3: Similarity comparison between node embeddings from DySTAGE and ground truth.

Table 5: Ablation study results in terms of MAPE.

Model	Russell	MLFI	S&P
w/o Importance	62.7537	55.3018	13.0674
w/o Temporal	62.6115	55.0411	13.1801
w/o Spatial	62.5868	54.8906	13.0734
w/o Edge	62.5943	54.9654	13.0690
DySTAGE	**62.5027**	**54.9632**	**13.0602**

highlights the effectiveness of DySTAGE as a powerful tool for financial analysis and decision-making.

5.7 Ablation Study (RQ4)

To quantify the improvement from each component, we implement the following variants of DySTAGE to examine the impact of removing each component:

- **DySTAGE w/o Importance** removes Asset-wise Importance Encoding
- **DySTAGE w/o Spatial** removes Pair-wise Spatial Encodings
- **DySTAGE w/o Edge** removes Edge-wise Correlation Encodings
- **DySTAGE w/o Temporal** replaces the Temporal module with a simple MLP layer

From Table 5, it is obvious that the temporal module significantly boosts the performance, especially a decrease of 0.11% on Russell and 0.12% on S&P in MAPE. Moreover, it demonstrates that DySTAGE equipped solely with the Topological Module is remarkably powerful, outperforming almost all benchmarks. In addition, all graph encodings consistently contribute to the model performance. Among them, *Asset-wise Importance Encoding* proves to be the most influential, especially on monthly data. Removing it leads to a 0.25% increase in Russell and a 0.34% in MLFI. Additionally, *Pair-wise Spatial Encoding* effectively improves MAPE of DySTAGE by revealing hidden connections between asset pairs.

6 Conclusion

This paper addresses a critical gap in the existing asset pricing literature. While prior research focuses on learning the interrelationships among assets within financial networks, it often overlooks how asset composition and relationships change over time. To bridge this gap, we introduce DySTAGE, a novel framework for dynamic graph representation learning in asset pricing from two aspects: the first universal formulation from asset time series to dynamic graphs accomodating changes in asset composition and relationships, and a graph learning model enriched with spatio-temporal and graph encodings with financial insights to capture the evolving nature of financial networks over time. Extensive experiments on monthly and daily assets demonstrate that DySTAGE outperforms popular benchmarks in price prediction and offers valuable insights for profitable investment strategies in real-world scenarios.

Future research may enhance DySTAGE's dynamic feature-capturing capabilities by integrating multimodal data, such as news sentiment and economic indicators, for richer market context. Additionally, incorporating advanced techniques, including graph generative adversarial networks and Large Language Models, could improve the model's adaptability, predictive precision, and and interpretability in response to evolving market structures.

References

[1] Jiandong Bai, Jiawei Zhu, Yujiao Song, Ling Zhao, Zhixiang Hou, Ronghua Du, and Haifeng Li. 2021. A3t-gcn: Attention temporal graph convolutional network for traffic forecasting. *ISPRS International Journal of Geo-Information* (2021).

[2] Filippo Maria Bianchi, Daniele Grattarola, Lorenzo Livi, and Cesare Alippi. 2021. Graph neural networks with convolutional arma filters. *IEEE transactions on pattern analysis and machine intelligence* 44, 7 (2021), 3496–3507.

[3] Andrew Y. Chen and Tom Zimmermann. 2022. Open Source Cross-Sectional Asset Pricing. *Critical Finance Review* (2022).

[4] Jinyin Chen, Xueke Wang, and Xuanheng Xu. 2022. GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction. *Applied Intelligence* (2022).

[5] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *CIKM*. 1655–1658.

[6] Rui Cheng and Qing Li. 2021. Modeling the momentum spillover effect for stock prediction via attribute-driven graph attention networks. In *AAAI*.

[7] Guillaume Coqueret and Tony Guida. 2023. *Machine Learning for Factor Investing: Python Version*. CRC Press.

[8] Luca F Di Cerbo and Stephen Taylor. 2021. Graph theoretical representations of equity indices and their centrality measures. *Quantitative Finance* (2021).

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

[10] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2018. Enhancing stock movement prediction with adversarial training. *arXiv preprint arXiv:1810.09936* (2018).

[11] Jingyi Gu, Fadi P Deek, and Guiling Wang. 2023. Stock broad-index trend patterns learning via domain knowledge informed generative network. *arXiv preprint arXiv:2302.14164* (2023).

[12] Jingyi Gu, Wenlu Du, and Guiling Wang. 2024. RAGIC: Risk-Aware Generative Adversarial Model for Stock Interval Construction. *arXiv preprint arXiv:2402.10760* (2024).

[13] Jingyi Gu, Sarvesh Shukla, Junyi Ye, Ajim Uddin, and Guiling Wang. 2023. Deep learning model with sentiment score and weekend effect in stock price prediction. *SN Business & Economics* 3, 7 (2023), 119.

[14] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[15] Bryan T Kelly, Seth Pruitt, and Yinan Su. 2019. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics* (2019).

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

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJU4ayYgl

[18] Guang Liu, Yuzhao Mao, Qi Sun, Hailong Huang, Weiguo Gao, Xuan Li, Jianping Shen, Ruifan Li, and Xiaojie Wang. 2020. Multi-scale Two-way Deep Neural Network for Stock Trend Prediction.. In *IJCAI*.

[19] Wen Long, Zhichen Lu, and Lingxiao Cui. 2019. Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems* (2019).

[20] Terence C Mills. 1990. *Time series techniques for economists*. Cambridge University Press.

[21] Sharan Narang, Hyung Won Chung, Yi Tay, William Fedus, Thibault Fevry, Michael Matena, Karishma Malkan, Noah Fiedel, Noam Shazeer, Zhenzhong Lan, et al. 2021. Do transformer modifications transfer across implementations and applications? *arXiv preprint arXiv:2102.11972* (2021).

[22] Brian K Nelson. 1998. Time series analysis using autoregressive integrated moving average (ARIMA) models. *Academic emergency medicine* (1998).

[23] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *ICLR*.

[24] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *AAAI*.

[25] AM Muntasir Rahman, Ajim Uddin, and Guiling Grace Wang. 2023. HODL: The Hold of Reddit Over the Stock Market. In *2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 36–39.

[26] Carlos A Ramírez. 2024. Firm Networks and Asset Returns. *The Review of Financial Studies* 37, 10 (2024), 3050–3091.

[27] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. 2020. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks. In *WSDM*. 519–527.

[28] Sreelekshmy Selvin, R Vinayakumar, EA Gopalakrishnan, Vijay Krishna Menon, and KP Soman. 2017. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *ICACCI*. IEEE.

[29] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. 2021. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *IJCAI-21*.

[30] Ajim Uddin, Xinyuan Tao, and Dantong Yu. 2021. Attention based dynamic graph learning framework for asset pricing. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1844–1853.

[31] Ajim Uddin, Xinyuan Tao, and Dantong Yu. 2024. The Network Factor of Equity Pricing: A Signed Graph Laplacian Approach. *Journal of Financial Econometrics* (2024), nbae010.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[33] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ

[34] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *ICML*. PMLR.

[35] Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. 2019. Understanding and improving layer normalization. *Advances in Neural Information Processing Systems* 32 (2019).

[36] Wei Yao, Jingyi Gu, Wenlu Du, Fadi P Deek, and Guiling Wang. 2022. Adpp: A novel anomaly detection and privacy-preserving framework using blockchain and neural networks in tokenomics. *International Journal of Artificial Intelligence & Applications* (2022).

[37] Junyi Ye, Bhaskar Goswami, Jingyi Gu, Ajim Uddin, and Guiling Wang. 2024. From Factor Models to Deep Learning: Machine Learning in Reshaping Empirical Asset Pricing. *arXiv preprint arXiv:2403.06779* (2024).

[38] Junyi Ye, Jingyi Gu, Ankan Dash, Fadi P Deek, and Guiling Grace Wang. 2023. Prediction with time-series mixer for the S&P500 index. In *2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 20–27.

[39] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. 2021. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *KDD*. 2037–2045.

[40] Haonan Yuan, Qingyun Sun, Xingcheng Fu, Cheng Ji, and Jianxin Li. 2024. Dynamic Graph Information Bottleneck. In *The Web Conference 2024*.

[41] Kaike Zhang, Qi Cao, Gaolin Fang, Bingbing Xu, Hongjian Zou, Huawei Shen, and Xueqi Cheng. 2023. DyTed: Disentangled Representation Learning for Discrete-time Dynamic Graph. In *KDD*.

[42] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems* (2019).