

# Factor Investing with a Deep Multi-Factor Model

Zikai Wei<sup>1</sup> Bo Dai<sup>2</sup> Dahua Lin<sup>1,2</sup>

<sup>1</sup>The Chinese University of Hong Kong <sup>2</sup>Shanghai AI Laboratory  
wz018@ie.cuhk.edu.hk daibo@pjlab.org.cn dhlin@ie.cuhk.edu.hk

## Abstract

Modeling and characterizing multiple factors is perhaps the most important step in achieving excess returns over market benchmarks. Both academia and industry are striving to find new factors that have good explanatory power for future stock returns and good stability of their predictive power. In practice, factor investing is still largely based on linear multi-factor models, although many deep learning methods show promising results compared to traditional methods in stock trend prediction and portfolio risk management. However, the existing non-linear methods have two drawbacks: 1) there is a lack of interpretation of the newly discovered factors, 2) the financial insights behind the mining process are unclear, making practitioners reluctant to apply the existing methods to factor investing. To address these two shortcomings, we develop a novel deep multi-factor model that adopts industry neutralization and market neutralization modules with clear financial insights, which help us easily build a dynamic and multi-relational stock graph in a hierarchical structure to learn the graph representation of stock relationships at different levels, *e.g.*, industry level and universal level. Subsequently, graph attention modules are adopted to estimate a series of deep factors that maximize the cumulative factor returns. And a factor-attention module is developed to approximately compose the estimated deep factors from the input factors, as a way to interpret the deep factors explicitly. Extensive experiments on real-world stock market data demonstrate the effectiveness of our deep multi-factor model in the task of factor investing.

## 1 Introduction

Factor investing is one of the most popular solutions in portfolio analysis and investment management for building an active portfolio with a desirable investment objective, which often relies on the linear multi-factor model as its core module [5, 8, 2]. However, the assumption of a linear relationship between factor exposures and expected returns is quite restrictive, as it ignores interactions between different factors and does not always hold in reality, and an increase in factor exposures may not lead to an increase in expected returns. Moreover, the linear multi-factor model characterizes the relationships of different stocks implicitly via their common factors, whereas their relationships are in fact much more complicated.

On the other hand, with the recent rapid development of deep neural networks, they have demonstrated great potential in learning the relationships between different factors [13, 11]. However, although deep neural networks have been widely applied to stock market prediction [7, 1, 14, 15, 4], there is only a factor investing method [5] dated back to 1995 that adopts neural networks to find the nonlinear relationship between factors and expected returns. One major reason that prevents practitioners in factoring investing from using deep neural networks is that the financial insights behind the deep models are not clear and often considered as a “black box”, making the learned deep factors difficult to interpret.

To introduce the advance of deep neural networks to factor investing without losing the underlying financial insights, in this paper we propose a novel deep multi-factor framework that, through stock selection, finds deep factors via graph attention modules on top of a multi-relational stock graph. Specifically, the proposed framework consider stocks as nodes in a dynamic graph with two types of edges corresponding to stock relationships at different granularities. In this way, the graph can account for the the unchronological changes in listing or de-listing of stocks and the changes in the multi-level relationships among stocks. The first type of edges connects nodes in the same industry to form an industry graph, from which we can learn the industry influence within the same industry. The second type of edges connects all nodes into a universe graph, from which we can learn the influence of other stocks with the same universe. On top of this multi-relational stock graph, the proposed deep multi-factor model then adopts graph attention module summarize the industry and universe influences on other stocks. Based on these influences, we can distill out the unique information from the stock context at different granularities, i.e., the original context, the unique context after neutralizing the industry influence, and the unique context after neutralizing the industry and universe influence. Later, these contexts at different granularities can help the proposed model learn a set of deep factors to distinguish attractive and unattractive stocks and maximize the cumulative factor returns and information ratio. Finally, a factor-attention module is further introduced to find the relationship between deep factors and the original input factors such as style factor, macroeconomic factors, which indicates how deep factors are composed from the original ones. In this way, practitioners, such as fund managers, can thoroughly understand the market logic and economic insights behind each deep factor used in stock selection.

To validate the effectiveness of our approach, we conduct a comprehensive study on real-world data containing three broad-based stock indexes which covers more than 2800 stocks in the Chinese stock market from 2010 to 2022. In summary, the contributions of this work include 1) developing a deep multi-factor model based on financial insights, where each component has a concrete financial meaning, 2) defining stock relationships via a dynamic and multi-relational graph with hierarchical levels of edges, 3) interpreting how our deep factor emerges from the original stock information.

## 2 Related works

The multifactor model has received considerable attention from researchers and academics for several decades to determine the exact nature of the common factors that influence risk and return in various assets and markets [8], where existing work classify it in two categories *w.r.t* from different perspectives: linear vs nonlinear, and cross-sectional vs time-series. Comparing the cross-sectional and time-series models, it has been found that the cross-sectional factors can provide better explanation of the average stock returns than the model uses time-series factors [2]. This inspires us develop a deep learning architecture to learn cross-sectional factors.

With the rapid development of deep learning and its advantages in learning nonlinear relationships from big data in finance [4, 5], deep learning methods are applied to learn nonlinear relationships between style factors and expected returns of stocks [10, 9]. However, these methods do not consider the relationships between stocks in their models, and the financial insights behind their building blocks are not clear. To incorporate the multi-relational nature of stocks into our model, we turn our attention to graph neural networks [12], which have recently done great work in learning cross-stock relationships over a knowledge graph to solve a variety of finance problems [13], such as stock movement prediction [14, 15, 7, 1] event prediction, and risk management [6]. In our model, we use two graph attention networks to learn the influence of other stocks in the same industry and the influence of other stocks in the same universe.

## 3 Method

In this work, we develop a deep learning framework to find the cross-sectional factors that can consistently explain the average stock returns well, making good use of the relationships between stocks and the insights from finance.

### 3.1 Graph Construction

**Definition 1. Stock Graph.** The stock graph is defined as a directed graph as in [15],  $\mathcal{G} = \langle \mathcal{S}, \mathcal{R}, \mathcal{M} \rangle$ , where  $\mathcal{S}$  denotes the set of constituents of a broad-based stock index and  $\mathcal{R}$  is the set of relations between two stocks.  $\mathcal{M}$  is the set of adjacent matrices. For an adjacent matrix  $\mathbf{M}^r \in \mathcal{M}$  of relation  $r \in \mathcal{R}$ , where  $\mathbf{M}^r \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{R}|}$ ,  $\mathbf{M}_{ij}^r = 1$  means that there is a relation  $r$  from stock  $s_j$  to stock  $s_i$  and  $\mathbf{M}_{ij}^r = 0$  means that there is no such relation. However, the edges of the directed graph have an orientation [13], so this stock graph does not always hold in reality: stocks in the same industry always influence each other. In our work, we define the stock graph as a *dynamic* and *multi-relational* graph. The nodes and edges could be dynamic because the listed stocks change over time and a stock can be classified into a different industry, which leads to a change in the nodes and edges. We use a time-varying adjacent matrix  $\mathbf{M}_t^r$  to represent this dynamic property. We distill the stock context into different levels: Industry level and Universe level, where the edges at the different levels represent different types of relationships. To learn the graph representation of stock relationships at different levels, we define the multi-related graph in a two-level hierarchy: an undirected graph describing the intra-industry relationship and an undirected graph interpreting the universe relationship.

**Definition 2. Industry Graph and Universe Graph.** We define the industry graph as a stock graph  $\mathcal{G}_s$  where stocks have relations to each other if they belong to the same industry. Similarly, the universe graph is defined as stock graph  $\mathcal{G}_g$  that all constituents of a broad-based stock index are related. We use a time-variant graph for both cases of industry and universe, since the constituents in a industry or a index change over time.

### 3.2 Architecture

**Stock Context Encoder.** We define  $n$  as the number of constituents inside a broad-based stock index, and  $m$  as the number of types of stock context that covers fundamental, trading information and analysts' estimates.

$$\mathbf{C}_t = \text{MLP}(\text{BatchNorm}(\mathbf{F}_t)), \quad (1)$$

where  $\mathbf{F}_t \in \mathbb{R}^{n \times m}$  is a matrix represents  $m$  original factors *w.r.t*  $n$  stocks, and  $\mathbf{C}_t \in \mathbb{R}^{n \times m_1}$  is the stock context matrix with  $m_1$  hidden features extracted from the original factors. The financial insight of the batch normalization [3] used here is the similar to the *z-score normalization* generally applied in data preprocessing for factor developing.

**Learning the Industry Influence.** Fundamentally, stocks are comparable inside the same industry, which makes the inner-industry relations important in industry-based analysis. Stocks in the same industry have mutual influence on each other beyond their common properties. To learn this industry influence, we apply Graph Attention Network (GAT) [12] over the pre-defined industry graph, where the time-variant industry mask,  $\mathbf{M}_t \in \mathbb{R}^{n \times n}$ , is symmetric and defined as

$$\mathbf{M}_{ij}^t = \begin{cases} 1, & \text{if stock } s_i \text{ and stocks } s_j \text{ are in the same industry} \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where  $t$  indicates the time stamp and  $1 \leq i, j \leq n$ . Then, we have the industry influence as

$$\mathbf{H}_I^t = \text{GAT}(\mathbf{M}_t \mathbf{C}_t). \quad (3)$$

This industry influence is useful for neutralizing the industry information.

**Definition 3. Industry Neutrality.** We define the industry neutrality  $\bar{\mathbf{C}}_I^t$  as the stock context  $\mathbf{C}_t$  subtracted from the industry influence  $\mathbf{H}_I^t$ , i.e.,

$$\bar{\mathbf{C}}_I^t = \mathbf{C}_t - \mathbf{H}_I^t. \quad (4)$$

Industry neutrality interprets the different context of a stock compared to other stocks in the same industry, which makes stocks comparable across different industries.

**Learning the Universe Influence.** Similar to learning the industry influence, we adopt GAT to capture the mutual influence of any two stocks over the universe graph:

$$\mathbf{H}_U^t = \text{GAT}(\bar{\mathbf{C}}_I^t), \quad (5)$$

where the universe is defined as a set of all constituents in a board-based stock index. The universe influence should learn from the stock context after eliminating industry influence, which can help to learn the cross-stock influence without industry effects.

**Definition 4. Universe Neutrality.** We define the universe neutrality  $\bar{\mathbf{C}}_U^t$  as the industry-neutralized stock context  $\mathbf{C}_t$  excluding the universe influence  $\mathbf{H}_U^t$ ,

$$\bar{\mathbf{C}}_U^t = \bar{\mathbf{C}}_I^t - \mathbf{H}_U^t. \quad (6)$$

The financial insight behind universal neutrality is that it contains the stock independent context with eliminating not only the effect of the industry on this stock but also the effect of the universe context.

**Learning Deep Factor from Hierarchical Contexts.** Formally, we learn the deep factor  $\mathbf{f}_t \in \mathbb{R}^n$  from different granularity: stock original context  $\mathbf{C}^t$ , industry neutralized context  $\bar{\mathbf{C}}_I^t$ , and universe-neutralized context  $\bar{\mathbf{C}}_U^t$ :

$$\mathbf{f}_t = \text{LeakyReLU} \left( \mathbf{W}^T \left( \mathbf{C}^t \parallel \bar{\mathbf{C}}_I^t \parallel \bar{\mathbf{C}}_U^t \right) \right), \quad (7)$$

where  $\mathbf{W}^T$  is a single-layer feed-forward neural network and  $\parallel$  represents concatenation.

**Multi-Head Learning.** We design  $K$  output heads for learning ultimate factors corresponding to multiple forward periods be aware of practitioners, *i.e.*  $k$ -forward trading days, where  $k = 3, 5, 10, 15, 20$ . The Chinese mutual fund regulation does not allow fund managers sell the securities hold less than two trading dates. The minimum forward horizon, therefore, is set to three trading days. The multiple of five indicates the number of weeks that a security is hold. Based on Eq. 7, the deep factor learned *w.r.t* different forward horizon is defined as

$$\mathbf{f}_k^t = \text{LeakyReLU} \left( \mathbf{W}_k^T \left( \mathbf{C}^t \parallel \bar{\mathbf{C}}_I^t \parallel \bar{\mathbf{C}}_U^t \right) \right), \quad (8)$$

where  $\mathbf{f}_k^t \in \mathbb{R}^n$  is the deep factor targets for explain the future information on the  $k$ -forward trading day and  $\mathbf{W}_k^T$  is a single-layer feed-forward neural network corresponding to  $k$ -forward trading day.

**Interpretation of Deep Factor  $\mathbf{f}^t$  via Factor Attention Module.** We introduce factor attention module to investigate the relationship between the deep factor and the original input factors or features. We employ attention mechanism [11] to attend our deep factor to learn the token importance of the original features. The normalized attention weight through a softmax function illustrates “how much information influx in the deep factor from an original feature”.

$$\mathbf{U}_k^t = \text{LeakyReLU} \left( \mathbf{W}_{a,k}^T \mathbf{F}_t \right), \quad (9)$$

$$\mathbf{A}_k^t = \text{softmax} \left( \mathbf{U}_k^t \right), \quad (10)$$

$$\bar{\mathbf{a}}_k^t = \frac{1}{n} \sum_{i \in n} \mathbf{a}_{ik}^t \quad (11)$$

where  $\mathbf{W}_{a,k}^T \in \mathbb{R}^{n \times n}$  is a single-layer feed-forward neural network corresponding to the  $k$ -forward trading day.  $\mathbf{A}_k^t \in \mathbb{R}^{n \times m}$  is the attention weight matrix for the deep factor  $\mathbf{f}_k^t$  and  $\mathbf{a}_{ik}^t$  is  $i$ -th column of  $\mathbf{A}_k^t$ . Here,  $\mathbf{a}_k^t$  is its corresponding attention weight vector of  $m$ . We define *the estimated deep factor via factor attention*,  $\hat{\mathbf{f}}_k^t \in \mathbb{R}^m$ , as

$$\hat{\mathbf{f}}_k^t = \mathbf{F}_t^T \bar{\mathbf{a}}_k^t, \quad (12)$$

where  $\hat{\mathbf{f}}_k^t$  is the “attention estimate” of the deep factor  $\mathbf{f}_k^t$ . When  $\hat{\mathbf{f}}_k^t$  is very close to  $\mathbf{f}_k^t$ , the attention weight  $\bar{\mathbf{a}}_k^t$  can interpret the portion of quantity comes from the original input factors.

### 3.3 Loss Design

Now we ponder the design of loss function to fulfill the learning objectives of deep factor.

**Maximizing Factor Stability.** One of our goals is to find a factor that has good stability of its predictive power, usually measured by the information ratio of information coefficient (ICIR) [7]. We use  $c_k$  to denote the ICIR of the deep factor  $\mathbf{f}_k^t$ , where  $c_k$  measures the stability of the predictive power of the deep factor  $\mathbf{f}_k^t$  for the future return over the next  $k$  trading days  $\mathbf{r}_{t+k}$ . Thus, since higher

ICIR indicates higher stability, we look for a stable factor that maximizes ICIR. For more details on ICIR and information ratio, see our supplementary material.

**Maximizing Factor Return.** Factor returns are the cross-sectional regression coefficients [6] that indicate the return attributable to a particular common factor:  $\hat{r}_{t+k} = b_k^t \mathbf{f}_k^t$ , where  $b_k^t \in \mathbb{R}$  is the factor return at time  $t$ . We introduce a local optimizer to learn the parameter  $b_k^t$ , which is independent of the global optimizer. The deep factor is expected to have a higher cumulative factor return,  $\sum_{t \in \mathcal{T}} b_k^t$ .

**Improving ‘‘Attention Estimate’’.** We use  $L_2$ -norm to evaluate the deviation between the deep factor  $\mathbf{f}_k^t$  and its corresponding attention estimate  $\hat{\mathbf{f}}_k^t$ , which can be calculated as  $d_k^t = \|\mathbf{f}_k^t - \hat{\mathbf{f}}_k^t\|_2$ .

The overall loss function is

$$\mathcal{L} = \frac{1}{|\mathcal{K}||\mathcal{T}|} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} d_k^t - b_k^t - c_k \quad (13)$$

where  $\mathcal{T}$  is the set of dates in the training period,  $\mathcal{K}$  is the set of forward horizons.

### 3.4 Optimization

**Cross-sectional optimizer.** The cross-sectional optimizer is a local optimizer designed to learn a cross-sectional regression coefficient (factor returns) at each time  $t \in \mathcal{T}$ , by minimizing the mean squared errors of the predicted returns and the target returns. This cross-sectional optimizer is independent of the global optimizer we use to train our deep multifactor model.

**Global optimizer.** The global optimizer is designed to update the parameters of our deep multifactor model. It is set to simultaneously optimize the parameters in the stock context encoder, the industry and universe influences learning modules, the multi-head module, and the factor attention module.

## 4 Experiment

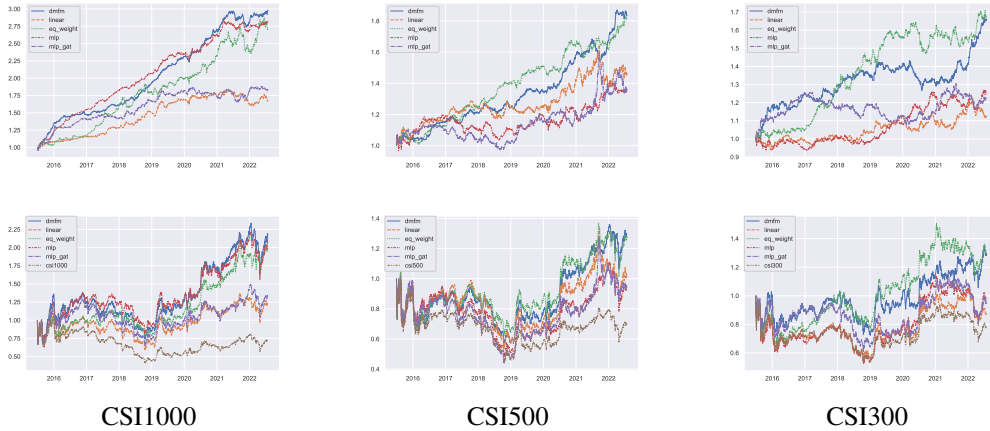


Figure 1: Comparison of performance between our deep multifactor models and other baseline methods in different stock universes. The figures in the top rows show the curve of cumulative excess returns over each benchmark, while the second row shows the net values achieved by each method.

**Stock Pool.** In order to capture the most representative stocks of China’s stock market, we select all constituents of the three major broad-based indices, namely CSI 300, CSI 500 and CSI 1000, which include 1800 stocks with the highest rank in terms of market capitalization and liquidity at the same time. Taking into account the changes caused by adjustments in the constituents, they cover more than 2800 stocks in the period from 2010 to mid-2022. By September 26, 2022, they cover 80.56% of the total market capitalization of all stocks in the Chinese market.

**Industry Classification Standard.** For industry graph, we choose the CITIC Securities Industry Classification Standard Level 1 (CICSL1) as a guideline to define the industry relationships among

Table 1: Experimental results on different universes over the period 2015 to mid-2022 (**best/2nd best**)

Method	CSI 1000				CSI 500				CSI 300			
	$\alpha(\%)$	ICIR	IR	SR	$\alpha(\%)$	ICIR	IR	SR	$\alpha(\%)$	ICIR	IR	SR
Linear	7.79	1.08	1.35	0.19	5.62	0.89	0.77	0.04	1.73	0.98	0.32	-0.09
EW	14.65	<b>1.48</b>	1.92	0.47	<u>8.91</u>	<u>1.13</u>	<u>1.86</u>	<u>0.18</u>	<u>7.57</u>	0.90	<b>1.20</b>	0.19
MLP	15.16	<u>1.44</u>	<b>3.53</b>	<u>0.49</u>	4.48	1.03	0.67	-0.01	3.56	<b>1.27</b>	0.65	0.00
MGAT	9.24	1.02	1.63	0.24	4.96	0.83	0.62	0.01	3.17	0.48	0.56	-0.02
Ours	<b>16.36</b>	1.25	<u>3.46</u>	<b>0.50</b>	<b>9.03</b>	<b>1.38</b>	<b>1.89</b>	<b>0.18</b>	<b>7.85</b>	<u>1.13</u>	<u>1.09</u>	<b>0.20</b>

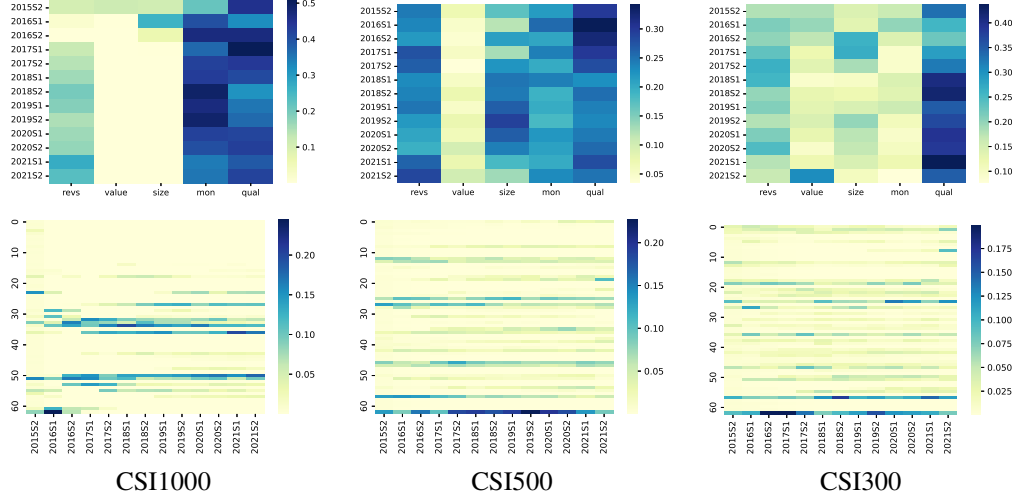
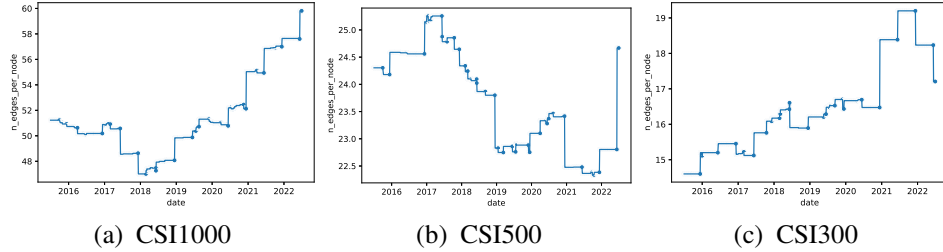
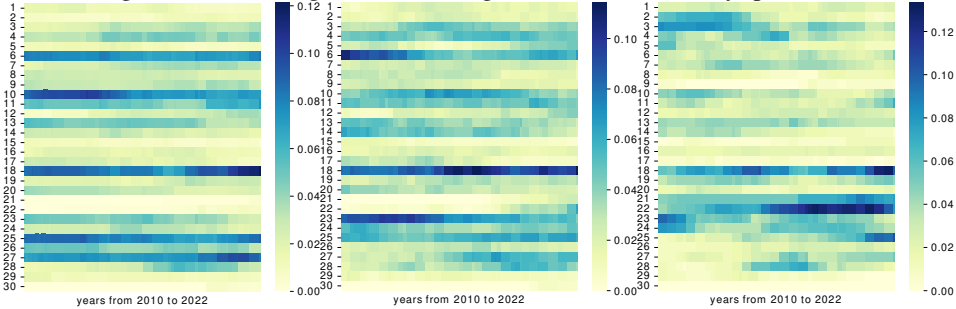


Figure 2: The heat map shows the attention weights of the deep factors across each of the original factors: the first row shows the average attention weights for each group of factors, while the second row shows the average attention weights for each of the sixty-three style factors.



(a) CSI1000 (b) CSI500 (c) CSI300

Figure 3: The average number of edges in the same industry (per stock)



(a) CSI1000 (b) CSI500 (c) CSI300

Figure 4: The heat map for the industry proportion in various broad-based indices from 2010 to mid-2022, with numbers from 1 to 30 representing various industries.

stocks. The CICS11 is widely used by institutional investors investing in Chinese market and was developed based on two well-known global standards: Global Industry Classification Standard (GICS)

and Industry Classification Benchmark (ICB). Figure 4 shows the heat map of industry portion in CSI 300, CSI 500 and CSI 1000, according to CICS1 standard.

**Dataset Construction.** Our original inputs are developed from fundamental and market data collected from publicly available data from tushare.pro. The institutional investor can also obtain the data from Refinitiv’s database or WIND. We construct our dataset on a daily basis and divide the original factors into five groups derived from the Barra Global Equity Model: reversal, value, size, momentum, quality[8]. Our dataset includes the factors that have been found to be informative in Chinese stock market [7]. Based on the factors mentioned in [8, 7], we create other derivatives of these factors by using twelve trailing months, quarterly changes, annual changes, and factor values over yields. Finally, we create 63 factors including the base factors and their derivatives. All factors are calculated using point-in-time data, which means that there is no information leakage. For example, a company issues a financial report at time  $t_1$  which is publicly available or accessible in a financial database at time  $t_2$ . Later, it make a restatement of that report at time  $t_3$ , while the information is available to investors at  $t_4$ . Here,  $t_1 < t_2 < t_3 < t_4$ . It will never use the modified information before time  $t_4$ . We follow the temporal order to divide the data into 14 data groups: Each group contains a training set, a validation set, and a test set. The training set is extended every six months. The validation set consists of the data over the following six-month period. However, we exclude the last month data from the validation set to avoid information leakage. The reason is that maximizing the information ratio involves using future information, *i.e.*, future returns over the next  $k$  trading days. Therefore, we truncate the information that might overlap with the information in the test set. Thus, while the test set uses data from the next six-month period, there is no overlap between the validation and test sets. For instance, the three datasets in the first group include training (2010/1/1~2015/6/30), validation (2015/7/1~2015/11/30), test (2016/1/1~2016/6/30), and the last group includes training (2010/1/1~2021/6/30), validation (2021/7/1~2021/11/30), test (2022/1/1~2022/6/30).

**Portfolio Construction.** Our goal is to select stocks from a given universe that are considered more attractive than others and build a portfolio that can outperform the benchmark of that universe. During the testing period, we select the best model with the lowest validation loss and use it to produce the deep factors. We then rank the stocks in ascending order according to their factor scores. We assume that the stocks with higher values are more attractive, which means that they could be responsible for future excess returns over the market benchmarks, *i.e.*, broad-based indices. Therefore, we construct an equally weighted portfolio consisting of the 10% most attractive stocks. The portfolio is rebalanced monthly, with transaction costs for buying and selling set at 4‰ of the transaction amount.

**Baselines and Model Implementation.** We compare our proposed method with the following baselines: 1) **Linear** is a linear regression model where the model learns a weight for each style factor. 2) **EW** is a linear model where each descriptor (original factor) is equally weighted. EW is an important baseline that shows a result of combining factors that has nothing to do with data mining or in-sample optimization, and any deviations from equal weighting shall be determined by economic rationality [8]. 3) **MLP** is a nonlinear model that learns factors from the stock context based on neural networks and consists of a context encoder and a feature decoder. 4) **MLP& GAT** is a model based on a context encoder, a GAT, which learns the interactions between stocks in a universe graph, and a decoder, which learns a deep factor from the stock context and the universal relationships of the stocks. See supplementary materials for more details in model implementation.

**The Performance of Portfolios Constructed by Different Methods.** In Figure 1 we show the performance curve of the portfolio created with each method, and in Table 1 we show the evaluation metrics of each method, *i.e.*, the active return denoted by  $\alpha(\%)$ , ICIR, information ratio (IR), and Sharpe ratio (SR) (see our supplementary material for more details.). These results show that our methods generally outperform others in the long run. Our deep multifactor model (DMFM) can always achieve higher active returns and Sharpe ratio than linear, EW, MLP and MLP& GAT (MGAT) models. Although SR of the three benchmarks is -0.11, -0.20, and -0.18, respectively, our model always has a positive SR. Since CSI1000, CSI500 and CSI300 represent the small-cap, mid-cap and large-cap of Chinese stock markets, the better performance of our model in these three universes shows its generalizability at different levels of market capitalization.

**Interpretation of Deep Factor.** To make our deep factor more transparent and to clarify its insight from the financial world, we visualize the attention weight of original style factors by groups or by each of them. We take the deep factor associated with the forward twentieth trading day ( $k = 20$ ) as

an example. The average attention weights over each test period are shown in Figure 2. In this case, the style factors belonging to the quality group receive more attention weights than other groups among all constituents of the three broad-based indexes. The heat maps for the different trading days ( $k = 3, 5, 10, 15$ ) can be found in the supplementary materials.

**Edge Analysis for Dynamic Graph.** This dynamic property of a stock universe can manifest itself in changes in the relationships among stocks (see Figure 3) and in changes in its constituents (see Figure 4). A change in index components or industry classification standard (ICS) contributes to a change in the industry share for each sector in a stock universe. Since the ICS does not change too often, the change in the industry proportion is mainly caused by the change in index constituents. Therefore, Figure 4 can show most of the changes in the nodes of the stock graph. To illustrate the changes in the stock relationship, in Figure 3 we show the curve of average number of edges a stock has on each trading day over our industry graph in different universes (CSI1000, CSI500 and CSI300).

## 5 Conclusion

In this paper, we propose a deep cross-sectional multifactor model that incorporates financial insights and stock relationships on a dynamic and multi-relational graph in the task of factor investing. Our work makes three contributions: First, we are the first to incorporate an attention-based interpretation module into a deep learning framework that more directly shows how the original style factors explain our deep factor. Second, we define a stock graph with a hierarchical level of edges to represent the dynamic and multi-relational properties of stock interactions. Third, we build our model based on certain economic rationales and each module within our model represents a concrete financial meaning. In the future, we plan to apply our method to develop enhanced index funds (EIFs) with additional modules.

## References

- [1] Chi Chen, Li Zhao, Jiang Bian, Chunxiao Xing, and Tie-Yan Liu. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2376–2384, 2019.
- [2] Eugene F Fama and Kenneth R French. Comparing cross-section and time-series factor models. *The Review of Financial Studies*, 33(5):1891–1926, 2020.
- [3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [4] Weiwei Jiang. Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, 184:115537, 2021.
- [5] Asriel Levin. Stock selection via nonlinear multi-factor models. *Advances in Neural Information Processing Systems*, 8, 1995.
- [6] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. Deep risk model: a deep learning solution for mining latent risk factors to improve covariance matrix estimation. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–8, 2021.
- [7] Hengxu Lin, Dong Zhou, Weiqing Liu, and Jiang Bian. Learning multiple stock trading patterns with temporal routing adaptor and optimal transport. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1017–1026, 2021.
- [8] Dimitris Melas. Best practices in factor research and factor models. *MSCI Research Insight*, 2018.
- [9] Kei Nakagawa, Tomoki Ito, Masaya Abe, and Kiyoshi Izumi. Deep recurrent factor model: interpretable non-linear and time-varying multi-factor model. *arXiv preprint arXiv:1901.11493*, 2019.
- [10] Kei Nakagawa, Takumi Uchida, and Tomohisa Aoshima. Deep factor model. In *ECML PKDD 2018 Workshops*, pages 37–50. Springer, 2018.
- [11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.



- [12] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.
- [13] Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications. *arXiv preprint arXiv:2111.15367*, 2021.
- [14] Wentao Xu, Weiqing Liu, Lewen Wang, Yingce Xia, Jiang Bian, Jian Yin, and Tie-Yan Liu. Hist: A graph-based framework for stock trend forecasting via mining concept-oriented shared information. *arXiv preprint arXiv:2110.13716*, 2021.
- [15] Wentao Xu, Weiqing Liu, Chang Xu, Jiang Bian, Jian Yin, and Tie-Yan Liu. Rest: Relational event-driven stock trend forecasting. In *Proceedings of the Web Conference 2021*, pages 1–10, 2021.