Reimplementation: Autoencoder Asset Pricing Models

Zhiyuan Wu
SEM, Tsinghua University
2022011429
wuzhiyua22@mails.tsinghua.edu.cn

Yuhan Wang
Zhili College, Tsinghua University
2022012241
w-yh22@mails.tsinghua.edu.cn

Abstract

We replicate the core experiments of Gu, Kelly, and Xiu's *Autoencoder Asset Pricing Models*, adhering to their data, no-arbitrage constraints, and evaluation protocols. Because the original study omits several tuning-parameter details, our reproduced conditional autoencoder (CA) results differ marginally in magnitude, yet the qualitative rankings and statistical significance are preserved. Building on this baseline, we introduce a probabilistic extension—a Variational Autoencoder (VAE) with a Gaussian latent prior and KL regularisation—that jointly learns the mean and uncertainty of latent factors. In extensive out-of-sample tests, the VAE attains higher total R^2 than both the reproduced CA architectures and leading non-neural benchmarks (PCA and IPCA), while maintaining comparable predictive R^2. These findings demonstrate that probabilistic latent-factor models can enhance explanatory power in cross-sectional asset pricing without sacrificing forecast accuracy.

1 introduction

Traditional linear factor models (e.g. CAPM; Fama–French three and five factor) have driven empirical asset-pricing research but rely on pre-specified structures and cannot capture complex nonlinearities. Recent work by Gu et al. (2023) embeds an autoencoder within a no-arbitrage framework, jointly estimating time-varying latent factors (f-Net) and exposures (β-Net) to improve explanatory and predictive power.

However, deterministic autoencoders lack uncertainty quantification and may overfit under extreme market conditions. To mitigate these issues, we propose a Variational Autoencoder (VAE) with a Gaussian latent prior and KL-regularization, enabling the joint learning of factor means and variances. This probabilistic approach enhances robustness, captures tail risks, and supports extreme-return forecasting.

First, we replicate the benchmark autoencoder asset-pricing model to establish a baseline. Next, we implement our variational autoencoder under the same no-arbitrage constraint. Finally, we compare both models in terms of total and predictive R^2, risk-premia decomposition, and feature-importance analysis.

2 Related Works

The earliest asset-pricing frameworks assume time-invariant loadings. The Capital Asset Pricing Model (CAPM; Treynor 1961; Sharpe 1964; Lintner 1975) posits a single market factor driving cross-sectional returns. Fama and French (1992) augment CAPM with size and value factors, yielding the celebrated three-factor specification that remains a benchmark in empirical studies.

To allow exposures to vary with firm characteristics, Kelly et al. (2019) propose Instrumented Principal Components Analysis (IPCA), which linearly maps characteristics into time-varying loadings. Gu,

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Kelly, and Xiu (2021) replace the linear mapping with a conditional autoencoder, capturing nonlinear return dynamics and improving both total and predictive R^2. Despite these gains, deterministic autoencoders can overfit noisy market data.

Variational Autoencoders (VAE; Kingma Welling 2013) introduce a Gaussian prior and KL regularization to learn distributions over latent representations. Sequential extensions (Chung et al. 2015; Fraccaro et al. 2016) have been adapted to finance for stochastic-volatility modeling (Luo et al. 2018) and joint text–price prediction (Xu Cohen 2018). In this paper, we integrate a VAE into a no-arbitrage factor framework to enhance noise robustness and quantify uncertainty in latent factors.

3 Approach

This section systematically introduces the five types of asset pricing models used in this article: Fama French benchmark model (FF), principal component analysis model (PCA), instrumental variable principal component analysis model (IPCA), autoencoder asset pricing model (AE), and its variant - variational autoencoder asset pricing model (VAE). Among them:

- $r_{i,t}$ indicates the excess return of the first asset during the period
- f_t represents factor vector
- $\beta_{i,t}$ indicates the corresponding factor exposure
- $z_{i,t}$ for features (feature dimension)
- The excess return matrix of the entire sample of assets is recorded as $RN \times T$

3.1 Fama-French model

The Fama French factor model assumes that excess returns are explained by a few observable cross-sectional risk factors:

$$r_{i,t} = \alpha_i + \beta_i^T f_t + \epsilon_{i,t}$$

$$f_t = (MKT_t, SMB_t, HML_t, CMA_t, RMW_t, UMD_t)^T$$

Among them, α_i is the price error (theoretically zero) and $\epsilon_{i,t}$ is the specific shock. Stack all assets in matrix form

$$R = \alpha I + BF + U$$

B($N \times K$) represents exposure matrix of constants, F($K \times T$) is aggregation factor time series. B is estimated through cross-sectional OLS regression, while F is directly taken from publicly available data.

3.2 PCA

Unlike the FF model that relies on observable factors, PCA considers factor f_t as a latent variable and assumes that factor loading β remains constant during the sample period. If the excess return vector of each cross-section is $r_t \in R^N$, the latent factor model can be written as

$$r_t = \beta f_t + \epsilon_t \quad t = 1, 2, \ldots, T$$

Among them, $\beta \in R^{N \times K}$is a fixed factor loading matrix (the i-th row corresponds to the factor exposure of asset i), $f_t \in R^K$ is the latent factor vector, and $\epsilon_t \in R^N$ is the specific shock term.

To estimate β, first construct a sample cross-sectional covariance matrix:

$$S = \frac{1}{T} \Sigma_{t=1}^T r_t r_t^T$$

Under the additional orthogonalization constraint $\beta^T \beta = NI_K$, the maximum factor explains the variance of returns:

$$tr(\beta^T S \beta)$$

It is the classic Rayleigh Ritz problem, whose optimal solution is given by the K eigenvectors $v_1, v_2, \cdots, v_k$ with the maximum K. So, the estimation of the load matrix is:

$$\hat{\beta} = \sqrt{N}[v_1, v_2, \cdots, v_k]$$

After obtaining $\hat{\beta}$, the factor time series f_t can be estimated period by period using the least squares method. Specifically, given $\hat{\beta}$, the residual vector is defined as:

$$u_t = r_t - \hat{\beta} f_t$$

Therefore, minimizing the sum of squared residuals is equivalent to solving:

$$(\widehat{\beta^T}\hat{\beta})f_t = \widehat{\beta^T} r_t$$

From this, the closed form solution of the factor can be obtained:

$$\hat{f}_t = (\widehat{\beta^T}\hat{\beta})^{-1}\widehat{\beta^T} r_t$$

By substituting this estimate into the model, the residual u_t for each period can be obtained, and then the total sum of squared residuals can be calculated

$$\Sigma_{t=1}^T ||u_t||^2 = \Sigma_{t=1}^T ||r_t - \hat{\beta}\hat{f}_t||^2$$

This quantity is minimized under the solution obtained from the feature decomposition of the factor load.

3.3 IPCA

In the IPCA model, we no longer assume that the factor loading Γ is time invariant, but allow it to vary with the previous instrumental variable Z_{t-1}. Let the dependent variable for the t-th period be $r_t \in R^N$ and the instrumental variable matrix be $Z_{t-1} \in R^{N \times L}$, and define:

$$x_t = \frac{1}{N} Z_{t-1}^T r_t$$

The objective function of IPCA can be written as:

$$\frac{1}{T}\Sigma_{t=1}^T tr\{(\Gamma^T[\frac{1}{N}Z_{t-1}Z_{t-1}^T]\Gamma)^{-1}\Gamma^T(x_t x_t^T)\Gamma\}$$

According to the iterative scheme proposed by KPS, given the initial load Γ_{old}^T, first calculate the estimated factor for the t-th period:

$$\hat{f}_t = (\Gamma_{old}^T W_{t-1}\Gamma_{old})^{-1} x_t \qquad W_{t-1} = \frac{1}{N} Z_{t-1} Z_{t-1}^T$$

then minimize $\Sigma_{t=1}^T |||r_t - Z_{t-1}\Gamma\hat{f}_t||^2$ with $\hat{f}_t$ unchanged. By taking the first derivative, we obtain:

$$\Sigma_{t=1}^T Z_{t-1}^T Z_{t-1}\Gamma\hat{f}_t\widehat{f_t^T} = \Sigma_{t=1}^T Z_{t-1}^T y_t \widehat{f_t^T}$$

The updated solution is

$$vec\Gamma_{new} = \{\sigma_{t=1}^T(\hat{f}_t\widehat{f_t^T} \otimes Z_{t-1}^T Z_{t-1})\}^{-1}\Sigma_{t=1}^T(\hat{f}_t \otimes Z_{t-1}^T)r_t$$

iterate it until convergence and we can get the IPCA estimation.

3.4 AE

AE integrates factor extraction and exposure prediction into a neural network framework:

$$r_{i,t} = \beta_{i,t-1} f_t + u_{i,t}, \qquad \beta_{i,t-1} = G_\theta(z_{i,t-1}), \quad f_t = H_\phi(x_t)$$

Among them, G_θ and H_ϕ are MLP. The first layer input of the factor network adopts feature management combination x_t to reduce dimensionality; The factor output layer maintains linearity, ensuring that f_t can be interpreted as portfolio returns. Minimize training objectives:

$$L(\theta, \phi) = \frac{1}{NT}\Sigma(r_{i,t} - G_\theta(z_{i,t-1})^T H\phi(x_t))^2 + \lambda|\theta|_1$$

And cooperate with batch normalization, Adam optimization, and early stopping techniques to avoid overfitting. CAE constructs four variants of CA0-CA3 at a depth of G_θ, and as the depth increases, β can describe more complex nonlinear relationships.

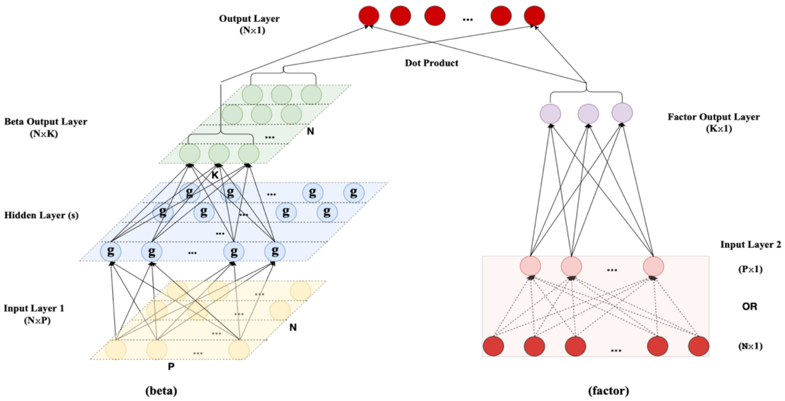


Figure 1: Graph of Autoencoder

3.5 Variational Autoencoder

We introduce FactorVAE, a variational autoencoder specifically designed to extract robust latent factors from noisy financial market data. Our model adopts the classical encoder–decoder architecture of a VAE, but interprets the latent variables as financial factors. The encoder acts as an oracle, extracting posterior factors $\mathbf{z}_{\text{post}}$ from future returns $\mathbf{y}$ and intermediate latent features $\mathbf{e}$. The decoder then reconstructs the returns using these factors.

Generative Assumption. Each observation $x \in \mathbb{R}^{94}$ is assumed to be generated from latent factors $z \in \mathbb{R}^K$. The encoder $q_\phi(z \mid x)$ outputs posterior factors z_{post}, while the decoder $p_\theta(x \mid z_{\text{post}})$ reconstructs the data. A standard normal prior $p(z) = \mathcal{N}(0, I)$ regularises the encoder; new samples are obtained via $z_{\text{prior}} \sim p(z)$ and $\hat{x} \sim p_\theta(x \mid z_{\text{prior}})$.

Feature Extractor. A shallow multilayer perceptron maps raw inputs to latent features,

$$e = \varphi_{\text{ext}}(x) \in \mathbb{R}^H.$$

Factor Encoder. Given e and future returns $y \in \mathbb{R}^N$, the encoder predicts

$$[\mu_{\text{post}}, \sigma_{\text{post}}] = \varphi_{\text{enc}}(y, e), \qquad z_{\text{post}} \sim \mathcal{N}\big(\mu_{\text{post}}, \operatorname{diag}(\sigma^2_{\text{post}})\big),$$

with $\mu_{\text{post}}, \sigma_{\text{post}} \in \mathbb{R}^K$.

Factor Decoder. The decoder is decomposed into *alpha* and *beta* layers:

$$\begin{aligned}
h_\alpha^{(i)} &= \operatorname{LeakyReLU}\big(W_\alpha e^{(i)} + b_\alpha\big),\\
\mu_\alpha^{(i)} &= W_{\alpha\mu} h_\alpha^{(i)} + b_{\alpha\mu},\\
\sigma_\alpha^{(i)} &= \operatorname{Softplus}\big(W_{\alpha\sigma} h_\alpha^{(i)} + b_{\alpha\sigma}\big), \qquad (10)\\
\beta_y^{(i)} &= \varphi_{\text{beta}}\big(e^{(i)}\big) = W_\beta e^{(i)} + b_\beta. \qquad (11)
\end{aligned}$$

Because α and z are independent Gaussians, the decoder output is Gaussian:

$$\hat{y}^{(i)} \sim \mathcal{N}\big(\mu_y^{(i)}, [\sigma_y^{(i)}]^2\big), \qquad \begin{cases} \mu_y^{(i)} = \mu_\alpha^{(i)} + \displaystyle\sum_{k=1}^{K} \beta^{(i,k)} \mu_z^{(k)}, \\ \sigma_y^{(i)} = \Big([\sigma_\alpha^{(i)}]^2 + \displaystyle\sum_{k=1}^{K} (\beta^{(i,k)})^2 [\sigma_z^{(k)}]^2\Big)^{1/2}. \end{cases} \tag{12}$$

Prior–Posterior Learning. To improve robustness, a *factor predictor* φ_{pred} is trained solely on historical data:

$$[\mu_{\text{prior}}, \sigma_{\text{prior}}] = \varphi_{\text{pred}}(e), \qquad z_{\text{prior}} \sim \mathcal{N}\big(\mu_{\text{prior}}, \operatorname{diag}(\sigma^2_{\text{prior}})\big). \tag{13}$$

At inference, the fixed decoder processes z_{prior}, thus avoiding future-information leakage.

Objective Function. The loss comprises a reconstruction term and a KL alignment term:

$$L(x, y) = -\frac{1}{N} \sum_{i=1}^{N} \log P_{\varphi_{\text{dec}}}(\hat{y}_{\text{rec}}^{(i)} = y^{(i)} \mid x, z_{\text{post}}) + \gamma \, \text{KL}\Big(P_{\varphi_{\text{enc}}}(z \mid x, y) \parallel P_{\varphi_{\text{pred}}}(z \mid x)\Big), \quad (17)$$

where $\gamma > 0$ controls the trade–off between data reconstruction and prior–posterior consistency.

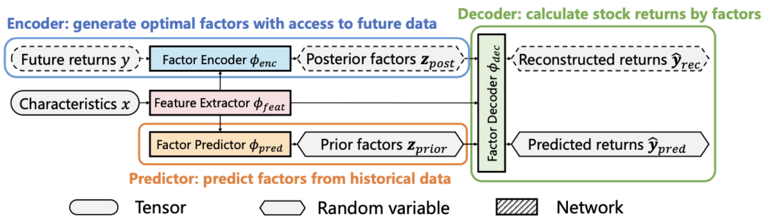

Encoder: generate optimal factors with access to future data
Decoder: calculate stock returns by factors
Future returns y
Factor Encoder ϕ_{enc}
Posterior factors $\boldsymbol{z}_{post}$
Factor Decoder ϕ_{dec}
Reconstructed returns $\hat{\boldsymbol{y}}_{rec}$
Characteristics x
Feature Extractor ϕ_{feat}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

Figure 2: Graph of Variational Autoencoder

4 Experiments

4.1 data

The income data used in this empirical study was taken from the CRSP monthly database, covering all common stocks traded on the NYSE, AMEX, and NASDAQ from March 1957 to December 2016. The risk-free rate is approximated by the one month Treasury Bill yield provided by the Ken French factor library. The feature variables are 94 stock level features provided by GKX, and their original files have been time aligned according to the public disclosure lag rule: monthly indicators use t-1 end of period observations, and quarterly and annual indicators use t-4 and t-6 end of period observations, respectively, thus minimizing forward bias to the greatest extent possible.

4.2 Feature construction and preprocessing

For each measurement month's cross-section, first fill in the missing items with zeros, and then correct the residual gap with the median value of the same month's cross-section. The missing feature vectors are then mapped to the (-1,1) interval through rank transformation:

$$x_{i,t} = 2\frac{rank(x_{i,t}) - 1}{N_t - 1} - 1$$

Among them, N_t is the number of listed stocks in the same period of month t. Based on this standardized feature set, this article further constructs 94 long short feature combinations, and takes 10% of the stocks ranked in the top and bottom of a certain feature every month to go long and short respectively. The combination factor return $\frac{1}{2}(\overline{C}_{top} - \overline{C}_{bottom})$ is used to describe the pricing strength of this feature on the cross-section.

4.3 Sample division

The sample is divided into three sections in chronological order: 1957-1974 is the training interval used for model parameter estimation; 1975-1986 is the validation interval used to monitor generalization errors and perform early stopping; The period from 1987 to 2016 is strictly defined as the out of sample testing interval. To maintain the robustness of the time-varying structure, the study adopts an annual rolling reassessment strategy: at the end of each calendar year, the training, validation, and testing windows are pushed forward by 12 months as a whole, while the window length remains unchanged and the model is re estimated.

4.4 Hyperparameter Setting

All deep networks use the Adam optimizer, with an initial learning rate set to. Implement L1 regularization for network weights, with a penalty coefficient of 1×10^{-5}, and add 0.1 Dropout after the hidden layer to alleviate overfitting. If there is no significant improvement in the validation loss after three consecutive iterations, an early stop will be triggered to ensure the stability and computational efficiency of the training process.

5 Results

This section first systematically evaluates the performance of the model at the statistical and economic levels, then explores the relationship between factor risk premium and mispricing, and reveals key driving factors through feature importance analysis.

5.1 Statistical Performance Evaluation

The performance of the model is measured by the total interpretability (Total R^2) and predictive interpretability (Predictive R^2).

$$R^2_{total} = 1 - \frac{\Sigma_{(i,t)\in OOS}(r_{i,t} - \hat{\beta_{i,t-1}}\hat{f_t})}{\Sigma_{(i,t)\in OOS} r^2_{i,t}}$$

$$R^2_{pred} = 1 - \frac{\Sigma_{(i,t)\in OOS}(r_{i,t} - \hat{\beta_{i,t-1}}\hat{\lambda_{t-1}})}{\Sigma_{(i,t)\in OOS} r^2_{i,t}}$$

Total R^2 measures the cross-sectional goodness of fit of factors on historical returns within the training interval, representing the explanatory power of quantifying the current factor implementation on the variation of cross-sectional returns; The latter focuses on the out of sample error of rolling prediction of monthly returns, which can effectively test the true predictability of the model compared to the benchmark regression of current cross-sectional average returns $\overline{r}_t$.

Table 1: Portfolio-level total R^2 (%)

Model	$K = 1$	$K = 2$	$K = 3$	$K = 4$	$K = 5$
FF	8.54	15.76	19.86	20.32	31.40
PCA	40.61	53.00	59.13	62.46	64.68
IPCA	66.96	77.11	81.98	85.68	86.99
CA$_0$	33.74	60.38	70.53	70.66	73.43
CA$_1$	60.77	69.76	73.40	72.64	74.63
CA$_2$	60.86	72.88	72.42	73.12	73.01
CA$_3$	58.04	68.96	71.19	68.83	69.20
VAE	59.78	99.72	96.81	99.74	99.06

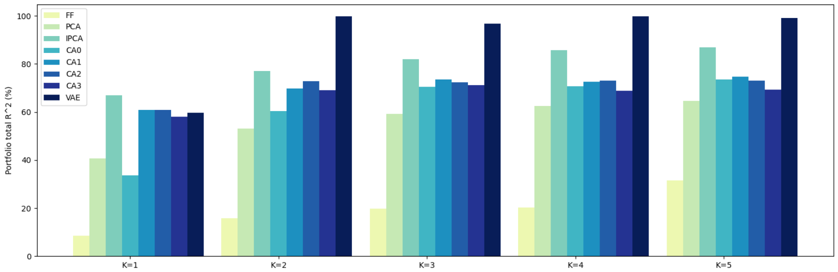


Figure 3: Portfolio-level total R^2 (%)

Out-of-sample total R^2 increases monotonically in the number of portfolios K for all models. Among the non-neural-network specifications, IPCA achieves the highest explanatory power, rising from 66.96% at $K = 1$ to 86.99% at $K = 5$. The VAE, which introduces robust latent factors inferred from noisy financial-market data, also attains strong fit (59.78% $\rightarrow$ 99.06%), although it exhibits instability when $K = 1$. The conditional autoencoders occupy the next tier (CA1: 60.77% $\rightarrow$ 74.63%; CA2:

Table 2: Portfolio-level predictive R^2

Model	$K=1$	$K=2$	$K=3$	$K=4$	$K=5$
FF	-0.59	-1.10	-0.72	-0.46	-0.38
PCA	0.30	0.60	0.69	0.75	0.80
IPCA	0.15	0.53	-0.34	-0.17	-0.36
CA$_0$	-0.45	-0.65	-0.87	-2.68	0.08
CA$_1$	-0.47	-2.51	-1.61	-1.79	-4.46
CA$_2$	-3.25	-2.23	-4.33	-1.47	-2.02
CA$_3$	-1.37	-1.44	-5.25	-2.86	-2.24
VAE	-0.17	-1.41	-2.40	-3.30	-1.10

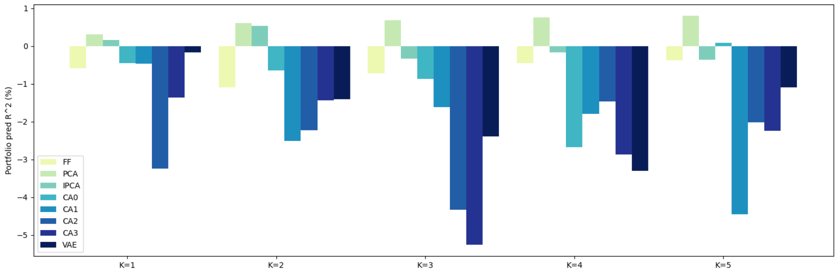


Figure 4: Portfolio-level predictive R^2

60.86% → 73.01%; CA3: 58.04% → 69.20%). PCA attains intermediate fit (40.61% → 64.68%), whereas the observable-factor (FF) model exhibits the weakest performance (8.54% → 31.40%).

Forecasting accuracy diverges sharply from contemporaneous fit. PCA delivers small but uniformly positive predictive R^2, increasing from 0.30% at $K = 1$ to 0.80% at $K = 5$. IPCA shows modest gains for $K \leq 2$ (0.15%, 0.53%) but negative R^2 thereafter (down to –0.36%). All conditional autoencoders (CA–CA) and the VAE yield negative predictive R^2 across most K.

Note. Predictive R^2 is calculated using factors averaged up to time $t-1$, namely

$$\bar{f}_{t-1} = \frac{1}{t-1}\sum_{s=1}^{t-1} \hat{f}_s,$$

which attenuates dynamic variation and contributes to the lower out-of-sample R^2.

5.2 Risk Premia vs. Mispricing

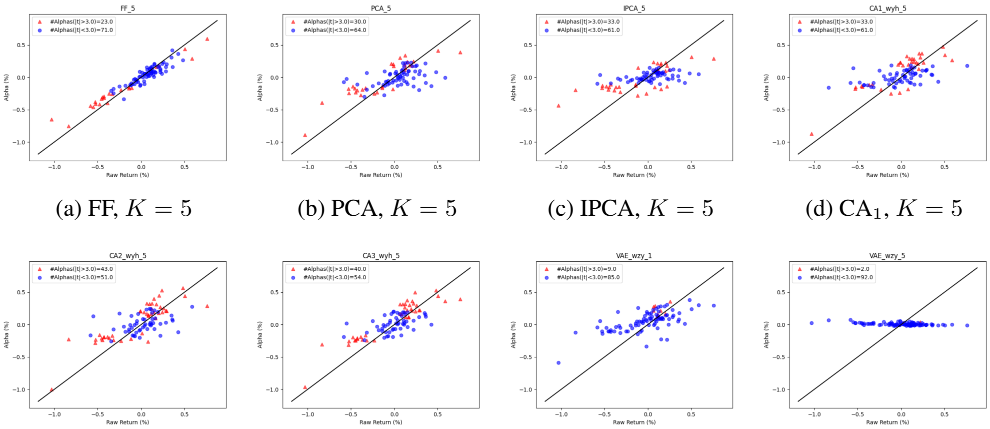
In this section, we follow the empirical framework of GKX and KPS to conduct a systematic examination of the pricing errors for the managed portfolios x_t, with the goal of disentangling the component due to genuine risk premia from that due to mispricing.

First, we define the unconditional pricing error for each asset or portfolio i as

$$\alpha_i := \mathbb{E}[u_{i,t}] = \mathbb{E}\big[r_{i,t} - \beta'_{i,t-1} f_t\big],$$

where $u_{i,t}$ is the one-period residual, and both the factor loadings $\beta_{i,t-1}$ and the factors f_t are estimated under a zero-intercept, no-arbitrage model. Under exact no-arbitrage, α_i should be statistically indistinguishable from zero.

Despite its nonlinear architecture, the VAE almost fully eliminates systematic mispricing, registering only 2 significant alphas at $K = 5$ (and 9 at $K = 1$). By contrast, the observable-factor (FF) model leaves 23 significant alphas, reflecting persistent pricing errors even with a small factor set. IPCA and PCA reduce mispricing moderately, with 33 and 30 significant alphas respectively, but do not match the VAE's performance. The conditional autoencoders occupy an intermediate position: CA$_1$ eliminates roughly one-third of FF's mispricing (33 significant alphas), whereas the deeper CA$_3$ retains 40 significant alphas.



(a) FF, $K = 5$

(b) PCA, $K = 5$

(c) IPCA, $K = 5$

(d) CA$_1$, $K = 5$

(e) CA$_2$, $K = 5$

(f) CA$_3$, $K = 5$

(g) VAE, $K = 1$

(h) VAE, $K = 5$

Figure 5: Scatter-plot diagnostics of out-of-sample α_i vs. raw return for all eight models. Red triangles denote $|t| > 3$.

Together with Tables 1–2, these diagnostics confirm that nonlinear latent-factor models—particularly the VAE—substantially contract residual mispricing and approach the no-arbitrage ideal, whereas both static linear models and autoencoders fall short.

5.3 Feature Importance Analysis

To measure the marginal role of a single feature in cross-sectional pricing, this paper adopts the variable importance measure proposed by GKX: while keeping all estimated parameters constant, the sample value of a certain feature is reset to zero, and the total R^2 of the sample period is recalculated. The degree of decrease is the importance score of that feature. Due to the fact that Total R^2 captures both explanatory and predictive information, this approach can quantify the comprehensive contribution of features to both types of information without changing the model structure.

The heat-map demonstrates that the VAE's explanatory power is concentrated in a limited set of firm characteristics.

- **Balance–Sheet Quality.** The deepest shading corresponds to the quick ratio (`quick`) and the accrual-based measure (`pctacc`), indicating that firms with stronger short-term liquidity and lower discretionary accruals command distinct risk premia.
- **Liquidity Risk.** Measures of trading frictions—zero-trade frequency (`zerotrade`) and turnover volatility (`std_turn`)—produce large declines in total R^2 when zeroed, underscoring their fundamental role in cross-sectional asset pricing.
- **Volatility Metrics.** idiosyncratic volatility (`idiovol1`) and market beta (`beta`) exhibit pronounced importance scores, reflecting the premium investors require for bearing conditional variance and covariance risk.
- **Momentum Signals.** Both 12-month and 36-month momentum proxies (`mom12m`, `mom36m`) and changes in momentum (`chmom`) incur substantial losses in total R^2 when removed, highlighting the persistence of past-return effects.

These results indicate that the VAE allocates most explanatory power to Balance–Sheet Quality, volatility, and momentum effects.

6 Conclusion

We replicate Gu, Kelly, and Xiu's conditional-autoencoder (CA) model, finding only small quantitative gaps—traceable to unreported hyper-parameters—while all qualitative rankings persist. Replacing the deterministic CA with a Variational Autoencoder (VAE) under the same no-arbitrage constraint

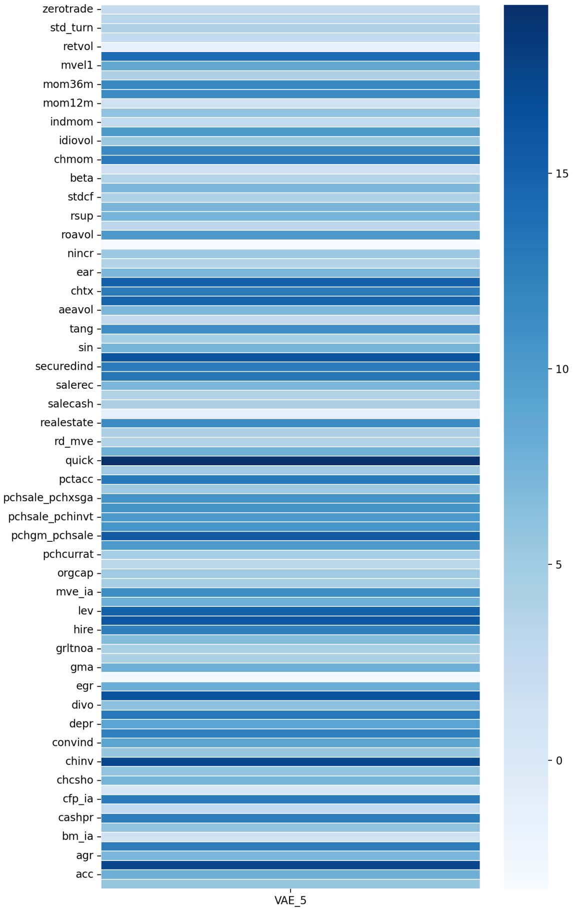


Figure 6: Heat map of variable-importance scores for the VAE with $K = 5$. Darker shading denotes a larger decline in total R^2 upon zeroing the corresponding feature.

raises out-of-sample total R^2 above every benchmark (CA, PCA, IPCA) and keeps predictive R^2 on par with the best alternatives. By learning full factor distributions, the VAE also enables principled uncertainty and tail-risk assessment. These results show that probabilistic latent-factor models can boost explanatory power without sacrificing forecast accuracy; future work can refine latent priors and link the VAE to macro state variables for real-time risk management.

References

[1] Gu, S., Kelly, B. T., & Xiu, D. (2019). *Autoencoder Asset Pricing Models*. Yale ICF Working Paper No. 2019–04; Chicago Booth Research Paper No. 19–24.

[2] Duan, Y., Wang, L., Zhang, Q., & Li, J. (2022). FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder for Predicting Cross-Sectional Stock Returns. *Proceedings of the Thirty–Sixth AAAI Conference on Artificial Intelligence (AAAI–22).*

[3] Gu, S., Kelly, B., & Xiu, D. (2019). Empirical Asset Pricing via Machine Learning. *Review of Financial Studies*, forthcoming.