

Attention Based Dynamic Graph Learning Framework for Asset Pricing

Ajim Uddin

New Jersey Institute of Technology
Newark, New Jersey, USA
au76@njit.edu

Xinyuan Tao

New Jersey Institute of Technology
Newark, New Jersey, USA
xinyuan.tao@njit.edu

Dantong Yu

New Jersey Institute of Technology
Newark, New Jersey, USA
dtyu@njit.edu

ABSTRACT

Recent studies suggest that financial networks play an essential role in asset valuation and investment decisions. Unlike road networks, financial networks are neither given nor static, posing significant challenges in learning meaningful networks and promoting their applications in price prediction. In this paper, we first apply the attention mechanism to connect the “dots” (firms) and learn dynamic network structures among stocks over time. Next, the end-to-end graph neural networks pipeline diffuses and propagates the firms’ accounting fundamentals into the learned networks and ultimately predicts stock future returns. The proposed model reduces the prediction errors by 6% compared to the state-of-the-art models. Our results are robust with different assessment measures. We also show that portfolios based on our model outperform the S&P-500 index by 34% in terms of Sharpe Ratio, suggesting that our model is better at capturing the dynamic inter-connection among firms and identifying stocks with fast recovery from major events. Further investigation on the learned networks reveals that the network structure aligns closely with the market conditions. Finally, with an ablation study, we investigate different alternative versions of our model and the contribution of each component.

CCS CONCEPTS

- Mathematics of computing → **Graph theory; Graph algorithms;**
- Applied computing → **Forecasting; Economics;**
- Computing methodologies → **Neural networks.**

KEYWORDS

FinTech; Graph Neural Networks; Graph Attention; Diffusion Recurrent Convolution; Asset Pricing; Stock Price Prediction

ACM Reference Format:

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 and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3459637.3482413>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8446-9/21/11...\$15.00

<https://doi.org/10.1145/3459637.3482413>

1 INTRODUCTION

As the machine learning gains great successes in many applications, its application in predicting the price of financial assets has been increasingly interesting to both academic researchers and practitioners. Traditional pricing models mainly focus on firm-specific and market/macro factors, which cannot capture the inter-connection among assets. However, firms are not operating independently and one firm might be affected by the others through multiple networks. As a result, a firm’s stock price depends not only on its own characteristics but also on the characteristics of other relevant firms, i.e., the interconnection among firms affects each other’s market price. In this paper, we propose a novel two-step graph learning model to capture the dynamic interconnections among firms (connect the “dots”) and investigate their contributions to the stock price movement (network-aware prediction).

The idea that the interconnections in networks affect stock prices is intuitive. However, many challenges exist in how to capture the network structure of the equity market as it can be dynamic and complex. Recent efforts in network representation learning and Spatio-temporal modeling show that the network information improves traffic prediction [12, 15, 20, 29, 42] and COVID-19 trend forecasting [6]. These models are largely based on the Graph Convolutional Networks proposed in [25] that focus primarily on a static network with predefined topologies. Unlike social networks or road networks, the equity market’s network structure is unknown. Finance studies typically use the pairwise Pearson correlation of firms’ historical returns to represent firms’ network structure [11, 30, 38]. The Pearson correlation only reveals the linear relationship among entities and might not be sufficient to model the inter-dependency among stocks. Moreover, it is challenging to integrate the correlation networks in graph structure data as they contain both positive and negative coefficients. Existing approaches typically use absolute values with the assumption that significant correlation represents high similarity regardless of the sign. This assumption violates the core idea that a positive correlation indicates convergence, while a negative correlation indicates divergence. In [21], authors use sector similarity as the linkage among firms. However, as Figure 1 shows, falling in the same sector or industry group does not necessarily ensure similar returns. Many other latent factors, such as institutional holdings, may affect firms’ underlying connections. Moreover, unlike traditional networks, financial networks are often dynamic. Firms enter or exit the market, and change their business models, capital structures, and supply chains. As a result, relationships among firms continuously evolve (AEM vs. WMT in Figure 1). Therefore, directly applying techniques developed for static-known graphs [12, 15, 20, 29, 42] into dynamic equity market network may produce sub-optimal results.

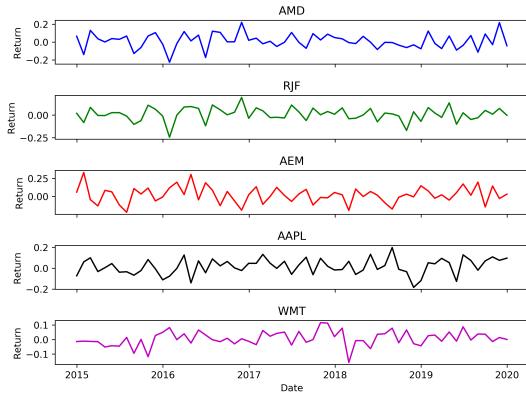


Figure 1: Rate of return from five assets over time. The timing of earning returns by AMD (Technology) is similar to that of RJF (Financial Services) and almost opposite to that from AEM (Materials-Mining). AMD and AAPL (Technology) are from the same industry sector, whereas their returns vary significantly. The dynamic nature of the changing relations among firms' return is also visible from the trend lines of AEM and WMT (Consumer Discount Stores). AEM and WMT started with little correlation in the early 2017, began to have strong co-movements between early 2017 and mid-2018, and then diverged into opposite movements from mid-2018 to December 2019.

This paper attempts to capture the time-varying networks of the equity market, model non-linear connectivity and dependency relationships, and use them for prediction. The main idea is: in the first step, an attention mechanism is used to learn the dynamic network structure of the US equity market, and in the second step, a recurrent diffusion convolution network is applied to model the spatial and temporal dependency among firms. The first step helps overcome the problem associated with unknown graphs. The second step considers both positive and negative connections and learns the graph embeddings with better predictive power on stock prices. The highlights of this paper are summarized as follows:

- To the best of our knowledge, it is the very first paper to propose the dynamic graph learning framework that tracks and follows the global and local patterns in the equity market over time. Our work enables financial analysis to be network-aware and minimizes the uncertainty associated with the prediction over stand-alone assets.
- We adopt a flexible attention mechanism to learn new networks from scratch or improve upon initial networks. The learned network topologies are non-linear and superior to the commonly used Pearson correlation and can capture the relationships of assets in the complex market environment.
- Unlike previous work, we use graph neural networks to integrate heterogeneous datasets and utilize the fundamentals, historical returns, and the US equity market network structure to improve the asset price prediction.
- The model is superior in prediction accuracy and portfolio performance compared to the conventional asset pricing methods and other off-the-shelf machine learning models.

The rest of the paper is organized as follows: Section 2 introduces the relevant work on graph application in asset pricing. Section 3 details the methodology of our work. Section 4 describes the experimental settings, data, and analysis results. Section 5 discusses the network learning capacity of our model and displays the evolution of network structure over time. We provide the results of the ablation study in Section 6 and offer the conclusions in Section 7.

2 RELATED WORKS

A surge of spectral-based graph neural network follows the seminal work in [25]. Many of these efforts target node classification and link prediction [9, 41]. Several spatiotemporal models enhanced by graph neural networks attain impressive performance in traffic prediction [12, 15, 20, 29, 42] and COVID-19 trend forecasting [6]. These models are largely based on the Graph Convolutional Networks proposed in [25] that focus primarily on a static network with predefined topologies. Financial networks, on the other hand, are time-varying without predefined structure. In this work, we propose a dynamic graph neural network framework, which can learn the time-varying network structure on its own, and show its usefulness in asset price prediction.

Predicting the asset price is of great importance to academic research and real-world investment. Traditional asset pricing models mainly focus on uncovering risk factors [14, 23]. Over the years, economists identify hundreds of factors to explain the variability in returns among assets [17]. The advancements in machine learning and deep learning allow researchers to use the full spectrum of these factors (for example, market, macroeconomic, and firm's accounting fundamentals) to predict returns or asset prices [18, 19, 27, 39, 43, 44]. Among them, auto-encoder is applied in [18, 39] to forecast stock returns from historical data. In [19], authors use Multi-layer-perceptron (MLP) on 94 characteristics variables to predict future returns of stocks. Nevertheless, these works ignore the network interconnections among financial assets and their importance in return prediction.

Recent finance research attempts to understand the network dynamics of financial assets and suggests that networks play an important role in information flow and shock transmission among financial assets [13, 26, 31, 36] and the corresponding institutions [34, 35]. They also show that network connectedness, centrality, and clustering coefficient evolve during crisis and pandemic situations [26, 34–36]. There are only a handful of studies that attempt to fill the gap and examine how the network enhances the predictability of the future stock returns [16, 21, 24, 28]. Our paper differs from these network-based approaches in multiple ways. In [16], authors use temporal graph convolutional neural network to rank assets for portfolio optimization. Unlike our work, they use feature similarities to learn the initial graph. The asset pricing model in [21] uses the multi-sector information to build a static input-output network. In contrast, our model captures the time-varying information and overcomes the drawback of the assumption on sector similarity. In [28], authors propose the LSTM Relational Graph Convolutional Network to predict the overnight stock movement. They use a static correlation matrix as the network among firms, which fails to capture both the latent components of the network and the dynamic changes in the network. The stock prediction model

in [24] applies hierarchical attention for learning node representations with only spatial convolution. Also, they do not consider any temporal dynamics. By contrast, our model in the recurrent diffusion step captures both spatial and temporal dependencies.

3 METHODOLOGY

In this section, we first define the problem of asset pricing prediction and then present the building blocks of the DYnamic Graph learning model for Asset Pricing (DY-GAP).

3.1 Problem Definition

The problem related to asset pricing here is defined as how to identify the intrinsic value of assets. One successful investment strategy is to identify undervalued (the current market price of the asset is lower than the intrinsic value) or overvalued (its market price is higher than the intrinsic value) stocks. Then, investors take long positions on (buy) undervalued stocks, or short positions on (sell) overvalued stocks, or both to make profits with the expectation that the market prices will eventually converge to the intrinsic values. The intrinsic values are not observed and need to be estimated by some asset pricing models. Asset pricing models can be grouped into two broader categories based on the input used: (i) time-series models and (ii) factor models.

Historical Return: Time series models are mainly based on historical returns. Researchers find that there exists time-varying pattern of stock returns. Stocks can maintain the performance in the short term named as the momentum effect, i.e., stocks with high (low) returns in the past are likely going to have high (low) returns in the near future. However, in the long term, stock prices could reverse. According to these theories, stock returns of a firm can be estimated with a function of its historical returns.

$$\hat{y}_{t+1} = f_1(y_t, \dots, y_{t-K})$$

here, K is the window size of historical return.

Factor Universe: Factor models are based on the assumptions that asset returns can be expressed as a linear function of a variety of macroeconomic, market, and security-specific factors. These factors include market risk premia, volatility, trading volume, and accounting fundamentals such as firm size, profitability, and liability.

$$\hat{y}_{t+1} = f_2(x_1, \dots, x_P)$$

Here, P are the total number of factors.

We formulate the asset pricing problem that combines these two types of inputs, the time-series returns and accounting fundamentals, to explain the return variability. First, we define multivariate temporal graphs at time t as $\mathcal{G}_t = (V_t, A_t)$, where V_t is the set of firms (nodes) $|V_t| = N$ and $A_t \in \mathbb{R}^{N \times N}$ is a weighted adjacency matrix representing firms' quantitative proximity at time t . The value of A_{ij} indicates the strength of the interdependence. $A_{ij} = 0$ indicates that the firms i and j are independent to each other. The input signal on graph at time t is $X_t = \{x_{ip}\} \in \mathbb{R}^{N \times P}$ and the output is $Y_{t+1} = \{y_i\} \in \mathbb{R}^{N \times 1}$, where N is the number of firms (nodes). Instead of using pre-defined networks, we use neural networks to learn the adjacency matrix A_t at time t from the historical returns $Y_t = \{y_{ik}\} \in \mathbb{R}^{N \times K}$, where K is the rolling window size of historical returns.

Given the observed returns of previous K timestamp Y_t, \dots, Y_{t-K} and graph signal X_t , the objective of the model is to learn an effective network structure \mathcal{G}_t at each time step and in the meanwhile to predict the next time step \hat{Y}_{t+1} with the integrated model of Graph Neural Networks and Recurrent Neural Networks. We use neural network to implement the forecasting function $f(\cdot)$ with parameters $\Theta = \theta_1, \theta_2, \theta_3$ (θ_1 represents the neuron weights for “connecting the dots”, θ_2 contains the parameters for the graph convolutional filter g_{θ_2} defined on \mathcal{G}_t , and θ_3 represents the neuron weights for transforming the input node features.):

$$\begin{aligned} G_t &= \text{Attn}(Y_t, \dots, Y_{t-K}; \theta_1) \\ \hat{Y}_{t+1} &= f(g_{\theta_2} \star X_t; \theta_2, \theta_3) \end{aligned} \quad (1)$$

3.2 DY-GAP Model Framework

Figure 2 presents the architecture of our proposed model. The model consists of three different learning functions: (i) Embedding learning, (ii) Graph learning, and (iii) Spectral and temporal dynamics learning. We follow [3] and apply the self-attention function to learn dynamic network structures from the historical return data. The return data is often noisy and large. An embedding learning layer is used to clean the noisy data and obtain the firm's condensed representation before performing the attention function. Finally, a spectral-temporal recurrent convolution function is performed on the firms' fundamentals using the learned network.

3.2.1 Embedding Learning Layer. Principle Component Analysis (PCA) is widely used in finance for dimension reduction and feature extraction from time series data [4, 22]. At each time period t , we perform Singular Value decomposition (SVD) on historical returns and extract principle components for each firm. For simplicity, we drop the time index for the remaining discussion. Given $Y \in \mathbb{R}^{N \times K}$, we perform SVD on Y as follows:

$$Y = USV^\top \quad (2)$$

where, U is a unitary matrix and S is the diagonal matrix of singular values corresponding to the Eigenvalues in the correlation matrix and $H = US$ is the principle components. We perform dimension reduction and choose the first L principal components $\sum_{i=1}^L U_{:i} S_{:i}^\top$ as the embedding matrix of return data, where $L < K$. With N firms, $H = \{h_i\} \in \mathbb{R}^{N \times L}$ holds the initial node embeddings (features).

3.2.2 Graph Attention Learning Layer. The graph learning layer employs the attention mechanism proposed in [3] to learn the edge coefficient between two nodes. The attention function is performed on each firm's learned embedding H . Following [40], each node in the attention pair is first undertaken a linear transformation with weight matrix $W \in \mathbb{R}^{L' \times L}$ before attended by shared attention mechanism $a : \mathbb{R}^{L'} \times \mathbb{R}^{L'} \rightarrow \mathbb{R}$. The learned attention coefficient is:

$$e_{ij} = a(Wh_i, Wh_j) \quad (3)$$

The learned value e_{ij} indicates the importance of firm j 's return on that of firm i . We adopt a nonlinear activation function to add nonlinearity to the dependence relation among firms and apply the softmax operation to make the coefficient easily comparable across nodes. The final network edge α_{ij} is defined as follows:

$$\alpha_{ij} = \frac{\exp(\sigma(a^\top [Wh_i || Wh_j]))}{\sum_{j=1}^n \exp(\sigma(a^\top [Wh_i || Wh_j]))} \quad (4)$$

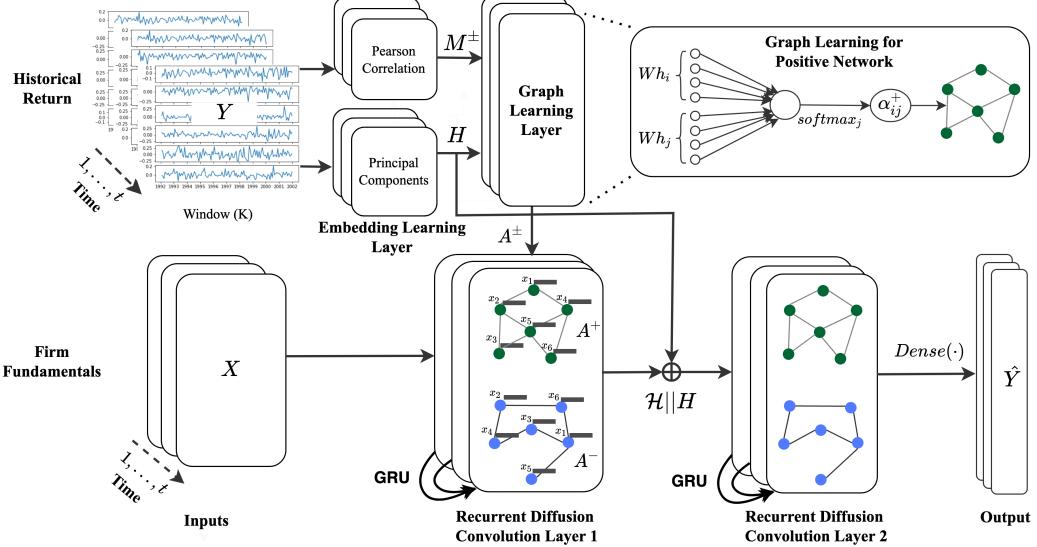


Figure 2: The three-stage DY-GAP model architecture. (1) PCA on the historical return learns latent embedding for each firm. (2) Self-attention on the latent embedding learns the network architecture. Two different attention mechanisms are performed to learn both positive (green) and negative (blue) networks. Pearson correlation of historical return ensures masked attention. (3) A diffusion convolution on firms’ signals uses the learned network for learning spatial dependency, and GRU recurrent neural network learns temporal dependency. Two diffusion layers are used: the first one with the firm fundamentals and the second one after concatenating latent embedding with the output of first diffusion layer. Solid arrows indicate the flow of information.

Here \top , \parallel , and σ represent transpose, concatenation and nonlinear activation operation, respectively. $a \in \mathbb{R}^{2L'}$ is a weight vector parametrizing the attention mechanism $a(Wh_i, Wh_j)$. The softmax operation satisfies $\sum_{j=1}^n \alpha_{ij} = 1$ and $\alpha_{ij} >= 0$, and thereby, provides normalization across the learned network.

In [40], the authors suggest that the desired objective be achieved with a simple attention strategy attending all node pairs while ignoring the structural information. However, as discussed in Section 1, the interactions among firms are not uniform and rather complex: some of them are converging and render the market towards the same direction, while the others are diverging and lead to the heterogeneous behaviors in the market. Later, we show in Section 5 that the distribution of the interaction relationships aligns with the market conditions.

A firm is not necessarily connected to all other firms, and a spurious relationship might do more harm than no connection due to the overfitting problem and excessive computation costs. Therefore, to expedite the graph learning process, we use the masked attention function where only the closely relevant firms of the target firm are attended. We use the Pearson correlation coefficients of the historical return data to determine a firm’s initial connection and potential neighbors. The Pearson’s correlation, ρ_{ij} , between firm i and j at time t with a rolling window K is defined as follows:

$$\rho_{ij} = \frac{\text{Cov}(r_i, r_j)}{\sqrt{\text{Var}(r_i)\text{Var}(r_j)}}$$

We keep the rolling window size to be the same as the embedding learning framework window for both daily and monthly data.

The original return correlation matrix is dense, with almost all firms’ returns (positively or negatively) correlated to others. The values close to zero do not provide useful information about firms’ similarity but noise. To enhance the signal-to-noise ratio (SNR), we adopt the noise filtering technique proposed in [11]. The signal enhancement algorithm first denoises the empirical Pearson covariance matrix by performing an eigendecomposition and replacing the noisy eigenvalues with their average to preserve the trace of the correlation matrix. The denoised matrix might not be a positive definite symmetrical matrix. Finally, the signal enhancement algorithm applies convex optimization to construct a correlation matrix that is the closest to the denoised matrix.

As a result, the correlation matrix has positive and negative coefficients. A positive correlation between i and j indicates that i and j reassemble to each other, while a negative one means the opposite. A single attention head treats both positive and negative coefficients indiscriminately and violates the correlation property. Therefore, to learn meaningful network representations, we use two separate attention heads, each attending the positive and negative components of the firm correlation matrix separately. We first decompose the new correlation matrix into two mask matrices: M^+ , where $M_{i,j}^+ = 1$ if $\rho_{i,j} \geq 0$, otherwise 0; and M^- where $M_{i,j}^- = 1$ if $\rho_{i,j} < 0$, otherwise 0. Then we perform the following masked attentions according to the positive and negative matrices:

$$\alpha_{ij}^\pm = \frac{M_{ij}^\pm \exp(\sigma(a^\top [Wh_i || Wh_j]))}{\sum_{j=1}^n M_{ij}^\pm \exp(\sigma(a^\top [Wh_i || Wh_j]))} \quad (5)$$

The learned attention coefficients α_{ij} are assembled into two new affinity matrices $A^+ = \{\alpha_{ij}^+\} \in \mathbb{R}^{N \times N}$ and $A^- = \{\alpha_{ij}^-\} \in \mathbb{R}^{N \times N}$.

3.2.3 Recurrent Diffusion Convolution Layer. We use the recurrent diffusion function to model the spatial relationship (network) and temporal dependency among firm fundamentals based on the learned network structure. We follow the diffusion process in [29] to propagate the firm information to their neighbors.

Graph Diffusion. We apply two separate diffusions in two networks A^\pm learned from the attention function. The diffusion process consists of a random walk on Graph \mathcal{G} with the state transition matrix (A^+ or A^-) and its random walk normalization $D^{\pm^{-1}}$ where D^\pm is the diagonal matrix of the node degree, $D_{ij}^\pm = \deg(v_i)$ if $i = j$, otherwise 0.¹ Following [29], we define the diffusion convolution on graph filter f_θ and each input channel $X_{:,p} \in \mathbb{R}^N$ ($1 \leq p \leq P$) of the graph signal consisting of firm fundamentals as follows:

$$f_\theta \star_{\mathcal{G}} X_{:,p} = \sum_{s=0}^{S-1} \theta_{s,1} (D^{+^{-1}} A^+)^s X_{:,p} \| \sum_{s=0}^{S-1} \theta_{s,2} (D^{-^{-1}} A^-)^s X_{:,p} \quad (6)$$

where f_θ denotes the graph diffusion filters, $\|$ concatenates the positive and negative diffusions of X , and $\theta \in \mathbb{R}^{S \times 2}$ are the parameters for filter. $D^{+^{-1}} A^+, D^{-^{-1}} A^-$ are the (random-walk normalized) transition matrices of the diffusion process learned from the positive and negative correlations, respectively. $(D^{\pm^{-1}} A^\pm)^s X$ is a s -step matrix power iteration on input X (s -step diffusion). The weighted sum parameterized by θ represents the learned features from the up-to- S -hop neighborhood via diffusion. S denotes the maximum number of diffusion steps. In [29], the authors show that a sufficient large number of diffusion will converge to a stationary distribution. The S -step diffusion network becomes a spectral graph convolutional neural network once we replace the attention matrices A^\pm with symmetrical matrices $\hat{A}^\pm = \frac{A^\pm + A^{\pm \top}}{2}$ and then perform symmetrical normalization. Equation 6, with these necessary modifications, can be transformed into a generalized form of the polynomial filter defined on graph filter matrix with the learnable parameters θ (Equation 3 in [10]).

It is important to note that, unlike the work in [29], our model is based on the undirected graphs with the positive and negative relationships, performs diffusion on these two different networks, and concatenates two outputs. Instead of transposing the original network, the second term in Eqn 6 employs the negative network. The two-term diffusion allows us to learn the spatial dependencies from the positively connected neighborhood and negatively connected neighborhood. The diffusion convolution layer takes $X \in \mathbb{R}^{N \times P}$ as the input and generates the tensor output $\mathcal{H} \in \mathbb{R}^{N \times Q \times 2}$, where P is the number of input channels and Q is the number of output channels. For each output channel $q \in 1, \dots, Q$, the output contains two components: one from the positive diffusion and the other one from the negative diffusion, and is in the form of:

$$\mathcal{H}_{:,q} = \sigma \left(\sum_{p=1}^P f_{\Theta_{p,q,:}} \star_{\mathcal{G}} X_{:,p} \right) \quad (7)$$

¹In our case, all row sums in A^\pm equal to 1, and degree matrices D^\pm become an identity matrix. We still keep D^\pm in the subsequent convolutional diffusion equations and make them generalize to any other un-normalized affinity matrix A^\pm .

here, $\Theta_{p,q,:} \in \mathbb{R}^{S \times 2}$. $\{f_{\Theta_{p,q,:}}\}$ are the diffusion filters and σ is the activation function, i.e., ReLU.

GRU for Temporal Modeling. Asset pricing models suggest that past returns (prices) of a stock have the predictive power for its future returns (prices). It is essential to model the temporal dependency along with the spatial dependence. Gated Recurrent Units (GRU) [8] is proved to be effective for temporal dependency modeling. As suggested in [29], we replace the common linear transformation matrices for all GRU gates with the graph diffusion and augment the diffusion convolution with GRU as follows:

$$\begin{aligned} r^t &= \sigma(\Theta_r \star_{\mathcal{G}} [X^t, \mathcal{H}^{t-1}] + b_r) \\ u^t &= \sigma(\Theta_u \star_{\mathcal{G}} [X^t, \mathcal{H}^{t-1}] + b_u) \\ C^t &= \sigma(\Theta_c \star_{\mathcal{G}} [X^t, (r^t \odot \mathcal{H}^{t-1})] + b_u) \\ \mathcal{H}^t &= u^t \odot \mathcal{H}^{t-1} + (1 - u^t) \odot C^t \end{aligned} \quad (8)$$

where $X^t, r^t, u^t, \mathcal{H}^t$, denote the input, reset, update gate and the hidden state at time t , respectively. $\star_{\mathcal{G}}$ denotes the graph convolution diffusion and $\Theta_{r,u,c}$ contains the parameters of the respective filters for each gate in GRU.

The embedding H serves two roles in the proposed framework in Figure 2: learning attention coefficients and supplying the stock market information of each firm. To support the second role, we design two recurrent diffusion convolution layers to process the firm signals in Figure 2. The first layer performs the diffusion convolution on the firm fundamentals and the hidden state $[X^t, \mathcal{H}^{t-1}]$. The second layer concatenates the output \mathcal{H}^t from the first diffusion layer and the embedding H from the embedding learning module, and applies another recurrent diffusion on $\mathcal{H}^t || H^t$. This concatenation function ensures that the return data is directly incorporated into the recurrent diffusion convolution framework. Consequently, we avoid the vanishing gradient problem and significantly improve the learning process. The recurrent diffusion using GRU defined in Eqn 8 is applied in these two layers, as shows in Figure 2.

Finally, for prediction, the diffusion network simply uses one layer MLP to regress the prediction output $\hat{Y}_{t+1} \in \mathbb{R}^{N \times P}$ on the hidden state of the second recurrent diffusion convolution layer.

4 EXPERIMENTAL DETAILS

4.1 Data

For the experiment, we collect two sets of data with different frequencies – monthly and daily. The monthly data include 1098 stocks from Russell 3000 Index. The sample period is from January 01, 1990, to December 31, 2019, and divided into three folds: training (January 1990 to December 2009), validation (January 2010 to December 2013), and test (January 2014 to December 2019). The daily data involves all the stocks in the S&P-500 index. The sample period is from January 01, 2010, to December 30, 2020, and divided into training, validation, and test sub-sample as January 01-2010 to December 31-2015, January 01-2016 to December 31-2017, and January 01-2018 to December 31-2020, respectively. For both monthly and daily data, selected firms are based on the index constituents by October 31, 2020. Few inconsistent firms are removed from the sample. The monthly and daily stock returns are from CRSP, and firms' fundamental variables are from Compustat. These two datasets are

Table 1: Fundamentals Variables

Firm Characteristics	Calculation Procedure
Total assets to market	Total asset / market value of equity
Size	$\log(p_t \times share\ outstanding)$
Turnover	Volume / share outstanding
Growth rate of volume	$V_t - V_{t-1}/V_{t-1}$
Growth rate of share outstanding	$SO_t - SO_{t-1}/SO_{t-1}$
Closeness to past year high	$P_{t-1} - \max(P_{t-1}, \dots, p_{t-12})/\max(p_{t-1}, \dots, p_{t-12})$
Closeness to past year low	$P_{t-1} - \min(P_{t-1}, \dots, p_{t-12})/\min(p_{t-1}, \dots, p_{t-12})$
Spread	$p_{t-1}^h - p_{t-1}^l$, monthly [daily] high minus low price
Opening and closing spread	$(p_{t-1}^o - p_{t-1}^c)$, daily opening minus closing price
Capital gain	Value is 0, if no capital gain is recorded
EPS	Earning per share
Dividend	Dividend paid in cash
Total volatility	Price volatility of last 60 months [last 252 days].
Idiosyncratic volatility	Total volatility - market volatility
Market return	Return on S&P-500 index
CAPM market Beta	Beta on Fama-French market factor
Small minus big beta	Beta on Fama-French size factor
High minus low beta	Beta on Fama-French value factor
1 week momentum	$(p_{t-1} - p_{t-5})/p_{t-5}$ - daily only
2-week momentum	$(p_{t-1} - p_{t-10})/p_{t-10}$ - daily only
1-month momentum	$(p_{t-1} - p_{t-2})/p_{t-2}$ $[(p_{t-1} - p_{t-21})/p_{t-21}]$
2-month momentum	$(p_{t-1} - p_{t-42})/p_{t-42}$ - daily only
3 month momentum	$(p_{t-1} - p_{t-3})/p_{t-3}$ $[(p_{t-1} - p_{t-63})/p_{t-63}]$
6 month momentum	$(p_{t-1} - p_{t-6})/p_{t-6}$ $[(p_{t-1} - p_{t-126})/p_{t-126}]$
12-month momentum	$(p_{t-1} - p_{t-12})/p_{t-12}$ - monthly only

available at [<https://wrds-www.wharton.upenn.edu/>]. Based on the data availability, fundamental variables included in the daily and monthly analysis are slightly different. Following the asset pricing literature, we calculate 21 monthly features and 24 daily features from firms' fundamentals. The detailed calculation procedure is in Table 1.

4.2 DY-GAP Setting

We implement all machine learning-based models in Python using Pytorch [33] and/or TensorFlow [1]. For monthly data, the window size K for historical returns for embedding layer and for Pearson correlation is 36 months; for daily data, it is 122 days (6 months). In the embedding layer, we use embedding size $L = 10$. For attention, we use eight attention heads, and for diffusion convolution, we use ten diffusion steps. Other hyperparameters for the attention and diffusion convolution layer are selected based on the validation result. We use early stopping criteria for model training and stop training if validation loss does not decrease in 10 Epoch.

4.3 Baselines

We first compare our model with a series of multi-factor pricing models and time-series methods. For multi-factor pricing models, we consider the well-acknowledged Fama-French five-factor model [14], multivariate regression using all fundamentals [2] and recently proposed empirical asset pricing via machine learning (EAP-ML) [19] model. For time-series models, we use the classic ARIMA and several advanced deep neural network-based approaches, including fully connected long short-term memory (FC-LSTM) [37] and the state-of-the-art neural basis expansion analysis for interpretable time series forecasting (N-BEATS) [32]. Finally, as our approach is inspired and combines two models: graph attention network (GAT) [40] and diffusion convolution recurrent neural network (DCRNN) [29], we compare the performance of our model with them in order

to ensure the advantage of integration. For most models, we use authors' source codes, if available, with necessary modifications. The best hyper-parameters are chosen based on the validation dataset. For MR, FF-5, and ARIMA, coefficients are determined with the training and validation data, and then the learned coefficients are used to estimate the performance values for test data.

4.4 Forecasting Future Returns

The model performance is first evaluated in terms of the prediction accuracy of future returns. There are three matrices to assess the prediction performance, including RMSE, MAE, and MAPE. The small values represent small prediction errors and high accuracy. Table 2 presents the prediction performance of each model. The left panel represents results from the monthly data (Russel-3000 Index), and the right panel represents results from the daily data (S&P-500 Index). Reported results are from the test data sets only. The stochastic nature of machine learning models may lead to different forecasts with different initialization. Therefore, we run each model ten times with different random seeds and report the average performance and the standard deviation.

Our proposed model DY-GAP outperforms other models in all three performance metrics for forecasting with monthly data. In RMSE, DY-GAP outperforms FF-5 and EAP-MLP by 9% and 12%, respectively. The two best performing models among all machine learning-based baseline models are FC-LSTM and DCRNN. Nevertheless, DY-GAP still outperforms these two models by 10%, and 6% in RMSE and 7% and 5% in MAPE, respectively.

The prediction performance of our model (DY-GAP) is consistently better using daily data with only one exception where DCRNN is the best model in terms of RMSE. However, in terms of MAE and MAPE, our model is still the best. In the later section, we show that DCRNN has a high risk that lowers its Sharpe Ratio. The prediction error of DY-GAP in terms of MAPE is smaller than DCRNN by 7%, GAT by 11%, N-BEAT by 10%, and FF-5 by 25%. It is worth noting that the superiority of sophisticated predictive models is also visible in the result of daily data. As the data frequency and size increase, sophisticated models can take advantage of large datasets to minimize their overfitting problem and use their increased learning capacity to reduce bias. As a result, the state-of-the-art N-BEATS outperforms the earlier deep learning model (FC-LSTM) by 5% in terms of RMSE with daily data and becomes the third-best model.

The integration of two powerful methods and nonlinearity explain the superior performance of DY-GAP. The use of both attention and diffusion function enables our model to harness the advantage of both GAT and DCRNN and outperforms the application of these two models individually. DY-GAP has clear superiority over time series models, such as ARIMA, FC-LSTM, and N-BEATS, as DY-GAP considers both time series and factor information. In addition, these time series models are applied for individual firms and therefore, information from the market or related firms is ignored in these models. Compared with multi-factor pricing models, such as FF-5 and EAP-MLP, our model considers the spatial connectedness and nonlinear interaction among asset returns and attains a great performance improvement compared to these traditional models.

Table 2: Forecasting results

	Monthly Data			Daily Data		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
MR	0.1211	0.0944	66.4063	0.0310	0.0251	24.5875
FF-5	0.0941	0.0843	58.9432	0.0274	0.0209	20.8055
ARIMA	0.1141	0.0852	62.8968	0.0351	0.0252	32.6051
EAP-MLP	0.0969±.009	0.0726±.007	61.5435±7.3	0.0519±.018	0.0524±.017	42.0046±9.1
FC-LSTM	0.0949±.006	0.0723±.006	56.1883±4.2	0.0254±.003	0.0183±.001	19.6615±0.9
N-BEATS	0.1065±.001	0.0739±.001	61.4771±2.3	0.0241±.008	0.0173±.002	17.3975±1.1
DCRNN	0.0912±.002	0.0727±.001	54.8688±1.2	0.0230±.005	0.0178±.002	16.8916±1.0
GAT	0.0953±.006	0.0778±.005	61.7650±5.5	0.0292±.006	0.0195±.005	17.4611±1.6
DY-GAP	0.0853±.002	0.0632±.001	52.2400±1.3	0.0233±.003	0.0161±.001	15.6098±0.7

Table 3: Portfolio performance

	Monthly Data			Daily Data		
	Average (%)	STD (%)	Sharpe Ratio	Average (%)	STD (%)	Sharpe Ratio
MR	1.0046	4.0400	0.8614	0.0120	1.7669	0.1081
FF-5	1.0980	4.4340	0.8578	0.0116	1.7450	0.1055
EAP-MLP	1.1455	5.1880	0.7649	0.0733	2.2850	0.5092
ARIMA	1.1840	4.9540	0.8279	0.0401	1.6876	0.3772
LSTM	1.1790	4.8877	0.8356	0.0322	2.0711	0.2468
N-BEATS	1.1046	4.0400	0.9471	0.0424	1.8450	0.3648
DCRNN	0.9121	5.1650	0.6117	0.0218	1.6104	0.2149
GAT	1.0820	4.7730	0.7853	0.0220	1.7669	0.1980
S&P-500	0.8236	3.2534	0.8769	0.0449	1.4892	0.4785
DY-GAP	1.5500	4.5617	1.1771	0.0854	1.9500	0.6955

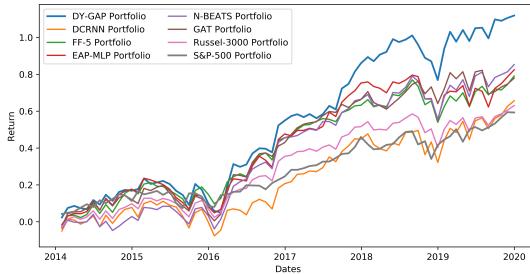


Figure 3: The portfolio performance from the monthly data. At the beginning of each month based on each model prediction, we hold the top 10% stocks and at the end of the month, we liquidate the stocks. The thick blue line is the cumulative return of our model in the test period and thick gray line is the S&P-500 index return during the test period. Russel-3000 is the average return of all studied firms.

4.5 Portfolio Performance

In this section, we show the economic benefits of our model via the portfolio analysis. For the monthly data, we take a long position on (i.e., buy) the top 10% stocks with the highest predicted returns at the beginning of each month, hold the position until the end of the month, and then liquidate (i.e., sell) all stocks. Figure 3 represents the cumulative returns of the portfolios by different

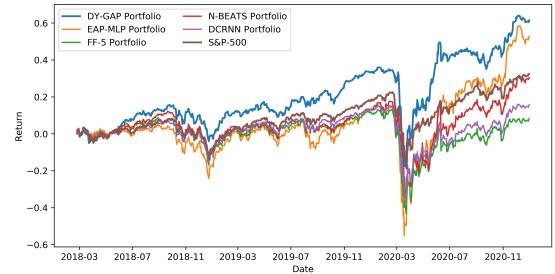


Figure 4: The portfolio performance from the daily data. For each trading day, we take the long position at (i.e., buy) the top 10% of highest predicted stocks. The thick blue line is the cumulative return of our model in the test period

models from January 2014 to December 2019. The thick blue line represents the cumulative return of our model in the test period, and the thick gray line is for the S&P-500 index return during the test period. Although in the early years, the performance of the DY-GAP portfolio is indistinguishable from others, our model stands out over time. The cumulative return of our model doubles both the S&P-500 index return and the weighted average return of all stocks in the Russel-3000 index. The finding suggests a profitable investment strategy by identifying the "success" group (i.e., the top 10% stocks with the highest returns) with our model. Among all

other baseline models, EAP-MLP and N-BEATS also perform well in the cumulative return.

Table 3 reports the monthly average rate of return, monthly standard deviation, and annualized Sharpe ratio of different portfolios. Sharpe ratio is an essential measure of portfolio performance because it provides the return by the unit of risk and exposes which portfolio offers the best returns (rewards) by taking the same amount of risk. The higher the Sharpe ratio indicates the better predictability of a model. Following the formula from Morningstar, Inc, we calculate the annualized (yearly) Sharpe ratio from monthly returns as $SR = \frac{\sum_{m=1}^M r_m}{\sigma_r} \times \sqrt{12}$. Our proposed model earns a high return rate (1.5% monthly) with relatively low risk (3%). Although DCRNN has high predictive power in Section 4.4, the portfolio analysis shows that the risk associated with DCRNN is much higher. As a result, the Sharpe ratio of DCRNN is lower than our model. S&P-500 index portfolio has the lowest risk, but it generates a low rate of return. The Sharpe ratio of the S&P-500 index portfolio is similar to that of MR, FF-5, ARIMA, and LSTM. The Sharpe Ratio of our model, DY-GAP, is the highest with 1.18 and significantly higher than that of the second-best model, N-BEATS, with 0.95.

The performance decline of DCRNN and the improvement in N-BEATS in portfolio evaluation attribute to their model formulation. During training, the models similar to DCRNN learn global parameters by minimizing errors over all firms, whereas N-BEATS learns individualized parameters for each firm. In portfolio analysis, only the selected (top 10%) firms are included. Other firms may have a better prediction, but do not fall in the leading 10% group, and are excluded from the portfolio. However, the evaluation results suggest that our model is less susceptible to this issue, has a better prediction accuracy, and provides a good portfolio performance.

We also perform a portfolio analysis on the S&P-500 stocks using daily data. Figure 4 shows the cumulative return from the portfolio constructed based on the daily prediction. Our model, DY-GAP, maintains superior performance over all other models throughout the test period. Our cumulative return is more than doubled compared with the S&P-500 index return. There is an interesting pattern that the up-down trend in the portfolio performance from all models echoes the market return. All models, including the DY-GAP, are affected by the market's downside, particularly for unexpected events like the COVID-19 pandemic. However, the incorporation of network and spatial dependency allows DY-GAP to select the stock groups that generate relatively higher returns in the middle of the pandemic. EAP-MLP is the second-best, especially in the latter part of 2020. This finding suggests that our model is able to identify stocks (firms) with faster recovery during recession periods.

5 GRAPH LEARNING

This section evaluates the network learning capacity of our proposed model. The dynamism of the US equity market is evident in the learned networks from the S&P-500 stocks daily data. Figure 5 presents the network structure at six different time points: June 26, 2016 - one day after Brexit Referendum (5a); June 26, 2017 - a normal network (5b); March 23, 2020 - three days after State-wide stay-at-home order declared in New York (5c); June 25, 2020 - three days after NYC meets the conditions for Phase 2 reopening (5d); November 5, 2020 - two days after the U.S. general election (5e);

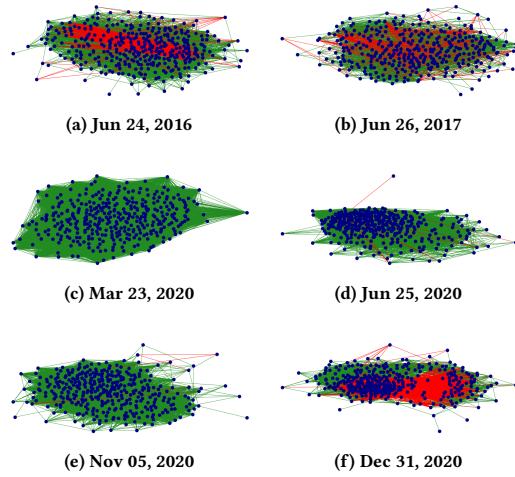


Figure 5: The network structure of S&P-500 stocks at different points of time. Green represents positive edges and red represents negative edges. During the normal economic condition, the market consists of a mixture of positive and negative edges (sub-figures b and f). However, during the economic or political event, the network is dominated by positive edges, representing the commonality across firms (sub-figures c and e).

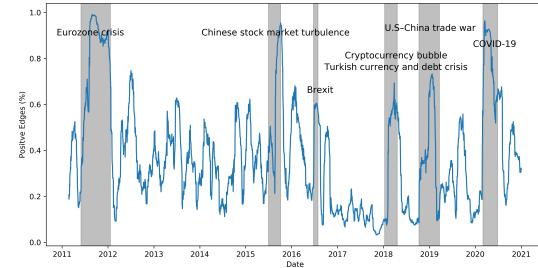


Figure 6: The learned positive edges from S&P-500 daily data. There are visible spikes in the degree of positive edges during financial, political, and economic events.

and December 31, 2020 - that is the last day of our study period and another example of normal network structure (5f).² The analysis shows that stocks become more homogeneous during market shocks, i.e., there are similar upward and downward trends among stocks. The pattern is also obvious in Figure 6. In-depth observations show that the impacts of all events are not identical. During the Brexit, the network structure changes, but all the negative edges do not vanish overnight. During the initial COVID-19 lockdown and the U.S. general election, almost all edges turn to be positive. However, when the states start to introduce phased reopening, the negative edges begin to appear again (sub-figure 5d) and over time, the market resumes its original structure (sub-figure 5f). These results are consistent with a number of recent studies where authors

²Because of holidays and weekends, these days are the next trading days after the events.

find that the correlation coefficient, network centrality, and clustering coefficient among financial assets increase during COVID-19 [5, 26, 36].

6 ABLATION STUDY

To gain a better understanding of the contributions of individual components of our model, we perform a series of ablation studies: (i) without any network structure (DY-GAT), (ii) considering both positive and negative relationships in a single network (DY-GAP-Single), (iii) with a new fund network from investment firm data (DY-GAP-FUND), and (iv) replacing diffusion convolution with simple graph convolution (GCN) as suggested in [25] (DY-GCN). In the ablation study, we only consider monthly data. Figure 7 shows the validation loss curves among all ablation models. Table 4 reports the prediction accuracy from alternative models.

DY-GAT ignores the constraints on network connectivity and allows all firms to attend all other firms. This change to the model causes the performance to decrease by 24% compared to the model abiding by the restrictions. In DY-GAP-Single and DY-GCN, the performance drops by 12% and 2%, respectively. Figure 7 shows that the validation error of these three models converges to approximately identical values.

The final and most promising alternative is the mutual connection network of the investment funds and assets. The investment fund shareholding is quarterly data providing the holding information of a firm's share by each portfolio manager. This data provides important information about institutional investors and their stock holding positions. The underlying assumption is that if the shares of two firms appear in the same portfolio, these two firms should have some connections. Although the holding positions on the shares of two firms by one portfolio appear to be coincidental, multiple holdings of the same two firms' shares are informative. The idea of using investment behavior is similar to [7], but we use the weighted side graph of the asset (and associated firm) nodes generated from the bipartite network between stocks and investment funds. First, we build a bipartite network between firms and investment funds from the quarterly mutual fund investment data and represent it as a matrix $B \in \mathbb{R}^{N \times N'}$, where N is the number of firms. Second, we project the bipartite network onto the firm nodes and obtain the side network for firms with the affinity matrix BB^\top . The elements in the normalized adjacency matrix represent the connectivity between two firms with the value equal to the ratio between the number of the funds investing on the two firms and the total number of funds. Finally, we use a threshold function to remove noisy edges, i.e., $M = (\rho_{ij})$ if $\rho_{ij} \geq \bar{\rho}_i$ otherwise 0. Here, ρ_{ij} represents the edge between firm i and firm j , and $\bar{\rho}$ represents the median value of all edges connecting to firm i . Using the threshold function, we enforce our earlier assumption that the coexistence of two firms in one (or a small number of) portfolios may be random. The coexistence of multiple portfolios represents similarity. Finally, we use this M to replace M in subsection 3.2.2 to perform masked attention on Eq. 5.

Although the loss function in our model that incorporates the fund network converges slower than that in other models, it has the second-lowest validation error after DY-GAP. Its prediction accuracy is also higher than other alternatives. The enhanced performance attributes to the information at the portfolio management

Table 4: Forecasting results from alternative versions

	RMSE	MAPE (%)
DY-GAP	0.0853	52.2400
DY-GAT	0.1121	61.7160
DY-GCN	0.0936	53.1476
DY-GAP-FUN	0.0908	52.6881
DY-GAP-Single	0.1072	58.4270

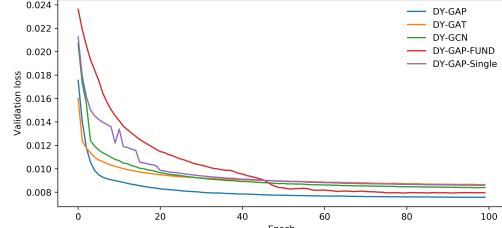


Figure 7: Loss curve with our proposed model and some other alternative of network and convolution operation. DY-GAP considers both positive and negative networks and diffusion convolution achieves the lowest validation error.

level, where the fund managers may hold some information advantage over the general market due to their skills and experience. The fund network allows us to incorporate the additional information into the asset pricing model.

7 CONCLUSION

In this paper, we propose a graph neural network-based approach to asset pricing. This work offers a novel solution to two critical problems in asset pricing: the interrelation among the firms and the evolution of the interrelation. Our model outperforms many traditional asset pricing models and advanced machine learning models in terms of prediction accuracy and portfolio performance. We also conduct analyses on several alternative models in the ablation study and show that DY-GAP with the positive and negative correlated networks and diffusion convolution layer performs the best. This paper confirms that the firm interconnection is bidirectional and relevant to the market analysis. The positive and negative relations must be treated differently because they serve different roles in the market. The incorporation of an effective network representation and the model for spatial and temporal relations enhances the stock price prediction and more profoundly, improves our understanding of the network structure in the financial market. Our model is still sensitive to some market conditions during abnormal events, e.g., COVID-19, and follows market downturns. The future research direction includes improving the model under various market conditions and design mitigation strategies.

ACKNOWLEDGMENTS

Research reported in this publication was supported by the National Center for Advancing Translational Sciences (NCATS), a component of the National Institutes of Health (NIH) under award number UL1TR003017. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

REFERENCES

- [1] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).
- [2] Tobias Adrian, Richard K Crump, and Emanuel Moench. 2015. Regression-based estimation of dynamic asset pricing models. *Journal of Financial Economics* 118, 2 (2015), 211–244.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.
- [4] Ruslan Bikbov and Mikhail Chernov. 2010. No-arbitrage macroeconomic determinants of the yield curve. *Journal of Econometrics* 159, 1 (2010), 166–182.
- [5] Elie Bouri, Oguzhan Cepni, David Gabauer, and Rangan Gupta. 2021. Return connectedness across asset classes around the COVID-19 outbreak. *International review of financial analysis* 73 (2021), 101646.
- [6] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. 2020. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. *Advances in Neural Information Processing Systems* 33 (2020).
- [7] Chi Chen, Li Zhao, Jiang Bian, Chunxiao Xing, and Tie-Yan Liu. 2019. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2376–2384.
- [8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
- [9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016), 3844–3852.
- [10] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems* (Barcelona, Spain) (NIPS’16). Curran Associates Inc., Red Hook, NY, USA, 3844–3852.
- [11] Luca F Di Cerbo and Stephen Taylor. 2020. Graph theoretical representations of equity indices and their centrality measures. *Quantitative Finance* (2020), 1–15.
- [12] Zulong Diao, Xin Wang, Dafang Zhang, Yingru Liu, Kun Xie, and Shaoyao He. 2019. Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 890–897.
- [13] Matthew Elliott, Benjamin Golub, and Matthew O Jackson. 2014. Financial networks and contagion. *American Economic Review* 104, 10 (2014), 3115–53.
- [14] Eugene F Fama and Kenneth R French. 2015. A five-factor asset pricing model. *Journal of financial economics* 116, 1 (2015), 1–22.
- [15] Shen Fang, Qi Zhang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2019. GSTNet: Global Spatial-Temporal Network for Traffic Flow Prediction.. In *IJCAI*. 2286–2293.
- [16] Fulí Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–30.
- [17] Guanhao Feng, Stefano Giglio, and Dacheng Xiu. 2020. Taming the factor zoo: A test of new factors. *The Journal of Finance* 75, 3 (2020), 1327–1370.
- [18] Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2020. Autoencoder asset pricing models. *Journal of Econometrics* (2020).
- [19] Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2020. Empirical asset pricing via machine learning. *The Review of Financial Studies* 33, 5 (2020), 2223–2273.
- [20] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 922–929.
- [21] Bernard Herskovic. 2018. Networks in production: Asset pricing implications. *The Journal of Finance* 73, 4 (2018), 1785–1818.
- [22] Januj Juneja. 2012. Common factors, principal components analysis, and the term structure of interest rates. *International Review of Financial Analysis* 24 (2012), 48–56.
- [23] Bryan T Kelly, Seth Pruitt, and Yinan Su. 2019. Characteristics are covariances: A unified model of risk and return. *Journal of Financial Economics* 134, 3 (2019), 501–524.
- [24] 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).
- [25] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net.
- [26] Trung Hai Le, Hung Xuan Do, Duc Khuong Nguyen, and Ahmet Sensoy. 2020. Covid-19 pandemic and tail-dependency networks of financial assets. *Finance research letters* (2020), 101800.
- [27] Chang Li, Dongjin Song, and Dacheng Tao. 2019. Multi-task recurrent neural networks and higher-order Markov random fields for stock price movement prediction: Multi-task RNN and higher-order MRFs for stock price classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1141–1151.
- [28] Wei Li, Ruihan Bao, Keiichi Harimoto, Deli Chen, Jingjing Xu, and Qi Su. 2020. Modeling the Stock Relation with Graph Network for Overnight Stock Movement Prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). 4541–4547. <https://doi.org/10.24963/ijcai.2020/626>
- [29] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.
- [30] A Namaki, AH Shirazi, R Raei, and GR Jafari. 2011. Network analysis of a financial market based on genuine correlation and threshold method. *Physica A: Statistical Mechanics and its Applications* 390, 21–22 (2011), 3835–3841.
- [31] Ashadun Nobi, Seong Eun Maeng, Gyeong Gyun Ha, and Jae Woo Lee. 2014. Effects of global financial crisis on network structure in a local stock market. *Physica A: Statistical Mechanics and its Applications* 407 (2014), 135–143.
- [32] Boris N Oreshkin, Dmitrii Carpor, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*.
- [33] Adam Paszke and et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems* 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [34] Leonard CG Rogers and Luitgard AM Veraart. 2013. Failure and rescue in an interbank network. *Management Science* 59, 4 (2013), 882–898.
- [35] Samuel Rönnqvist and Peter Sarlin. 2015. Bank networks from text: interrelations, centrality and determinants. *Quantitative Finance* 15, 10 (2015), 1619–1635.
- [36] Mike KP So, Amanda MY Chu, and Thomas WC Chan. 2020. Impacts of the COVID-19 pandemic on financial market connectedness. *Finance Research Letters* (2020), 101864.
- [37] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27 (2014), 3104–3112.
- [38] Michele Tumminello, Fabrizio Lillo, and Rosario N Mantegna. 2010. Correlation, hierarchies, and networks in financial markets. *Journal of economic behavior & organization* 75, 1 (2010), 40–58.
- [39] Ajim Uddin and Dantong Yu. 2020. Latent factor model for asset pricing. *Journal of Behavioral and Experimental Finance* 27 (2020), 100353.
- [40] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
- [41] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
- [42] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 3634–3640.
- [43] Pengfei Yu and Xuesong Yan. 2020. Stock price prediction based on deep neural networks. *Neural Computing and Applications* 32, 6 (2020), 1609–1628.
- [44] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*.