Adaptive Long-Short Pattern Transformer for Stock Investment Selection

Heyuan Wang[1,3], **Tengjiao Wang**[1,3], **Shun Li**[2*], **Jiayi Zheng**[1,3], **Shijie Guan**[1,3] and **Wei Chen**[1,3]

[1]School of Computer Science, National Engineering Laboratory for Big Data Analysis and Applications, Peking University, China

[2]University of International Relations

[3]Institute of Computational Social Science, Peking University (Qingdao)

{wangheyuan, tjwang, jiayizheng, guanshijie, pekingchenwei}@pku.edu.cn, lishunmail@foxmail.com

Abstract

Stock investment selection is a hard issue in the *Fintech* field due to non-stationary dynamics and complex market interdependencies. Existing studies are mostly based on RNNs, which struggle to capture interactive information among fine granular volatility patterns. Besides, they either treat stocks as isolated, or presuppose a fixed graph structure heavily relying on prior domain knowledge. In this paper, we propose a novel Adaptive Long-Short Pattern Transformer (ALSP-TF) for stock ranking in terms of expected returns. Specifically, we overcome the limitations of canonical self-attention including context and position agnostic, with two additional capacities: (i) *fine-grained pattern distiller* to contextualize queries and keys based on localized feature scales, and (ii) *time-adaptive modulator* to let the dependency modeling among pattern pairs sensitive to different time intervals. Attention heads in stacked layers gradually harvest short- and long-term transition traits, spontaneously boosting the diversity of stock representations. Moreover, we devise a *graph self-supervised regularization*, which helps automatically assimilate the collective synergy of stocks and improve the generalization ability of overall model. Experiments on three exchange market datasets show ALSP-TF's superiority over state-of-the-art stock forecast methods.

1 Introduction

As an essential ingredient of modern financial ecosystem, the forecast of stock market has aroused extensive research interest due to scientific and investment merits [Feng *et al.*, 2019b]. Different from the general time series modeling, it is inherently difficult to assess stock's evolving trend confronting with highly volatile and interrelated natures of the market. Traditional literatures leverage machine learning algorithms based on manual indicators [Nayak *et al.*, 2015; Khaidem *et al.*, 2016], where the hypothetical stochastic process may become stranded in catching non-stationary oscillations. In recent studies, deep neural networks have

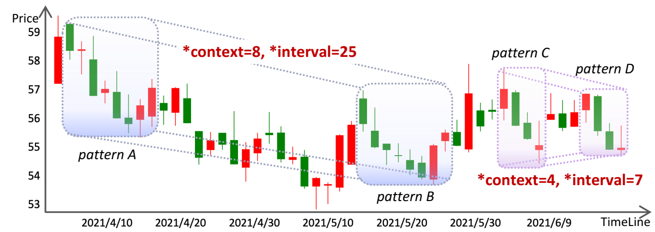


Figure 1: Interaction of patterns along stock (*SMIC*) dynamic series. $\{A, B\}$ and $\{C, D\}$ have varied context spans and time intervals.

shown encouraging prospects in characterizing stock dynamics, especially the RNNs are the dominant choice. For instance, Zhang *et al.* [2017] used state frequency memory in the LSTM network to decompose stock transaction patterns. Sawhney *et al.* [2021] equipped LSTM with temporal attention to reward driving hidden signals. In view of the short of RNNs in capturing granular feature units, Wang *et al.* [2021] proposed to apply gated causal CNNs to convolve price subsequences. However, both of them are ineffective to learn the dependencies of long-range discontinuous temporal states.

As an alternative, the well-known Transformer [Vaswani *et al.*, 2017] preliminarily designed in the NLP field has gained huge success in learning sequential data [Ding *et al.*, 2020; Zhou *et al.*, 2021]. The core of Transformer is *multi-head self-attention*, which explicitly performs information exchange between input tokens to fix the deficiencies of RNN and CNN structures. However, directly applying the canonical encoder to modeling stock movements is problematic in two non-neglectable aspects: (1) The global self-attention focuses on point-wise token similarities without contextual insights [Xu *et al.*, 2020]. Since stock fluctuations are conditioned on composite signals over manifold time spans, the lack of pattern-wise interaction hinders the adequate discrimination of stock tendency and is susceptible to noise points. (2) The basic query-key matching paradigm is position agnostic. Though sinusoidal position embedding is inserted to the sequential input, it may not be optimal due to the inability to reveal precise distances [Wu *et al.*, 2021]. As an empirical example, in Fig. 1, the subsequences $\{A, B\}$ and $\{C, D\}$ reflect stock wave patterns in the context of different long- and short-term spans; Intuitively, conducting multi-granular matching between them instead of simple dot-point projec-

[*]Corresponding Author.

tions is more conducive to characterizing the evolution status. Additionally, the patterns C and D are closer than A and B, which means that their interaction is more responsible for observing high-frequency regularities. This inspires us to factor in the elapsed time between stock change patterns so as to modulate the self-attention on stock trading series.

The synergy effect is another prominent trait of stock market, i.e, related stocks are apt to exhibit synchronous changes, offering a desiderative pointcut for trend predictions. Nevertheless, it is non-trivial to anchor full-scale stock interdependencies given that relationship sources may originate from various aspects (*industry rotation, common shareholder, supply chain, etc*). Previous works mostly treat individual stocks as isolated [Wang *et al.*, 2020; Ding *et al.*, 2020]. Some studies presuppose a graph structure by resorting to limited domain knowledge [Chen *et al.*, 2018; Feng *et al.*, 2019b; Sawhney *et al.*, 2021], which may lead to information bias in revealing intricate market factors. More particularly, task-specific graph building requires massive expertise, which prevents the model from being applied to extended scenarios.

Along these lines, in this paper, we present *Adaptive Long-Short Pattern Transformer* (ALSP-TF) for stock investment selection. The model is structurally innovated for hierarchical representation and interaction of stock price series at different context scales. In addition, with the help of a learnable sigmoid function, we make the self-attention aware of the weighted time intervals between patterns, in order to adaptively adjust their dependencies beyond similarity matching. Further, to get rid of reliance on any prior knowledge, we devise a *graph self-supervised regularization* which automatically learns stock topology through dynamic path alignment, and thereby boosting the generalization capacity of the overall model. Our major contributions are as follows:

- We delve into the issues of basic Transformer in modeling stock price series, and present a reformed self-attention encoder to exploit adaptive pattern-wise interactions supported by temporal representations at different grain levels.
- We construct a data-driven adjacency graph to uncover the implicit similarities in volatility across different stocks. It helps reduce the stochasticity of stock input sequences and serves as a self-supervision signal to guide the model's representation learning.
- All the components are seamlessly integrated and jointly trained to predict stock expected returns. Experiments on datasets from NYSE, NASDAQ and TSE markets show the effectiveness of proposed model for investment selection.

2 Related Work

Technical Analysis. TA is at the heart of stock trend prediction, which is developed on top of price-volume indicators from historical quote data. Traditional mathematical algorithms such as HMM, SVM [Kavitha *et al.*, 2013; Nayak *et al.*, 2015; Khaidem *et al.*, 2016] leverage manual feature engineering and hypothetical stochastic processes. Later studies move to exploiting deep neural networks to model the hidden dependency of stock dynamics, where RNNs are commonly utilized [Qin *et al.*, 2017; Zhang *et al.*, 2017; Wang *et al.*, 2020]. To enhance the capacity of handling fine-granular transition signals, some efforts have explored other architectures such as hybrid SAE-LSTM units [Bao *et al.*, 2017], adversarial training [Feng *et al.*, 2019a], and gated causal convolutions [Wang *et al.*, 2021]. These approaches have made progress in several stock prediction tasks, whereas they are commonly framed for the regression of prices or classification of bucketing stock movements. The latest work [Sawhney *et al.*, 2021] claimed that the absence of optimization toward expected returns will harm practical investment choices. They reformulated stock forecasts as a learning to rank task and realized state-of-the-art profitability.

Market Relation Modeling. A new line of studies revolve around employing the collective synergy among stocks based on their metadata relevance. For instance, Lai *et al.* [2017] acquired stock relatedness by querying company *collaboration* and *competition* from the search engine, then made inferences based on unary and binary potentials in Markov random fields. Chen *et al.* [2018] built a graph of corporations based on their shareholding properties, and transferred stock prediction to node classification using graph convolutional network (GCN) [Kipf and Welling, 2017]. Feng *et al.* [2019b] clustered stocks from the same industries and supply chains to make the temporal price encoder aware of inter-stock relations. Sawhney *et al.* [2021] augmented the corporate relevance based on Wikidata and used hypergraph convolution to propagate higher-order neighbor's information. Despite advances in graph-based stock forecasts, the preset knowledge-based stationary graph structure requires extensive domain expertise and may strain model's extensibility.

Transformer. A powerful attention neural model that is preliminarily designed for machine translation [Vaswani *et al.*, 2017] and now has attained huge success in various fields such as computer vision, multimodal reasoning and video classification [Gao *et al.*, 2019; Dosovitskiy *et al.*, 2021]. There are several recent studies in applying Transformer to time series modeling. Zhou *et al.* [2021] devoted to optimizing the efficiency of time complexity and memory usage of Transformer on extremely long time series. Ding *et al.* [2020] first tried to exploit Transformer on trading sequences to classify stock price movements. Differently, we delve and revamp the deficiencies of basic Transformer in grasping important context and temporal information of stock volatility patterns.

3 Methodology

3.1 Problem Formulation

To avoid the gap between stock movement prediction and investment profit, we follow the setup of [Sawhney *et al.*, 2021] and adopt a learning to rank formulation for stock selection. Given the candidate set $\mathcal{S} = \{s_1, \ldots, s_N\}$ of N stocks, on any trading day t, each stock s_i entails a feature sequence $\mathcal{X}_i = [\boldsymbol{x}_{t-\Delta T}, \ldots, \boldsymbol{x}_{t-1}] \in \mathbb{R}^{\Delta T \times F}$ of historical ΔT time steps (where F is the feature dimension), an associated closing price p_i^t and a 1-day return ratio $r_i^t = \frac{p_i^t - p_i^{t-1}}{p_i^{t-1}}$. The model $\mathcal{F}_\theta(\mathcal{X}_{[1:N]})$ aims to output an ordering of all stocks $\mathcal{Y}^t = \{y_1^t > y_2^t \ldots > y_N^t\}$, where the top-ranked ones are expected to gain more investment revenues on day t.

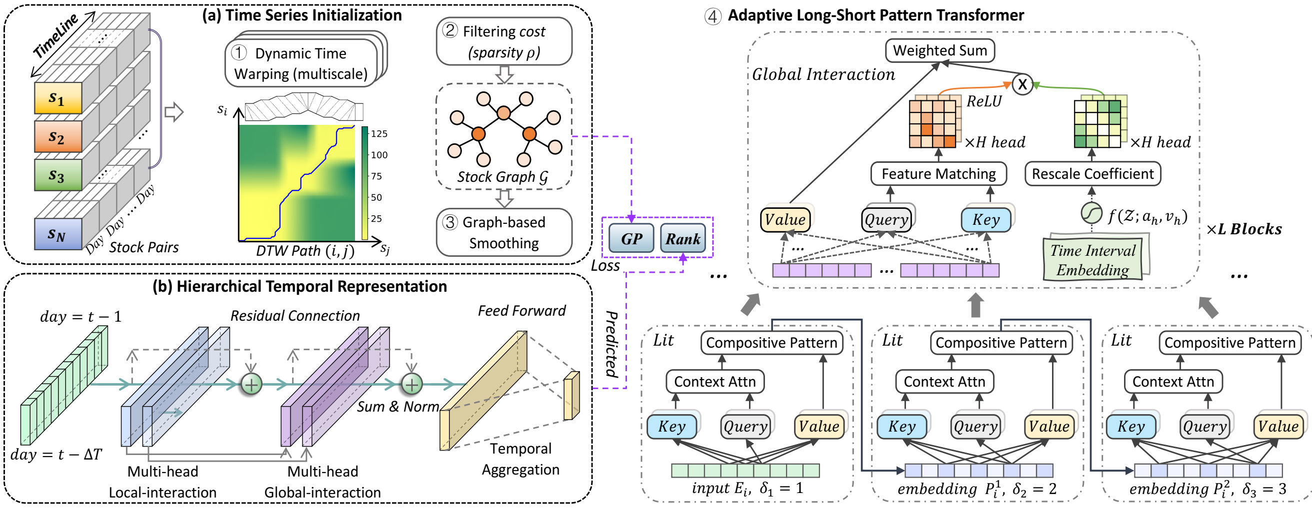


Figure 2: Overview of the proposed ALSP-TF.

3.2 Framework Overview

Fig. 2 illustrates the overview of **ALSP-TF**. It consists of three major parts: (i) *Graph-based Initialization* to shift the dynamic path alignment of stock pairs into a topology structure, and apply a gated graph smoothing operation to enrich the initial embeddings of stock time series; (ii) *Reformed Transformer Encoder* with hierarchical blocks of multi-head self-attention to exploit interdependencies between different stock volatility patterns. Each block jointly leverages *fine-grained pattern distiller* and *time-adaptive modulator* to calculate the attention scores on the basis of contextualized feature units; (iii) At last, regularized graph self-supervision signals are added to tune the entire stock ranking framework.

3.3 Time Series Embedding Initialization

Stock Graph Construction

The future movement of a stock is conditioned on its historical dynamic patterns as well as its neighbors' synchronous information. To incorporate inter-stock dependencies, recent research relies to a large extent on prior domain knowledge, which can be labor-intensive, difficult to adequately reveal intricate market factors, and may narrow the applicability to new prediction tasks. In this regard, we propose to learn the implicit collective synergy among all candidate stocks in a data-driven manner. Specifically, we utilize a proximity function by applying multi-dimensional Dynamic Time Warping (DTW) [Jeong *et al.*, 2011] on the input signals $\mathcal{X}_{[1:N]}$. It finds the minimum alignment cost of each pair of stock sequences through dynamic programming on a path matrix $\mathcal{D}$:

$$cost(s_i, s_j) \Leftarrow DTW(\{\mathcal{D}_{pq}\}_{\Delta T \times \Delta T})\,. \tag{1}$$

Here we set each pixel $\mathcal{D}_{pq} = \sum_{f=1}^{F}(\mathcal{X}_i^{p,f} - \mathcal{X}_j^{q,f})^2$. Then, we attach an edge between the stock pairs whose cost value is less than a limit threshold (by sorting alignment costs and controlling the graph sparsity ρ) to build the stock graph $\mathcal{G}$.

Graph-based Sequence Smoothing

To further reduce the stochasticity of stock sequences, we develop a graph-based smoothing operation which substantiates stock proximity correlations using neighboring features. Let $\widehat{\mathcal{A}} \in \mathbb{R}^{N\times N}$ denote $\mathcal{G}$'s normalized adjacency matrix with self-loops and $\mathbf{e}^0_{[1:N]} \in \mathbb{R}^{(N\times\Delta T)F}$ the input signals of $\mathcal{X}_{[1:N]}$, we utilize the aggregation layer of graph attention (GAT) network [Velickovic *et al.*, 2017] to convolve the k-order neighbor messages of stock s_i into embedding of $\mathbf{a}_i^k$. Then we engage a GRU-like gate manner to rectify the fusing proportion in node updating process ($\mathbf{e}_i^k = \mathrm{Gate}(\mathbf{e}_i^{k-1}, \mathbf{a}_i^k)$):

$$\begin{cases} \boldsymbol{z}_i^k &= \sigma(\boldsymbol{W}_z \boldsymbol{a}_i^k + \boldsymbol{U}_z \boldsymbol{e}_i^{k-1} + \boldsymbol{b}_z)\,, \\ \boldsymbol{r}_i^k &= \sigma(\boldsymbol{W}_r \boldsymbol{a}_i^k + \boldsymbol{U}_r \boldsymbol{e}_i^{k-1} + \boldsymbol{b}_r)\,, \\ \tilde{\boldsymbol{e}}_i^k &= \tanh(\boldsymbol{W}_h \boldsymbol{a}_i^k + \boldsymbol{U}_h (\boldsymbol{r}_i^k \odot \boldsymbol{e}_i^{k-1}) + \boldsymbol{b}_h)\,, \\ \boldsymbol{e}_i^k &= \tilde{\boldsymbol{e}}_i^k \odot \boldsymbol{z}_i^k + \boldsymbol{e}_i^{k-1} \odot (1 - \boldsymbol{z}_i^k)\,, \end{cases} \tag{2}$$

where $\odot$ is Hadamard product, $\boldsymbol{W}_*$, $\boldsymbol{U}_*$, $\boldsymbol{b}_*$ are trainable weights and biases. $\boldsymbol{r}_i^k$ and $\boldsymbol{z}_i^k$ are reset and update gates, which are responsible for catching irrelevant information to forget and the part of past state to move forward, respectively.

3.4 Transformer Encoder

Next, we describe our reformed Transformer encoder to adaptively capture the interactive information between short- and long-term volatility patterns with different time intervals. The representation hierarchy consists of L blocks of multi-head self-attention and feed-forward layers. Taken initialized stock embedding sequences $\boldsymbol{E} \in \mathbb{R}^{N\times\Delta T\times\bar{d}}$ as input, the canonical self-attention in [Vaswani *et al.*, 2017] performs information exchange between every pair of time points for each stock s_i. Specifically, it transforms $\boldsymbol{E}_i \in \mathbb{R}^{\Delta T\times\bar{d}}$ (the series of s_i) into query $\boldsymbol{Q}_{i,h} = \boldsymbol{E}_i \boldsymbol{W}_h^Q$, key $\boldsymbol{K}_{i,h} = \boldsymbol{E}_i \boldsymbol{W}_h^K$ and value matrices $\boldsymbol{V}_{i,h} = \boldsymbol{E}_i \boldsymbol{W}_h^V$ with distinct linear projection parameters, where $h = 1, \ldots, H$ is the head index, and

$\boldsymbol{W}_h^Q, \boldsymbol{W}_h^K, \boldsymbol{W}_h^V \in \mathbb{R}^{\bar{d} \times d_f}$. Then scaled dot-product attention scores are computed to acquire a weighted sum of the sequential values. Afterwards, the final layer output is represented by the concatenation of all attention heads:

$$\begin{aligned} \boldsymbol{F}_i &= \text{Multihead}(\boldsymbol{Q}_{i,h}, \boldsymbol{K}_{i,h}, \boldsymbol{V}_{i,h}) \\ &= ||_{h=1}^{h=H} \text{Softmax}(\boldsymbol{Q}_{i,h} \boldsymbol{K}_{i,h}^\top / \sqrt{d_f}) \boldsymbol{V}_{i,h} \,, \end{aligned} \tag{3}$$

where $||$ is the concatenation operator and d_f is the dimension of projected feature space. Thereby all stocks' representations are formed as $\boldsymbol{F} = [\boldsymbol{F}_1; \boldsymbol{F}_2; \ldots; \boldsymbol{F}_N] \in \mathbb{R}^{N \times \Delta T \times H d_f}$ followed by feed-forward layers.

Enhancing Locality with Fine-grained Pattern Distiller

The dot-product attention calculated on top of point-wise tokens exhibits a powerful ability in extracting global dependencies for words in NLP and regions in CV. Whereas, the compound patterns implied in different short- and long-term local time spans are both important to shed light on the intricate stock market dynamics, while cannot be well exploited in such scheme. To solve it, in each block, we inject a local interaction (*Lit*) layer before global matching such that compositive signals are shifted into new contextualized query-key tuples. To save computational consumptions, we borrow the concept of dilated causal convolutions in WaveNet [Oord *et al.*, 2016], and skip interval "holes" on cascading *Lit* layers instead of making projection over contiguous subsequences in $\boldsymbol{E}_i$ to obtain wider receptive fields hierarchically.

Regarding *Lit* at the 1_{st} block, at any time-step τ we apply self-attention to process its surrounding w-length context, i.e, $\bar{\boldsymbol{E}}_i^{1\to\tau} = [\boldsymbol{E}_i^{\tau-w+1}, \ldots, \boldsymbol{E}_i^{\tau}] \in \mathbb{R}^{w \times \bar{d}}$ (we use padding when $\tau \le w$). Note that the context is a predecessor time period from τ, which is like a position mask to filter out temporal attention to future information. Therefrom we represent transformed key-value tuples as $\bar{\boldsymbol{K}}_{i,h}^\tau = \bar{\boldsymbol{V}}_{i,h}^\tau = \bar{\boldsymbol{E}}_i^{1\to\tau} \bar{\boldsymbol{W}}_h^Q \in \mathbb{R}^{w \times d_f}$, and form the state of current time-step as a query matrix $\bar{\boldsymbol{Q}}_{i,h}^\tau = \boldsymbol{E}_i^\tau \bar{\boldsymbol{W}}_h^Q \in \mathbb{R}^{1 \times d_f}$. After that, we exploit the context dependency and attentively sum up localized elements as a specific *compound pattern* for the moment τ:

$$\boldsymbol{P}_i^{1\to\tau} = \text{Multihead}(\bar{\boldsymbol{Q}}_{i,h}^\tau, \bar{\boldsymbol{K}}_{i,h}^\tau, \bar{\boldsymbol{V}}_{i,h}^\tau) \,. \tag{4}$$

By concatenating all time-steps, the embedding output of this layer is denoted as $\boldsymbol{P}_i^1 \in \mathbb{R}^{\Delta T \times H d_f}$. Further, in the higher $(l+1)_{th}$ block, we pass $\boldsymbol{P}_i^l$ as input to *Lit* and δ_{l+1} a wider skipping distance to handle longer-term local contexts $\bar{\boldsymbol{E}}_i^{l+1\to\tau} = ||_{\mu=0}^{\mu=w-1} \boldsymbol{P}_i^{l\to(\tau-\mu\delta_{l+1})} \in \mathbb{R}^{w \times H d_f}$ along time-steps. In this way, the local receptive fields in the stacking *Lit*s are exponentially expanded, supported by linearly growing parameters (e.g, we can obtain hierarchical patterns over 3→7→15 days by setting the draw-out size as $w = 3$ and skipping distances as 1→2→4). The $(l+1)$-level *compound patterns* $\boldsymbol{P}_i^{l+1}$ are derived in the same manner of Eq. 4 with different parameters $\bar{\boldsymbol{W}}_h^Q$. Starting with $\delta_1 = 1$ to ensure no loss of coverage on the sequence, we devise $\delta_{[2:L]}$ according to the performance in validation to gradually distill L different granularities of transition patterns for all stock sequences.

Pattern Interaction with Time-adaptive Modulator

Building on the fine-grained patterns distilled from *Lit* at each block, we turn to capturing global intra dependencies across the entire sequence. To this end, the position information of time series plays a critical role in measuring the variant of elapsed time between patterns. The canonical Transformer adds sinusoidal relative position embeddings to the input, which is proved weak due to loss of precise distance information. Let $\mathcal{Z}_{\tau,\mu} = |\tau - \mu|$ denote the distance (i.e, temporal intervals) between the τ_{th} and μ_{th} moments, we construct a matrix $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{\Delta T \times \Delta T}$ to indicate temporal distance signals between every pair of patterns in the embedding sequence $\boldsymbol{P}_i^l$ $(i \in \{1,..,N\}, l \in \{1,..,L\})$. Motivated by [Wu *et al.*, 2021], we use a learnable sigmoid function $f(\cdot)$ to rescale raw temporal values into an appropriate range, which is bounded, tunable and monotone to guarantee the model training:

$$\hat{\boldsymbol{\mathcal{Z}}}_h = f(\boldsymbol{\mathcal{Z}}; a_h, v_h) = \frac{1 + \exp(v_h)}{1 + \exp(v_h - a_h \boldsymbol{\mathcal{Z}})} \,, \tag{5}$$

where a_h is a weight parameter that determines whether preferring to capture long-distance interactions (with a positive value) or focusing on short-distance dependencies (with a negative value). v_h controls the curve's upperbound and ascending steepness, whose larger value means intenser effect of the time distance between patterns. Both a_h and v_h are tailored for the h_{th} global attention head in a learnable way.

Afterwards, we use the time-adaptive coefficients to adjust the attention operation inside Eq. 3. At each level of granularity l, we linearly transform the local pattern sequence $\boldsymbol{P}_i^l$ and compute $\text{Multihead}(\boldsymbol{Q}_{cross}^{l,i,h}, \boldsymbol{K}_{cross}^{l,i,h}, \boldsymbol{V}_{cross}^{l,i,h})$ as follows:

$$\boldsymbol{F}_{l,i} = ||_{h=1}^{h=H} \text{softmax}\Big(\frac{\text{ReLU}(\boldsymbol{Q}_{cross}^{l,i,h} \boldsymbol{K}_{cross}^{l,i,h\top}) * \hat{\boldsymbol{\mathcal{Z}}}_h}{\sqrt{d_f}}\Big) \boldsymbol{V}_{cross}^{l,i,h} \,, \tag{6}$$

where $\boldsymbol{Q}_{cross}^{l,i,h} = \boldsymbol{P}_i^l \tilde{\boldsymbol{W}}_h^Q$, $\boldsymbol{K}_{cross}^{l,i,h} = \boldsymbol{P}_i^l \tilde{\boldsymbol{W}}_h^K$, $\boldsymbol{V}_{cross}^{l,i,h} = \boldsymbol{P}_i^l \tilde{\boldsymbol{W}}_h^V$, $*$ means element-wise product, ReLU activation is applied to keep non-negativity and sharpen the original attention weights. Hence, the feature similarity and time distance are jointly measured to make the volatility patterns in the l_{th} block crossly attend to each other. We keep one feed-forward network and residual connection of canonical Transformer to further process $\boldsymbol{F}_{l,i}$. The transformation matrices are shared for all stocks. Finally, by averaging all steps of each layer and then concatenating multi-granular embeddings, we represent the stock set $\mathcal{S}$ as a compact tensor $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{N \times (L \times H d_f)}$.

3.5 Prediction and Network Optimization

Rank Loss. For ranking optimization, we acquire stock predicted return ratios on day t by feeding stock representations $\boldsymbol{\mathcal{O}}$ to a dense layer with the activation of Leaky-ReLU. Then we jointly compute the point-wise regression and pair-wise ranking loss with a weighting coefficient α, to minimize the discrepancy between predicted $\hat{r}_{[1:N]}^t$ and ground-truth $r_{[1:N]}^t$ meanwhile maintaining the relative order of stocks:

$$\mathcal{L}_{\mathcal{R}} = \sum_{i=1}^{N} \left\| \hat{r}_i^t - r_i^t \right\|^2 + \alpha \sum_{i=1}^{N} \sum_{j=1}^{N} \max\left(0, -\left(\hat{r}_i^t - \hat{r}_j^t\right)\left(r_i^t - r_j^t\right)\right) \,. \tag{7}$$

	Methods		NASDAQ		NYSE		TSE	
			SR	*IRR*	*SR*	*IRR*	*SR*	*IRR*
CLF	**ARIMA** [Wang and Leu, 1996]	Recurrent neural network using features extracted from ARIMA analyses	0.55	0.10	0.33	0.10	0.47	0.13
CLF	**Adv-ALSTM** [Feng *et al.*, 2019a]	Simulate the stochasticity of stock dynamics with adversarial training	0.97	0.23	0.81	0.14	1.10	0.43
CLF	**HGCluster** [Luo *et al.*, 2014]	Translate stock trend prediction into clustering of the hypergraph	0.06	0.10	0.10	0.11	0.20	0.10
CLF	**HATS** [Kim *et al.*, 2019]	Use hierarchical attention network to model stock multigraphs	0.80	0.15	0.73	0.12	0.96	0.31
CLF	**HMG-TF** [Ding *et al.*, 2020]	Apply Gaussian Transformer on daily and weekly trading series	0.83	0.19	0.75	0.13	1.05	0.33
CLF	**LSTM-RGCN** [Li *et al.*, 2021]	LSTM + GCN for modeling stock relational graph	0.75	0.13	0.70	0.10	0.90	0.28
CLF	**HATR** [Wang *et al.*, 2021]	Temporal-relational modeling with gated convolution + diffusion GCN	0.92	0.31	0.76	0.14	0.98	0.36
REG	**SFM** [Zhang *et al.*, 2017]	RNNs + DFT-based state frequency memory to extract volatility patterns	0.16	0.09	0.19	0.11	0.08	0.07
REG	**DA-RNN** [Qin *et al.*, 2017]	RNNs + Dual-stage attentions to reward driving input and hidden states	0.71	0.14	0.66	0.13	0.86	0.25
RL	**DQN** [Carta *et al.*, 2021]	Ensemble of deep Q-learning agents to maximize a return function	0.93	0.20	0.72	0.12	1.08	0.31
RL	**iRDPG** [Liu *et al.*, 2020]	Imitate RDPG model to learn trading policies for reward of Sharpe Ratio	1.32	0.28	0.85	0.18	1.10	0.55
RL	**RAT** [Xu *et al.*, 2020]	Relation-aware Transformer under RL framework for portfolio selection	1.37	0.40	1.03	0.22	*1.20*	*0.64*
RAN	**SAE-LSTM** [Bao *et al.*, 2017]	Stacked autoencoders + LSTM to forecast stock price for ranking	0.95	0.22	0.79	0.12	0.73	0.21
RAN	**RSR-E** [Feng *et al.*, 2019b]	Temporal GCN using similarity of feature vectors as relation strength	1.12	0.26	0.88	0.20	1.07	0.50
RAN	**RSR-I** [Feng *et al.*, 2019b]	Temporal GCN using neural net to compute relation strength	1.34	0.39	0.95	0.21	1.08	0.53
RAN	**STHAN-SR** [Sawhney *et al.*, 2021]	Attentive LSTM + hypergraph attention on multiple relationships	*1.42*	*0.44*	*1.12*	*0.33*	1.19	0.62
RAN	**ALSP-TF** (Ours)	Adaptive Long-Short Pattern Transformer with self-supervised regularization	**1.55***	**0.53***	**1.24***	**0.41***	**1.27***	**0.71***

Table 1: Profitability comparison with *Classification (CLF), Regression (REG), Reinforcement Learning (RL)*, and *Ranking (RAN)* baselines. Relation-exploited studies leverage the knowledge of industry, first- and second-order Wikidata corporate relationships defined in STHAN-SR. Bold & underlines depict the best & second-best results. $\star$ means the improvement over SOTA is statistically significant ($p < 0.01$).

Graph Proximity Loss. To ensure that the learned stock embeddings can effectively capture the correlation information stored in the alignment graph structure $\mathcal{G}$, we further introduce a graph reconstruction strategy, which regulates stock representations by explicitly drawing closer the node neighbors (we denote $\mathcal{N}_i$ as the 1-hop neighbors of s_i including itself) and pushing farther the negative ones in feature space:

$$\mathcal{L}_{\mathcal{GP}}(i) = -\sum_{j\in\mathcal{N}(i)} \log(\sigma(\boldsymbol{o}_i\boldsymbol{o}_j^\top)) - \sum_{p\in\mathcal{S}-\mathcal{N}_i} \log(\sigma(-\boldsymbol{o}_i\boldsymbol{o}_p^\top)) \,, \quad (8)$$

where $\boldsymbol{o}_i$, $\boldsymbol{o}_j$, $\boldsymbol{o}_p$ denote the embeddings of stock node i and its neighbor j in a pair, as well as one sampled negative node.

Combining the supervisory ranking signals and the self-supervised graph proximity loss, we reach the complete end-to-end loss function with a weighting coefficient η:

$$\mathcal{L} = \mathcal{L}_{\mathcal{R}} + \eta\mathcal{L}_{\mathcal{GP}} \,. \quad (9)$$

4 Experiments

4.1 Experimental Setup

Datasets. We examine *ALSP-TF* on three real-world datasets from *US* and *Japanese* Exchange markets. Table 2 shows the detailed statistics. The first dataset [Feng *et al.*, 2019b] comprises 1,026 shares from fairly volatile US S&P 500 and NASDAQ Composite Indexes; The second dataset [Feng *et al.*, 2019b] targets at 1,737 stocks from NYSE, which is by far the world's largest stock exchange w.r.t. market capitalization of listed companies and is relatively stable compared to NASDAQ; The third dataset [Li *et al.*, 2021] corresponds to the popular TOPIX-100 Index, which includes 95 stocks with the largest market capitalization in Tokyo stock exchange.

Implementation Details. Our model is implemented with PyTorch. We collect daily quote data of all stocks including normalized *opening-high-low-closing* prices (OHLC) and *trading volumes* from professional *Wind-Financial Terminal*[1]. For fair comparison, we follow [Sawhney *et al.*, 2021] and generate samples by moving a 16-day lookback window along trading days. We keep $\rho = 0.85, 0.85, 0.90$ for *NASDAQ*, *NYSE* and *TSE* respectively, and set the hop of graph convolutional operation to 2. For temporal modeling, we test stacking 1-5 *Lit* layers with varied skipping rates. The reported results utilize a 3-layers' hierarchy assigning $\delta_{[1:3]}$ to 1→2→3 and the number of attention heads H to 6 according to scores on validation set. The dimension of hidden feature space d_f is 16. The loss factors are set to $\alpha = 4$ and $\eta = 0.5$. We tune the model and ablation variants on a GeForce RTX 3090 GPU by Adam optimizer [Kingma and Ba, 2015] for 100 epochs, the learning rate is 1e-3 and batch size is 16.

	NASDAQ	NYSE	TSE
# Stocks	1,026	1,737	95
Train Period (Days)	01/13-12/15 (756)	01/13-12/15 (756)	11/15-08/18 (693)
Val Period (Days)	01/16-12/16 (252)	01/16-12/16 (252)	08/18-07/19 (231)
Test Period (Days)	01/17-12/17 (237)	01/17-12/17 (237)	07/19-08/20 (235)

Table 2: Statistics of datasets.

Metrics. Following previous studies [Feng *et al.*, 2019b; Sawhney *et al.*, 2021], we adopt a daily *buy-hold-sell* trading strategy to assess the profitability of *ALSP-TF* in terms of Sharpe ratio (SR) and cumulative investment return ratio (IRR). That is, the trader buys κ stocks with the highest expected revenues once the market is closed on day t, then sells off these shares on next day's close market. Formally, $\text{IRR}^t = \sum_{i\in\hat{\mathcal{S}}^t} \frac{p_i^{t+1} - p_i^t}{p_i^t}$, where $\hat{\mathcal{S}}^t$ denotes the stocks in portfolio on day t. SR$= \frac{E[R_p]-R_f}{std[R_p]}$ is a measure of risk-adjusted return describing the additional earnings an investor receives for per unit of increase in risk. We also compare the model's ranking ability adopting the widely used metric nDCG@κ. We report the mean results of five individual runs for $\kappa = 5$.

[1] https://www.wind.com.cn/en/wft.html

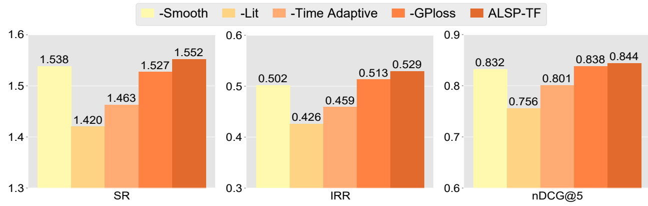
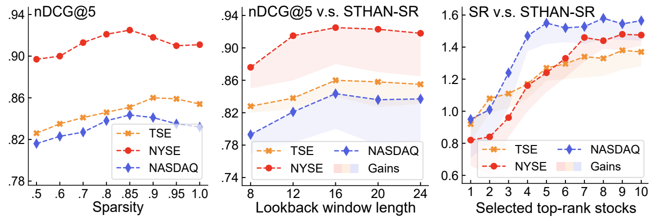


Figure 3: Ablation study over different components (*Graph smoothing operation*, *Local interaction (Lit)* & *Time-adaptive modulator* in Transformer, *Self-supervised graph proximity loss*) on NASDAQ.



(a) Graph structure (b) Lookback window (c) Trade stocks

Figure 4: Influence of hyper-parameters ρ, ΔT, and κ.

4.2 Overall Performance

We consider four categories of baselines for comparison. The results are shown in Table 1, from which we have several observations: 1) In general, RL and ranking approaches (e.g., *iRDPG*, *RSR*) perform better in investment returns than conventional price classification and regression methods (e.g., *HATR*, *SFM*), which justifies the effectiveness of *learning to rank* optimization toward stock selection. 2) Transformer-based encoders (*HMG-TF*, *RAT*) appreciably have better capability of modeling stock temporal dependencies, while it is hard for RNN-based models (e.g., *SFM*, *DA-RNN*, *SAE-LSTM*) to harvest fine-granular interactive information among discontinuous time steps. 3) Exploiting inter-stock relationships is demonstrated to be conducive to investment forecasting (e.g., *RSR*, *STHAN-SR*). This accentuates the collective synergistic effect of stock dynamics. Whereas, the requirement to predefine graph topologies based on domain knowledge raises the difficulty of generalizing these methods in new scenarios. 4) By revamping Transformer to hierarchically perform pattern-wise interactions being aware of time intervals and fusing self-supervision signals of stock proximity, our proposed *ALSP-TF* obtains the best results across all datasets. Specifically, it fetches an average relative performance gain of 8.57% and 18.54% in regard of risk adjusted returns and cumulative profits (t-test $p < 0.01$) over the best baselines. In addition, greater degrees of improvements are observed on *NASDAQ* and *NYSE* datasets than *TSE*, which may indicate the advantage of *ALSP-TF* in dealing with large candidate pools for stock portfolio selection.

4.3 In-depth Analysis

Next we conduct further analysis to learn the influence of various components and key hyperparameters in *ALSP-TF*.

Ablation Study. We investigate the effect of ablated variants from perspectives of both temporal and relational embedding. Due to space limitation, we depict the results on *NASDAQ* in Fig. 3, similar regularities can be observed on other datasets. As shown, different components jointly contribute to the performance. The main benefits stem from the *locality* and *time-sensitive* peculiarities inside Transformer blocks, which serve for extracting multi-grained dynamic patterns and endow different attention heads with adaptive ability to time intervals. The variant only retaining the temporal module can beat other Transformer baselines (*HMG-TF*, *RAT*) as well as state-of-the-art *STHAN-SR* that integrates LSTM and hypergraph convolution of pre-fixed corporate relations. Besides, introducing the graph-based smoothing and regularization further helps generate more stable and profitable stock selections. This inspires us that the data-driven relation embedding channel can furnish useful hidden dependency information when prior domain knowledge is unavailable.

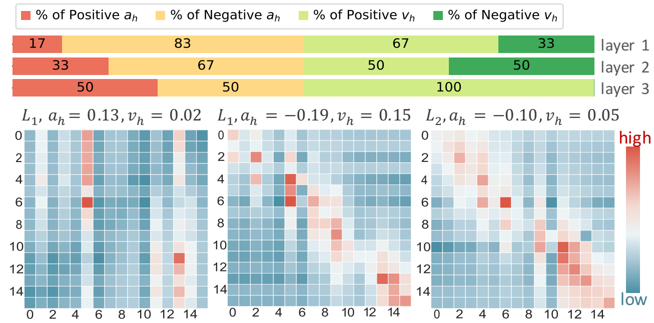


Figure 5: The time-adaptive coefficients and global self-attention weights learned by different attention heads on NASDAQ.

Parameter Sensitivity. We next look closer to building stock graphs with different sparsities. It can be seen from Fig. 4a that either free of the proximity guidance (i.e., *sparsity* $= 1.0$) or pushing too many edges will cause degradation of performance. This is intuitive because among the vast number of inter-stock connections, only a few are meaningful enough to significantly influence the dynamic of related stocks. Fig. 4b elaborates the impact of varied lookback lengths. We find that the advantage of *ALSP-TF* compared to *STHAN-SR* is greater in the case of longer input, revealing the merit of mining elaborate interactions among sequential patterns. We also explore the change of profitability regarding the number of selected top-rank stocks. Fig. 4c demonstrates *ALSP-TF*'s suitability to trading strategies accompanied by different risk appetites.

Interpretation of Time-adaptive Self-attention. We further interpret what is of importance the time-adaptive modulator in self-attention learns. Taking *NASDAQ* as example, the upper part of Fig. 5 shows the proportion of positive/negative a_h and v_h parameters tuned for the learnable rescale function $f(\cdot)$ over all attention heads. It is intriguing that the number of positive a_h gradually increases along with the stacking hierarchy. It means that lower-layer attention heads prefer to capture interactions around local contexts, while the heads at higher layers are responsible for modeling both short- and long-term dependencies. In addition, most values of v_h are positive, which may indicate that time information has relatively strong impact on weighing the pattern interactions. Moreover, the bottom half of Fig. 5 visualizes the attention heatmaps of a stock sequence produced by different heads. We find the attention scores are structurally sharper supported by the time-adaptive modulator. Specifically, the first heatmap concerns more on long interval dependencies, the matching between time-steps $5_{th}, 13_{th}$ and global contexts is prominently highlighted. In contrast, the latter two heatmaps put more emphasis on the information exchange inside local contexts. The interactive areas upon L_2 layer's embedding are broader, probably because self-attention acts on patterns of a higher-level granularity. These results show that our model can flexibly catch dynamic time factors and adjust attention on multi-grained sequential signals.

5 Conclusion

In this paper, we present *ALSP-TF*, a new temporal-relational embedding framework for stock selection. The temporal module performs hierarchical representation and interaction of stock dynamic patterns based on a modified Transformer encoder. Different from vanilla self-attention that is context and position agnostic, our model can adaptively capture short- and long-term pattern matching signals taking advantage of locality and time-aware peculiarities. For relational view, we propose a graph self-supervised regularization, which integrates collective synergies of stocks while relieving the dependence on prior domain knowledge. Through quantitative and qualitative analyses on three global market datasets, we probe the effectiveness of *ALSP-TF* and set forth its applicability in investment forecast and recommendation.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (No. 2021ZD0111202), the National Natural Science Foundation of China (No.62176005, 91846303), PKU-Haier "Smart Home" Qingdao Joint Innovation Lab, Project 2020BD002 supported by PKU-Baidu Fund, and Research Funds for NSD Construction, University of International Relations.

References

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

[Carta *et al.*, 2021] Salvatore Carta, Anselmo Ferreira, Alessandro Sebastian Podda, Diego Reforgiato Recupero, and Antonio Sanna. Multi-dqn: An ensemble of deep q-learning agents for stock market forecasting. *Expert Syst. Appl.*, 164:113820, 2021.

[Chen *et al.*, 2018] Yingmei Chen, Zhongyu Wei, and Xuanjing Huang. Incorporating corporation relationship via graph convolutional neural networks for stock price prediction. In *CIKM*, pages 1655–1658, 2018.

[Ding *et al.*, 2020] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo, and Jian Guo. Hierarchical multi-scale gaussian transformer for stock movement prediction. In *IJCAI*, pages 4640–4646, 2020.

[Dosovitskiy *et al.*, 2021] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[Feng *et al.*, 2019a] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. Enhancing stock movement prediction with adversarial training. In *IJCAI*, pages 5843–5849, 2019.

[Feng *et al.*, 2019b] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *TOIS*, 37(2):1–30, 2019.

[Gao *et al.*, 2019] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *CVPR*, pages 6639–6648, 2019.

[Jeong *et al.*, 2011] Young-Seon Jeong, Myong K Jeong, and Olufemi A Omitaomu. Weighted dynamic time warping for time series classification. *Pattern recognition*, 44(9):2231–2240, 2011.

[Kavitha *et al.*, 2013] G Kavitha, A Udhayakumar, and D Nagarajan. Stock market trend analysis using hidden markov models. *arXiv preprint arXiv:1311.4771*, 2013.

[Khaidem *et al.*, 2016] Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey. Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*, 2016.

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

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Lai *et al.*, 2017] Lin Lai, Chang Li, and Wen Long. A new method for stock price prediction based on mrfs and SSVM. In *ICDM*, pages 818–823, 2017.

[Li *et al.*, 2021] Wei Li, Ruihan Bao, Keiko Harimoto, Deli Chen, Jingjing Xu, and Qi Su. Modeling the stock relation with graph network for overnight stock movement prediction. In *IJCAI*, pages 4541–4547, 2021.

[Liu *et al.*, 2020] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *AAAI*, pages 2128–2135, 2020.

[Luo *et al.*, 2014] Yongen Luo, Jicheng Hu, Xiaofeng Wei, Dongjian Fang, and Heng Shao. Stock trends prediction based on hypergraph modeling clustering algorithm. In *PIC*, pages 27–31, 2014.

[Nayak *et al.*, 2015] Rudra Kalyan Nayak, Debahuti Mishra, and Amiya Kumar Rath. A naïve svm-knn based stock market trend reversal analysis for indian benchmark indices. *Applied Soft Computing*, 35:670–680, 2015.

[Oord *et al.*, 2016] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *ISCA*, page 125, 2016.

[Qin *et al.*, 2017] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *IJCAI*, pages 2627–2633, 2017.

[Sawhney *et al.*, 2021] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, Tyler Derr, and Rajiv Ratn Shah. Stock selection via spatiotemporal hypergraph attention network: A learning to rank approach. In *AAAI*, pages 497–504, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Velickovic *et al.*, 2017] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2017.

[Wang and Leu, 1996] Jung-Hua Wang and Jia-Yann Leu. Stock market trend prediction using arima-based neural networks. In *ICNN*, volume 4, pages 2160–2165, 1996.

[Wang *et al.*, 2020] Heyuan Wang, Tengjiao Wang, and Yi Li. Incorporating expert-based investment opinion signals in stock prediction: A deep learning framework. In *AAAI*, pages 971–978, 2020.

[Wang *et al.*, 2021] Heyuan Wang, Shun Li, Tengjiao Wang, and Jiayi Zheng. Hierarchical adaptive temporal-relational modeling for stock trend prediction. In *IJCAI*, pages 3691–3698, 2021.

[Wu *et al.*, 2021] Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. Da-transformer: Distance-aware transformer. In *NAACL-HLT*, pages 2059–2068, 2021.

[Xu *et al.*, 2020] Ke Xu, Yifan Zhang, Deheng Ye, Peilin Zhao, and Mingkui Tan. Relation-aware transformer for portfolio policy learning. In *IJCAI*, pages 4647–4653, 2020.

[Zhang *et al.*, 2017] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *KDD*, pages 2141–2149, 2017.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, pages 11106–11115, 2021.