Agree to Disagree: Personalized Temporal Embedding and Routing for Stock Forecast

Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Weijun Chen, and Wei Chen

***Abstract*—Stock forecast is a crucial yet challenging task in modern quantitative trading. Given theoretical and investment merits, recently a variety of deep learning methods have been proposed for automatically simulating stock movements from historical time series. However, these methods typically follow the i.i.d. assumption that actually contradicts the complex trading environment. In reality, individual stocks often exhibit diverse volatility patterns, while macro market scenarios may also change over time, jointly resulting in distribution shifts and weak generalization. To combat these bottlenecks, in this paper we propose a new learning architecture called *Personalized Temporal Embedding and Routing* (*PTER*) to improve stock forecast by forming a relaxed weight-sharing paradigm. The key of PTER is introducing hypernetworks to guide tailoring target network parameters, such that stock time series are embedded adapting to multi-object multi-scenario data disparities. Specifically, in the encoding stage, PTER first captures hyper-knowledge characterizing the similarity and peculiarity of different stocks and market scenarios. The knowledge space is then projected onto the temporal parameter space, enabling the customization of protruded features from chaotic observation signals. In the inference stage, each sample is dispatched to orthogonal predictor heads to dynamically output expected returns based on market conditions. Through experiments on benchmark datasets spanning over five years on four of the world's largest exchange markets, we show that PTER improves the cumulative and risk-adjusted revenue performance by a significant margin.**

***Index Terms*—Intelligent finance, stock forecast, distribution shift, hypernetwork.**

I. Introduction

AMONG a myriad of investment channels, the stock market has achieved continuous development, with global capitalization exceeding $93 trillion as of 2020.[1] Tallying with early Efficient Market Hypothesis [1] and its critics, many studies have found that real-world stock markets are incompletely efficient and can be predicted to a certain extent. In effect, stock forecast has become a fundamental yet challenging task in the *Fintech* field. Through data-driven analysis of quote and auxiliary information, it benefits market organizations and investors in various ways such as robo-advisory, intelligent monitoring, and risk control [2], [3], [4].

Formally, most existing stock prediction approaches strive to train a single parameterized model $\mathcal{F}_\Theta$ to simulate the patterns of stock fluctuations, i.e., mapping per stock's observation features $\boldsymbol{x}_i \in \mathcal{X}$ into a predicted label $y_i \in \mathcal{Y}$ toward the objectives of future price regression, classification, or expected return ranking. To pursue superior predictive ability, a variety of model architectures including RNNs [5], GNNs [6], Attention [7], Transformer [8] etc, are employed to instantiate $\mathcal{F}_\Theta$. However, despite technically feasible and effective in some scenarios, many methods are vulnerable to weak generalization [9], [10]. Through evidence analyses, we attribute this issue to the fact that the complex financial ecosystem involving blended trading patterns can invalidate the strict *independent and identically distributed* (*i.i.d.*) assumption. Our work is motivated by both microscopic and macroscopic perspectives of the real-world stock market. At the micro-level, influenced by latent factors such as shareholder behavior and hotspot reaction, constituent stock objects of listed firms often exhibit diverse volatility characteristics. As an example, Fig. 1(a) plots the histograms illustrating the deviation distributions of daily volatility of two stocks relative to the market index, spanning over a five-year period from 2016 to 2020. It can be observed that *Goodix* manifests a high potential of bucking the global market's trend, while *Huaxia Bank* is a tight market-follower, indicating their extremely distinct investment opportunities. Consequently, an undifferentiated feature encoder is far from adequate and credible in supporting the varied representation of individual stock traits. Apart from micro-object diversity, the macro-market scenarios may also shift over time because of non-stationary economic cycles [10], [11]. Striking evidence can be found in classic asset pricing theories, e.g., the *Momentum Effect* [12] and *Reversal Effect* [13] coexist but hold exactly opposite beliefs about tendency persistence or reversal, the *Merrill Lynch Investment Clock* [14] reports cyclical inflections in market returns. Fig. 1(b) displays the leading fund allocation styles in the A-share market, which unveils an overall rotation of prevailing investment directions.[2] All of these

Manuscript received 19 June 2023; revised 15 February 2024; accepted 1 March 2024. Date of publication 12 March 2024; date of current version 7 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62176005, in part by the Art Project of The National Social Science Fund of China under Grant 2022CC02195. Recommended for acceptance by G. Wang. *(Corresponding author: Shun Li.)*

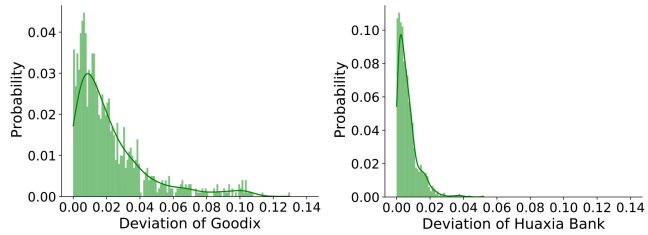
Heyuan Wang, Tengjiao Wang, Jiayi Zheng, Weijun Chen, and Wei Chen are with the School of Computer Science, National Engineering Laboratory for Big Data Analysis and Applications, Peking University, Beijing 100871, China, and also with the Institute of Computational Social Science, Peking University, Qingdao 266555, China (e-mail: wangheyuan@pku.edu.cn; tjwang@pku.edu.cn; jiayizheng@pku.edu.cn; oncecwj@stu.pku.edu.cn; pekingchenwei@pku.edu.cn).

Shun Li is with the University of International Relations, Beijing 100000, China (e-mail: lishunmail@foxmail.com).

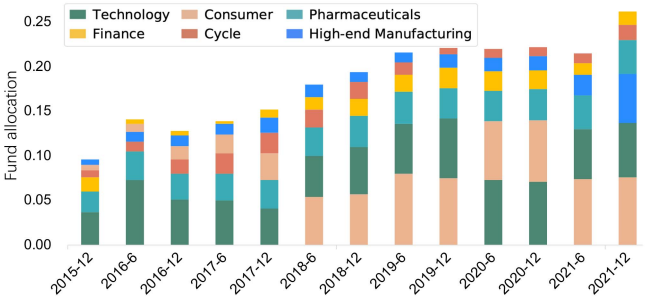
Digital Object Identifier 10.1109/TKDE.2024.3374373

[1]World Federation of Exchanges database: https://data.worldbank.org/indicator/CM.MKT.LCAP.CD/

[2]Statistics are from "Huatai Securities Report: 2022-06-10", towards 2460 active Ashare funds disclosed after the 2021 Fund Annual Report.



(a) Deviation distribution of stock volatility relative to market Index



(b) Rotation of leading fund allocation styles in the Chinese market

Fig. 1. Non-i.i.d. challenges in stock forecast.

put forward the necessity of customizing stock embeddings accounting for the diversity of different stock objects during different market scenarios to relieve the non-*i.i.d.* problems and facilitate smarter prediction.

However, it is not trivial to jointly harness the micro-object and macro-market distribution information. The severe challenges lie in 1) How to differentiate the personalized characteristics of individual stock objects while their idiosyncrasies are affected by heterogeneous factors? Fortunately, the *Momentum Spillover Effect* proves that related stocks often exhibit synchronous volatility patterns, which inspires us to analyze the interrelationships among firms as a flexible way to infer implicit knowledge of stock dynamic correlations. 2) Given the stock features, how to design a self-adaptive strategy to make stock forecasts sensitive to the dynamic of macro-market scenarios? Particularly, traditional approaches to deal with the non-*i.i.d.* problem such as rotation learning (constantly updating a new model with the latest data), exclusive learning (training an exclusive model for each scenario), and domain generalization (learning invariants in multiple source domains) [15], [16] are inapplicable for the stock prediction task, since the prior domain identifiers of market dynamics are not available and can be hardly determined even with massive expert efforts.

Building on the above analyses, in this paper we propose PTER: a **P**ersonalized **T**emporal **E**mbedding and **R**outing architecture for stock forecast. As shown in Fig. 2, PTER first employs a hyper-knowledge learner with a graph contrast module to delineate the personality traits of stock objects and encapsulate macro-market scenario indicators as a latent vector. Subsequently, the knowledge is fed into a hypernetwork-enhanced time series encoder to facilitate the customization of weights and biases in RNN and information bottleneck backbones. Notably, the hypernetwork [17] acts as a knowledge concentrator, implementing a low-dimensional manifold mapping knowledge

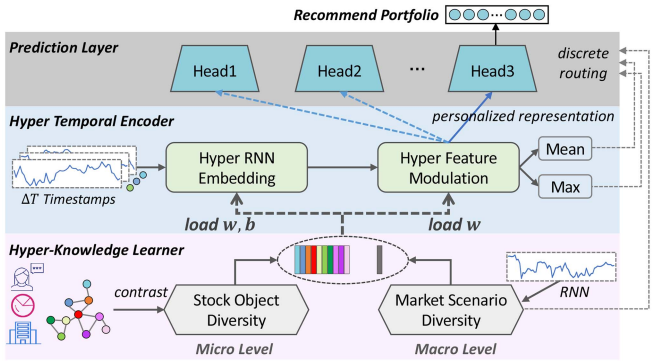


Fig. 2. Our system diagram.

space to the parameter space within a target network. Hence without incurring the overhead of additional complex designs, a relaxed weight-sharing paradigm can be opened up. It enables the extraction of diverse fluctuation patterns and the modulation of redundant channels by populating the embedding architecture with self-adaptive weights. Lastly, during the prediction stage, we follow Lin et al. [10] regarding the stock market as a mixture of finite distributions, and dispatch each sample to an orthogonal multi-head predictor according to macro-market conditions. The overall architecture can be seen as a decoupling of conflicting patterns in the input signals, addressing the issue of weak generalization caused by feature diversity and distribution shift in non-*i.i.d.* data. We combine portfolio ranking loss and unsupervised learning terms to optimize the entire framework, and conduct a series of experiments on four long-term real-world benchmark datasets over five years. The contributions of this paper are summarized as follows:

- We study the widely-explored yet intricate stock forecast problem at the intersection of data mining and finance. To our knowledge, we are the first to explicitly and jointly capture the diversity inherent in micro-level stock objects and macro-level market scenarios, so as to relieve non-*i.i.d.* issues and better characterize stock volatility trends.
- We develop a relaxed weight-sharing paradigm to accommodate the unique patterns exhibited by different stocks during different market scenarios when modeling confounded quote series. The time series encoder is not fixed but smartly parameterized, which enables achieving stronger personalization and more accurate prediction.
- By simulating quantitative trading on four of the world's largest exchange markets including NASDAQ, NYSE, TSE and Ashare&HK, we demonstrate that PTER outperforms state-of-the-art approaches by a significant margin.

II. Related Work

A. Stock Forecast

Different from general time series modeling problems, stock forecast is inherently difficult due to the high degree of non-stationarity and intricate interplay of multifactors. Given the historical features of a candidate set of stocks, the predictive

objective of a stock forecast model can involve price regression, movement classification, or ranking of expected returns.[3] Existing studies in this field can be broadly categorized into Fundamental Analysis (FA) and Technical Analysis (TA). Based on alternative financial data such as corporate news, public reviews and earnings reports, FA technology makes predictions by identifying the collective optimism or pessimism of investors in society [18], [19], [20], [21]. For example, Liu et al. [19] proposed a hierarchical attention network to distill financial news semantics for mapping to stock movements. Wang et al. [20] described an expert mining framework to screen credible investment opinion signals from online platforms. Sawhney et al. [21] investigated aligning vocal and textual clues in merger and acquisition calls. The line of TA is dedicated to probing historical price-volume trading data using mathematical algorithms such as ARIMA, SVM, Hidden Markov [22], [23] and RNN-dominated deep neural networks [5], [7], [24]. For example, Bao et al. [24] combined stacked autoencoders and LSTM to learn stock price time series. Zhang et al. [5] used Discrete Fourier Transform (DFT) to enable LSTM to discover varied frequencies of fluctuation patterns. Qin et al. [7] designed an attention-based LSTM to extract salient input and hidden states in quote data. Recent effort has also explored gated convolution [25], adversarial training [9], reinforcement learning [26], [27], Transformer-based [8], [28] and temporal-relational models [6], [29] to enhance expressiveness. Generally, all these methods follow the assumption that stock samples meet an independent identical distribution. Despite progress achieved in some cases, they inevitably suffer from non-i.i.d. issues and limited flexibility, leading to the bottleneck of generalization capability. Nevertheless, there is much less work on pertinently solving the diversity, shifts, and even conflicts of stock market characteristics. Lin et al. [10] formulated the market's temporal shift by carefully approximating an optimal transport [30] constraint, and found that dispensing samples in different periods to different predictors could bring practical gains compared to using a single predictor. However, dedicated modeling of the inherent traits of stock objects and market scenarios mingled in both micro and macro perspectives has not been effectively resolved.

B. Solution for Distribution Shift

Handling distribution shift of non-i.i.d. training and test data has raised broad attention from the community. Common solutions are causal learning and domain generalization, which aim to find invariant causal correlations between features and class variables [16], [31], [32]. There are also a variety of tactics in practice, such as exclusive learning, meta learning, robust optimization, data augmentation, style transfer and mixup [33], [34], [35], [36], [37]. Whereas, it is hard to uncover the causality in the evolution of financial time series. Most of the aforementioned techniques are unsuitable for stock forecast due to the requirement for explicit domain identifiers and lack of consideration for temporal angle. Though rotation learning can update new models with the latest data, evolving training weights is a tedious process that can only alleviate the shift in macro-market rather than micro-level diversity among stock objects.

On the other hand, inheriting the idea of parametric function approximators and fast-weights, hypernetworks (HNs) are neural networks that predict the parameters of another neural network responsible for the task of interest [17], [38]. The predicted weights are conditioned on particular genotype features and subsequently applied to the evolutionary phenotype features. This enables HNs to effectively counter distribution shift by establishing a relaxed weight-sharing paradigm. HNs have shown promising effects in various fields such as intelligent transportation [39], [40], language modeling [41], image segmentation [42], hyperparameter optimization [38], and federated learning [43], etc. For example, Pan et al. [39] extracted geographic knowledge to generate layer parameters for urban traffic prediction. Zhang et al. [40] modeled the behavior of advertisers in multiple e-commerce statuses using a meta tower to output dynamic weights. Ma et al. [42] proposed a hyper-convolution that implicitly represents kernel weights as a function of grid coordinates, making the number of learnable parameters decoupled from the kernel size. Shamsian et al. [43] trained a central hypernetwork to generate a unique model for each client in federated learning. Differently, stock forecast faces challenges of intra- and inter-series correlation, micro- and macro-level data disparity, and susceptibility to chaotic observation noise. In this work, we extend HNs to build a knowledge-driven self-adaptive model that is compatible with diverse traits of stock objects and market scenarios through personalized temporal embedding and routing. The proposed model can be flexibly applied to realistic quantitative trading platforms and provide a certain degree of interpretability.

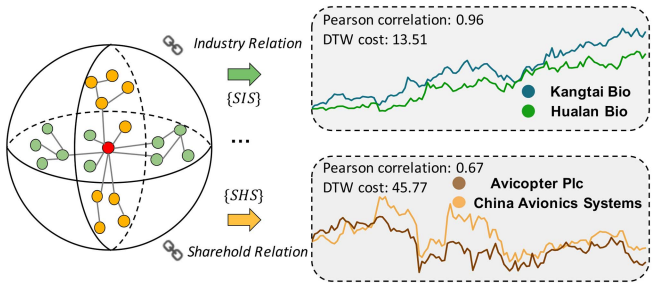


Fig. 3. Spatio-temporal correlation among stock pairs.

III. Preliminaries

Definition 1. Corporate Multiplex Graph: The momentum spillover effect reveals that related stocks tend to exhibit synchronous volatility patterns. We show the evidence of two pairs of stock price curves in Fig. 3 , where *SIS* and *SHS* refer to stocks belonging to the same industry or held by a common major shareholder, respectively. As a result of the COVID-19 outbreak, the biomedical-related stocks *Kangtai* and *Hualan* demonstrate conspicuous upward trends. Obvious

[3]Note that we focus on machine learning's ability in forecasting individual stock performance, rather than the portfolio management topic in the financial field that emphasizes risk diversification, planning of holding periods and investment objectives.

synchronization also exists between *China Avionics Systems* and *Avicopter Plc*, both of which are held by *JonHon*, China's leading aviation manufacturer. This suggests that the interdependence among listed firms provides an effective bridge for acquiring the hyper-knowledge of stock objects on behalf of their commonality and personality in the financial ecosystem. Formally, we define the corporate multiplex graph as $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_{|\mathcal{R}|}\}$, where $\mathcal{G}_r = \{\mathcal{S}, \mathcal{E}^{(r)}\}$ is a homogeneous graph for the set of candidate stocks $\mathcal{S} = \{s_1, \ldots, s_N\}$ under a specific relation view $r \in \mathcal{R}$, $\mathcal{A}^{(r)} \in \{0,1\}^{N\times N}$ is the corresponding adjacency matrix indicating the connection intensity in $\mathcal{G}_r$.

Definition 2: Stock Time Series. At timestamp t, the input time series of N stock objects forms a tensor $\mathcal{X}^t = (X_1^t, \ldots, X_N^t) \in \mathbb{R}^{N\times\Delta T\times F}$, where $X_i^t = (x_i^{t-\Delta T}, \ldots, x_i^{t-1})$ is s_i's historical quote indicators with the lookback window as ΔT and the dimension as F.

Definition 3: Market Time Series. To be aware of the macro context in stock market dynamics, we take the quote time series of global market Index $\mathcal{C}^t = (c^{t-\Delta T}, \ldots, c^{t-1})$ as input to represent scenario information at timestamp t.

Problem 1: Stock Forecast and Investment Recommendation. Following the setup of recent studies [6], [29], [44], we treat the stock forecast as a learning to rank problem, which is beneficial to practical applications like investment recommendation. Let p_i^t be the closing price of s_i at timestamp t, $y_i^t = \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$ is the associated 1-day return ratio. Given the corporate graph $\mathcal{G}$, the input time series of stock objects $\mathcal{X}^t$ and market scenario $\mathcal{C}^t$, our aim is to train a unified model $\mathcal{F}(\cdot)$. It predicts an ordering of all stocks $\mathcal{O}^t = \{o_1^t > o_2^t \ldots > o_N^t\}$ where the higher ranked ones are expected to earn more revenues at t, formulated as $\mathcal{O}^t = \mathcal{F}(\mathcal{G}, \mathcal{X}^t, \mathcal{C}^t)$. The optimal result ranks $o_i^t > o_j^t$ between any $s_i, s_j \in \mathcal{S}$, if $y_i^t > y_j^t$.

Theorem 1: Theoretical Understanding of Hypernetworks. The central to the encoder of PTER is using hypernetworks (denoted as $\mathcal{F}_H(\cdot;\Psi)$) to dynamically generate the weights of target networks (denoted as $\mathcal{F}_T(\cdot;\Theta)$) for the main task, conditioned on micro-object and macro-scenario particular identifiers (hyper-knowledge). In this paradigm, Ψ replaces Θ as the directly trainable weights. During the forward pass of training, $\mathcal{F}_H$ takes the hyper-knowledge and returns the parameters Θ for $\mathcal{F}_T$, then $\mathcal{F}_T$ takes an input $\mathcal{X}^t$ and returns an output $\mathcal{O}^t$ to calculate the loss. During the backward pass, the gradients of Θ are back-propagated through the hyernetworks to Ψ. Hence, the learning algorithm optimizes Ψ to yield Θ, thereby the target task's performance is optimized. That is, the optimization process is changed from the typical $\min_{\Theta} \mathcal{F}_T(\cdot;\Theta)$ to $\min_{\Psi} \mathcal{F}_T(\cdot;\Theta) = \mathcal{F}_T(\cdot;\mathcal{F}_H(\cdot;\Psi))$, and the hypernetwork lays on a low-dimensional manifold from the knowledge space to the space of target network parameters. The theoretical effectiveness of hypernetworks has been assessed from different perspectives. The work [45] proved that *convexity* can be achieved when the dimensionality of hypernetwork's output tends to infinity. The work [46] proved the *modularity* of hypernetwork, indicating the efficiency of mapping any conditioning signal (hyper-knowledge) to a function compared to embedding-based methods. By applying the central limit theorem for maximum-likelihood estimators [43], the hypernetwork can also be viewed as performing denoising on the extreme exclusive training where no information is shared across objects.

IV. Methodologies

As shown in Fig. 4, our PTER framework consists of three main parts: 1) Pre-training micro-level hyper-knowledge $micro\_HK_{[1:N]}$ that characterizes individual stocks' personality from structural clues of corporate multiplex graph; 2) Combining hypernetworks $\mathcal{F}_H(\cdot;\Psi)$ parametrized by Ψ with target networks $\mathcal{F}_T(\cdot;\Theta)$ parametrized by Θ as the temporal encoder. For stock s_i and timestamp t, the former network hunts specific hyper-knowledge to dynamically customize parameters for the latter, i.e., $\Theta_{i,t} = \Theta_{i,t}(\Psi) := \mathcal{F}_H([micro\_HK_i, macro\_HK_t];\Psi)$, where $macro\_HK_t$ is macro-level hyper-knowledge indicating market scenarios; 3) In the last part, each sample is routed to orthogonal predictor heads to make predictions. We depict how each module works in following subsections.

A. Stock Hyper-Knowledge Learner

Based on the observation in Definition 1, corporate relationships significantly reflect the autocorrelation of stock movements, which can be used to decompose useful hyper-knowledge to guide the learning of personalized networks. To this end, we devise a self-supervised contrastive coding module, which explicitly discerns the margins of stock synergy by pulling together subgraphs from related nodes while pushing apart unrelated ones. Note that unlike in general spatio-temporal works [3], [6], where observation node input is propagated throughout the entire graph, we learn node representations purely based on the topological structure. This approach helps reduce the adverse impacts of noise accumulation and stochastic disturbance given the low Signal Noise Ratio (SNR) inherent in stock data. Moreover, since single relationship signals may be incomplete, robust correlation clues should reconcile the variance between multi-view structures to identify invariant transferable information. Specifically, we adopt three relation views to instantiate $\mathcal{G} = \{$*Industry Graph* $\mathcal{G}_I$, *Topicality Graph* $\mathcal{G}_T$, *Shareholding Graph* $\mathcal{G}_H\}$.

1) Graph Construction: Industry Graph: The industry categorization is a systematic view for interpreting stock dynamics. Affected by common favorable factors and risk factors, stocks belonging to the same industry usually exhibit similar reactions to new messages, thus displaying a pronounced lead-lag structure [47]. In this regard, we explore the division of corporate industries from the Wind-Financial Terminal,[4] and establish edges between intra-industry stock nodes.

Topicality Graph: With the growth of web information, the relationship between corporations can also be observed from numerous online resources [6], [48]. For example, the first-order linkage in Wikidata[5] $A \xrightarrow{R_1} B$ denotes that stocks A and B has relation R_1 such as supplier-consumer, while the second-order

[4][Online]. Available: https://www.wind.com.cn/en/wft.html
[5][Online]. Available: https://www.wikidata.org/

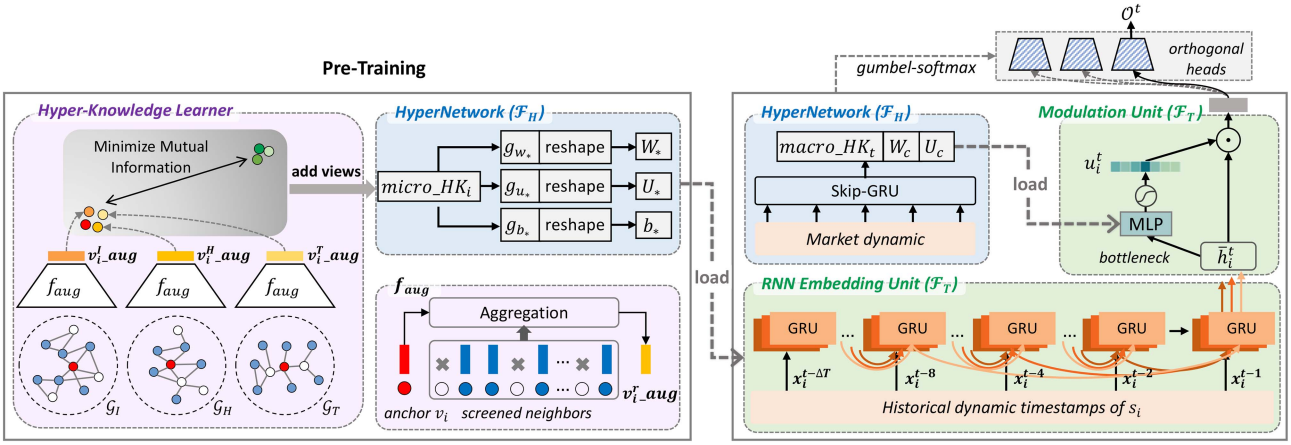


Fig. 4. Overall architecture of our proposed PTER. In the encoding stage, micro- and macro-level hyper-knowledge is fed into hypernetworks $\mathcal{F}_H$, which dynamically customize parameters for the target networks $\mathcal{F}_T$ including a hyper RNN unit and a hyper modulation unit. In the inference stage, each sample is routed to orthogonal predictor heads to forecast expected returns.

linkage $A \xrightarrow{R_1} E \xleftarrow{R_2} B$ means that A and B are connected by an intermediary entity E. In addition, the social forums enable millions of investors to discuss their portfolios and trading opinions. Stocks co-mentioned in the same review often embody user-perceived relatedness. For US and Chinese exchange markets, we collect reviews from StockTwits[6] and Xueqiu[7] to explore such connections respectively. Moreover, in order to avoid user subjectivity and ensure the credibility of extracted relations, we screen stable stock pairs abiding by the following strategies: 1) Reviews involving over 5 stock entities are discarded because they are usually sketchy records, the co-mentions are more inclined to be unintentional and meaningless; 2) Following the work [20], a financial lexicon is leveraged to identify stocks with consistent bullish/bearish semantics; 3) Finally top 15% co-occurring pairs are reserved to form edges.

Shareholding Graph: The shareholding connection also plays an important role in characterizing corporate relatedness. We gather 10 major shareholders per stock and amalgamate those from the same organization based on syntactic matching such as $F(A)$, $F - A$ and $F \cdot A$, where F is the generic term indicating one shareholder while A denotes the accessory annotation. Thereby we create an edge between two nodes if they have the same shareholders.

2) Subgraph Instances: The real-world inter-stock linkages can occupy strong skewness where a few corporations have a large degree of connections, for example, popular stocks *Kweichow Moutai* and *Apple Inc* are most broadly discussed at Chinese investment forums. Intuitively, the hub nodes can exert disproportionate influence and exacerbate over-smoothing when embedding the protogenic topology. Inspired by the word context factorization in NLP [49], we first apply PPMI to reduce the tilt of $\mathcal{G}$ by performing random surfing with restart on each graph view $\mathcal{G}_r$. Let λ be the restart weight, row vector $f_k^{r,i}$ denotes a reachable path rooted at node s_i, whose jth entry means the probability of reaching s_j after k-step recursions:

$$f_k^{r,i} = \lambda \cdot f_{k-1}^{r,i} \mathcal{A}^{(r)} + (1-\lambda) f_0^{r,i}. \tag{1}$$

In specific, $f_0^{r,i}$ is a one-hot vector with the ith entry being 1. Then, the r-type adjacency matrix is transformed into $\bar{A}^r$, where $\bar{A}^r_{i,*} = \sum_{k=1}^{K} f_k^{r,i}$ is the ith row indicating the frequency s_i occurring in surrounding neighborhood within a window size of K. Thereby the PPMI values between nodes are measured as

$$PPMI^r(i,j) = \max\left(\log\left(\frac{\bar{A}^r_{i,j} \cdot \sum_{i',j'} \bar{A}^r_{i',j'}}{\sum_j \bar{A}^r_{i,j} \cdot \sum_i \bar{A}^r_{i,j}}\right), 0\right), \tag{2}$$

where a larger value means more robust relatedness between a node and its neighborhood. The role of hubs is inhibited and more latent dependencies can be highlighted. Next, we embed the subgraphs of each node to form contrastive instances. Given a threshold ρ_r, we apply semantic attention to integrate s_i's high-correlated neighbors $\mathcal{N}_i^r = \{s_j | s_j \in \mathcal{S} \ and \ PPMI^r(i,j) \geq \rho_r\}$ in $\mathcal{G}_r$

$$\boldsymbol{v}_i^r\_aug = \sigma\left(\sum\nolimits_{s_j \in \mathcal{N}_i^r} \alpha_{i,j}^r \cdot \boldsymbol{v}_j\right), \tag{3}$$

$$\alpha_{i,j}^r = \frac{\exp\left(LeakyReLU\left(\boldsymbol{a}_r^\top \cdot [\boldsymbol{v}_i || \boldsymbol{v}_j]\right)\right)}{\sum_{s_k \in \mathcal{N}_i^r} \exp\left(LeakyReLU\left(\boldsymbol{a}_r^\top \cdot [\boldsymbol{v}_i || \boldsymbol{v}_k]\right)\right)}, \tag{4}$$

where $\boldsymbol{a}_r$ is an attention vector for the rth view, $||$ is concatenation operator. The graph nodes are randomly initialized as learnable vectors. Following Wang et al. [50], practically we do not aggregate all the information of $\mathcal{N}_i^r$, but randomly sample a part of unrepeated neighbors every epoch. By this means the multiplicity of subgraph instances can be promoted and the contrastive learning task is more challenging.

3) View-Collaborative Contrastive Learning: After getting $\boldsymbol{v}_i^r\_aug$ from each view of $\mathcal{G}$, we map them into a shared space where contrast operations are performed to enhance the collaboration between views

$$\boldsymbol{v}_i^r\_inter = \boldsymbol{W}^{(2)} ReLU(\boldsymbol{W}^{(1)} \boldsymbol{v}_i^r\_aug + \boldsymbol{b}^{(1)}) + \boldsymbol{b}^{(2)}. \tag{5}$$

[6]The well-known Twitter-like investment forum https://www.stocktwits.com
[7]A popular Chinese investment forum https://www.xueqiu.com

Then, for each stock s_i, we sort the set of nodes in descending order based on the sum of their relevance to s_i across all views, i.e, $\mathbb{D}_i(j) = \sum_{r\in\mathcal{R}_{[I,T,H]}} PPMI^r(i,j)$. Accordingly, the subgraph embeddings of the former $N_{pos'} = \min(N_{pos}, |\mathbb{D}_i(\cdot) > 0|)$ nodes are selected as positive instances $\mathbb{P}_i$, where N_{pos} is a maximum threshold. The remaining nodes are naturally treated as negative instances $\mathbb{N}_i$. On this basis, we perform a full grasp of inter-stock relevance information by optimizing the whole contrastive loss between $\binom{|\mathcal{R}|}{2}$ pairs of structural views

$$\mathcal{L}_{hyper} = \sum_{1\le q<r\le|\mathcal{R}|}\sum_{i=1}^{|\mathcal{S}|} - \log\frac{\sum_{v_{i_+}\in\mathbb{P}_i}\exp\left(\boldsymbol{v}_{i\_}^{q}inter^{\top}\boldsymbol{v}_{i_+\_}^{r}inter/\tau_1\right)}{\sum_{v_j\in\{\mathbb{P}_i\bigcup\mathbb{N}_i\}}\exp\left(\boldsymbol{v}_{i\_}^{q}inter^{\top}\boldsymbol{v}_{j\_}^{r}inter/\tau_1\right)}, \tag{6}$$

where τ_1 is a temperature coefficient. Unlike the general contrastive objective that focuses on one positive pair in the numerator, we enrich the learner's ability by transferring knowledge across different views. At the end of self-supervised graph coding, we add $\{\boldsymbol{v}_i^r\_aug|r\in\mathcal{R}\}$ as s_i's hyper-knowledge $micro\_HK_i$ for downstream predictions.

B. Hyper Temporal Encoder

We next embed stocks' time series for predicting future movements. As individual stocks have personalized volatility patterns and the pattern significance may vary under shifting market scenarios, assigning undifferentiated parameters to a simply shared network such as canonical RNN is insufficient to model such diversity. With hyper-knowledge derived above, we propose a hyper temporal encoder that consists of two components: a hyper RNN unit and a hyper modulation unit to customize stock representations.

1) Hyper RNN Unit: The regularities of stock fluctuation are implicit in manifold short- and long-term time horizons. Traditional time series models primarily focus on capturing consecutive signals while not adept at understanding patterns with varying periodicities. In this regard, we engage skip-GRU [51] as the backbone for temporal embedding and catch multi-periodic price features by applying increasing skip intervals. Given input time series $X_i^t = (x_i^{t-\Delta T}, \ldots, x_i^{t-1})$ of s_i, the hyper RNN strategy is defined as

$$\boldsymbol{h}_i^{\tau} = HyperGRU_{skip}(\boldsymbol{x}_i^{\tau}, \boldsymbol{h}_i^{\tau-\delta}|\boldsymbol{W}_*, \boldsymbol{U}_*, \boldsymbol{b}_*), \tag{7}$$

where δ is the length of skipping timestamps, e.g., for daily-frequency trading, $\delta = 1$ mines the recent stock fluctuation trend, $\delta = 5$ captures weekly patterns. $\boldsymbol{h}_i^{\tau}$ is the hidden state at τ's step computed by

$$\begin{aligned}\boldsymbol{z} &= \sigma(\boldsymbol{W}_z\boldsymbol{x}_i^{\tau} + \boldsymbol{U}_z\boldsymbol{h}_i^{\tau-\delta} + \boldsymbol{b}_z)\\ \boldsymbol{r} &= \sigma(\boldsymbol{W}_r\boldsymbol{x}_i^{\tau} + \boldsymbol{U}_r\boldsymbol{h}_i^{\tau-\delta} + \boldsymbol{b}_r)\\ \boldsymbol{h}_i^{\tau} &= \boldsymbol{h}_i^{\tau-\delta}\circ(1-\boldsymbol{z}) + \phi(\boldsymbol{W}_h\boldsymbol{x}_i^{\tau} + \boldsymbol{U}_h(\boldsymbol{r}\circ\boldsymbol{h}_i^{\tau-\delta}) + \boldsymbol{b}_h)\circ\boldsymbol{z},\end{aligned} \tag{8}$$

where $\circ$ is element-wise multiplication, $\sigma(\cdot)$ and $\phi(\cdot)$ are sigmoid and tanh functions. Specifically, the hypernetwork $\mathcal{F}_H$ is responsible for generating personalized weights and biases based on the hyper-knowledge of stock object

$$\begin{aligned}\boldsymbol{W}_* &= \Omega(g_{w_*}(micro\_HK_i))\ \boldsymbol{U}_* = \Omega(g_{u_*}(micro\_HK_i))\\ \boldsymbol{b}_* &= \Omega(g_{b_*}(micro\_HK_i)),\end{aligned} \tag{9}$$

where Ω is the reshape operator to match the sizes of corresponding parameters. $\{g_{w_*}, g_{u_*}, g_{b_*}|* \in \{z, r, h\}\}$ are implemented with a fully-connected network, which are shared across all objects. Unlike existing stock forecast studies employing concatenation-based data fusion or relying on spatio-temporal signal propagation, here the temporal embedding process can be seen as self-tuning which coordinates multi-object's guiding knowledge. Thus, the universality and personality among stock objects are simultaneously accommodated while mitigating noise disturbances.

2) Hyper Modulation Unit: Regarding the low SNR of stock data that different parts of extracted features are not equally significant for prediction, we further utilize SEblock [52] as the backbone to protrude important feature channels and compress redundancy. Specifically, we take the final hidden state produced by applying skip-GRU on the input time series of market Index $\mathcal{C}^t$ to represent the macro-market hyper-knowledge $macro\_HK_t$ at timestamp t. Then the weights of SEblock are dynamically generated by a hypernetwork

$$\boldsymbol{W}_c = \Omega(g_{w_c}(macro\_HK_t))\ \boldsymbol{U}_c = \Omega(g_{u_c}(macro\_HK_t)), \tag{10}$$

where Ω is the reshape operator, $\boldsymbol{W}_c \in \mathbb{R}^{\frac{\bar{d}}{p}\times\bar{d}}$ and $\boldsymbol{U}_c \in \mathbb{R}^{\bar{d}\times\frac{\bar{d}}{p}}$ are dimensionality reduction-recovery parameters with an abstraction ratio of p, g_{w_c} and g_{u_c} are fully-connected layers. By concatenating the multi-periodic hidden states of stock object's time series $\boldsymbol{h}_i^t$ into a compact vector $\bar{\boldsymbol{h}}_i^t \in \mathbb{R}^{\bar{d}}$, the customized SEblock performs information bottleneck by

$$\boldsymbol{u}_i^t = \sigma(\boldsymbol{U}_c ReLU(\boldsymbol{W}_c\bar{\boldsymbol{h}}_i^t)), \tag{11}$$

where σ is the sigmoid function. Finally, the stock representation is produced by rescaling extracted features with the activation $\boldsymbol{u}_i^t$ through element-wise multiplication

$$\tilde{\boldsymbol{h}}_i^t = \boldsymbol{u}_i^t \circ \bar{\boldsymbol{h}}_i^t. \tag{12}$$

Since we give heed to macro-market scenarios and incorporate such information to help determine the significant dynamic features of individual stock objects, this unit not only aids the efficacy of stock representation but also measures the contemporaneous stock-to-market correlation.

C. Prediction Router

After characterizing the input time series, we next introduce the prediction component to output the ranking of expected returns for the candidate pool. Recent work [10] investigated that dispensing different samples to different predictors could bring practical gains compared to using a single fixed predictor. We borrow this idea and apply a multi-head operator to simulate adaptive routing in inference. Let $\hat{\mathcal{P}} = \{\mathcal{P}_1, \ldots, \mathcal{P}_M\}$ be a M-head predictor where each head consists of an MLP layer with Leaky-ReLU activation. Given stock representations $\boldsymbol{H}_t = [\tilde{\boldsymbol{h}}_i^t]_{i=1}^N \in \mathbb{R}^{N\times\bar{d}}$ at timestamp t, the estimated return

ratios are $Y^t = \hat{\mathcal{P}}(\boldsymbol{H}_t) = [\mathcal{P}_1(\boldsymbol{H}_t), \ldots, \mathcal{P}_M(\boldsymbol{H}_t)]^\top$. We apply orthogonal regularization to $\hat{\mathcal{P}}$ to prevent learning redundant heads in the prediction space that may lead to trivial solutions. By concatenating the weights of each head into a matrix $\boldsymbol{W}_{\mathcal{P}} \in \mathbb{R}^{M \times \bar{d}}$ and normalizing it as $\tilde{\boldsymbol{W}}_{\mathcal{P}} = \boldsymbol{W}_{\mathcal{P}} / \|\boldsymbol{W}_{\mathcal{P}}\|_2$, the regularization term is computed by

$$\mathcal{L}_{OR} = \|\tilde{\boldsymbol{W}}_{\mathcal{P}} \tilde{\boldsymbol{W}}_{\mathcal{P}}^\top - I\|_F \,, \tag{13}$$

where $\|\cdot\|_F$ is Frobenius norm for traversing and measuring the disagreements on matrix rows, $I \in \mathbb{R}^{M \times M}$ is an identity matrix. The routing process to output ranking scores is

$$l^t = \boldsymbol{V}_p [meanpool(\boldsymbol{H}_t) || maxpool(\boldsymbol{H}_t) || macro\_HK_t] + \boldsymbol{b}_p \,, \tag{14}$$

$$\hat{y}^t = \gamma^\top Y^t, \; \gamma_m = \frac{\exp((l_m^t + \epsilon_m)/\tau_2)}{\sum \exp((l^t + \epsilon)/\tau_2)} \,, \tag{15}$$

where $\boldsymbol{V}_p \in \mathbb{R}^{M \times 3\bar{d}}$ is an attention vector, $\epsilon = [\epsilon_1, \ldots, \epsilon_M]$ is noise sampled from Gumbel(0,1) distribution, τ_2 is a temperature controlling the output sharpness. The router combines the market scenario as well as the overall and salient representations of constituent stocks to calculate a discrete selection among Y^t. Note that Gumbel-softmax [53] is used to exert reparameterization on the attention logits to ensure differentiability.

D. Optimization

Rank Loss: To optimize PTER, we jointly compute point-wise and pair-wise ranking losses with a weighting coefficient, which minimizes the discrepancy between predicted and ground-truth returns while preserving the relative order of the top-ranked stocks

$$\mathcal{L}_{Rank} = \sum_{i=1}^{N} \left\| \hat{y}_i^t - y_i^t \right\|^2 + \alpha \sum_{i=1}^{N} \sum_{j=1}^{N} \max\left(0, -\left(\hat{y}_i^t - \hat{y}_j^t\right)\left(y_i^t - y_j^t\right)\right) \,. \tag{16}$$

Freezing versus Full-Tuning: We investigate two strategies to train the whole framework: 1) In the freezing mode, we pre-train the self-supervised stock hyper-knowledge learner by minimizing $\mathcal{L}_{hyper}$ and freeze its parameters as a static feature extractor of $micro\_HK_{[1:N]}$. On this foundation, the hyper temporal encoder and prediction router cater for the downstream investment recommendation task

$$\min \mathcal{L} = \mathcal{L}_{Rank} + \eta \mathcal{L}_{OR} \,, \tag{17}$$

where η is a weighting factor. 2) In the full-tuning mode, the pre-trained hyper-knowledge learner is fine-tuned together with the downstream encoder and predictor. Both strategies are evaluated and compared in the experiments.

V. Experiments

In this section, we probe the effectiveness of PTER and seek to answer the following research questions:

1) *RQ1:* Can PTER outperform other state-of-the-art studies that jointly learn spatio-temporal information of stocks?
2) *RQ2:* What are the contributions of different components?
3) *RQ3:* How do hyperparameter settings, such as look-back window, dimensions of hyper-knowledge and hidden units, the number of predictor's heads affect the performance?

TABLE I
Detailed Statistics of the Datasets

	NASDAQ	NYSE	TSE	Ashare&HK
#Stocks (Nodes)	1,026	1,737	95	85
Train Timespan	01/13-12/15	01/13-12/15	11/15-08/18	12/14-5/18
Val Timespan	01/16-12/16	01/16-12/16	08/18-07/19	05/18-6/19
Test Timespan	01/17-12/17	01/17-12/17	07/19-08/20	06/19-7/20
#Days Tra:Val:Test	756:252:237	756:252:237	693:231:235	756:252:285
$\mathcal{G}_I$ **Avg./Max Degree**	52.25 / 156	163.58 / 500	5.46 / 11	5.77 / 14
$\mathcal{G}_T$ **Avg./Max Degree**	2.89 / 55	5.88 / 114	7.73 / 29	14.09 / 53
$\mathcal{G}_H$ **Avg./Max Degree**	43.98 / 185	106.81 / 357	7.81 / 20	9.66 / 24

A. Experimental Setup

1) Datasets: Different from many prior stock forecast arts that evaluate only dozens of stocks, we conduct extensive evaluations on four large real-world datasets from US, Japanese and Chinese exchanges: *1) NASDAQ* [48] collects price-volume records of 1026 equity stocks from the fairly volatile US S&P 500 and NASDAQ Indices; *2) NYSE* [48] targets 1,737 stocks from NYSE, the world's largest stock exchange by far w.r.t. the capitalization of listed corporations. It covers Dow Jones Industrial Average, S&P 500 and NYSE Composite Indices and is relatively stable compared to NASDAQ; *3) TSE* [54] is from the popular TOPIX-100 Index and includes 95 constituent stocks with the largest capitalization in Tokyo Exchange; *4) Ashare&HK* [55] consists of 85 top-traded Chinese stocks from Shanghai, Shenzhen and Hong Kong markets. Detailed statistics are summarized in Table I. We crawl trading quote data including split-adjusted *opening-high-low-closing* prices (OHLC) and *volumes*, from professional Wind-Financial Terminal for all individual stocks and composite indices. All indicators are preprocessed by z-score normalization. Training/validation/test sets are strictly partitioned in chronological order with non-overlapping timespans to avoid the data leakage problem.

2) Comparative Methods: We compare PTER against four groups of stock prediction baselines: *1) Classification* methods discretize stock movements into [up, neutral, down] classes and pick stocks that are expected to rise. *ARIMA* [56] is a typical autoregressive model devoted to linear statistical analysis of stock time series; *Adv-ALSTM* [9] applies LSTM and adversarial training to simulate the stochasticity of stock dynamics; *HATS* [57] feeds price vectors as node input and then embeds stock graphs via a hierarchical attention mechanism; *HMG-TF* [8] designs Gaussian Transformer to extract daily and weekly fluctuation patterns of stock time series; *LSTM-RGCN* [54] feeds news events as node input and then performs RGCN to model stock relationships; *HATR* [25] is a sequential spatio-temporal model combining gated causal convolution and GCN modules. *2) Regression* methods are dedicated to regressing absolute stock prices. *SFM* [5] adapts the RNN memory with discrete Fourier transform to discover multi-frequency trading signals;

TABLE II
PROFITABILITY COMPARISON WITH CLASSIFICATION (CLF), REGRESSION (REG), REINFORCEMENT LEARNING (RL), AND RANKING (RAN) BASELINES

	Methods	NASDAQ		NYSE		TSE		Ashare&HK	
		SR ↑	*IRR* ↑	*SR* ↑	*IRR* ↑	*SR* ↑	*IRR* ↑	*SR* ↑	*IRR* ↑
CLF	ARIMA	$0.55\pm1e^{-3}$	$0.10\pm6e^{-3}$	$0.33\pm3e^{-3}$	$0.10\pm5e^{-3}$	$0.47\pm2e^{-3}$	$0.13\pm1e^{-3}$	$0.37\pm4e^{-3}$	$0.43\pm3e^{-3}$
	Adv-ALSTM	$0.97\pm5e^{-3}$	$0.23\pm3e^{-3}$	$0.81\pm4e^{-3}$	$0.14\pm7e^{-3}$	$1.10\pm1e^{-3}$	$0.43\pm9e^{-2}$	$0.83\pm4e^{-3}$	$0.80\pm6e^{-3}$
	HATS	$0.80\pm6e^{-3}$	$0.15\pm7e^{-3}$	$0.73\pm5e^{-3}$	$0.12\pm2e^{-3}$	$0.96\pm4e^{-3}$	$0.31\pm2e^{-3}$	$0.77\pm1e^{-3}$	$0.72\pm5e^{-3}$
	HMG-TF	$0.83\pm2e^{-3}$	$0.19\pm4e^{-3}$	$0.75\pm2e^{-3}$	$0.13\pm1e^{-3}$	$1.05\pm3e^{-3}$	$0.33\pm2e^{-3}$	$0.88\pm6e^{-3}$	$0.83\pm7e^{-3}$
	LSTM-RGCN	$0.75\pm4e^{-3}$	$0.13\pm1e^{-3}$	$0.70\pm3e^{-3}$	$0.10\pm6e^{-3}$	$0.90\pm7e^{-3}$	$0.28\pm5e^{-3}$	$0.73\pm6e^{-3}$	$0.75\pm3e^{-3}$
	HATR	$0.92\pm6e^{-3}$	$0.31\pm3e^{-3}$	$0.76\pm3e^{-3}$	$0.14\pm4e^{-3}$	$0.98\pm5e^{-3}$	$0.36\pm4e^{-3}$	$1.10\pm5e^{-3}$	$0.90\pm3e^{-3}$
REG	SFM	$0.16\pm6e^{-3}$	$0.09\pm5e^{-3}$	$0.19\pm1e^{-3}$	$0.11\pm2e^{-3}$	$0.08\pm2e^{-3}$	$0.07\pm4e^{-3}$	$0.21\pm1e^{-3}$	$0.31\pm6e^{-3}$
	DA-RNN	$0.71\pm3e^{-3}$	$0.14\pm3e^{-3}$	$0.66\pm2e^{-3}$	$0.13\pm1e^{-3}$	$0.86\pm2e^{-3}$	$0.25\pm4e^{-3}$	$0.72\pm5e^{-3}$	$0.73\pm3e^{-3}$
RL	DQN	$0.93\pm5e^{-3}$	$0.20\pm6e^{-3}$	$0.72\pm5e^{-3}$	$0.12\pm4e^{-3}$	$1.08\pm5e^{-3}$	$0.31\pm7e^{-3}$	$0.69\pm2e^{-3}$	$0.71\pm4e^{-3}$
	iRDPG	$1.32\pm5e^{-3}$	$0.28\pm4e^{-3}$	$0.85\pm7e^{-3}$	$0.18\pm3e^{-3}$	$1.10\pm2e^{-3}$	$0.55\pm1e^{-3}$	$1.16\pm6e^{-3}$	$0.87\pm5e^{-3}$
	RAT	$1.37\pm4e^{-3}$	$0.40\pm5e^{-3}$	$1.03\pm6e^{-3}$	$0.22\pm2e^{-3}$	$1.17\pm3e^{-3}$	$0.64\pm4e^{-3}$	$1.17\pm3e^{-3}$	$0.92\pm5e^{-3}$
RAN	SAE-LSTM	$0.95\pm4e^{-3}$	$0.22\pm2e^{-3}$	$0.79\pm1e^{-3}$	$0.12\pm6e^{-3}$	$0.73\pm1e^{-3}$	$0.21\pm5e^{-3}$	$0.74\pm7e^{-3}$	$0.74\pm3e^{-3}$
	RSR-E	$1.12\pm5e^{-3}$	$0.26\pm4e^{-3}$	$0.88\pm6e^{-3}$	$0.20\pm3e^{-3}$	$1.07\pm1e^{-3}$	$0.50\pm7e^{-3}$	$0.82\pm5e^{-3}$	$0.81\pm2e^{-3}$
	RSR-I	$1.34\pm6e^{-3}$	$0.39\pm5e^{-3}$	$0.95\pm1e^{-3}$	$0.21\pm3e^{-3}$	$1.08\pm6e^{-3}$	$0.53\pm4e^{-3}$	$0.85\pm1e^{-3}$	$0.86\pm2e^{-3}$
	ALSTM-TRA	$1.22\pm5e^{-3}$	$0.30\pm6e^{-3}$	$0.89\pm4e^{-3}$	$0.21\pm5e^{-3}$	$1.14\pm3e^{-2}$	$0.59\pm6e^{-2}$	$0.86\pm8e^{-3}$	$0.90\pm5e^{-3}$
	HyperStockGAT	$1.40\pm7e^{-3}$	$0.44\pm1e^{-2}$	$1.10\pm8e^{-3}$	$0.25\pm9e^{-3}$	$1.20\pm8e^{-3}$	*$0.75\pm6e^{-3}$*	$1.25\pm9e^{-3}$	$1.05\pm2e^{-3}$
	STHAN-SR	$1.42\pm1e^{-2}$	$0.44\pm4e^{-3}$	$1.12\pm8e^{-3}$	$0.33\pm7e^{-3}$	$1.19\pm2e^{-2}$	$0.62\pm5e^{-3}$	$1.28\pm6e^{-3}$	*$1.09\pm3e^{-3}$*
	ALSP-TF	*$1.55\pm2e^{-2}$*	*$0.53\pm7e^{-3}$*	*$1.24\pm9e^{-3}$*	*$0.41\pm8e^{-3}$*	*$1.27\pm2e^{-2}$*	$0.71\pm7e^{-3}$	*$1.31\pm1e^{-2}$*	$1.08\pm8e^{-3}$
	PTER (freeze)	**1.57**$^{\star}\pm6e^{-3}$	**0.54**$^{\star}\pm5e^{-3}$	**1.26**$^{\star}\pm7e^{-3}$	**0.43**$^{\star}\pm8e^{-3}$	**1.33**$^{\star}\pm1e^{-2}$	**0.80**$^{\star}\pm6e^{-3}$	**1.42**$^{\star}\pm5e^{-3}$	**1.18**$^{\star}\pm4e^{-3}$
	PTER (full)	**1.58**$^{\star}\pm1e^{-2}$	**0.56**$^{\star}\pm7e^{-3}$	**1.27**$^{\star}\pm5e^{-3}$	**0.46**$^{\star}\pm9e^{-3}$	**1.31**$^{\star}\pm9e^{-3}$	**0.79**$^{\star}\pm8e^{-3}$	**1.42**$^{\star}\pm1e^{-2}$	**1.17**$^{\star}\pm8e^{-3}$
	%Improv. SOTA	+2.39%	+6.42%	+2.83%	+13.17%	+5.21%	+6.79%	+8.02%	+8.26%

Blue and violet depict the best performance and baselines. *Means the improvement over SOTAs is statistically significant (t-test $p < 0.01$, wilcoxon's signed-rank test $p < 0.05$). The brackets denote training PTER with freezing / full-tuning strategies.

DA-RNN [7] equips RNN with dual-stage attention to highlight driving input and hidden states. *3) Reinforcement Learning* methods optimize trading actions in a RL framework taking gotten revenues as rewards. *DQN* [27] is an ensemble of deep Q-learning agents for stock forecast; *iRDPG* [26] automatically develops quantitative strategy by RL and imitation learning; *RAT* [28] uses relation-aware Transformer and RL for portfolio selection. *4) Ranking* methods select stocks with higher expected returns as a learning-to-rank task. *SAE-LSTM* [24] combines stacked autoencoders and LSTM to embed stock time series; *RSR-E* [48] and *RSR-I* [48] adopt GCN by weighting stock correlation with the similarity of feature vectors and neural transformation; *ALSTM-TRA* [10] uses attentive LSTM and optimal transport to learn varying patterns; *HyperStockGAT* [44] blends hyperbolic graph convolution and temporal convolution to capture stock dynamics; *STHAN-SR* [6] sequentially applies attentive LSTM and hypergraph attention for stock embedding; *ALSP-TF* [29] enhances Transformer to capture interactions within stock price series under a DTW-graph reconstruction constraint. Table III compares the traits of all baseline architectures.

3) Implementation Details: 1) Hyperparameters. For fair comparisons, we follow previous work [6], [29] and partition samples by moving a 16-day lookback window along trading time series. In the stock hyper-knowledge learner, we set the size of neighborhood $K = 5$ and restart weight $\lambda = 0.8$ for surfing on corporate graphs. To construct contrastive instances, we select positive samples setting $N_{pos} = 6$ for Ashare&HK and 10 for other datasets under each relation view. The dimension of knowledge embeddings is 32. In the temporal encoder, the hidden unit of hyper RNN has a dimension of 32 and skipping intervals are $\delta \in \{1, 3, 5\}$, the reduction ratio p for hyper feature modulation is set to 8. For the final prediction, we test applying 1→10 numbers of predictor heads and utilize $M = 4$ for NASDAQ and NYSE, $M = 5$ for the others. Dropout [58] with a rate of 0.3 is applied at the end of each layer. Loss factors are set to $\alpha = 0.2$ and $\eta = 0.01$. We use Adam optimizer [59]

TABLE III
COMPARISONS OF BASELINE ARCHITECTURES

Methods	Temporal Info.		Relation Info.		Distribution Shift.	
	SP	MP	SR	MR	Micro	Macro
ARIMA	√	×	×	×	×	×
Adv-ALSTM	√	×	×	×	×	√
HATS	√	×	√	√	×	×
HMG-TF	√	√	×	×	×	×
LSTM-RGCN	×	×	√	√	×	×
HATR	√	√	√	√	×	×
SFM	×	√	×	×	×	×
DA-RNN	√	×	×	×	×	×
DQN	√	√	×	×	×	×
iRDPG	√	×	×	×	×	×
RAT	√	√	√	×	×	×
SAE-LSTM	√	×	×	×	×	×
RSR-E	√	×	√	√	×	×
RSR-I	√	×	√	√	×	×
ALSTM-TRA	√	×	×	×	×	√
HyperStockGAT	√	×	√	√	×	×
STHAN-SR	√	×	√	√	×	×
ALSP-TF	√	√	√	×	×	×
PTER	√	√	√	√	√	√

To embed quote temporal information, "SP" represents a single periodicity, and "MP" denotes multiple periodicities. To embed relation information, "SR" denotes single-view relationship, "MR" denotes multiplex relationships between listed firms. To deal with the problem of distribution shift, "Micro" denotes the diverse characteristics of stock objects at micro-level, and "Macro" denotes the shift of global market scenarios at macro-level.

to tune model parameters for 200 epochs with early stopping patience as 40, the initial learning rate is 1e-3. We conduct a grid search and find optimal hyperparameters according to the performance of the validation set.

2) Platform Configuration and Training Time: Our model was implemented by PyTorch version 1.5.0 with CUDA version 11.4. Experiments were conducted on Linux servers with an Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10 GHz, 256 GB RAM and a single Nvidia RTX 2080Ti 11 GB GPU. It takes 8.21, 12.35, 1.33, 1.23 secs/epoch to train PTER on NASDAQ, NYSE, TSE and China&HK datasets, respectively. Compared to state-of-the-art models, the training time is on par with STHAN-SR and ALSP-TF, significantly better than HyperStockGAT which requires 1.9, 3.4, 1.4 and 1.3 minutes per epoch for training on the four datasets.

3) Evaluation Metrics: We use the classic Normalized Discounted Cumulative Gain (NDCG@κ) as the metric of PTER's ranking ability, and adopt profitability metrics including the cumulative investment return ratio (IRR) and annualized Sharpe ratio (SR) to assess profit generation ability of all methods. Following previous studies [6], [29], [44], [48], we employ a daily buy-hold-sell trading strategy. That is, when the market closes on trading day t, the trader buys top-κ stocks with the highest expected returns in the model's predicted ranking list, and then sells off those purchased shares on the close market of the next trading day. This strategy ensures that profitability comparisons result from the capabilities of stock prediction models, rather than by changing the rules of portfolio management. Each experiment is run ten times and the mean results are reported. Specifically, NDCG is commonly used to measure the prediction efficacy of a ranking system. It gives more credit to items with higher positions in a ranking list to obtain discounted cumulative gain (DCG), and then measures the quotient of DCG in the actual order and the ideal order. We thus calculate NDCG over all stock objects on every trading day in the test duration. IRR is the cumulative return on an investment over time. We calculate IRR by summing the return ratios of selected stocks on each test day, formulated by $\text{IRR}^t = \sum_{i \in \hat{\mathcal{S}}^t} \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$, where $\hat{\mathcal{S}}^t$ denotes the portfolio on day t and p_i^t is the close price of s_i. Differently, SR measures the risk-adjusted returns. It characterizes how well an investment earning R_p compensates a trader for the borne risk, i.e., SR= $\frac{E[R_p]-R_f}{std[R_p]}$, where R_f is the risk-free rate.[8],[9] These metrics enable a unified comparison between ranking models and other non-ranking stock prediction methods. The larger values of NDCG, IRR and SR indicate better performance. For both returns and ranking comparison, we report results with $\kappa = 5$ by default. The performance with varying top κ is analyzed in Section V-D.

B. Profitability Comparison (RQ1)

Table II lists the results of different methods, from which we obtain several key observations. First, nonlinear models usually surpass the linear algorithm based on statistical analysis (*ARIMA*). RL and ranking methods generally perform superior to conventional studies utilizing classification or regression objectives. This illustrates the prospect of deep neural networks in tackling intricate Fintech issues, and the methods directly optimizing stock selection may be more conducive to pursuing investment returns in practice.

Second, considering the interrelationship among stocks (e.g., *LSTM-RGCN*, *HATR*, *HyperStockGAT* and *STHAN-SR*) often yields better results than regarding individual stock's time series in isolation, which demonstrates the effect of the collective synergy in stock dynamics. Especially, the Transformer-based *ALSP-TF* and the *STHAN-SR* which utilizes multi-graph information for spatio-temporal modeling are strong competitors, yet our model still outperforms them on each metric. We attribute the gains to three reasons. 1) Previous studies are generally built on the i.i.d. assumption while neglecting the diversity of individual stock objects under different market scenarios. Hence, the expressiveness of target network is tightly coupled with the limited number of shared parameters. In contrast, *PTER* has the ability to perceive personalized prior knowledge of arbitrary dimensions while maintaining the size of target network. 2) *STHAN-SR* follows a sequential framework, where the output of its temporal encoder is fed into a graph convolutional module as input. However, the diffusion of observed features in highly volatile stock prices may exacerbate noise accumulation and make the model susceptible to local disturbances. Instead, *PTER* embeds the relationships among listed firms into stable hyper-knowledge of stocks in an unsupervised manner. Aided by hypernetworks, the knowledge can be flexibly integrated with the target network to achieve personalized time series embedding. 3) By conducting feature modulation and dynamic routing prediction, *PTER* becomes more sensitive to the macro-market scenarios, thereby enhancing its adaptability to temporal distribution shifts.

Third, *PTER* consistently achieves the optimal results on all datasets. On average, it fetches 4.6% and 8.6% improvements in terms of risk-adjusted and cumulative profits over the best baselines. In freezing mode, the stock hyper-knowledge learner is positioned outside the training objective of portfolio ranking. Nevertheless, the performance of *PTER (freeze)* closely approximates that of the full-tuning counterpart *PTER (full)*, and surpasses existing spatio-temporal stock prediction studies. This indicates that the pre-trained graph embedding is credible as robust hyper-knowledge for stocks, and can serve as the start of downstream investment recommendation networks.

C. In-Depth Analysis (RQ2)

1) Ablation Study: To look closer at how different components affect the performance of *PTER*, we study four ablation variants:

- *w/o hyper RNN unit*: Apply vanilla skip-GRU as the simple backbone to embed stock time series input.
- *w/o hyper modulation unit*: Apply vanilla SEblock as the bottleneck for denoising extracted volatility features.
- *w/o hyper temporal encoder*: Discard the knowledge-aware hypernetworks and train a fixed set of parameters for the above target networks following i.i.d. assumption.

[8]T-Bill Rate: https://home.treasury.gov/

[9]Chinabond: https://yield.chinabond.com.cn/cbweb-czb-web/czb/moreInfo?locale=enUS

TABLE IV
COMPARISON OF PROFITABILITY AND RANKING PERFORMANCE AMONG CORE MODEL COMPONENTS (AVERAGE OF 5 INDEPENDENT RUNS)

Methods	NASDAQ			NYSE			TSE			Ashare&HK		
	SR↑	IRR↑	NDCG↑	SR↑	IRR↑	NDCG↑	SR↑	IRR↑	NDCG↑	SR↑	IRR↑	NDCG↑
w/o Hyper RNN	1.37	0.45	0.77*	1.14	0.36	0.90	1.22*	0.70*	0.81	1.26	1.07	0.84
w/o Hyper Modulation	1.52*	0.48*	0.81*	1.19*	0.38*	0.91	1.27*	0.72*	0.84	1.35*†	1.09*†	0.89
w/o Hyper Temporal Encoder	1.33	0.42	0.71	1.05	0.24	0.87	1.16	0.66	0.77	1.24	1.02	0.83
w/o Prediction Router	1.51*	0.50*	0.82*	1.22*	0.41*	0.92*	1.30*†	0.75*†	0.86*	1.38*†	1.14*†	0.91*†
PTER	**1.58***†	**0.56***†	**0.85***†	**1.27***†	**0.46***†	**0.94***†	**1.33***†	**0.80***†	**0.89***†	**1.42***†	**1.18***†	**0.92***†

*and † mean the improvements over state-of-the-art STHAN-SR and ALSP-TF, respectively, are significant ($p < 0.05$) under t-test and Wilcoxon's signed-rank test.

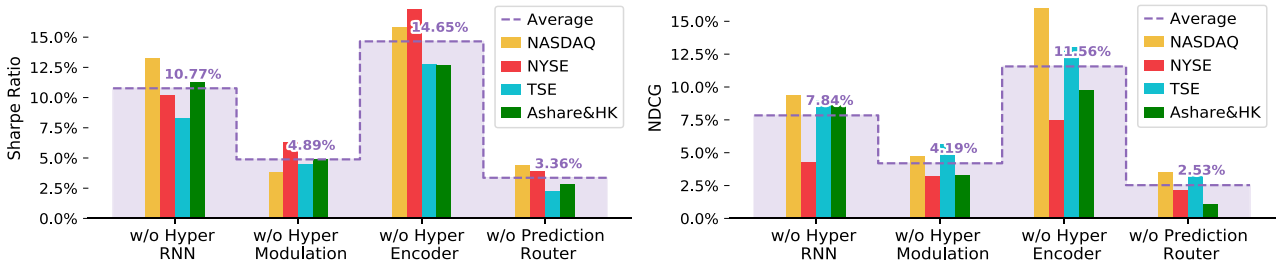


Fig. 5. Decrease in performance of the ablated models compared to the full PTER.

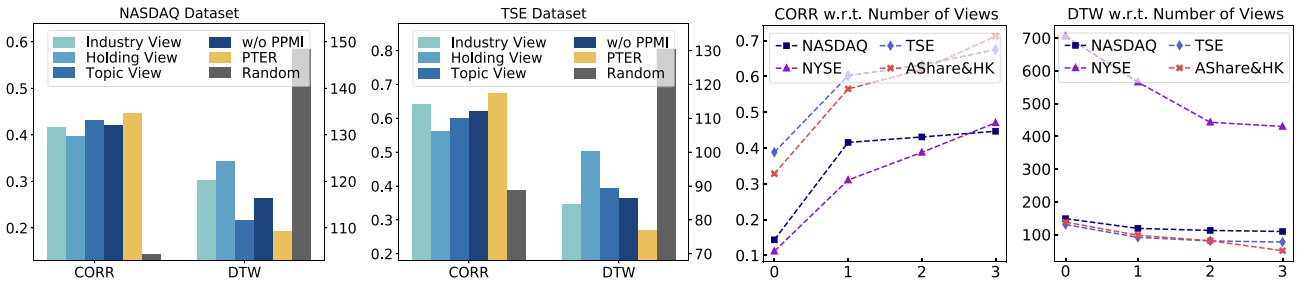


Fig. 6. Dynamic synchronization between stocks and their neighbors identified from the hyper-knowledge embedding space. Larger CORR and smaller DTW stand for higher similarity. In the line charts, hyper-knowledge is trained by increasing the number of views from 0 (i.e., random) to 3.

- *w/o prediction router*: Degenerate the predictor into one head with an MLP layer and Leaky-ReLU activation.

Table IV and Fig. 5 show the evaluation results. It can be observed that all components contribute to the overall framework. Specifically, the hyper RNN and modulation units responsible for accommodating micro and macro diversity to represent stock objects and market scenarios have the most significant impact. The average degradation of SR exceeds 10.7% and 4.8% when they are ablated. Moreover, by equipping the predictor with lightweight routing manner for inference, *PTER* can further promote the efficacy of stock forecast and investment recommendation.

2) Evaluation on Hyper-Knowledge: Next, we probe into the significance of the mined hyper-knowledge in reflecting the correlations and peculiarities of stock price movements. To verify this, for each stock object, we find its $N_{nearest}$ neighbors in the graph embedding space formed by the stock hyper-knowledge learner. Then we calculate the synchronization between the trading curves of all node-neighbor pairs on the test dataset. Both Pearson Correlation (CORR) and Dynamic Time Warping (DTW) cost [60] are taken as measurements

$$CORR(p_i, p_b) = \frac{\sum_\tau (p_i^\tau - \bar{p}_i)(p_b^\tau - \bar{p}_b)}{\sqrt{\sum_\tau (p_i^\tau - \bar{p}_i)^2}\sqrt{\sum_\tau (p_b^\tau - \bar{p}_b)^2}}$$
$$DTW(p_i, p_b) = DTW([(p_i^\tau - p_b^\mu)^2]_{\Delta L \times \Delta L}), \quad (18)$$

where p_i, p_b are the closing price series of two stocks with the length of ΔL, $\bar{p}_i, \bar{p}_b$ are the mean values, and τ, μ denote timestamps. The larger value of CORR and smaller value of DTW mean stronger correlations. We compare the effectiveness of the hyper-knowledge learned from *PTER*, a single homogeneous view of relations $\{\mathcal{G}_I, \mathcal{G}_T, \mathcal{G}_H\}$, as well as the exclusion of PPMI-based graph preprocessing. Fig. 6 shows the average results over all stocks with $N_{nearest}$=10. It can be seen that the multi-view co-generated knowledge characterizes the dynamic correlation of stocks more effectively beyond every single view. Fig. 7 shows the structure heatmaps of TSE constituent stocks. The model endows the multifaceted interrelated ones with more

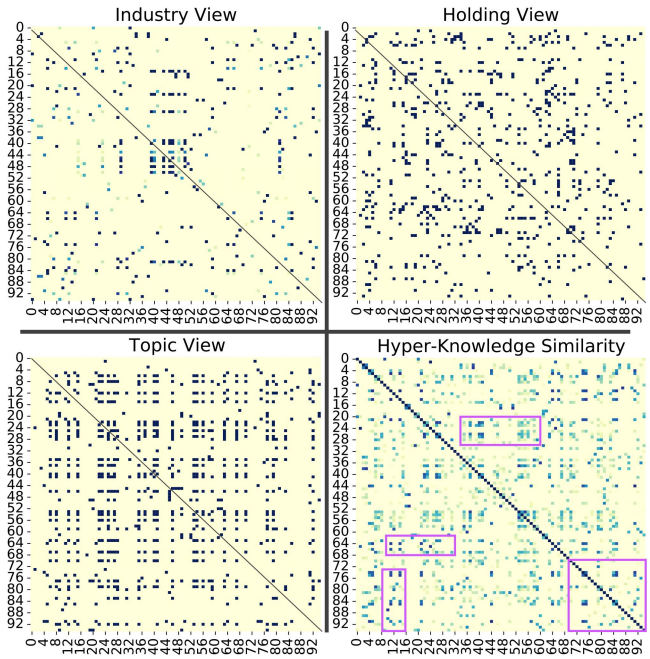


Fig. 7. Structure heatmaps of TSE constituent stocks in different relational views and the hyper-knowledge space.

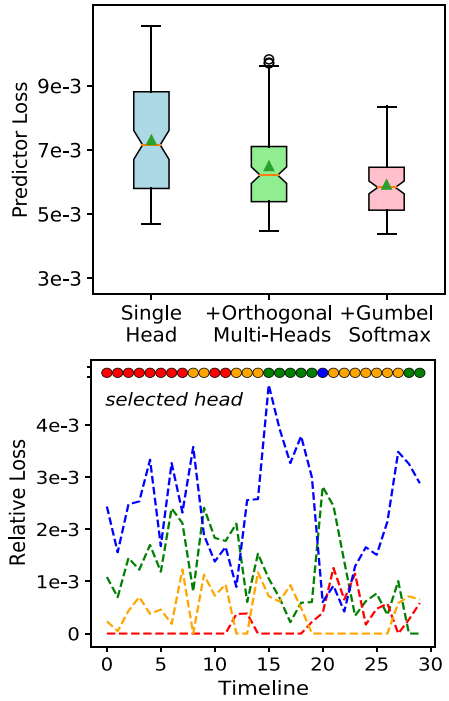


Fig. 8. Above: Overall distribution of PTER loss. Below: Comparison of losses when using different predictor heads (normalized by subtracting the minimum value), where each point corresponds to a test sample of NASDAQ dataset and is color-coded by the model's selected head.

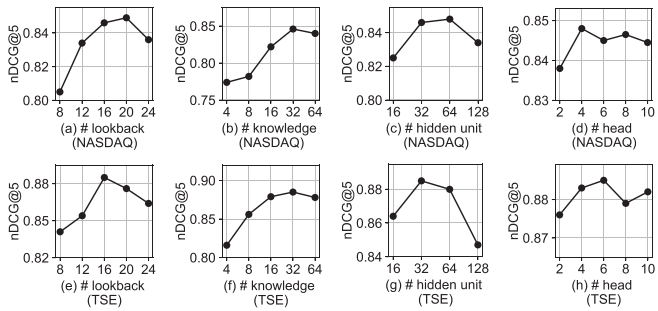


Fig. 9. Influence of different hyperparameters.

similar hyper-knowledge. This builds a reasonable bridge for building personalized stock representations. In addition, given the scale-free nature of real-world corporate graphs, applying PPMI to mitigate the hub biases is also beneficial to highlight essential dependency clues.

3) Evaluation on Prediction Router: We further investigate the role of our multi-head predictor by visualizing the loss when using different prediction settings as well as routing assignments along the timeline. Fig. 8 shows the results on the NASDAQ dataset (similar patterns observed on other datasets have been omitted due to space limitations). We find that augmenting a single prediction layer with the discrete selection of multi-heads can improve overall optimization, which aligns with the finding in Lin et al. [10] that suggests allocating stock samples to different predictors. Comparing the normalized loss values in the figure below, the relative advantage of each head varies over time. In most cases, our model can pick the optimal one with a smaller forecast loss, illustrating the efficacy of our time series encoder and predictor in shedding light on diverse volatility patterns of the stock market.

D. Hyperparameter Analysis (RQ3)

In this section, we analyze the settings of key hyperparameters in our model. We first change the lookback window ΔT of stock input time series to study its influence on model performance. The results are shown in Fig. 9(a) and (e). As ΔT increases, the performance first improves by retrieving more historical signals and then begins to decline, probably due to the diminishing relevance of outdated information to the future trend in highly volatile markets. Next, we tune the dimensions of stock hyper-knowledge. According to Fig. 9(b) and (f), the performance is suboptimal when the dimension is too small, given that relation signals between listed corporations are not fully exploited. A moderate value such as 32 is appropriate for the datasets. Note that the dimension of hyper-knowledge is independent of the size of the target network for time series embedding. It turns out the hyper-knowledge pre-trained from corporate structures is an essential contributing factor in customizing stock representation parameters.

Then, we search the effect of coefficients in the temporal encoder and predictor. Fig. 9(c) and (g) depict the dimension of hidden units in the RNN cell, which determines the network's expressiveness in extracting stock and market dynamic features. Due to the aggravation of overfitting, performance degradation is manifested when the dimension goes too large. Fig. 9(d) and (h) plot different numbers of predictor heads. Likewise, there is a peak on the performance curve. When $M < 4$ obvious improvements are achieved by using a multi-head manner, while adding more heads subsequently brings little effect. Lastly, we observe *PTER*'s profitability with varied numbers of selected

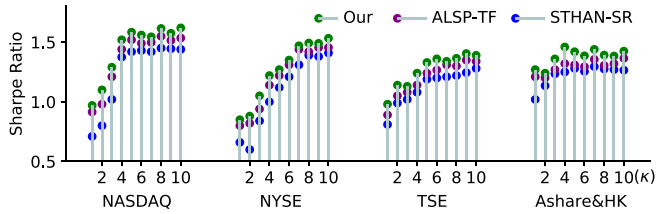


Fig. 10. Profitability with varying numbers of selected stocks.

stocks κ, which indicates the model's adaptability to trading strategies accompanied by different risk appetites. As shown in Fig. 10, *PTER* consistently obtains better results compared to state-of-the-art *STHAN-SR* and *ALSP-TF*, revealing its advantage in smartly learning network parameters to achieve stronger personalization and more accurate prediction.

VI. Conclusion

We explore the distribution shift property of stock data and propose PTER, a personalized temporal embedding and routing model for stock forecast. By forming a relaxed weight-sharing paradigm via hypernetworks, PTER can customize stock representations simultaneously considering the diversity of micro stock objects and macro market scenarios. It first mines hyper-knowledge characterizing stock similarities and peculiarities from the relationships among listed firms. Then the knowledge space is projected onto the temporal parameter space, enabling extracting and modulating protruded fluctuation patterns with self-adaptive dynamic weights. Finally, each encoded sample is dynamically dispatched to orthogonal predictor heads based on market conditions for inference. Extensive experiments on four benchmark datasets show the effectiveness and applicability of PTER. In the future, we are committed to optimizing the hyper-knowledge of stock markets, and expanding this work to broader spatio-temporal Fintech tasks such as defaulter detection and credit risk rating.

References

[1] B. G. Malkiel, "The efficient market hypothesis and its critics," *J. Econ. Perspectives*, vol. 17, no. 1, pp. 59–82, 2003.
[2] V. Rajput and S. Bobde, "Stock market forecasting techniques: Literature survey," *Int. J. Comput. Sci. Mobile Comput.*, vol. 5, no. 6, pp. 500–506, 2016.
[3] Y. Chen, Z. Wei, and X. Huang, "Incorporating corporation relationship via graph convolutional neural networks for stock price prediction," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1655–1658.
[4] R. Sawhney, A. Wadhwa, S. Agarwal, and R. Shah, "Fast: Financial news and tweet based time aware network for stock trading," in *Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2021, pp. 2164–2175.
[5] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 2141–2149.
[6] R. Sawhney, S. Agarwal, A. Wadhwa, T. Derr, and R. R. Shah, "Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 497–504.
[7] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 2627–2633.
[8] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, "Hierarchical multi-scale Gaussian transformer for stock movement prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 4640–4646.
[9] F. Feng, H. Chen, X. He, J. Ding, M. Sun, and T.-S. Chua, "Enhancing stock movement prediction with adversarial training," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 5843–5849.
[10] H. Lin, D. Zhou, W. Liu, and J. Bian, "Learning multiple stock trading patterns with temporal routing adaptor and optimal transport," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1017–1026.
[11] R. Næs, J. A. Skjeltorp, and B. A. Ødegaard, "Stock market liquidity and the business cycle," *J. Finance*, vol. 66, no. 1, pp. 139–176, 2011.
[12] E. F. Fama and K. R. French, "Size, value, and momentum in international stock returns," *J. Financial Econ.*, vol. 105, no. 3, pp. 457–472, 2012.
[13] I. Figelman, "Stock return momentum and reversal," *J. Portfolio Manage.*, vol. 34, no. 1, pp. 51–67, 2007.
[14] T. Greetham and H. Hartnett, "Investment clock special report 1: Making money from macro," Merrill Lynch, 2004.
[15] X. Yi, J. Zhang, Z. Wang, T. Li, and Y. Zheng, "Deep distributed fusion network for air quality prediction," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 965–973.
[16] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8052–8072, Aug. 2023.
[17] D. Ha, A. Dai, and Q. V. Le, "HyperNetworks," 2017, *arXiv:1609.09106*.
[18] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 2327–2333.
[19] Q. Liu, X. Cheng, S. Su, and S. Zhu, "Hierarchical complementary attention network for predicting stock price movements with news," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1603–1606.
[20] H. Wang, T. Wang, and Y. Li, "Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 971–978.
[21] R. Sawhney, M. Goyal, P. Goel, P. Mathur, and R. Shah, "Multimodal multi-speaker merger & acquisition financial modeling: A new task, dataset, and neural baselines," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics*, 2021, pp. 6751–6762.
[22] G. Kavitha, A. Udhayakumar, and D. Nagarajan, "Stock market trend analysis using hidden Markov models," 2013, *arXiv:1311.4771*.
[23] R. K. Nayak, D. Mishra, and A. K. Rath, "A naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices," *Appl. Soft Comput.*, vol. 35, pp. 670–680, 2015.
[24] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS One*, vol. 12, no. 7, 2017, Art. no. e0180944.
[25] H. Wang, S. Li, T. Wang, and J. Zheng, "Hierarchical adaptive temporal-relational modeling for stock trend prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 3691–3698.
[26] Y. Liu, Q. Liu, H. Zhao, Z. Pan, and C. Liu, "Adaptive quantitative trading: An imitative deep reinforcement learning approach," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 2128–2135.
[27] S. Carta, A. Ferreira, A. S. Podda, D. R. Recupero, and A. Sanna, "Multi-DQN: An ensemble of deep Q-learning agents for stock market forecasting," *Expert Syst. Appl.*, vol. 164, 2021, Art. no. 113820.
[28] K. Xu, Y. Zhang, D. Ye, P. Zhao, and M. Tan, "Relation-aware transformer for portfolio policy learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 4647–4653.
[29] H. Wang, T. Wang, S. Li, J. Zheng, S. Guan, and W. Chen, "Adaptive long-short pattern transformer for stock investment selection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2022, pp. 3970–3977.
[30] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2292–2300.
[31] K. Muandet, D. Balduzzi, and B. Schölkopf, "Domain generalization via invariant feature representation," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 10–18.
[32] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, "Invariant risk minimization," 2019, *arXiv:1907.02893*.
[33] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3490–3497.
[34] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[35] Q. Qian, S. Zhu, J. Tang, R. Jin, B. Sun, and H. Li, "Robust optimization over multiple domains," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4739–4746.
[36] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer, "Robust and generalizable visual representation learning via random convolutions," in *Proc. Int. Conf. Learn. Representations*, 2021.
[37] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, "Domain generalization with mixstyle," in *Proc. Int. Conf. Learn. Representations*, 2021.
[38] M. MacKay, P. Vicol, J. Lorraine, D. Duvenaud, and R. Grosse, "Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions," 2019, *arXiv:1903.03088*.
[39] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1720–1730.
[40] Q. Zhang, X. Liao, Q. Liu, J. Xu, and B. Zheng, "Leaving no one behind: A multi-scenario multi-task meta learning approach for advertiser modeling," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 1368–1376.
[41] J. Suarez, "Language modeling with recurrent highway hypernetworks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3267–3276.
[42] T. Ma, A. V. Dalca, and M. R. Sabuncu, "Hyper-convolution networks for biomedical image segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2022, pp. 1933–1942.
[43] A. Shamsian, A. Navon, E. Fetaya, and G. Chechik, "Personalized federated learning using hypernetworks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 9489–9502.
[44] R. Sawhney, S. Agarwal, A. Wadhwa, and R. Shah, "Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading," in *Proc. Web Conf.*, 2021, pp. 11–22.
[45] E. Littwin, T. Galanti, L. Wolf, and G. Yang, "On infinite-width hypernetworks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 13226–13237.
[46] T. Galanti and L. Wolf, "On the modularity of hypernetworks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 10409–10419.
[47] M. Steliaros and D. C. Thomas, "The cross-sectional variability of stock-price returns: Country and sector effects revisited," *J. Asset Manage.*, vol. 7, pp. 273–290, 2006.
[48] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Trans. Inf. Syst.*, vol. 37, no. 2, pp. 1–30, 2019.
[49] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2177–2185.
[50] X. Wang, N. Liu, H. Han, and C. Shi, "Self-supervised heterogeneous graph neural network with co-contrastive learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1726–1736.
[51] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1227–1235.
[52] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[53] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," 2016, *arXiv:1611.01144*.
[54] W. Li, R. Bao, K. Harimoto, D. Chen, J. Xu, and Q. Su, "Modeling the stock relation with graph network for overnight stock movement prediction," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 4541–4547.
[55] J. Huang, Y. Zhang, J. Zhang, and X. Zhang, "A tensor-based sub-mode coordinate algorithm for stock prediction," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace*, 2018, pp. 716–721.
[56] J.-H. Wang and J.-Y. Leu, "Stock market trend prediction using ARIMA-based neural networks," in *Proc. Int. Conf. Neural Netw.*, 1996, pp. 2160–2165.
[57] R. Kim, C. H. So, M. Jeong, S. Lee, J. Kim, and J. Kang, "HATS: A hierarchical graph attention network for stock movement prediction," 2019, *arXiv:1908.07999*.
[58] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[60] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, 2011.

Heyuan Wang received the PhD degree from the School of Computer Science, Peking University, Beijing, China. She has published several papers at top-tier conferences and journals, such as *IEEE Transactions on Knowledge and Data Engineering*, ACL, IJCAI, AAAI, CIKM, etc. Her research interests include graph neural networks, time series modeling, and recommender system.

Tengjiao Wang is a professor with the School of Computer Science, Peking University. He is also dean of the Institute of Computational Social Science, Peking University (Qingdao). He is mainly engaged in research and development of Big Data management, data warehouse and data mining, artificial intelligence. He has published dozens of papers with SIGMOD, SIGKDD, VLDB, *IEEE Transactions on Knowledge and Data Engineering*, IJCAI, AAAI, ACL, etc.

Shun Li received the both BS and PhD degrees from Peking University. He is an associate professor of computer science with the University of International Relations (UIR). His research interests include data mining, machine learning, and Big Data analysis.

Jiayi Zheng received the both BS and MS degrees from the School of Computer Science, Peking University. His research interests include graph neural networks, natural language understanding, and recommender systems. He has published several papers with the top conferences including IJCAI, *IEEE Transactions on Knowledge and Data Engineering*, etc.

Weijun Chen is currently working toward the PhD degree with the School of Computer Science, Peking University (PKU), Beijing, China. He is working in National Engineering Laboratory for Big Data Analysis and Applications, PKU. His research interests include graph neural networks, data mining, and time-series forecast.

Wei Chen is currently an associate researcher with the School of Computer Science, Peking University. Her research interests include artificial intelligence and Big Data management.