# MULTI-RELATIONAL GRAPH DIFFUSION NEURAL NETWORK WITH PARALLEL RETENTION FOR STOCK TRENDS CLASSIFICATION

*Zinuo You[1], Pengju Zhang[1], Jin Zheng[2], John Cartlidge[2]*

[1]School of Computer Science, University of Bristol, UK
[2]School of Engineering Mathematics and Technology, University of Bristol, UK

## ABSTRACT

Stock trend classification remains a fundamental yet challenging task, owing to the intricate time-evolving dynamics between and within stocks. To tackle these two challenges, we propose a graph-based representation learning approach aimed at predicting the future movements of multiple stocks. Initially, we model the complex time-varying relationships between stocks by generating dynamic multi-relational stock graphs. This is achieved through a novel edge generation algorithm that leverages information entropy and signal energy to quantify the intensity and directionality of inter-stock relations on each trading day. Then, we further refine these initial graphs through a stochastic multi-relational diffusion process, adaptively learning task-optimal edges. Subsequently, we implement a decoupled representation learning scheme with parallel retention to obtain the final graph representation. This strategy better captures the unique temporal features within individual stocks while also capturing the overall structure of the stock graph. Comprehensive experiments conducted on real-world datasets from two US markets (NASDAQ and NYSE) and one Chinese market (Shanghai Stock Exchange: SSE) validate the effectiveness of our method. Our approach[1] consistently outperforms state-of-the-art baselines in forecasting next trading day stock trends across three test periods spanning seven years.

***Index Terms***— stock market prediction, graph neural network, graph representation learning, financial application

## 1. INTRODUCTION

Predicting the stock market has long been of interest to both academia and industry. In recent years, deep learning techniques have emerged as powerful tools to uncover latent stock market behaviors, particularly in forecasting prices or price movements [1]. Early works typically adopt Recurrent Neural Networks (RNNs) to predict the exact price of a stock based on temporal sequence formed by a single stock indicator such as daily closing prices [2]. While these methods achieved some success in predicting target stock prices, they often overlooked the intricate interconnections between stocks. This can result in catastrophic degradation when generalizing to multiple stocks [3]. Contemporary approaches bridge the gap by introducing different attention mechanisms [4, 5] or by constructing stock graphs [6, 7]. On the one hand, the attention mechanism considers the relations between stocks by utilizing the element-wise relevance of encoded long temporal sequences. On the other hand, the stock graphs represent stocks as nodes with inter-stock relationships as edges. While the attention-based models [5, 8] demonstrate their efficacy in forecasting stock prices and movements, they have

limited ability to model the inter-series relations and capture the temporal features. Graph Neural Networks (GNNs) have recently emerged as practical approaches for analyzing complex networks and multivariate time series [9, 10]. Since the stock market can be regarded as a complex network [11], GNNs offer a well-suited framework for analyzing a diverse set of stocks. Specifically, GNNs explicitly represent individual stocks as nodes, with their time-series data as node attributes, and capture the intricate inter-stock relations through edges.

However, current GNN-based methods for predicting the future status of stocks face two primary challenges: they often neglect the dynamic and asymmetric nature of inter-stock relations [12], and they overlook hierarchical intra-stock features [13]. Traditional GNN-based methods, like HATS [6], rely on static stock graphs based on industry-corporate or firm-specific relations. Such static stock graphs fail to encapsulate the ever-changing dynamics of stock markets, where inter-stock relations can shift rapidly. Additionally, the hierarchical features intrinsic to individual stocks are frequently sidelined by existing GNN methodologies. Many of these models are built on top of message-passing GNNs, where information propagation and graph representation learning are deeply coupled. This entanglement can result in severe loss of hierarchical intra-stock features [14], which are crucial for learning both individual and collective stock patterns [13].

To address the two challenges, we present the Multi-relational Graph Diffusion Neural Network with Parallel Retention (MGDPR), a novel graph-based representation learning approach. The contributions of this work are threefold. First, unlike conventional methods, MGDPR dynamically captures the relations between stocks by leveraging the concepts of information entropy and signal energy. This approach quantifies the directionality and intensity of relationships between stocks at each timestamp, providing a more granular view of the complicated inter-stock dynamics. Moreover, our multi-relational diffusion mechanism refines the generated stock graphs, adaptively learning task-optimal edges that further filter the noisy edges and recover task-relevant edges. Second, we propose a decoupled graph representation learning scheme with parallel retention, preserving the hierarchical intra-stock features and the long-term dependencies of stock indicator time series. Last, through extensive experiments on three real-world datasets covering 2,893 stocks over seven years, we demonstrate the effectiveness of MGDPR in forecasting future stock trends.

## 2. RELATED WORK

### 2.1. Attention-based Models for Stock Prediction

The attention mechanism has achieved great success in natural language processing and time series problems, which stems from its

---

[1]https://github.com/pixelhero98/MGDPR

efficacy in handling long sequences and modeling intra-sequence dependencies. Therefore, many attention-based models have been developed for stock selection, price prediction, and movement prediction. For instance, DA-RNN [15] employs a dual-stage attention mechanism within an encoder-decoder framework to forecast future stock prices. This mechanism selectively concentrates on different features of the input time series and further learns hidden inter-series relations. HMG-TF [8] enhances the extraction of hierarchical intra-stock features using a Multi-Scale Gaussian Prior and mitigates redundant learning in multi-head attention through orthogonal regularization. Additionally, DTML [5] endeavors to understand the asymmetric and dynamic relations between stocks using a multi-level context transformer in an end-to-end fashion. However, a recurring limitation in these models is their inability to explicitly capture inter-series relations and temporal dynamics, both of which are crucial in multivariate contexts like stock prediction.

## 2.2. GNN-based Models for Complex Networks

In complex evolving networks, entities are often characterized by various time-series features [16]. The stock market, a typical example of such networks, can be effectively modeled by analyzing the inter-relation and intra-features of entities' time series. For instance, HATS [6] integrates Long Short-term Memory and Gated Recurrent Unit layers to distill temporal features as node attributes, further employing a hierarchical attention mechanism to learn graph representations from manually crafted stock graphs. Similarly, Hyper-StockGAT [13] harnesses hyperbolic graph learning to capture the scale-free nature of inter-stock relations and hierarchical intra-stock features, utilizing a hypergraph attention mechanism to represent spatial stock correlations on Riemannian manifolds. While these studies underscore the importance of retaining hierarchical intra-stock features, their reliance on static, manually crafted stock graphs overlooks the dynamic and asymmetric nature of stock markets. In contrast, GraphWaveNet [17] presents an innovative architecture that combines an adaptive dependency matrix with stacked dilated 1D convolution components, adeptly capturing dynamic spatial inter-entity dependencies and managing extended temporal sequences. Furthermore, TPGNN [18] captures dynamic inter-entity relations by constructing a matrix polynomial using static matrix bases and time-sensitive coefficients. Nevertheless, these models overlook the complex interplays among various inter-entity relations.

## 3. PRELIMINARY

### 3.1. Notations

Let $\mathcal{G}_t(\mathcal{V}, \mathcal{E}_t, \mathcal{R})$ denote a weighted multi-relational stock graph at trading day $t$, where $\mathcal{V}$ is the set of nodes $\{v_0, ..., v_{N-1}\}$, $\mathcal{E}_t$ is the set of edges, and $\mathcal{R}$ is the set of relations. Then, let $\mathbf{A}_{t,r} \in \mathbb{R}^{N \times N}$ denote the adjacency matrix for relation $r \in \mathcal{R}$. Let $\mathbf{X}_{t,r} = \{x_{t,r,0}, x_{t,r,1}, ..., x_{t,r,N-1}\} \in \mathbb{R}^{N \times \tau}$ denote the node feature matrix for relation $r \in \mathcal{R}$, where $\tau$ denotes the historical lookback window size and $x_{t,r,i}$ represents the time series for node $v_i$ under relation $r$. Let $C_t \in \mathbb{R}^{N \times 1}$ denote the label matrix at trading day $t$, and $L$ denotes the number of layers in the model. Let $\mathbf{W}_l^n$ denote the learnable weight matrix, $b_l^n$ denote the bias term, and $\sigma(\cdot)$ denotes the activation function.

### 3.2. Problem Formulation

Our approach predicts future trends of multiple stocks by representing them as temporally ordered multi-relational graphs. Hence,

we transform this into a node classification task, namely the next trading day stock trend classification, which aligns with previous works [8, 13]. Meanwhile, we adopt a common yet effective label-generation rule for stocks during training, validation, and test periods. In general, the mapping relationship of this work is defined as,

$$f(\mathcal{G}_t(\mathcal{V}, \mathcal{E}_t, \mathcal{R})) \rightarrow C_{t+\tau_0}, \quad (1)$$

where $f(\cdot)$ denotes the proposed MGDPR and $\tau_0$ denotes the prediction step. In this work, we consider the single-step case with $\tau_0 = 1$.

## 4. METHODOLOGY

### 4.1. Dynamic Multi-relational Stock Graph Generation

Most existing GNN-based approaches for modeling multiple stocks stick to rigid graph generation rules that produce time-invariant stock graphs according to domain or expert knowledge (e.g., industry or firm-specific relations). However, these methods contradict the true nature of the stock market, which changes stochastically over time. According to financial studies [19, 20], the stock market can be modeled as a complicated communication system with stocks as transmitters and receivers of various information. Accordingly, we can further represent the evolving interactions between numerous stocks through a series of stock graphs ordered in time.

Concretely, we propose a novel multi-relational edge generation method based on information entropy and signal energy. Compared to adopting estimation-complex transfer entropy to obtain the directionality as prior research proposed [21], this allows us to more effectively approximate the directionality and intensity of different relations between stocks at each timestamp. The entry $(a_{t,r})_{i,j}$ of $\mathbf{A}_{t,r}$ is expressed by the following equation,

$$E(x_{t,r,i}) = \sum_{o=0}^{\tau-1} |x_{t,r,i}[o]|^2,$$

$$H(x_{t,r,i}) = -\sum_{m=0} p(x'_{t,r,i}[m]) \ln p(x'_{t,r,i}[m]),$$

$$p(x'_{t,r,i}[m]) = \frac{\sum_{o=0}^{\tau-1} \delta(x'_{t,r,i}[m] - x_{t,r,i}[o])}{\tau},$$

$$(a_{t,r})_{i,j} = \frac{E(x_{t,r,i})}{E(x_{t,r,j})} e^{H(x_{t,r,i}) - H(x_{t,r,j})}, \quad (2)$$

where $E(\cdot)$ denotes the signal energy, $H(\cdot)$ denotes the information entropy, $x'_{t,r,i}$ denotes the non-repeating subsequence of $x_{t,r,i}$, $p(\cdot)$ denotes the probability of a value, and $\delta(\cdot)$ represents the Dirac delta function. Note that $(a_{t,r})_{i,j}$ represents a directed and weighted edge from $v_i$ to $v_j$, indicating different impacts on each other. This approach facilitates information transmission from nodes with more predictabilities (smaller entropy) to nodes with less predictabilities (larger entropy) while preserving possibilities for the inverse cases.

### 4.2. Multi-relational Graph Diffusion

Conventional message-passing GNNs, such as Graph Convolutional Neural Networks (GCNs) [22] and Graph Attention Neural Networks (GATs) [23], propagate information by aggregating features from immediate neighbors of nodes. Nonetheless, this may overlook the potential incompatibility between the provided graph and the task objective, where certain edges are task-irrelevant and lead to sub-optimal performance [24]. To alleviate this issue, we propose a multi-relational diffusion process to spread different types

6546

**Table 1**: Dataset description for the three markets.

|                          | NASDAQ          | NYSE            | SSE             |
|--------------------------|-----------------|-----------------|-----------------|
| Training Period (Tr)     | 01/2013-12/2014 | 01/2013-12/2014 | 01/2015-12/2016 |
| Validation Period (Val)  | 01/2015-06/2015 | 01/2015-06/2015 | 01/2017-12/2017 |
| Test Period (Test)       | 07/2015-12/2017 | 07/2015-12/2017 | 01/2018-12/2019 |
| # Days (Tr:Val:Test)     | 483:103:631     | 468:123:631     | 488:241:503     |
| # Stocks                 | 1026            | 1737            | 130             |
| # Stock Indicators ($|\mathcal{R}|$) | 5   | 5               | 5               |

of information through learned task-optimal edges adaptively. The multi-relational graph diffusion at layer $l$ is defined as,

$$\mathbf{S}_{l,r} = \left( \sum_{k=0}^{K-1} \gamma_{l,r,k} \mathbf{T}_{l,r,k} \right) \odot \mathbf{A}_{t,r}, \quad \sum_{k=0}^{K-1} \gamma_{l,r,k} = 1,$$

$$\mathbf{H}_l = \sigma \left( \text{Conv2d}_{1 \times 1} \left( \Delta (\mathbf{S}_{l,r} \mathbf{H}_{l-1} \mathbf{W}_l^r)_{r=0}^{|\mathcal{R}|-1} \right) \right), \quad (3)$$

where $\gamma_{l,r,k}$ represents the weight coefficient, $\mathbf{T}_{l,r,k}$ denotes the column-stochastic transition matrix, $\mathbf{S}_{l,r}$ denotes the diffusion matrix, $K$ denotes the expansion step, $\text{Conv2d}_{1 \times 1}(\cdot)$ denotes the 2D convolutional layer with a 1x1 kernel, $\Delta(\cdot)_{r=0}^{|\mathcal{R}|-1}$ denotes the function that stacks all relational node representations, and $\mathbf{H}_l$ represents the latent diffusion representation.

The weight coefficient should satisfy the constraint to prevent graph signals from being amplified or reduced during the multi-relational graph diffusion. As the relations of graphs are temporally evolving in this work, we make $\gamma_{l,r,k}$ and $\mathbf{T}_{l,r,k}$ as learnable parameters rather than fixing on the invariant mappings for fixed-relation graphs (e.g., CORA, PubMed, CiteSeer, etc.) of the original graph diffusion [25].
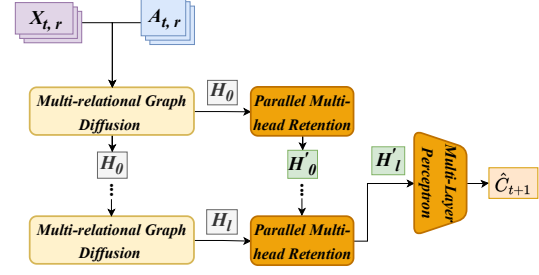
### 4.3. Graph Representation Learning with Parallel Retention

From the perspective of the expressiveness of GNNs, the deeply intertwined process of information propagation and representation learning hinders learning unique information of individual nodes [26]. This can result in a less expressive learned graph representation, degrading the model performance [14]. Prior research, such as DAGNN [14], instead of directly utilizing the representation from the final GNN layer, adopts an attention mechanism to learn a weighted graph representation based on representations from different layers. Similarly, approaches like HyperStockGAT [13] learn the graph representation in the hyperbolic space with attention mechanisms. This preserves distinctive hierarchical features of stock graphs, leading to substantial model improvements. On the one hand, the attention mechanism primarily concentrates on the relevance between different segments of the input sequence but falls short in modeling long-term dependencies. On the other hand, the retention mechanism is designed to capture the long-term dependencies while retaining the ability to model the content-wise relevance [27].

Consequently, we propose replacing the attention mechanism with a parallel retention mechanism to better incorporate long-term dependencies in the stock time series. The overall architecture of the proposed MGDPR is presented in Fig. 1. The layer update rule of MGDPR is defined by,

$$\mathbf{H}_l^{'} = \sigma \left( \left( \eta \left( \mathbf{H}_l \right) || (\mathbf{H}_{l-1}^{'} \mathbf{W}_l^1 + b_l^1) \right) \mathbf{W}_l^2 + b_l^2 \right) \quad (4)$$

where $\mathbf{H}_l^{'}$ denotes the hidden graph representation, $||$ denotes con-



**Fig. 1**: Schematic of graph representation learning scheme.

catenation, and $\eta(\cdot)$ denotes the parallel retention.[2]

Parallel retention is defined by,

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{Z}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{Z}\mathbf{W}_V$$

$$\mathbf{D}_{ij} = \begin{cases} \zeta^{i-j}, & \text{if } i \geq j \\ 0, & \text{if } i \leq j \end{cases}$$

$$\eta(\mathbf{Z}) = \phi \left( (\mathbf{Q}\mathbf{K}^T \odot \mathbf{D})\mathbf{V} \right). \quad (5)$$

Here, $\phi(\cdot)$ denotes the group normalization, $\mathbf{D}_{ij}$ denotes the entry of $\mathbf{D} \in \mathbb{R}^{\tau \times \tau}$ which is the masking matrix composing causal masking and decay factors along the relative distance, and $\zeta$ denotes the decay coefficient.

### 4.4. Objective Function

Considering Eq. 3 and Eq. 1, the model is optimized towards minimizing the objective function as defined,

$$\mathcal{J} = \frac{1}{B} \sum_{t=0}^{B-1} \mathcal{L}_{\mathcal{CE}}(\hat{C}_{t+1}, C_{t+1}) + \sum_{l=0}^{L-1} \sum_{r=0}^{|\mathcal{R}|-1} \left( \sum_{k=0}^{K-1} \gamma_{l,r,k} - 1 \right). \quad (6)$$

Here, $\hat{C}_{t+1}$ denotes the predicted label matrix of the trading day $t+1$, $\mathcal{L}_{CE}(\cdot)$ denotes the cross-entropy loss, the second term denotes the multi-relational graph diffusion constraint for all relations and all layers, and $B$ denotes the batch size.

## 5. EXPERIMENTS

### 5.1. Experiment Setup

All methods were implemented by Pytorch, Cuda Version 12.0, with 2x Navida A100.

**Dataset**. Following previous works, the stock data (active on 98% trading days) used in this work are collected from two US stock markets (NASDAQ and NYSE) and one China stock market (SSE). The stock indicators of NASDAQ, NYSE, and SSE are open price, high price, low price, close price, and trading volume. The details of the three datasets are presented in Table 1.

**Evaluation Metric**. Following prior research on stock classification and stock prediction, we use three metrics to evaluate the performance of the model on the downstream task, classification accuracy (Acc), Matthews correlation coefficient (Mcc), and F1-Score (F1).

**Model Setting**. The historical lookback window size $\tau$ is set to 21, which coincides with values from previous works and professional

---

[2]We have compared a variant of the model with multi-head attention [28]. The model with parallel retention has slightly higher performance and also benefits from computational efficiencies in training.

**Table 2**: Test period evaluation for MGDPR and baselines. Higher values indicate better performance for all metrics; bold denotes best.

| Method | NASDAQ | | | NYSE | | | SSE | | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc(%) | Mcc | F1 | Acc(%) | Mcc | F1 | Acc(%) | Mcc | F1 |
| DA-RNN [15] | 56.33±1.15 | 0.04±4.06×10⁻³ | 0.54±0.02 | 57.28±0.76 | 0.05±2.23×10⁻³ | 0.56±0.01 | 57.03±0.42 | 0.04±2.41×10⁻³ | 0.55±0.02 |
| HMG-TF [8] | 58.32±0.41 | 0.10±1.79×10⁻³ | 0.57±0.01 | 59.11±0.35 | 0.09±3.81×10⁻³ | 0.59±0.01 | 58.90±0.36 | 0.11±3.01×10⁻³ | 0.59±0.01 |
| DTML [5] | 57.56±0.67 | 0.06±1.98×10⁻³ | 0.58±0.01 | 58.78±0.45 | 0.08±2.00×10⁻³ | 0.60±0.01 | 59.63±0.21 | 0.09±5.42×10⁻³ | 0.59±0.01 |
| HATS [6] | 50.37±1.80 | 0.01±4.79×10⁻³ | 0.48±0.02 | 51.93±0.76 | 0.02±6.55×10⁻³ | 0.50±0.03 | 53.13±0.47 | 0.02±5.07×10⁻³ | 0.50±0.01 |
| GraphWaveNet [17] | 59.19±0.55 | 0.06±6.83×10⁻³ | 0.60±0.02 | 62.14±1.08 | 0.07±3.20×10⁻³ | 0.59±0.02 | 60.78±0.23 | 0.06±2.93×10⁻³ | 0.57±0.01 |
| HyperStockGAT [13] | 57.23±0.71 | 0.06±5.36×10⁻³ | 0.59±0.02 | 59.34±0.46 | 0.08±5.73×10⁻³ | 0.61±0.02 | 58.36±0.22 | 0.09±4.10×10⁻³ | 0.58±0.02 |
| TPGNN [18] | 60.42±0.49 | 0.10±3.45×10⁻³ | 0.61±0.02 | 61.81±0.19 | 0.11±4.45×10⁻³ | 0.60±0.02 | 62.69±0.10 | 0.12±1.66×10⁻³ | 0.60±0.02 |
| MGDPR | **62.77±0.65** | **0.13±4.49×10⁻³** | **0.62±0.01** | **64.54±0.20** | **0.13±1.88×10⁻³** | **0.63±0.01** | **63.90±0.32** | **0.14±2.01×10⁻³** | **0.62±0.02** |

financial studies [29]. The number of layers $L$ for NASDAQ and NYSE is set to 8, and for SSE is set to 5. The embedding dimension is set to 256. The number of layers of MLP is set to 2 for NASDAQ, NYSE, and SSE. The batch size is set to full batch, the learning rate is set to $2.5 \times 10^{-4}$, the number of training epochs is set to 900, and the optimizer is set to Adam. The decay coefficient $\zeta$ is set to 1.27, and the expansion step $K$ is set to 7 for NASDAQ, 8 for NYSE, and 3 for SSE. All hyperparameters are tuned on the validation periods.
**Baseline**. We further compare our approach with methods related to stock movements prediction, including attention-based methods: DA-RNN [15], DTML [5], and HMG-TF [8], and GNN-based methods: HATS [6], GraphWaveNet [17], HyperStockGAT [13], and TPGNN [18].

## 5.2. Experimental Result and Analysis

We evaluate the performance of our proposed model MGDPR against other baseline methods on the next trading day stock trend classification across three test periods. From the results summarized in Table 2, we can make three observations. (i) Graph-based methods like GraphWaveNet, TPGNN, MGDPR, and HyperStockGAT, which explicitly model the relationships between entities, outperform those such as DA-RNN, HMG-TF, and DTML, which rely on attention mechanisms for implicit inter-entity relations. (ii) Among GNN-based models, those (TPGNN, MGDPR, and GraphWaveNet) capturing dynamic links and broader neighborhoods yield superior results compared to HyperStockGAT and HATS. This advantage may stem from the latter methods' reliance on pre-defined static stock graphs and updates limited to immediate neighbors. (iii) Among models incorporating dynamic relations, MGDPR performs better over TPGNN and GraphWaveNet, which only consider single inter-entity relations. In summary, the proposed MGDPR consistently outperforms other models across all evaluation metrics. These observations indicate that neither prior domain knowledge nor manually constructed graphs adequately capture the complex, evolving relationships between stocks. Previous studies have relied on undirected and unweighted graphs, which are insufficient for modeling the dynamic nature of stock markets. In the meantime, the downstream task involves labeling multiple stocks at each timestamp, necessitating models that incorporate the likelihood that highly similar stocks may belong to different classes or seemingly unrelated stocks may fall into the same class over time. Therefore, it is crucial to consider the time-series features of individual stocks in terms of both piece-wise relevance and long-term dependencies.

## 5.3. Ablation Study

We conduct an ablation study to investigate the effectiveness of each module in our proposed approach, shown in Fig. 2. The proposed
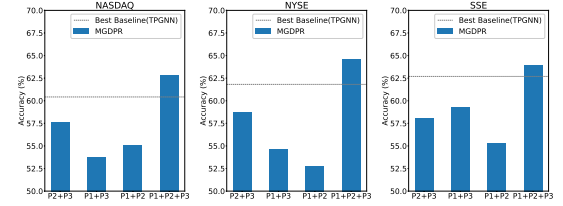


**Fig. 2**: Ablation study results. Only the full MGDPR (P1+P2+P3) has higher accuracy than the best baseline in each market.

MGDPR incorporates three critical components: Dynamic Multi-relational Stock Graph Generation (P1), Multi-relational Graph Diffusion (P2), and Graph Representation Learning with Parallel Retention (P3). Specifically, removing P1 leads to an average decrease in classification accuracy by 3.42%, underscoring the importance of dynamically capturing multiple inter-stock relations. Eliminating P2 has varying impacts across the three datasets, likely due to differences in graph sizes (i.e., NASDAQ and NYSE have more stocks than SSE). The diffusion process proves more effective in larger stock graphs, possibly as it adaptively disseminates information across broader neighborhoods. Lastly, replacing P3 results in an average decline in classification accuracy by 6.32%. This suggests that the parallel retention mechanism better captures long-term dependencies and temporal features of stock time series.

## 6. CONCLUSION

We have presented MGDPR, a novel graph learning framework designed to capture evolving inter-stock relationships and intra-stock features. To address the limitations of traditional graph-based models, we conceptualize the stock market as a communication system. First, we use information entropy and signal energy to quantify the connectivity and intensity between stocks on each trading day. Then, we propose the multi-relational diffusion process for the generated stock graph, aiming to learn task-optimal edges. This step alleviates the discordance between the task objectives and the input graph. Finally, we adopt a decoupled graph representation learning scheme with parallel retention. This module is designed to preserve hierarchical features within individual stocks and the long-term dependencies in their time-series features. Empirical results validate the efficacy of MGDPR in forecasting future stock trends. One limitation is MGDPR's focus on quantitative representations of information entropy, while neglecting statistical properties. Future work will optimize the entropy-based algorithm through statistical physics theories.

# 7. REFERENCES

[1] Ryo Akita, Akira Yoshihara, Takashi Matsubara, and Kuniaki Uehara, "Deep learning for stock prediction using numerical and textual information," in *15th Int. Conf. on Computer and Information Science (ICIS)*. IEEE, 2016, pp. 1–6.

[2] Wei Bao, Jun Yue, and Yulei Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PloS one*, vol. 12, no. 7, pp. e0180944, 2017.

[3] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z Pan, and Huajun Chen, "Knowledge-driven stock trend prediction and explanation via temporal convolutional network," in *World Wide Web Conference (WWW), Companion Proceedings*, 2019, pp. 678–685.

[4] Yumo Xu and Shay B Cohen, "Stock movement prediction from tweets and historical prices," in *56th Ann. Meeting Assoc. for Computational Linguistics*, 2018, vol. 1, pp. 1970–1979.

[5] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang, "Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts," in *Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 2037–2045.

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

[7] Sheng Xiang, Dawei Cheng, Chencheng Shang, Ying Zhang, and Yuqi Liang, "Temporal and heterogeneous graph neural network for financial time series prediction," in *31st ACM Int. Conf. on Information & Knowledge Management*, 2022, pp. 3584–3593.

[8] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo, "Hierarchical multi-scale gaussian transformer for stock movement prediction.," in *IJCAI*, 2020, pp. 4640–4646.

[9] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in *AAAI*, 2020, pp. 5363–5370.

[10] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu, "Traffic flow prediction via spatial temporal graph neural network," in *The Web Conference (WWW)*, 2020, pp. 1082–1092.

[11] Ruey S Tsay, *Analysis of financial time series*, John Wiley & Sons, 2005.

[12] Thanh Trung Huynh, Minh Hieu Nguyen, Thanh Tam Nguyen, Phi Le Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer, "Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction," in *Web Search and Data Mining (WSDM)*, 2023, pp. 850–858.

[13] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah, "Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading," in *The Web Conference (WWW)*, 2021, pp. 11–22.

[14] Meng Liu, Hongyang Gao, and Shuiwang Ji, "Towards deeper graph neural networks," in *Knowledge Discovery & Data Mining (KDD)*, 2020, pp. 338–348.

[15] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[16] Chang Liu and Nior Arunkumar, "Risk prediction and evaluation of transnational transmission of financial crisis based on complex network," *Cluster Computing*, vol. 22, pp. 4307–4313, 2019.

[17] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[18] Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen, "Multivariate time-series forecasting with temporal polynomial graph neural networks," in *NeurIPS*, 2022, pp. 19414–19426.

[19] Syed Jawad Hussain Shahzad, Jose Areola Hernandez, Mobeen Ur Rehman, Khamis Hamed Al-Yahyaee, and Muhammad Zakaria, "A global network topology of stock markets: Transmitters and receivers of spillover effects," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 2136–2153, 2018.

[20] Rama Cont and Jean-Philipe Bouchaud, "Herd behavior and aggregate fluctuations in financial markets," *Macroeconomic Dynamics*, vol. 4, no. 2, pp. 170–196, 2000.

[21] Peng Yue, Yaodong Fan, Jonathan A Batten, and Wei-Xing Zhou, "Information transfer between stock market sectors: A comparison between the usa and china," *Entropy*, vol. 22, no. 2, pp. 194, 2020.

[22] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

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

[24] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun, "Measuring and relieving the over-smoothing problem for graph neural networks from the topological view," in *AAAI*, 2020, pp. 3438–3445.

[25] Johannes Gasteiger, Stefan Weißenberger, and Stephan Günnemann, "Diffusion improves graph learning," in *NeurIPS*, 2019, pp. 13333–13345.

[26] Hanqing Zeng, Muhan Zhang, Yinglong Xia, Ajitesh Srivastava, Andrey Malevich, Rajgopal Kannan, Viktor Prasanna, Long Jin, and Ren Chen, "Decoupling the depth and scope of graph neural networks," in *NeurIPS*, 2021, pp. 19665–19679.

[27] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei, "Retentive network: A successor to transformer for large language models," *arXiv preprint arXiv:2307.08621*, 2023.

[28] Zinuo You, Zijian Shi, Hongbo Bo, John Cartlidge, Li Zhang, and Yan Ge, "DGDNN: Decoupled graph diffusion neural network for stock movement prediction," in *16th Int. Conf. on Agents and Artificial Intelligence (ICAART)*, Feb 2024.

[29] Klaus Adam, Albert Marcet, and Juan Pablo Nicolini, "Stock market volatility and learning," *The Journal of finance*, vol. 71, no. 1, pp. 33–82, 2016.