

HireVAE: An Online and Adaptive Factor Model Based on Hierarchical and Regime-Switch VAE

Zikai Wei¹, Anyi Rao², Bo Dai³ and Dahua Lin^{1,3}

¹The Chinese University of Hong Kong

²Stanford University

³Shanghai AI Laboratory

{wz018, dhlin}@ie.cuhk.edu.hk, anyirao@stanford.edu, daibo@pjlab.org.cn

Abstract

Factor model is a fundamental investment tool in quantitative investment, which can be empowered by deep learning to become more flexible and efficient in practical complicated investing situations. However, it is still an open question to build a factor model that can conduct stock prediction in an online and adaptive setting, where the model can adapt itself to match the current market regime identified based on only point-in-time market information. To tackle this problem, we propose the first deep learning based online and adaptive factor model, **HireVAE**, at the core of which is a **hierarchical** latent space that embeds the underlying relationship between the market situation and stock-wise latent factors, so that HireVAE can effectively estimate useful latent factors given only historical market information and subsequently predict accurate stock returns. Across four commonly used real stock market benchmarks, the proposed HireVAE demonstrate superior performance in terms of active returns over previous methods, verifying the potential of such online and adaptive factor model.

1 Introduction

Originating in asset pricing, factor model is the foundation of factor-based investing, which simplifies the high-dimensional characteristics in quantitative investment via an experimental way to model how factors drive returns and risks. This can significantly help institutional investors construct diverse and customized portfolios. But, a better-developed capital market brings out more challenges in achieving excess returns in the market using traditional methods due to more similar investing actions and overlapping trading positions. Although the recent nonlinear data-driven methods [Duan *et al.*, 2022; Wei *et al.*, 2022; Gu *et al.*, 2021] enrich the conventional tool library built largely from linear methods [Ng *et al.*, 1992; Fama and French, 2020], they still face serious issues in handling large, abrupt, and sustained market changes and some factors that worked well in the previous market situation become less effective in the new situation. Therefore, it is of

great importance to model the current market regime and develop an adaptive factor model associated with different market situations.

Both practitioners and academics make many attempts to model the regime change in markets using statistical learning [Guidolin and Timmermann, 2007], machine learning [Uysal and Mulvey, 2021], or hybrid methods [Akiyamen *et al.*, 2020] from different perspectives of the market. Existing data-driven approaches view regime switching as a clustering problem and represent each regime as a cluster, subdividing the market regime based on the information absorbed from the entire training data. However, this setting is not suitable for learning a regime-switching model intended for later use in investment practice. Practice requires an online learning setting that all data used for learning should be point-in-time data, meaning that there is no future information leakage at any timestamp in the training data set.

In this paper, we thus propose a novel end-to-end neural factor model, referred to as HireVAE, that offers an online and adaptive regime-switching capability, targeting two critical questions: **Q1**: How to keep the regime cluster labels changing consistently and centers of cluster shifting smoothly during training? **Q2**: How to adaptive learn regime-oriented factor model under different regimes?

To tackle **Q1**, we develop a module for identifying regimes, in which it learns a set of dynamic distributions associated with different regime clusters. To maintain this regime identifier as a stable clustering module, we design a linear stabilization algorithm to supervise the identifier learn a consistent clustering that smoothly alternates with a market latent variable. Specifically, a market encoder, which is learned in a prior-posterior learning framework, will encode the market information into a market latent variable. This framework can help the market encoder learn the hidden logic between the market information and the average future stock returns in the whole market, which will serve as prior knowledge for the market encoder in predicting future stock returns given only current market information. The market latent variable plays two roles in HireVAE: 1) at first it is used to estimate the means and variances of K Gaussian distributions, representing K regimes. 2) Subsequently, it will also be used to predict K regime indicators measuring the likelihood of each regime being the most suitable one, from which the one with the largest likelihood can thus be identified. It's worth not-

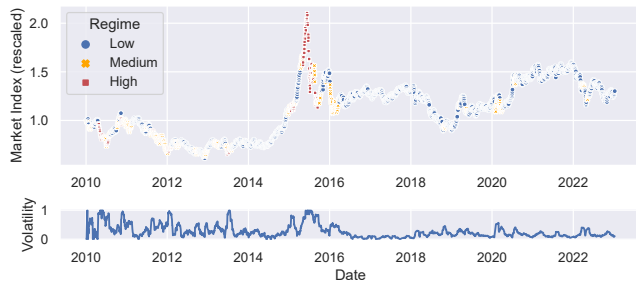


Figure 1: Identifying market regimes can help make better investment decisions, and there is an example that identifies regime changes based on volatility, where volatility acts as an experience-and-knowledge based indicator that is widely used in practice.

ing that we will sort K estimated regimes by sorting their means in descending order, as pointed out by [Botte and Bao, 2021], where returns and risk-adjusted measures are linearly correlated across market regimes, so that we can partition the whole market situation into market regions following a consecutive order. With the help of the market latent variable, we can then adaptively select the most suitable regime given current market information. However, one question remains: How to smoothly update the regime centers and make a consistent regime assignment? To ensure consistency of cluster assignment during training, the cluster ID is always reordered in the descending order. In this way, the market latent variable can always be associated with a consistent regime cluster and the additional burden of checking for a “wrong assignment” is eliminated. Moreover, to smoothly update the regime clusters, we use a dynamically weighted moving average algorithm to update the means and standard deviations of the clusters, giving more weight to the historical clusters. To cope with **Q2**, we develop an encoder with a hierarchical latent space of market and stock latent variables and a set of regime-based stock decoders, each of which recovers or predicts stock returns based on regime, market, and stock latent factors. With this linear stabilization algorithm for learning market regimes, the end-to-end training framework can adaptively learn different market conditions as well as the corresponding regime-specific factor model for stock prediction. In a later section, we will show the advantages of this algorithm for regime switching with adaptive learning compared to rule-based regime switching, e.g., categorizing market regimes with respect to different volatility levels.

Example 1 (Determining regimes helps better decision making). *Figure 1 shows the structural changes of the CSI All Share Index, composed of all stocks in China stock market, in terms of volatility level. When volatility is high, there is usually a change from a bullish market to a bearish market. In contrast, medium volatility usually occurs during the period when the market is changing trend at a more steady pace. The market begins a long bull run when the low volatility prevails most of the time.*

The contribution of our paper are as follows: 1) To the best of our knowledge, we are the first to propose regime switching with online regime learning in an end-to-end training

framework. 2) We provide an adaptive learning framework for training a regime switch factor model with a hierarchical latent space that incorporates global observation of stock market and stock-oriented latent factors. The decoders are also online trained based on different dynamic market conditions, each of which takes advantage of regime, market, and stock information in the following downstream tasks. 3) We develop a linear stabilization algorithm that helps the regime identification process to learn a more consistent and smoother clustering deep learning technique.

2 Related Work

Factor Models. The factor model are traditionally regarded as a pricing model in academia [Fama and French, 1992; Fama and French, 2020], while practitioners use its anomalies as factors to capture excess returns from mispricing [Levin, 1995]. In recent years, stock forecasting studies based on machine learning methods are emerging. Machine learning based stock prediction methods are closely related to factor models and can be categorized as factor mining and factor composition. The former learns a new logical factor from new data sources or perspective, while the latter finds a better way to compose the existing factors, which can be considered as a machine learning based factor model. In terms of factor mining, existing works on stock prediction study trading patterns [Wang *et al.*, 2022; Ding *et al.*, 2020; Zhang *et al.*, 2017], investment reviews [Wang *et al.*, 2021b], overnight patterns [Li *et al.*, 2021], and the temporal relationship of stocks [Wang *et al.*, 2021a]. As for factor composition as nonlinear factor models, existing works learn optimal latent factors via a variational autoencoder (Factor-VAE) [Duan *et al.*, 2022], a deep multifactor model (DMFM), which builds on hierarchical stock graphs [Wei *et al.*, 2022], and a deep risk factor model that fuses a set of uncorrelated risk factors from the original style factors [Lin *et al.*, 2021]. Unlike existing methods, our model aims to learn an adaptive factor model that can recognize the current market regime and use a learnable and more appropriate factor composition for that market regime, which can enable a market-driven investment decision.

Regime Switching [Hamilton, 1989] provide a statistical modeling framework to address the problem of regime switching in various circumstances, such as regime switching in interest rates [Ang and Bekaert, 2002], asset allocation [Guidolin and Timmermann, 2007], and financial markets [Zhu, 2022]. In the cryptocurrency market, [Zhu, 2022] shows that regime switching methods perform better than methods without regime switching. Machine learning based methods open another main direction in modeling regime switching, such as Hidden Markov Model [Wang *et al.*, 2020] and Gaussian Mixed Models [Botte and Bao, 2021]. As a hybrid method, [Akiyamen *et al.*, 2020] applies principal component analysis and k -means clustering to identify regimes in financial markets. Deep learning based regime switching models of energy commodity prices [Mari and Mari, 2022]. Unlike the existing regime switching method, in investment practice, we not only need to use an online learning algorithm to identify market regimes, but also need to

learn an adaptive factor model based on the determination of the market regime.

Variational Autoencoder. Variational Autoencoder (VAE) [Kingma and Welling, 2013], one of the major families of deep generative models, provides a prior-posterior framework that encodes the observation as prior knowledge in a latent space, and later [Vahdat and Kautz, 2020] introduce a hierarchical pipeline into VAE. [Chung *et al.*, 2015] is the first to extend VAE to sequential modeling. Then, the variational methods are extended to neural machine translation [Su *et al.*, 2018] and trajectory prediction. In the area of financial applications, VAE includes topics such as stochastic volatility models [Luo *et al.*, 2018] and factor models. The latter VAE family includes applications in factor mining based on social media in the context of stocks [Xu and Cohen, 2018], applications in factor composition where [Duan *et al.*, 2022] aims to learn representative latent factors from existing technical factors, and [Gu *et al.*, 2021] aims to learn a pricing model.

3 Preliminaries

In this section, we first define the stock prediction problem and then introduce the problem setting from the perspective of learning a data-driven method.

Definition 1 (Stock prediction). *In general, stock prediction problem can be defined as learning a mapping $f(\cdot; \Theta)$ from the past information set $\mathcal{F}_{<t}$ to future stock return $\mathbf{y}_{t+\Delta_t}$, where Δ_t is the prediction length, $\hat{\mathbf{y}}_{t+\Delta_t} = \mathbb{E}(\mathbf{y}_{t+\Delta_t} | \mathcal{F}_{<t})$ is its corresponding expected future return, and Θ is the model parameter of data-driven method $f(\cdot)$.*

In our setting, we have a set of samples labeled with time stamps sorted in chronological order: $\mathcal{D} = \{\mathcal{D}_{t_0}, \dots, \mathcal{D}_{t_n}\}$, where n is the total number of trading days covered by the dataset. For a single data sample $\mathcal{D}_t = \{\mathbf{X}_t, \mathbf{M}_t\}$, $\mathbf{X}_t \in \mathbb{R}^{T \times N \times C}$ and $\mathbf{M}_t \in \mathbb{R}^{T \times C_m}$ are sequential features of stocks and the market, where T , N and $C_{(\cdot)}$ represent the length of the historical sequence, the number of stocks and the number of features. Moreover, the overall pattern of the market (global observation) consists of information from d modalities, based on which it can be segmented as $\mathbf{M}_t = [\mathbf{M}_t^1, \dots, \mathbf{M}_t^d]$. The above are general input dataset.

Besides, the corresponding target set of \mathcal{D} is $\mathcal{Y} = \{\mathcal{Y}_{t_0}, \dots, \mathcal{Y}_{t_n}\}$, where $\mathcal{Y}_t = \{\mathbf{y}_{t+\Delta_t}\}$, Δ_t is the length of prediction horizon, and $\mathbf{y}_{t+\Delta_t} \in \mathbb{R}^N$ is the future cross-section returns of N stocks in the market over the time period from t to $t + \Delta_t$.

Our target is to learn an optimal Θ^* of a data driven model $f(\cdot; \Theta)$ with learnable parameter Θ such that $\hat{\mathcal{Y}} \leftarrow f(\mathcal{D}; \Theta)$.

4 Methodology

In this section, we introduce HireVAE to identify the market regime with online learning in conjunction with a linear stabilization clustering algorithm and learn an adaptive factor model to extract the latent logic behind the market and stocks under different regimes. The brief framework of HireVAE is given in Figure 2.

The architecture of our model is derived from a VAE encoder-decoder structure. In contrast to the classical

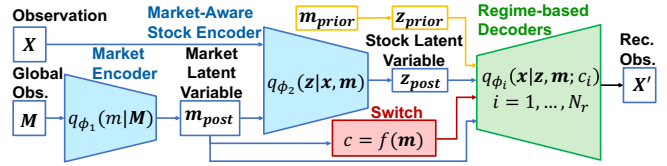


Figure 2: The brief architecture of HireVAE.

pipeline, we develop a hierarchical encoder with a two-level latent space representing a market-stock hierarchy, and couple it with a set of regime-specific decoders that reconstruct or predict stock returns based on different market situations. First, the market encoder extracts market latent factors from multimodal features. Then, the stock encoder plays the role of a conditional encoder that extracts stock latent factors based on stock features and market latent factors. This hierarchical structure can help the stock latent factor better recognize the market situation. In the middle of this encoder-decoder architecture, a regime-switching module acts as a mediator to match an appropriate regime-specific decoder to the latent market situation.

Specifically, this architecture consists of four parts: an explanation of the feature extraction procedure from sequential data, an overview of the hierarchical encoder-decoder structure, a regime switching learning algorithm, and the learning objectives in prior-posterior learning.

4.1 Extracting Features from Sequential Data

We establish two inherent feature extractors, *i.e.*, stock-specific and market-specific extractors.

Stock Features. The stock feature extractor ϕ_s^{feat} , extracts features \mathbf{e}_s from the historical stock characteristics: $\mathbf{e}_s = \phi_s^{feat}(\mathbf{X})$, where $\mathbf{X} \in \mathbb{R}^{T \times N \times C}$ and $\mathbf{e}_s \in \mathbb{R}^{N \times H}$ represents the extracted stock features with a hidden size H . Specifically, \mathbf{e}_s is the latest hidden state obtained by a gated recurrent unit which summarizes all temporal information.

Market Features. For sequential market data $\mathbf{M}_t \in \mathbb{R}^{T \times C_m}$ consists of information with d multimodal features, we introduce a set of extractors $\phi_{m,i}^{feat}(\cdot)$ associated with different modalities $\mathbf{M}_t^i \in \mathbb{R}^{T \times C_i}$, $i = 1, \dots, d$, and $\sum_{i=1}^d C_i = C_m$. We use $d!$ cross attention modules [Hou *et al.*, 2019] to learn pairwise features from each modality pair. Later, we extract the market features \mathbf{v} from all these pairwise features by concatenation. In our setting, a market can be represented by an index composed of all the stocks in the market. The different modalities are the market momentums (defined as the cumulative return over a period of time), the levels of relative trading volume, and the market volatilities over different time frames.

4.2 Hierarchical Encoder-Decoder Structure

To help the model better identify the market situation and better interpret future stock returns in different market situations, we design the latent space in a hierarchical structure that represents latent factors from the market and stocks. A pair of encoders is designed to learn latent variables from market and stocks based on the market and stock features.

Posterior Structure in Training Phase

In the training phase, the market encoder generates a market latent variable, based on which and in conjunction with both future stock returns and historical stock features a market-aware stock encoder can learn stock latent features. After the current market regime is identified by a regime switching algorithm, a regime-specific decoder can reconstruct future stock returns.

Market Encoder. The market encoder ϕ_{enc}^m extracts posterior market latent factor \mathbf{m} from the extracted market information \mathbf{v} , and future average return of all stocks $\bar{\mathbf{y}}$:

$$[\boldsymbol{\mu}_{post}^m, \boldsymbol{\sigma}_{post}^m] = \phi_{enc}^m(\mathbf{v}, \bar{\mathbf{y}}),$$

$$\mathbf{m} \sim \mathcal{N}(\boldsymbol{\mu}_{post}^m, \text{diag}(\boldsymbol{\sigma}_{post}^m)),$$

where $\mathbf{m} \in \mathbb{R}^{H_m}$ is the market latent factor following a Gaussian distribution, parameterized by mean $\boldsymbol{\mu}_{post}^m$ and standard deviation $\boldsymbol{\sigma}_{post}^m$ and H_m is the market hidden size.

Market-Aware Stock Encoder. The stock encoder ϕ_{enc}^s extracts posterior stock latent factors \mathbf{z} from the market latent factor \mathbf{m} , stock features \mathbf{e}_s , and future stock returns \mathbf{y} :

$$[\boldsymbol{\mu}_{post}^s, \boldsymbol{\sigma}_{post}^s] = \phi_{enc}^s(\mathbf{m}, \mathbf{e}_s, \mathbf{y}),$$

$$\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_{post}^s, \text{diag}(\boldsymbol{\sigma}_{post}^s)),$$

where $\mathbf{z} \in \mathbb{R}^{H_s}$ is the stock latent factors following a Gaussian distribution, parameterized by mean $\boldsymbol{\mu}_{post}^s$ and standard deviation $\boldsymbol{\sigma}_{post}^s$ and H_s is the stock hidden size.

Regime-Switch Factor Decoder. We develop a learning algorithm to distinguish the current market situation from different regimes in the latter part of the paper (Section 4.3). To learn an adaptive factor model for different market situations, we design a regime-switch factor decoder consisting of N_k sub-decoders that can reconstruct future stock returns $\hat{\mathbf{y}}$ based on stock latent factors \mathbf{z} and market latent factors \mathbf{m} :

$$\hat{\mathbf{y}} = \phi_{dec}(\mathbf{z}, \mathbf{m}, \mathbf{e}_s; c),$$

where $c \in \{1, \dots, N_k\}$ denotes a market regime. More precisely, $c = f(\mathbf{m})$ where $f(\cdot)$ is a switching function (algorithm) that can identify the regime based on the market latent factors \mathbf{m} .

Prior Structure in Prediction Phase

First, the market prior generates market latent factors \mathbf{m}_0 based on the extracted market information \mathbf{v} from various modalities. Later, the stock prior generates stock latent factors \mathbf{z}_0 based on the market prior variables \mathbf{m}_0 and stock characteristics \mathbf{e}_s . The decoder predicts future returns based on the market-stock priors and the regime c determined by the switch function. Specifically,

$$[\boldsymbol{\mu}_0^m, \boldsymbol{\sigma}_0^m] = \phi_{prior}^m(\mathbf{v}),$$

$$\mathbf{m}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0^m, \text{diag}(\boldsymbol{\sigma}_0^m)),$$

$$[\boldsymbol{\mu}_0^s, \boldsymbol{\sigma}_0^s] = \phi_{prior}^s(\mathbf{m}_0, \mathbf{e}_s),$$

$$\mathbf{z}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0^s, \text{diag}(\boldsymbol{\sigma}_0^s)),$$

$$\hat{\mathbf{y}} = \phi_{dec}(\mathbf{z}_0, \mathbf{m}_0, \mathbf{e}_s; c) \text{ with } c = f(\mathbf{m}_0),$$

where $\mathbf{m}_0 \in \mathbb{R}^{H_m}$ follows a Gaussian distribution with parameters $\boldsymbol{\mu}_0^m$ and $\boldsymbol{\sigma}_0^m$, and $\mathbf{z}_0 \in \mathbb{R}^{H_s}$ follows a Gaussian distribution with parameters $\boldsymbol{\mu}_0^s$ and $\boldsymbol{\sigma}_0^s$.

Algorithm 1 Linear stabilization clustering algorithm

Input: Training data $\mathcal{D}^{train} = \{\mathcal{D}_1^{batch}, \dots, \mathcal{D}_{N_b}^{batch}\}$ s.t. all data items are sorted in chronological order

Parameter: Model parameters Φ , regime distribution parameters $\boldsymbol{\mu}_r \in \mathbb{R}^{N_r}$ and $\boldsymbol{\sigma}_r \in \mathbb{R}_+^{N_r}$

Output: Regime $\mathbf{c}_i, i = 1, \dots, N_b$

- 1: Randomly initialize $\boldsymbol{\mu}_r$ and $\boldsymbol{\sigma}_r$ with $\boldsymbol{\sigma}_r \geq 0$
 - 2: **while** $i \leq N_b$ **do**
 - 3: Projection to 1D score space: $\mathbf{s}_i \leftarrow \text{Score}(\mathcal{D}_i^{batch})$
 - 4: Pred. cluster center: $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i \leftarrow \text{DistShift}(\mathcal{D}_i^{batch})$
 - 5: Sort cluster center: $\boldsymbol{\mu}_i^*, \boldsymbol{\sigma}_i^* \leftarrow \text{Sort}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$
 - 6: Update cluster center:

$$\boldsymbol{\mu}_r \leftarrow \beta \boldsymbol{\mu}_r + (1 - \beta) \boldsymbol{\mu}_i^*$$

$$\boldsymbol{\sigma}_r \leftarrow \beta \boldsymbol{\sigma}_r + (1 - \beta) \boldsymbol{\sigma}_i^*$$
 - 7: Calc. regime prob.: $\mathbf{p}_i \leftarrow \mathcal{N}(\mathbf{s}_i; \boldsymbol{\mu}_r, \text{diag}(\boldsymbol{\sigma}_r^2))$
 - 8: Pred. regime: $\mathbf{c}_i \leftarrow \text{ArgMax}(\mathbf{p}_i)$
 - 9: **end while**
 - 10: **return** regime $\mathbf{c}_i, i = 1, \dots, N_b$
-

4.3 Regime Switching with Online Learning

In order to learn an adaptive factor model that can handle various regimes, the first step is to identify these regimes based on point-in-time (PIT) market information. To achieve consistent and smooth market clustering based on a market latent variable $\mathbf{m} \in \mathbb{R}^{H_m}$ extracted from the mixed market features $\mathbf{v} \in \mathbb{R}^{H_c}$, we develop an online learning clustering algorithm with linear stabilization summarized in Algorithm 1. First, we project the market latent variable onto a 1D measure space, and consider its relative score in this 1D space as a random variable from a market regime distribution. Then, we compute the log-likelihood of this score *w.r.t* the distributions of different market regimes and choose the regime with largest log-likelihood as the predicted regime.

Projection to 1D Space. A project layer ϕ^{proj} projects the market latent variable $\mathbf{m} \in \mathbb{R}^{H_m}$ onto a 1D regime measure space. We define $s = \phi^{proj}(\mathbf{m})$, $s \in \mathbb{R}$, as the ‘‘score’’ of the current market situation and it plays a role as a 1D measure of the overall market status. This market score s is considered as a given sample in the sample spaces of the different regimes. Later, we can measure how much more likely this market score s belongs to a given market regime.

Determination of Market Regimes. Denote the number of market regimes as N_r . We assume that \mathbf{r} is a Gaussian random vector given by $\mathbf{r} \sim \mathcal{N}(\boldsymbol{\mu}_r, \text{diag}(\boldsymbol{\sigma}_r))$, where $\boldsymbol{\mu}_r \in \mathbb{R}^{N_r}$ and standard deviation $\boldsymbol{\sigma}_r \in \mathbb{R}_+^{N_r}$ are the representatives of regime centers and regime deviations *w.r.t* N_r regime clusters. To keep each regime as different as possible, we use Kullback-Leibler divergence (KL) to measure the difference between two distributions and define the loss function in terms of market regimes as

$$L_{reg} = - \sum_{\substack{i=1, \dots, N_r \\ j=i+1, \dots, N_r}} \text{KL}(\mathcal{N}(\mu_i, \sigma_i^2) \parallel \mathcal{N}(\mu_j, \sigma_j^2)),$$

where μ_i and σ_i are the i -th element of the mean vector $\boldsymbol{\mu}_r$ and the deviation vector $\boldsymbol{\sigma}_r$, respectively.

Regime Learner. To determine the centers of each regime, the regime learner ϕ_{reg} predicts the parameters of the market regime distribution, *i.e.* $[\boldsymbol{\mu}_r, \boldsymbol{\sigma}_r] = \phi_{reg}(\mathbf{m})$. Specifically,

$$\begin{aligned} \boldsymbol{\mu}_r &= \mathbf{w}_{\mu,r} \mathbf{v} + \mathbf{b}_{\mu,r}, \\ \boldsymbol{\sigma}_r &= \text{Softplus}(\mathbf{w}_{\sigma,r} \mathbf{v} + \mathbf{b}_{\sigma,r}). \end{aligned}$$

To keep the shift of regimes consistent and smooth in 1D measure space, a sorting function is applied to the mean values of all regime centers $\boldsymbol{\mu}_r \in \mathbb{R}^{N_r}$ in descending order and the corresponding deviation terms are reordered *w.r.t.* the new order of sorted mean values. Formally,

$$\begin{aligned} [\boldsymbol{\mu}_r^*, \mathbf{I}_r^*] &= \text{Sort}(\boldsymbol{\mu}_r), \\ \boldsymbol{\sigma}_r^* &= \boldsymbol{\sigma}_r[\mathbf{I}_r^*], \end{aligned}$$

where $\boldsymbol{\mu}_r^*$ is a reordered mean vector, \mathbf{I}_r^* is its corresponding reordered index *w.r.t.* its original index, and $\boldsymbol{\sigma}_r^*$ is a reordered deviation vector *w.r.t.* this reordered index \mathbf{I}_r^* . These reordered mean and deviation vectors obtained in each batch are later used to update each regime center ($\boldsymbol{\mu}_r$ and $\boldsymbol{\sigma}_r$) with a weighted moving average, allowing the regime center to be updated gradually.

Regime Prediction. We assume that the market score s is a random variable generated from a market regime distribution defined as a Gaussian distribution. For N_k regimes, we have N_k market regime distributions, for each of which we can calculate the log-likelihood of the current market score s :

$$l(s; \boldsymbol{\mu}_r^i, \boldsymbol{\sigma}_r^i) = -\ln(\boldsymbol{\sigma}_r^i) - \frac{1}{2} \ln(2\pi) - \frac{1}{2} \left(\frac{(s - \boldsymbol{\mu}_r^i)^2}{\boldsymbol{\sigma}_r^i} \right),$$

where $i = 1, \dots, N_k$. The regime is predicted to be the regime with the maximum log-likelihood among all regimes:

$$c = \text{argmax}_j [l(s; \boldsymbol{\mu}_r^i, \boldsymbol{\sigma}_r^i), i = 1, \dots, N_k],$$

where c indicates the predicted market regime.

4.4 Learning Objective

Our learning objective consists of three parts, where the first part is to learn a better future stock return reconstruction, the second part is to learn latent market features and latent stock features in a hierarchical structure, and the third part is to learn a more differentiated regime clusters. Thus, the overall loss function of our HireVAE is as follows,

$$\begin{aligned} L_{rec} &= - \sum_{r=1}^{N_r} \mathbb{1}_{f(\mathbf{m})}(r) \ln P_{\phi_{dec}}(\hat{\mathbf{y}} = \mathbf{y} | \mathbf{X}, \mathbf{M}, \mathbf{z}, \mathbf{m}; r) \\ L_{hier} &= \text{KL}(P_{\phi_{enc}}(\mathbf{m} | \mathbf{M}, \bar{\mathbf{y}}) || P_{\phi_{prior}}(\mathbf{m} | \mathbf{M}, \bar{\mathbf{y}})) + \\ &\quad \text{KL}(P_{\phi_{enc}}(\mathbf{z} | \mathbf{X}, \mathbf{M}, \mathbf{m}, \mathbf{y}) || P_{\phi_{prior}}(\mathbf{z} | \mathbf{X}, \mathbf{M}, \mathbf{m})) \\ L_{overall} &= L_{rec} + L_{hier} + L_{reg}. \end{aligned}$$

5 Experiments

In this section, we evaluate the proposed HireVAE on real stock market data, choosing one of the largest emerging markets as representative, and demonstrate the effectiveness of

Method	IC	Rank IC	Rank ICIR
Linear	0.030	0.031	0.322
GRU	0.046	0.050	0.484
MLP	0.053	0.051	0.537
Trans	0.050	0.040	0.253
GAT	0.029	0.032	0.466
IGAT	0.012	0.018	0.210
DMFM	0.014	0.015	0.480
VAE	0.049	0.059	0.539
CVAE	0.053	0.063	0.628
HiReVAE	0.058	0.066	0.734

Table 1: Evaluation the performance of various methods in terms of their predictive power.

our model through various experiments. These experiments are designed to investigate the following research questions.

Exp 1: Does our method outperform the state-of-the-art methods in terms of factor investing?

Exp 2: Is the improvement in effectiveness mainly due to the additional market information?

Exp 3: How does regime switch help adaptive learning in factor models?

Exp 4: Is the hierarchical structure necessary in our design?

5.1 Experiment Settings

Dataset Construction. We conduct the experiment over the entire China stock market, with price-volume and fundamental data obtained from the WIND database, and all raw data processed on a point-in-time basis. The stock pool is constructed based on all on-listing stocks, except for stocks that receive special treatments (ST) and those that have been on the market for less than three months. When a company is labeled as ST, which means it suffer losses for two or more consecutive years. We follow the same paradigm as in [Wei *et al.*, 2022] to construct stock features and select 58 features that have a good coverage over the entire stock market. The length of the sequential data and the prediction horizon are $T = 20$ and $\Delta t = 20$, respectively. The stock returns are calculated on a basis of volume weighted average prices. We choose the dataset and data split settings as in [Wei *et al.*, 2022] instead of [Duan *et al.*, 2022] because the former better covers the period from 2010 to mid-2022 and has a longer test set from mid-2015 to mid-2022 than the latter from 2019 to 2020. There are 14 groups of training, validation, and test datasets. To prevent information leakage, we delete the data items from the validation dataset when the future stock returns in the validation dataset are covered by the time period of the test dataset.

Baselines. To allow a fair comparison, we provide the same sequential data to all methods. We compare our HiReVAE to the factor models as follows: ① **Linear** is a linear factor model derived from [Fama and French, 1992], with average pooling on the sequential dimension. ② **GRU** [Chung *et al.*, 2014] is a factor model in which a GRU extracts the sequential features and, in conjunction with a linear layer, predicts future returns. ③ **Trans** [Ding *et al.*, 2020] is a factor model

Method	CSI 300		CSI 500		CSI 1000		CNI 2000		CSI ALL	
	AR↑	IR↑	AR↑	IR↑	AR↑	IR↑	AR↑	IR↑	AR↑	IR↑
Linear	11.94	0.30	8.81	0.50	6.98	0.65	5.04	0.49	5.82	0.89
GRU	19.40	0.60	15.04	0.80	12.89	0.96	11.27	0.84	13.25	2.20
MLP	23.04	0.78	20.20	1.10	18.12	1.22	16.87	<u>1.09</u>	19.25	3.53
Trans	7.21	0.18	4.63	0.28	3.03	0.41	0.68	0.24	0.22	0.27
GAT	5.77	0.14	5.65	0.19	4.33	0.29	2.30	0.16	2.08	0.31
IGAT	12.67	0.41	10.32	0.61	8.47	0.69	6.85	0.54	8.43	2.04
DMFM	3.16	0.05	2.13	0.20	0.78	0.35	1.50	0.18	2.24	0.32
VAE	26.49	0.82	25.04	1.11	23.32	1.23	19.05	1.02	23.72	4.38
CVAE	27.78	0.91	25.84	<u>1.24</u>	<u>24.00</u>	1.36	<u>22.76</u>	0.99	<u>24.91</u>	<u>5.46</u>
HiReVAE	30.89	<u>0.89</u>	29.05	<u>1.23</u>	27.14	<u>1.35</u>	25.94	1.18	28.02	5.92

Table 2: The performances on portfolio construction over the period from mid-2015 to mid-2022 (**best/2nd best**).

based on Transformer. ④ **MLP** [Levin, 1995] is the first non-linear factor model with multilayer perceptrons, where we equip it with a GRU to extract time-series features. ⑤ **GAT** [Veličković *et al.*, 2018] is a graph attention work with a fully connected stock graph. ⑥ **IGAT** [Wei *et al.*, 2022] is a GAT, based on a stock-industry graph where there is an edge between two nodes if two stocks belong to the same industry. ⑦ **DMFM** [Wei *et al.*, 2022] is a deep multifactor model built on top of a stock graph. ⑧ **VAE** [Duan *et al.*, 2022] is a stock prediction method based on VAE that treats latent stock variables as the optimal factors thus learned. ⑨ **CVAE** [Gu *et al.*, 2021] is a factor model with a conditional autoencoder.

5.2 Exp 1: Effectiveness of HireVAE

In this experiment, we verify the effectiveness of our HireVAE with respect to factor investing. In this context, factor models are the essential tools in factor investing, where the objective is to answer two practical questions: 1) whether the model explains (future) stock returns well and 2) whether it provides good stock selection.

Predictive Power

To evaluate the predictive power of the compared methods, we use two metrics widely used in industry and academia [Duan *et al.*, 2022; Wei *et al.*, 2022], *i.e.*, the rank information coefficient (IC), the rank information coefficient (RankIC), and the information ratio of RankIC (RankICIR), where RankICIR is the z-score of RankIC. To reconcile deterministic and generative methods in prediction, we use the means of the hierarchical priors as the deterministic values and determine the predicted stock return by $\hat{y} = \phi_{dec}(\mu_0^s, \mu_0^m; e_s; c)$. By comparison in Table 1, HireVAE outperforms the compared methods, illustrating its effectiveness in stock prediction.

Stock Selection

The most widely used method to test the ability of a factor model in portfolio construction is *Top-1-of-G Groups*. Specifically, this involves sorting stocks in descending order *w.r.t* the expected returns estimated by a factor model and then dividing the sorted stocks into *G* groups. In addition, a long-short portfolio can be created by buying the *Top-1* group and short selling the *Bottom-1* group simultaneously. Since short selling in Chinese stock markets is not always possible for

Method	Mkt	rIC	rICIR	AR	IR
GRU	✓	0.051	0.484	13.180	2.202
GRU-s	✗	0.050	0.485	13.249	2.107
MLP	✓	0.051	0.537	19.247	3.525
MLP-s	✗	0.051	0.540	19.251	3.454
VAE	✓	0.059	0.539	23.724	4.377
VAE-s	✗	0.059	0.540	23.589	4.267

Table 3: Ablation study of the difference in the use of market information. RIC rICIR, AR and IR are abbreviations for rank IC, rankICIR and active return, portfolio information ratio, respectively. The “-s” stands for the exclusive use of stock characteristics.

each stock, we construct only the portfolio holding the *Top-1* group. The frequency of rebalancing is monthly. All transaction costs and taxes are included in our backtesting.

The CSI300, the CSI500, the CSI1000 and the CNI2000 are widely used benchmarks in China stock markets and consist of the most representative stocks in market hierarchy. In particular, the CSI300 consists of the 300 most liquid and largest stocks. The CSI500, representing mid-caps, consists of the 500 most liquid and largest stocks. The CSI1000 consists of the following 1000 stocks representing small caps. In addition, the CNI2000 is another representative of small caps and consists of 2000 stocks.

To test the potential of subsequent use in the construction of enhanced index funds (EIFs), we compare the different methods. We report two metrics that are widely used in investment practice: the annualized active return (AR) relative to an investment benchmark and the information ratio of a portfolio (IR), the latter being a risk-adjusted measure. We use \uparrow to indicate that a larger value is better. As shown in Table 2, the HireVAE can achieve better active returns over a long investment period across different market hierarchies.

5.3 Exp 2: Improvement Not From Extra Info

Theoretically, see our experiment settings in Section 5.1, we use all stocks included in the CSI All A Share Index, an overall market representative of Chinese stock markets. At its core, the index is roughly calculated by a capitalization-weighted sum of the prices of all stocks, which means that

Method	Algo	Enc	rIC	rICIR	AR
RVAE-v	Rule	✗	0.05	0.67	21.62
RVAE	Clus	✗	0.05	0.50	13.72
RCVAE-v	Rule	✗	0.04	0.38	14.85
RCVAE	Clus	✗	0.03	0.32	2.61
HiReVAE-v	Rule	✓	0.06	0.67	24.54
HiReVAE	AOL	✓	0.07	0.73	28.02

Table 4: Ablation study on the rule-based and the end-to-end regime switching identifier in terms of their predictive power and stock selection performance on all Chinese A-shares. The rIC, rICIR, AR are abbreviations for rank IC, rankICIR and active return, respectively. “-v” stands for regime change based on rule-based algorithm.

this index is calculated based on the most basic stock characteristics, price and market capitalization. Some of the original stock characteristics are also calculated based on these two basic characteristics. Thus, the market index is a “global observation” of all stocks and no additional information is added to the original raw data.

To provide empirical support for the previous argument, we conduct two experiments. First, all methods use the same input characteristics for stocks and the market, and the corresponding results are reported in Table 1 and 2. Second, in Table 3, we show the comparisons between the baseline methods in terms of using market information, selecting the baseline methods that performed well in Exp 1 as our baseline methods in Exp 2. The results show that the additional market information does not help to improve the performance of a model.

5.4 Exp 3: Adaptive Learning is Better

Determining the regime on the basis of volatility is a way of establishing regimes on the basis of rules of thumb [Botte and Bao, 2021]. Inspired by this knowledge and experience, we develop a rule-based method shown in Algorithm 2. We consider this method as a baseline for regime switching to its online and adaptive counterpart. Table 4 shows the performance of the different regime-switching algorithms in stock prediction and portfolio construction, where we select the most recommended stocks from the entire market (10% stocks from the market). In different modeling settings, our adaptive regime switching algorithm can always outperform the rule-based and neural clustering counterparts.

5.5 Exp 4: Hierarchy is Necessary

In this experiment, we analyze the necessity of using a hierarchical structure in two ways: first, does a hierarchical latent space (HVAE) encode market information better than a single latent space (VAE)? Second, does the hierarchical structure help the end-to-end clustering algorithm in adaptive learning of market regimes.

Table 5 can answer these two questions accordingly. As for the first question, encoding market information alone does not provide significant improvement in the non-regime-switching modeling, since the metrics of VAE and HiVAE are similar. This is reasonable, as we showed in Section 5.3, where the market information itself does not provide more information

Algorithm 2 Volatility-based regime identification

Input: Volatility data $\mathcal{D}_\sigma = \{\sigma_{t_0}, \dots, \sigma_{t_n}\}$ where all volatility data items are sorted in chronological order, and the number of regimes K ($K \geq 2$)

Output: Regime c_t

- 1: Initialize $\sigma_{min} = +\infty$ and $\sigma_{max} = -\infty$
- 2: Initialize $I_K = \left[\frac{(K-1)}{K}, 1 \right]$ and $I_k = \left[\frac{(k-1)}{K}, \frac{k}{K} \right)$, for $k = 1, \dots, K - 1$
- 3: **while** $t \leq t_n$ **do**
- 4: $\sigma_{min} \leftarrow \min(\sigma_{min}, \sigma_t)$
- 5: $\sigma_{max} \leftarrow \max(\sigma_{max}, \sigma_t)$
- 6: $\sigma_{level} \leftarrow (\sigma_t - \sigma_{min}) / (\sigma_{max} - \sigma_{min})$
- 7: **while** $k \leq K$ **do**
- 8: **if** $\sigma_{level} \in I_k$ **then**
- 9: $c_t \leftarrow k$.
- 10: **end if**
- 11: **end while**
- 12: **end while**
- 13: **return** regime c_t

Method	Hier	MLV	rIC	rICIR	AR
VAE	✗	✓	0.06	0.63	23.72
HiVAE	✓	✓	0.07	0.68	23.13
RVAE	✗	✓	0.05	0.50	13.72
RCVAE	✗	✓	0.03	0.32	2.61
HiReVAE	✓	✓	0.07	0.73	28.02

Table 5: The result of the experiment showing the need for a hierarchical structure in learning an online and adaptive regime switching identifier. The MLV is the abbreviation for the market latent variable, which indicates whether the MLV participates in regime switching learning.

than the raw data. However, in the case of regime switching, the market latent variable plays an important role in end-to-end regime switching learning, as it is used to estimate the regime region and predict the corresponding indicators that measure the probability of being in each regime. This can be empirically demonstrated by the fact that HireVAE outperforms the other two regime switching methods without considering the market latent variable, RVAE and RCVAE, in terms of predictive power and stock selection ability.

6 Conclusion

In this paper, we present HireVAE, a novel end-to-end neural factor model that can identify current market regime according to point-in-time market information, and subsequently adapt itself for better prediction. HireVAE achieves such an ability by a pair of hierarchically organized encoders respectively process global market situation and stock-wise latent factors. Being the first online and adaptive regime-switch factor model, HireVAE achieves superior performance on real stock market data, outperforming classical linear methods and recent non-linear data-driven methods by a large margin.

Acknowledgments

This project is funded in part by Shanghai AI Laboratory, CUHK Interdisciplinary AI Research Institute, and the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnoHK.

References

- [Akiyamen *et al.*, 2020] Peter Akiyamen, Yi Zhou Tang, and Hussien Hussien. A hybrid learning approach to detecting regime switches in financial markets. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–7, 2020.
- [Ang and Bekaert, 2002] Andrew Ang and Geert Bekaert. Regime switches in interest rates. *Journal of Business & Economic Statistics*, 20(2):163–182, 2002.
- [Botte and Bao, 2021] Alex Botte and Doris Bao. A machine learning approach to regime modeling. *Two Sigma*, 2021.
- [Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [Chung *et al.*, 2015] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.
- [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.
- [Duan *et al.*, 2022] Yitong Duan, Lei Wang, Qizhong Zhang, and Jian Li. Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. In *36th AAAI Conference on Artificial Intelligence*, 2022.
- [Fama and French, 1992] Eugene F Fama and Kenneth R French. The cross-section of expected stock returns. *the Journal of Finance*, 47(2):427–465, 1992.
- [Fama and French, 2020] 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.
- [Gu *et al.*, 2021] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Autoencoder asset pricing models. *Journal of Econometrics*, 222(1):429–450, 2021.
- [Guidolin and Timmermann, 2007] Massimo Guidolin and Allan Timmermann. Asset allocation under multivariate regime switching. *Journal of Economic Dynamics and Control*, 31(11):3503–3544, 2007.
- [Hamilton, 1989] James D Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica: Journal of the econometric society*, pages 357–384, 1989.
- [Hou *et al.*, 2019] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. *Advances in Neural Information Processing Systems*, 32, 2019.
- [Kingma and Welling, 2013] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [Levin, 1995] Asriel Levin. Stock selection via nonlinear multi-factor models. *Advances in Neural Information Processing Systems*, 8, 1995.
- [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.
- [Lin *et al.*, 2021] 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.
- [Luo *et al.*, 2018] Rui Luo, Weinan Zhang, Xiaojun Xu, and Jun Wang. A neural stochastic volatility model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [Mari and Mari, 2022] Carlo Mari and Emiliano Mari. Deep learning based regime-switching models of energy commodity prices. *Energy Systems*, pages 1–22, 2022.
- [Ng *et al.*, 1992] Victor Ng, Robert F Engle, and Michael Rothschild. A multi-dynamic-factor model for stock returns. *Journal of Econometrics*, 52(1-2):245–266, 1992.
- [Su *et al.*, 2018] Jinsong Su, Shan Wu, Deyi Xiong, Yaojie Lu, Xianpei Han, and Biao Zhang. Variational recurrent neural machine translation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [Uysal and Mulvey, 2021] A Sinem Uysal and John M Mulvey. A machine learning approach in regime-switching risk parity portfolios. *The Journal of Financial Data Science*, 3(2):87–108, 2021.
- [Vahdat and Kautz, 2020] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.
- [Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- [Wang *et al.*, 2020] Matthew Wang, Yi-Hong Lin, and Ilya Mikhelson. Regime-switching factor investing with hidden markov models. *Journal of Risk and Financial Management*, 13(12), 2020.
- [Wang *et al.*, 2021a] Heyuan Wang, Shun Li, Tengjiao Wang, and Jiayi Zheng. Hierarchical adaptive temporal-relational modeling for stock trend prediction. In *IJCAI*, pages 3691–3698, 2021.

- [Wang *et al.*, 2021b] Heyuan Wang, Tengjiao Wang, Shun Li, Shijie Guan, Jiayi Zheng, and Wei Chen. Heterogeneous interactive snapshot network for review-enhanced stock profiling and recommendation. In *IJCAI*, pages 4541–4547, 2021.
- [Wang *et al.*, 2022] Heyuan Wang, Tengjiao Wang, Shun Li, Jiayi Zheng, Shijie Guan, and Wei Chen. Adaptive long-short pattern transformer for stock investment selection. In *IJCAI*, pages 3970–3977, 2022.
- [Wei *et al.*, 2022] Zikai Wei, Bo Dai, and Dahua Lin. Factor investing with a deep multi-factor model. In *NeurIPS Workshop on GLInDA*, 2022.
- [Xu and Cohen, 2018] Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1970–1979, 2018.
- [Zhang *et al.*, 2017] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 2141–2149, 2017.
- [Zhu, 2022] Brian Zhu. Regime-switching factor models with applications to portfolio selection and demand estimation. https://economics.yale.edu/sites/default/files/Forms/zhu_brian_senior_essay_econ_492.pdf, 2022. Online; accessed 10 January 2023.