BerkeleyHaas

230ZA: Deep Learning and Applications I

University of California Berkeley: Haas School of Business

Master of Financial Engineering

Paper Review: FactorVAE by Duan et Al.

Authors:

Di Venti, Matteo Mario; Ng, Anson; Pan, Jianming; Shi, Jiacheng David; Vialard, Victor

Date: October 25, 2023

1 Introduction

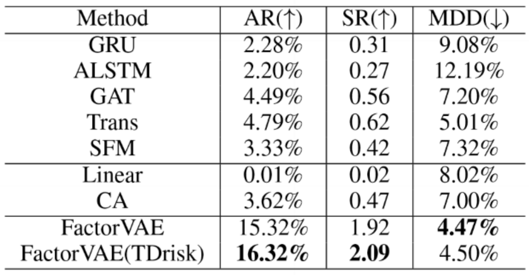
In the paper 'FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder for Predicting Cross-Sectional Stock Returns', Yitong Duan, LeiWang, Qizhong Zhang, Jian Li combine dynamic factor models with Variational Auto-encoders (VAE) to predict the cross-section of stock returns. In their approach, latent factors prior distribution are derived through a prior-posterior learning method to guide learning from the highly noisy data. They provide evidence that such model structure provides strong excessive returns and outperform various benchmark dynamic factor and deep learning models.

The model factor learning employs an encoder-decoder structure. During training, it uses the full training sample to identify the best factors for reproducing those returns. Subsequently, the authors develop a predictor using only past data to predict optimal factors. In the forecasting stage, only the predictor and decoder are used, ensuring there's no inadvertent use of future data. As a result, the VAE model produces probabilistic returns, which are also beneficial for risk assessment.

As can be seen in Figures 7 13, FactorVAE outperforms all benchmark models both on return prediction, robustness and portfolio investments. Furthermore, as can be seen from the paper results in Table 1, combining the FactorVAE predictions with TDRisk, a model considering risk aversion in stock-picking, the portfolio performance reaches a 16.32% return and 2.09 Sharpe ratio.

In the remainder of this report, we will talk about the literature surrounding Duan et Al's paper, cover our current implementation of the paper together with future possible extensions and provide a critical review of their paper covering positives, negatives, and model improvements.

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



Method	AR(↑)	SR(↑)	MDD(↓)
GRU	2.28%	0.31	9.08%
ALSTM	2.20%	0.27	12.19%
GAT	4.49%	0.56	7.20%
Trans	4.79%	0.62	5.01%
SFM	3.33%	0.42	7.32%
Linear	0.01%	0.02	8.02%
CA	3.62%	0.47	7.00%
FactorVAE	15.32%	1.92	**4.47%**
FactorVAE(TDrisk)	**16.32%**	**2.09**	4.50%

Figure 1: Duan et Al. original result: 3: The portfolio performance (Absolute Returns, Sharpe Ratio and Maximum Drawdown) relative to the benchmark. ↑ means the larger the better while ↓ means the smaller the better.

2 Literature Review

Duan et Al. start building their model from the factor models literature. In their paper, they cite the models of Fama and French [2021] and Daniel et al. [2020] as evidence of capacity to predict cross-sectional stock returns. In particular, the authors adopt the dynamical factor models (DFM) framework which enables factor loadings to change through time. Some classical references about the technique are Sargent and Sims [1977] for macroeconomic business cycle analysis, Stock and Watson [2002] for using PCA to extract factors and Forni et al. [2000] for generalized factor models.

Some more recent application of the models to finance are the dynamic trading model of Gârleanu and Pedersen [2013] that uses dynamic factors to optimally balance between the current optimal Markowitz portfolio and the expected Markowitz portfolio in the future. While for the extraction of factors, statistical methods like PCA have been refined by various current papers. Pelger [2019] builds latent diffusion and jump factors for high frequency data and Lettau and Pelger [2020] put forward the RP-PCA method to better capture the latent factors. A further PCA method based on instrumental variables is put forward by Kelly et al. [2019] and Gu et al. [2021] utilise DFM together with ML to extract the factors. This last paper is particularly important in the context of the Duan et Al. paper as it employs a conditional autoencoder for the latent factors, thus being a direct comparable for the need of Variational Autoencoders.

The use of machine learning into finance is becoming increasingly important and thus sparking a fertile literature. Prediction of stock returns has been done also through alternative datasets like news (Hu et al. [2018]), social media (Xu and Cohen [2018]), knowledge graphs (Cheng et al. [2020]). Using more technical data, Duan et Al. cite as benchmarks the models of Zhang et al. [2017] using a variant of LSTM. Finally the method used, the Variational Autoencoder has been put forward by the seminal paper of Kingma and Welling [2013].

The paper of Duan et Al, although relatively recent, has already been cited a number of times. In particular, Wei et al. [2023a] showcase it as an example of 'Deep Factors' in their example of End-to-End Deep Learning Framework. Ye et al. [2023] develop in contrast a deep learning method based on MLP-Mixers. Finally, Wei et al. [2023b] use it as a benchmark, together with the Conditional Autoencoder of Gu et al. [2021] to showcase their very promising advancement that adds regime-based factors and hierarchical latent space to the VAE configuration.

3 Implementation - Victor

3.1 Dataset

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 CSI500, which includes the top 300 stocks traded on the Shanghai Stock Exchange and the Shenzhen Stock Exchange. We use different time periods between training, testing and validation to reduce information leakage between the datasets. Summary statistics can found on table 2. We use the off-the-shelf features provided by the QLib (Alpha158)[2], which are 158 technical feature extracted from price/volume data. For our time series model, we further use a selected set of 20 features.

[1]https://github.com/microsoft/qlib
[2]The full description of the features can be found on GitHub - QLib.

Parameter	Value
Stock universe	CSI300 Index
Train period	2017 - 2017
Validation period	2018 - 2019
Test period	2020 - 09/2022
Number of features	158

Figure 2: Dataset description

3.2 Model

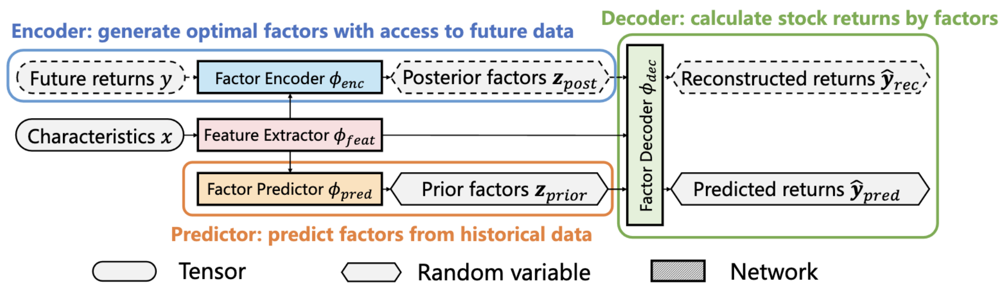

Encoder: generate optimal factors with access to future data
Decoder: calculate stock returns by factors
Future returns y
Factor Encoder ϕ_{enc}
Posterior factors z_{post}
Reconstructed returns $\hat{y}_{rec}$
Characteristics x
Feature Extractor ϕ_{feat}
Factor Decoder ϕ_{dec}
Factor Predictor ϕ_{pred}
Prior factors z_{prior}
Predicted returns $\hat{y}_{pred}$
Predictor: predict factors from historical data
Tensor
Random variable
Network

Figure 3: Overall framework of FactorVAE

Figure 3 provides an overview of our model.

3.2.1 Feature pre-processing

Our model first consists of a *feature encoder* that extracts stock latent features in a sequential manner, using a *Gate Recurrent Unit (GRU)*. GRUs have a gating mechanism that mitigates vanishing gradient problems and captures long-term dependencies more effectively than traditional RNNs while being computationally efficient with fewer parameters. Before being fed to the GRU, the input features are also fed to a *Layer Normalization* and a linear layer to improve stability of our training.

3.2.2 Model description

In the training phase, the posterior factors are computed using the *Posterior Encoder*, which outputs the mean and the variance of our posterior distribution, under the assumption that are sampled using a Gaussian Distribution. Note that the posterior factors not only have access to the stock latent features, but also the future stock returns. Figure 4 provides more

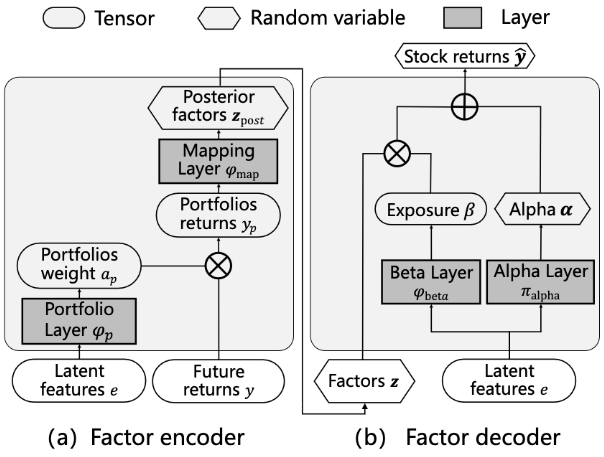


Figure 4: Encoder-Decoder architecture

information about this encoding process. Because the number of individual stocks is large, we also construct a set of portfolios, dynamically re-weighted based on the latent features to reduce the dimensionality of our problem. The stock returns are then predicted using these new factors and the latent features that were previously produced. The alpha and beta layers, which corresponds to the idiosyncratic returns and the factor exposures, are computed by the decoder. The alpha are assumed to follow a Gaussian distribution, and the factor exposures are computed from using a linear mapping. Thus, because the latent features are assumed to follow a Gaussian distribution independent from the idiosyncratic returns, the predicted stock returns are also assumed to follow a Gaussian distribution.
The prior factors are computed using the factor predictor, which computes the distribution of the prior factors using a multi-head global attention mechanism. This is desirable since our goal is to extract the diverse global representations of the market in parallel, and extract factors that correspond to diverse risk premiums of the market. The distribution network (formally fully connected layers) is then used to compute the parameters (mean and variance) of the Gaussian distributions associated to each prior.

3.2.3 Training

$$L(x,y) = -\frac{1}{N}\sum_{i=1}^{N} \log P_{\phi_{\text{dec}}}\left(\widehat{y}_{\text{rec}}^{(i)} = y^{(i)} \mid x, \mathbf{z}_{\text{post}}\right) + \gamma \cdot \text{KL}\left(P_{\phi_{\text{enc}}}(z \mid x,y), P_{\phi_{\text{pred}}}(z \mid x)\right) \tag{1}$$

During training, out model is given access to future returns as shown on figure 3, and our goal is to minimize the Kullback–Leibler divergence between the distribution of the prior and posterior factors. Formally, the RHS (right hand side) of this equation is used to maximize the log-likelyhood of generating real data, while the LHS (left hand side) minimizes the difference between the real and estimated distributions. In Variational Bayesian methods, this equation of known as the ELBO (Evidence Lower Bound), but its derivation is out of the scope of this study. Note that we use the Forward KL divergence, which ensures that the optimized variational distribution covers the entire posterior distribution.

3.2.4 Inference

During prediction, our model predicts stock returns only using the predictor and the decoder, without any future information leakage.

4 Positive Breakthrough

4.1 Multi-Head Global Attention

As we are passing time-series of latent features into the Feature Predictor, using simple multi-layer perceptron cannot fully utilize the extracted features as there maybe cross dependencies within the set of features, returns and stocks. Therefore, using the attention mechanism can overcome these shortfalls. With the multi-head global attention as the first layer of the Feature Predictor, it can learn to dynamically weigh the feature importance and capture different time range dependencies.

4.2 Posterior Prior Learning

One of the most notable traits that distinguish the FactorVAE model is the posterior-prior learning mechanism. For most of the proposed models in the industry, they tend to map features directly to actual returns via different neural network architectures with the goal to lower the loss between the actual return and the predicted returns. Although these mod-

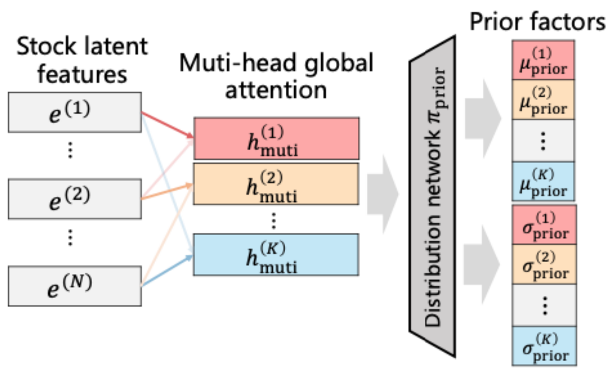

Stock latent features
$e^{(1)}$
$e^{(2)}$
$e^{(N)}$
Muti-head global attention
$h^{(1)}_{\text{muti}}$
$h^{(2)}_{\text{muti}}$
$h^{(K)}_{\text{muti}}$
Distribution network π_{prior}
Prior factors
$\mu^{(1)}_{\text{prior}}$
$\mu^{(2)}_{\text{prior}}$
$\mu^{(K)}_{\text{prior}}$
$\sigma^{(1)}_{\text{prior}}$
$\sigma^{(2)}_{\text{prior}}$
$\sigma^{(K)}_{\text{prior}}$

Figure 5: Multi-Head Global Attention

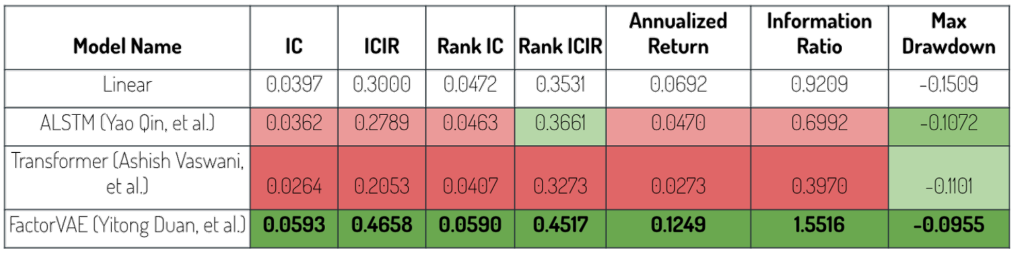
els can capture non-linearity between features and returns, they neglected the fact that the signal-to-noise ratio is extremely low in stock returns.

Therefore, using a probabilistic approach is beneficiary in this setting. Instead of mapping the returns directly, the Variational Autoencoder guides the Feature Predictor in learning the correct factors' distribution in the market and can capture both the inherent structure of the data and the variability introduced by noise. If you look at the loss function of FactorVAE, you can notice that the predicted returns and actual returns are not compared directly. Instead, the prior and posterior factor distributions are connected with the Kullback–Leibler divergence, to enforce a similar distribution, which is assumed to be a Gaussian distribution as noise in returns are widely regarded as normal.

$$L(x,y) = -\frac{1}{N}\sum_{i=1}^{N} \log P_{\phi_{\text{dec}}}\left(\hat{\boldsymbol{y}}^{(i)}_{\text{rec}} = y^{(i)} \mid x, \mathbf{z}_{\text{post}}\right) + \gamma \cdot \mathrm{KL}\left(P_{\phi_{\text{enc}}}(z \mid x, y), P_{\phi_{\text{pred}}}(z \mid x)\right)$$

Figure 6: Loss Function of FactorVAE

Although our results on ALSTM and Transformer does not align with what the paper is proposing, we agree on FactorVAE delivering superior results comparing to the baseline Linear Model. Using attention and encoder-decoder structure does not improve the performance independently but with Variational Autoencoder, it enabled the model to learn a smoother and more generalized representations of the stock returns.



Model Name	IC	ICIR	Rank IC	Rank ICIR	Annualized Return	Information Ratio	Max Drawdown
Linear	0.0397	0.3000	0.0472	0.3531	0.0692	0.9209	-0.1509
ALSTM (Yao Qin, et al.)	0.0362	0.2789	0.0463	0.3661	0.0470	0.6992	-0.1072
Transformer (Ashish Vaswani, et al.)	0.0264	0.2053	0.0407	0.3273	0.0273	0.3970	-0.1101
FactorVAE (Yitong Duan, et al.)	**0.0593**	**0.4658**	**0.0590**	**0.4517**	**0.1249**	**1.5516**	**-0.0955**

Figure 7: Comparison between different models

5 Shortcoming of the Proposed Framework

5.1 Economical Linkage of Latent Factors

Since the proposed framework is rather black-boxed and we used Microsoft's Qlib Alpha158 features which are constructed from price and volume, there is actually no visibility of the characteristic of the latent factors being learnt. It could be possible that our superior results is just purely exposure towards momentum in an inefficient market and will not be transferable to an efficient market. We can already see that the model experienced a significant drawdown and a higher error between the predicted and actual volatility in 2020 when COVID happened. When undergoing regime switching, where retraining is not feasible due to the lack of new data, without good interpret-ability of the latent factors that can accurately decompose risk, it will be difficult to react immediately to the market dynamics.

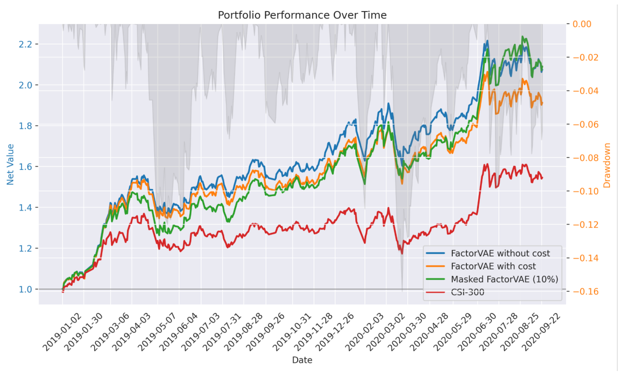


Figure 8: Performance of Different Variation of FactorVAE

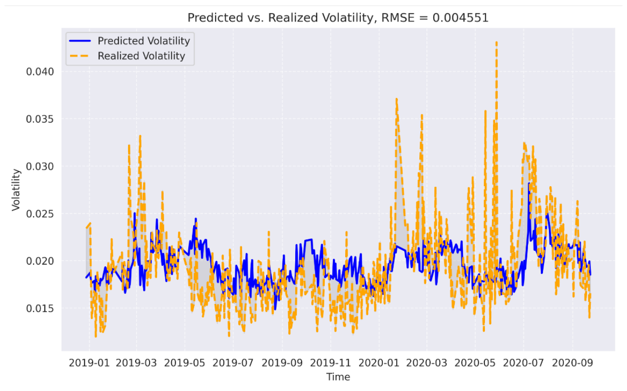


Figure 9: Predicted vs Realized Volatiltiy of FactorVAE

5.2 Complexity of the Model

As visualized below, there are 4 parts of the model to be trained, namely the Feature Extractor, Feature Predictor, Factor Encoder and Factor Decoder. Compare the other models, here the author used the Variational Autoencoder to assist the learning process of Feature Predictor, which introduced another layer of complexity. Apart from that, with the Multi-Head global attention layer, the trainable parameters of the model increased exponentially, which has 446,582 trainable parameters, and significantly hindered our training time. Within our experiment, training for 100 epochs take almost 10-hours of time.

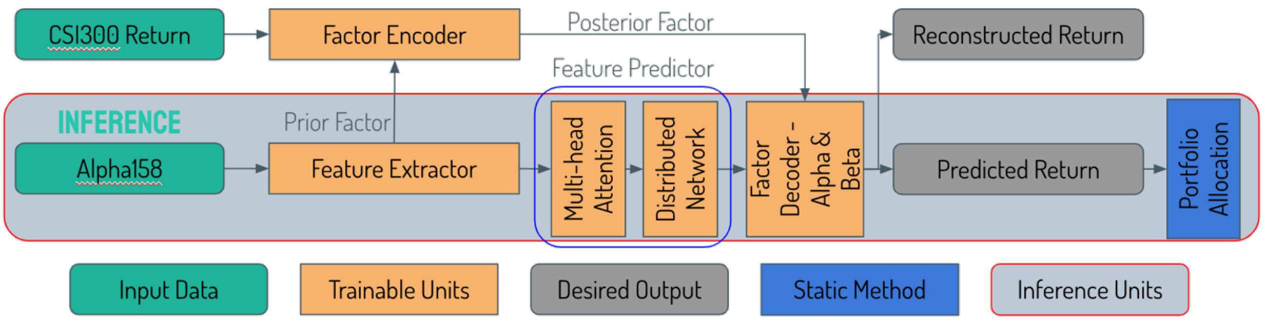
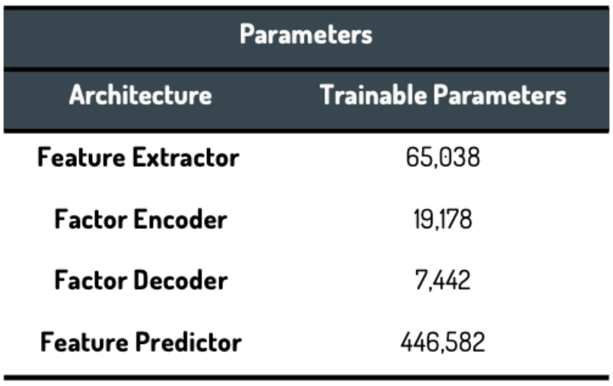
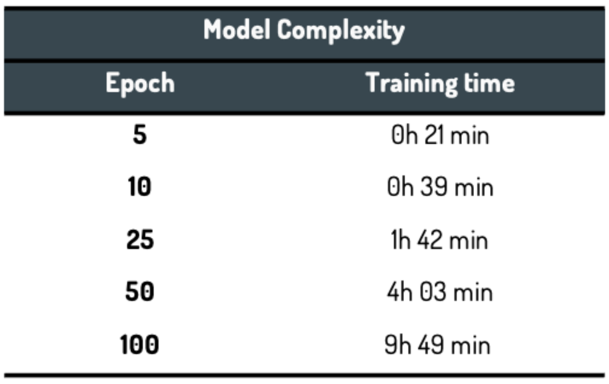


Figure 10: Visualization of FactorVAE



Parameters	
Architecture	**Trainable Parameters**
Feature Extractor	65,038
Factor Encoder	19,178
Factor Decoder	7,442
Feature Predictor	446,582

Figure 11: Trainable Parameters



Model Complexity	
Epoch	**Training time**
5	0h 21 min
10	0h 39 min
25	1h 42 min
50	4h 03 min
100	9h 49 min

Figure 12: Model Training Time

6 Potential Improvements

6.1 Parallel Computing

Typically, when dealing with highly complex models, we employ distributed computing methods to break down the entire training task into smaller subtasks, which are executed simultaneously on multiple processors known as worker nodes.

In our specific scenario, we aim to utilize Data Parallelism. This approach involves dividing the initial training batch into numerous smaller minibatches, which are then distributed among the worker nodes. Each node independently performs its computations and acquires its own set of parameters. These parameters are subsequently aggregated and updated across all nodes using either a parameter server architecture or the all-reduce algorithm.

Algorithm 1 ParallelSGD ($\{c^1, ...c^m\}$, Learning Rate η, Machines k)

```
Define $T = \lfloor m/k \rfloor$
Randomly partition the examples, giving T examples to each machine.
for $i \in \{1, ...k\}$ parallel do
    Randomly shuffle the data on machine $i$
    Initialize $\omega_{i,0}$
    for $t \in 1, ....T$ : do
        Get the $t$th example on the $i$th machine, $c^{i,t}$
        $w_{i,t} \leftarrow w_{i,t-1} - \eta \partial_\omega c^i(\omega_{i,t-1})$
    end
end
Aggregate from all computers $v = \frac{1}{k} \sum_{i=1}^{k} \omega_{i,t}$ and return $v$
```

Algorithm 2 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]
```

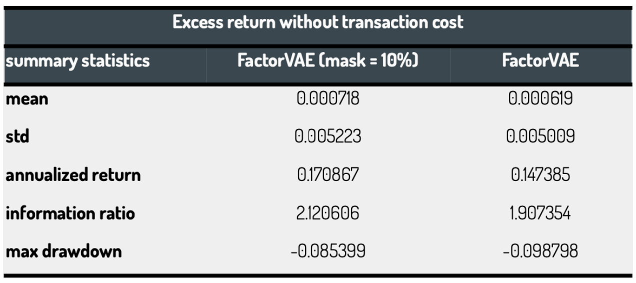
Parameter Server Architecture: In this architecture, there is a central parameter server that maintains the global model parameters. Worker nodes communicate with the parameter server to update and retrieve parameters.

All-Reduce Algorithm: The all-reduce algorithm is a collective communication operation that allows all worker nodes to efficiently synchronize their parameter updates. It involves summing or averaging the updates from all nodes and then distributing the result back to each node.

6.2 Robustness

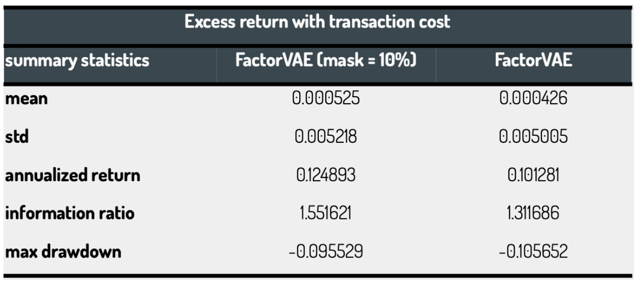
We conducted robustness tests on the model. Initially, we masked 10% of the training data by setting their features to 0 in each epoch, then we trained the model, and finally we tested the trained model on the entire CSI300 dataset.

The result of this trained model still yields consistent favorable results. This observation strongly suggests that the model is not learning specific factors unique to individual stocks but rather focusing on broader market factors.



Excess return without transaction cost		
summary statistics	FactorVAE (mask = 10%)	FactorVAE
mean	0.000718	0.000619
std	0.005223	0.005009
annualized return	0.170867	0.147385
information ratio	2.120606	1.907354
max drawdown	-0.085399	-0.098798

Figure 13: Excess return without transaction cost



Excess return with transaction cost		
summary statistics	FactorVAE (mask = 10%)	FactorVAE
mean	0.000525	0.000426
std	0.005218	0.005005
annualized return	0.124893	0.101281
information ratio	1.551621	1.311686
max drawdown	-0.095529	-0.105652

Figure 14: Excess return with transaction cost

To further assess its robustness, we plan to increase the percentage of masked stocks during training phase and subsequently test the trained model on the entire CSI300 dataset.

Moreover, a crucial aspect of our assessment is to gauge the model's performance on the S&P500 dataset. This analysis will not only provide valuable insights into the model's behavior in a more efficient market but also help us determine its robustness in terms of transfer learning. More precisely, it will enable us to investigate whether the factors identified by the model in the Chinese market are universally applicable or specific to certain geographical regions.

6.3 Economic understanding of factors

In our experiments, we discovered that the model performs optimally when we increase the number of risk factors to 60. This implies that there might be approximately 60 latent risk factors that could potentially explain most of the variance in returns within the equity markets.

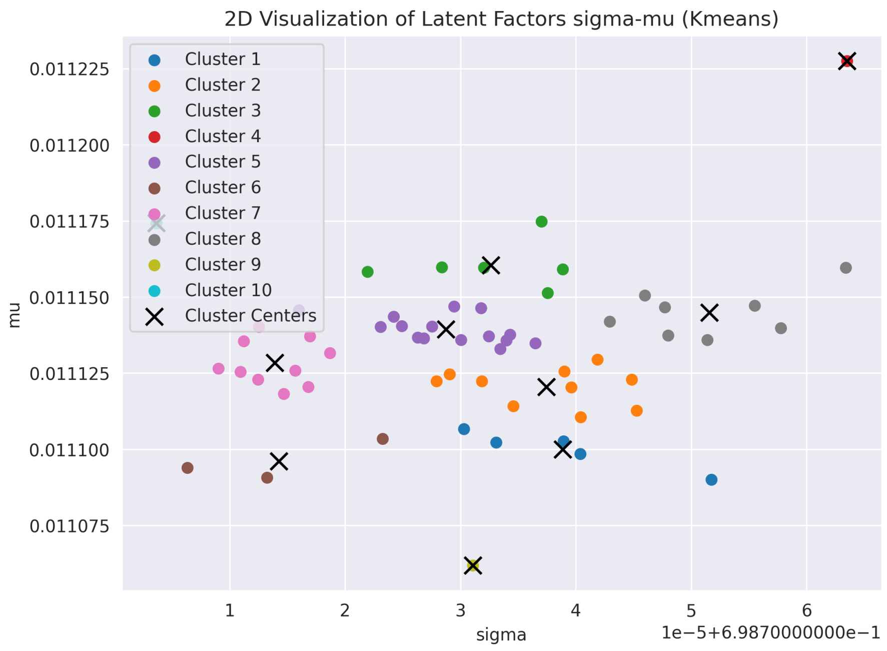


Figure 15: 2-D Visualization of latent factors

As expected, we observe a positive relationship between the risk premium's μ (expected return) and its σ (volatility).

We have grouped the risk factors into 10 clusters, each potentially associated with a distinct economic factor and accompanied by a specific economic rationale. In our future work, we plan to compare these clusters with traditional economic factors by measuring the distance in the 2D Euclidean space.

6.4 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.

7 Future works

7.1 Portfolio Construction & Asset Class

In the original paper, the authors only compare FactorVAE to the Top-k strategy. We aim to provide a thorough analysis of a variety of portfolios including Mean-Variance optimization, Minimum Variance Optimization, Risk-Parity Strategy, etc. To provide a detailed understanding of how the model performed on both return and risk prediction.

Furthermore, the idea of the underlying factor model in asset pricing is prevalent across all financial assets, including fixed income, futures, and even options. Though the factor model inside different assets is widely different, we are planning on applying this FactorVAE to different asset classes. The model of different assets shall differ according to its natural, e.g., in option pricing, the factors (Greeks) are often non-linear to the price, which is a best practise of applying the machine learning model and testing for the theorem.

7.2 Risk Model

The model adopted the traditional factor model as inspiration to propose Posterior Prior Learning, while does not count on the systematic risk of each asset. The $\mu_{prior}, \sigma_{prior}$ pairs are assumed idiosyncratic risk drawn from independent normal distribution. In a more realistic scenario, the factors are not only related to the idiosyncratic risk of the model, but also account for the systematic risk of the portfolio, i.e., we are intentionally include the risk model into the training of FactorVAE. The risk model estimates assets' variance-covariance matrix as two parts: 1) systematic risk and 2) idiosyncratic risk.

$$\Sigma = \mathbf{F}\Sigma_{\mathbf{Factor}}\mathbf{F}^{\top} + \Delta. \tag{2}$$

The systematic risk is determined by factor exposure F and the variance-covariance matrix of factor returns and the idiosyncratic risk is determined by individual stocks. We plan on learning Σ by two networks: 1) $\pi_{sys}(e) \in \mathbb{R}^{n\times n}$ systematic layer outputs factor risk matrix and 2) $\pi_{ido}(e, \mu_{prior}, \sigma_{prior})$ idiosyncratic layer outputs individual stock risk.

7.3 Generative Learning

The core idea of this paper is to learn the factor structure of the model instead of learning the return itself. In order to decode the true return into multiple factors, the paper adopted the VAE model and interpreted the hidden layer as factor. Breaking inputs into small hidden parts and reformulating them from pure noise is a typical generative learning task. However, in large-scale works, VAE shows inefficiency in capturing non-unimodal distribution likelihood function in comparison with advanced generative learning models like Diffusion or Normalizing flows. Assume the likelihood of return is simply unimodal is not practical in practice and may hinder the power of VAE. Therefore, we intentionally switch the VAE part in FactorVAE and plan on using other generative models.

References

Eugene F Fama and Kenneth R French. The value premium. **The Review of Asset Pricing Studies**, 11(1):105–121, 2021. pages 2

Kent Daniel, David Hirshleifer, and Lin Sun. Short-and long-horizon behavioral factors. **The review of financial studies**, 33(4):1673–1736, 2020. pages 2

T. J. Sargent and C. A. Sims. Business cycle modeling without pretending to have too much a priori economic theory. Technical report, Working Paper, Federal Reserve Bank of Minneapolis, 1977. pages 2

J. H. Stock and M. W. Watson. Forecasting using principal components from a large number of predictors. **Journal of the American Statistical Association**, 2002. pages 2

M. Forni, M. Hallin, M. Lippi, and L. Reichlin. The generalized dynamic factor model: Identification and estimation. **The Review of Economics and Statistics**, 2000. pages 2

Nicolae Gârleanu and Lasse Heje Pedersen. Dynamic trading with predictable returns and transaction costs. **The Journal of Finance**, 68(6):2309–2340, 2013. pages 2

Markus Pelger. Large-dimensional factor modeling based on high-frequency observations. **Journal of Econometrics**, 208(1):23–42, 2019. pages 2

Martin Lettau and Markus Pelger. Factors that fit the time series and cross-section of stock returns. **The Review of Financial Studies**, 33(5):2274–2325, 2020. pages 2

Bryan T Kelly, Seth Pruitt, and Yinan Su. Characteristics are covariances: A unified model of risk and return. **Journal of Financial Economics**, 134(3):501–524, 2019. pages 2

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Autoencoder asset pricing models. **Journal of Econometrics**, 222(1):429–450, 2021. pages 2, 3

Ziniu Hu, Weiqing Liu, Jiang Bian, Xuanzhe Liu, and Tie-Yan Liu. Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. In **Proceedings** of the eleventh ACM international conference on web search and data mining, pages 261–269, 2018. pages 3

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. pages 3

Dawei Cheng, Fangzhou Yang, Xiaoyang Wang, Ying Zhang, and Liqing Zhang. Knowledge graph-based event embedding framework for financial quantitative investments. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 2221–2230, 2020. pages 3

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. pages 3

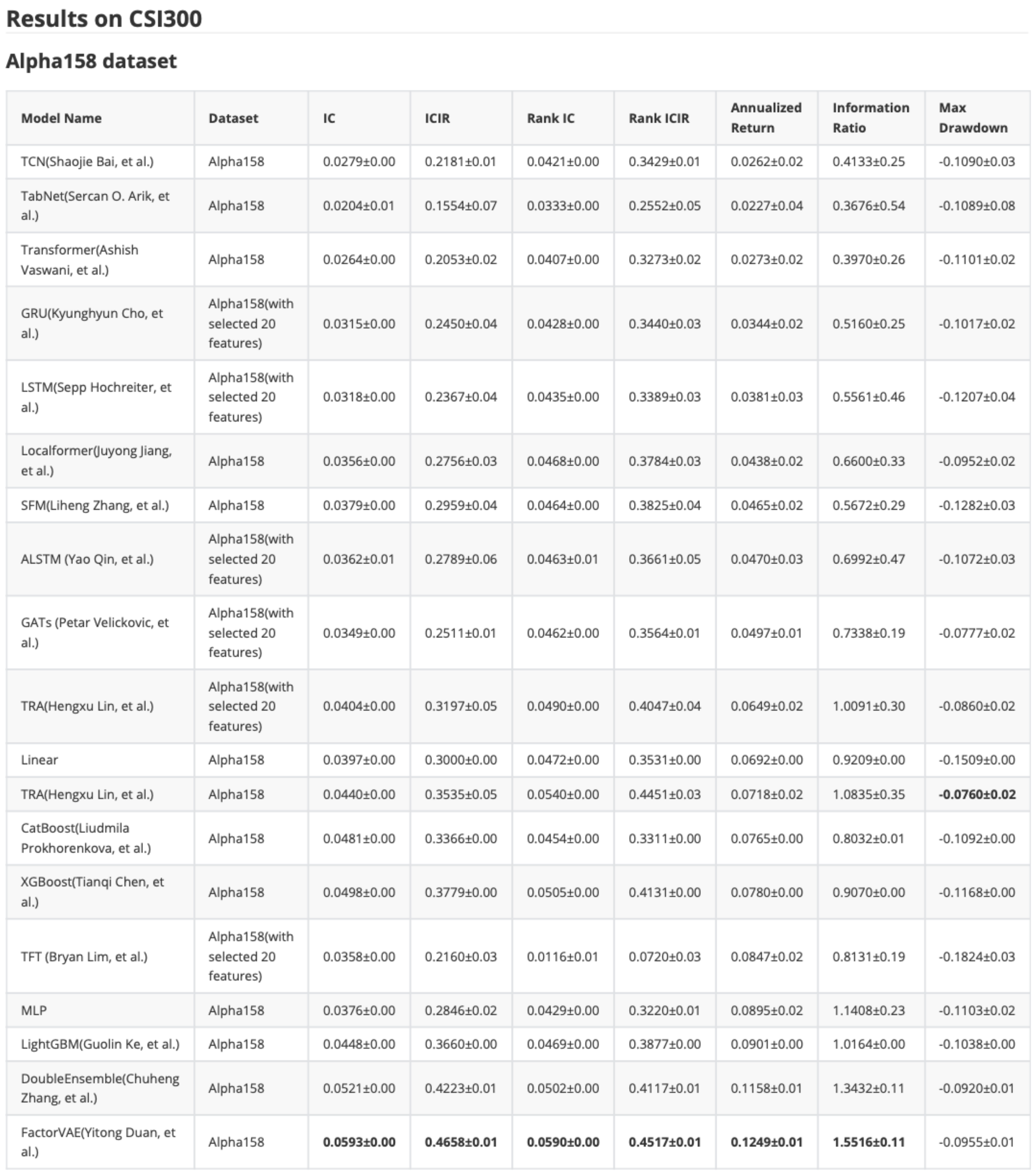
Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013. pages 3

Zikai Wei, Bo Dai, and Dahua Lin. E2eai: End-to-end deep learning framework for active investing. arXiv preprint arXiv:2305.16364, 2023a. pages 3

Junyi Ye, Jingyi Gu, Ankan Dash, Fadi P Deek, and Guiling Grace Wang. Prediction with time-series mixer for the s&p500 index. In 2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW), pages 20–27. IEEE, 2023. pages 3

Zikai Wei, Anyi Rao, Bo Dai, and Dahua Lin. Hirevae: An online and adaptive factor model based on hierarchical and regime-switch vae. arXiv preprint arXiv:2306.02848, 2023b. pages 3

8 Appendix A



Results on CSI300

Alpha158 dataset

Model Name	Dataset	IC	ICIR	Rank IC	Rank ICIR	Annualized Return	Information Ratio	Max Drawdown
TCN(Shaojie Bai, et al.)	Alpha158	0.0279±0.00	0.2181±0.01	0.0421±0.00	0.3429±0.01	0.0262±0.02	0.4133±0.25	-0.1090±0.03
TabNet(Sercan O. Arik, et al.)	Alpha158	0.0204±0.01	0.1554±0.07	0.0333±0.00	0.2552±0.05	0.0227±0.04	0.3676±0.54	-0.1089±0.08
Transformer(Ashish Vaswani, et al.)	Alpha158	0.0264±0.00	0.2053±0.02	0.0407±0.00	0.3273±0.02	0.0273±0.02	0.3970±0.26	-0.1101±0.02
GRU(Kyunghyun Cho, et al.)	Alpha158(with selected 20 features)	0.0315±0.00	0.2450±0.04	0.0428±0.00	0.3440±0.03	0.0344±0.02	0.5160±0.25	-0.1017±0.02
LSTM(Sepp Hochreiter, et al.)	Alpha158(with selected 20 features)	0.0318±0.00	0.2367±0.04	0.0435±0.00	0.3389±0.03	0.0381±0.03	0.5561±0.46	-0.1207±0.04
Localformer(Juyong Jiang, et al.)	Alpha158	0.0356±0.00	0.2756±0.03	0.0468±0.00	0.3784±0.03	0.0438±0.02	0.6600±0.33	-0.0952±0.02
SFM(Liheng Zhang, et al.)	Alpha158	0.0379±0.00	0.2959±0.04	0.0464±0.00	0.3825±0.04	0.0465±0.02	0.5672±0.29	-0.1282±0.03
ALSTM (Yao Qin, et al.)	Alpha158(with selected 20 features)	0.0362±0.01	0.2789±0.06	0.0463±0.01	0.3661±0.05	0.0470±0.03	0.6992±0.47	-0.1072±0.03
GATs (Petar Velickovic, et al.)	Alpha158(with selected 20 features)	0.0349±0.00	0.2511±0.01	0.0462±0.00	0.3564±0.01	0.0497±0.01	0.7338±0.19	-0.0777±0.02
TRA(Hengxu Lin, et al.)	Alpha158(with selected 20 features)	0.0404±0.00	0.3197±0.05	0.0490±0.00	0.4047±0.04	0.0649±0.02	1.0091±0.30	-0.0860±0.02
Linear	Alpha158	0.0397±0.00	0.3000±0.00	0.0472±0.00	0.3531±0.00	0.0692±0.00	0.9209±0.00	-0.1509±0.00
TRA(Hengxu Lin, et al.)	Alpha158	0.0440±0.00	0.3535±0.05	0.0540±0.00	0.4451±0.03	0.0718±0.02	1.0835±0.35	**-0.0760±0.02**
CatBoost(Liudmila Prokhorenkova, et al.)	Alpha158	0.0481±0.00	0.3366±0.00	0.0454±0.00	0.3311±0.00	0.0765±0.00	0.8032±0.01	-0.1092±0.00
XGBoost(Tianqi Chen, et al.)	Alpha158	0.0498±0.00	0.3779±0.00	0.0505±0.00	0.4131±0.00	0.0780±0.00	0.9070±0.00	-0.1168±0.00
TFT (Bryan Lim, et al.)	Alpha158(with selected 20 features)	0.0358±0.00	0.2160±0.03	0.0116±0.01	0.0720±0.03	0.0847±0.02	0.8131±0.19	-0.1824±0.03
MLP	Alpha158	0.0376±0.00	0.2846±0.02	0.0429±0.00	0.3220±0.01	0.0895±0.02	1.1408±0.23	-0.1103±0.02
LightGBM(Guolin Ke, et al.)	Alpha158	0.0448±0.00	0.3660±0.00	0.0469±0.00	0.3877±0.00	0.0901±0.00	1.0164±0.00	-0.1038±0.00
DoubleEnsemble(Chuheng Zhang, et al.)	Alpha158	0.0521±0.00	0.4223±0.01	0.0502±0.00	0.4117±0.01	0.1158±0.01	1.3432±0.11	-0.0920±0.01
FactorVAE(Yitong Duan, et al.)	Alpha158	**0.0593±0.00**	**0.4658±0.01**	**0.0590±0.00**	**0.4517±0.01**	**0.1249±0.01**	**1.5516±0.11**	-0.0955±0.01

Figure 16: Models' performance on CSI300 using Alpha158 dataset