Exploring the Scale-Free Nature of Stock Markets: Hyperbolic Graph Learning for Algorithmic Trading

Ramit Sawhney*
ramits@iiitd.ac.in
IIIT Delhi
India

Shivam Agarwal*
shivamag99@gmail.com
Manipal Institute of Technology
India

Arnav Wadhwa
arnavw96@gmail.com
MIDAS, IIIT Delhi
India

Rajiv Ratn Shah
rajivratn@iiitd.ac.in
IIIT Delhi
India

ABSTRACT

Quantitative trading and investment decision making are intricate financial tasks in the ever-increasing sixty trillion dollars global stock market. Despite advances in stock forecasting, a limitation of most existing neural methods is that they treat stocks independent of each other, ignoring the valuable rich signals between related stocks' movements. Motivated by financial literature that shows stock markets and inter-stock correlations show scale-free network characteristics, we leverage domain knowledge on the Web to model inter-stock relations as a graph in four major global stock markets and formulate stock selection as a scale-free graph-based learning to rank problem. To capture the scale-free spatial and temporal dependencies in stock prices, we propose HyperStockGAT: Hyperbolic Stock Graph Attention Network, the first model on the Riemannian Manifolds for stock selection. Our work's key novelty is the proposal of modeling the complex, scale-free nature of inter-stock relations through temporal hyperbolic graph learning on Riemannian manifolds that can represent the spatial correlations between stocks more accurately. Through extensive experiments on long-term real-world data spanning over six years on four of the world's biggest markets: NASDAQ, NYSE, TSE, and China exchanges, we show that HyperStockGAT significantly outperforms state-of-the-art stock forecasting methods in terms of profitability by over 12%, and risk-adjusted Sharpe Ratio by over 4%. We analyze HyperStockGAT's components' contributions through a series of exploratory and ablative experiments to demonstrate its practical applicability to real-world trading. Furthermore, we propose a novel hyperbolic architecture that can be applied across various spatiotemporal problems on the Web's commonly occurring scale-free networks.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; • **Social and professional topics** → *Economic impact*.

*Equal contribution.

WWW 2021, April 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450095

KEYWORDS

stock market, hyperbolic learning, graph neural network, finance

ACM Reference Format:
Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn Shah. 2021. Exploring the Scale-Free Nature of Stock Markets: Hyperbolic Graph Learning for Algorithmic Trading. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3450095

1 INTRODUCTION

The stock market, a financial ecosystem involving transactions between businesses and investors, observed a market capitalization exceeding *$60 trillion* globally as of the year 2019.[1] Stock trading presents opportunities that increasingly attract traders and investors to utilize the market as a platform for investing and forecasting risk to maximize profits. However, making the right investment decisions and designing profitable trading strategies has many challenges due to the market's highly volatile and non-stationary nature [1]. Recent advances in deep learning present a promising prospect in stock forecasting by analyzing online news [25], earnings calls [48], and social media [62] to learn latent patterns that affect stock prices [33]. However, most existing neural works treat stock movements as independent of each other, which is contrary to true market function [17], and makes forecasting challenging.

In reality, stocks are often related to each other, and there exist rich signals in relationships between stocks (or companies) [46]. Recent advances in graph-based deep learning [61] have led to the rise of graph neural networks (GNNs) that can model the relationships between related stocks. Publicly available company information from the Web can be used to identify connections between stocks that might influence other stocks' prices, such as those having the same CEO or belonging to the same industry. These connections can be leveraged to represent inter-stock relations as graphs, where nodes represent stocks and connections define the edges. Some recent studies show the effectiveness of modeling stock interdependence for improving the predictive power of neural stock forecasting networks [20, 36]. However, these methods do not account for the *scale-free* nature [3] in stock markets, where the stock graphs may exhibit strong node-degree heterogeneity, and the node degrees follow a power-law distribution.

[1]https://data.worldbank.org/indicator/CM.MKT.LCAP.CD/

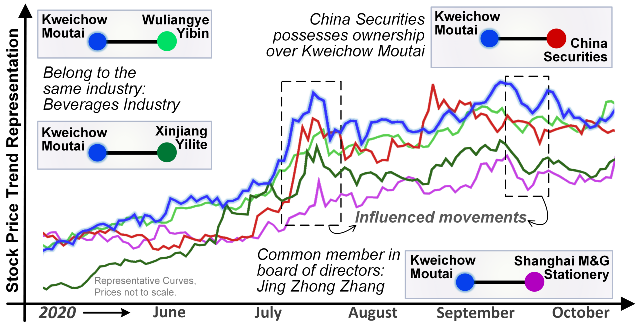

Figure 1: Movements in price of highly connected, influential stocks impact the prices of related stocks. Here, we show how movements in Kweichow Moutai impact related stocks.

The scale-free nature in stock markets arises due to the presence of a few highly connected influential companies, also called as hubs in the graph [3]. The price fluctuations in these usually large hubs impact the movement of other related stocks [6]. For instance, consider the effects of movement and fluctuations in stock prices of Kweichow Moutai, a highly influential China A-shares stock [26], over related stocks, as shown in Figure 1. Existing works [20, 36] that adopt graph-based methods to model stocks do not factor the hierarchical [43] and asymmetric nature of inter-stock correlations. Further, commonly used graph-based methods such as graph convolution networks (GCNs) [37] represent scale-free graphs in Euclidean space, which leads to high distortion in node embeddings [14, 49]. In contrast, typical properties of the scale-free graphs such as the long tail degree distribution and the presence of highly connected nodes (stocks) generalize well in *hyperbolic* space [51, 52]. The recent advances in graph representation learning in hyperbolic space [13, 40], and the ability of the hyperbolic space to better model real-world scale-free graphs together provide a promising unexplored research avenue for financial applications.

Building on the limitations of existing research and accounting for the complex, scale-free nature of the stock market, we propose HyperStockGAT: **H**yperbolic **S**tock **G**raph **A**ttention **N**etwork, a neural architecture (**Sec.4**) in the Riemannian manifold that jointly extracts spatial and temporal features from inter-stock correlations and historical prices for stock selection. We formulate stock prediction as a learning to rank problem (**Sec.4.1**) and directly optimize HyperStockGAT towards ranking stocks in terms of profits. HyperStockGAT models the temporal evolution of stock prices using attentive temporal convolutions (**Sec.4.2**) and learns the synergy between related stocks' movements using attentive spatial hyperbolic graph convolutions (**Sec.4.3**). Through trading simulations on real-world data of four of the world's biggest markets (**Sec.5.1**), we show that HyperStockGAT significantly outperforms state-of-the-art methods in terms of profit: return, risk-adjusted return: *Sharpe Ratio*, and ranking metrics (**Sec.6**). We perform an ablation study (**Sec.6.1**) and conduct exploratory analyses (**Sec.6.2, Sec.6.4**), to contextualize each component's effectiveness for stock selection and pave the future directions towards more spatiotemporal problems involving scale-free networks (**Sec.7**).

The contributions of our work can be summarized as:

- We explore the scale-free nature of stock markets and propose to leverage the hyperbolic space instead of the commonly used Euclidean space to better represent scale-free stock graphs.
- We present a novel Attentive Spatio-Temporal Hyperbolic Graph Convolution Network optimized for stock selection that combines temporal convolutions with spatial hyperbolic graph convolutions in an end-to-end fashion to capture correlations in the movements of related stocks and the temporal evolution of historic stock features.
- Through experiments on real-world data of 2,943 stocks belonging to NASDAQ, NYSE, TSE and China exchanges spanning over six years, we show that HyperStockGAT significantly outperforms state-of-the-art stock forecasting methods in terms of profitability by over 12%, risk-adjusted returns by 4%, and demonstrate HyperStockGAT's practical applicability to quantitative trading.

2 RELATED WORK

Conventional Methods: Stock prediction spans numerous methods, commonly formulated as either classification or regression problems [33]. Predicting the prices or directional trends in stock movements finds practical applications that include designing investment strategies [8], portfolio management [23], etc. To predict stock movements, financial models conventionally focus on technical analysis (TA) and rely only on numeric features like historical prices [44, 60] and macroeconomic indicators like GDP [24]. These TA methods include discrete [9], and neural approaches [44]. Newer models based on Efficient Market Hypothesis (EMH), categorized under fundamental analysis (FA) [16], account for stock affecting factors beyond numerical ones such as investor sentiment through news [25], social media [53], earnings calls [48], etc. Despite their success, a limitation of existing TA and FA methods is that they assume stock movements to be independent of each other. This assumption hinders the model's ability to learn latent patterns for modeling price movements of interrelated stocks. Another major limitation is that prior works are not directly optimized for maximizing profit as they do not explicitly select the top stocks with the highest expected revenue.

Contemporary Methods: Another newer line of work revolves around the use of graph-based methods to represent relations between stocks using metadata, such as stock-industry data, and links between company CEOs [20, 39]. For instance, Kim et al. [36] propose an attention-based graph neural network for stock movement prediction. They show that all stocks are not equally correlated, and equal treatment of stock relations may increase the noise in graphs, thereby negatively affecting the predictive performance. Feng et al. [20] augment graph convolution networks (GCNs) with temporal convolutions and demonstrate the utility of augmenting temporal stock price evolution methods with inter-stock relations. Despite these advancements, a limitation in existing graph-based methods is that they ignore the scale-free nature of stock graphs [35] and represent these graphs in the Euclidean space, which leads to distortions in their embeddings [51, 52].

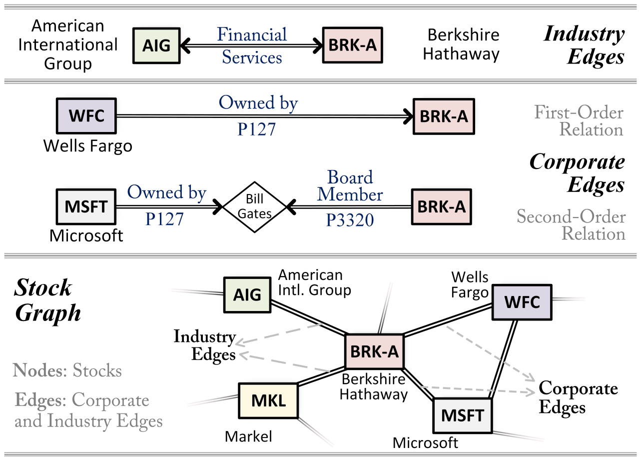

Figure 2: Stock graph construction using stocks as nodes, and corporate and industry relations as edges.

Hyperbolic Representation Learning: Hyperbolic geometry has been proved to be effective in representing information where relation among data points are scale-free in nature [2]. Learning in the hyperbolic space has been applied to various natural language processing [22, 57] and computer vision tasks [34, 47], owing to its ability to better represent hierarchical and asymmetric relationships. Recently, shallow hyperbolic embedding methods such as Poincaré embeddings [45] have been developed for encoding hyperbolic properties of graphs by learning via hyperbolic distance metric and Riemannian optimization for reconstructing trees [52]. However, the Poincaré embedding methods do not leverage rich node features, which can be crucial for tasks such as node classification, and they scale poorly as the number of model parameters grows linearly with the number of nodes. Further, Chami et al. [13], propose hyperbolic graph convolution networks that perform graph convolutions with attention in hyperbolic space to obtain low distortion embeddings of scale-free and hierarchical graphs. A gap in existing works on hyperbolic graph neural networks is that they are not specifically designed for temporally learning from time-evolving features such as daily stock prices.

3 STOCK NETWORKS

3.1 Stock Graph Construction

We model stock interdependence using graphs, where the nodes of the graph represent the stocks, and the edges of the graph represent the relations between the stocks. We introduce domain knowledge between stocks into the network based on two types of relations: corporate relations and sector-industry relations.

Corporate Edges: Following Feng et al. [20], we make use of Wiki corporate relations. Using Wikidata,[2] we extract first-order and second-order relations between the stocks. As shown in Figure 2 (middle), a first-order relation is defined as $X \xrightarrow{\text{R1}} Y$ where X and Y denote entities in Wikidata that correspond to the two stocks. A second-order relation is defined by $X \xrightarrow{\text{R2}} Z \xleftarrow{\text{R3}} Y$ where Z denotes another entity connecting the two entities X and Y. R1, R2, and R3, defined in Wikidata, are different types of entity-relations. For instance, Wells Fargo and Bank of America are related to Berkshire Hathaway via a first-order corporate relation "owned by." Another example is Microsoft and Berkshire Hathaway that are related through Bill Gates (second-order relation: "owned by" - "is a board member of") since Bill Gates possesses ownership over Microsoft and is a Board member of Berkshire Hathaway. We make use of a total of 57 first-order and second-order relations as defined by Feng et al. [20].

Industry Edges: Stocks that belong to the same industry, experience similar price movement trends based on the industry's performance [42]. To leverage this signal, we define sector-industry relations between stocks as per the Global Industry Classification Standard.[3] As shown in Figure 2 (top), a sector-industry relation is defined for stocks belonging to common sector-industry groupings as $X \xleftrightarrow{\text{SI}} Y$ where X and Y denote the stocks and SI denotes the common sector-industry grouping they belong to.

Based on these relations we define the stock relation network as a graph $\mathcal{G}(\mathcal{S}, \mathcal{E})$ where $\mathcal{S}$ denotes the set of nodes, and $\mathcal{E}$ is the set of edges. As shown in Figure 2 (bottom), each node $s \in \mathcal{S}$ represents a stock, and two stocks $s_1, s_2 \in \mathcal{S}$ are joined by an edge $e \in \mathcal{E}$ if s_1, s_2 are linked by a first-order, second-order, or a sector-industry relation.

3.2 Scale-free Nature of Stock Graphs

Financial studies suggest that stock graphs are often skewed [3, 43] and characterized by presence of large hubs [6]. These hubs have a very high degree of connectivity with other stocks and are hence highly influential stocks [27]. For instance, one of the commonly occurring hubs are conglomerate companies [18][4] which tends to be less affected by industry-specific factors and more affected by global market behaviours. Another instance of commonly occurring hubs are energy stocks [63], which often influence other businesses like technology stocks [38]. Intuitively, these influential companies are generally large-capitalization stocks [35] and share a high degree of correlation with other stocks [59]. This also suggests that the behaviour of important "influential" nodes in the graph might be more predictive of fluctuations in the market price of other underlying stocks [50]. Such correlation between stocks leads to the emergence of scale-free nature in stock markets [56] where, the highly influential nodes are hubs in the stock graph.

The scale-free structure of stock graphs manifests itself as a power-law distribution on the node degrees, at least asymptotically [5]. That is, the fraction $P(k)$ of nodes in the graph having k connections to other nodes follows the distribution,

$$P(k) \sim k^{-\gamma} \tag{1}$$

where, γ is the exponent. Furthermore, scale-free graphs show tree-like properties which are well represented in the hyperbolic space [51, 52]. We compute Gromovs δ hyperbolicity [21][5], that measures how tree-like a graph is. The lower δ, the more hyperbolic

[2] https://www.wikidata.org/wiki/Wikidata:List_of_properties/all

[3] wikipedia.org/wiki/Global_Industry_Classification_Standard

[4] A conglomerate company is a combination of more firms operating in different business sectors.

[5] The implementation of hyperbolicity estimation can be found in: https://github.com/HazyResearch/hgcn.

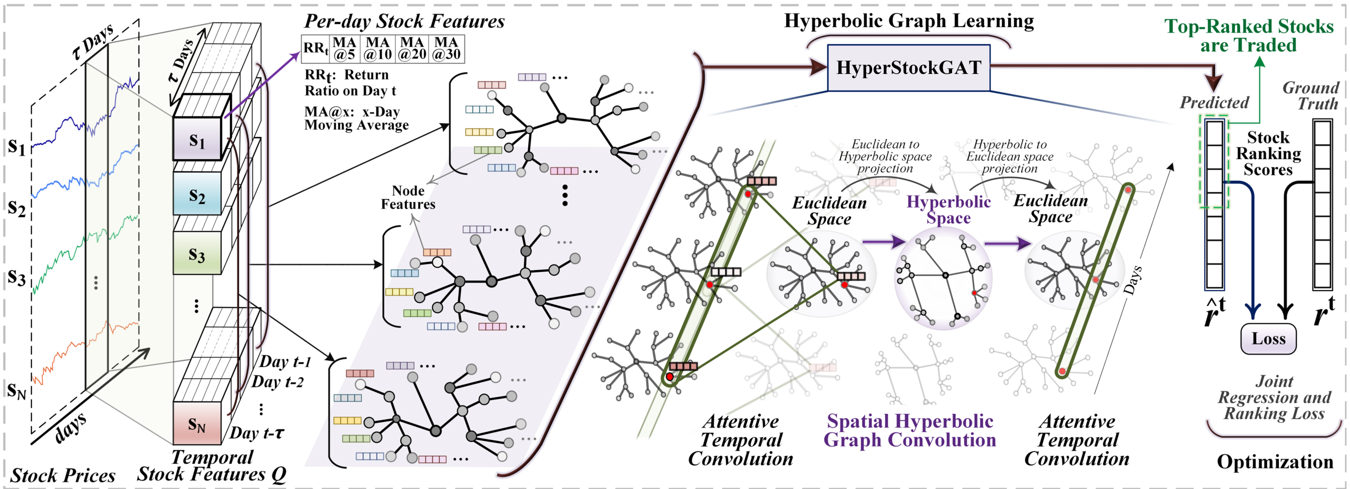

Figure 3: A high-level overview of HyperStockGAT, feature extraction, spatiotemporal hyperbolic learning, and optimization

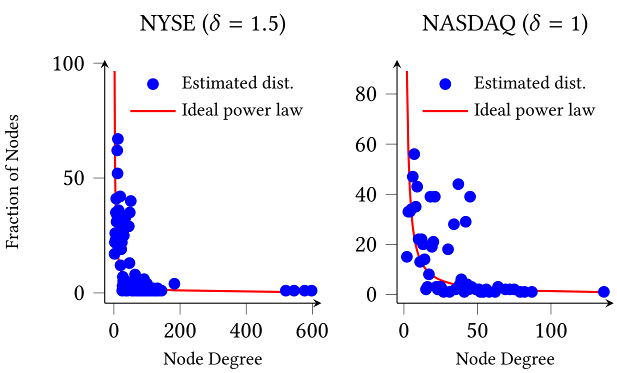

Figure 4: Node (stock) degree frequency distribution.

is the stock graph. The δ hyperbolcity of the graph $\mathcal{G}$ is calculated as follows. Let $a, b, c, d \in \mathcal{S}$ be four vertices of the graph $\mathcal{G}$, let S_1, S_2, S_3 be defined by,

$$S_1 = \text{dist}(a, b) + \text{dist}(d, c) \tag{2}$$

$$S_2 = \text{dist}(a, c) + \text{dist}(b, d) \tag{3}$$

$$S_3 = \text{dist}(a, d) + \text{dist}(b, c) \tag{4}$$

where, $\text{dist}(x, y)$ denotes the distance between two nodes x, y in the graph $\mathcal{G}$. Let M_1 and M_2 be the two largest values among S_1, S_2, and S_3. We define $hyp(a, b, c, d) = M_1 - M_2$, and the hyperbolicity δ of the graph $\mathcal{G}$ is the maximum of $hyp(a, b, c, d)$ over all possible 4-tuples (a, b, c, d) divided by two. That is,

$$\delta = \frac{1}{2} \max_{a,b,c,d \in \mathcal{S}} hyp(a, b, c, d) \tag{5}$$

We study four of the world's biggest and well-studied stock markets: NYSE, NASDAQ, TSE and China exchanges. These markets are highly traded and span the S&P 500 index, Dow Jones Industrial Average, TOPIX 100 index and China A-Shares. We find the degree distribution and hyperbolicity of NYSE and NASDAQ markets in Figure 4. Both TSE and China exchanges have similar degree distributions as compared to NYSE and NASDAQ with hyperbolicity 0. We observe that the degree distribution of the stock graphs follow a power-law distribution and have low δ hyperbolicities, indicating the presence of hubs in these graphs. TSE and China exchanges are more hyperbolic as compared to NYSE and NASDAQ since they have lower δ values. These observations collectively tie up with other financial studies [3, 29], which suggest that stock markets have a few highly influential stocks and show scale-free nature. We now present HyperStockGAT, which is defined on the Hyperbolic space that can better represent scale-free stock graphs.

4 METHODOLOGY

4.1 Problem Setting

We formulate stock prediction as a *learning to rank* problem. Let $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$ denote the set of N stocks, where for each stock $s_i \in S$ on trading day t, there is an associated closing price p_i^t and a 1-day return ratio $r_i^t = \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$. On any given trading day t, there exists an optimal ranking $Y^t = \{y_1^t > y_2^t \cdots > y_N^t\}$ of the stocks, such that there exists a total order between the ranks $y_i^t > y_j^t$ for any two stocks $s_i, s_j \in \mathcal{S}$, if $r_i^t > r_j^t$. Such an ordering of stocks $\mathcal{S}$ on a trading day t represents a ranking list, where stocks achieving higher ranking scores Y are expected to achieve a higher investment revenue (profit) on day t. Formally, given stock data for a historical lookback window of length τ (i.e., $[t - \tau, t - 1]$), we aim to learn a ranking function that outputs a score y_i^t to rank each stock s_i on day t in terms of expected profit.

We present an overview of our proposed HyperStockGAT in Figure 3. In the following subsections, we first explain how we extract temporal features from the evolution of historical stock prices using an attentive temporal convolution (**Sec.4.2**). We then describe how we aggregate stock (node) features that are related to each other using hyperbolic graph convolutions and attention over the constructed inter-stock graphs (**Sec.4.3**). We then show

how we combine the temporal and spatial hyperbolic graph convolutions components (**Sec.4.4**) to capture the temporal evolution of stock features over scale-free networks in the hyperbolic space. Finally, we optimize HyperStockGAT to rank stocks in terms of their expected profitability based on the learned spatiotemporal stock features (**Sec.4.5**).

4.2 Attentive Temporal Convolution

Historical stock prices have shown to be a strong indicator of future stock trends [44], and widely used across financial literature [20, 36]. Further, studies have shown that the stock features of each day has a different impact on future prices [25]. To this end, we employ a temporal attention mechanism [31] which learns to weigh critical days that impact the future price prediction. Thus, we use attentive temporal convolutions to capture the temporal dependencies in stock features over time. We first describe a single attentive temporal convolution layer here, which is used throughout the HyperStockGAT framework. The attentive temporal convolution consists of an attention mechanism followed by a temporal convolution. We feed stock features $\boldsymbol{X}^l \in \mathbb{R}^{N\times C\times\tau}$ corresponding to all N stocks for a historical lookback period τ to the l^{th} attentive temporal convolution layer. Here, C is the number of features per stock. This attention mechanism learns a weight matrix $\boldsymbol{E}$, such that the value of each element $E_{i,j}$ of $\boldsymbol{E}$ signifies the strength of the temporal dependency between stock features of days i and j. We then normalize temporal attention matrix $\boldsymbol{E}$ as $\boldsymbol{E'} = \text{softmax}(\boldsymbol{E})$ to make the attention coefficients comparable across different days. We define this mechanism as:

$$\boldsymbol{E} = \boldsymbol{V} \cdot \sigma\left((\boldsymbol{X}^{l^T}\boldsymbol{U}_1)\boldsymbol{U}_2(\boldsymbol{U}_3\boldsymbol{X}^l) + \boldsymbol{b}\right) \tag{6}$$

$$E'_{i,j} = \frac{\exp(E_{ij})}{\sum_{j=1}^{T}\exp(E_{i,j})} \tag{7}$$

where, $\boldsymbol{V}, \boldsymbol{b} \in \mathbb{R}^{\tau\times\tau}, \boldsymbol{U}_1 \in \mathbb{R}^N, \boldsymbol{U}_2 \in \mathbb{R}^{C\times N}, \boldsymbol{U}_3 \in \mathbb{R}^C$ are learnable parameters and, we use logistic sigmoid as the activation function. We apply the normalized temporal attention matrix $\boldsymbol{E'}$ to the input $\boldsymbol{X}^l$ and learn an adaptive feature vector $\boldsymbol{X}^l_{\text{attn}} \in \mathbb{R}^{N\times C\times\tau}$ that learns to weigh features from more critical days selectively. Formally,

$$\boldsymbol{X}^l_{\text{attn}} = \boldsymbol{X}^l\boldsymbol{E}' = (\boldsymbol{X}^l_1, \boldsymbol{X}^l_2, \cdots, \boldsymbol{X}^l_\tau)\boldsymbol{E}' \tag{8}$$

We then use a 1-dimensional *causal* convolution operation $*$ with a learnable kernel $\boldsymbol{\phi}^l$ on the attentive features $\boldsymbol{X}^l_{\text{attn}}$ to extract the temporal dependencies within these features. We use causal convolutions to prevent information "leakage" from future features to the past [15, 64]. The 1D causal convolution along the temporal dimension updates the features of each stock by merging features from neighbouring time steps. We devise an attentive temporal convolution operator $*_{\text{attn}}$ which updates stock features $\boldsymbol{X}^l$ to feature vector $\boldsymbol{X}^{l+1} \in \mathbb{R}^{N\times C\times\tau}$ via the temporal attention mechanism followed by a standard temporal convolution operation $*$. Formally, the update rule of this attentive temporal convolution $*_{\text{attn}}$ is given by:

$$\boldsymbol{X}^{l+1} = \boldsymbol{\phi}^l *_{\text{attn}} \boldsymbol{X}^l = \text{ReLU}(\boldsymbol{\phi}^l * \boldsymbol{X}^l_{\text{attn}}) = \text{ReLU}(\boldsymbol{\phi}^l * (\boldsymbol{X}^l\boldsymbol{E}')) \tag{9}$$

where, the activation function is ReLU, $\boldsymbol{\phi}^l$ denotes the learnable parameters of the temporal convolution kernel.

4.3 Spatial Hyperbolic Graph Convolution

To capture the dependencies in the price movements of related stocks, we employ graph convolutions over the inter-stock relation graph $\mathcal{G}$. Building on the scale-free nature of inter-stock relation graph $\mathcal{G}$ (**Sec.3.2**), we propose the use of graph convolutions in the hyperbolic space as opposed to the standard graph convolutions [37] in the euclidean space. We leverage the hyperbolic space as the graph volume grows exponentially lending more robustness and expressiveness to stock (node) embeddings [40], as opposed to the polynomial growth of the graph volume in the euclidean space which leads to high distortion node embeddings [51, 52].

Here, we describe an attentive hyperbolic graph convolution, which is a generalization of the graph convolution in hyperbolic space that benefits from the expressiveness of both graph neural networks and hyperbolic embeddings [13]. First, we formally describe the hyperboloid model for the hyperbolic space. Let $\langle .,.\rangle_{\mathcal{L}} : \mathbb{R}^{C+1}\times\mathbb{R}^{C+1} \to \mathbb{R}$ denote the Minkowski inner product, $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}} = -x_0y_0 + x_1y_1 \cdots x_Cy_C$ [12]. Following [13][6], we denote $\mathbb{H}^{C,K}$ as the hyperboloid manifold in C dimensions with a constant negative curvature $-1/K, (K > 0)$ defined as:

$$\mathbb{H}^{C,K} = \{\boldsymbol{x} \in \mathbb{R}^{C+1} : \langle \boldsymbol{x}, \boldsymbol{x}\rangle_{\mathcal{L}} = -K, x_0 > 0\} \tag{10}$$

We denote the tangent space (Euclidean) centered at point $\boldsymbol{x}$ as $\mathcal{T}_{\boldsymbol{x}}\mathbb{H}^{C,K}$. A tangent space defined as the local, first-order linear approximation of the hyperbolic manifold $\mathbb{H}^{C,K}$ at $\boldsymbol{x}$, is given by:

$$\mathcal{T}_{\boldsymbol{x}}\mathbb{H}^{C,K} = \{\boldsymbol{v} \in \mathbb{R}^{C+1} : \langle \boldsymbol{v}, \boldsymbol{x}\rangle_{\mathcal{L}} = 0\} \tag{11}$$

Additionally, we form a bijection between the hyperbolic space and the tangent space at a point via exponential and logarithmic mappings [13, 40]. Formally, given a point $\boldsymbol{x} \in \mathbb{H}^{C,K}$ and a tangent vector $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathbb{H}^{C,K}(\boldsymbol{v} \neq 0)$, the exponential mapping $\exp^K_{\boldsymbol{x}} : \mathcal{T}_{\boldsymbol{x}}\mathbb{H}^{C,K} \to \mathbb{H}^{C,K}$, maps the tangent vector $\boldsymbol{v}$ to the point $\exp^K_{\boldsymbol{x}}(\boldsymbol{v})$ on the hyperboloid manifold as:

$$\exp^K_{\boldsymbol{x}}(\boldsymbol{v}) = \cosh\left(\frac{||\boldsymbol{v}||_{\mathcal{L}}}{\sqrt{K}}\right)\boldsymbol{x} + \sqrt{K}\sinh\left(\frac{||\boldsymbol{v}||_{\mathcal{L}}}{\sqrt{K}}\right)\frac{\boldsymbol{v}}{||\boldsymbol{v}||_{\mathcal{L}}} \tag{12}$$

where, $||\boldsymbol{v}||_{\mathcal{L}} = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v}\rangle_{\mathcal{L}}}$ is the norm of $\boldsymbol{v} \in \mathcal{T}_{\boldsymbol{x}}\mathbb{H}^{C,K}$. Similarly, the logarithmic mapping $\log^K_{\boldsymbol{x}} : \mathbb{H}^{C,K} \to \mathcal{T}_{\boldsymbol{x}}\mathbb{H}^{C,K}$ is the reverse map of the exponential mapping that maps a point $\boldsymbol{y} \in \mathbb{H}^{C,K}(\boldsymbol{y} \neq \boldsymbol{x})$ back to a point $\log^K_{\boldsymbol{x}}(\boldsymbol{y})$ on the tangent space at $\boldsymbol{x}$, given by:

$$\log^K_{\boldsymbol{x}}(\boldsymbol{y}) = \sqrt{K}\text{arcosh}\left(\frac{-\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}}}{K}\right)\frac{\boldsymbol{y} + \frac{1}{K}\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}}\boldsymbol{x}}{||\boldsymbol{y} + \frac{1}{K}\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\mathcal{L}}\boldsymbol{x}||_{\mathcal{L}}} \tag{13}$$

Building on these mappings between tangent space and hyperbolic spaces, we now describe graph convolutions on the hyperboloid manifold. The input to the attentive hyperbolic convolution is a set of Euclidean node features (stock) $\boldsymbol{P} \in \mathbb{R}^{N\times C}$, where C is the number of features per stock. Since, the input node features $\boldsymbol{P}$ are in the Euclidean space, we first map node features to the hyperboloid manifold via the exponential mapping to apply hyperbolic graph convolutions. Given Euclidean node features $\boldsymbol{x}^E_i \in \mathbb{R}^C$ of a single stock $s_i \in \mathcal{S}$ we map it to the hyperbolic features $\boldsymbol{x}^H_i \in \mathbb{H}^{C,K}$ (superscript E and H denote Euclidean and hyperbolic features, respectively). Let $\boldsymbol{o} = \{\sqrt{K}, 0, \cdots, 0\} \in \mathbb{H}^{C,K}$ denote the origin of

[6]We use the Poincarè ball model for our implementation as found in: https://github.com/HazyResearch/hgcn.

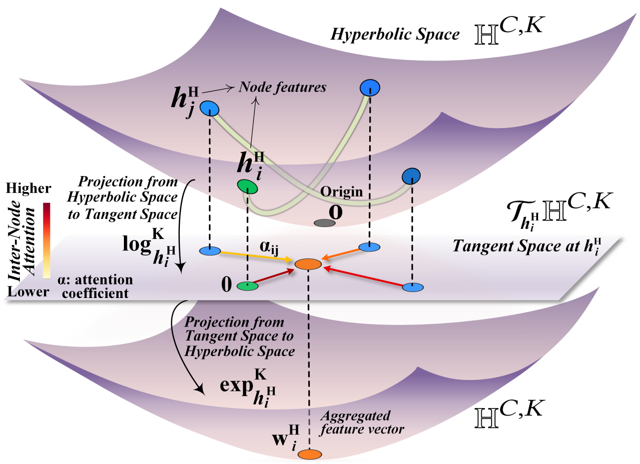

Figure 5: HyperStockGAT's spatial attentive neighborhood aggregation. Node features are first mapped to the tangent space where node feature aggregation is performed. The updated features are mapped back to the hyperbolic space.

the hyperboloid manifold $\mathbb{H}^{C,K}$, which we use as a reference point to perform tangent space operations. Following [13], we consider $(0, \boldsymbol{x}_i^E)$ as a point in the tangent space at the origin $\mathcal{T}_{\boldsymbol{o}}\mathbb{H}^{C,K}$ and map it to the hyperboloid manifold using the exponential map as:

$$\boldsymbol{x}_i^H = \exp_{\boldsymbol{o}}^K((0, \boldsymbol{x}_i^E)) = \left(\sqrt{K}cosh\left(\frac{||\boldsymbol{x}_i^E||_2}{\sqrt{K}}\right), \sqrt{K}sinh\left(\frac{||\boldsymbol{x}_i^E||_2}{\sqrt{K}}\right)\frac{\boldsymbol{x}_i^E}{||\boldsymbol{x}_i^E||_2}\right) \tag{14}$$

We now apply the spatial hyperbolic graph convolution to the hyperbolic node features $\boldsymbol{x}_i^H$. The spatial hyperbolic graph convolution layer has two main operations, namely, hyperboloid linear transform and neighbourhood based aggregation.

Hyperbolic Linear Transform. We apply a hyperbolic linear transform on the hyperbolic node feature $\boldsymbol{x}_i^H$ to learn more expressive features $\boldsymbol{h}_i^H$. The hyperbolic linear transform constitutes matrix vector multiplication of the feature vectors by a weight matrix, followed by a bias translation. We denote the matrix vector multiplication in the hyperbolic space by $\otimes^K$, which multiples input features $\boldsymbol{x}_i^H$ with learnable weight matrix $\boldsymbol{W} \in \mathbb{R}^{C' \times C}$ where, C' is the number of output features. Following [13], we define the hyperbolic matrix multiplication $\otimes^K$ as:

$$\boldsymbol{W} \otimes^K \boldsymbol{x}_i^H = \exp_{\boldsymbol{o}}^K(\boldsymbol{W}\log_{\boldsymbol{o}}^K(\boldsymbol{x}_i^H)) \tag{15}$$

where, $\log_{\boldsymbol{o}}^K(\cdot)$ is on $\mathbb{H}^{C,K}$ and $\exp_{\boldsymbol{o}}^K(\cdot)$ maps to $\mathbb{H}^{C',K}$. We now define the bias vector addition in the hyperbolic space, denoted as $\oplus^K$, which adds a bias vector $\boldsymbol{b}$ to the above matrix multiplication expression. Following [13], we use a parallel transport $P^K_{\boldsymbol{o}\to\boldsymbol{x}_i^H}(\cdot)$ to transport the bias vector $\boldsymbol{b}$ from the tangent space of origin $\mathcal{T}_{\boldsymbol{o}}\mathbb{H}^{C',K}$ to the tangent space of the point of interest $\mathcal{T}_{\boldsymbol{x}_i^H}\mathbb{H}^{C',K}$. The hyperbolic bias addition $\oplus^K$ is defined as:

$$\boldsymbol{x}_i^H \oplus^K \boldsymbol{b} = \exp^K_{\boldsymbol{x}_i^H}\left(P^K_{\boldsymbol{o}\to\boldsymbol{x}_i^H}(\boldsymbol{b})\right) \tag{16}$$

We now define the hyperbolic linear transform on input node features $\boldsymbol{x}_i^H$ to learn a new feature vector $\boldsymbol{h}_i^H$ of stock s_i, given by:

$$\boldsymbol{h}_i^H = (\boldsymbol{W} \otimes^K \boldsymbol{x}_i^H) \oplus^K \boldsymbol{b} \quad \text{(Hyberbolic linear transform)} \tag{17}$$

Attentive Aggregation. To learn the interdependence between related stocks, we update the extracted hyperbolic features $\boldsymbol{h}_i^H$ of node (stock) s_i by aggregating neighbouring stocks' features. To this end, we employ an attentive aggregation in the hyperbolic space to capture the varying degree of influence each stock may have on other stocks, as shown in Figure 5. Formally, given a node $s_i \in \mathcal{S}$ and its associated neighbor $s_j \in \mathcal{N}(i)$, where, $\mathcal{N}(i)$ denotes the neighborhood of node s_i, we first map the hyperbolic features $\boldsymbol{h}_i^H$ and $\boldsymbol{h}_j^H$ to the tangent space of the origin using a logarithmic projection. Then, we compute an attention coefficient α_{ij} using a Euclidean Multi-layer Perceptron (MLP), given by:

$$\alpha_{ij} = \text{softmax}_{j\in\mathcal{N}(i)}\left(\text{MLP}\left(\log_{\boldsymbol{o}}^K(\boldsymbol{h}_i^H)||\log_{\boldsymbol{o}}^K(\boldsymbol{h}_j^H)\right)\right) \tag{18}$$

where, || denotes concatenation. The learnt attention coefficients α_{ij} are aggregated to update node s_i's features to new hyperbolic features $\boldsymbol{w}_i^H \in \mathbb{H}^{C',K}$. We perform the neighbourhood aggregation on the tangent space of the centre point $\boldsymbol{h}_i^H$, as this is where the Euclidean approximation is best, and there is lower distortion in relative distances between nodes leading to better performance [13]. Formally, we define this node feature aggregation as:

$$\boldsymbol{w}_i^H = \exp^K_{\boldsymbol{h}_i^H}\left(\sum_{j\in\mathcal{N}(i)}\alpha_{ij}\log^K_{\boldsymbol{h}_i^H}(\boldsymbol{h}_j^H)\right) \quad \text{(neighborhood aggregation)} \tag{19}$$

We combine the hyperbolic linear transform with this attentive neighbourhood aggregation of node features to devise an attentive hyperbolic convolution, HAttn$(\cdot)$. This attentive hyperbolic graph convolution updates a set of input node features $\boldsymbol{P} \in \mathbb{R}^{N\times C}$ to learnt output features $\boldsymbol{P'} \in \mathbb{R}^{N\times C'}$ where, C' is the number of output features per node. We obtain the feature vector $\boldsymbol{P'}$ by mapping the updated node features $\boldsymbol{w}_i^H$ of each node to the tangent space of the origin using the logarithmic projection $\log_{\boldsymbol{o}}^K(\cdot)$. Formally, the update rule of this attentive hyperbolic graph convolution is given by:

$$\boldsymbol{P'} = \log_{\boldsymbol{o}}^K\left(\text{HAttn}\left(\exp_{\boldsymbol{o}}^K(0, \boldsymbol{P})\right)\right) \tag{20}$$

We now generalize the above attentive hyperbolic graph convolution layer HAttn$(\cdot)$ to 3D input temporal features $\boldsymbol{X}^l \in \mathbb{R}^{\tau\times N\times C}$ and denote it as HAttn$^{3D}(\cdot)$. The generalised attentive hyperbolic convolution layer HAttn$^{3D}(\cdot)$ applies the same HAttn$(\cdot)$ layer to all time steps of the input and, produces a new set of features $\boldsymbol{X}^{l+1} \in \mathbb{R}^{\tau\times N\times C'}$. Formally,

$$\boldsymbol{X}^{l+1} = \log_{\boldsymbol{o}}^K\left(\text{HAttn}^{3D}\left(\exp_{\boldsymbol{o}}^K\left(0, \boldsymbol{X}^l\right)\right)\right) \tag{21}$$

We now describe how we combine the general attentive temporal and spatial convolutions for end-to-end stock ranking.

4.4 Spatiotemporal Hyperbolic Convolution

We sandwich the spatial hyperbolic graph convolution between two attentive temporal convolutions, as shown in Figure 6. The spatial hyperbolic convolution acts as a bridge between the two

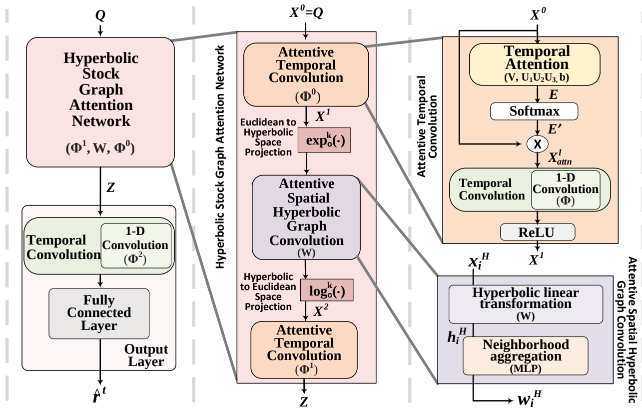

Figure 6: Component-wise layout of HyperStockGAT showing end-to-end ranking $\hat{r}^t$ from input features Q.

attentive temporal convolutions. This design choice allows the propagation of spatially updated features along the time axis through temporal convolutions. Formally, we feed temporal price features $Q \in \mathbb{R}^{\tau \times N \times C}$ to HyperStockGAT's first layer X^0. For all N stocks, we feed $C = 5$ temporal features: 1-day return ratio, 5, 10, 20 and 30 day moving averages which represent the daily, weekly and monthly trends over a lookback of τ days. As shown in Figure 6, we feed these temporal features Q to the attentive temporal convolution layer and map its Euclidean outputs X^1 to the hyperbolic space using the exponential projection. Then, we apply the attentive hyperbolic graph convolution and map its outputs X^2 to the tangent space of the origin with a logarithmic projection followed by a second attentive temporal convolution. We summarize these transformations made by HyperStockGAT to the input stock features Q to obtain the learned spatiotemporal stock representations Z as:

$$Z = \left(\phi^1 *_{\text{attn}} \log_{o}^{K}\left(\text{HAttn}^{3\text{D}}\left(\exp_{o}^{K}\left(0, \phi^0 *_{\text{attn}} Q\right)\right)\right)\right) \quad (22)$$

ϕ^0, ϕ^1 are weights of the first and second temporal convolutions.

4.5 HyperStockGAT Optimization and Learning to Rank

To rank stocks, we employ an output layer after the HyperStockGAT, as shown in Figure 6 (left), which maps the learned representations to a single time step prediction. Specifically, the output layer consists of a temporal convolution followed by a fully connected layer to predict the stock ranking $\hat{r}^t \in \mathbb{R}^N$, given by:

$$\hat{r}^t = W\left(\phi^2 * Z\right) + b \quad (23)$$

where, ϕ^2 are the parameters of the learnable temporal convolution kernel of the output layer and W, b are the parameters of the fully connected layer. We optimize HyperStockGAT using a combination of a pointwise regression and pairwise ranking-aware loss to minimize the difference between the predicted and actual return ratios while maintaining the relative order of top ranked stocks

Table 1: Dataset statistics detailing chronological date splits of the four markets and their corresponding graphs.

	NASDAQ	NYSE	TSE	China & HK
Train (Tr) Period	01/13-12/15	01/13-12/15	11/15-08/18	12/14-5/18
Val Period	01/16-12/16	01/16-12/16	08/18-07/19	05/18-6/19
Test Period	01/17-12/17	01/17-12/17	07/19-08/20	06/19-7/20
# Days Tr:Val:Test	756:252:237	756:252:237	693:231:235	756:252:285
# Stocks (Nodes)	1,026	1,737	95	85
# Edges	28,161	1,46,732	467	157
Avg. Node Degree	54.9 ± 58.7	168.9 ± 213.4	9.8 ± 12.8	3.7 ± 9.2
Max Node Degree	157	598	95	85
Hyperbolicity δ	1	1.5	0	0

with higher expected return for investment as:

$$L = \|\hat{r}^t - r^t\|^2 + \beta \sum_{i=0}^{N} \sum_{j=0}^{N} \max\left(0, -\left(\hat{r}_i^t - \hat{r}_j^t\right)\left(r_i^t - r_j^t\right)\right) \quad (24)$$

where, $\hat{r}^t$ and r^t are the predicted and actual ranking scores, respectively, and β is a weighting parameter.

5 EXPERIMENTAL SETUP

5.1 Stock Markets and Datasets

For a comprehensive evaluation of HyperStockGAT, we evaluate it on *four* real-world datasets from *US*, *Japanese* and, *Chinese* stock markets spanning over *six* years. We summarize statistics about the datasets in Table 1, and elaborate on them next:

NASDAQ [20]. is a volatile [54] US exchange. We evaluate HyperStockGAT on 1, 026 equity stocks from the NASDAQ Global and Capital markets that span the S&P 500 and NASDAQ Composite indexes.

NYSE [20]. is the world's largest stock exchange by market capitalization and is stable as compared to NASDAQ [32]. We evaluate HyperStockGAT on 1,737 stocks spanning the Dow Jones Industrial Average, S&P 500, and NYSE Composite indexes.

TSE [39]. is a smaller market contrasting with US markets [30]. We evaluate HyperStockGAT on the 95 largest stocks by market capitalization in Japan, in the TOPIX 100 index.

China and Hong Kong [28]. comprise the Shanghai, Shenzhen, and Hong Kong Stock Exchange. We evaluate HyperStockGAT on 85 top-traded China A-share stocks in these markets.

We collect historical stock prices using Google Finance[7] for all stocks. For a fair comparison, we follow the same data preprocessing as adopted by the works that introduced the datasets.

5.2 Evaluation Metrics

Returns. We compare the Sharpe ratio and cumulative investment return ratio (IRR) to assess profit generation ability of all methods. Following [20], we adopt a daily buy-hold-sell trading strategy that is, when the market closes on trading day $t-1$ the trader uses the method to get a ranked list of the predicted return ratio for each stock. The trader then buys the top k stocks and

[7]Google Finance: https://www.google.com/finance

Table 2: Ablation over HyperStockGAT (mean of 10 independent runs). * and † indicate the improvement over Temporal Convolution and state-of-the-art RSR-I, respectively, is statistically significant ($p < 0.005$), under Wilcoxon's signed rank test.

Model Component	**NYSE** ($\delta = 1.5$)			**NASDAQ** ($\delta = 1.0$)			**TSE** ($\delta = 0$)			**China** ($\delta = 0$)		
	SR@5	IRR@5	NGCG@5	SR@5	IRR@5	NGCG@5	SR@5	IRR@5	NGCG@5	SR@5	IRR@5	NGCG@5
Temporal convolution (T-Conv)	0.78	0.13	0.25	0.98	0.23	0.23	0.76	0.23	0.37	0.65	0.70	0.35
T-Conv+Temporal Attention (T-Attn)	0.81	0.14	0.30	1.01	0.26	0.24	0.80	0.27	0.40	0.70	0.74	0.38
Hyperbolic Convolution (H-Conv)+T-Conv	0.87*†	0.16	0.70*†	1.20*†	0.30*†	0.53*†	1.12*†	0.64*†	0.71*†	1.10*†	0.87*†	0.71*†
H-Conv+T-Conv+TAttn	0.91*†	0.17	0.75*†	1.30*†	0.38*†	0.58*†	1.16*†	0.69*†	0.73*†	1.16*†	0.92*†	0.77*†
Hyperbolic Attention+T-Conv	*1.02*†*	*0.22*†*	*0.86*†*	*1.35*†*	*0.41*†*	*0.67*†*	*1.16*†*	*0.72*†*	*0.77*†*	*1.20*†*	*0.96*†*	*0.86*†*
HyperStockGAT	**1.10*†**	**0.25*†**	**0.90*†**	**1.40*†**	**0.44*†**	**0.71*†**	**1.20*†**	**0.75*†**	**0.83*†**	**1.25*†**	**1.05*†**	**0.89**

then sells the bought stocks on the market close of the trading day t. The IRR is thus, the cumulative return on an investment over time, independent of the length of the duration. The IRR on day t is defined as, $\text{IRR}^t = \sum_{i \in \mathcal{S}^{t-1}} \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$ where, $\mathcal{S}^{t-1}$ denotes the set of stocks in the portfolio on day $t-1$ and p_i^t, p_i^{t-1} is the closing price of the stock i on day t and $t-1$ respectively. We also calculate the **Sharpe ratio (SR)**, which is a measure of the return of a portfolio compared to its risk [55].

Ranking. We evaluate HyperStockGAT's ranking ability using Normalized Discounted Cumulative Gain (NDCG@k). NDCG@k sums the true scores ranked in the order induced by the predicted scores, after applying logarithmic discount. For both returns and NDCG, we report results for top $k = 5$ stocks, and present performance variation with the top k in Section 6.4.

5.3 Training Setup

We perform all experiments on a Tesla P100 GPU. We use the grid search to find optimal hyperparameters based on validation Sharpe Ratio for all models. We explore lookback window length $\tau \in \text{range}[2, 12]$, loss weighting factor $\phi \in \text{range}[1, 10]$. We set the output space of the hyperbolic linear transform C' to 32. We use RiemannianSGD [10, 65] to optimize in the hyperbolic space. We set the learning rate $\in (1e^{-5}, 1e^{-3})$ and train HyperStockGAT for 50 epochs on all datasets. It takes 2, 3.5, 1.5 and 1.4 hours to train and test on NASDAQ, NYSE, TSE, and China and HK datasets, respectively.

5.4 Baseline Models

We compare HyperStockGAT with baselines of various formulations: regression, classification, reinforcement learning and ranking.

Regression. methods predict return ratios using historic price data and trade the top stocks.

- **SFM [66]** State Frequency Memory (SFM) recurrent network decomposes prices into signals of different frequencies via Discrete Fourier Transform (DFT). The DFT coefficients are fed into an LSTM with separate memory states for each frequency.
- **LSTM [4]** Vanilla LSTM network which feeds on the historic price features. We use a single layer LSTM with hidden size 32.

Classification. methods classify stock movements as [up, down, neutral] and trade the stocks where prices are expected to rise.

- **ARIMA [60]** Auto Regressive Integrated Moving Average (ARIMA) is fitted to non-stationary historic stock price data to forecast future price movements.
- **A-LSTM [19]** Adversarial LSTM is optimised using adversarial training which simulates stochasticity during training thereby enhancing stock forecasting performance.
- **GCN [39]** An Euclidean graph convolution network (GCN) to learn from relationships between stocks. We use a GCN model with two convolution layers and one prediction layer.
- **HATS [36]** A hierarchical graph attention model in the Euclidean space that uses a multi-graph to represent different kinds of relations. It uses hierarchical attention to selectively aggregate information from different relationship types.

Reinforcement Learning (RL). approaches optimize quantitative trading through RL by using profit as the reward.

- **DQN [11]** Ensemble of Deep Q-Learning agents that take daily open-high-low-close prices as input and outputs an action (long, short). We use Sharpe Ratio and IRR as the reward function.
- **iRDPG [41]** Imitative Recurrent Deterministic Policy Gradient (iRDPG), uses the Sharpe Ratio as it's reward function, and outputs an action –long or short position.

Ranking. methods trade the most profitable top-ranked stocks.

- **Rank LSTM [4]** Vanilla LSTM network which feeds on the historic price features. We use a single layer LSTM with hidden size 32 and optimise using the ranking loss function.
- **GCN [37]** A graph convolution network (GCN) to learn from relationships between stocks. We use a GCN model with two convolution layers and one prediction layer and optimise it using the ranking loss function.
- **RSR-E [20]** Uses an LSTM to encode temporal price features and Euclidean Temporal Graph Convolution to model stock relations. We use the cosine similarity between pairs of node features as the weight of each edge for a given time step.
- **RSR-I [20]** Same as RSR-E except that, RSR-I uses a one-layer MLP to estimate the weight of each edge. We feed pairs of temporal node features to the MLP and obtain a weight for the corresponding edge for a given time step.

6 RESULTS AND ANALYSIS

6.1 Ablation Study

We probe HyperStockGAT's stock ranking ability and profitability benefits from each of its components in Table 2. First, we observe

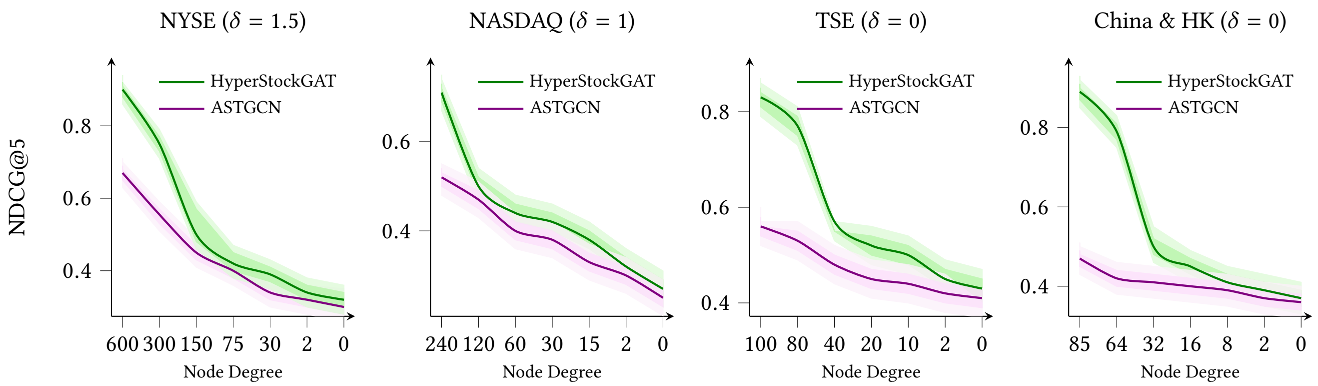

Figure 7: Impact of domain knowledge and hyperbolic learning. ASTGCN is the Euclidean variant of HyperStockGAT obtained by replacing hyperbolic graph attention with GAT.

Table 3: Profit and Ranking comparison (Mean of 10 independent runs) of hyperbolic GCN (HConv+Tconv+TAttn) and hyperbolic graph attention (HAttn+TConv+TAttn) with Euclidean GCN (GCN+TConv+TAttn) and graph attention (GAT+TConv+TAttn), respectively. $*$ and † indicate imorovements over GCN and GAT, respectively are statistically significant ($p < 0.005$) under Wilcoxon's signed rank test.

Model	NYSE $\delta = 1.5$		NASDAQ $\delta = 1$		TSE $\delta = 0$		China $\delta = 0$	
	SR	NDCG	SR	NDCG	SR	NDCG	SR	NDCG
TConv+TAttn	0.81	0.30	1.01	0.24	0.80	0.40	0.70	0.38
+GCN	0.88	0.55	0.92	0.42	0.75	0.52	0.74	0.40
+HConv	**0.91***	**0.75***	**1.30***	**0.58***	**1.16***	**0.73***	**1.16***	**0.77***
% Improv GCN	3.40	36.36	41.30	38.09	54.66	40.3	56.75	92.5
+GAT	0.91	0.67	0.99	0.52	0.80	0.56	0.80	0.47
+HAttn	**1.10†**	**0.90†**	**1.40†**	**0.71†**	**1.20†**	**0.83†**	**1.25†**	**0.89†**
% Improv GAT	20.87	34.3	41.41	36.53	50.00	48.21	56.25	89.36

that temporal attention significantly ($p < 0.005$) improves temporal convolutions, as it can effectively learn important long-term and short-term temporal dependencies. We note the biggest improvement on adding the hyperbolic graph convolution, validating the effectiveness of factoring in inter-stock relations, particularly via the hyperbolic space which better captures the scale-free nature of the stock market. Next, we note that adding hyperbolic attention leads to gains, likely because some relations between stocks such as having the *same parent company*, are more important than other relations, such as having the *same country of origin*. Finally, we observe that the spatial hyperbolic graph attention and temporal attention components complement each other by capturing latent spatio-temporal correlations via hyperbolic learning in stock markets that are inherently scale-free in nature. Noting the biggest improvements on adding the hyperbolic graph convolution, we now investigate the impact of modelling stock relationships via ordinary GCNs and learning on the hyperbolic space to further contextualise these performance improvements.

6.2 Impact of Hyperbolic Learning

Comparing Hyperbolic GNNs with Euclidean GNNs. : To further quantify the improvements from hyperbolic learning in HyperStockGAT, we compare it with its Euclidean counterpart in Table 3. Specifically, we replace the hyperbolic convolution with a traditional graph convolution (GCN) [37] and the hyperbolic graph attention with an ordinary graph attention network (GAT) [58]. We observe that complementing price-only methods with graph-based learning (GCN, GAT) usually leads to significant improvements ($p < 0.005$), reiterating the importance of exploiting spatial correlations amongst movements of related stocks. Next, we observe significant ($p < 0.005$) improvements on introducing hyperbolic learning for representing the underlying graph structure. This improvement empirically validates that, hyperbolic learning equips HyperStockGAT with geometrically appropriate inductive biases for better representing the inherent scale-free nature of stock graphs and capturing the impact of highly influential stocks on other stocks. Additionally, hyperbolic learning enables HyperStockGAT to produce more robust node embeddings, in contrast to Euclidean methods such as GCNs and GATs [51, 52]. We now investigate the ability of HyperStockGAT to capture the impact of highly influential stocks on other stocks to further analyze the gains in HyperStockGAT.

Quantifying the impact of high influence stocks. : We probe the impact of the highly influential stocks (hubs) and the effectiveness of domain knowledge on HyperStockGAT's ranking performance. We start by sorting the nodes in the decreasing order of their degree to identify the hubs of the scale-free network, and successively remove the edges of the corresponding hubs. Then, we compare our model with its Euclidean variant, by replacing the hyperbolic graph attention network with an ordinary GAT in Figure 7. We observe that the NDCG@5 decreases for both models as we remove edges, and they perform the worst after all edges are removed, essentially degenerating HyperStockGAT and GAT to a temporal model that does not account for inter-stock relations. An interesting observation is that, as we isolate the most influential hubs, HyperStockGAT's performances drops sharply, as the most influential nodes are now isolated, and HyperStockGAT is unable to learn the strong price correlations these hubs form with other stocks.

Table 4: Profitability comparison with classification (CLS), regression (REG), reinforcement learning (RL) and ranking methods (mean of 10 individual runs). Purple and Pink show best and second best (SOTA) results, respectively.◇ indicates the improvement RSR-I is statistically significant ($p < 0.005$), under Wilcoxon's signed rank test.

	Model	NYSE ($\delta = 1.5$)		NASDAQ ($\delta = 1$)	
		SR@5	IRR@5	SR@5	IRR@5
REG	SFM [66]	$0.19{\pm}1e^{-3}$	$0.11{\pm}2e^{-3}$	$0.16{\pm}6e^{-3}$	$0.09{\pm}5e^{-3}$
REG	LSTM [4]	$0.13{\pm}3e^{-3}$	$0.09{\pm}7e^{-3}$	$0.48{\pm}2e^{-3}$	$0.13{\pm}4e^{-3}$
CLS	ARIMA [60]	$0.33{\pm}3e^{-3}$	$0.10{\pm}5e^{-3}$	$0.55{\pm}1e^{-3}$	$0.10{\pm}6e^{-3}$
CLS	A-LSTM [19]	$0.81{\pm}4e^{-3}$	$0.14{\pm}7e^{-3}$	$0.97{\pm}5e^{-3}$	$0.23{\pm}3e^{-3}$
CLS	GCN [39]	$0.70{\pm}3e^{-3}$	$0.10{\pm}6e^{-3}$	$0.75{\pm}4e^{-3}$	$0.13{\pm}1e^{-3}$
CLS	HATS [36]	$0.73{\pm}5e^{-3}$	$0.12{\pm}2e^{-3}$	$0.80{\pm}6e^{-3}$	$0.15{\pm}7e^{-3}$
RL	DQN [11]	$0.72{\pm}5e^{-3}$	$0.12{\pm}4e^{-3}$	$0.93{\pm}5e^{-3}$	$0.20{\pm}6e^{-3}$
RL	iRDPG [41]	$0.85{\pm}7e^{-3}$	$0.18{\pm}3e^{-3}$	$1.32{\pm}5e^{-3}$	$0.28{\pm}4e^{-3}$
RANKING	Rank LSTM [4]	$0.79{\pm}1e^{-3}$	$0.12{\pm}6e^{-3}$	$0.95{\pm}4e^{-3}$	$0.22{\pm}2e^{-3}$
RANKING	GCN [37]	$0.72{\pm}7e^{-3}$	$0.16{\pm}3e^{-3}$	$0.46{\pm}4e^{-3}$	$0.13{\pm}5e^{-3}$
RANKING	RSR-E [20]	$0.88{\pm}6e^{-3}$	$0.20{\pm}3e^{-3}$	$1.12{\pm}5e^{-3}$	$0.26{\pm}4e^{-3}$
RANKING	RSR-I [20]	*$0.95{\pm}1e^{-3}$*	*$0.21{\pm}3e^{-3}$*	*$1.34{\pm}6e^{-3}$*	*$0.39{\pm}5e^{-3}$*
RANKING	**HyperStockGAT**	**$1.10^{\diamond}$**$\pm 8e^{-3}$	**$0.25^{\diamond}$**$\pm 9e^{-3}$	**$1.40^{\diamond}$**$\pm 7e^{-3}$	**$0.44^{\diamond}$**$\pm 1e^{-2}$
	% Improv. (SOTA)	15.78	19.04	4.47	12.82

Table 5: Profitability comparison with classification (CLS), regression (REG), reinforcement learning (RL) and ranking methods (mean of 10 individual runs). Purple and Pink show best and second best (SOTA) results, respectively. ◇ indicates the improvement over iRDPG is statistically significant ($p < 0.005$), under Wilcoxon's signed rank test.

	Model	TSE ($\delta = 0$)		China ($\delta = 0$)	
		SR@5	IRR@5	SR@5	IRR@5
REG	SFM [66]	$0.08{\pm}2e^{-3}$	$0.07{\pm}4e^{-3}$	$0.21{\pm}1e^{-3}$	$0.31{\pm}6e^{-3}$
REG	LSTM [4]	$0.63{\pm}2e^{-3}$	$0.20{\pm}5e^{-3}$	$0.17{\pm}3e^{-3}$	$0.63{\pm}8e^{-3}$
CLS	ARIMA [60]	$0.47{\pm}2e^{-3}$	$0.13{\pm}1e^{-3}$	$0.37{\pm}4e^{-3}$	$0.43{\pm}3e^{-3}$
CLS	A-LSTM [19]	$1.10{\pm}1e^{-3}$	$0.43{\pm}9e^{-2}$	$0.83{\pm}4e^{-3}$	$0.80{\pm}6e^{-3}$
CLS	GCN [39]	$0.90{\pm}7e^{-3}$	$0.28{\pm}5e^{-3}$	$0.73{\pm}6e^{-3}$	$0.75{\pm}3e^{-3}$
CLS	HATS [36]	$0.96{\pm}4e^{-3}$	$0.31{\pm}2e^{-3}$	$0.77{\pm}1e^{-3}$	$0.72{\pm}5e^{-3}$
RL	DQN [11]	$1.08{\pm}5e^{-3}$	$0.31{\pm}7e^{-3}$	$0.69{\pm}2e^{-3}$	$0.71{\pm}4e^{-3}$
RL	iRDPG [41]	*$1.10{\pm}2e^{-3}$*	*$0.55{\pm}1e^{-3}$*	*$1.16{\pm}6e^{-3}$*	*$0.87{\pm}5e^{-3}$*
RANKING	Rank LSTM [4]	$0.73{\pm}1e^{-3}$	$0.21{\pm}5e^{-3}$	$0.74{\pm}7e^{-3}$	$0.74{\pm}3e^{-3}$
RANKING	GCN [37]	$0.81{\pm}3e^{-3}$	$0.27{\pm}4e^{-3}$	$0.73{\pm}6e^{-3}$	$0.79{\pm}3e^{-3}$
RANKING	RSR-E [20]	$1.07{\pm}1e^{-3}$	$0.50{\pm}7e^{-3}$	$0.82{\pm}5e^{-3}$	$0.81{\pm}2e^{-3}$
RANKING	RSR-I [20]	$1.08{\pm}6e^{-3}$	$0.53{\pm}4e^{-3}$	$0.85{\pm}1e^{-3}$	$0.86{\pm}2e^{-3}$
RANKING	**HyperStockGAT**	**$1.20^{\diamond}$**$\pm 8e^{-3}$	**$0.75^{\diamond}$**$\pm 6e^{-3}$	**$1.25^{\diamond}$**$\pm 9e^{-3}$	**$1.05^{\diamond}$**$\pm 2e^{-3}$
	% Improv. (SOTA)	9.25	36.36	7.07	20.68

This observation is in line with financial literature that shows there exist few very highly influential stocks that drastically dominate a market and impact prices for all other stocks [3, 29]. Through this experiment, we note that hyperbolic learning effectively captures the scale-free nature in stock markets, particularly for highly influential stocks. We now analyze HyperStockGAT's profitability compared to state-of-the-art methods.

6.3 Profitability Comparison with Baselines

We compare HyperStockGAT with state-of-the-art methods of varying formulations in terms of profitability in Table 4, 5. We observe that HyperStockGAT consistently generates significantly ($p < 0.005$) higher risk-adjusted returns than all baselines across all markets. Generally, ranking and RL methods that are inherently optimized for higher returns, are more profitable than classification and regression methods, which do not necessarily select the most profitable stocks to trade. This observation validates our premise of formulating stock prediction as a learning to rank problem. We also observe that amongst the best performing ranking and RL models, those that model stock interdependence (RSR-I, HyperStockGAT) outperform price-only methods (LSTM, DQN, iRDPG), as they capture the spatial correlations amongst movements of related stocks.

Next, we observe that HyperStockGAT achieves an average of 9% more risk-adjusted return compared to the best baselines for markets with high hyperbolicity (lowest δ). This observation suggests that graph neural networks can significantly benefit from hyperbolic geometry, especially in stock markets that exhibit scale-free nature [35, 40]. We postulate this improvement to HyperStockGAT's ability to adaptively learn the latent impact of highly influential stocks on other stocks, capturing the scale-free nature of stock networks better [3]. This impact is not well captured by ordinary graph-based models (RSR-I, RSR-E) because, they operate in the Euclidean space [51, 52], unable to capture the asymmetric and latent

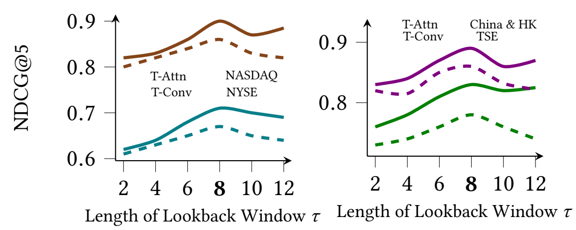

Figure 8: Sensitivity to length of lookback window τ

scale-free nature of stock correlations. Furthermore, we note larger improvements in stock markets with higher hyperbolicity (low δ). This observation is in-line with other hyperbolic graph neural networks [13, 40] that significantly outperform competitive methods on graphs with high hyperbolicity ($\delta < 2$). These observations collectively demonstrate HyperStockGAT's utility as a spatiotemporal learning model for ranking stocks. We now further probe into each of HyperStockGAT's sensitivity to various hyperparameters.

6.4 Sensitivity to Hyperparameters

Lookback window length τ: We study the variation in HyperStockGAT's performance with varying lengths of lookback window in Figure 8. First, we observe that HyperStockGAT using temporal attention outperforms temporal convolution due to its ability to capture important long term and short temporal dependencies in historical prices. Next, we note that using shorter lookbacks leads to poorer performance, likely because of lower market information in the window. As we increase the length of lookback window τ, we find that larger lookbacks allow the inclusion of stale information from older days having relatively lower influence on prices [7], deteriorating the ranking performance. However, we note that HyperStockGAT using temporal attention is able to selectively filter

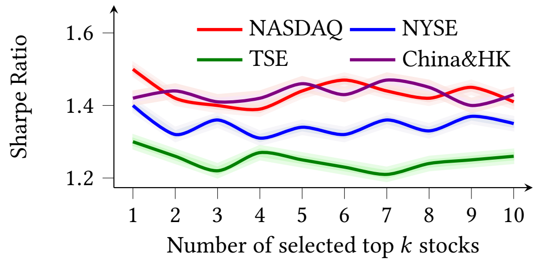

Figure 9: Sensitivity to number of top-stocks selected k

out crucial information from larger windows to an extent. Overall we observe that HyperStockGAT is *fairly* robust over varying window lengths τ and works best with mid-sized windows.

Number of selected top stocks $\mathbf{k}$*:* We analyze HyperStockGAT's profitability (Sharpe Ratio) variation with the number of selected top stocks k from the ranked stocks in Figure 9. We find that HyperStockGAT performs well and is consistent on varying k, showing suitability to strategies with different risk taking appetites.

7 CONCLUSION

We formulate stock selection as a *learning to rank* problem and model inter-stock relations based on domain knowledge as a graph. Building on financial theories on stock markets, we analyze the scale-free nature of stock networks and propose HyperStockGAT, the first neural scale-free graph-based learning network for stock selection. HyperStockGAT leverages spatial graph convolutions in the hyperbolic space to capture the scale-free nature of inter-stock relations. We devise a novel spatiotemporal hyperbolic graph learning component that blends attentive hyperbolic graph convolutions with temporal convolutions to capture the complex dependencies in related stocks' movements and the temporal evolution of stock prices simultaneously. HyperStockGAT significantly ($p < 0.005$) outperforms state-of-the-art methods by over 12%, and euclidean graph-based variants by over 3% in terms of risk-adjusted returns in four global markets over six years. Through ablative experiments, we probe HyperStockGAT's effectiveness and set forth its practical applicability for algorithmic trading. Our proposed HyperStockGAT can be directly generalized for hyperbolic spatiotemporal learning over hierarchical and scale-free networks across problems in varying domains, such as time-evolving social and citation networks.

REFERENCES

[1] Klaus Adam, Albert Marcet, and Juan Pablo Nicoli. 2016. Stock Market Volatility and Learning. *The Journal of Finance* 71, 1 (2016), 33–82.

[2] Rodrigo Aldecoa, Chiara Orsini, and Dmitri Krioukov. 2015. Hyperbolic graph generator. *Computer Physics Communications* 196 (2015), 492–496.

[3] Nitin Arora, Babu Narayanan, and Samit Paul. 2006. Financial Influences and Scale-Free Networks. In *Computational Science – ICCS 2006*, Vassil N. Alexandrov, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 16–23.

[4] 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.

[5] Albert-László Barabási and Réka Albert. 1999. Emergence of Scaling in Random Networks. *Science* 286, 5439 (Oct. 1999), 509–512. https://doi.org/10.1126/science.286.5439.509

[6] Stefano Battiston, James B. Glattfelder, Diego Garlaschelli, Fabrizio Lillo, and Guido Caldarelli. 2010. *The Structure of Financial Networks*. Springer London, London, 131–163. https://doi.org/10.1007/978-1-84996-396-1_7

[7] Dan Bernhaedt and Jianjun Miao. 2004. Informed Trading When Information Becomes Stale. *The Journal of Finance* 59, 1 (2004), 339–390. http://www.jstor.org/stable/3694898

[8] David C Blitz and Pim Van Vliet. 2007. The volatility effect. *The Journal of Portfolio Management* 34, 1 (2007), 102–113.

[9] Tim Bollerslev. 1986. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31, 3 (1986), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

[10] S. Bonnabel. 2013. Stochastic Gradient Descent on Riemannian Manifolds. *IEEE Trans. Automat. Control* 58, 9 (2013), 2217–2229.

[11] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. 2021. Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting. *Expert Systems with Applications* 164 (2021), 113820. https://doi.org/10.1016/j.eswa.2020.113820

[12] Francesco Catoni, Dino Boccaletti, Roberto Cannata, Vincenzo Catoni, Enrico Nichelatti, and Paolo Zampetti. 2008. *The mathematics of Minkowski space-time: with an introduction to commutative hypercomplex numbers*. Springer Science & Business Media.

[13] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic Graph Convolutional Neural Networks. 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., 4868–4879. http://papers.nips.cc/paper/8733-hyperbolic-graph-convolutional-neural-networks.pdf

[14] Wei Chen, Wenjie Fang, Guangda Hu, and Michael W. Mahoney. 2013. On the Hyperbolicity of Small-World and Tree-Like Random Graphs. arXiv:1201.1717 [cs.SI]

[15] D. Daiya, M.-S. Wu, and C. Lin. 2020. Stock Movement Prediction That Integrates Heterogeneous Data Sources Using Dilated Causal Convolution Networks with Attention, In ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2020-May (2020), 8359–8363. https://doi.org/10.1109/ICASSP40776.2020.9053479 cited By 0; Conference of 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020 ; Conference Date: 4 May 2020 Through 8 May 2020; Conference Code:161907.

[16] Ilia D. Dichev and Vicki Wei Tang. 2009. Earnings volatility and earnings predictability. *Journal of Accounting and Economics* 47, 1 (2009), 160–181. https://doi.org/10.1016/j.jacceco.2008.09.005 Accounting Research on Issues of Contemporary Interest.

[17] F.X. Diebold and K. Yilmaz. 2014. On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics* 182, 1 (2014), 119–134. https://doi.org/10.1016/j.jeconom.2014.04.012 cited By 416.

[18] Corwin D Edwards. 1955. Conglomerate bigness as a source of power. In *Business concentration and price policy*. Princeton University Press, 331–359.

[19] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 5843–5849. https://doi.org/10.24963/ijcai.2019/810

[20] 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* 37, 2 (March 2019), 1–30. https://doi.org/10.1145/3309547

[21] M. Gromov. 1987. Hyperbolic Groups. In *Essays in Group Theory*. Springer New York, 75–263. https://doi.org/10.1007/978-1-4613-9586-7_3

[22] Çaglar Gülçehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter W. Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2019. Hyperbolic Attention Networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 583–591. https://openreview.net/forum?id=rJxHsjRqFQ

[23] Alexandre Hocquard, Sunny Ng, and Nicolas Papageorgiou. 2013. A constant-volatility framework for managing tail risk. *The Journal of Portfolio Management* 39, 2 (2013), 28–40.

[24] Ehsan Hoseinzade, Saman Haratizadeh, and Arash Khoeini. 2019. U-CNNpred: A Universal CNN-based Predictor for Stock Markets.

[25] 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* (Marina Del Rey, CA, USA) *(WSDM '18)*. Association for Computing Machinery, New York, NY, USA, 261–269. https://doi.org/10.1145/3159652.3159690

[26] Chuangxia Huang, Shigang Wen, Mengge Li, Fenghua Wen, and Xin Yang. 2021. An empirical evaluation of the influential nodes for stock market network: Chinese A-shares case. *Finance Research Letters* 38 (2021), 101517. https://doi.org/10.1016/j.frl.2020.101517

[27] Chuangxia Huang, Shigang Wen, Mengge Li, Fenghua Wen, and Xin Yang. 2021. An empirical evaluation of the influential nodes for stock market network: Chinese A-shares case. *Finance Research Letters* 38 (2021), 101517. https://doi.org/10.1016/j.frl.2020.101517

[28] Jieyun Huang, Yunjia Zhang, Jialai Zhang, and Xi Zhang. 2018. A Tensor-Based Sub-Mode Coordinate Algorithm for Stock Prediction. In *Third IEEE International Conference on Data Science in Cyberspace, DSC 2018, Guangzhou, China, June 18-21, 2018*. IEEE, 716–721. https://doi.org/10.1109/DSC.2018.00114

[29] Wei-Qiang Huang, Xin-Tian Zhuang, and Shuang Yao. 2009. A network analysis of the Chinese stock market. *Physica A: Statistical Mechanics and its Applications* 388, 14 (July 2009), 2956–2964. https://doi.org/10.1016/j.physa.2009.03.028

[30] Andreas Humpe and Peter Macmillan. 2009. Can macroeconomic variables explain long-term stock market movements? A comparison of the US and Japan. *Applied Financial Economics* 19, 2 (2009), 111–119.

[31] Ziyu Jia, Youfang Lin, Jing Wang, Ronghao Zhou, Xiaojun Ning, Yuanlai He, and Yaoshuai Zhao. 2020. GraphSleepNet: Adaptive Spatial-Temporal Graph Convolutional Networks for Sleep Stage Classification. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 1324–1330. https://doi.org/10.24963/ijcai.2020/184 Main track.

[32] Christine X. Jiang, Jang-Chul Kim, and Robert A. Wood. 2011. A comparison of volatility and bid–ask spread for NASDAQ and NYSE after decimalization. *Applied Economics* 43, 10 (2011), 1227–1239. https://doi.org/10.1080/00036840802600376 arXiv:https://doi.org/10.1080/00036840802600376

[33] Weiwei Jiang. 2020. Applications of deep learning in stock market prediction: recent progress.

[34] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan V. Oseledets, and Victor S. Lempitsky. 2020. Hyperbolic Image Embeddings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 6417–6427. https://doi.org/10.1109/CVPR42600.2020.00645

[35] H. Kim, I. Kim, Y. Lee, and B. Kahng. 2002. Scale-Free Network in Stock Markets. *Journal of the Korean Physical Society* 40 (2002), 1105–1108.

[36] 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.

[37] 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. https://openreview.net/forum?id=SJU4ayYgl

[38] Baris Kocaarslan and Ugur Soytas. 2019. Dynamic correlations between oil prices and the stock prices of clean energy and technology firms: The role of reserve currency (US dollar). *Energy Economics* 84 (2019), 104502.

[39] Wei Li, Ruihan Bao, Keiko 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.). International Joint Conferences on Artificial Intelligence Organization, 4541–4547. https://doi.org/10.24963/ijcai.2020/626 Special Track on AI in FinTech.

[40] Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic Graph Neural Networks. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., 8230–8241. https://proceedings.neurips.cc/paper/2019/file/103303dd56a731e377d01f6a37badae3-Paper.pdf

[41] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. 2020. Adaptive Quantitative Trading: An Imitative Deep Reinforcement Learning Approach. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 02 (April 2020), 2128–2135.

[42] Miles Livingston. 1977. Industry movements of common stocks. *The Journal of Finance* 32, 3 (1977), 861–874.

[43] Rosario N Mantegna. 1999. Hierarchical structure in financial markets. *The European Physical Journal B-Condensed Matter and Complex Systems* 11, 1 (1999), 193–197.

[44] Thi-Thu Nguyen and Seokhoon Yoon. 2019. A Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning. *Applied Sciences* 9, 22 (2019). https://doi.org/10.3390/app9224745

[45] Maximillian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., 6338–6347. https://proceedings.neurips.cc/paper/2017/file/59dfa2df42d9e3d41f5b02bfc32229dd-Paper.pdf

[46] 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. https://doi.org/10.1016/j.physa.2014.03.083

[47] Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. 2020. Mix Dimension in Poincaré Geometry for 3D Skeleton-based Action Recognition. arXiv:2007.15678 [cs.CV]

[48] Yu Qin and Yi Yang. 2019. What You Say and How You Say It Matters: Predicting Stock Volatility Using Verbal and Vocal Cues. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 390–401. https://doi.org/10.18653/v1/P19-1038

[49] Erzsébet Ravasz and Albert-László Barabási. 2003. Hierarchical organization in complex networks. *Phys. Rev. E* 67 (Feb 2003), 026112. Issue 2. https://doi.org/10.1103/PhysRevE.67.026112

[50] Ram Babu Roy and Uttam Kumar Sarkar. 2011. Identifying influential stock indices from global stock markets: A social network analysis approach. *Procedia Computer Science* 5 (2011), 442–449. https://doi.org/10.1016/j.procs.2011.07.057

[51] Frederic Sala, Christopher De Sa, Albert Gu, and Christopher Ré. 2018. Representation Tradeoffs for Hyperbolic Embeddings. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018 (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer G. Dy and Andreas Krause (Eds.). PMLR, 4457–4466. http://proceedings.mlr.press/v80/sala18a.html

[52] Rik Sarkar. 2011. Low Distortion Delaunay Embedding of Trees in Hyperbolic Plane. In *Proceedings of the 19th International Conference on Graph Drawing* (Eindhoven, The Netherlands) *(GD'11)*. Springer-Verlag, Berlin, Heidelberg, 355–366. https://doi.org/10.1007/978-3-642-25878-7_34

[53] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Ratn 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)*. Association for Computational Linguistics, Online, 8415–8426. https://doi.org/10.18653/v1/2020.emnlp-main.676

[54] G William Schwert. 2002. Stock volatility in the new millennium: how wacky is Nasdaq? *Journal of Monetary Economics* 49, 1 (2002), 3–26.

[55] William F Sharpe. 1994. The sharpe ratio. *Journal of portfolio management* 21, 1 (1994), 49–58.

[56] Mieko Tanaka-Yamawaki and Masanori Yamanaka. 2019. Market efficiency is truly enhanced in sub-second trading market. *Procedia Computer Science* 159 (2019), 544–551. https://doi.org/10.1016/j.procs.2019.09.209

[57] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. 2019. Poincare Glove: Hyperbolic Word Embeddings. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=Ske5r3AqK7

[58] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[59] Gang-Jin Wang, Chi Xie, and H. Eugene Stanley. 2016. Correlation Structure and Evolution of World Stock Markets: Evidence from Pearson and Partial Correlation-Based Networks. *Computational Economics* 51, 3 (Oct. 2016), 607–635. https://doi.org/10.1007/s10614-016-9627-7

[60] Jung-Hua Wang and Jia-Yann Leu. 1996. Stock market trend prediction using ARIMA-based neural networks. In *Proceedings of International Conference on Neural Networks (ICNN'96), Washington, DC, USA, June 3-6, 1996*. IEEE, 2160–2165. https://doi.org/10.1109/ICNN.1996.549236

[61] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.* 32, 1 (2021), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

[62] 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)*. Association for Computational Linguistics, Melbourne, Australia, 1970–1979.

[63] Kedong Yin, Zhe Liu, Chong Huang, and Peide Liu. 2020. TOPOLOGICAL STRUCTURAL ANALYSIS OF CHINA'S NEW ENERGY STOCK MARKET: A MULTI-DIMENSIONAL DATA NETWORK PERSPECTIVE. *Technological and Economic Development of Economy* 26, 5 (May 2020), 1030–1051. https://doi.org/10.3846/tede.2020.12723

[64] Fisher Yu and Vladlen Koltun. 2015. Multi-Scale Context Aggregation by Dilated Convolutions. arXiv:1511.07122 [cs.CV]

[65] Hongyi Zhang, Sashank J. Reddi, and Suvrit Sra. 2016. Riemannian SVRG: Fast Stochastic Optimization on Riemannian Manifolds. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 4592–4600. http://papers.nips.cc/paper/6515-riemannian-svrg-fast-stochastic-optimization-on-riemannian-manifolds.pdf

[66] 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* (Halifax, NS, Canada) *(KDD '17)*. Association for Computing Machinery, New York, NY, USA, 2141–2149. https://doi.org/10.1145/3097983.3098117