Factor-Based Conditional Diffusion Model for Portfolio Optimization

Xuefeng Gao
CUHK
xfgao@se.cuhk.edu.hk

Mengying He
CUHK
myhe@se.cuhk.edu.hk

Xuedong He
CUHK
xdhe@se.cuhk.edu.hk

Abstract

We propose a novel conditional diffusion model for portfolio optimization that learns the cross-sectional distribution of next-day stock returns conditioned on asset-specific factors. The model builds on the Diffusion Transformer with token-wise conditioning, linking each asset's return to its own factor vector while capturing cross-asset dependencies. Generated return samples are used for daily mean–variance optimization under realistic constraints. Empirical results on the Chinese A-share market show that our approach consistently outperforms benchmark methods based on standard empirical and shrinkage-based estimators across multiple metrics.

1 Introduction

Diffusion-based generative models is a powerful class of probabilistic generative AI models that can generate samples from high-dimensional data distributions [Song and Ermon, 2019, Ho et al., 2020]. The key idea is to use a forward process to gradually turn the unknown target distribution to a simple noise distribution, and then reverse this process to generate new samples. Diffusion models have achieved state-of-the-art performances and outperformed GANs (Generative Adversarial Nets Goodfellow et al. [2014]) in applications such as image and audio generations [Dhariwal and Nichol, 2021, Rombach et al., 2022, Ramesh et al., 2022]. However, due to the complexities of financial markets, the use of diffusion models for tackling financial decision making problems like portfolio optimization, is still a relatively underexplored area of study.

In this paper, we introduce a factor-based conditional diffusion model for portfolio optimization, which generates the full cross-sectional distribution of next-day stocks returns conditional on asset-specific factors. Motivated by its proven success in conditional image and video generation, our model adapts the Diffusion Transformer (DiT, Peebles and Xie [2023]) architecture by incorporating a token-wise conditioning mechanism that integrates assets-level factor information for each asset. Empirical results on the Chinese A-share market show that mean-variance portfolios constructed through daily optimization based on the generated return distributions significantly outperform those based on standard empirical and shrinkage estimators. To our best knowledge, this is the first study to combine factors with conditional diffusion models for portfolio optimization, highlighting the value of modeling the full return distribution for financial decision-making.

Our work is closely related to two working papers, and we next explain the key differences. (1) Chen et al. [2025] propose an *unconditional* diffusion model with a *latent (unobservable)* factor structure aimed primarily at dimensionality reduction in high-dimensional return modeling. In contrast, we develop a *conditional* diffusion model that leverages a rich set of *observable* firm-level characteristics to generate the next-day cross-sectional return distribution, serving as a direct input for portfolio optimization. In addition, unlike their approach which updates portfolio weights annually in their empirical study, our framework generates one-day-ahead return distributions, and we perform daily weight updates and portfolio rebalancing in response to changing market conditions; (2) Cetingoz and Lehalle [2025] employ PCA-based latent factor extraction, followed by a GAN-based generative model to produce synthetic data that replicates statistical properties of real financial time series.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Generative AI in Finance.

They only evaluate performance of equal-weighted portfolios under real and synthetic datasets. In contrast, our model does not aim to replicate stylized facts for data simulation; instead, it is trained to generate next-day return distributions conditioned on firm-specific factors, with the explicit objective of improving mean-variance portfolio allocation performance.

Finally, although not specifically applied to finance, diffusion models have been employed in time series forecasting tasks. Notable examples include Rasul et al. [2021] and Li et al. [2022]. Moreover, other deep learning approaches such as FNN [Gu et al., 2020] or CNN [Jiang et al., 2023] have also been used to predict future mean returns or the direction of future returns for building long-short portfolios. In contrast, our generative approach can learn the entire return distribution, enabling the construction of more diverse portfolios. Our framework is highly versatile, applicable not only to mean-variance portfolio optimization but also to other mean-risk models, such as mean-CVaR.

2 Methodology

We follow the Arbitrage Pricing Theory [Ross, 2013] and the general asset pricing framework used in Gu et al. [2020] to model stock returns using the following statistical factor model:

$$R_{t+1} = f(X_t) + u_{t+1},$$

where $R_{t+1} \in \mathbb{R}^D$ is the return vector of D stocks in the period from time t to $t+1$, $X_t = (x_{1,t}, x_{2,t}, ..., x_{D,t})' \in \mathbb{R}^{D\times K}$ is a factor matrix observable at time t, u_{t+1} is a random shock independent of the information at time t, and f is a deterministic function prescribing the dependence of the stock return on the factors. The form of f and the distribution of u_{t+1} are unknown, so the conditional distribution of R_{t+1} given X_t is also unknown and will be learned from data.

We adopt the conditional denoising diffusion probabilistic model (DDPM; Ho et al., 2020) to learn the distribution of R_{t+1} given X_t. More precisely, given a pair (X_t, R_{t+1}), we add Gaussian noise to R_{t+1} using a forward diffusion process:

$$R^{(0)}_{t+1} = R_{t+1}, \quad R^{(n)}_{t+1} = \sqrt{1-\beta_n} R^{(n-1)}_{t+1} + \sqrt{\beta_n}\epsilon_n,\ n = 1, \ldots N,$$

where ϵ_n's are i.i.d. D-dimensional standard Gaussian random variable and $\{\beta_n\}_{n=1}^N \in (0,1)$ is a predefined variance schedule. Denote $\alpha_n = 1-\beta_n$ and $\bar{\alpha}_n = \prod_{s=1}^n \alpha_s$. In the reverse (denoising) process, we draw R^N_{t+1} from $\mathcal{N}(0, I)$ and $R^{(n-1)}_{t+1}$ from

$$\mathcal{N}\left(\frac{1}{\sqrt{\alpha_n}}\left(R^{(n)}_{t+1} - \frac{\beta_n}{\sqrt{1-\bar{\alpha}_n}}\boldsymbol{\epsilon}_\theta\left(R^{(n)}_{t+1}, n; X_t\right)\right), \sigma_n^2 I_D\right),\ n = N, \ldots, 1,$$

where $\sigma_n^2 := \frac{1-\bar{\alpha}_{n-1}}{1-\bar{\alpha}_n}\beta_n$ and $\boldsymbol{\epsilon}_\theta\left(R^{(n)}_{t+1}, n; X_t\right)$ is modeled by a neural network with parameter θ and will be trained using data. Ho et al., 2020 set the loss function to be a variant of the usual variational bound on the negative log likelihood of $R^{(0)}_{t+1}$ in the reverse process, which is given by

$$L(\theta) := \frac{1}{N}\sum_{n=1}^N \mathbb{E}\left[\|\epsilon - \boldsymbol{\epsilon}_\theta\left(R^{(n)}_{t+1}, n; X_t\right)\|^2\right],$$

where ϵ is a D-dimensional standard Gaussian random variable and the expectation is taken with respect to (X_t, R_{t+1}) and ϵ. See Section 3.4 in Ho et al., 2020. The loss function can be estimated by using the data $(X_t, R_{t+1}), t = 0, \ldots, T-1$ as T samples of (X_t, R_{t+1}). The stochastic gradient descent algorithm and its variants can be applied to find the optimal parameter value θ^*.

We adopt the DiT architecture as the denoising network for $\boldsymbol{\epsilon}_\theta$, motivated by its effectiveness in conditional image and video generation tasks. To adapt it to financial cross-sectional stock return data, we introduce three key modifications: (1) Unlike the original DiT, which operates on latent image patches, our model works directly in the raw data space. Each token is a single asset's noisy return, eliminating the need for a VAE-style encoder. (2) In contrast to global conditioning in original DiT, we implement token-wise conditioning to capture asset-specific heterogeneity. Each token is conditioned on its own factor vector $x_i \in \mathbb{R}^K$, producing a condition vector $c_i = \text{MLP}(x_i) + e_n$, where MLP is a multi-layer perceptron and e_n is the embedding of diffusion step n. While conditioning is local, cross-asset dependencies are modeled globally via multi-head self-attention block, enabling the network to learn the return covariance structure. (3) Consequently, the AdaLN-Zero modulation parameters are computed locally for each token from its unique condition vector c_i via $\text{MLP}_{\text{Ada}}(c_i)$, rather than being shared globally. This allows the network to apply tailored transformations to each asset, thereby taking into account the heterogeneity across assets.

Given the latest observable factor matrix X_t, the model generates a large set of synthetic return samples from the learned conditional distribution. These samples are then used to estimate the mean and covariance of returns, which serve as direct inputs for mean-variance portfolio optimization.

3 Experimental Results

3.1 Data and experiments setup

In this section, we empirically demonstrate the effectiveness of our proposed factor-based conditional diffusion model in single-period mean-variance portfolio selection [Markowitz, 1952, DeMiguel et al., 2009] on the Chinese A-share market. We consider daily investment and use daily stock return data for the CSI 300 Index constituents in the period 4-Jan-2017 (time 1) to 9-Apr-2025, obtained from the Wind Database (`https://www.wind.com.cn/`). The corresponding stock-level factor data covering the period 3-Jan-2017 to 8-Apr-2025 is sourced from Datayes (`https://mall.datayes.com/`). We focus on price-volume-based factors, resulting in 208 factors per stock. We retain 113 stocks from the CSI 300 that have complete daily observations throughout the sample period. For each day of the training period, factor values standardized across all stocks and winsorized at three standard deviations to mitigate the influence of outliers, with missing values imputed by the mean across all stocks of that factor. Stock returns are winsorized similarly.

The dataset is split chronologically into a training period 4-Jan-2017 to 29-Dec-2023 and a test period 2-Jan-2024 to 9-Apr-2025, maintaining an 8:2 train/test ratio. The diffusion model is trained for 30 epochs using the Adam optimizer with a learning rate of 0.003 and a batch size of 16.

3.2 Portfolio optimization

We first consider the case in which transaction costs are not taken into account. In this case, the mean-variance portfolio selection problem is formulated as

$$\max_{\boldsymbol{\omega}} \quad \boldsymbol{\omega}^\top \boldsymbol{\mu} - \frac{\gamma}{2}\boldsymbol{\omega}^\top \boldsymbol{\Sigma}\boldsymbol{\omega} \quad \text{s.t.} \quad \boldsymbol{\omega}^\top \mathbf{1} = 1, \omega_i \geq 0, \quad \forall i, \tag{1}$$

where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_D)'$, γ, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ denote the portfolio weight vector, the investor's risk aversion degree, the mean and covariance matrix of the stock return vector, respectively, and the short sale constraint $\omega_i \geq 0$ is imposed because short sales are not allowed in the A-share market.

We set $\gamma = 100$ so as to generate a reasonable amount of risk taking. At each time t, we estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, compute the optimal portfolio for the coming day, and implement the portfolio. We use three estimation methods: (i) Factordiff, which estimates the mean and covariance based on 500 sets of predictive samples generated by our conditional diffusion model; (ii) Emp, which computes the sample mean $\bar{\boldsymbol{\mu}}_t$ and sample covariance $\bar{\boldsymbol{\Sigma}}_t$ using historical data from time 1 to t; and (iii) ShrEmp, the James–Stein shrinkage estimator [James and Stein, 1992]. We compare the performance of the portfolios based on these three estimation methods as well as the performance of EW, the equally weighted portfolio [DeMiguel et al., 2009], in terms of the mean, standard deviation, Sharpe ratio, Sortino ratio, Calmar ratio, and Return-to-CVaR (RtC) [Huang et al., 2024]. The results are shown in the first five columns of Table 1, showing that our model Factordiff yields better performance than the other three strategies if transaction costs are not taken into account.

In practice, trading incurs transaction fees. In the A-share market, the fees include trading commissions, stamp tax (applied only to seller), and slippage, which amount to approximately 7.5 basis points (bps) for buying orders and 12.5 bps for selling orders per unit trading amount [Leippold et al., 2022]. Columns 6–9 of Table 1 show the mean return, standard deviation, Sharpe ratio, Sortino ratio, Calmar ratio, and Return-to-CVaR of the portfolio with transaction fees deducted. There are two observations: First, the transaction fees are negligible for the EW, Emp, and ShrEmp portfolios. Note that the portfolio weights are constant in EW and nearly constant in Emp and ShrEmp because a single new data point has negligible impact on the moment estimates given a large amount of existing data and, consequently, the day-by-day update of the empirical mean and covariance of stock returns is minimal. The daily stock price change is small, e.g., within 6%, so the transaction fees due to portfolio rebalancing in the case of maintaining constant portfolio weights over time are minimal. Second, with transaction fees deducted, Factordiff underperforms Emp and ShrEmp, which is due to the large amount of transaction fees incurred in Factordiff. This is because the portfolio weights in Factordiff vary significantly over time, as shown in the top left panel of Figure 1.

The above findings show the importance of considering transaction fees in the construction and evaluation of portfolio strategies, whereas these fees are ignored in some studies in the literature. To

Table 1: Performance of the EW portfolio and the optimal portfolio of (1) (with $\gamma = 100$) with Factordiff (500 samples), Emp, and ShrEmp estimates of stock return moments.

	Transaction Fees Ignored				Transaction Fees Deducted			
Metric	EW	Factordiff	Emp	ShrEmp	EW	Factordiff	Emp	ShrEmp
Mean (%)	0.056	**0.133**	0.096	0.098	0.055	0.077	0.094	**0.096**
Std (%)	1.313	1.155	**0.957**	0.962	1.313	1.154	**0.957**	0.962
Sharpe	0.043	**0.116**	0.100	0.102	0.042	0.067	0.098	**0.100**
Sortino	0.066	**0.168**	0.149	0.151	0.064	0.094	0.146	**0.148**
Calmar	0.004	**0.011**	0.011	0.011	0.004	0.006	0.011	**0.011**
RtC	0.021	**0.049**	0.044	0.045	0.021	0.028	0.044	**0.044**

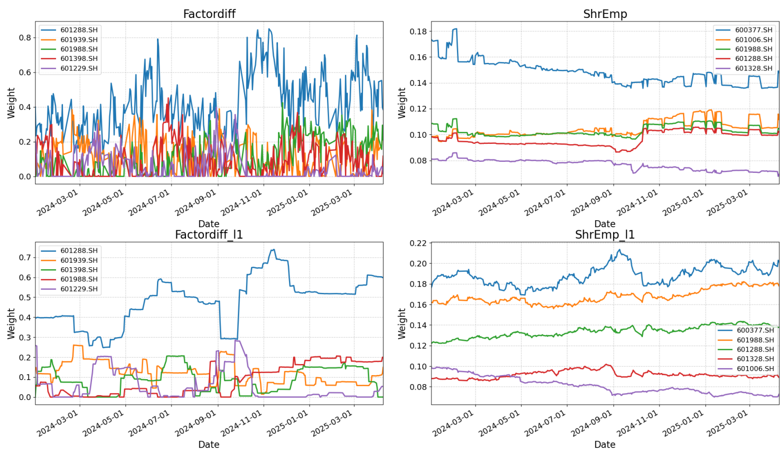

Figure 1: Portfolio weights over time for the top 5 stocks in the optimal portfolio of (1) (top panels) and in the optimal portfolio of (2) (bottom panels) with stock return moments estimated by 500 samples generated in Factordiff (left panels) and by ShrEmp (right panels).

account for transaction fees, we consider the optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\omega_t}, \boldsymbol{b_t}, \boldsymbol{s_t}} \quad & \boldsymbol{\omega_t}^\top \boldsymbol{\mu_t} - \frac{\gamma}{2} \boldsymbol{\omega_t}^\top \boldsymbol{\Sigma_t} \boldsymbol{\omega_t} - \left(0.00075 \boldsymbol{b_t}^\top \mathbf{1} + 0.00125 \boldsymbol{s_t}^\top \mathbf{1}\right) \\ \text{s.t.} \quad & \boldsymbol{\omega_t}^\top \mathbf{1} = 1, \boldsymbol{b_t} \geq 0, \boldsymbol{s_t} \geq 0, 0 \leq \omega_{t,i} \leq 1, \omega_{t,i} - \omega_{t,i}^d = b_{t,i} - s_{t,i}, \quad \forall i, \end{aligned} \tag{2}$$

where $\boldsymbol{b_t}$ and $\boldsymbol{s_t}$ stand for the vectors of buying and selling amounts, per unit wealth, of the stocks due to portfolio rebalancing at time t and $\omega_{t,i}^d$ denotes the dollar amount of stock i's holding at time t before portfolio rebalancing. Table 2 presents the performance of the EW portfolio and the performance of the optimal portfolio of (2) with estimates of the mean and convariance of stock returns based on Factordiff (100, 500, and 1000 samples generated), Emp, and ShrEmp, taking into account the transaction fees. We can see that Factordiff outperforms the other three strategies. We can also observe from the bottom left panel of Figure 1 that the Factordiff portfolio in (2) entails much smoother portfolio weights over time than the Factordiff portfolio in (1), so the former incurs much lower transaction fees, explaining its superior performance.

Table 2: Performance of the EW portfolio and the optimal portfolio of (2) (with $\gamma = 100$) with Factordiff (100, 500, and 1000 samples), Emp, and ShrEmp estimates of stock return moments. Transaction fees are included.

Metric	EW	Factordiff (100)	Factordiff (500)	Factordiff (1000)	Emp	ShrEmp
Mean (%)	0.055	0.100	0.123	**0.138**	0.095	0.099
Std (%)	1.313	1.151	1.172	1.188	**0.983**	0.999
Sharpe	0.042	0.087	0.105	**0.116**	0.096	0.099
Sortino	0.064	0.125	0.154	**0.172**	0.141	0.144
Calmar	0.004	0.008	0.010	**0.011**	0.011	0.010
RtC	0.021	0.036	0.045	**0.049**	0.042	0.043

References

Adil Rengim Cetingoz and Charles-Albert Lehalle. Synthetic data for portfolios: A throw of the dice will never abolish chance. *arXiv preprint arXiv:2501.03993*, 2025.

Minshuo Chen, Renyuan Xu, Yumin Xu, and Ruixun Zhang. Diffusion factor models: Generating high-dimensional returns with factor structure. *arXiv preprint arXiv:2504.06566*, 2025.

Victor DeMiguel, Lorenzo Garlappi, Francisco J Nogales, and Raman Uppal. A generalized approach to portfolio optimization: Improving performance by constraining portfolio norms. *Management science*, 55(5):798–812, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5):2223–2273, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Yilie Huang, Yanwei Jia, and Xun Yu Zhou. Mean–variance portfolio selection by continuous-time reinforcement learning: Algorithms, regret analysis, and empirical study. *arXiv preprint arXiv:2412.16175*, 2024.

William James and Charles Stein. Estimation with quadratic loss. In *Breakthroughs in statistics: Foundations and basic theory*, pages 443–460. Springer, 1992.

Jingwen Jiang, Bryan Kelly, and Dacheng Xiu. (re-) imag (in) ing price trends. *The Journal of Finance*, 78(6):3193–3249, 2023.

Markus Leippold, Qian Wang, and Wenyu Zhou. Machine learning in the chinese stock market. *Journal of Financial Economics*, 145(2):64–82, 2022.

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35: 23009–23022, 2022.

Harry Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, Mar. 1952.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International conference on machine learning*, pages 8857–8868. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Stephen A Ross. The arbitrage theory of capital asset pricing. In *Handbook of the fundamentals of financial decision making: Part I*, pages 11–30. World Scientific, 2013.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32, 2019.