BerkeleyHaas

230ZB: Deep Learning and Applications II

University of California Berkeley: Haas School of Business

Master of Financial Engineering

Beyond FactorVAE

Authors:

Di Venti, Matteo Mario; Ng, Anson; Pan, Jianming; Shi, Jiacheng David; Xu, Yucheng

Date: October 25, 2023

1 Introduction

In the previous course, we initially replicated the paper 'FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder for Predicting Cross-Sectional Stock Returns', Yitong Duan, LeiWang, Qizhong Zhang, Jian Li.

In the paper, the authors blend dynamic factor models with Variational Auto-encoder (VAE) to forecast stock return cross-sections. Their approach involves deriving the prior distribution of latent factors through a prior-posterior learning method, facilitating learning from noisy data. They present compelling evidence that this model structure yields significant excess returns and outperforms numerous benchmark dynamic factor and deep learning models.

FactorVAE demonstrates superior performance across return prediction, robustness, and portfolio investments when compared to all benchmark models. It's worth noting, however, that the study is based on data from the Chinese stock markets. Further improvements to the model are required to address issues such as computational efficiency, risk modeling, and other relevant concerns.

In the subsequent sections of this report, our primary focus will be on discussing the encountered challenges, proposing potential solutions, and presenting the modified model along with its performance evaluation.

The code is available at **https://github.com/jmp41/230ZB**.

2 Existing Challenges

2.1 Computational Efficiency

The initial experiment was conducted using the CSI-300 dataset, encompassing a maximum of 300 stocks for both training and inference. This dataset served as a reasonable testbed

for the FactorVAE model. However, when we attempted to apply FactorVAE to the S&P 500, we encountered a notable computational efficiency challenge. The stark contrast in dataset size and complexity led to significantly longer training times, which became a substantial hindrance to practical implementation.

To provide some context, when we conducted a 25-epoch training run on the original dataset, the process took a manageable 1 hour and 42 minutes. However, when dealing with the S&P 500, the training time escalated dramatically to a staggering 7 hours and 11 minutes. Such extensive training times are neither efficient nor sustainable in real-world scenarios, rendering the model less practical for applications involving larger datasets.

Table 1: Training Duration Across Various Stock Universes

Model Complexity (CSI300)		Model Complexity(S&P500)	
Epoch	Training time	Epoch	Training time
5	0h 21 min	**5**	1h 23 min
10	0h 39 min	**10**	3h 17 min
25	1h 42 min	**25**	7h 11 min
50	4h 03 min	**50**	18h 22 min
100	9h 49 min	**100**	51h 12min

2.2 Risk Modeling

While drawing inspiration from traditional factor models for the development of our model, we introduced the concept of Posterior Prior Learning. However, our model initially omitted consideration of each asset's systematic risk. Specifically, it assumed that pairs μ_{prior} and σ_{prior} represented idiosyncratic risk, sampled independently from a normal distribution.

In a more realistic scenario, our model recognizes that factors are not solely tied to idiosyncratic risk but also encompass the systematic risk of the portfolio. Therefore, we intentionally integrated the risk model into the FactorVAE training process. This risk model estimates the variance-covariance matrix of assets, distinguishing between two components: systematic risk and idiosyncratic risk.

2.3 Alternative data features

Our model was originally built using the Alpha158 dataset from the Microsoft Qlib platform. This dataset containes 158 features extracted from price-volume data, which gave rise to the 60 risk factors we previously discussed. To enhance predictive capabilities and overall model performance, we will be working on alternative data sources to extract additional features. Factors usually represents a certain type of risk premium within the market. Using alternative data sources that are not readily available to the broader market participants, we aim to capture potentially profitable risks that investors can leverage to achieve excess returns.

3 Improvement

3.1 Dataset-S&P 500 & Alpha360

Our dataset was retrieved using the QLib library[1], developed by Microsoft, which provide an integrated platform for quantitative investing. Our stock universe corresponds to the S&P 500, which includes the top 500 stocks traded on the S&P 500 Index. We use different time periods between training, testing and validation to reduce information leakage between the datasets. Summary statistics can found on table 1. We use the off-the-shelf features provided by the QLib (Alpha360), which are 360 price-volume feature extracted from past 60 days price/volume data.

Parameter	Value
Stock universe	S&P 500 Index
Train period	2007 - 2017
Validation period	2018 - 2019
Test period	2020 - 09/2022
Number of features	360

Figure 1: Dataset description

[1] https://github.com/microsoft/qlib

3.2 Model Modifications

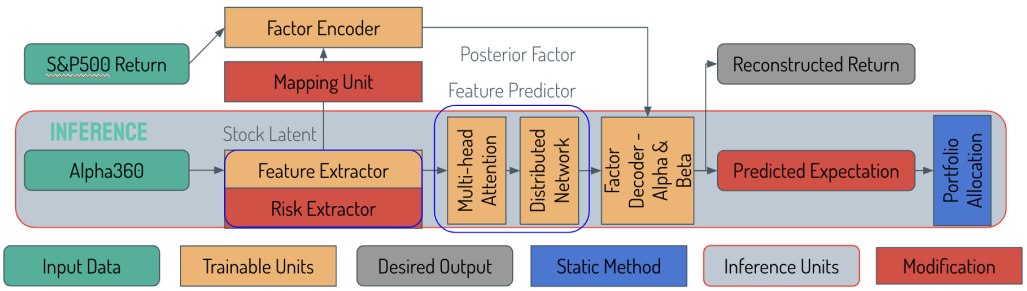

Figure 2: Overall framework of Modified FactorVAE

Our model comprises several key components, starting with a sequential feature extraction process involving a 'Feature Extractor' and a 'Risk Extractor.' The 'Feature Extractor' employs a 'Gate Recurrent Unit (GRU)' for extracting latent stock features, while the 'Risk Extractor' utilizes a 'Graph Attention Network (GAT)' to capture latent risk-related information.

Following the stock latent feature extraction, we employ a 'Mapping Unit' designed to reduce the high dimensionality of these latent factors. This reduction is achieved through an 'Auto Encoder' layer, which compresses the concatenation of latent factors, effectively reducing their scale.

A noteworthy modification occurs during the inference stage. In the original model, FactorVAE produced an output that included random sampled noise, which limited its robustness. In contrast, our modified approach deploys the mean of the last output layer as the output, a more common and stable choice for VAE-like models. This adjustment enhances the reliability and interpretability of the model's output.

3.3 Conditional VAE

A Conditional Variational Autoencoder (CVAE) is an extension of the Variational Autoencoder (VAE) that incorporates additional context or conditions during data generation. This context, which can be any relevant information, allows the CVAE to produce data that aligns with the given conditions. For example, in image generation, the condition might specify the type of object to generate, while in text generation, it could represent a particular topic or sentiment. This capability makes CVAEs a valuable tool in tasks requiring controlled and

guided data generation, such as image-to-image translation, style transfer, and various other conditional data generation tasks. The original model does not include information during factor encoding part, we therefore introduce risk extractor and mapping unit to guide the generation of factor mu and factor sigma. We are using the conditional ELBO as training loss:

$$\begin{aligned} \mathbb{E}_{x,y\sim\mathcal{D}} \log p_\theta(y \mid x) \geq \mathbb{E}_{x,y\sim\mathcal{D}}\left[\mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(y \mid x,z)\right] \\ -\mathbb{E}_{x,y\sim\mathcal{D}}\left[\mathrm{KL}\left(q_\phi(z \mid x,y)\|p(z \mid x)\right)\right] \end{aligned} \tag{1}$$

3.4 Graph Attention Network

GAT Veličković et al. [2017] is a novel neural network architectures that operate on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations.

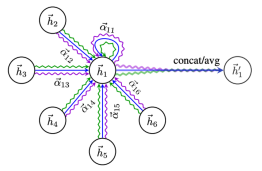

Figure 3: Graph Attention Mechanism

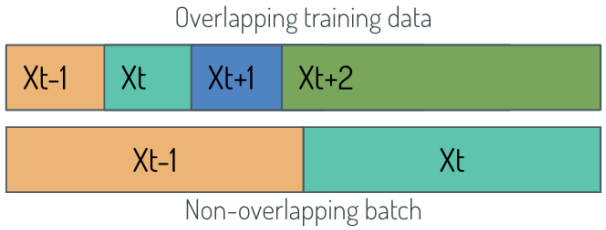

Figure 4: Efficient Batch Sampling Algorithm

An illustration of multihead attention (with K = 3 heads) by node 1 on its neighborhood. Different arrow styles and colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain h'_1. We are using the fully-connected GAT to extract information from Alpha360 and take it as the representation of risk features.

3.5 Computation efficiency

3.5.1 Non-overlapping batch sampling

The original dataset comprises price-volume data spanning the past 60 days. When training a model with such data, a significant issue arises due to substantial overlapping between consecutive data points. This overlap results in redundancy, as each consecutive dataset

essentially provides only one new data column when we are dealing with 360 features. To fully maximize the utility of the data, we employ a non-overlapping dataset approach. This involves using a batch lap mechanism to control the transition between two distinct batches of data. By doing so, we effectively reduce the batch length, mitigating the issue of redundancy and enabling more efficient utilization of the available information.

3.5.2 Efficient Batch Sampling Algorithm

Algorithm 1 Cache dataloader

Define cache

...

iteration through batch

for $i \in \{1, ...k\}$ **batch length do**

    yield indices$[i*batchsize*nonoverlappingsize:(i+1)*batchsize*nonoverlappingsize]$

end

get batch data (slicing index)

return cache[slicing index]

3.5.3 Accelerate Performance

Our model training process is 3x faster than the previous iteration, as evident from the results presented in the Results section, which also demonstrate improved prediction metrics.

Table 2: Time Duration for Training After Efficiency Improvement

Model Complexity		
Epoch	**Training time (FactorVAE)**	**Training time (ours)**
5	1h 23 min	**0h 31 min**
10	3h 17 min	**1h 19 min**
25	7h 11 min	**2h 32 min**
50	18h 22 min	**5h 03 min**
100	51h 12min	**13h 23 min**

4 Results

4.1 Model Performance

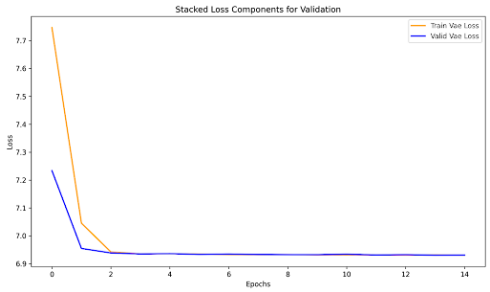

Figure 5: Training and Validation Loss

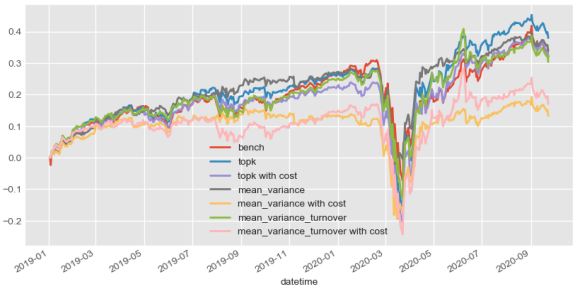

Figure 6: Cummulative return of different optimization methods

Figure 5 illustrates the training and validation loss. Loss quickly decreases and converges within three epochs during both trading and validation. That means our model learns the objective function well.

Table 3: Comparison between different models

Model Name	IC	ICIR	Rank IC	Rank ICIR	Annualized Return	Information Ratio	Max Drawdown
Linear	0.0197	0.0200	0.0172	0.0231	0.0092	0.11091	-0.25091
MLP	0.0010	0.0100	0.0163	0.0161	-0.03243	-0.38392	-0.22249
FactorVAE (Yitong Duan, et al.)	0.0143	0.0105	0.0107	0.0173	-0.04981	-021970	-0.24013
FactorVAE (Ours.)	**0.0343**	**0.0465**	**0.0390**	**0.0451**	**0.03709**	**0.34915**	**-0.19171**

Table 3 summarizes the performance of benchmark models, the original model in the paper, and our improved model in predicting SP500 return from 2017-01-01 to 2020-08-01. As we can observe from the table, the original paper performs not even better than the linear model on SP500. However, with our improvements, new FactorVAE performs much better than the original one. Its Rank IC achieves 3.9%, roughly 3% higher than that of original FactorVAE. Rank ICIR, which is defined as the ratio of the mean of Rank IC to the volatility of Rank IC, achieves 4.51%, roughly 3% higher than before. That is, the new prediction is more accuate and stable than before. We can safely conclude that the improvements we did significantly improve model performance on SP500.

4.2 Portfolio Investment

Table 4: Performance of Different Optimization Methods

	ann return	ann std	Sharpe	MDD	excess ann return	excess ann std
topk	0.207181	0.295416	0.721653	-0.441224	0.032691	0.108288
mean variance	0.184707	0.208869	0.909959	-0.312934	0.010218	0.143033

	information ratio	excess ann return with cost	excess ann std with cost
topk	0.310647	-0.000819	0.108279
mean variance	0.073508	0.102040	0.143060

The paper only adapts TopK-Drop strategy, which equally invests in 50 stock with highest predicted return, to measure the model performance in portfolio investing. We further implemented mean variance optimization, which is more commonly used in the industry.

Table 4 and Figure 6 show the result of portfolio investment based on the signals of improved FactorAVE. Ignoring the cost, the mean-variance portfolio achieves a Sharpe of 0.9. But this methods requires frequent position changes and results in high cost. TopK-Drop strategy, simpler but more robust, has smaller tracking error and lower cost. The TopK-Drop portfolio built on our signals has an information of 0.3 and equally good performance as PS500 after considering cost. Our model can achieve a profitable investment in real US market.

5 Analysis

5.1 Interpretability of Latent Factors

To better understand the latent factors, we analyse their connection to financial time series data and market indicators, to determine whether any factor can be traced to existing stocks, literature factors or industry indexes.

First of all, from the model training experience, we see that the number of factors utilised is an important hyperparameter that impacts performance of the model and factor quality. Performing the mean-variance clustering that we previously performed on the CSI-300 universe

factors seems to be be less effective in giving clear factor clusters. However, looking at just the factors returns time series, factors do seem to present some correlation groupings which can further visualized with a dendrogram. If the correlation is particularly strong, another thing that could be considered for future works is an hierarchical factor model like the one explored in a recently published paper that picks up from the FactorVAE model and adds hierarchical and regime-dependent factors (Wei et al. [2023]).

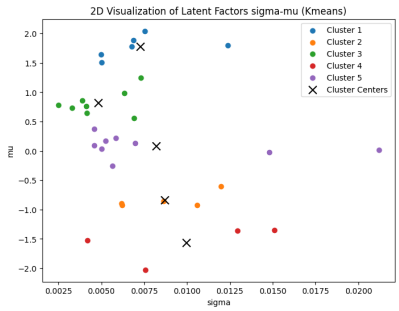

Figure 7: Mean-Variance clustering of factors

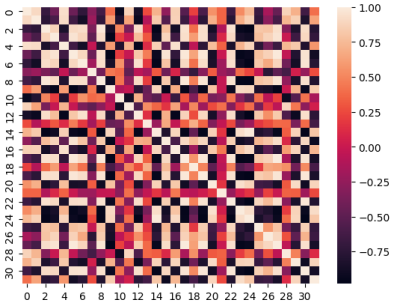

Figure 8: Correlation of factors based on returns time series

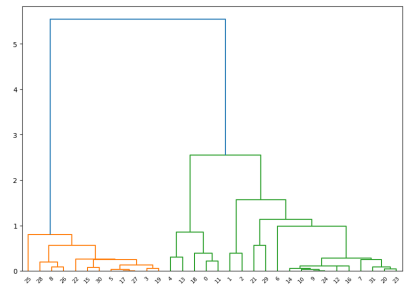

Figure 9: Dendrogram of clustered factors based on returns correlation

To analyze the factors direct connection to the data, we first run correlation and regression between the factors and the stock returns. We do not find any factor to correlate highly with or have a high explanation by a single stock. As this last analysis is very high-dimensional, it does not provide additional interpretability beyond a safety check.

Therefore, we check correlation of each factors with the return of each of the 11 industry indexes as well as running the regression of each industry index on the whole factor panel.

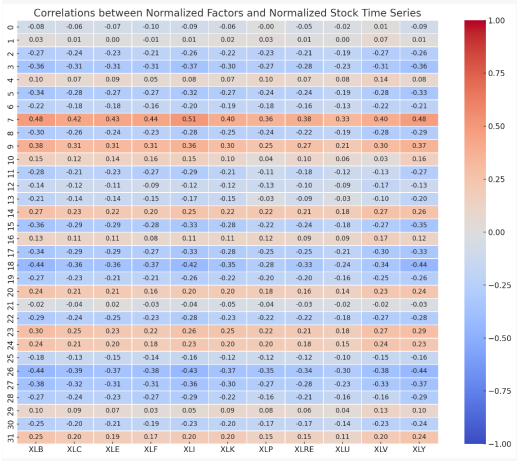

Figure 10: Correlation industry - factor

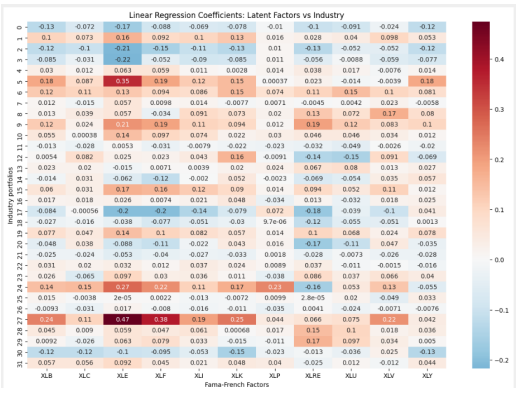

Figure 11: Normalised regression of factors on the industry

As can be seen, for most factors there is no clear connection to industry portfolios. However, it seems that the energy sector XLE loads significantly higher weight on factors 5 and 27 and the consumer sector XLP has predominantly weight on factor 24. Similarly we analyse the connection of the factors with the most known factors in the industry: the Fama-French factors. From the analyis, we see that there seems some weak connections between RMB and factor 24 as well as HML with a few specific factors. CMA factor seem to be highly uncorrelated with the latent factors. Overall, there is no clear direct factor attribution and it seems the correlation is quite small. This was in turn expected as the prediction horizon is 1 day and FF-factors (since are built from company fundamentals) are most useful for long prediction horizons.

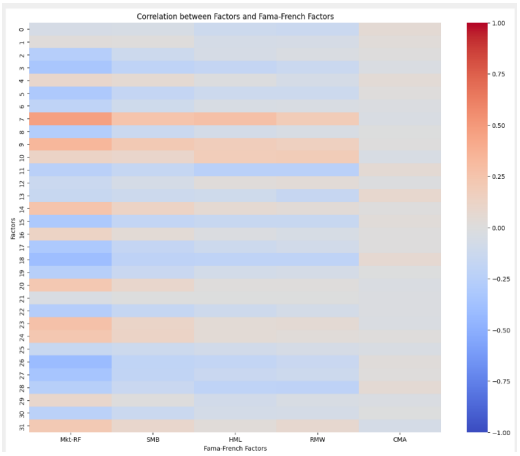

Figure 12: Correlation FF5 - factors

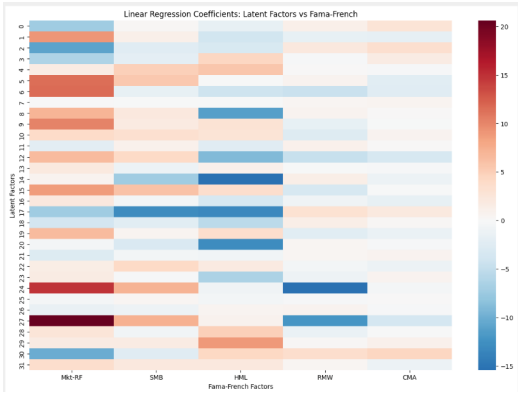

Figure 13: Normalised regression of FF5 on the factors

The factor analysis seems to highlight that some specific factors are partially tied to industry characteristic while overall the factor model is not particularly relatable with Fama-French factors or industries. As for time constraint, the factor analysis was exploratory, more time series depicting different market characteristics could potentially shine more light on the factors. However, interpretability for Machine Learning models remains a very open topic of research. On the modeling side, as the most recent papers suggest, hierarchical factors could be further explored to improve model performance.

References

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. **arXiv preprint arXiv:1710.10903**, 2017. pages 5

Zikai Wei, Bo Dai, and Dahua Lin. E2eai: End-to-end deep learning framework for active investing. **arXiv preprint arXiv:2305.16364**, 2023. pages 9