

DYCOR: Capturing Hidden Stock Relationships for Stock Trend Prediction

Kangmin Choi*

Hanyang University
Seoul, Republic of Korea
arceus2@hanyang.ac.kr

Jungwoo Yang

Seoul National University
Seoul, Republic of Korea
jwyang0213@snu.ac.kr

Geon Shin*

Hanyang University
Seoul, Republic of Korea
shingeon94@hanyang.ac.kr

Hyunjoon Kim†

Hanyang University
Seoul, Republic of Korea
hyunjoonkim@hanyang.ac.kr

Abstract

Stock trend prediction, the task of forecasting future trends of stocks from their historical feature sequences, remains highly challenging due to the complex and dynamic nature of financial markets. In reality, stocks form diverse relationships that transcend traditional sector boundaries as market conditions evolve, i.e., stocks within the same sector may display different trends, while those in different sectors often exhibit similar movements. However, most existing stock prediction methods rely on predefined static relationships, lacking flexibility to adapt to changing market dynamics. Furthermore, objectives widely adopted in prior work have limitations in capturing complex patterns and relationships in stock market data. To address these limitations, we propose DYCOR, a novel stock trend prediction method that integrates two key innovations: (i) dynamic stock clustering, which captures market characteristics without relying on predefined relationship data by adaptively discovering hidden stock relationships; and (ii) correlation-aware training, which aligns predicted and ground-truth stock trends by reflecting their correlations in a fine-grained manner. We evaluate DYCOR on three datasets NASDAQ, NYSE, and S&P 500 widely used in existing research, and this method demonstrates superior performance across correlation-based and retrieval-based metrics compared to state-of-the-art baseline methods, while maintaining competitive runtime efficiency.

CCS Concepts

- Computing methodologies → Machine learning; Artificial intelligence;
- Applied computing → Economics.

Keywords

stock trend prediction; dynamic stock clustering; correlation-aware training

*Co-first authors.

†Corresponding author.



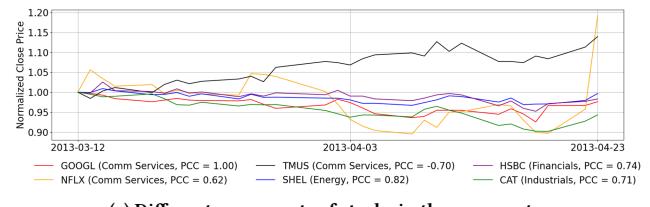
This work is licensed under a Creative Commons Attribution 4.0 International License.

CIKM '25, Seoul, Republic of Korea

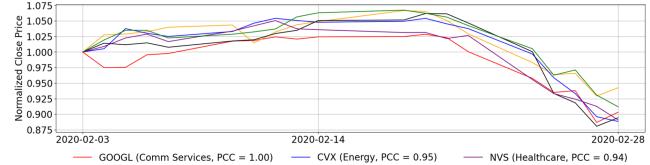
© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2040-6/2025/11

<https://doi.org/10.1145/3746252.3761413>



(a) Different movements of stocks in the same sector



(b) Similar movements of stocks in different sectors

Figure 1: Motivation for dynamic stock relation modeling. Co-movements of GOOGL and cross-sector stocks overlooked by static sector-based relationships.

ACM Reference Format:

Kangmin Choi, Geon Shin, Jungwoo Yang, and Hyunjoon Kim. 2025. DYCOR: Capturing Hidden Stock Relationships for Stock Trend Prediction. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*, November 10–14, 2025, Seoul, Republic of Korea. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3746252.3761413>

1 Introduction

Forecasting stock trends accurately can significantly benefit financial institutions, fund managers, and policymakers in navigating volatile market conditions and finding investment strategies. However, this problem has long been a critical challenge in both academia and industry, due to the highly dynamic and complex nature of financial markets. With the advent of deep learning and the proliferation of financial data, numerous methods [4, 8, 9, 13, 15, 18, 26, 34, 37] have been proposed to improve the accuracy of predicting stock trends. Existing methods have been designed based on either *fundamental analysis* or *technical analysis*.

Fundamental analysis methods leverage not only stock prices but also external data sources to enrich stock relations such as news articles [38], bank-firm networks [25], fund-holdings [7], and executive linkages [42]. However, their reliance on such external data may not fully capture nuanced and time-varying interactions

between stocks, and the predefined static relationships between stocks [9, 26], such as those represented by sectors [36] or Wikidata [29] may not generalize to the highly volatile nature of the stock market [4, 8, 34].

Technical analysis methods are grouped into four major approaches: (i) an **individual stock prediction approach** focuses solely on the historical data of each stock; (ii) an **all-pair-correlation-based approach** models pairwise dependencies between stocks; (iii) a **hypergraph-based approach** captures group-wise stock interactions. Yet, all of these approaches may not flexibly reflect time-varying and various market behaviors viewed from different market and sector perspectives, which affect the movement of individual stocks: approach (i) does not take into account each stock’s interaction with its market or other stocks, treating each stock in isolation; approach (ii) is limited to capturing only pairwise interactions without higher-order structures of multiple stocks; and approach (iii) constructs stock relation graphs from statistical measures such as Pearson correlation coefficient (PCC) [10, 15] and dynamic time warping (DTW) [34] over the entire training period, but the graphs computed from such in-sample statistics are fixed throughout training, making them unable to capture evolving market conditions in inference.

Figure 1 illustrates the limitations of using static stock relationships by visualizing stock co-movement patterns of GOOGL and other stocks showing a high PCC with GOOGL over two different periods. In Figure 1a, while GOOGL and NFLX within the same sector often move together as indicated by high positive PCC, GOOGL and TMUS within the same sector exhibit divergent behavior, showing negative correlations. Interestingly, GOOGL, HSBC, SHEL, and CAT from entirely different sectors sometimes demonstrate stronger co-movement than stocks from the same sector. In Figure 1b, a major market shock leads to widespread co-movements across sectors, indicating that large-scale disruptions can override sectoral boundaries. These observations highlight that static sector-based or precomputed relational graphs may not effectively capture time-varying stock dependencies, risking misaligned relational modeling under dynamic market conditions.

Second, Figure 2 illustrates two fundamental limitations of widely-used loss functions: mean squared error (MSE) and pairwise ranking-aware loss [43] (hereafter simplified as ranking loss). Note that MSE over the predicted returns P1 and their ground truth returns is very low, but this is not because the model accurately approximates ground-truth returns, but because the model might find it easy to minimize MSE by predicting values near zero across all stocks, without considering the relative differences among them. Despite exhibiting a low ranking loss, the predicted returns P2 reveal its limitations. Ranking loss imposes a penalty between stocks s_2 and s_3 due to the mismatch in the sign of their pairwise differences between the ground truth and P2. However, it fails to penalize the discrepancies between s_1 and s_2 , or s_1 and s_3 , as the relative orderings are preserved, even though the magnitude differences remain substantial, e.g., while the ground truth difference between stocks s_1 and s_2 is $0.5 - 0.4 = 0.1$, the predicted difference is $0.51 - (-0.21) = 0.72$. This limitation highlights how ranking loss can fail to capture magnitude mismatches when the sign of the pairwise differences is consistent, potentially leading to misleading evaluations of model performance.

Stocks	s_1	s_2	s_3	s_4	Ranking Loss	MSE
Ground Truth Returns	0.5 (^{1st})	0.4 (^{2nd})	-0.2 (^{3rd})	-0.3 (^{4th})		
Predicted Returns P1	-0.0010 (^{3rd})	0.0020 (^{2nd})	-0.0011 (^{4th})	0.0021 (^{1st})	3.96e-4	0.1351
Predicted Returns P2	0.51 (^{1st})	-0.21 (^{3rd})	0.41 (^{2nd})	-0.31 (^{4th})	4.65e-2	0.1861

Figure 2: Limitations of MSE and pairwise ranking-aware loss.

To address these limitations, we propose **DYCOR**, a stock trend forecasting method designed to capture both evolving and latent relationships between stocks, and to learn how each stock’s trend correlates with those of other stocks, thereby enhancing forecasting accuracy. DYCOR consists of four key components: (a) **dynamic stock clustering** groups all stocks at each time step based on their similarity to evolving representations of latent segments, enabling flexible modeling of latent interactions beyond static sector boundaries, (b) **intra-stock correlation** associates each stock with its highly correlated stocks discovered by dynamic stock clustering, (c) in **stock-wise inter-cluster aggregation**, DYCOR aggregates contextualized representations of each stock obtained from different latent segments by weighting them according to how likely that stock is to belong to each latent segment, thereby capturing multi-segment influences that reflect the complex market dynamics, and (d) **correlation-aware training** ensures that predictions align with real-world market behaviors.

Our contributions are summarized as follows:

- **Important discovery.** We show that existing methods struggle to reflect dynamic stock interactions due to their reliance on individual features, fixed pairwise relationships, and static external knowledge, which limit their adaptability to evolving market conditions.
- **Novel framework.** We introduce a novel stock trend prediction framework DYCOR designed to capture latent and time-varying stock relationships.
- **Extensive evaluation.** We evaluate DYCOR on NASDAQ, NYSE, and S&P 500, and we observe that DYCOR mostly outperforms state-of-the-art baseline methods. On average, DYCOR achieves 53.3% higher IC and 11.2% higher Rank IC compared to the best baseline across all benchmarks, confirming its effectiveness in modeling dynamic stock relationships and enhancing prediction stability under volatile market conditions.

2 Related Work

Various methods for the task of stock trend prediction have been proposed by a large amount of literature, which can be categorized into two groups: (1) *fundamental analysis* methods try to understand the true value of a stock by taking advantage of external data sources other than stock prices, and (2) *technical analysis* methods focus on how the stock price moves over time by seeking for patterns in technical indicators such as price trends and volume.

Fundamental analysis methods utilize external information such as news data [38], executive networks [42], market knowledge graphs [6, 7, 9, 16, 25, 26, 35]. RSR [9] tries to captures dynamic stock relationships by leveraging predefined relationships between stocks such as Wikidata [29] and sector information [23]. STHAN-SR [26] employs a hypergraph that integrates stock interdependencies and temporal trend movement of stocks. ESTIMATE [15] captures non-pairwise correlations between stocks via hypergraph convolution

over a predefined hypergraph based on the industry of the companies. HATR-I [30] groups stocks into soft clusters, obtains stock representations based on explicit relationships in predefined graphs consisting of industries, topicalities, and shareholdings. These methods heavily rely on predefined concepts, relationships, and rules, and create static correlation graphs. However, the predefined data substantially constrain the diversity of potential interactions among stock markets, which are characterized by high volatility.

Most technical analysis methods belong to one of the following three approaches.

Individual stock prediction approach. Previous work to forecast the trend of stocks individually relies solely on a single stock's features in a lookback window to predict the trend of that stock, e.g., ARIMA [1, 32], Recurrent Neural Networks (RNNs) [3, 5, 10, 24, 41], Convolutional Neural Networks (CNNs) [27], the combination of CNNs and RNNs [19, 20], Transformers [22, 31], and a diffusion-based method [17]. However, these methods take a single stock as input, failing to incorporate inter-stock relationships that could enhance overall prediction accuracy.

All-pair-correlation-based approach. This approach focuses on utilizing pairwise correlations between stocks to explicitly capture inter-stock dependencies. DTML [39] refines the understanding of stock relationships by computing asymmetric and dynamic inter-stock dependencies via a Transformer encoder. Similarly, MASTER [18] computes stock correlations dynamically across time by performing intra-stock and inter-stock aggregations through the self-attention mechanism. MATCC [4] not only correlates different stocks but also captures temporal dependencies across different time periods of the same stock. However, this approach is limited to capture a direct pairwise relationship between every pair of stocks, which may overlook high-order or group-wise interactions that could provide additional insights into market dynamics.

Hypergraph-based approach. This approach captures complex and high-order stock relationships beyond simple pairwise stock interactions. CI-STHPAN [34] constructs dynamic hypergraphs based on temporal similarities of stock features using *dynamic time warping*. StockMixer [8] implicitly captures stock relationships without requiring predefined or precomputed hypergraphs. This method aggregates stock-level information into several market representations, and then distributes the market-influenced information back to individual stocks, which in turn generates a self-learnable hypergraph.

3 Preliminaries

Definition 3.1. (Stock Trend). The trend of stock s at time t is defined as the return ratio of the next trading day:

$$r_{s,t} = \frac{c_{s,t} - c_{s,t-1}}{c_{s,t-1}},$$

where $c_{s,t}$ represents the closing price of stock s at time t . The return ratio $r_{s,t}$ serves as the target value for forecasting. Each stock s has F features at each time step of a lookback window of length T , forming its feature matrix $\mathbf{X}_s = [\mathbf{x}_{s,t-T}, \mathbf{x}_{s,t-T+1}, \dots, \mathbf{x}_{s,t-1}] \in \mathbb{R}^{F \times T}$.

Definition 3.2. (Stock Trend Prediction). Given a set S of stocks, let $\{\mathbf{X}_s\}_{s \in S} \in \mathbb{R}^{|S| \times F \times T}$ represent the features of all stocks. The

Symbol	Definition
$\mathbf{X}_s \in \mathbb{R}^{F \times T}$	Feature matrix of stock s
$\mathbf{e}_s \in \mathbb{R}^d$	Stock encoding output (embedding) for stock s
$\mathbf{p}_k \in \mathbb{R}^d$	k -th principal component
C_k	Set of stocks belonging to latent segment k (cluster k)
$C_{k,l}$	Subset of stocks in latent segment k belonging to sub-segment l (sub-cluster l of cluster k)
$P(C_k s)$	Membership probability of stock s for cluster k
$\mathbf{h}_s^{(k)} \in \mathbb{R}^d$	Contextualized representation of stock s
$\mathbf{z}_s \in \mathbb{R}^d$	Final aggregated representation of stock s
$\hat{r}_{s,t}$	Predicted return ratio for stock s at time t
$r_{s,t}$	Actual return ratio for stock s at time t

Table 1: Key notations used throughout the paper.

goal of stock trend prediction is to forecast the return ratio $r_{s,t}$ for every stock using its historical features up to time t . Formally, the task is to learn a function f_θ parameterized by θ which predicts the return ratios of all stocks:

$$f_\theta(\{\mathbf{X}_s\}_{s \in S}) \rightarrow \{r_{s,t}\}_{s \in S},$$

4 Proposed Method

4.1 Overview

Figure 3 illustrates the overall architecture of DYCOR. Our framework consists of five main stages. The **stock encoding** stage encodes historical features of all stocks to generate their embeddings. From these embeddings, the **dynamic stock clustering** stage computes principal components through PCA, and divides all stocks into multiple clusters and sub-clusters based on their similarity to the components. The **intra-stock correlation** stage models relationships between stocks within each sub-cluster by using the self-attention mechanism. The **stock-wise inter-cluster aggregation** stage combines multiple representations of each stock from the perspectives of different principal components, weighted by their membership probabilities, to form a final representation. The **prediction** stage forecasts the return ratio of each stock.

4.2 Stock Encoding

Stock encoder takes features $\mathbf{X}_s \in \mathbb{R}^{F \times T}$ of each stock s in a look-back window as input, and encodes the input features of the look-back window into their latent representation, inspired by the recent work [4, 8] for stock trend prediction. Specifically, the stock encoder consists of three key components: (1) stock trend decomposition to extract medium-term trends and short-term fluctuations using moving average, (2) time mixing for progressively representing patterns from sequences of increasing lengths within a lookback window, and (3) indicator mixing to capture dependency between the features of a stock.

4.2.1 Stock Trend Decomposition. First, we decompose the input features $\mathbf{X}_s \in \mathbb{R}^{F \times T}$ of each stock s into two components using a moving average filter [40] to separate trends from short-term fluctuations:

$$\mathbf{X}_s^{tr} = \text{AvgPool1d}(\mathbf{X}_s), \quad \mathbf{X}_s^{fl} = \mathbf{X}_s - \mathbf{X}_s^{tr} \quad (1)$$

where \mathbf{X}_s^{tr} represents the trend component capturing medium-term price movements, \mathbf{X}_s^{fl} represents the fluctuation component

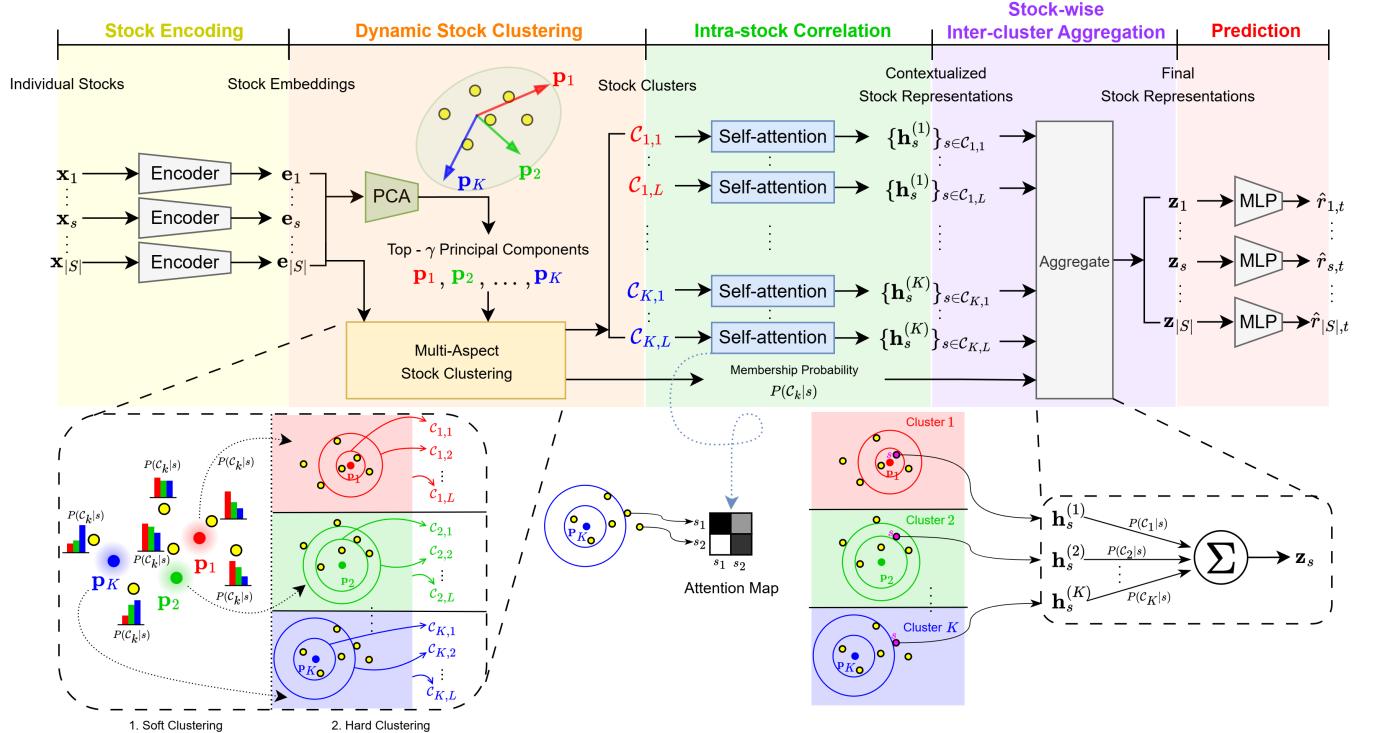


Figure 3: Overview of DYCOR.

capturing short-term volatility, and AvgPool1d(\cdot) performs average pooling with kernel size k over the lookback window of each feature to smooth local variations.

To obtain latent representations of the features, we apply linear projections to both components \mathbf{X}_s^{tr} and \mathbf{X}_s^{fl} :

$$\mathbf{X}'_s = (\mathbf{W}_1 \mathbf{X}_s^{tr} + \mathbf{b}_1) + (\mathbf{W}_2 \mathbf{X}_s^{fl} + \mathbf{b}_2) \quad (2)$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{F \times F}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^F$ are learnable parameters. In this way, $\mathbf{X}'_s \in \mathbb{R}^{F \times T}$ captures both smoothed trends and short-term volatility of each stock s .

4.2.2 Time Mixing. We perform information exchange between different time steps of each feature within a lookback window of every stock s by visiting all possible prefixes of the lookback window, i.e., we ensure that information from later time steps does not leak into the earlier ones, more in line with the temporal nature [8]:

$$\hat{\mathbf{X}}_s = [\hat{\mathbf{x}}_{s,t-T}, \hat{\mathbf{x}}_{s,t-T+1}, \dots, \hat{\mathbf{x}}_{s,t-1}] = \text{TimeMixing}(\mathbf{X}'_s) \quad (3)$$

$$\hat{\mathbf{x}}_{s,t-T+p} = \text{MLP}_p([\mathbf{x}'_{s,t-T}, \dots, \mathbf{x}'_{s,t-T+p}]) \quad p \in \{0, 1, \dots, T-1\} \quad (4)$$

where MLP_p takes each prefix $[\mathbf{x}'_{s,t-T}, \dots, \mathbf{x}'_{s,t-T+p}] \in \mathbb{R}^{F \times p}$ from the latent feature matrix \mathbf{X}'_s as input, and generates a hidden unit $\hat{\mathbf{x}}_{s,t-T+p} \in \mathbb{R}^F$. The temporal representation $\hat{\mathbf{X}}_s \in \mathbb{R}^{F \times T}$ incorporates historical information up to every time step $p \in \{0, 1, \dots, T-1\}$. In other words, this representation integrates temporal patterns across varying horizons to capture evolving stock behaviors.

4.2.3 Indicator Mixing. Finally, we exchange information between hidden features of every time step for each stock s to take into account interactions among technical indicators in the temporal representation $\hat{\mathbf{X}}_s \in \mathbb{R}^{F \times T}$:

$$\text{IndicatorMixing}(\hat{\mathbf{X}}_s) = \hat{\mathbf{X}}_s + \mathbf{W}_4 \sigma(\mathbf{W}_3 \text{LayerNorm}(\hat{\mathbf{X}}_s)) \quad (5)$$

where $\mathbf{W}_3 \in \mathbb{R}^{F \times F}$ and $\mathbf{W}_4 \in \mathbb{R}^{F \times F}$ are learnable parameters, σ is a non-linear activation function such as Gaussian Error Linear Unit (GELU) [11] and LayerNorm [2] represents layer normalization.

In our implementation, we apply time mixing and indicator mixing to multiply the compressed sequences obtained from one-dimensional convolution over the lookback window with different kernel sizes, e.g., $j \in \{\frac{T}{2}, T\}$:

$$\bar{\mathbf{X}}_s^{(j)} = \text{IndicatorMixing}(\text{TimeMixing}(\text{Conv1d}_j(\mathbf{X}'_s))) \quad (6)$$

where Conv1d_j is one-dimensional convolution with a kernel size of j . Then a single-layer MLP aggregates all of F hidden features in the above representations $\bar{\mathbf{X}}_s^{(j)} \in \mathbb{R}^{F \times j}$, and produces the stock embedding $\mathbf{e}_s \in \mathbb{R}^d$ of each stock s where $d = \frac{T}{2} + T$:

$$\mathbf{e}_s = \text{MLP}([\bar{\mathbf{X}}_s^{(T/2)}, \bar{\mathbf{X}}_s^{(T)}]) \quad (7)$$

In summary, the resulting latent representation \mathbf{e}_s captures (i) mid-term trends and short-term fluctuations via stock trend decomposition, (ii) temporal dependencies learned through time mixing, and (iii) cross-indicator relationships modeled by indicator mixing.

4.3 Dynamic Stock Clustering

Stocks normally exhibit correlated behaviors within a sector or an industry group, whereas stock prices across the market often show correlated movements in response to certain political or economic events. To reflect these phenomena, this stage is designed to capture the group dynamics through data-driven stock clustering that adapts to varying market conditions by identifying and leveraging similar behavioral patterns among similar stocks. Unlike existing approaches that utilize predefined classification of industries/sectors to capture static correlations of stocks [6, 7, 9, 15, 16, 25, 26, 30, 35], our data-driven dynamic stock clustering is designed to capture the phenomena in which stocks exhibit complex interactions depending on market conditions beyond the static stock classifications.

4.3.1 Latent Segment Representations. From all stock embeddings obtained from the previous stage, we extract orthogonal representations that reflect diverse aspects of the market from different perspectives. Given all stock embeddings $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{|S|}] \in \mathbb{R}^{|S| \times d}$, we perform principal component analysis (PCA). Let a set $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\}$ of top- γ principal components be a smallest set of principal components determined by selecting the minimum number of components whose cumulative explained variance ratio exceeds γ as follows:

$$\frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^{|S|} \lambda_i} \geq \gamma \quad (8)$$

where λ_i denotes the eigenvalue corresponding to the i -th principal component, with components are sorted in descending order of explained variance.

4.3.2 Multi-Aspect Stock Clustering. To reflect the phenomenon that a stock is simultaneously influenced by multiple market factors to different extents, we adopt two clustering methods: (i) soft clustering to estimate how likely the stock is to align with each latent segment, and (ii) hard clustering to group stocks with similar patterns in a fine-grained manner based on the latent segment.

First, we associate each principal component \mathbf{p}_k with a cluster C_k of stocks. Let C_k be defined as latent segment k . Then our soft clustering strategy computes the membership probability of each stock s belonging to latent segment C_k :

$$P(C_k|s) = \frac{\exp(\text{sim}(\mathbf{e}_s, \mathbf{p}_k)/\tau)}{\sum_{k'=1}^K \exp(\text{sim}(\mathbf{e}_s, \mathbf{p}_{k'})/\tau)} \quad (9)$$

where $\text{sim}(\cdot, \cdot)$ stands for cosine similarity. The softmax operation with temperature parameter τ transforms K similarities into probabilities. This temperature parameter controls the sharpness of the distribution, with higher values of τ yielding softer probability distributions that allow stocks to be associated with multiple latent segments simultaneously.

Our hard clustering strategy further divides the stocks within each latent segment C_k into L latent sub-segments to correlate only the stocks affected by a latent segment to a similar extent in the subsequent stage. For this, we sort all stocks in C_k in descending order of $P(C_k|s)$, and divide them into L sub-clusters $C_{k,1}, C_{k,2}, \dots, C_{k,L}$ such that every sub-cluster includes the same number of stocks, where L is a hyperparameter. Intuitively, each latent sub-segment $C_{k,l}$ represents a group of stocks closely associated with each other. This fine-grained clustering allows us to detect subtle differences of

varying degrees of alignment of the stocks in C_k , which will offer more nuanced stock representations in the next step.

4.4 Intra-stock Correlation

For every latent sub-segment $C_{k,l}$, we compute the sub-segment-aware contextualized representation $\mathbf{H}_{k,l}$ of all stocks $s \in C_{k,l}$ based on their correlations by utilizing the self-attention mechanism:

$$\mathbf{Q} = \mathbf{E}_{k,l} \mathbf{W}_Q, \quad \mathbf{K} = \mathbf{E}_{k,l} \mathbf{W}_K, \quad \mathbf{V} = \mathbf{E}_{k,l} \mathbf{W}_V \quad (10)$$

$$\mathbf{H}_{k,l} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V} \quad (11)$$

where $\mathbf{E}_{k,l} \in \mathbb{R}^{|C_{k,l}| \times d}$ represents the concatenation of stock embeddings \mathbf{e}_s for all stocks $s \in C_{k,l}$, and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$ are learnable parameters.

From the perspective of each stock s , K contextualized representations are obtained by this stage: a contextualized representation from at least one sub-segment for every latent segment C_k , which is a soft cluster. These K contextualized representations $\{\mathbf{h}_s^{(k)}\}_{k \in \{1, \dots, K\}}$ can capture the stock relationships from multiple aspects: how much the stock is aligned with orthogonal latent segments in the embedding space, and how much the stock is aligned with the dynamics of the latent sub-sector of that stock through self-attention.

In practice, Equations (10) and (11) can be applied to every sub-segment of each segment C_k in parallel by calculating masked self-attention between all stocks in S , in which each stock attends to only the stocks in the latent sub-segment the stock belongs to.

4.5 Stock-wise Inter-cluster Aggregation

Stocks can be affected by multiple latent segments to different extents, e.g., a corporation in the semiconductors industry may behave somewhat similarly to other corporations in the same industry, and can be simultaneously affected by its customer corporations in other industries. Note that each stock is represented over its embedding spaces viewed from multiple unique perspectives of latent segments in the previous stage. In this stage, we aggregate the multiple representations of the stock obtained from various aspects by weighting them according to the influence of each aspect.

Given the contextualized representations $\{\mathbf{h}_s^{(k)}\}_{k \in \{1, \dots, K\}}$ of each stock s , we compute its final representation \mathbf{z}_s by summing up these representations weighted by their membership probabilities:

$$\mathbf{z}_s = \sum_{k=1}^K P(C_k|s) \cdot \mathbf{h}_s^{(k)} \quad (12)$$

We assign higher weights to perspectives from the latent segment with which the stock s more associated. Consequently, this method can naturally reflect the complex reality in which stocks simultaneously respond to multiple segments that exert influence on the stock to varying degrees.

4.6 Prediction

Finally, a two-layer MLP takes the final representation \mathbf{z}_s of each stock s to predict its return $\hat{r}_{s,t}$:

$$\hat{r}_{s,t} = \mathbf{W}_6 \cdot \sigma(\text{LayerNorm}(\mathbf{W}_5 \cdot \mathbf{z}_s + \mathbf{b}_5)) + \mathbf{b}_6 \quad (13)$$

	NASDAQ	NYSE	S&P 500
#(Stocks)	1,026	1,737	646
Start date	2013-01-02	2013-01-02	2003-01-01
End date	2017-12-08	2017-12-08	2023-12-14
#(Train. days)	756	756	3,775
#(Val. days)	252	252	503
#(Test days)	273	273	994

Table 2: Statistics of datasets used for evaluation.

where $\mathbf{W}_5 \in \mathbb{R}^{d \times d}$, $\mathbf{W}_6 \in \mathbb{R}^{d \times d}$, $\mathbf{b}_5 \in \mathbb{R}^d$, $\mathbf{b}_6 \in \mathbb{R}^d$ are learnable parameters, and σ stands for an activation function, e.g., GELU.

4.7 Correlation-aware Training

Most existing stock trend prediction methods jointly adopt both regression loss such as MSE and ranking loss [26, 43]. However, ranking loss may overlook the directional consistency and relative magnitude of stock returns, which are critical to capture meaningful correlations between predicted and actual stock movements. Ranking loss focuses on maintaining relative order between stocks, but its loss function $\sum_{u \in S} \sum_{v \in S} \max(0, -(r_{u,t} - r_{v,t})(\hat{r}_{u,t} - \hat{r}_{v,t}))$ incurs no penalty at all, when the differences $(\hat{r}_{u,t} - \hat{r}_{v,t})$ for predicted returns and $(r_{u,t} - r_{v,t})$ for ground-truth returns have the same sign. This leads to the inability to reflect significant magnitude discrepancies even when the two differences are directionally consistent but substantially mismatched in magnitude.

To tackle the above potential shortcoming, we train our method based on composite loss between predicted returns $\{\hat{r}_{s,t}\}_{s \in S}$ and ground-truth returns $\{r_{s,t}\}_{s \in S}$:

$$\mathcal{L} = (1 - \lambda_{corr}) \cdot \mathcal{L}_{reg} + \lambda_{corr} \cdot \mathcal{L}_{corr} \quad (14)$$

where \mathcal{L}_{reg} represents Huber loss [14] which is robust to outliers, correlation-aware loss \mathcal{L}_{corr} is defined below, and hyperparameter λ_{corr} is the weight that balances the two losses. We empirically determined the value of λ_{corr} to be 0.65.

$$\mathcal{L}_{corr} = 1 - \text{corr}(\{\hat{r}_{s,t}\}_{s \in S}, \{r_{s,t}\}_{s \in S}) \quad (15)$$

We represent corr as the Pearson correlation coefficient (PCC) between $\{\hat{r}_{s,t}\}_{s \in S}$ and ground-truth returns $\{r_{s,t}\}_{s \in S}$ below:

$$\text{corr}(\{\hat{r}_{s,t}\}_{s \in S}, \{r_{s,t}\}_{s \in S}) = \frac{\sum_{s \in S} (\hat{r}_{s,t} - \hat{\mu}_t)(r_{s,t} - \mu_t)}{\hat{\sigma}_t \cdot \sigma_t} \quad (16)$$

where $\hat{\mu}_t$ and $\hat{\sigma}_t$ denote the mean and standard deviation of the predicted returns, respectively, while μ_t and σ_t denote those of the ground-truth returns.

The correlation-aware loss \mathcal{L}_{corr} is designed to maximize PCC by encouraging accurate capture of relative relationships among the returns. For each stock s at each time step t , the standard score $(r_{s,t} - \mu_t)/\sigma_t$ based on the ground-truth return in PCC serves as a per-stock loss weight, so the model focuses more on predicting high or low returns for stocks that largely deviate from the mean relative to the standard deviation. These mathematical properties ensure that the model learns both the relative ordering and the proportional relationships between returns especially for top- and bottom-performing stocks, which will be validated in Figure 7b.¹

¹We conducted experiments using joint loss that combines three aforementioned losses, i.e., Huber loss, ranking loss, and correlation-aware loss, but these competing objectives not only increase a training overhead but also result in inferior performance.

5 Experiments

5.1 Experiment Settings

5.1.1 Datasets. For evaluation, we use three real-world U.S. stock market datasets, i.e., NASDAQ, NYSE, and S&P 500, summarized in Table 2. The NASDAQ and NYSE datasets were curated by [9]. The S&P 500 dataset was sourced from Yahoo Finance [36]. All datasets are preprocessed to remove abnormal patterns and penny stocks, ensuring data quality and representativeness [9].

5.1.2 Evaluation Metrics. The metrics below are adopted for evaluation.²

- **Information Coefficient (IC)** is the Pearson correlation coefficient between predictions and targets averaged over all time steps in the test period to evaluate their linear relations.
- **Rank Information Coefficient (Rank IC)** is the Spearman correlation coefficient between predictions and targets averaged over all time steps in the test period to evaluate the linear relations between the ranks of the predictions and the targets.
- **Information Ratio of IC (ICIR)** is a normalized metric of IC, i.e., IC divided by its standard deviation.
- **Information Ratio of Rank IC (Rank ICIR)** is a normalized metric of Rank IC, i.e., Rank IC divided by its standard deviation.
- **Recall@n** is the proportion of top- n ground-truth stocks that are correctly identified within the top- n predictions.
- **NDCG@n** is a measure that evaluates both the relevance and the ranking quality of the top- n predictions.

5.1.3 Baselines. We compare our method not only with widely-used models such as LSTM [12] and Transformer [28], but also with state-of-the-art time-series forecasting models such as TimeMixer [33] and iTransformer [21], as well as the following state-of-the-art stock trend prediction models:

- **RSR** [9]: learns to rank stock returns, and utilizes temporal graph convolution operations to model dynamic stock relations.
- **DTML** [39]: models asymmetric and dynamic stock correlations by combining temporal patterns within individual stocks and their pairwise inter-stock relationships using Transformer.
- **STHAN-SR** [26]: captures stock inter-dependencies and temporal patterns of price movements through *hypergraph attention*.
- **StockMixer** [8]: adopts an MLP-based architecture which performs indicator mixing, time mixing, and stock mixing to model stock-to-market and market-to-stock influences.
- **CI-STHPAN** [34]: incorporates Transformer and hypergraph attention network that applies to channel-independent hypergraphs constructed from pairwise inter-stock similarities, modeling both intra- and inter-stock correlations.
- **MATCC** [4]: decomposes given stock features into long-term market trends and fluctuations, and models cross-time correlations from both time and stock dimensions.

²While these metrics are indicative, they can be misleading when ground-truth return ratios for certain stocks are missing at some time steps during the test period. In prior work, such missing values are imputed as zeros, which allows them to be included in metric calculations, potentially overestimating the ranking performance. [8, 9, 18, 26]. To mitigate this issue, we compute the evaluation metrics using only the return ratios that are available at every time step within the test period. We refer to this evaluation setting as *filtered*.

Model	NASDAQ				NYSE				S&P 500			
	IC	Rank IC	ICIR	Rank ICIR	IC	Rank IC	ICIR	Rank ICIR	IC	Rank IC	ICIR	Rank ICIR
LSTM	0.012 (0.004)	0.023 (0.006)	0.125 (0.043)	0.218 (0.054)	0.002 (0.001)	0.010 (0.003)	0.015 (0.009)	0.081 (0.029)	0.009 (0.003)	0.011 (0.002)	0.058 (0.021)	0.066 (0.013)
Transformer	0.019 (0.003)	0.031 (0.002)	0.196 (0.032)	0.299 (0.025)	0.003 (0.011)	0.004 (0.016)	0.007 (0.088)	0.020 (0.108)	0.008 (0.010)	0.005 (0.006)	0.036 (0.045)	0.020 (0.025)
TimeMixer	0.009 (0.007)	0.007 (0.008)	0.102 (0.081)	0.063 (0.070)	0.003 (0.006)	0.003 (0.005)	0.037 (0.075)	0.028 (0.048)	0.004 (0.004)	0.006 (0.004)	0.029 (0.026)	0.034 (0.023)
iTransformer	0.018 (0.009)	0.017 (0.012)	0.210 (0.094)	0.176 (0.121)	0.014 (0.001)	0.013 (0.003)	0.117 (0.023)	0.104 (0.026)	0.012 (0.005)	0.003 (0.005)	0.067 (0.023)	0.013 (0.026)
RSR	0.001 (0.001)	0.013 (0.004)	0.012 (0.016)	0.122 (0.035)	0.006 (0.003)	0.008 (0.005)	0.066 (0.028)	0.066 (0.037)	0.016 (0.001)	0.009 (0.001)	0.093 (0.009)	0.043 (0.003)
DTML	0.022 (0.003)	0.031 (0.001)	0.222 (0.029)	0.301 (0.008)	0.014 (0.006)	0.020 (0.003)	0.101 (0.060)	0.135 (0.028)	0.001 (0.003)	-0.003 (0.003)	0.006 (0.018)	-0.016 (0.017)
STHAN-SR	0.003 (0.001)	0.009 (0.002)	0.041 (0.010)	0.082 (0.017)	0.000 (0.000)	0.016 (0.000)	0.047 (0.003)	0.125 (0.002)	0.004 (0.001)	0.001 (0.003)	0.029 (0.005)	0.007 (0.014)
StockMixer	0.021 (0.009)	0.031 (0.011)	0.265 (0.120)	0.366 (0.168)	0.018 (0.003)	0.011 (0.003)	0.213 (0.034)	0.140 (0.024)	0.016 (0.004)	0.018 (0.007)	0.082 (0.020)	0.089 (0.042)
CI-STHPAN	0.009 (0.002)	0.018 (0.003)	0.106 (0.024)	0.172 (0.025)	0.015 (0.004)	0.021 (0.002)	0.122 (0.038)	0.193 (0.019)	0.008 (0.002)	0.010 (0.001)	0.040 (0.010)	0.048 (0.007)
MATCC	0.024 (0.005)	0.028 (0.005)	0.264 (0.058)	0.269 (0.047)	0.015 (0.007)	0.019 (0.005)	0.132 (0.059)	0.151 (0.047)	-0.006 (0.009)	-0.005 (0.011)	-0.035 (0.072)	-0.031 (0.079)
DYCOR w/o DC	0.025 (0.004)	0.029 (0.004)	0.313 (0.049)	0.304 (0.050)	0.014 (0.005)	0.016 (0.003)	0.131 (0.068)	0.122 (0.050)	0.020 (0.003)	0.014 (0.004)	0.081 (0.017)	0.054 (0.024)
DYCOR w/o IC	0.030 (0.002)	0.033 (0.004)	0.324 (0.027)	0.308 (0.035)	0.023 (0.001)	0.023 (0.004)	0.225 (0.016)	0.245 (0.019)	0.019 (0.004)	0.015 (0.001)	0.084 (0.022)	0.053 (0.009)
DYCOR w/o IA	0.031 (0.002)	0.033 (0.002)	0.337 (0.032)	0.321 (0.026)	0.024 (0.004)	0.022 (0.003)	0.310 (0.039)	0.293 (0.031)	0.022 (0.001)	0.015 (0.000)	0.096 (0.006)	0.077 (0.004)
DYCOR w/o \mathcal{L}_{corr}	0.023 (0.003)	0.032 (0.003)	0.264 (0.027)	0.329 (0.025)	0.017 (0.002)	0.020 (0.004)	0.178 (0.036)	0.201 (0.063)	0.015 (0.001)	0.015 (0.000)	0.069 (0.003)	0.064 (0.002)
DYCOR	0.037 (0.001)	0.036 (0.001)	0.390 (0.012)	0.366 (0.012)	0.028 (0.003)	0.027 (0.003)	0.358 (0.065)	0.347 (0.050)	0.024 (0.001)	0.016 (0.000)	0.119 (0.002)	0.079 (0.001)

Table 3: A comparison of different methods on NASDAQ, NYSE, and S&P 500, along with an ablation study of DYCOR. The best results are in bold, and the second-best results are underlined. Values in parentheses represent the standard deviation over 5 runs.

Model	NASDAQ		NYSE		S&P 500	
	Recall@ n	NDCG@ n	Recall@ n	NDCG@ n	Recall@ n	NDCG@ n
LSTM	0.073 (0.002)	0.080 (0.002)	0.073 (0.001)	0.075 (0.001)	0.083 (0.017)	0.092 (0.017)
Transformer	0.081 (0.001)	0.079 (0.001)	0.068 (0.013)	0.073 (0.010)	0.005 (0.006)	0.006 (0.006)
TimeMixer	0.026 (0.004)	0.026 (0.004)	0.017 (0.002)	0.017 (0.003)	0.033 (0.008)	0.032 (0.010)
iTransformer	0.032 (0.011)	0.033 (0.012)	0.036 (0.012)	0.038 (0.014)	0.035 (0.016)	0.037 (0.015)
RSR	0.080 (0.002)	0.081 (0.002)	0.068 (0.004)	0.077 (0.005)	0.123 (0.004)	0.130 (0.003)
DTML	0.082 (0.001)	0.082 (0.002)	0.074 (0.008)	0.083 (0.007)	0.086 (0.010)	0.100 (0.009)
STHAN-SR	0.076 (0.001)	0.075 (0.001)	0.080 (0.001)	0.088 (0.001)	0.116 (0.002)	0.128 (0.003)
StockMixer	0.066 (0.012)	0.073 (0.011)	0.068 (0.003)	0.073 (0.003)	0.103 (0.010)	0.115 (0.010)
CI-STHPAN	0.000 (0.000)	0.000 (0.000)	0.014 (0.027)	0.014 (0.026)	0.084 (0.002)	0.092 (0.003)
MATCC	0.056 (0.004)	0.062 (0.003)	0.039 (0.005)	0.044 (0.006)	0.076 (0.020)	0.081 (0.021)
DYCOR	0.084 (0.002)	0.093 (0.001)	0.082 (0.008)	0.084 (0.007)	0.126 (0.005)	0.138 (0.004)

Table 4: A comparison of different methods on NASDAQ, NYSE, and S&P 500 in terms of Recall@ n and NDCG@ n .

5.1.4 Implementation Details. For all experiments, we report results averaged over five independent runs. We conduct hyper-parameter tuning through grid search. We explore the lookback window length $T \in \{8, 16, 32, 64\}$, with $T = 16$ yielding the best performance across all datasets. For dynamic stock clustering, we explore the number of sub-clusters $L \in \{1, 2, 4, 6, 8\}$, and select $L = 4$ for NASDAQ and S&P 500, and $L = 6$ for NYSE. We test thresholds $\gamma \in \{0.87, 0.9, 0.93, 0.95\}$ for the cumulative explained variance ratio used in PCA, and select $\gamma = 0.93$, which delivers optimal performance across all datasets, ensuring that principal components are selected to explain 93% of the data variance while maintaining the model efficiency. We test the softmax temperature parameters $\tau \in \{0.1, 0.2, 0.5, 1.0\}$, and select $\tau = 0.2$ that consistently delivers the best performance in all datasets, balancing the sharpness of cluster assignments with the flexibility to capture complex market relationships. Our method is trained with the Adam optimizer, exploring learning rates $\{1e-4, 5e-4, 1e-3, 1e-2\}$, with $1e-4$ consistently performing best. All experiments are conducted using PyTorch 2.4.1 with CUDA 12.1. The hardware configuration consists of an Intel i9-10980XE CPU, dual NVIDIA RTX 4090 GPUs, and 256GB RAM. The source code of DYCOR is available³.

5.2 Performance Comparison

Table 3 presents the performance of DYCOR and baselines across all datasets. DYCOR demonstrates superior performance on most evaluation metrics. DYCOR achieves the highest performance in

all metrics on NASDAQ, with significant improvements of approximately 54% in IC over the best baseline MATCC, and 16% in Rank IC over DTML. ICIR and Rank ICIR also show notable enhancements, indicating reliable prediction capability of DYCOR in varying markets. For NYSE, DYCOR outperforms in every metric, with IC and ICIR improving by approximately 56% and 68% over the best baseline StockMixer. On S&P 500, DYCOR achieves the best performance in IC and ICIR with improvements of 50% and 42% over the best baseline RSR. While ranking second in Rank IC and Rank ICIR behind StockMixer in S&P 500, DYCOR outperforms the majority of baselines, and maintains competitive performance.

Most baseline methods struggle to perform well on the S&P 500 dataset due to its higher volatility and extended time frame, i.e., 3,775 training days, which contribute to the increased variability inherent in this long-term period. The larger time span introduces more market shifts, making it more challenging for models to maintain consistent predictive power over time, as evidenced by the higher volatility of S&P 500 (i.e., 0.0236) compared to those of NASDAQ (i.e., 0.0205) and NYSE (i.e., 0.0164).

Table 4 further confirms the effectiveness of DYCOR with retrieval metrics. DYCOR leads in both Recall@ n and NDCG@ n on NASDAQ and S&P 500. We evaluate these metrics with $n \in \{5, 10, 20\}$, and Table 4 presents the results for $n = 20$. The performance trends remain consistent across all values of n .

5.3 Ablation Study

To verify the contribution of key components in our method, we conduct an ablation study by removing different components. Table 3 presents performance comparisons between DYCOR and its variants with specific components removed. First, the variant w/o DC eliminates dynamic stock clustering and performs self-attention across all stocks without any grouping. This variant shows notable performance degradation across all datasets, with IC decreasing by 32.4%, 50.0%, and 16.7% on NASDAQ, NYSE, and S&P 500, respectively. The most dramatic performance drop occurs in NYSE, which contains the largest number of stocks. This finding indicates that dynamic stock clustering plays a crucial role as the market size grows. Second, the variant w/o IC removes intra-stock correlation, instead directly using stock embeddings in stock-wise inter-cluster aggregation. This variant shows moderate performance degradation. The

³<https://github.com/FinancialDeepLearning/DYCOR>

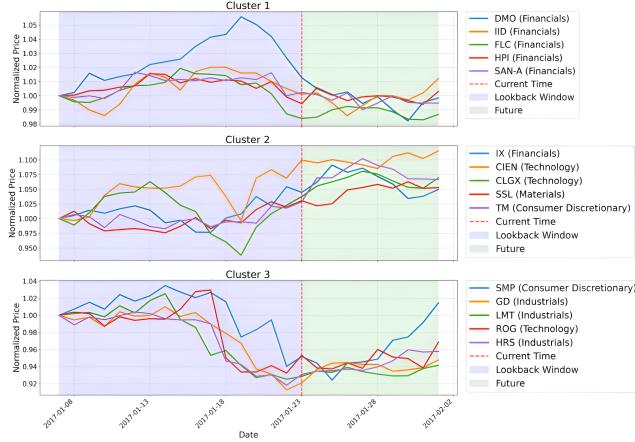


Figure 4: Normalized prices of major stocks in each cluster of NYSE discovered by DYCOR. The blue-shaded area represents the lookback window, the red vertical dashed line marks its final time point, and the green-shaded area indicates the corresponding ground-truth future prices.

results indicate that modeling correlations between stocks within the same cluster contributes to accuracy. The variant w/o IA removes stock-wise inter-cluster aggregation, instead using uniform weights $1/K$ to aggregate representations from different clusters, showing minor performance drops. The variant w/o \mathcal{L}_{corr} uses only Huber loss and ranking loss, and it shows consistent performance degradation on all datasets, with IC decreasing by 37.8%, 39.3%, and 37.5% on NASDAQ, NYSE, and S&P 500, respectively. The substantial performance drop highlights the importance of incorporating supervision of stock correlations in training.

5.4 Analysis

5.4.1 Analysis of Dynamic Stock Clustering. We evaluate the quality of the dynamic stock clustering results using Dynamic Time Warping (DTW): (1) for each pair of stocks within a cluster, we compute the mean DTW distance (intra-cluster distance) across their 16-day lookback and 10-day future windows of stock prices, and (2) for each pair of stocks from different clusters, we calculate their DTW distance (inter-cluster distance). The results show that intra-cluster distances are consistently lower than inter-cluster distances across all markets, in both the lookback and future windows, indicating stronger similarity within clusters.

Figure 4 visualizes the prices of the top-5 stocks s most similar to each principal component p_k based on $\text{sim}(e_s, p_k)$ at a timestamp randomly selected from the test period of NYSE. Here, the prices of each stock are normalized relative to the price of that stock at the first timestamp of the lookback window. The figure demonstrates that stocks within each cluster exhibit similar price patterns, while stocks from different clusters show distinctly contrasting movement patterns. Notably, these stocks with similar movements belong to different industry sectors, supporting our claim that stock relationships extend beyond traditional industry classifications and evolve dynamically. This cross-industry composition within cluster is further illustrated in Figure 5, which shows the diverse distribution of industry sectors within Cluster 3.

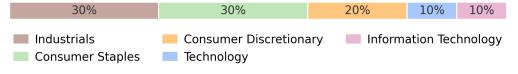


Figure 5: The distribution of industry sectors within Cluster 3.

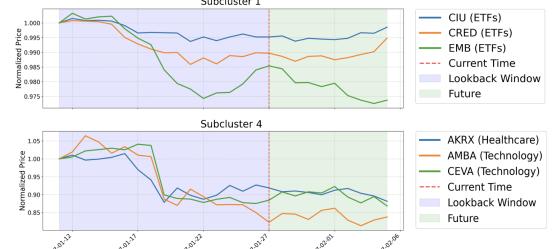


Figure 6: Normalized prices of major stocks from different sub-clusters within a specific NASDAQ cluster discovered by DYCOR.

Further analysis of Figure 4 demonstrates that dynamic stock clustering in DYCOR effectively captures latent business relationships, supply chain connections, or similar exposures to external economic factors that are difficult to identify through predefined and static industry classifications in existing methods. First, Cluster 1 illustrates the tendency of stocks within the same industry sector to group together, e.g., DMO, IID, FLC, HPI, and SAN-A all belonging to Financials. Second, Cluster 2 and Cluster 3 contain stocks from multiple distinct industry sectors, e.g., SMP from Consumer Discretionary, GD, LMT, HRS from Industrials, and ROG from Technology in Cluster 3. Despite their different sector classifications, these stocks exhibit remarkably similar price movement patterns. Analyzing these relationships more deeply reveals hidden business connections that transcend traditional industry boundaries. GD, LMT, and HRS are major defense contractors developing and manufacturing military equipment and technologies. ROG, a specialized materials and electronic components manufacturer, likely supplies critical components to these defense companies. Although classified under consumer discretionary, SMP specializes in engine management systems, ignition and emissions components, and sensor technologies, all of which have potential applications in military vehicles and equipment. These may share technical specifications with defense industry requirements, potentially positioning the company within the defense supply chain.

Furthermore, Figure 4 shows that stocks within each cluster maintain similar movements in the future, i.e., the green-shaded area, so DYCOR may provide valuable insight not only for single-day prediction but also for mid-term forecasting tasks. Although these results suggest potential directions for future research in extending our approach to multi-step prediction, this work focuses solely on single-step prediction, in line with existing state-of-the-art methods to enable direct comparison.

Figure 6 illustrates the normalized price movements of stocks which belong to different sub-clusters but are highly similar to a randomly-selected cluster in NASDAQ. While both sub-clusters show an overall downward trend, they show distinct behaviors around mid-January. Sub-cluster 1 exhibits a smooth and gradual decline, whereas sub-cluster 4 experiences a sharp drop. This contrast highlights the effectiveness of our sub-clustering strategy in capturing differences in movements of its stocks in a fine-grained

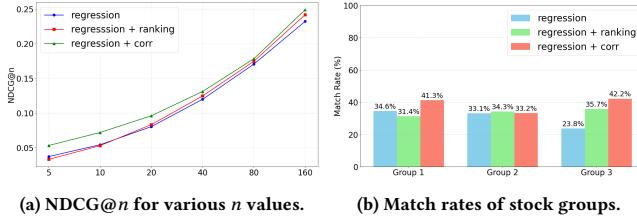


Figure 7: Performance comparison of different loss function configurations on NASDAQ.

manner, though the stocks display similar overall movements. These subtle differences are effectively modeled by intra-stock correlation. For every time step, we compute the variance of the next-day return ratios of all stocks in S to analyze how market volatility affects clustering results. We observe that DYCOR forms more clusters as variance increases, while achieving high IC and Rank IC under medium variance.

5.4.2 Effectiveness of Correlation-aware Loss. Figure 7a compares NDCG@ n of DYCOR and its variants supervised by different loss configurations: (1) regression, i.e., a regression loss only, and (2) regression + ranking, i.e., a joint loss combining regression and ranking objectives, and (3) regression + corr (DYCOR), i.e., a joint loss combining regression and correlation-aware objectives. Using only regression loss leads to suboptimal performance due to insufficient supervision of inter-stock dependencies, while regression + ranking does not fully capture the differences in magnitude if prediction and ground truth returns share the same sign. Indeed, the performance gap between regression + corr and regression + ranking becomes more pronounced for smaller n , i.e., NDCG@5 and NDCG@10. This confirms that regression + ranking struggles to predict higher returns for top- n stocks than for the other stocks. In contrast, regression + corr provides better guidance for identifying high-impact stocks. These results empirically validate the limitations discussed in Section 4.7. To further validate the effectiveness of correlation-aware loss, we sort all stocks in descending order of their predicted returns for a randomly-selected lookback window, and partition them into three groups with equal depth. For all stocks in each group, we calculate the percentage of the stocks correctly classified into that group based on their ground truth trends. Figure 7b shows the match rates for the three loss configurations, and regression + corr achieves the highest match rate on Groups 1 and 3. This may stem from the implication of maximizing PCC, in which each stock's contribution is proportional to $(\hat{r}_{s,t} - \mu_{\hat{r},t})(r_{s,t} - \mu_{r,t})$. Stocks with their ground truth returns far from their mean contribute more heavily to PCC when their predictions closely match the ground truths in both direction and magnitude.

5.5 Time Efficiency

To evaluate computational efficiency, we report the training time per epoch and the inference time of the competing methods on the NASDAQ dataset under identical hardware settings. As shown in Table 5, DYCOR achieves fast inference while maintaining strong predictive performance, and shows favorable training efficiency with significantly lower per-epoch time than state-of-the-art competitors such as CI-STHPAN and MATCC.

Model	Training (sec/epoch)	Inference (sec)
LSTM	17.58	2.73
Transformer	45.50	3.37
TimeMixer	82.16	4.71
iTransformer	21.67	5.07
RSR	34.70	2.70
STHAN-SR	11.82	0.71
StockMixer	54.92	3.15
CI-STHPAN	47.89	7.35
MATCC	48.27	13.59
DYCOR	22.48	2.99

Table 5: The training and inference time on the NASDAQ dataset.

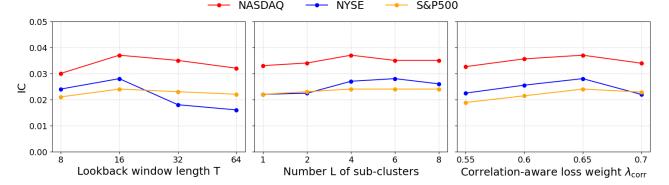


Figure 8: The sensitivity to hyperparameters T , L , and λ_{corr} .

5.6 Hyperparameter Sensitivity

Figure 8 shows the results for sensitivity analysis varying the values of three key hyperparameters: length T of a lookback window, the number L of sub-clusters, and the correlation-aware loss weight λ_{corr} . All three datasets show optimal performance at a lookback window length of $T = 16$. DYCOR with $L = 6$ reaches its highest IC of 0.028 on NYSE, and DYCOR with $L = 4$ achieves peak performance with IC of 0.037 and 0.024 on NASDAQ and S&P 500, respectively. We examine the sensitivity to the correlation-aware loss weight λ_{corr} . Its optimal value is $\lambda_{corr} = 0.65$ on all datasets.

6 Conclusion

We propose DYCOR that addresses two limitations of existing stock trend prediction methods: (1) prior approaches using the pre-defined static relationships struggle to capture the dynamic and cross-sectoral relationships observed in real-world stock markets, and (2) while MSE penalizes errors and pairwise ranking-aware loss aims to relative order between stocks, both do not fully capture how closely predicted stock trends correlate with ground-truth stock trends. DYCOR harmonizes dynamic stock clustering, intra-stock correlation, and stock-wise inter-cluster aggregation to effectively represent the time-varying correlations. Correlation-aware training encourages DYCOR to align its predicted trends with the relative movements of ground-truth trends, thereby capturing directional consistency over time. Extensive experiments demonstrate superior performance of DYCOR. DYCOR successfully captures complex stock relationships, encompassing both within-sector and across-sector co-movements, transcending traditional boundaries between sectors. Future research will focus on expanding DYCOR to multi-step prediction and assessing its performance in trading applications.

Acknowledgments

This work was partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2020-II201373, Artificial Intelligence Graduate School Program(Hanyang University)) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2022R1G1A1011645).

GenAI Usage Disclosure

No GenAI tools were used in any stage of the research, nor in the writing.

References

- [1] Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. 2014. Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*. IEEE, 106–112.
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer Normalization. *arXiv preprint arXiv:1607.06450* (2016).
- [3] Wei Bao, Jun Yue, and Yulei Rao. 2017. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one* 12, 7 (2017), e0180944.
- [4] Zhiyuan Cao, Jiayu Xu, Chengqi Dong, Peiwen Yu, and Tian Bai. 2024. MATCC: A Novel Approach for Robust Stock Price Prediction Incorporating Market Trends and Cross-time Correlations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 187–196.
- [5] Kai Chen, Yi Zhou, and Fangyan Dai. 2015. A LSTM-based method for stock returns prediction: A case study of China stock market. In *2015 IEEE international conference on big data (big data)*. IEEE, 2823–2824.
- [6] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. 2018. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 1655–1658.
- [7] Chaoran Cui, Xiaojie Li, Chunyun Zhang, Weili Guan, and Meng Wang. 2023. Temporal-relational hypergraph tri-attention networks for stock trend prediction. *Pattern recognition* 143 (2023), 109759.
- [8] Jinyong Fan and Yanyan Shen. 2024. StockMixer: A Simple Yet Strong MLP-Based Architecture for Stock Price Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8389–8397.
- [9] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. 2019. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–30.
- [10] Huyen Giang Thi Thu, Thuy Nguyen Thanh, and Tai Le Quy. 2022. Dynamic sliding window and neighborhood LSTM-based model for stock price prediction. *SN Computer Science* 3, 3 (2022), 256.
- [11] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). In *arXiv preprint arXiv:1606.08415*.
- [12] S Hochreiter. 1997. Long Short-term Memory. *Neural Computation MIT-Press* (1997).
- [13] Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. 2018. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 261–269.
- [14] Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*. Springer, 492–518.
- [15] Thanh Trung Huynh, Minh Hieu Nguyen, Thanh Tam Nguyen, Phi Le Nguyen, Matthias Weidlich, Quoc Viet Hung Nguyen, and Karl Aberer. 2023. Efficient integration of multi-order dynamics and internal dynamics in stock movement prediction. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 850–858.
- [16] Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. 2019. Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999* (2019).
- [17] Kelvin JL Koa, Yunshan Ma, Ritchie Ng, and Tat-Seng Chua. 2023. Diffusion variational autoencoder for tackling stochasticity in multi-step regression stock price prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1087–1096.
- [18] Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. 2024. MASTER: Market-Guided Stock Transformer for Stock Price Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 162–170.
- [19] Tao Lin, Tian Guo, and Karl Aberer. 2017. Hybrid neural networks for learning the trend in time series. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence*. 2273–2279.
- [20] Guang Liu, Yuzhao Mao, Qi Sun, Hailong Huang, Weiguo Gao, Xuan Li, Jianping Shen, Ruifan Li, and Xiaojie Wang. 2020. Multi-scale Two-way Deep Neural Network for Stock Trend Prediction.. In *IJCAI* 4555–4561.
- [21] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).
- [22] Tashreef Muhammad, Anika Binteet Aftab, Muhammad Ibrahim, Md Mainul Ah-san, Maishameem Meherin Muhu, Shahidul Islam Khan, and Mohammad Shafiqul Alam. 2023. Transformer-based deep learning model for stock price prediction: A case study on Bangladesh stock market. *International Journal of Computational Intelligence and Applications* 22, 03 (2023), 2350013.
- [23] NASDAQ. n.d.. Industry Screening. <https://www.nasdaq.com/SCREENING/INDUSTRIES.ASPX> Accessed: February 10, 2025.
- [24] David MQ Nelson, Adriano CM Pereira, and Renato A De Oliveira. 2017. Stock market's price movement prediction with LSTM neural networks. In *2017 International joint conference on neural networks (IJCNN)*. Ieee, 1419–1426.
- [25] Hao Qian, Hongting Zhou, Qian Zhao, Hao Chen, Hongxiang Yao, Jingwei Wang, Ziqi Liu, Fei Yu, Zhiqiang Zhang, and Jun Zhou. 2024. Mdgnn: Multi-relational dynamic graph neural network for comprehensive and dynamic stock investment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 14642–14650.
- [26] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. 2021. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 497–504.
- [27] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kannainen, Moncef Gabbouj, and Alexandros Isidoris. 2017. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th conference on business informatics (CBI)*, Vol. 1. IEEE, 7–12.
- [28] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).
- [29] Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57, 10 (2014), 78–85.
- [30] Heyuan Wang, Tengjiao Wang, Shun Li, and Shijie Guan. 2022. HATR-I: Hierarchical adaptive temporal relational interaction for stock trend prediction. *IEEE Transactions on Knowledge and Data Engineering* 35, 7 (2022), 6988–7002.
- [31] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. 2022. Adaptive Long-Short Pattern Transformer for Stock Investment Selection.. In *IJCAI*. 3970–3977.
- [32] 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)*, Vol. 4. IEEE, 2160–2165.
- [33] Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. 2024. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616* (2024).
- [34] Hongjie Xia, Huijie Ao, Long Li, Yu Liu, Sen Liu, Guannan Ye, and Hongfeng Chai. 2024. CI-STHPAN: Pre-trained Attention Network for Stock Selection with Channel-Independent Spatio-Temporal Hypergraph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 9187–9195.
- [35] Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. Hist: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. *arXiv preprint arXiv:2110.13716* (2021).
- [36] Yahoo Finance. 2024. Yahoo Finance. <https://finance.yahoo.com/>
- [37] Linyi Yang, Jiazheng Li, Ruihai Dong, Yue Zhang, and Barry Smyth. 2022. Numhtml: Numeric-oriented hierarchical transformer model for multi-task financial forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11604–11612.
- [38] Nan Yin, Fuli Feng, Zhigang Luo, Xiang Zhang, Wenjie Wang, Xiao Luo, Chong Chen, and Xian-Sheng Hua. 2022. Dynamic hypergraph convolutional network. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1621–1634.
- [39] Jaemin Yoo, Yejun Soun, Yong-chan Park, and U Kang. 2021. Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2037–2045.
- [40] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.
- [41] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2141–2149.
- [42] Yu Zhao, Huaming Du, Ying Liu, Shaopeng Wei, Xingyan Chen, Fuzhen Zhuang, Qing Li, and Gang Kou. 2022. Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks. *IEEE transactions on knowledge and data engineering* 35, 8 (2022), 8559–8571.
- [43] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. 2007. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 287–294.