Causal Inference based Graph Neural Network for Stock Market Prediction

Anonymous Author(s)*

ABSTRACT

Recent graph neural network(GNN)-based stock price prediction models have received extensive attention due to the mutual influence among stock prices, and have shown SOTA performances compared to traditional linear or time-series models. However, a vital issue is still under-explored, that is these works still stay at the stock price correlation level. Specifically, the weight of the aggregation step under the graph neural network framework only considers the correlation but not the causality. Thus for some pairs of stocks that do not have the real causal influence supported by business behavior such as contracts, the wrong link weights will be modeled, leading to wrong stock price forecasts. Therefore, we propose a causal inference and GNN based dual autoencoder network (DAN) to solve this problem. Specifically, we divide the observed datas into two sets and feed them into two autoencoder networks to separately learn the control and counterfactual outcome. Then the conditional average treatment effect (CATE) is regarded as the GNN weights to better model the impact between stock prices. We conduct extensive experiments to demonstrate that DAN achieves significant improvements over the state-of-the-art methods. Further analysis shows that DAN offers interpretability for the influence among stock prices.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Causal reasoning and diagnostics**; • **Mathematics of computing → Graph algorithms**.

KEYWORDS

Stock prediction, Causal inference, Graph neural network

ACM Reference Format:

Anonymous Author(s). 2018. Causal Inference based Graph Neural Network for Stock Market Prediction. In *Proceedings of The 17th ACM International Conference on Web Search and Data Mining (WSDM2024).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Stock prediction is one of the most attractive topics in the Fintech area [4], which motivates further exploration of stock prediction techniques to seek higher revenue [34]. In the industry, that is, hedge funds, stock price forecasting is a key step in multi-factor investing[10, 18]. Multi-factor stock quantitative investment usually consists of three parts: mining factors, factor combinations and portfolio optimization. This framework is widely used by famous hedge funds such as Millennium Management and Chinese LingJun Investment. Among these three steps, factor combinations aim to predict stock price with price-volume or fundamental factors as input, which have an important impact on portfolio return.

Early models adopts the idea of linear regression, and are divided into static (e.g. CAPM[19], FamaFrench three-factor model[12]) and dynamic models according to whether the factor exposure varies with time[17, 21]. However, they don't recognize nonlinear signals. In order to solve this shortcoming, some machine learning [27, 32] and deep learning [2, 9] methods improve price prediction performance from different aspects, such as using conditional autoencoder network to model the non-linearity in the return dynamics[11] , and using NLP to model the impact of news events on stock prices[31], and using RNN-based[39] or Transformer-based[26] models to predict stock price trends.

In recent years, graph nueral network (GNN) based models[5, 16, 25, 30, 40] concerning the relations between stocks have received a lot of attention. The core idea is to calculate a weight coefficient for a pair of companies to represent the influence of their stock prices. For example, stocks under the same sector or industry might have similar long-term trends. The stock of a supplier company might impact the stock of its consumer companies. These models adopt the following four-step schema. (1) Historical factors are fed into an RNN-based(e.g. GRU, LSTM) layer as features to output node (company) embeddings for each stock. The factors can be intra-daily or longer-term (e.g. exponential moving average) features.(2) An adjacency matrix is created through pre-prepared knowledge graph such as Wikidata, to connect companies through multi-kind relations like the same industry, the same shareholder, and one supply chain. (3) The node embeddings created by the first step and the adjacency matrix are combined and fed into a GNN layer to aggregate and update the node embeddings. Note that the weight of neighbor nodes aggregated to the central node is mostly based on the similarity of their latent vectors (e.g. dot product or cosine similarity). (4) The updated embeddings are fed into the final output layer, mostly a fully connected layer, to predict stock prices.

Although GNN-based studies achieve the state-of-the-art performances compared with the previous linear or nonlinear models, we point out that **they are still at the level of correlation research**, that is, the correlation of observed data is used to calculate the weight of edges, and the explainability and reasons behind it have not been explored from the perspective of causality. Therefore, this paper integrates the causal inference method into the current GNN-based model, aiming to discover the causality of stocks rising and falling in lockstep. Specifically, we compute the propagation weights of graph neural networks based on their causal effects rather than vector correlations. In order to follow the sechema of causal inference, we split the observed data and make customized

WSDM2024, March 04–08, 2024, Mérida, Yucatán

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

settings to build a weight calculation framework based on causal inference. Moreover, we design a dual autoencoder network to accurately learn the weights.

To sum up, the main objective of this paper is to explore the causality between stock prices and the interpretability behind the corresponding graph neural network, which is an improvement over previous work on correlation studies [16, 24, 35]. The key contributions of the paper are summarized as follows.

- We propose a causal inference framework to learn the effect between stock prices, which is an improvement from past correlation to causality studies.
- We propose a novel graph neural network based dual autoencoder network to compute causal inference based weights
- We conduct extensive experiments to demonstrate superiority of our model and further analysis shows the interpretability for the influence among stock prices.

2 PRELIMINARIES

2.1 Problem Setup

In this section, we formulate the problem with some notations.

For all S stocks, we have a set of historical sequence factor data X and returns y. A set of stock historical sequence data at day t is represented by $\mathbf{X_t} = \left\{\mathbf{X}_t^1, \cdots, \mathbf{X}_t^S\right\} \in \mathbb{R}^{S\times T\times K}$, where T is the length of time series and K is the dimension of factors. The return y_s^{t+1} is defined as the 1-day return ratio $\left(r_s^{t+1} - r_s^t\right)/r_s^t$ where r_s^t is the closing price at day t. For simplicity, we eliminate the superscript, which is t by default, a.k.a. $y_s = y_s^t$, unless we specifically notate it.

We also have an directed stock relation graph $G = \{V, E\}$, where V is a set of $|V| = S$ nodes representing stocks; $E = e_{ij}$ is a set of edges, with e_{ij} indicating the relation from stock i to j. To determine the relation between a pair of stocks, we predefine some connection rules. For example, two stocks that belong to the same sector or have a supply chain relationship connect an edge. For details on the graph construction, see the dataset description in Section **??**.

2.2 Past GNN-based methods

As mentioned in the introduction, this paper aims to improve the modeling of inter-stock effects from correlation to causality. Therefore, we first introduce the past GNN-based model for a better understanding.

Given the historical factors of stock s, $\mathbf{X}_s^t = \left\{\mathbf{x}_s^{t-T+1}, \cdots, \mathbf{x}_s^t\right\}$, we input it into the RNN-based networks. Here we use the GRU model as follows:

$$\mathbf{h}_s^i = \mathrm{GRU}\left(\mathbf{x}_s^i, \mathbf{h}_s^{i-1}\right) \quad t - T \le i \le t \tag{1}$$

where, $h_s^i \in \mathbb{R}^d$ is the last hidden state and d is the number of hidden units in GRU.

We use the embedding learned from the above sequential model as the initial representation of the graph node. Then the neighbor nodes are aggregated to the central node to generate the final representation followed by the prediction layer. Note that unlike graph models in other domains (e.g. the recommender system) which need multiple message passes (aggregations), graph models for stock forecasting mostly have only one layer. More analysises about GNN layers are shown in section 4.4

$$\overline{\mathbf{h}_s^t} = \sum_{j\in\mathcal{N}_s} k_{sj}\mathbf{h}_j^t \tag{2}$$

$$\mathbf{h}_s^{t,final} = f_{\mathrm{agg}}(\overline{\mathbf{h}_s^t}, \mathbf{h}_s^t) \tag{3}$$

$$\hat{y}^t = f_{\mathrm{pre}}(\mathbf{h}_s^{t,final}) \tag{4}$$

where $\mathcal{N}_s$ is the the neighbor nodes of stock s, $\overline{\mathbf{h}_s^t}$ encodes the impacts coming from other stocks that have relations with stock i at time t, k_{sj} represents the edge weight learned from different aggregation methods, f_{agg} usually adopts mean, sum, or concatenation operation, f_{pre} mostly is a feedforward neural network like MLP.

It is clear that the aggregation process, a.k.a. k_{sj}, is the key point, reflecting the innovation and contribution of one model. Table 1 lists some examples about k_{sj} and illustrates their ideas. We can see that the above design determines the influence between stocks based on the embedding similarity, that is, it still stays in the correlation study and the causality behind it is not explored. Sepecifically, each stock is influenced by multiple other stocks, while the direct calculation based on embedding similarity does not extract the influence of confounders, leading to that current study is at associational, a.k.a. correlation, level under the Pearl Causal Hierarchy (PCH) framework [28].

2.3 Base theory of causal inference

To make this paper self-contained, we first briefly introduce the base theory of causal inference including two parts as:

(1) **CATE.** The core target of causal inference is to to quantitatively measure how much a treatment W has a causal effect on the whole population. The common metric is Conditional Average Treatment Effect (CATE) as

$$\mathrm{CATE} = \mathbb{E}[\mathrm{Y}(W=1) \mid X = x] - \mathbb{E}[\mathrm{Y}(W=0) \mid X = x] \tag{5}$$

where X is the feature of the population and is sometimes called the confounder. $\mathrm{Y}(W=1)$ and $\mathrm{Y}(W=0)$ are two outputs given opposite treatments. Note that these two outcomes of one group cannot be observed simultaneously, since two treatments cannot be administered to the same member (also called unit [38]) at the same time. Usually, the unrealized outcome is called the counterfactual (a.k.a. potential treated) outcome. Therefore, we need to simulate one of them. Techinically, two learners are trained to recognize the mapping fuctions from X to Y in the control/treated group, which are denoted as $\mu_0(x) = \mathbb{E}[Y(W=0) \mid X = x]$ and $\mu_1(x) = \mathbb{E}[Y(W=1) \mid X = x]$ respectively. Note that despite each learned estimater is induced from part members, a.k.a one group, they can be used predict the outcome for whole population. The latent assumption is that all members in whole population are independent and identically distributed (i.i.d.) after we have consider all confounders.

(2) **SCM.** A structural causal model (SCM) is defined by [29] as $\mathfrak{C} := (\mathbf{S}, P(\mathbf{U}))$ where $P(\mathbf{U})$ is a product distribution over exogenous unmodelled variables $\mathbf{U}$ and $\mathbf{S}$ is defined to be a set of n structural equations

$$V_i := f_i\left(\mathrm{pa}\left(V_i\right), U_i\right), \quad \text{where } i = 1, \ldots, s \tag{6}$$

Table 1: Some examples about k_{sj}

Method	Formula	Illustration
RSR [16]	$\frac{1}{N_s}$	mean-pooling operation
	$\mathbf{h}_t^{s\mathrm{T}}\mathbf{h}_t^{j}$	dot product
	$\phi\left(\mathbf{w}^T\left[\mathbf{h}_t^{s\mathrm{T}},\mathbf{h}_t^{j\mathrm{T}}\right]^{\mathrm{T}}+b\right)$	MLP
MAN-SF [30]	$\frac{\exp\left(\text{LeakyReLU}\left(\mathbf{h}_t^{s\mathrm{T}}\left[\mathbf{W}\mathbf{h}_t^{s}\oplus\mathbf{W}\mathbf{h}_t^{j}\right]\right)\right)}{\sum_{k\in N_s}\exp\left(\text{LeakyReLU}\left(\mathbf{h}_t^{s\mathrm{T}}\left[\mathbf{W}\mathbf{h}_t^{s}\oplus\mathbf{W}\mathbf{h}_t^{k}\right]\right)\right)}$	MLP following softmax
GNN-Roll[25]	$\frac{\phi(\mathbf{W}^T\mathbb{A}_{sj}+b)}{d_s}$	static graph convolution
	$\mathbf{h}_s^T\mathbf{h}_j\times\frac{\phi(\mathbf{W}^T\mathbb{A}_{sj}+b)}{d_s}$	temporal graph convolution
TRAN [40]	$\frac{\exp\left(\mathbf{h}_t^{s\mathrm{T}}\phi\left(\mathbf{W}\mathbf{h}_t^{j}+b\right)\right)}{\sum_{k\in N_s}\exp\left(\mathbf{h}_t^{s\mathrm{T}}\phi\left(\mathbf{W}\mathbf{h}_t^{k}+b\right)\right)}$	MLP following softmax

$\mathbb{A}$ is the adjacecy matrix
d_s is the degree of node s acting as a normalization factor
$\mathbf{W}$ and b are learnable parameters
ϕ is a non-linear activation

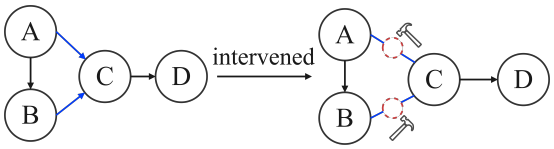

Figure 1: An example of intervention.

with pa (V_i) representing the parents of variable V_i in graph $G(\mathfrak{C})$ which contains s nodes. An intervention $do(\mathbf{W}), \mathbf{W}\subset\mathbf{V}$ on a SCM $\mathfrak{C}$ as defined in equation 6 occurs when (multiple) structural equations are being replaced through new non-parametric functions. The most common intervention is that some edges are removed, which means the intervened local neighborhood is the subset of the regular graph neighborhood shown in figure 1. An important property of interventions often referred to as "modularity" or "autonomy" states that interventions are fundamentally of local nature, and intervening on a variable V_i only changes the causal mechanism for V_i, leaving the other mechanisms invariant. Therefore, an important consequence of autonomy based on the the Markov condition of a causal directed acyclic graph (DAG) is the truncated factorization as

$$p(\boldsymbol{V}\mid do(\mathbf{W}))=\prod_{V\notin\mathbf{W}}p(V\mid \mathrm{pa}(V)) \tag{7}$$

For simplification, we use $p(\boldsymbol{V}\mid do(\mathbf{W}))$ and $p(\boldsymbol{V})$ to denote the interventional and original distribution in graph $G(\mathfrak{C})$, respectively.

[20] postulates a set of natural, intuitive requirements that a measure of causal influence should satisfy. Finally, they conclude that the KL-divergence is a suitable measure as

$$\mathrm{ATE}(\mathbf{W})=D_{\mathrm{KL}}\left[p(\boldsymbol{V})\|p(\boldsymbol{V}\mid do(\mathbf{W}))\right] \tag{8}$$

where we treat the intervention as a treatment and thus the average treatment effect of all intervened edges in graph $G(\mathfrak{C})$ are denoted as ATE($\mathbf{W}$).

3 METHODOLOGY

3.1 Overview

We model the causal inference based weight.

Given an edge e_{ij} in stock relation graph G, to learn the causal effect from stock i to j, we follow the idea of equation 5 as: (1) calcaute the price change of stock j motivated by two conditions that stock i goes up or down, (2) subtract these two values as the causal effect from stock i to j. However, we cannot observe two changes simultaneously since stock i has only one realized return. Therefore, we propose the following algorithm to tackle this issue:

1. We divide all the edges into two subsets according to the positive and negative of the source node as $E=\{E_1,E_2\}$, where $E_1=\{e_{ij},y_i\geq 0\}$ and $E_2=\{e_{ij},y_i\leq 0\}$.

2. We perform an intervention $\mathbf{W}_1$ that removes E_2 from the original graph. It is clear that the intervened graph $\mathfrak{C}_1$ only keeps the causal relations where source stocks go up. Then we learn the weighted adjacency matrix $A_1\in\mathbb{R}^{s\times s}$ of $\mathfrak{C}_1$.

3. Finally, an global shared function f_1 like a feed forward neural network is learned to map the feature $[\mathbf{X}_i||\mathbf{X}_j]$ of an edge e_{ij} to its weight a_{1ij}, the element in the i-th row and the j-th column of A_1.

4. Similar with the above steps, we perform another intervention $\mathbf{W}_2$ by removing E_1. The corresponding notations are $\mathfrak{C}_2$, A_2, and f_2.

5. We use f_1/f_2 to predict the price change when stock i increases/falls $\Delta\%$, and calculate the causal inference based weight via equation 5 as

$$\begin{aligned}k_{ij}&=\frac{(f_1([\mathbf{X}_i||\mathbf{X}_j])\cdot\Delta\%-f_2([\mathbf{X}_i||\mathbf{X}_j])\cdot(-\Delta\%))}{2\Delta\%}\\&=\frac{f_1([\mathbf{X}_i||\mathbf{X}_j])+f_2([\mathbf{X}_i||\mathbf{X}_j])}{2}\end{aligned} \tag{9}$$

Obviously, the key is how to learn the intervened adjacency matrix A_1 and A_2.

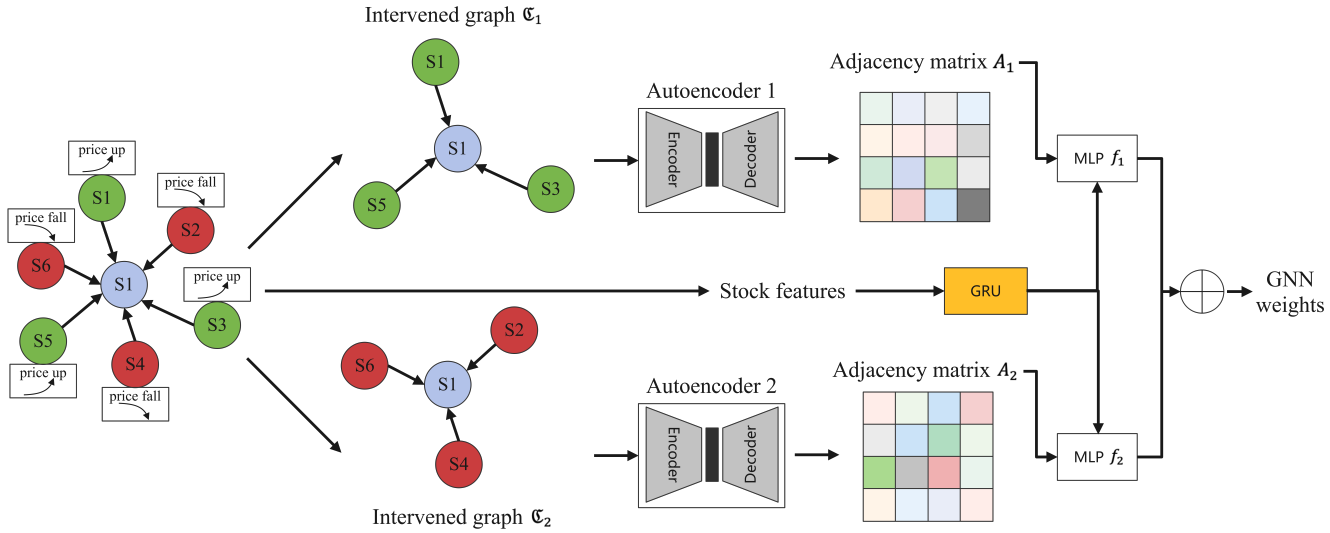

Figure 2: The framework of DAN.

3.2 Dual autoencoder based network

Motivated by the succees of the linear structural equation model (SEM) [45] and variational autoencoder [41] in DAG structure learning, we propose a dual autoencoder based network (DAN) to learn A_1 and A_2 jointly. Figure 2 shows the framework of DAN.

We build a variational autoencoder for each intervened graph. Since $\mathfrak{C}_1$ and $\mathfrak{C}_2$ are symmetric, we inllustrate $\mathfrak{C}_1$ as an example and use the subscript to distinguish the notations in these two graphs. **The core idea is using linear SEM to describe the causal effect among stock prices** as

$$Y_1 = A_1^T Y_1 + Z_1 \tag{10}$$

where $Z_1 \in \mathbb{R}^{s\times 1}$ is the noise matrix, mostly regarded as a random signal to generate Y_1 as

$$Y_1 = \left(I - A_1^T\right)^{-1} Z_1 \tag{11}$$

Equation 11 is a generative model, similar with the decoder of a variational autoencoder, motivating us to design the following model as

$$Z_1 = \left(I - A_1^T\right) f_{\text{en1}}(Y_1) \tag{12}$$

$$\hat{Y}_1 = \left(I - A_1^T\right)^{-1} f_{\text{de1}}(Z_1) \tag{13}$$

where equation 12 and 13 are the encoder and decoder, respectively. f_{en1} and f_{de1} are designed as two-layer MLP

$$f_{\text{en1}}(Y_1) = \text{ReLU}\left(Y_1 W_1^1\right) W_1^2 \tag{14}$$

$$f_{\text{de1}}(Z_1) = \text{ReLU}\left(Y_1 W_1^3\right) W_1^4 \tag{15}$$

where W_1^1, W_1^2, W_1^3, and W_1^4 are all learnable parameters.

The purpose of the variational autoencoder is to maximize the log-evidence $p(Y_1)$, that is maximizing the probability of the observed datas. Therefore, we first illustrate what distributions these variables belong to as the following three items and then give the loss function to maximize $p(Y_1)$ according to [**?**].

- Z_1 is a prior distribution typically modeled as the standard Gaussian distribution $p(Z_1) = \mathcal{N}(0, I)$.
- For the inference model, we use the variational posterior $q(Z_1 \mid Y_1)$ to approximate the actual posterior $p(Z_1 \mid Y_1)$. Sepcifically, $q(Z_1 \mid Y_1)$ is a factored Gaussian distribution with mean $M_{Z_1} \in \mathbb{R}^{s\times 1}$ and standard deviation $S_{Z_1} \in \mathbb{R}^{s\times 1}$, computed from the encoder 12 as

$$\left[M_{Z_1} \mid \log S_{Z_1}\right] = \left(I - A_1^T\right) f_{\text{en1}}(Y_1) \tag{16}$$

- For the generative model, $q(Y_1 \mid Z_1)$ is also a factored Gaussian distribution with mean $M_{Y_1} \in \mathbb{R}^{s\times 1}$ and standard deviation $S_{Y_1} \in \mathbb{R}^{s\times 1}$, computed from the decoder 13 as

$$\left[M_{Y_1} \mid \log S_{Y_1}\right] = \left(I - A_1^T\right)^{-1} f_{\text{de1}}(Z_1) \tag{17}$$

Based on the above distributions, the evidence lower bound (ELBO) based loss function [**?**] is designed as

$$L_{\text{ELBO1}} = -D_{\text{KL}}\left(q\left(Z_1 \mid Y_1\right) \| p(Z_1)\right) + \text{E}_{q(Z_1|Y_1)}\left[\log p\left(Y_1 \mid Z_1\right)\right] \tag{18}$$

[41] simplifies this loss to a simple but calculable result as

$$L_{\text{ELBO1}} = \frac{1}{2}\left(\|Y_1 - \hat{Y}_1\|_2 + \|Z_1\|_2\right) \tag{19}$$

where the first term is the reconstruction error, a closed form of $\text{E}_{q(Z_1|Y_1)}\left[\log p\left(Y_1 \mid Z_1\right)\right]$. The second term is a regularization of the latent space, a closed form of the KL-divergence term $-D_{\text{KL}}\left(q\left(Z_1 \mid Y_1\right) \| p(Z_1)\right)$.

We also have another variational autoencoder for $\mathfrak{C}_2$ with the loss function

$$L_{\text{ELBO2}} = \frac{1}{2}\left(\|Y_2 - \hat{Y}_2\|_2 + \|Z_2\|_2\right) \tag{20}$$

Note that Y_1 and Y_2 are both unobservable and have the following relations. According to our intervention rules, any edge in the original graph either belongs to graph $\mathfrak{C}_1$ or graph $\mathfrak{C}_2$. That is to say, for any node j, in the set of all parent nodes that affect its stock price $\{i, e_{ij} \in E\}$, the nodes whose prices are positive are put in $\mathfrak{C}_1$, while the nodes whose prices are negative are placed in $\mathfrak{C}_2$. What's

more, we use the linear SEM to fit the causal influence, where the effects from nodes in $\{i, e_{ij} \in E\}$ to node j accumulate linearly into the final stock price of node j. Therefore, we have the following conclusion as

$$Y_1 + Y_2 = Y \tag{21}$$

Therefore, we revise the reconstruction error as

$$\begin{aligned} L_{\text{ELBO}} =& L_{\text{ELBO1}} + L_{\text{ELBO2}} \\ \approx& \frac{1}{2}\left(\|Y - (\hat{Y}_1 + \hat{Y}_2)\|_2 + \|Z_1\|_2 + \|Z_2\|_2\right) \end{aligned} \tag{22}$$

What's more, we propose an ATE loss. Below we take $\mathfrak{C}_1$ as an example. As mentioned in equation 8, the KL-divergence is proportional to the causal effect of an intervention as

$$\begin{aligned} \text{ATE}(\mathbf{W}_1) =& D_{\text{KL}}\left[p(Y) \| p(Y \mid do(\mathbf{W}_1))\right] \\ =& D_{\text{KL}}\left[p(Y) \| p(Y_1)\right] \end{aligned} \tag{23}$$

where $\text{ATE}(\mathbf{W}_1)$ is the causal effect of removing all edges where source nodes price are negtive. Therefore, its influence is close to the average weight of these removed edges, a.k.a. the average of A_2, as

$$\text{ATE}(\mathbf{W}_1) = \frac{\|A_2\|_1}{|E_2|} \tag{24}$$

As mentioned in equation 17, the output of decoder is a Gaussian distribution, leading to $p(Y) = p(Y_1) + p(Y_2) = \mathcal{N}(M_{Y_1} + M_{Y_2}, S_{Y_1} + S_{Y_2})$. Therefore, the KL-divergence of two Gaussian distributions is a analytical solution as

$$\begin{aligned} D_{\text{KL}}\left[p(Y) \| p(Y_1)\right] =& D_{\text{KL}}\left[\mathcal{N}(M_{Y_1} + M_{Y_2}, S_{Y_1} + S_{Y_2}) \| \mathcal{N}(M_{Y_1}, S_{Y_1})\right] \\ =& \sum_{i=1}^{s} \log\left(\frac{(S_{Y_1} + S_{Y_2})_i}{(S_{Y_1})_i}\right) - \frac{1}{2} \\ &+ \frac{(S_{Y_1} + S_{Y_2})^2 + ((M_{Y_1} + M_{Y_2}) - M_{Y_1})^2}{2(S_{Y_1})^2} \\ =& \sum_{i=1}^{s} \log\left(\frac{(S_{Y_1} + S_{Y_2})_i}{(S_{Y_1})_i}\right) - \frac{1}{2} + \frac{(S_{Y_1} + S_{Y_2})_i^2 + (M_{Y_2})_i^2}{2(S_{Y_1})_i^2} \end{aligned} \tag{25}$$

Based on equation 24 and 25, we have

$$\frac{\|A_2\|_1}{|E_2|} = \sum_{i=1}^{s} \log\left(\frac{(S_{Y_1} + S_{Y_2})_i}{(S_{Y_1})_i}\right) - \frac{1}{2} + \frac{(S_{Y_1} + S_{Y_2})_i^2 + (M_{Y_2})_i^2}{2(S_{Y_1})_i^2} \tag{26}$$

It has a dual conclusion for $\mathfrak{C}_2$ as

$$\frac{\|A_1\|_1}{|E_1|} = \sum_{i=1}^{s} \log\left(\frac{(S_{Y_1} + S_{Y_2})_i}{(S_{Y_2})_i}\right) - \frac{1}{2} + \frac{(S_{Y_1} + S_{Y_2})_i^2 + (M_{Y_1})_i^2}{2(S_{Y_2})_i^2} \tag{27}$$

Therefore, the final ATE loss is designed as

$$\begin{aligned} L_{\text{ATE}} =& \left\| \frac{\|A_2\|_1}{|E_2|} - \sum_{i=1}^{s} \log\left(\frac{(S_{Y_1} + S_{Y_2})_i}{(S_{Y_1})_i}\right) - \frac{1}{2} + \frac{(S_{Y_1} + S_{Y_2})_i^2 + (M_{Y_2})_i^2}{2(S_{Y_1})_i^2} \right\|_2 \\ &+ \left\| \frac{\|A_1\|_1}{|E_1|} - \sum_{i=1}^{s} \log\left(\frac{(S_{Y_1} + S_{Y_2})_i}{(S_{Y_2})_i}\right) - \frac{1}{2} + \frac{(S_{Y_1} + S_{Y_2})_i^2 + (M_{Y_1})_i^2}{2(S_{Y_2})_i^2} \right\|_2 \end{aligned} \tag{28}$$

Table 2: Statistics of two datasets.

Datasets		NASDAQ	NYSE
Stock Historical Sequence	#Training Days	747	747
	#Validation Days	249	249
	#Testing Days	249	249
Stock Relation Graph	#Nodes	1026	1737
	#Edges	52586	98065
Stock Description Document	#Descriptions	5130	5955
	#Words	319297	427932

3.3 Model Optimization

As aforementioned, we first jointly learn two adjacency matrixes A_1 and A_2 with the loss fuction as

$$L = L_{\text{ELBO}} + L_{\text{ATE}} \tag{29}$$

Finally two map functions f_1 and f_2 are trained by the Mean Squared Error (MSE) loss.

For online services, we calculate the weight via equation 9 once after the market closes on day t, and then predict the return on day $t+1$ via equation 2, 3 and 4.

4 EXPERIMENTS

4.1 Experimental Settings

We evaluate our method to answer the following research questions:

- **RQ1**: Does our proposed DAN outperform the state-of-the-art methods?
- **RQ2**: Can STKGN provide potential explanations about the motivation of this paper, that is the improvement from the correlation to causality?
- **RQ3**: How do the critical hyperparameter (i.e. GNN layer numbers, time length of feature) affect DAN?

Dataset Setup. We adopt the stocks from NASDAQ and NYSE markets that have historical sequences between 01/02/2013 and 12/08/2017, including 1,026 and 1,737 stocks, respectively. The entire dataset is divided into the training set, validation set and testing set via the rate 0.6/0.2/0.2. We extract 112 and 130 types of industry relations from the company classification hierarchy structure of NASDAQ and NYSE stocks, respectively. For more information on industries and stocks, please refer to the appendix section 7.1. We construct stock relation graph by connecting the edge for a pair of stocks which belong to the same industry of the same sector. What's more, some baselines (e.g. MAN-SF, TRAN and MAGNN) need to use side informations, we introduce the corresponding documents to construct the auxiliary datas and achieve the best potential of their original models. The final statistics of two datasets are shown in Table 2. All datas are publicly available in [16].

Baselines. To demonstrate the effectiveness, we compare DAN with two kinds of methods: time-series models (DFM [42], WSAEs-LSTM [3]) and GNN-based models (GCN [23], GAT [33], RSR [16], MAN-SF [30], GNN-Roll [25], TRAN [40], GC-GNN [5]). See more descriptions in section 7.2.

Table 3: Performance comparison. (e-4±e-7) means that the first number is based on e-4 and the second number is based on e-7. For example, "5.20±577" means "5.20e-4±577e-7". The best performance is boldfaced. The runner up is labeled with '*'

Methods	NASDAQ			NYSE		
	MSE(e-4±e-7)	MRR(e-2±e-3)	IRR	MSE(e-4±e-7)	MRR(e-2±e-3)	IRR
SFM	5.20±577	2.33±10.7	-0.25±0.52	3.81±930	4.82±4.95	0.49±0.47
WSAEs-LSTM	3.81±22.0	3.64±1.04	0.13±0.62	2.31±14.3	2.75±1.09	-0.90±0.73
GCN	3.80±22.4	3.45±8.36	0.24±0.32	2.27±1.30	5.01±5.56	0.97±0.56
GAT	3.79±13.7	3.61±5.63	0.30±0.42	2.26±1.18*	4.68±5.18	1.25±0.74
RSR-E	3.82±26.9	3.16±3.45	0.20±0.22	2.29±27.7	4.28±6.18	1.00±0.58
RSR-I	3.80±7.90	3.17±5.09	0.23±0.27	2.26±5.30	4.51±2.41	1.06±0.27
MAN-SF	3.82±19.7	3.55±6.17	0.37±0.42	2.33±7.74	4.77±7.99	1.21±0.80
MAN-SF-pure	3.83±18.2	3.51±7.14	0.32±0.37	2.36±9.12	4.71±7.22	1.09±0.69
GNN-Roll	3.78±5.11*	3.76±5.05	0.97±0.24*	2.28±1.77	5.02±4.99*	1.29±0.62
TRAN	3.79±3.90	3.81±4.37*	0.92±0.25	2.26±2.30	4.91±4.82	1.38±0.85*
TRAN-pure	3.81±2.37	3.74±3.12	0.85±0.11	2.29±1.96	4.72±3.14	1.13±0.72
GC-GNN	3.84±25.1	3.66±4.10	0.74±0.37	2.32±16.6	4.58±6.13	0.84±0.17
MAGNN	3.79±10.0	3.69±6.89	0.89±0.23	2.30±9.74	4.59±5.43	0.91±0.18
MAGNN-pure	3.84±6.67	3.55±3.01	0.84±0.16	2.27±10.67	4.31±4.90	0.79±0.23
DAN	**3.68±4.31**	**4.19±3.02**	**1.33±0.18**	**2.16±1.79**	**5.21±3.66**	**1.62±0.31**

Parameter Settings. We implement our DAN model in Pytorch and Deep Graph Library (DGL)[1]. More details about reproduction are in the section 7.3.

Evaluation Metrics. We employ three common used metrics, Mean Square Error (MSE), Mean Reciprocal Rank (MRR), and the cumulative investment return ratio (IRR). They represent three different and concerned aspects, which are regression, alpha[2] and return. Smaller value of MSE (≥ 0) and larger value of MRR ($[0, 1]$) and IRR indicate better performance.

4.2 Performance Comparision (RQ1)

We report the empirical results in Table 3. The observations are as follows:

- DAN consistently achieves the best performance on two datasets in terms of all measures. Specifically, it achieves significant improvements over the strongest baselines w.r.t. IRR by 0.36 and 0.24 in NASDAQ. The results prove the effectiveness of DAN.
- MAN-SF/TRAN/MAGNN outperforms MAN-SF-pure/TRAN-pure/MAGNN-pure, proving that the auxiliary textual information (description document) reports the petential influence caused by emerging events and brings improvement for prediction.
- Considering relations is more useful on NYSE as compared to NASDAQ. It could be attributed to that the industry relations reflect more of long-term correlations between stocks, since NASDAQ is considered as a much more volatile market as compared to NYSE and dominated by short-term factors.
- Jointly analyzing the methods of learning k_{ij} across all GNN-based baselines, correlation based methods (e.g. GAT, GNN-Roll, TRAN, etc.) have close performs while causality based method (DAN) significantly outperforms them. The reason is that DAN explores the causality behind observed datas and learns more accurate weights between stocks.
- Stock relation based models outperforms time series based models (SFM and WSAEs-LSTM), which proves that there is a phenomenon of rising and falling at the same time in the market and modeling relational signals helps to improve alpha, a.k.a. excess returns.

4.3 Case Study (RQ2)

This section aims to visualize the weights to illustrate the improvement brought about by causal inference compared to past correlation study. We choose sector *Water Supply* as the example. For more business information in this sector, please refer to the appendix section 7.4. We fix a time window from 4/27/2016 to 6/30/2016 and Figure 5 gives their returns based on the price on 4/27/2016. Since all stocks had significant price movements on 6/27/2016, we calculated the weights on this day shown in Table 4 to check whether different models could account for this phenomenon. Specifically, PRIM/CWCO/MYRG/AEGN share a similar recent trend, but PRIM is down on 4/27/2016 and the others are up. While ARTNA/CTWS/MSEX/YORW have an opposite recent trend but a similar movement on 4/27/2016 with PRIM. Therefore, we take the stock PRIM as the center node to check the weights with other companies and Figure 3 shows the topology.

At the correlation level, all baselines generate correlation-based weights, which have the similar distribution. Specifically, they uniformly present that PRIM has higher weights with AEGN/CWCO/MYRG and lower weights with ARTNA/CTWS/MSEX/YORW. What's more, we give their information coefficients (IC[3]) based on prices before 6/27/2016 in Figure 4. We have an important finding that **the**

[1]https://github.com/dmlc/dgl

[2]Alpha is a term used in investing to describe an investment strategy's ability to beat the market, indicating when a strategy, trader, or portfolio manager has managed to beat the market return or other benchmark over some period. Alpha is thus also often referred to as "excess return" relation to a benchmark.

[3]IC is the most widely used measure in hedge found to evaluate the skill of an investment analyst or an active portfolio manager. IC shows how closely the analyst's stock forecasts match actual stock results. The IC can range from 1.0 to -1.0, with -1 indicating the analyst's forecasts bear no relation to the actual results, and 1 indicating that the analyst's forecasts perfectly matched actual results.

ICs calculated based on past observed datas (prices) and the weights calculated by these baselines based on correlation study have an approximate distribution, like the ICs between PRIM and AEGN/CWCO/ MYRG are higher than that between PRIM and ARTNA/CTWS/MSEX/YORW. It proves that past works are still at correlation level. However, PRIM has the opposite movement with AEGN/CWCO/ MYRG on 4/27/2016, which means that higher weights with AEGN/CWCO/ MYRG and lower weights with ARTNA/CTWS/MSEX/YORW cannot explain this phenomenon and do not describe the mutual causal relationship behind stock prices.

At the causality level, our porposed model learns different weights. Specifically, MYRG has significantly higher weights with ARTNA and MSEX compared to baselines, which effectively explains why MYRG fall on 4/27/2016 because ARTNA and MSEX also fall on this day. What's more, we explore the causal reasons behind it. The reason is that they all focus on wastewater treatment, so the market trend of the same business makes their stock prices interact with each other. This case study demonstrates the improvement brought about by causal inference compared to past correlation study.

We also have an additional finding that the weight learned with auxiliary information is closer to DAN, like MAN-SF/TRAN/MAGNN outperform MAN-SF-pure/TRAN-pure/MAGNN-pure.The reason is that the news data contains the company's main business and implicitly models this fundamental knowledge and causal relationship.

Table 4: Weights learned by different models

Methods	p-AE	p-AR	p-CT	p-CW	p-MS	p-MY	p-Y
GAT	0.21	0.02	0.03	0.17	0.05	0.46	0.06
RSR-E	0.32	0.05	0.06	0.25	0.02	0.22	0.08
RSR-I	0.28	0.06	0.02	0.31	0.05	0.21	0.07
MAN-SF	0.18	0.12	0.01	0.36	0.06	0.16	0.11
MAN-SF-pure	0.19	0.08	0.01	0.4	0.03	0.21	0.08
GNN-Roll	0.29	0.03	0.06	0.15	0.09	0.33	0.05
TRAN	0.19	0.12	0.04	0.20	0.14	0.24	0.07
TRAN-pure	0.23	0.11	0.02	0.22	0.13	0.25	0.05
GC-GNN	0.21	0.09	0.04	0.17	0.09	0.36	0.04
MAGNN	0.21	0.09	0.07	0.23	0.12	0.24	0.04
MAGNN-pure	0.22	0.07	0.05	0.32	0.1	0.20	0.03
DAN	0.21	0.18	0.01	0.22	0.19	0.17	0.02

4.4 Study of DAN (RQ3)

Impact of GNN layer numbers. The number of GNN layers affects the scope of the local update. As mentioned in sector 2.2, most graph models for stock forecasting mostly have only one layer. Here we search the layer in the range of $\{1, 2, 3\}$ for all GNN-based baselines to explore its effect. Figure 6 gives the MSE performances in NASDAQ and we omit other metrics since they have the same conclusion. It is clear that stacking more layers decrease the performance of baselines because of the noise caused by multiple message transmissions, that is, the information of more than 2-hop on the graph has not been correctly aggregated on the central node. However, our model performs best after two layers of aggregation. The reason is the weights are more in line with the real causality, so

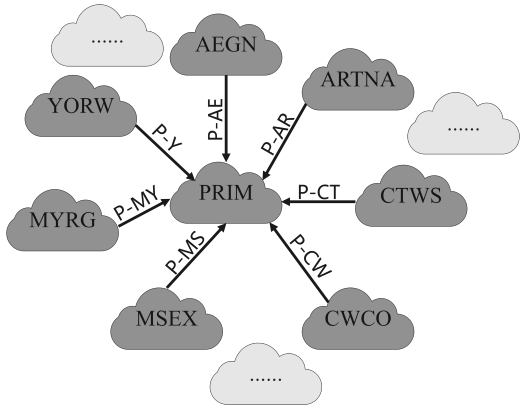

Figure 3: Subgraph.

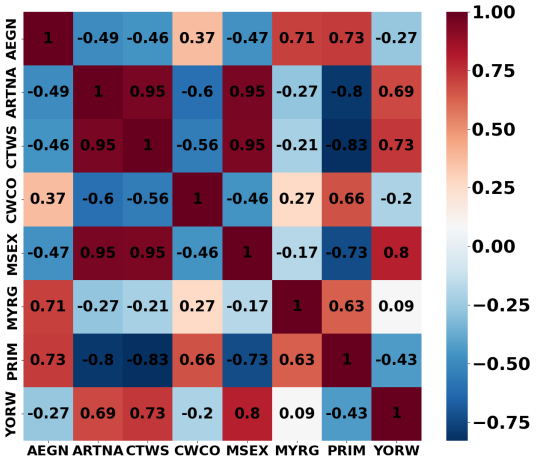

Figure 4: IC heatmap.

spreading twice will further collect more useful information in a larger receptive field, which proves the superiority of causal-based weights.

Impact of factor time length. As mentioned in equation 1, using GRU to recognize the pattern of historical sequence factos is the first step and the time length of factors affects the final performance. Here we search T in the range of $\{3, 5, 10, 15, 20, 30\}$ days. Figure 7 gives the MSE performances in NASDAQ. It is obvious that baselines get the best performances when using 10-day datas while DAN can use more(15-day) past factors. The reason is that our fitting target is one-day return, a relatively short-term return, so too long historical data means that some mid-to-long-term fundamental information has no effect on short-term trends. Therfore 10-day data is sufficient for these baselines. But our model explores the causality between stock prices, which is a relatively stable relationship. For example, if two companies have commercial contracts, their stock prices will be affected by this causal relationship for a long time. Therefore,

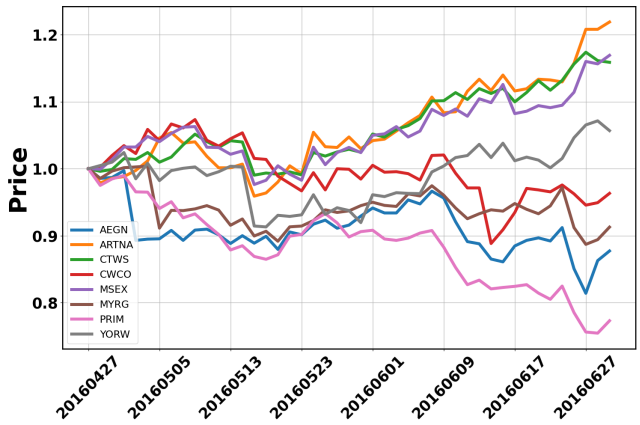

Figure 5: Stock price.

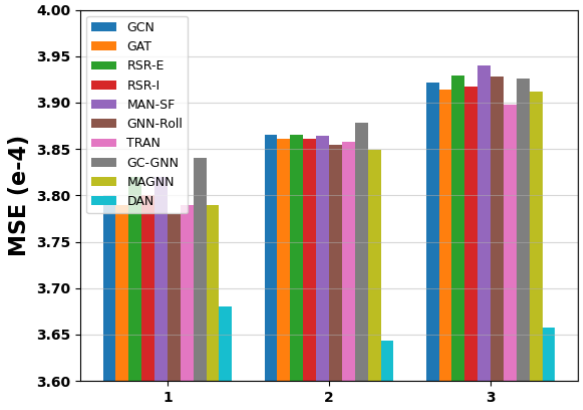

Figure 6: Impact of GNN layer number.

our model research, which is based on causal weights, can use more historical data to obtain more accurate weights and better performances. What's more, we leave how much time window data is needed for the longer-term return as our future work.

5 RELATED WORK

Our work is highly related to four subareas of stock prediction.

Multi-factor model. Factor models explain market phenomena and asset returns by various factors[8, 15], which can be fundamental, technical, macroeconomic, and so on. According to whether the factor exposure varies with time, factor models fall under two categories: static models[12, 19] and dynamic models[17, 21]. Recent research[14] indicates that dynamic models with time-varying factor exposure achieve better asset pricing performance than static methods, so dynamic models become increasingly popular.

Time-series model. Traditional solutions for stock prediction are based on time-series analysis models, such as Kalman Filtersc [37], Autoregressive Models and their extensions [1]. Given an indicator of a stock (e.g., stock price), this kind of models represents it as

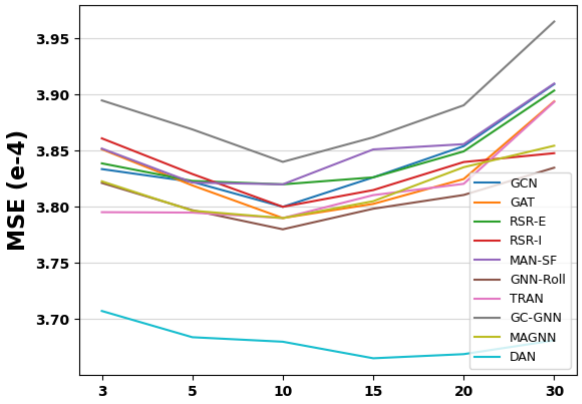

Figure 7: Impact of critical hyperparameters.

a stochastic process and takes the historical data of the indicator to fit the process. Due to the success of deep learning, advanced techniques like RNN have become a promising solution to substitute the traditional time-series models to predict the future trend or exact price of a stock [3, 42–44].

Event-based model. During recent years, many studies on predicting stock returns based on textual side information (events and news)[9] have emerged. They believe that the financial market is "informationally efficient"[13] and the price movement is in response to news or events. As web information grows, recent work has applied Natural Language Processing (NLP) techniques to explore financial news for predicting market volatility. They incorporates the fine-grained new events[18], social media information[36] and a financial knowledge graph[6] based on raw news texts into stock movement prediction.

GNN-based model. Recently, there have been interesting works [16, 22] which incorporate open source knowledge graphs (e.g.Wikidata) into their stock prediction models for individual companies by combining graph neural networks and sequential models such as Recurrent Neural Networks (RNN). They believe that there is mutual influence between stocks, such as upstream and downstream supply chains, companies with economic cooperation, and companies that compete with each other. Therefore, using graph neural network to model the connection between companies is conducive to more accurate stock price prediction[5, 16, 25, 30, 40].

6 CONCLUSIONS

This paper explores the causality between stock prices, which is an improvement of the traditional model in the correlation study of observed data. Our key contribution is to propose a causal inference-based link weight calculation algorithm for GNNs, using a dual autoencoder. Furthermore, we provide causal explanations behind some market phenomenas that violate correlation. Extensive experiments demonstrate the superiority of our model. Future work will focus on how to explore the relationship between causal inference-based weights and long-term returns.

REFERENCES

[1] Ayodele Ariyo Adebiyi, Aderemi Oluyinka Adewumi, Charles Korede Ayo, et al. 2014. Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics* 2014 (2014).

[2] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara. 2016. Deep learning for stock prediction using numerical and textual information. In *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*. IEEE, 1–6.

[3] Wei Bao, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one* 12, 7 (2017), e0180944.

[4] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of computational science* 2, 1 (2011), 1–8.

[5] Wei Chen, Manrui Jiang, Wei-Guo Zhang, and Zhensong Chen. 2021. A novel graph convolutional feature based convolutional neural network for stock trend prediction. *Information Sciences* 556 (2021), 67–94.

[6] Dawei Cheng, Fangzhou Yang, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. 2020. Knowledge graph-based event embedding framework for financial quantitative investments. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2221–2230.

[7] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. 2022. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition* 121 (2022), 108218.

[8] Kent Daniel, David Hirshleifer, and Lin Sun. 2020. Short-and long-horizon behavioral factors. *The review of financial studies* 33, 4 (2020), 1673–1736.

[9] Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2015. Deep learning for event-driven stock prediction. In *Twenty-fourth international joint conference on artificial intelligence*.

[10] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang. 2017. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance* 6, 3-4 (2017), 67–77.

[11] Yitong Duan, Lei Wang, Qizhong Zhang, and Jian Li. 2022. Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 4468–4476.

[12] Robert Faff. 2001. An examination of the Fama and French three-factor model using commercially available factors. *Australian Journal of Management* 26, 1 (2001), 1–17.

[13] Eugene F Fama. 1965. The behavior of stock-market prices. *The journal of Business* 38, 1 (1965), 34–105.

[14] Eugene F Fama and Kenneth R French. 2020. Comparing cross-section and time-series factor models. *The Review of Financial Studies* 33, 5 (2020), 1891–1926.

[15] Eugene F Fama and Kenneth R French. 2021. Multifactor explanations of asset pricing anomalies. *University of Chicago Press* (2021).

[16] Fuli 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] Shihao Gu, Bryan Kelly, and Dacheng Xiu. 2021. Autoencoder asset pricing models. *Journal of Econometrics* 222, 1 (2021), 429–450.

[18] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 261–269.

[19] Ravi Jagannathan, Ellen R McGrattan, et al. 1995. The CAPM debate. *Federal Reserve Bank of Minneapolis Quarterly Review* 19, 4 (1995), 2–17.

[20] Dominik Janzing, David Balduzzi, Moritz Grosse-Wentrup, and Bernhard Schölkopf. 2013. Quantifying causal influences. (2013).

[21] 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.

[22] 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).

[23] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

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

[25] Daiki Matsunaga, Toyotaro Suzumura, and Toshihiro Takahashi. 2019. Exploring graph neural networks for stock market predictions with rolling window analysis. *arXiv preprint arXiv:1909.10660* (2019).

[26] Tashreef Muhammad, Anika Bintee Aftab, Muhammad Ibrahim, Md Mainul Ahsan, Maishameem Meherin Muhu, Shahidul Islam Khan, and Mohammad Shafiul Alam. 2023. Transformer-based deep learning model for stock price prediction: A case study on Bangladesh stock market. *International Journal of Computational Intelligence and Applications* (2023), 2350013.

[27] Ramkrishna Patel, Vikas Choudhary, Deepika Saxena, and Ashutosh Kumar Singh. 2021. Review of stock prediction using machine learning techniques. In *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 840–846.

[28] Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress* 19, 2 (2000), 3.

[29] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

[30] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. 2020. Deep attentive learning for stock movement prediction from social media text and company correlations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8415–8426.

[31] Dev Shah, Haruna Isah, and Farhana Zulkernine. 2018. Predicting the effects of news sentiments on the stock market. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 4705–4708.

[32] Troy J Strader, John J Rozycki, Thomas H Root, and Yu-Hsiang John Huang. 2020. Machine learning stock market prediction studies: review and research directions. *Journal of International Technology and Information Management* 28, 4 (2020), 63–83.

[33] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[34] Huizhe Wu, Wei Zhang, Weiwei Shen, and Jun Wang. 2018. Hybrid deep sequential modeling for social text-driven stock prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1627–1630.

[35] Cong Xu, Huiling Huang, Xiaoting Ying, Jianliang Gao, Zhao Li, Peng Zhang, Jie Xiao, Jiarun Zhang, and Jiangjian Luo. 2022. HGNN: Hierarchical graph neural network for predicting the classification of price-limit-hitting stocks. *Information Sciences* 607 (2022), 783–798.

[36] Yumo Xu and Shay B Cohen. 2018. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1970–1979.

[37] Xu Yan and Zhang Guosheng. 2015. Application of kalman filter in the prediction of stock price. In *5th International Symposium on Knowledge Acquisition and Modeling (KAM 2015)*. Atlantis press, 197–198.

[38] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 5 (2021), 1–46.

[39] Yimian Yao. 2022. Data analysis on the computer intelligent stock prediction model based on LSTM RNN and algorithm optimization. In *2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*. IEEE, 480–485.

[40] Xiaoting Ying, Cong Xu, Jianliang Gao, Jianxin Wang, and Zhao Li. 2020. Time-aware graph relational attention network for stock recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2281–2284.

[41] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*. PMLR, 7154–7163.

[42] 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*. 2141–2149.

[43] Yao Zhang, Yun Xiong, Xiangnan Kong, and Yangyong Zhu. 2017. Learning node embeddings in interaction graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 397–406.

[44] Sendong Zhao, Quan Wang, Sean Massung, Bing Qin, Ting Liu, Bin Wang, and ChengXiang Zhai. 2017. Constructing and embedding abstract event causality networks from text snippets. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 335–344.

[45] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. 2018. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems* 31 (2018).

7 APPENDIX

7.1 Datasets

Table 5 shows part relations from the NASDAQ market.

7.2 Baselines

- **SFM** [42] decomposes the hidden states of memory cells into multiple frequency components, each of which models a particular frequency of latent trading pattern. Then the future stock prices

Table 5: Part relations among 1,026 stocks from the NASDAQ.

Sectors	Industries
Transportation	Transportation Services Air Freight/Delivery Services Trucking Freight/Courier Services Oil Refining/Marketing
Finance	Specialty Insurers, Commercial Banks Savings Institutions, Specialty Insurers Commercial Banks, Savings Institutions Real Estate, Major Banks Investment Managers, Investment Bankers/Brokers/Service, Life Insurance Finance: Consumer, Services, Banks Property-Casualty Insurers, Finance Companies
Energy	Electric Utilities: Central, Coal Mining Oil & Gas Production

are predicted as a nonlinear mapping of the combination of these components in an Inverse Fourier Transform (IFT) fashion.

- **WSAEs-LSTM** [3] This method is the vanilla LSTM, which first generates the denoised time series via discrete wavelet transform and then extracts the deep daily features via stacked autoencoders (SAEs), followed by a LSTM layer to predict the one-step-ahead output.
- **GCN** [23] GCN is the classical graph-based learning method. We replace equation 2 and 3 with a GCN layer.
- **GAT** [33] GAT enables (implicitly) specifying different weights to different nodes in a neighborhood, which means placing k_{sj} with the attention coefficient.
- **RSR** [16] Following the past GNN-based methods, RSR proposes two methods to calculate k_{sj}. RSR-E uses dot product as the explicit modeling while RSR-I adopts MLP as the implicit modeling.
- **MAN-SF** [30] It introduces an architecture to model the stock movements influenced by social media and correlations among stocks in a hierarchical temporal fashion. For a fair comparasion, we adopt two versions of MAN-SF. One is the orginal model and uses the social media data, called MAN-SF. The other drops the social media information and just keeps the temporal attention based price encoder, called MAN-SF-pure.
- **GNN-Roll** [25] It designs a novel backtesting schema using rolling window analysis to predict the movement of a certain company's stock price based on the performance of its suppliers or customers. We compare with two variants, termed GNN-Roll-s/t, which adopts the static/temporal graph convolution.
- **TRAN** [40] It captures time-varying correlation strength between stocks by the interaction of historical sequences and stock description documents for stock recommendation. Similar with MAN-SF, according to whether use the auxiliary textual information, we use two versions termed as MAN-SF and MAN-SF-pure.
- **GC-GNN** [5] It proposes a graph convolutional feature based convolutional neural network (GC−CNN) model to predict stock trend by combining IGCN and Dual-CNN, in which the stock market features and individual stock features are merged into joint features.
- **MAGNN** [7] It proposes a multi-modality graph neural network to learn from these multimodal inputs for financial time series prediction. Similar with MAN-SF and TRAN, for a fair comparasion, we use two versions termed as MAGNN and MAGNN-pure. MAGNN-pure does not use information other than price.

7.3 Reproducibility

For MAN-SF and TRAN, we can directly use the stock description documents as the side information. For MAGNN, three modalities (historical prices, a heterogeneous graph and medial news) are needed. The former two datas are also used by our DAN model while the last one comes from the description documents. For SFM, the number of frequencies is set as 20 for best performance. For GAT and MAN-SF, we use 8 attention heads. For TRAN, the topic numbers of NASDAQ and NYSE are 50 and 60, respectively. For MAGNN, the price modality just use the close price for a fair comparison. The length of sequential input and the embedding size of all models are set to be 10 and 64 for a fair comparison. We adopt Adam as the optimizer and the batch size is fixed at 1024 for all methods. We apply a grid search for hyper-parameters: the learning rate and the coefficient of L2 normalization are searched in $\{10^{-5}, 10^{-4}, \cdots, 10^{-1}\}$, and the dropout ratio is tuned in $\{0.0, 0.1, \cdots, 0.9\}$.

7.4 Water supply

Table 6 lists all stocks and their business activities in Water supply sector.

Table 6: All stocks in sector water supply.

Stock(Company)	Business
AEGN(Aegion Corp)	industry infrastructure wastewater service
ARTNA(Artesian Resources)	water supply wastewater service sewer line protection
CTWS(Connecticut Water Service)	water supply
CWCO(Consolidated Water Co Ltd)	potable water supply retail water water-related products water infrastructure
MSEX(Middlesex Water)	water supply wastewater utility
MYRG(MYR Group)	electrical infrastructure including waste-water treatment facilities
PRIM(Primoris Services Corporation)	pipelines for natural gas wastewater and water
YORW(The York Water Company)	water supply wastewater service