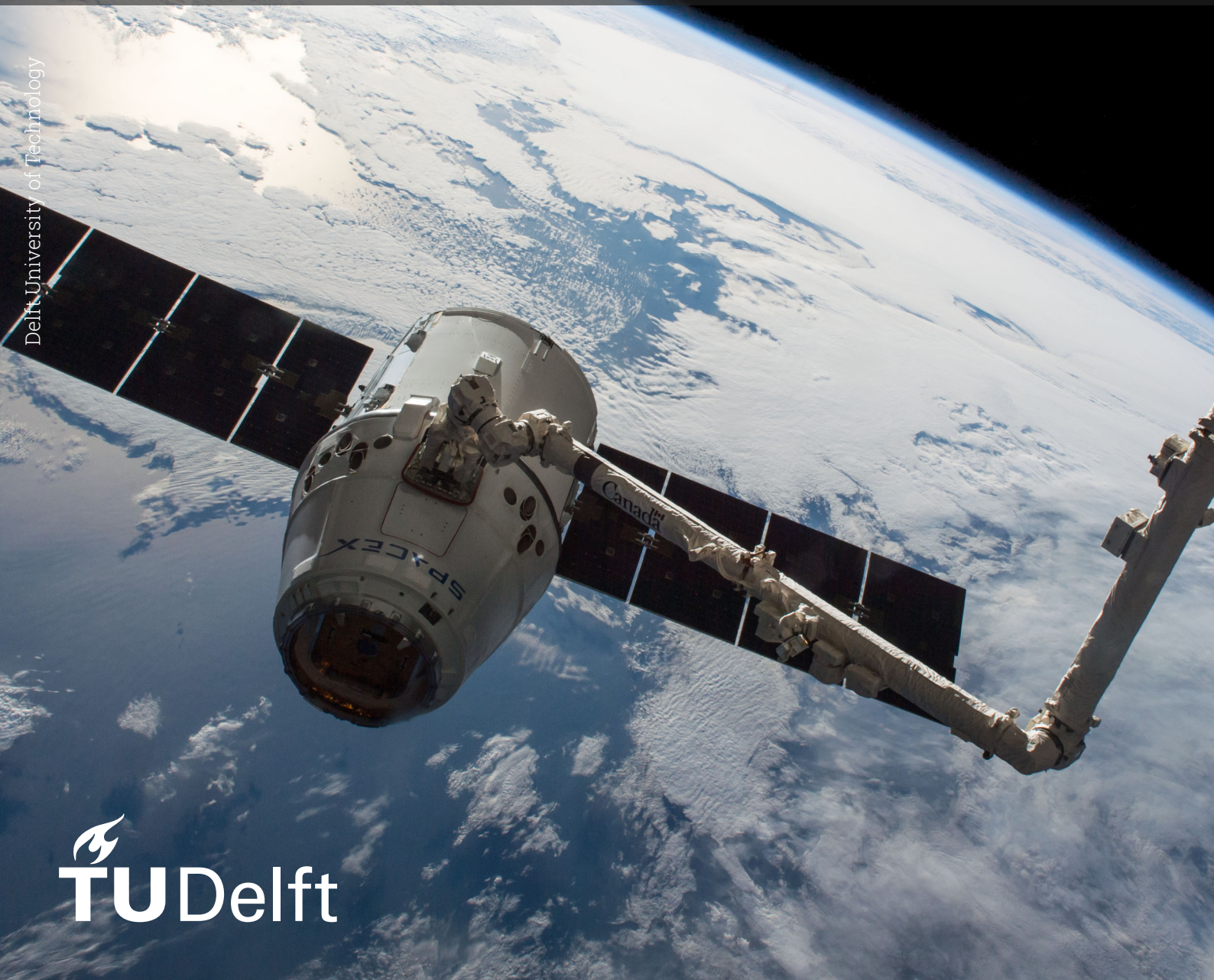
FaDiff: Conditional Factor Diffusion for Equity Return Scenario Generation

CS5000: MSc CS Thesis project

Renyi Yang

Delft University of Technology



TU Delft

FaDiff: Conditional Factor Diffusion for Equity Return Scenario Generation

by

Renyi Yang

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday, August 26, 2026.

Student number:	5470668	
Project duration:	January 5, 2026 – July 31, 2026	
Thesis committee:	Prof. dr. Lydia Chen,	TU Delft, Chair
	Dr. Shuaiqiang Liu,	TU Delft
	Dr. Burcu Özkan,	TU Delft
	Dr. Daniel Gradeci,	Robeco UK Ltd.
	Ewout Noort,	Robeco Nederland B.V.

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

This thesis concludes my Master of Science in Computer Science at Delft University of Technology. The research was carried out during an internship at Robeco, where I investigated diffusion-based generative models for equity return scenario generation.

I would like to thank my thesis advisor, Prof. Lydia Chen, for giving me the opportunity to carry out this thesis and for her guidance throughout the process. I also thank Burcu Özkan for taking the time to serve on my thesis committee. I am especially grateful to Shuaiqiang Liu, whose day-to-day supervision shaped this thesis in countless ways. His concrete and always constructive feedback on experimental design, on writing, and on how to think about a problem before rushing to solve it taught me what rigorous research looks like.

At Robeco, I would like to thank Daniel Gradeci, who proposed the research direction that became this thesis, and Ewout Noort, who introduced me to the factor perspective that fundamentally shaped the methodology. Beyond the research itself, their guidance during my internship gave me an invaluable view of how quantitative research is actually used in practice.

Renyi Yang
Delft, August 2026

Abstract

Quantitative investment strategies stand or fall on their behaviour in rare and extreme market events, yet the scenario generators used for stress testing fail on opposite sides of that requirement. Historical resampling cannot leave the observed record, and mutivarite Gaussian modeling extrapolates only along a thin-tailed elliptical law that understates joint extremes. This thesis asks whether a denoising diffusion model can close the gap, preserving the heavy-tailed joint distribution of equity returns, reaching counterfactuals that history does not contain, and doing so within a workflow a risk manager can actually use.

We propose FaDiff, a two-stage pipeline for the equity cross-section. A parametric factor model, estimated by ordinary least squares and then frozen, reduces each day's returns to a small vector of factor returns plus Student-t idiosyncratic residuals. The factor returns are generated by a Diffusion Transformer whose forward process is driven by anisotropically coloured stable Lévy noise, so that heavy tails and cross-factor co-movement are built into the corruption process. Stress scenarios are imposed at inference time through SNR-weighted gradient guidance, requiring no labels and no retraining per scenario, and a first-order autoregressive rollout extends the generator to multi-day paths.

Evaluated on a 25-year MSCI World index against resampling and Gaussian baselines that share every other component of the pipeline, the model reproduces the out-of-sample factor distribution more faithfully than either baseline, across marginals, joint tail and conditional structure, and its multi-day paths preserve compounding and drawdown behaviour. A second, self-built dataset of one-minute Dow 30 bars preserve the good result the unconditional comparison at a different frequency and on a cross-section modelled without the factor decomposition; Applied to stress testing, the pipeline evaluates live portfolios under factor scenarios with no historical precedent and surfaces exposures that a full year of realised performance leaves invisible. Code available at `https://github.com/yryMax/TailEnhancedFinDiff`.

Contents

Preface i

Abstract ii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Research Questions and Contributions . . . 3
- 1.4 Thesis Outline . . . 3

2 Preliminary 4
- 2.1 Diffusion Model . . . 4
  - 2.1.1 Conditional Sampling . . . 5
  - 2.1.2 Temporal Modeling . . . 6
- 2.2 Scenario Generation . . . 6
  - 2.2.1 Traditional Methods . . . 6
  - 2.2.2 Generative AI Methods . . . 7

3 Methodology 8
- 3.1 Parametric Factor Decomposition . . . 8
- 3.2 Factor Diffusion Model . . . 9
  - 3.2.1 Forward Process: Heavy-Tailed Noising . . . 10
  - 3.2.2 Reverse Process: (Conditional) Parameterised Denoising . . . 11
  - 3.2.3 Denoiser . . . 12
- 3.3 Temporal Factor Generation . . . 13
- 3.4 Generation Pipeline . . . 14

4 Experiments 16
- 4.1 Datasets . . . 16
- 4.2 Baselines . . . 18
- 4.3 Implementation Details . . . 19
- 4.4 Evaluation Metrics . . . 20

5 Results and Evaluation 23
- 5.1 Unconditional Cross-Sectional Generation . . . 23
  - 5.1.1 Statistical Moments . . . 23
  - 5.1.2 Marginal Distributions . . . 24
  - 5.1.3 Factor Correlations . . . 26
  - 5.1.4 Stock-Level Evaluation . . . 28
- 5.2 Conditional Cross-Sectional Generation . . . 31
  - 5.2.1 Volatility Conditioning . . . 31
  - 5.2.2 Momentum Conditioning . . . 33
  - 5.2.3 Stock-Level Propagation . . . 34
  - 5.2.4 Joint Conditioning . . . 34
- 5.3 Temporal Generation . . . 34
- 5.4 Downstream Risk Management . . . 37
  - 5.4.1 Conditional Generation on Factors . . . 38
  - 5.4.2 Benchmark Portfolio . . . 38
  - 5.4.3 Strategy Portfolios . . . 39

6 Ablation Studies 41
- 6.1 Diffusion Model on Learning Correlation Structure . . . 41

6.2 Data convergence on MSCI 42

7 Conclusion **45**
7.1 Summary 45
7.2 Discussion 46

References **48**

A Additional Generation Results **50**
A.1 Unconditional Generation 50
A.2 Conditional Generation 50

B Notation **57**

C Generated Samples **59**

D Use of Generative AI **61**

1

Introduction

This chapter sets up the thesis in four steps. Section 1.1 explains why scenario generation matters for risk management and why the two families of generator in current use each fail on some of the real need. Section 1.2 turns that motivation into a precise task, stating what is generated, in which forms, under what kinds of condition, and what may legitimately be claimed about each. Section 1.3 poses the four research questions that follow from it and lists the contributions. Section 1.4 maps the remainder of the thesis.

1.1. Motivation

In quantitative investment management, Strategies respond non-linearly to market stress, and episodes such as the 2008 global financial crisis or the COVID-19 shock of March 2020 compress years of ordinary risk into a few weeks. Whether a strategy survives such an episode is determined not by its expected return but by its behaviour in the tail of the return distribution. Robust risk management therefore turns on stress testing, the ability to ask, before the fact, what a portfolio would do under a market regime that has not occurred.

As an example we examines three live strategy portfolios that returned +0.3%, +0.3% and +2.2% per month against their benchmark over 2025, figures that make them nearly indistinguishable. Under generated factor-stress scenarios they separate immediately. One leans on the value factor, one on momentum, and one is close to factor-neutral. A full year of live profit and loss, in other words, says almost nothing about what a portfolio is exposed to. What risk management needs is a distribution over what could have happened and what could yet happen, not the single path that did.

Practice supplies that distribution with two families of scenario generator, and they fail on opposite sides of the requirement. Historical resampling, such as the stationary bootstrap [26], reproduces the empirical distribution faithfully but is backward-looking by construction, since it reshuffles observed days and can never produce a configuration absent from the record. Parametric Monte Carlo under Gaussian assumptions extrapolates freely, but only along a thin-tailed elliptical law, which is precisely the assumption that understates the heavy tails, skewness and joint extremes characteristic of equity returns under stress [6]. Neither family can answer the question a stress test actually poses. What happens to this portfolio if, say, the value factor rallies by thirty per cent in a month?

Making a modern generative model answer that question is not straightforward, for three reasons that shape the design of this thesis. First, dimensionality. The equity cross-section comprises one to two thousand stocks per day, while a quarter-century of daily data supplies only about six thousand observations, far too few to learn a distribution of that dimension directly. Second, the tails themselves. A diffusion model corrupted by Gaussian noise inherits Gaussian tails, so heavy-tailed behaviour must be built into the generative process rather than hoped for from the data. Third, conditioning. Stress regimes carry no labels, their closest historical analogues number in single-digit days, and retraining a model for every scenario a risk manager might pose is impractical.

This thesis addresses all three with a two-stage design that we call FaDiff (Factor Diffusion). A parametric factor model, estimated once by ordinary least squares and then frozen, reduces the cross-section to seven factor returns plus a per-stock Student-t idiosyncratic residual, so that the generative problem becomes low-dimensional. The factor returns are generated by a Diffusion Transformer whose forward process is driven by anisotropically coloured stable Lévy noise, so that heavy tails and cross-factor co-movement enter through the corruption process itself. Stress scenarios are imposed at inference time by signal-to-noise-weighted gradient guidance against a differentiable condition on the clean factor vector, which requires no labels and no retraining per scenario, and a first-order autoregressive rollout extends the generator from a single cross-section to multi-day paths. Throughout the empirical work, the factor model and the residual draws are held fixed and only the factor sampler changes between methods, so any difference in results comes from the generative component alone.

1.2. Problem Statement

The ultimate goal is to develop a generator of equity returns. The terms and scopes are developed as following.

The generated target. Fix a universe of N assets observed on a regular time grid indexed by t, and let $R_{i,t}$ denote the simple return of asset i over interval t. Nothing in the formulation depends on the length of that interval. The same construction applies at daily frequency and at a different frequency, for example intraday.

The shape. The generator must produce that quantity in two shapes. A *cross-section* is an instant observation of all equity returns at a single time t,

$$\mathbf{R}_t = (R_{1,t}, \ldots, R_{N,t}) \in \mathbb{R}^N,$$

and carries the contemporaneous co-movement between assets, which is what determines whether losses arrive together. A *path* is a sequence of H consecutive cross-sections,

$$\mathbf{R}_{t:t+H} = (\mathbf{R}_t, \ldots, \mathbf{R}_{t+H-1}) \in \mathbb{R}^{N \times H},$$

and adds the time dimension: compounding, volatility persistence and drawdowns. A cross-section is the case $H = 1$, but a path is not a collection of independent cross-sections, since it must also reproduce the dependence between successive intervals.

Conditions. A stress test asks for $p(\cdot \mid c)$ rather than $p(\cdot)$, where c is a constraint supplied at inference time. Conditions differ along two aspects.

The first aspects is which variable is constrained. An endogenous condition acts on the factor coordinates themselves, for example a thirty per cent monthly rally in the value factor. It lives in the same space as the generated vector and can therefore be imposed directly on a sample. An exogenous condition acts on a global indicator outside the equity cross-section, such as the VIX or a macroeconomic release. It is not a coordinate of the generated vector, so the model must carry it as an additional information.

The second aspect is how far the condition sits from the data. A normal condition has many historical instances, so its conditional law is estimable and a generator can be scored against it. An extreme condition has few, in the joint regimes studied here single digits, so a historical analogue exists but is too thin to calibrate against and supports claims of plausibility only. A counterfactual condition has none: no empirical conditional law exists, the generator must extrapolate beyond the record, and correctness cannot be established even in principle. This last case is the one stress testing most often needs, and it is the reason conditioning cannot be learned from labelled examples.

Criterion and scope. The three-way split above fixes what may be claimed where. Wherever a held-out realisation exists, unconditionally and along paths, agreement with it is a falsifiable criterion and the evaluation is a controlled comparison against baselines on that criterion. Where the condition is counterfactual, the evaluation can establish internal consistency and agreement with the closest historical analogues, but not correctness, and the two are kept apart throughout. It worth mentioning that this thesis models the distribution of returns, not their conditional mean, and nothing in it constitutes a forecast of future returns or a claim of predictive skill. A generated scenario is a statement about what is plausible under an assumed regime, and that regime is an input supplied by the user.

1.3. Research Questions and Contributions

RQ1 How can a generative model preserve the heavy-tailed, jointly extreme distribution of the equity factor cross-section, the property that elliptical models remove by construction?

RQ2 How can counterfactual scenarios that history does not contain be obtained at no extra cost, without labelled data, without per-scenario retraining and without new observations?

RQ3 How can temporal dependence, volatility memory, compounding and drawdowns, be preserved by a generator that is itself cross-sectional?

RQ4 Does any of this transfer to practice? Can such a generator support a stress-testing workflow on portfolios that are actually held?

RQ1 is addressed by the Lévy-noise diffusion model and evaluated unconditionally; RQ2 by inference-time guidance, evaluated against historical stress analogues; RQ3 by the autoregressive rollout, evaluated against the temporal stylized facts; and RQ4 by the downstream stress-testing study on live portfolios.

In answering them, the thesis makes four contributions:

- FaDiff, a factor-level diffusion generator whose heavy tails are carried by the forward process, via anisotropically coloured stable Lévy noise;
- a training-free conditioning mechanism that turns the unconditional generator into a scenario generator at inference time;
- a self-built intraday dataset, one-minute Dow 30 bars assembled from a public source with the retrieval and construction scripts released, which serves as a second evaluation panel at a different frequency and on a different cross-section and makes that part of the evaluation reproducible without a commercial data subscription;
- a stress test on portfolios that are actually held, in which three live strategies whose realised performance over a year is nearly indistinguishable are separated immediately by generated factor scenarios that recover the tilt each one carries.

1.4. Thesis Outline

Chapter 2 reviews the background on denoising diffusion models and generative scenario modelling that the thesis builds on. Chapter 3 develops FaDiff, covering the factor decomposition, the Lévy-noise factor diffusion with its guidance mechanism, and the temporal extension. Chapter 4 fixes the empirical setup, the MSCI World panel and the complementary Dow 30 intraday panel with their temporal splits, the two baselines, implementation details and the evaluation metrics. Chapter 5 evaluates the generator unconditionally, under conditioning, along multi-day paths and finally in the downstream stress-testing application. Chapter 6 reports ablations that separate the model's representational capacity from the data regime it is trained in. Chapter 7 summarises the findings and discusses their limitations and the directions they point to.

2

Preliminary

2.1. Diffusion Model

Diffusion models represent a powerful class of generative models that transform a simple prior distribution, typically standard Gaussian noise, into a complex data distribution by reversing a gradual noise-injection process. Built upon foundational pillars [36], this development includes Denoising Diffusion Probabilistic Models (DDPMs), Score-Based Generative Models (SGMs), and Score-based Stochastic Differential Equations (Score SDEs).

Definition: Diffusion Model
Let $P_r(\mathbf{x}_0)$ denote the unknown data distribution. A diffusion model is defined by the tuple $\mathcal{M} = (q, p_\theta, T, \beta)$, where T is the number of diffusion steps and $\beta = \{\beta_t\}_{t=1}^T$ is a variance schedule.

1. **Forward Diffusion Process:** A Markov chain that perturbs $\mathbf{x}_0$ by adding Gaussian noise:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \tag{2.1}$$

   The marginal distribution at step t is analytically derived as:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{a}_t}\mathbf{x}_0, (1-\bar{a}_t)\mathbf{I}) \tag{2.2}$$

   where $a_t = 1 - \beta_t$ and $\bar{a}_t = \prod_{i=1}^{t} a_i$. We write the schedule coefficients as a_t rather than the more common α_t, reserving α throughout this thesis for the stability index of the α-stable law introduced in Section 3.2.1.

2. **Generative Reverse Process:** A parameterized Markov chain p_θ that approximates the intractable reverse distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$:

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p(\mathbf{x}_T) = \mathcal{N}(0, \mathbf{I}) \tag{2.3}$$

   where $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$.

The generative capability of these models stems from their ability to reconstruct the original data from noise. While DDPMs [13, 30] focus on predicting the added noise to perform this reversal, SGMs [32, 33] approach the problem by estimating the score function, the gradient of the log-density, and using Langevin dynamics for sampling. These perspectives were eventually unified through the framework of Stochastic Differential Equations (SDEs) [34], where the forward and reverse processes are viewed as continuous-time evolutions. This unification allows for deterministic sampling through "Probability Flow ODEs," enabling exact likelihood computation and more efficient numerical solvers [17].

Over time, the architectural backbone of these models has evolved from basic multi-layer perceptrons to sophisticated U-Nets equipped with self-attention and group normalization. More recently, Transformer-based architectures like Diffusion Transformers (DiT) [25] and specialized structures for temporal data have gained prominence.

In the context of financial scenario generation, these models are attractive because the iterative denoising formulation places no parametric restriction on the distribution being learned, which makes it a flexible framework for both unconditional and conditional synthesis of extreme market regimes [36]. Whether the heavy tails and temporal dependence of that distribution are actually recovered is a separate question, and one this thesis returns to at length.

In order to perform our task, we focus on two key features of diffusion models, conditional sampling and temporal modeling. Conditional sampling allows us to generate scenarios that adhere to specific market regimes. Temporal modeling is crucial for capturing the volatility clustering and co-movement patterns especially during extreme market conditions. The two subsections below review these aspects in turn.

2.1.1. Conditional Sampling

Conditional sampling in diffusion models aims to generate samples $\mathbf{x}_0$ that satisfy specific external constraints or conditions $\mathbf{c}$, effectively modeling the posterior distribution $p(\mathbf{x}_0|\mathbf{c})$. In the domain of financial modeling, these conditions might represent specific market regimes, factor exposures, or macroeconomic indicators. Conditional sampling strategies can be broadly categorized into training-time (built-in) conditioning and inference-time (post-hoc) conditioning.

Training-time Conditioning

Built-in conditioning involves the joint learning of the data distribution and the condition during the training phase. In this approach, the denoiser $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ is explicitly dependent on the condition $\mathbf{c}$, allowing the model to learn $p(\mathbf{x}|\mathbf{c})$ directly. Achieving this typically requires architectural modifications that inject conditional information into the network. Discrete or continuous conditions are first mapped through learned embedding layers and then either concatenated with or added to the timestep embeddings; these augmented representations are injected into each residual block of the U-Net or Transformer architecture [24]. Another popular mechanism is adaptive normalization, where techniques such as Adaptive Group Normalization or Adaptive Layer Normalization scale and shift the normalized feature maps using parameters predicted from the condition, providing a more expressive way to modulate the generative process [7]. Finally, classifier-free guidance (CFG) offers a streamlined training scheme by jointly optimizing conditional and unconditional objectives, using a null token to represent the absence of a condition, and then combining the two predictions at inference time through an extrapolation factor; this improves sample quality and alignment at the cost of reduced diversity [14].

Inference-time Conditioning

Post-hoc conditioning allows constraints to be imposed on a pre-trained unconditional diffusion model $p(\mathbf{x})$ during the sampling process, obviating the need for retraining and enabling arbitrary guidance modalities. Methods in this category typically fall into two groups, soft and hard guidance. Soft guidance techniques add a penalty or guiding gradient at each reverse diffusion step to nudge the sample trajectory toward regions of high conditional probability. For example, classifier guidance trains a separate model $p(\mathbf{c}|\mathbf{x}_t)$ on the noisy data and uses the gradient $\nabla_{\mathbf{x}_t} \log p(\mathbf{c}|\mathbf{x}_t)$ to adjust the mean of the reverse step [7]. This idea can be generalized by employing the gradient of an arbitrary differentiable loss $\mathcal{L}(\mathbf{x}_t, \mathbf{c})$ or an energy-based parameterization, as proposed in recent work on loss-based and energy-based guidance [31]. Another extension, diffusion posterior sampling (DPS), makes use of manifold-constrained gradients to handle complex observation models, which is especially useful for inverse problem settings [5]. In contrast, hard guidance methods directly modify the intermediate latents $\mathbf{x}_t$ to enforce exact constraints. A straightforward tactic is replacement and masking (as in RePaint), where known or fixed regions, such as parts of an image in inpainting tasks, are substituted with their forward-diffused counterparts at each reverse step, compelling the model to fill in the remaining regions consistently [21]. Structural guidance techniques like Iterative Latent Variable Refinement (ILVR) and SDEdit further constrain generation by matching low-frequency components to a reference signal or by perturbing a guide image with noise and denoising it so that the overall structure is preserved while realistic detail is added [4, 23].

2.1.2. Temporal Modeling

Temporal modeling is a critical dimension of diffusion-based generation, particularly when dealing with time series where sequential dependencies and temporal coherence are paramount. The integration of temporal structures into the diffusion framework allows for the capture of complex dynamics that traditional autoregressive models often struggle to represent. In financial applications, this capability is essential for modeling phenomena such as volatility clustering, mean reversion, and the co-movement of multiple assets across different time scales.

The architectural evolution of temporal diffusion models has been marked by a transition from standard U-Nets to more specialized backbones designed to handle sequential data. One of the pioneering works in this area is TimeGrad [27], which employs an autoregressive recurrent neural network (RNN) to model the temporal evolution of the data distribution, with a diffusion model serving as the conditional generator at each time step. By predicting the gradient of the log-density of the next observation given the historical context, TimeGrad effectively captures the multi-modal nature of future trajectories in high-dimensional financial time series.

Beyond autoregressive approaches, conditional score-based models have also shown significant promise for structured temporal tasks. CSDI (Conditional Score-based Diffusion models for probabilistic time series Imputation) [35] represents a major advancement in this regard, utilizing a 2D-attention mechanism that simultaneously considers both the temporal dependencies and the correlations between different features or variables. This dual-attention structure allows the model to leverage information from observed values across both the time and feature dimensions to reconstruct missing data or generate future scenarios with high fidelity.

More recently, the adoption of Structured State Space Models (SSMs) and Transformer-based architectures has further enhanced the efficiency and expressive power of temporal diffusion models. SSSD (Structured State Space Diffusion) [1] replaces the traditional dilated convolutions or RNNs with state-space layers, which are particularly effective at capturing long-range dependencies while maintaining computational efficiency. Additionally, models like DiffWave [18] and its derivatives have demonstrated the utility of bidirectional temporal information and multi-scale feature extraction in generating coherent high-frequency signals. These developments collectively provide a robust toolkit for financial scenario generation, enabling the synthesis of realistic market paths that respect both short-term fluctuations and long-term structural trends.

2.2. Scenario Generation

Scenario generation is a foundational component of modern financial risk management, providing the necessary inputs for stress testing, capital allocation, and counterfactual analysis. In the context of risk management, the core objective is to synthesize a diverse set of alternative realizations, or scenarios, that accurately reflect the underlying joint distribution and temporal dependencies of market risk factors. By generating counterfactual versions of market conditions, these frameworks enable risk managers to probe the structural vulnerabilities of complex portfolios and evaluate the stability of financial systems under adverse or tail-risk regimes that are often underrepresented in historical datasets.

2.2.1. Traditional Methods

Traditional scenario generation methods are primarily divided into resampling-based techniques and parametric simulation models. Among the most influential is the Stationary Bootstrap [26], which samples blocks of random lengths following a geometric distribution to preserve the local temporal dependencies of market returns.

To bridge the gap between historical resampling and time-varying volatility, Filtered Historical Simulation (FHS) was proposed in [2]. By scaling historical residuals with current GARCH-based volatility estimates, FHS allows for the generation of scenarios that respect current market regimes while retaining the non-parametric nature of return innovation distributions.

For multi-asset modeling, Copula-based simulations are a standard industry tool for capturing non-linear dependencies. By decoupling marginal distributions from the dependence structure, these models [9] can effectively represent tail dependence and systemic risk, which are often overlooked by traditional linear correlation measures. Despite their utility, these approaches often struggle to

capture the high-dimensional complexities of global portfolios compared to modern machine-learning frameworks.

2.2.2. Generative AI Methods

The emergence of deep generative models has fundamentally shifted the frontier of scenario generation, enabling data-driven synthesis that goes beyond the parametric and resampling limitations of traditional approaches. Rather than relying on fixed distributional assumptions, these models learn the full joint distribution of asset returns from data, capturing complex non-linear dependencies and tail behaviour that classical methods cannot represent.

Early work in this direction applied VAE-based architectures to factor models. FactorVAE [8] proposes a probabilistic dynamic factor model that integrates the VAE framework with the dynamic factor model (DFM), treating latent factors as random variables to explicitly model noise. A GRU-based feature extractor encodes historical stock characteristics, and a prior-posterior learning scheme guides factor extraction. MacroVAE [19] instead focuses on conditioning generation on macroeconomic context. Using a convolutional ResNet encoder-decoder with Feature-wise Linear Modulation (FiLM), it generates 22-day multivariate return sequences conditioned on macro indicators, enabling counterfactual scenario analysis under macroeconomic conditions not observed historically.

More recently, diffusion models have been applied to cross-sectional return generation. FactorDiff [12] adapts the Diffusion Transformer (DiT) with token-wise conditioning, treating each asset as a token conditioned on its own observable factor vector via AdaLN-Zero modulation. Cross-asset dependencies are captured through multi-head self-attention. FactorDM [3] takes a different approach. Rather than conditioning on observable factors, it proposes an unconditional score-based diffusion model that embeds latent factor structure directly into the score function via time-varying orthogonal projections, decomposing the score into a low-dimensional subspace component and an idiosyncratic complement.

Table 2.1 summarises these four models across key design dimensions.

Table 2.1: Comparative analysis of generative AI models for financial scenario generation. N: assets; T: time steps; C: characteristics per asset; K: factors.

Model	Framework	Backbone	Output Shape	Condition c	Injection	Temporal
FactorVAE [8]	Probabilistic VAE	GRU + MLP	$(N,)$	Hist. characteristics $(N\times T\times C)$	GRU encoder → factor posterior	N
MacroVAE [19]	Conditional VAE	Conv ResNet2D + FiLM	$(T\times N)$	Macro indicators $c \in \mathbb{R}^{36}$	FiLM modulation in decoder	Y
FactorDiff [12]	Conditional DDPM	DiT (token-wise)	$(N,)$	Factor matrix $X_t \in \mathbb{R}^{N\times K}$	Token-wise AdaLN-Zero per asset	N
FactorDM [3]	Score-based DM	Score-decomposed MLP	$(N,)$	None (unconditional)	Latent factor in score decomp.	N

3

Methodology

Our proposed framework, FaDiff (*Factor Diffusion*), synthesises equity returns by combining two components. A parametric factor model that converts between stock returns and factor returns via fixed factor loadings and a Diffusion Transformer (DiT) that models the joint distribution of factor returns. Scenario generation starts from sampling factor returns from the DiT, and reconstruction back to stock returns given fixed factor loadings and a sample from the per-stock residual model.

The factor model is estimated once via ordinary least squares and treated as fixed throughout. The diffusion model is the only learned generative object, it captures the cross-factor co-movement, the heavy-tailed marginals of factor returns and other stylistic features of the data. The remainder of this chapter is organised as follows. Section 3.1 formalises the parametric factor model and details how loadings are estimated. Section 3.2 describes the factor diffusion model: the heavy-tailed forward process, the Gaussian-parameterised reverse process, and the Transformer-based denoiser with its training objective. Section 3.3 extends the model to temporally coherent paths by conditioning the denoiser on the previous day's cross-section and unrolling a first-order Markov chain. Section 3.4 ties the components together as algorithm boxes for training and sampling.

3.1. Parametric Factor Decomposition

The parametric factor model decomposes each stock return into a deterministic part driven by a small number of generic factors and a stochastic idiosyncratic part specific to each stock. Once the factor loadings are estimated, the only object that requires generative modelling is the joint distribution of factor returns, which lives in a much lower-dimensional space than the stock return vector itself.

The Capital Asset Pricing Model [29] states that the expected excess return of stock i is proportional to the expected excess return of the market portfolio:

$$\mathbb{E}_t[R_{i,t+1}] - R_t^f = \beta_{i,t}\left(\mathbb{E}_t[R_{t+1}^m] - R_t^f\right). \tag{3.1}$$

The CAPM is an economic model in expectations; turning it into an estimable form requires introducing a mean-zero residual $\varepsilon_{i,t+1}$ and treating β as constant within an estimation window. It has been long established that a single market factor does not fully explain the cross-section of expected returns [10], motivating the extension to a vector of K factors. The empirical multi-factor model is

$$R_{i,t+1} - R_t^f \;=\; a_i \;+\; \boldsymbol{\beta}_i^\top \mathbf{F}_{t+1} \;+\; \varepsilon_{i,t+1}, \tag{3.2}$$

where $\mathbf{F}_{t+1} \in \mathbb{R}^K$ is the realised factor return vector, $\boldsymbol{\beta}_i \in \mathbb{R}^K$ collects the per-stock loadings, a_i is a stock-specific intercept, and $\varepsilon_{i,t+1}$ captures the residual unexplained by the factors. Our proposed framework treats $\boldsymbol{\beta}_i$ and a_i as deterministic and estimated once on the training set, while the joint distribution of $\mathbf{F}$ and the per-stock distribution of ε_i are the only stochastic objects that subsequent components model.

We estimate all parts of Equation (3.2) in two sequential ordinary-least-squares (OLS) stages, following the cross-sectional regression methodology of Fama and MacBeth [11].

The first stage is a cross-sectional regression that extracts the factor returns. At each date t, the contemporaneous stock returns are regressed on the lagged firm characteristics $B_{t-1} \in \mathbb{R}^{N\times K}$:

$$R_{i,t} \;=\; f_{0,t} \;+\; \mathbf{b}_{i,t-1}^{\top}\mathbf{F}_t \;+\; u_{i,t}, \qquad i = 1,\dots,N, \tag{3.3}$$

where $f_{0,t}$ is the common intercept and $u_{i,t}$ the stock-level residual of the cross-sectional fit. This yields a single realisation $\mathbf{F}_t$ of the factor return vector for that date. Repeating this regression at every date produces the historical factor return series $\{\mathbf{F}_t\}_{t=1}^{T}$, the data on which the diffusion model is later trained. We treat the intercept $f_{0,t}$ as the market factor and append it to $\mathbf{F}_t$; Section 4.3 gives its economic interpretation. Given the factor return series, the second stage is to calculate the loadings of each individual stock via per-stock time-series OLS according to Equation (3.2).

Once the factors and factor loadings are frozen, the residual series $\{\varepsilon_{i,t}\}_{t=1}^{T}$ of each stock is treated as samples from a stationary, stock-specific distribution. Rather than assuming Gaussian residuals, which systematically underestimate equity tail risk [6], we fit a variance-normalised Student-t per stock:

$$\varepsilon_{i,t} \;=\; \sigma_i \sqrt{\frac{\nu_i - 2}{\nu_i}}\, z_{i,t}, \qquad z_{i,t} \sim t_{\nu_i}, \tag{3.4}$$

where $\sigma_i = \sqrt{\mathbb{E}[\varepsilon_{i,t}^2]}$ is the sample residual standard deviation and $\nu_i > 2$ is the degrees-of-freedom parameter, estimated by maximum likelihood on the standardised residuals $\varepsilon_{i,t}/\sigma_i$. The factor $\sqrt{(\nu_i-2)/\nu_i}$ normalises the Student-t variable to unit variance, so the tail shape is governed by ν_i and the scale by σ_i.

At sampling time a fresh $z_{i,s} \sim t_{\nu_i}$ is drawn per stock and per day, independently of the factor returns. Cross-sectional co-movement therefore comes entirely from $\boldsymbol{\beta}_i^{\top}\mathbf{F}_s$, and idiosyncratic volatility clustering is reproduced only in so far as it is transmitted through the factors.

Modelling factor returns instead of stock returns directly carries several practical advantages. First learning the joint distribution of factors rather than the full stock returns reduces the generative problem by more than two orders of magnitude. Then the decomposition is also robust to stock-level missingness that would severely undermine a model trained on all stocks jointly. In our proposed method, missing observations only affect the factor loading estimation for the affected stock, while the factor return series and the diffusion model itself remain unaffected.

Beyond these data-handling benefits, the decomposition also makes downstream usage cleaner. Conditioning the diffusion model on factor-level signals (for example, a low-momentum stress regime) is both economically meaningful and statistically efficient, because each factor aggregates information across the entire cross-section rather than being tied to a single stock. Likewise, stocks outside the training universe can be incorporated at sampling time without retraining the diffusion model. This is structurally impossible for a generative model that operates directly on the joint stock return distribution.

3.2. Factor Diffusion Model

The factor diffusion model contributes to the generative component in the proposed framework. It captures the joint distribution of factor returns $\mathbf{F}_t$, from which all stock-level scenarios are subsequently reconstructed via the projection introduced in Section 3.1. The model follows the denoising-diffusion paradigm reviewed in Section 2.1. Two design choices distinguish our implementation from a standard image-domain DDPM, each motivated by the structure of financial factor returns. First, the forward process replaces Gaussian noise with α-stable (Lévy) noise, anisotropically coloured by the empirical factor correlation matrix, so that the noising trajectory matches the fat-tailed marginal and cross-sectional dependence of the data. Second, the denoiser is a stack of Transformer blocks rather than a convolutional U-Net: multi-head self-attention naturally captures cross-factor co-movement, and AdaLN provides an efficient mechanism for injecting both the diffusion timestep and auxiliary conditioning signals.

Figure 3.1 summarises the resulting architecture. The remainder of this section describes each component in detail: the forward process (Section 3.2.1), the reverse process and conditional sampling (Section 3.2.2), and the denoiser together with its training objective (Section 3.2.3).

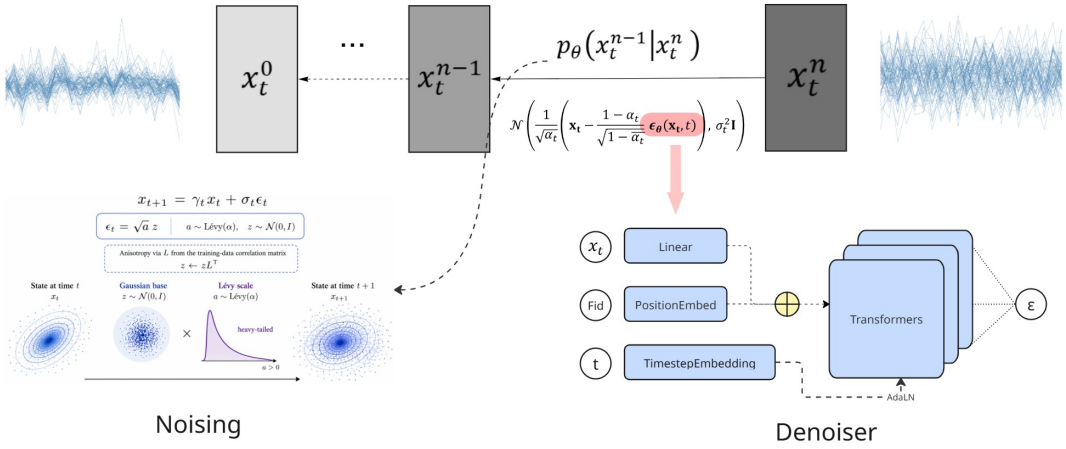


Figure 3.1: Architecture Overview of the factor diffusion model. The forward process (left) combines a Gaussian base with a Lévy scale to produce heavy-tailed perturbations of the factor return vector x_t, anisotropically coloured by the empirical factor correlation matrix. The denoiser (right) embeds the noised input x_t together with a per-factor positional embedding and the diffusion timestep t; these embeddings feed a stack of Transformer blocks whose layer normalisations are modulated by the timestep through AdaLN, and whose output predicts the noise component $\epsilon_\theta(x_t, t)$.

3.2.1. Forward Process: Heavy-Tailed Noising

The forward noising process follows the standard DDPM [13] with two modifications: the Gaussian increment is replaced by heavy-tailed alpha stable noise inspired by the Denoising Lévy Probabilistic Model (DLPM, [28]), and the Gaussian component of that noise is anisotropically coloured by the empirical factor correlation matrix following the blue-noise diffusion construction of Huang et al. [15].

Heavy-tailed α-stable forward chain. Following DLPM, we replace the Gaussian increment by a draw from the symmetric α-stable distribution $S(\alpha, 0)$, which recovers the standard normal at $\alpha = 2$ and is heavy-tailed for $\alpha < 2$:

$$\mathbf{x}_t \;=\; \gamma_t\, \mathbf{x}_{t-1} \;+\; \sigma_t\, \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \;\sim\; S(\alpha, 0). \tag{3.5}$$

The schedule coefficients are constrained to preserve the α-norm scale of the marginal, $\gamma_t^\alpha + \sigma_t^\alpha = 1$, generalising the variance-preserving DDPM constraint $a_t + \beta_t = 1$. We retain a cosine $\bar{a}_t$ schedule [24] and define $\gamma_t = a_t^{1/\alpha}$, $\sigma_t = \beta_t^{1/\alpha}$, with cumulative coefficients $\bar{\gamma}_t = \prod_{s \le t} \gamma_s$ and $\bar{\sigma}_t = (1 - \bar{\gamma}_t^\alpha)^{1/\alpha}$. Here a_t and β_t are the DDPM schedule coefficients of Section 2.1 and α is the stability index, two quantities that the usual notation would both call α. The closed-form marginal is

$$\mathbf{x}_t \;=\; \bar{\gamma}_t\, \mathbf{x}_0 \;+\; \bar{\sigma}_t\, \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim S(\alpha, 0). \tag{3.6}$$

At $\alpha = 2$, equations (3.5)–(3.6) reduce exactly to DDPM. For $\alpha < 2$ the increment $\boldsymbol{\varepsilon}_t$ has infinite variance, which precludes a direct MSE training objective. Following DLPM, we use the Gaussian scale-mixture representation of $S(\alpha, 0)$,

$$\boldsymbol{\varepsilon}_t \;=\; \sqrt{A_t}\, \mathbf{z}_t, \qquad A_t \sim S\!\left(\tfrac{\alpha}{2}, 1\right), \quad \mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3.7}$$

where the positive subordinator A_t carries the heavy-tailed scale and is sampled independently per factor, so that $A_t \in \mathbb{R}^K$ and every quantity built from it is diagonal in the factor basis. Conditional on the realised subordinator path, the marginal becomes Gaussian:

$$\mathbf{x}_t \mid \mathbf{x}_0,\, A_{1:t} \;\sim\; \mathcal{N}(\bar{\gamma}_t\, \mathbf{x}_0,\; \boldsymbol{\Sigma}_t), \quad \boldsymbol{\Sigma}_t = \sigma_t^2\, \mathrm{diag}(A_t) + \gamma_t^2 \boldsymbol{\Sigma}_{t-1}. \tag{3.8}$$

Since $\boldsymbol{\Sigma}_t$ is diagonal, the recursions below act coordinatewise across the K factors, and we write them as scalar expressions per coordinate.

Anisotropic colouring Huang et al. [15] showed for image diffusion that replacing such isotropic Gaussian noise with a correlated Gaussian, obtained by multiplying a white draw with a fixed Cholesky factor $\mathbf{L}$ of a target covariance, can substantially improve generation quality; the principle is the standard reparameterisation $\mathbf{Lz} \sim \mathcal{N}(\mathbf{0}, \mathbf{LL}^\top)$. We instantiate the same construction for factor returns by taking $\mathbf{L}$ as the Cholesky factor of the empirical factor correlation matrix $\mathbf{C} = \mathbf{LL}^\top$, computed once on the standardised training set, and colouring the Gaussian component of the scale mixture:

$$\boldsymbol{\varepsilon}_t \;=\; \sqrt{A_t}\,\mathbf{L}\,\mathbf{z}_t. \tag{3.9}$$

The subordinator A_t is still sampled independently per factor, so the heavy-tailed component stays axis-aligned while the Gaussian core respects the empirical co-movement structure. Setting $\mathbf{C} = \mathbf{I}$ recovers the original DLPM formulation; setting additionally $\alpha = 2$ recovers DDPM.

Stable training via Monte Carlo on the subordinator. In practice, the subordinator distribution $S(\alpha/2, 1)$ is itself heavy-tailed and a small fraction of draws produce values two-to-three orders of magnitude above the bulk even after the safety clip; a single such draw in a minibatch yields a loss spike that dominates the gradient and pulls the optimiser toward a degenerate constant-output solution. We therefore estimate the per-sample loss with a median-of-means scheme over independent subordinator draws [22]: at each training step we draw M outer subordinators $A^{(1)}, \ldots, A^{(M)}$ and, conditional on each, K_z inner Gaussian draws; the squared errors are averaged over the K_z inner draws and the median is taken across the M outer draws as the per-sample loss. The inner average is a Rao–Blackwell-style variance reduction on the Gaussian factor of the mixture; the outer median provides robustness to the rare extreme A.

3.2.2. Reverse Process: (Conditional) Parameterised Denoising

The forward chain has a heavy-tailed marginal but is conditionally Gaussian given the subordinator path $A_{1:T}$. We exploit this structure by parameterising the reverse process as a Gaussian denoiser conditioned on $A_{1:T}$, training it with the standard noise-prediction objective, and integrating out the subordinator at sampling time via Monte Carlo.

Conditional on $A_{1:t}$, the forward marginal in equation (3.8) is Gaussian, so the reverse posterior $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, A_{1:t})$ is also Gaussian and admits a closed form. Mirroring the DDPM derivation [13] but with the heavy-tailed scaling, one obtains

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0, A_{1:t}) \;=\; \mathcal{N}\big(\boldsymbol{\mu}_t,\; \Gamma_t\,\boldsymbol{\Sigma}_{t-1}\big)\,, \tag{3.10}$$

with posterior contraction factor and mean

$$\Gamma_t \;=\; 1 - \frac{\gamma_t^2\,\Sigma_{t-1}}{\Sigma_t}, \qquad \boldsymbol{\mu}_t \;=\; \frac{\mathbf{x}_t - \bar{\sigma}_t\,\Gamma_t\,\boldsymbol{\varepsilon}}{\gamma_t}. \tag{3.11}$$

We follow DDPM [13] and replace the unknown noise $\boldsymbol{\varepsilon}$ in (3.11) with a learned estimate $\hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t)$. The parameterised reverse step is

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t, A_{1:t}) \;=\; \mathcal{N}\!\left(\frac{\mathbf{x}_t - \bar{\sigma}_t\,\Gamma_t\,\hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t)}{\gamma_t},\; \Gamma_t\,\boldsymbol{\Sigma}_{t-1}\right). \tag{3.12}$$

The denoiser network does not need to model the subordinator: $A_{1:t}$ enters only through the schedule scalars $\gamma_t, \bar{\sigma}_t, \Sigma_t, \Gamma_t$. The training objective therefore reduces to a noise-prediction MSE, with the heavy-tailed structure folded entirely into the schedule and the subordinator sampling at inference time.

At sampling time we draw the subordinator path $A_{1:T}$ once per trajectory, accumulate $\boldsymbol{\Sigma}_t$ by the same recursion, initialise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_T)$ jointly with the same chain, and unroll equation (3.12) from $t = T$ down to $t = 1$.

To sample from a conditional distribution that biases trajectories toward a target regime $\mathbf{c}$, we encode such regimes as a differentiable loss function $\mathcal{L}(\mathbf{x}_0, \mathbf{c})$ on the clean factor return vector. Our modeling target becomes

$$p^*(\mathbf{x}_0 \mid \mathbf{c}) \;\propto\; p_\theta(\mathbf{x}_0)\,\exp\big(-s\,\mathcal{L}(\mathbf{x}_0, \mathbf{c})\big), \tag{3.13}$$

where $s > 0$ controls the strength of the bias. We approximate samples from p^* using the energy-based diffusion guidance scheme reviewed in Section 2.1.1 [31, 7]. At each reverse step we estimate $\mathbf{x}_0$ from the current $\mathbf{x}_t$ via the Tweedie identity adapted to the heavy-tailed schedule,

$$\hat{\mathbf{x}}_0(\mathbf{x}_t) \;=\; \frac{\mathbf{x}_t - \bar{\sigma}_t\, \hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t)}{\bar{\gamma}_t}, \tag{3.14}$$

and shift the posterior mean of equation (3.11) along the energy gradient,

$$\boldsymbol{\mu}_t \;\leftarrow\; \boldsymbol{\mu}_t \;-\; s\, w_t\, \Gamma_t \Sigma_{t-1}\, \nabla_{\mathbf{x}_t} \mathcal{L}(\hat{\mathbf{x}}_0(\mathbf{x}_t), \mathbf{c})\,. \tag{3.15}$$

Because the Tweedie estimate $\hat{\mathbf{x}}_0$ is unreliable at large timesteps, we set

$$w_t \;=\; \left(\frac{\mathrm{SNR}_t}{\mathrm{SNR}_t + 1}\right)^p, \qquad \mathrm{SNR}_t = \frac{\bar{\gamma}_t^2}{\bar{\sigma}_t^2}, \tag{3.16}$$

which damps the guidance update during the high-noise stage of the reverse chain. The factor $w_t \in [0, 1]$ is inspired by the Bayes-optimal denoiser weight in the EDM family [17]: $w_t \to 0$ as $t \to T$ and $w_t \to 1$ as $t \to 0$, so the guidance strength interpolates smoothly between the high-noise and low-noise regimes. The decay exponent p controls the sharpness of the transition; we use $p = 1$ throughout.

The exact target distribution p^* in (3.13) can also be sampled by soft rejection on top of the unconditional model: draw $\mathbf{x}_0 \sim p_\theta$ and accept with probability $\exp(-s\, \mathcal{L}(\mathbf{x}_0, \mathbf{c}))$. Soft rejection preserves the unconditional joint distribution exactly on the unconstrained coordinates and is the gold standard whenever the acceptance rate is tractable; gradient guidance is an inference-time approximation that becomes exact in the limit $s \to 0$ and small reverse step size [31]. We use guidance as the production sampler because of its lower compute cost.

3.2.3. Denoiser

The denoiser $\hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t)$ is a Transformer encoder operating on a sequence of factor tokens, mirroring the Diffusion Transformer (DiT) backbone [25]. We adopt this design over the convolutional U-Net used in image diffusion because multi-head self-attention treats every factor pair symmetrically and is well suited to capturing the dense cross-factor co-movement that the model is required to reproduce.

The noised input $\mathbf{x}_t \in \mathbb{R}^K$ is split into K scalar coordinates and lifted to a d-dimensional embedding by a per-coordinate linear projection. A learnable per-factor identity embedding $\mathbf{e}_k \in \mathbb{R}^d$ is added so that each token carries its factor index through the attention layers, playing the role of the positional encoding in a standard Transformer but indexing the factor identity rather than position in time:

$$\mathbf{h}_k^{(0)} \;=\; \mathbf{W}_{\mathrm{in}}\, x_{t,k} \;+\; \mathbf{e}_k, \qquad k = 1, \ldots, K. \tag{3.17}$$

Treating each factor as its own token, rather than concatenating the entire factor vector into a single token, preserves per-factor structure throughout the network and lets attention act directly on factor pairs.

The diffusion timestep t is encoded with a sinusoidal embedding followed by a small MLP, producing a continuous conditioning vector $\mathbf{c}_t$. We inject $\mathbf{c}_t$ into every Transformer block via adaptive layer normalisation (AdaLN), in which the affine parameters of LayerNorm are predicted from $\mathbf{c}_t$ rather than learned as free parameters [25, 7]. This concentrates the timestep dependence into a low-cost modulation while leaving the attention weights themselves unconditioned on t. The body of the denoiser is then a stack of such Transformer blocks; the final hidden states are projected back to scalars by a linear output head, yielding the predicted noise $\hat{\varepsilon}_{\theta,k}(\mathbf{x}_t, t)$ for each factor k.

The denoiser is trained with the standard noise-prediction MSE of [13], adapted to the heavy-tailed forward chain via the Gaussian scale-mixture representation of equation (3.7). For a sample $\mathbf{x}_0$ from the empirical factor return distribution and a uniformly drawn timestep $t \sim \mathcal{U}\{1, \ldots, T\}$, the per-sample loss is

$$\ell(\theta;\, \mathbf{x}_0, t) \;=\; \mathbb{E}_{A, \mathbf{z}}\left[\, \left\| \hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t) - \sqrt{A}\, \mathbf{L}\mathbf{z} \right\|_2^2 \right], \qquad \mathbf{x}_t = \bar{\gamma}_t\, \mathbf{x}_0 + \bar{\sigma}_t\, \sqrt{A}\, \mathbf{L}\mathbf{z}, \tag{3.18}$$

with $A \sim S(\alpha/2, 1)$ sampled per factor and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The expectation is estimated with the median-of-means scheme over subordinator draws.

3.3. Temporal Factor Generation

The model of Section 3.2 draws a single cross-section $\mathbf{F}$ from the unconditional factor distribution, with no notion of time. Most downstream uses, however, multi-day risk, path-dependent stress and path-level portfolio evaluation, require sequences of cross-sections that are temporally coherent rather than independent daily draws. We obtain such sequences by making the diffusion transformer *autoregressive*: the denoiser is conditioned on the previous day's factor cross-section, and a multi-day path is produced by unrolling a first-order Markov chain over calendar days. We index calendar days by d and write $\mathbf{F}_d \in \mathbb{R}^K$ for the factor cross-section on day d and H for the horizon; the diffusion timestep t and the noised state $\mathbf{x}_t$ retain their meaning from Section 3.2. We make the first-order Markov assumption

$$\mathbf{F}_d \;\perp\; \mathbf{F}_{<d-1} \;\mid\; \mathbf{F}_{d-1}, \tag{3.19}$$

so that the joint law of a path factorises as $p(\mathbf{F}_0, \dots, \mathbf{F}_{H-1}) = p(\mathbf{F}_0) \prod_{d=1}^{H-1} p(\mathbf{F}_d \mid \mathbf{F}_{d-1})$. Only the denoiser's conditioning input changes; the heavy-tailed forward process, the schedule and the parameterised reverse step of Section 3.2 are reused unchanged.

We augment the denoiser with the previous day's cross-section as an additional conditioning signal, written $\hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t, \mathbf{F}_{d-1})$. Following the DiT design [25], this signal enters through exactly the AdaLN pathway already used for the diffusion timestep in Section 3.2.3. The full previous cross-section is mapped to a context vector by a small MLP and combined with the per-factor identity embedding and the timestep context, so that the timestep conditioning vector $\mathbf{c}_t$ of Section 3.2.3 is replaced by a per-token condition

$$\mathbf{g}_k \;=\; \underbrace{\mathrm{MLP}_{\mathrm{cond}}(\mathbf{F}_{d-1})}_{\text{previous-day context}} \;+\; \mathbf{e}_k^{\mathrm{cond}} \;+\; \mathbf{c}_t^{\mathrm{time}}, \qquad k = 1, \dots, K, \tag{3.20}$$

where $\mathbf{c}_t^{\mathrm{time}} \equiv \mathbf{c}_t$ is the sinusoidal timestep embedding of Section 3.2.3 and $\mathbf{e}_k^{\mathrm{cond}}$ is a learnable per-factor conditioning identity, distinct from the token identity embedding $\mathbf{e}_k$ of equation (3.17). Encoding the *entire* previous cross-section, rather than each factor's own lag in isolation, lets every factor's next-day dynamics depend on the whole previous regime, so that cross-factor spillover is carried through time as well as within a day. The denoiser thereby learns the one-step transition density $p_\theta(\mathbf{F}_d \mid \mathbf{F}_{d-1})$.

This temporal condition is orthogonal to the regime condition $\mathbf{c}$ of Section 3.2.2: $\mathbf{F}_{d-1}$ enters the network as an *input* that shapes the learned transition, whereas $\mathbf{c}$ acts at inference time through the energy-gradient shift of equation (3.15). A single reverse step can therefore carry both, advancing the chain by one day while simultaneously biasing that day toward a target regime.

A purely transition-based model cannot start a path on its own, since the first day $\mathbf{F}_0$ has no predecessor. We resolve this with a learned *null* (zero) condition $\varnothing$, a single learnable vector that replaces $\mathrm{MLP}_{\mathrm{cond}}(\mathbf{F}_{d-1})$ in equation (3.20) and represents the absence of a previous day. To train both behaviours into one set of weights we use classifier-free-style condition dropout [14]: for each training pair the previous-day context is replaced by $\varnothing$ with probability p_{drop}. The denoiser then jointly learns the transition $p_\theta(\mathbf{F}_d \mid \mathbf{F}_{d-1})$ and the initial marginal

$$p_\theta(\mathbf{F}_0 \mid \varnothing) \;=\; p_\theta(\mathbf{F}_0), \tag{3.21}$$

which is exactly the unconditional factor distribution of Section 3.2. A path can thus be *self-started* from $\varnothing$ with no external seed, while the same weights still propagate any given day forward.

Generation unrolls the Markov chain of equation (3.19). We first draw $\mathbf{F}_0 \sim p_\theta(\cdot \mid \varnothing)$ from the null condition, then iterate

$$\mathbf{F}_d \;\sim\; p_\theta(\cdot \mid \mathbf{F}_{d-1}), \qquad d = 1, \dots, H-1, \tag{3.22}$$

feeding each freshly generated cross-section back as the condition for the next. Each draw in (3.22) is a complete reverse-diffusion pass over $t = T, \dots, 1$, identical to that of Section 3.2.2 except that the denoiser is queried as $\hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t, \mathbf{F}_{d-1})$; the heavy-tailed subordinator, schedule scalars and posterior step are unchanged. When a real history is available the rollout can instead be seeded with an out-of-sample day as $\mathbf{F}_0$, after which the chain proceeds identically. Because the temporal and regime conditions are orthogonal, the guidance update of equation (3.15) can be applied at every step to bias the entire trajectory toward a target regime.

Algorithm 3, stated after the pipeline in Section 3.4, summarises the resulting training and sampling changes relative to Algorithms 1 and 2; all unchanged components, the forward noising internals, the median-of-means loss estimator and the stock-level reconstruction, are reused verbatim and omitted there.

3.4. Generation Pipeline

The components developed in Sections 3.1 –3.2 compose into the two-stage pipeline that constitutes FaDiff; whenever later chapters refer to FaDiff, they refer to this pipeline as a whole. Algorithm 1 produces the fixed factor model, the diffusion schedule, the empirical correlation Cholesky factor and the trained denoiser; Algorithm 2 consumes those artifacts and emits stock-level scenarios, optionally biased toward a regime condition. For temporal generation (Section 3.3) the reverse loop of Algorithm 2 is invoked once per day with the previous day's cross-section as condition, as summarised in Algorithm 3.

Algorithm 1 FaDiff training: factor model and factor diffusion

Require: Panel data $\{(R_{i,t}, \mathbf{b}_{i,t})\}$, factor list, hyperparameters $\alpha, T, M, K_z, \ldots$

Stage 1: parametric factor model

$\forall t: \quad \mathbf{F}_t \leftarrow \arg\min_{\mathbf{F}} \sum_i (R_{i,t} - \mathbf{b}_{i,t-1}^\top \mathbf{F})^2$ (3.3)

$\forall i: \quad (a_i, \boldsymbol{\beta}_i) \leftarrow \arg\min \sum_t (R_{i,t+1} - R_t^f - a_i - \boldsymbol{\beta}_i^\top \mathbf{F}_{t+1})^2$ (3.2)

$\forall i: \quad \varepsilon_{i,t} \leftarrow R_{i,t+1} - R_t^f - a_i - \boldsymbol{\beta}_i^\top \mathbf{F}_{t+1}$

$\forall i: \quad (\sigma_i, \nu_i) \leftarrow \mathrm{MLE}\big(\{\varepsilon_{i,t}\}\big)$ (3.4)

Stage 2: factor diffusion

Standardise $\{\mathbf{F}_t\}$; $\mathbf{C} \leftarrow \mathrm{corr}(\mathbf{F})$, $\mathbf{L}\mathbf{L}^\top = \mathbf{C}$

Initialise schedule $\{\gamma_t, \sigma_t, \bar{\gamma}_t, \bar{\sigma}_t\}_{t=1}^T$ from cosine $\bar{\alpha}$ noise schedule

repeat

  Draw minibatch $\{\mathbf{F}^{(b)}\}_{b=1}^B$; $t \sim \mathcal{U}\{1, \ldots, T\}$

  $A^{(m)} \sim S(\alpha/2, 1)$, $m = 1, \ldots, M$ ▷ outer subordinator draws

  $\mathbf{z}^{(m,k)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $k = 1, \ldots, K_z$ ▷ inner Gaussian draws

  $\boldsymbol{\varepsilon}^{(m,k)} = \sqrt{A^{(m)}}\, \mathbf{L}\, \mathbf{z}^{(m,k)}$ (3.9)

  $\mathbf{x}_t^{(m,k)} = \bar{\gamma}_t\, \mathbf{F}^{(b)} + \bar{\sigma}_t\, \boldsymbol{\varepsilon}^{(m,k)}$ (3.6)

  $\ell \leftarrow \mathrm{median}_m\, \mathrm{mean}_k \left\| \hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t^{(m,k)}, t) - \boldsymbol{\varepsilon}^{(m,k)} \right\|^2$ (3.18)

  $\theta \leftarrow \theta - \eta\, \nabla_\theta \ell$

until converged

return $\{(a_i, \boldsymbol{\beta}_i, \sigma_i, \nu_i)\}_{i=1}^N$, $\hat{\boldsymbol{\varepsilon}}_\theta$, $\mathbf{L}$, $\{\gamma_t, \sigma_t, \bar{\gamma}_t, \bar{\sigma}_t\}$

Finally, Algorithm 3 records the changes needed for temporal generation (Section 3.3): a previous-day condition with dropout at training time, and a Markov-1 rollout that wraps the reverse loop of Algorithm 2 at sampling time. Only the parts that differ from Algorithms 1–2 are shown.

Algorithm 2 FaDiff sampling: factor scenarios and stock returns

Require: Trained artifacts from Algorithm 1; optional condition $(\mathbf{c}, \mathcal{L}, s, p)$

$A_t \sim S(\alpha/2, 1),\ t = 1, \dots, T$
$\Sigma_t \;=\; \sigma_t^2 A_t + \gamma_t^2 \Sigma_{t-1},\ t = 1, \dots, T$ (3.8)
$\mathbf{x}_T \;=\; \sqrt{\Sigma_T}\,\mathbf{L}\mathbf{z},\ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-1, \dots, 1$ **do**
  $\hat{\boldsymbol{\varepsilon}} \;\leftarrow\; \hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t)$
  $\Gamma_t \;=\; 1 - \gamma_t^2\, \Sigma_{t-1}/\Sigma_t$
  $\boldsymbol{\mu}_t \;=\; (\mathbf{x}_t - \bar{\sigma}_t\, \Gamma_t\, \hat{\boldsymbol{\varepsilon}})/\gamma_t$ (3.11)
  if condition supplied and $t > 1$ **then**
    $\hat{\mathbf{x}}_0 \;=\; (\mathbf{x}_t - \bar{\sigma}_t\, \hat{\boldsymbol{\varepsilon}})/\bar{\gamma}_t$ (3.14)
    $w_t \;=\; \big(\mathrm{SNR}_t/(\mathrm{SNR}_t + 1)\big)^p,\ \mathrm{SNR}_t = \bar{\gamma}_t^2/\bar{\sigma}_t^2$ (3.16)
    $\boldsymbol{\mu}_t \;\leftarrow\; \boldsymbol{\mu}_t - s\, w_t\, \Gamma_t\, \Sigma_{t-1}\, \nabla_{\mathbf{x}_t} \mathcal{L}\big(\hat{\mathbf{x}}_0, \mathbf{c}\big)$ (3.15)
  end if
  $\mathbf{x}_{t-1} \;=\; \boldsymbol{\mu}_t + \mathbf{1}_{[t>1]} \sqrt{\Gamma_t\, \Sigma_{t-1}}\,\mathbf{L}\mathbf{z},\ \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
end for
$\mathbf{F}_s \;\leftarrow$ inverse-standardise $\mathbf{x}_0$
for $i = 1, \dots, N$ **do**
  $\varepsilon_{i,s} \;=\; \sigma_i \sqrt{(\nu_i - 2)/\nu_i}\, z_{i,s},\ z_{i,s} \sim t_{\nu_i}$ (3.4)
  $R_{i,s} \;=\; a_i + \boldsymbol{\beta}_i^\top \mathbf{F}_s + \varepsilon_{i,s}$ (3.2)
end for
return $\{R_{i,s}\}_{i=1}^N$

Algorithm 3 Temporal conditioning: training and Markov-1 sampling (deltas vs. Algorithms 1–2)

Training delta (replaces the minibatch construction of Algorithm 1)
Build consecutive-day pairs $\{(\mathbf{F}_{d-1}, \mathbf{F}_d)\}_d$ from the standardised factor panel
Draw minibatch of pairs $\{(\mathbf{F}_{\text{prev}}^{(b)}, \mathbf{F}^{(b)})\}_{b=1}^B;\ t \sim \mathcal{U}\{1, \dots, T\}$
$\forall b:$ with prob. p_{drop} set $\mathbf{F}_{\text{prev}}^{(b)} \leftarrow \varnothing$ ▷ condition dropout [14]
Form $\mathbf{x}_t^{(b)}$ as in Algorithm 1; condition the denoiser on $\mathbf{F}_{\text{prev}}^{(b)}$ via (3.20)
$\ell \leftarrow \text{median}_m\, \text{mean}_k \left\| \hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t^{(m,k)}, t, \mathbf{F}_{\text{prev}}) - \boldsymbol{\varepsilon}^{(m,k)} \right\|^2$ (3.18)
Sampling delta (Markov-1 rollout, wraps Algorithm 2)
function Reverse($\mathbf{F}_{\text{cond}}$) ▷ lines 1–14 of Algorithm 2, denoiser $= \hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t, \mathbf{F}_{\text{cond}})$
  return $\mathbf{x}_0$
end function
$\mathbf{F}_0 \leftarrow$ Reverse($\varnothing$) ▷ self-start from the null condition (3.21)
for $d = 1, \dots, H-1$ **do**
  $\mathbf{F}_d \leftarrow$ Reverse($\mathbf{F}_{d-1}$) (3.22)
end for
return path $\{\mathbf{F}_d\}_{d=0}^{H-1}$ ▷ reconstruct stock returns per day as in Algorithm 2

4

Experiments

This chapter fixes the empirical setup. We first introduce the two datasets on which FaDiff is evaluated. The primary dataset is the MSCI World index at daily frequency, which carries the full pipeline and every experiment of the evaluation. As a complement, we introduce a self-built dataset of Dow 30 bars at minute frequency. We then introduce the two non-learning generators, historical resampling and a parametric Gaussian, report the denoiser architecture, training and sampling hyperparameters, and finally define the evaluation protocol and the metrics used at the factor and stock levels.

4.1. Datasets

The empirical analysis rests on two datasets. The MSCI World index at daily frequency is the primary one and carries every experiment in this thesis; And Dow 30 bars serve as a complementary dataset which has smaller stock coverage but more observations. The two differ in sampling frequency, by roughly two and a half orders of magnitude, and in the stock pool. On the MSCI dataset stocks are compressed into seven factors, whereas the thirty Dow tickers are modelled directly, without a factor decomposition. We use simple returns on both datasets.

MSCI World Index. The primary dataset is the constituent universe of the MSCI World Index, an index of large and mid cap stocks across 23 developed markets, weighted by free-float market capitalisation and covering roughly 85% of it in each country.[1] For each constituent we obtain daily total returns in USD and six MSCI Generic Factor exposures, cross-sectionally standardised stock-level scores produced by MSCI's Global Equity Model for the most widely used style factors. These exposures are the firm characteristics $\mathbf{b}_{i,t}$ that enter the cross-sectional regression of Equation (3.3).

The panel spans from 2001-01-01 to 2025-12-31, covering 6523 trading dates and 3942 unique equities across the Americas, EMEA and Asia-Pacific and all eleven GICS sectors. The cross-section grows from roughly 1000 stocks in 2001 to above 2000 after 2020, reflecting both the expansion of the developed-market universe and successive MSCI inclusion-rule updates. Figure 4.1 shows this evolution and confirms that the six factors are close to orthogonal in our working sample. Each cross-sectional distribution is approximately symmetric between -3 and +3, and the largest time-averaged pairwise correlation, between quality and volatility, is around 0.33, with every other pair inside 0.2 in absolute value.

Roughly 0.74% of the factor cells are missing in the raw panel, typically at the start of a firm's listing window or around corporate actions. We forward-fill within firm along the time dimension and then apply cross-sectional median imputation at each date for any cell that remains. Because the exposures are already winsorised and standardised at source, no further outlier treatment is applied. We retain the last calendar year which contains 261 trading days as the out-of-sample set and train on the remaining 24 years which contains 6262 trading days. The split is purely temporal, and 1924 of the 1979 out-of-sample securities also appear in training, so the out-of-sample comparison tests new returns for mostly known

[1] MSCI World Index Methodology, `https://www.msci.com/documents/10199/178e6643-6ae6-47b9-82be-e1fc565ededb`.

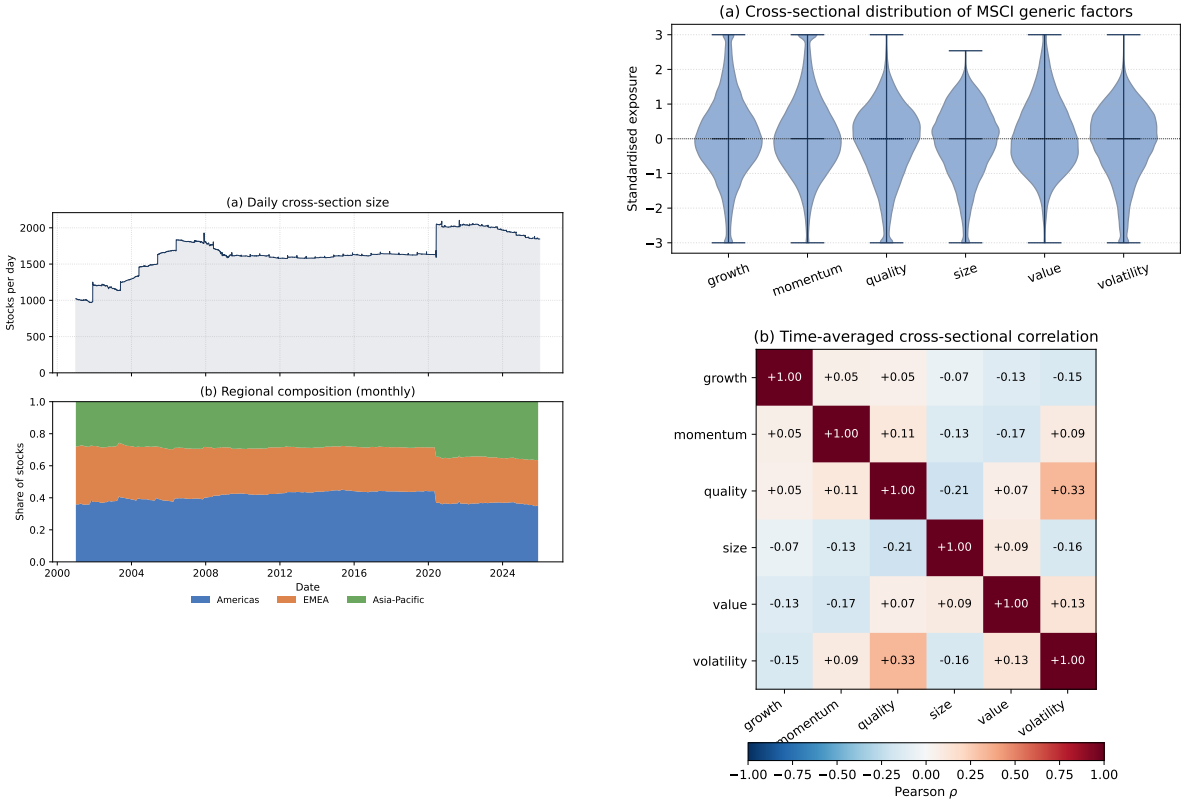


Figure 4.1: MSCI World working panel and factor exposures. **Left:** (a) daily cross-section size and (b) monthly regional composition over 2001-01-01 to 2025-12-31. **Right:** (a) cross-sectional distribution of the six MSCI generic factor exposures and (b) their time-averaged cross-sectional correlation matrix.

stocks rather than unseen firms. The stock-level evaluation uses only these 1924 stocks, aligned so that every comparison is made on the same firm in both windows.

Dow 30. The second dataset is one we build ourselves. The MSCI dataset is a commercial product behind a paid subscription, so results computed on it cannot be verified independently. We therefore build the Dow 30 panel from a public data source and release the scripts that construct it at `https://github.com/yryMax/TailEnhancedFinDiff`, so that at least one of the two evaluations in this thesis can be fully reproduced by anyone.

The dataset consists of minute-level bars for the thirty constituents of the Dow Jones Industrial Average, retrieved from the Polygon.io aggregates endpoint with split and dividend adjusted prices, covering 2024-07-08 to 2026-07-07 and 501 trading days. Within each day we keep the fixed window from 10:00 to 11:59 ET, 120 closes per ticker and day. The chosen window yields a cross-section observed synchronously for all thirty tickers; it contains the most heavily traded part of the session and excludes the opening half-hour, where trading behaves unlike the rest of the day.

In the raw data 0.58% of the grid is missing, because no trade occurs in that minute. We forward-fill these gaps within ticker and day, and set the few cells that remain to the previous day's closing price. We then compute one-minute simple returns within each ticker and trading day, so every ticker contributes 119 observations per day and no return spans a day boundary. The panel therefore forms a sequence of 501 self-contained intraday segments rather than one continuous series. We hold out the last 50 trading days (5950 samples) as the out-of-sample set and train on the preceding 451 (53669 bars).

Table 4.1 gives an overview of the two datasets.

Table 4.1: MSCI and Dow 30 dataset overview.

Property	**MSCI World**	**Dow 30**
Frequency	Daily	Minute
Sample Period	2001-01-01 to 2025-12-31	2024-07-08 to 2026-07-07
Size	6523	59619
Equities	3942	30
Coverage	23 developed markets	US
Missing Rate	0.74%	0.58%
Standard Deviation	0.0023	0.00075
Median Kurtosis	6.3	15.2
Median Skewness	-0.20	0.02
Mean Correlation	-0.04	0.23
Train/Test Split	≈96/4	≈90/10

4.2. Baselines

We restrict the comparison to non-learning generators rather than including third-party deep generative models, because embedding the latter in our pipeline would require non-trivial design choices and further adaptation to our datasets. The two baselines we retain are instead the canonical reference points for financial scenario generation, namely historical resampling and multivariate Gaussian modeling. Historical resampling is a fully non-parametric generator that reproduces the empirical marginals exactly but cannot extrapolate beyond observed history; the Gaussian generator extrapolates smoothly but assumes thin-tailed, elliptically symmetric co-movements.

Historical resampling (Resample). The first baseline is the empirical bootstrap. It admits two implementations, one resampling the factors and the other resampling the equity returns directly. We use each at the level where it is the natural reference for the quantity being compared.

At the factor level, an unconditional cross-sectional draw picks a day uniformly at random, with replacement, from the standardised historical factor-return series. This reproduces the historical marginal and contemporaneous factor dependence exactly, but by construction it can never generate a factor configuration that was not observed in the training window. For multi-day paths we use the stationary bootstrap of Politis and Romano [26]. A path is grown one step at a time by advancing the current index cyclically, and at each step restarting from a uniformly random date with probability 0.125,

so the resulting blocks have a geometric length with a mean of eight trading days, which preserves short horizon autocorrelation and volatility clustering while keeping the resampled path stationary. This is the variant used wherever the comparison itself is made in factor space, and it is a drop-in replacement for the diffusion factor sampler in the pipeline.

At the stock level we use the same resampling principle, but on individual equities rather than factors. The object of interest is the return of an individual stock, and the corresponding non-parametric reference is resampling the returns themselves. This variant bypasses the factor decomposition and the Student-t residual model. It reproduces each stock's own historical marginal, and the contemporaneous dependence across stocks. It is therefore a deliberately demanding reference at that level, stronger than what the shared pipeline could produce, and it is the sampler reported as Resample in the stock-level evaluation.

Parametric Gaussian (Gaussian). The second baseline replaces the empirical distribution with a multivariate normal whose mean and covariance are the sample mean and sample covariance of the training factor returns. This is the Monte Carlo counterpart to historical resampling. It can produce factor configurations never seen in the data, but only along the directions of an elliptically symmetric, thin tailed Gaussian, which is exactly the assumption shown to understate equity tail risk [6]. Because real volatility clusters in time, for multi-day paths we let the covariance evolve through the exponentially weighted moving average [16],

$$\boldsymbol{\Sigma}_t \;=\; \lambda\,\boldsymbol{\Sigma}_{t-1} \;+\; (1-\lambda)\,\mathbf{d}_t\mathbf{d}_t^{\top}, \qquad \mathbf{d}_t = \mathbf{F}_t - \hat{\boldsymbol{\mu}}, \tag{4.1}$$

with decay 0.94 and the recursion warm-started at the sample covariance. Each path step is drawn from a normal distribution with the current covariance, giving the baseline a GARCH-like time-varying second moment while keeping every conditional distribution Gaussian.

Conditional sampling. Both baselines admit the same regime-conditioning interface as FaDiff, so the conditional evaluation is well defined for all three. Lacking a reverse diffusion process to steer, the baselines realise the conditioning loss by rejection sampling. A candidate cross-section is generated from the unconditional sampler, standardised with the shared scaler, and accepted with a probability that decays exponentially in the conditioning loss, at the same guidance scale as the diffusion model. This targets the same reweighted distribution that the guided diffusion approximates, which keeps the conditional comparison on a common footing.

4.3. Implementation Details

Table 4.2 collects our implementation configuration on both datasets, and the following paragraphs describe it component by component.

Denoiser. We adopt a medium complexity Transformer denoiser. Each cross-sectional unit enters as a separate token, seven factors on the MSCI World data and thirty tickers on the Dow 30 data, followed by six Transformer blocks with eight attention heads each and an AdaLN layer that injects the diffusion timestep. Token and timestep embeddings are 128 dimensional. Only the number of tokens differs between the two datasets, since the Dow panel is modelled without a factor decomposition and its assets are the cross-sectional units themselves.

Training. Training operates on standardised inputs. A scaler fit on the training set normalises each series to zero mean and unit variance and is reused at sampling time. Optimisation uses AdamW with weight decay and gradient clipping at norm 1.0; both have little effect in well-behaved batches but absorb the occasional large gradients produced by heavy-tailed noise. The learning rate is 0.0001 throughout. Epoch count and batch size are the two settings that scale with the data. The Dow panel carries roughly nine times as many observations along the time axis as the MSCI one, so far fewer passes over it are needed and a larger batch is affordable. We use 200 epochs at batch 64 on MSCI against 40 epochs at batch 256 on Dow.

Scheduling. The noise schedule is the cosine schedule of [24], run over 100 diffusion steps. Clipping the subordinator to the range 0 to 2000 keeps rare astronomical draws from corrupting the schedule

arithmetic. The stability index alpha preserves the extreme moves characteristic of the target, at 1.9 on the MSCI data and 1.95 on the Dow data.

Loss. The Lévy training loss is estimated by a median-of-means scheme, again for stability since a single subordinator draw can be extreme. On the MSCI data we use five outer and five inner Monte Carlo draws per minibatch; on the Dow data two and three suffice, because the larger batch already averages the subordinator draw over four times as many samples within a single step.

Sampling. At sampling time we generate 4096 cross-sections per run on both datasets. Conditional sampling adds a guidance gradient whose strength decays with the per-step signal-to-noise ratio, so guidance acts strongly when there is still signal to steer and fades out near the noise floor. It is used at scale 5.0 with SNR weight 1.0 on the MSCI data and does not arise on the Dow data, which carries no conditional experiment.

Table 4.2: Diffusion model hyperparameters on the two datasets. A dash means the Dow 30 setting is identical to the MSCI one.

Group	Parameter	MSCI World	Dow 30
Denoiser	Input tokens	7 factors	30 tickers
	Token embedding dim d	128	–
	Number of Transformer blocks	6	–
	Attention heads	8	–
	Timestep embedding dim	128	–
Training	Optimiser	AdamW	–
	Learning rate	1×10^{-4}	–
	Weight decay	1×10^{-4}	–
	Epochs	200	40
	Batch size	64	256
Scheduling	Diffusion steps T	100	–
	Noise Scheduler	Cosine	–
	Stability index α	1.9	1.95
	Subordinator clip	$A_t \in [0, 2000]$	–
Loss	Outer MC draws M	5	2
	Inner MC draws K_z	5	3
Sampling	Samples per evaluation	4096	–
	Conditional guidance scale s	5.0	n/a
	SNR weight p	1.0	n/a

The Dow 30 dataset is used for the unconditional evaluation only. It cannot support the temporal evaluation for three reasons. Each day contributes only a fixed two-hour window, so a generated path does not correspond to a trading day. The 501 daily segments are separated by overnight gaps and do not form one continuous series. And temporal statistics depend on the sampling horizon, so values measured in minutes cannot be compared with values measured in days. It cannot support the conditional evaluation either, simply because guidance conditions on factor-level signals and this panel has no factors.

To obtain the factor return time series consumed by the diffusion model, we apply two standard estimators in parallel, a cross-sectional regression and a univariate portfolio sort. Figure 4.2 reports the resulting cumulative factor returns under both methods.

All following evaluation uses the regression-based factor returns. A final relabelling is applied. The cross-sectional intercept has the economic interpretation of the equally-weighted return left after removing the six style factors. We rename it the market factor, which is how the original six MSCI generic factors become the seven factors used as model inputs throughout this thesis.

4.4. Evaluation Metrics

We assess generation quality at two levels, the low-dimensional factor returns and the high-dimensional stock cross-section, and along the time dimension of individual stock paths. This section defines the metric used at each level together with the reason it is included. Every metric is either widely adopted in

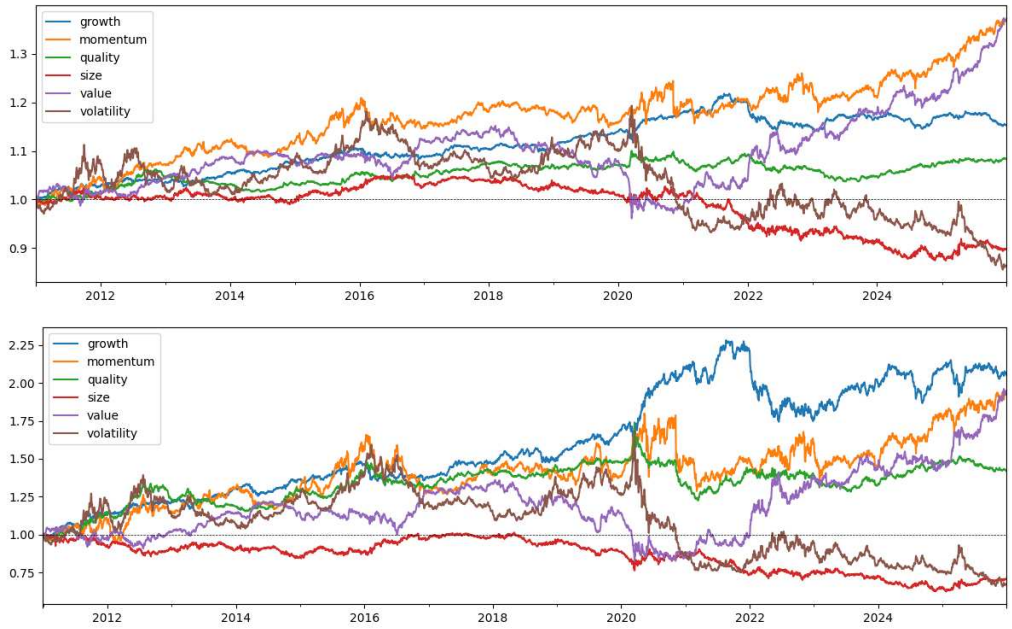


Figure 4.2: Cumulative factor returns over 2001 to 2025 under the cross-sectional regression estimator (top) and the quintile portfolio-sort estimator (bottom).

the distributional comparison and financial econometrics literature, or directly motivated by a stylized fact of equity returns [6] such as heavy tails or negative skew. All formulas are collected in Table 4.3.

At the factor level we begin with the Mahalanobis distance. Relative to a reference mean and covariance, a draw's squared Mahalanobis distance would follow a known chi-squared law if the generated factors were multivariate Gaussian, so a QQ plot of the ordered distances against those quantiles tests joint normality. Because equity returns are heavy-tailed [6], the distances should curve above the diagonal in the right tail, and a generator that keeps Gaussian-like joint tails would instead sit on the diagonal, which is where the elliptical Gaussian baseline is expected to break down.

Each factor's marginal distribution is then summarised by its first four moments, the mean for location, the standard deviation for dispersion, the skewness for the downside asymmetry typical of equity factors, and the excess kurtosis for the fat tails that are a defining stylized fact [6] and, because four moments do not pin down the full shape, by the Wasserstein-1 distance, the integrated absolute difference of two CDFs, which responds to the entire distribution rather than to any finite set of moments. Cross-factor dependence is captured by the correlation matrix. Co-movement drives diversification and joint tail risk, so a generator can match every marginal yet remain useless if it distorts the dependence structure. Throughout the thesis dependence is always compared on correlation matrices, never on raw covariances, so that the comparison is free of any scale mismatch between two samples. The whole-matrix discrepancy is collapsed into a single number in three ways. The Frobenius norm of the difference weights large entries most heavily; the mean absolute error over the off-diagonal entries weights every factor pair equally; and the relative (scale-free) version of either is obtained by normalising with the same norm of the reference matrix. We state which of the three is used at each point of the evaluation.

At the stock level the cross-section is summarised rather than inspected name by name. The same four time-series moments are computed per stock, with the standard-deviation panel the key check that individual stock volatilities are reproduced rather than averaged away by aggregation. A per-stock Kolmogorov-Smirnov test yields a p-value for each stock; the KS test is a widely adopted nonparametric test of whether two samples share a distribution, and aggregating across stocks turns it into a population statement, since under the null of identical distributions the p-values are uniform between zero and one. A histogram piling up near zero exposes systematic mismatch, and the fraction above 0.05 counts how many stocks pass. Finally, because each return decomposes into a systematic factor component and an idiosyncratic residual, the cross-sectional dispersion is split accordingly and, per

component, summarised by a dispersion ratio and the Wasserstein-1 distance between two dispersion series. Separating systematic from idiosyncratic dispersion is standard practice in cross-sectional asset pricing [10] and makes the diagnosis actionable. An inflated cross-section can be localised to factor returns or loadings that are too volatile, or to an overstated residual variance, which a single aggregate error could never separate.

The temporal generation task is judged along the time dimension of each simulated path, where the benchmarks are the stylized facts of return dynamics rather than the shape of any single marginal. We adopt the evaluation suite of Kubiak et al. [19], computing each metric per scenario per asset on the series and averaging over the scenarios and assets. The first three measures test the autocorrelation stylized facts catalogued by Cont [6]. The return autocorrelation gauges linear predictability. Efficient-market returns carry no exploitable serial correlation, so a faithful generator keeps it near zero. The volatility-clustering measure is the autocorrelation of absolute returns, which is positive and slowly decaying in real markets because large moves cluster in time. The coarse-fine measure is the lead-lag cross-correlation between coarse and fine grained volatility; its asymmetry encodes the volatility-cascade effect, whereby long-horizon volatility predicts short-horizon volatility more strongly than the reverse, and is expected to be positive. The variance ratio of Lo and MacKinlay [20] contrasts the realised variance of the multi-day aggregated return with the random-walk prediction that variance grows linearly in the horizon. A value of one signals a memoryless walk, above one momentum or trend, and below one mean reversion. Following the convention of Lo and MacKinlay [20], the aggregate is the sum of the daily returns rather than their compounded product; over the horizons considered here the two differ by less than 0.003 in the reported ratio. Finally the maximum drawdown records the largest peak-to-trough loss along the cumulative-return path, a path-dependent tail measure that risk managers care about directly and that no marginal moment can recover.

Table 4.3: Evaluation metrics used in this thesis.

Metric	**Formula**
Marginal moments	$\bar{x} = \frac{1}{n}\sum_i x_i, \quad \sigma = \sqrt{\frac{1}{n}\sum_i (x_i - \bar{x})^2}, \quad \gamma_1 = \mathbb{E}\Big[\big(\frac{x-\bar{x}}{\sigma}\big)^3\Big], \quad \gamma_2 = \mathbb{E}\Big[\big(\frac{x-\bar{x}}{\sigma}\big)^4\Big] - 3$
Wasserstein-1	$W_1(P,Q) = \int_{-\infty}^{\infty} \big\lvert F_P(x) - F_Q(x)\big\rvert \, dx$
Mahalanobis Q-Q	$d^2(\mathbf{F}) = (\mathbf{F} - \hat{\boldsymbol{\mu}})^\top \hat{\boldsymbol{\Sigma}}^{-1} (\mathbf{F} - \hat{\boldsymbol{\mu}})$, ordered against χ^2_K quantiles
Correlation error	$\rho_{ij} = \dfrac{\mathrm{Cov}\big(F^{(i)}, F^{(j)}\big)}{\sigma_i \sigma_j}, \qquad \boldsymbol{\Delta} = \boldsymbol{\rho}_P - \boldsymbol{\rho}_Q$; reported as one of (i) $\lVert\boldsymbol{\Delta}\rVert_F = \Big(\sum_{i,j} \Delta_{ij}^2\Big)^{1/2}$, relative form $\mathrm{Err}_F = \lVert\boldsymbol{\Delta}\rVert_F \big/ \lVert\boldsymbol{\rho}_Q\rVert_F$ (ii) $\mathrm{Err}_{\mathrm{mean}} = \dfrac{2}{K(K-1)} \sum_{i<j} \big\lvert\Delta_{ij}\big\rvert$ (off-diagonal mean absolute error) (iii) $\mathrm{Err}_{\mathrm{MAE}} = \mathrm{Err}_{\mathrm{mean}} \Big/ \dfrac{2}{K(K-1)} \sum_{i<j} \big\lvert\rho_{Q,ij}\big\rvert$ (relative MAE)
Kolmogorov-Smirnov	$D = \sup_x \big\lvert\hat{F}_P(x) - \hat{F}_Q(x)\big\rvert$; report the distribution of $\{p_n\}$ and $\frac{1}{N}\sum_n \mathbb{1}[p_n > 0.05]$
Dispersion comparison	$\mathrm{CSD}_t^{(c)} = \mathrm{std}_n\, r_{n,t}^{(c)}$, $r_{n,t}^{(\mathrm{sys})} = \boldsymbol{\beta}_n^\top \mathbf{F}_t$, $r_{n,t}^{(\mathrm{idio})} = \varepsilon_{n,t}$; report $R^{(c)} = \overline{\mathrm{CSD}}_P^{(c)} / \overline{\mathrm{CSD}}_Q^{(c)}$ and W_1 of the $\{\mathrm{CSD}_t^{(c)}\}$ series, $c \in \{\mathrm{total}, \mathrm{sys}, \mathrm{idio}\}$
Autocorrelation (ACF)	$\mathrm{ACF}(k) = \mathrm{Corr}\big(r_t,\, r_{t-k}\big)$; report the lag-averaged $\frac{1}{K}\sum_{k=1}^{K} \mathrm{ACF}(k)$, $K = 10$
Volatility clustering	$\mathrm{VolClust}(k) = \mathrm{Corr}\big(\lvert r_t\rvert,\, \lvert r_{t-k}\rvert\big)$; report the lag-averaged value, $K = 10$
Coarse-fine	$c_t = \Big\lvert \sum_{s=t-\tau+1}^{t} r_s \Big\rvert, \quad f_t = \sum_{s=t-\tau+1}^{t} \lvert r_s\rvert, \quad \overline{\mathrm{CF}} = \frac{1}{K}\sum_{k=1}^{K} \mathrm{Corr}\big(c_{t+k},\, f_t\big), \quad \tau = 5$
Variance ratio	$\mathrm{VR}(h) = \dfrac{\mathrm{Var}\big(\sum_{i=1}^{h} r_{t-i+1}\big)}{h\ \mathrm{Var}(r_t)}$; reported in standardized form $\sqrt{\mathrm{VR}(h)} = \dfrac{\sigma(h)}{\sqrt{h}\,\sigma(1)}$, with $\sigma(h)$ the standard deviation of the sum of h consecutive daily returns and value one under a random walk
Maximum drawdown	$V_t = \prod_{s \le t}(1 + r_s), \qquad \mathrm{MDD} = \max_t \dfrac{\max_{s \le t} V_s - V_t}{\max_{s \le t} V_s}$

5

Results and Evaluation

This chapter evaluates FaDiff against the two baselines on the out-of-sample year. Throughout, FaDiff denotes the full pipeline with the diffusion factor sampler. Depending on the section, figure legends label FaDiff as "Generated", "Diffusion" or "Ours".

The evaluation proceeds in four steps. We first test whether the unconditional samples reproduce the statistical structure of the out-of-sample data, at the factor level and, after reconstruction, at the stock level. We then put the generator into prescribed stress regimes and compare the conditional samples against historical stress days. Next we roll the generator out over multiple days and check the temporal stylized facts along the generated paths. Finally we apply the full pipeline to a stress-testing exercise on a benchmark and three live strategy portfolios. The Dow 30 dataset appears in the unconditional evaluation only, for the data reasons given in the previous chapter; every other experiment runs on the MSCI World.

5.1. Unconditional Cross-Sectional Generation

We begin by asking whether FaDiff reproduces the unconditional statistical structure of the factor cross-section. The marginal and joint distributions it produces have to match those of the real returns. Throughout we compare the generated sample against two baselines, an empirical Resample of the historical cross-section and an elliptical Gaussian fit, all evaluated on the same out-of-sample (OOS) period.

We organise the comparison around the per-factor marginals, first through their first four moments and then through full densities and Wasserstein distances, followed by the cross-sectional correlation structure that ties the seven factors together. Two supporting distributional diagnostics, a Mahalanobis check on the joint tail and per-factor quantile plots, are reported in Appendix A.1.

5.1.1. Statistical Moments

We first summarise each factor's marginal distribution by its first four moments. Figure 5.1 plots, for each of the seven factors, the moment of the generated and baseline samples against the corresponding value during the OOS period, so that a point on the dashed diagonal denotes an exact match and the per-factor spread of each method is visible directly. Table 5.1 then aggregates this comparison into a single mean absolute error (MAE) across factors, with the across-factor standard deviation in parentheses.

The three methods coincide on the mean and the standard deviation and separate only on the higher moments, which is where the heavy tails and downside asymmetry of equity returns live. Location and scale are therefore a sanity check that every method clears rather than a point of discrimination, and the comparison that matters is the third and fourth moments. Appendix A.1 verifies that the MSCI World factor returns are materially heavier-tailed than a Gaussian.

On both, the figure shows the Gaussian baseline collapsing onto the zero line irrespective of the OOS

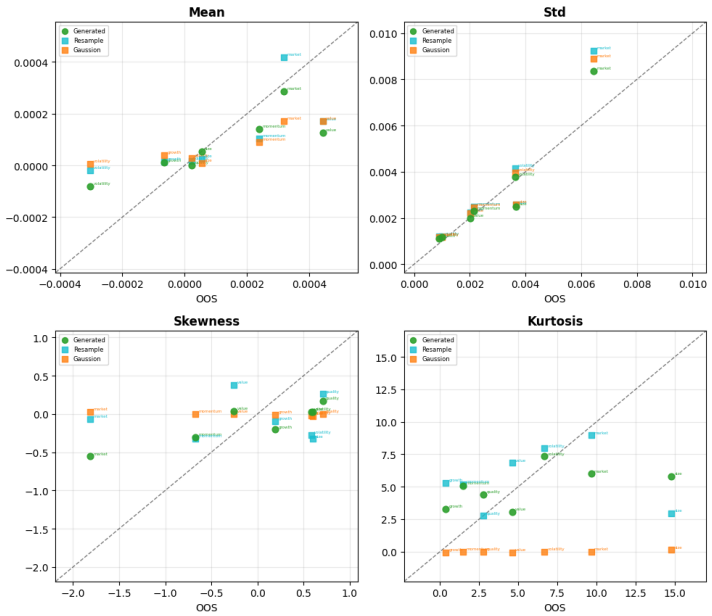


Figure 5.1: Marginal moment comparison at the factor level. Each panel plots one moment (mean, standard deviation, skewness, excess kurtosis) of the FaDiff (*Generated*, green) and baseline (Resample, Gaussian) samples on the vertical axis against the OOS value on the horizontal axis, one point per factor; points on the dashed diagonal match the OOS moment exactly.

Table 5.1: Per-moment MAE (↓, lower is better) across the seven factors between each method and the OOS sample, as mean (standard deviation) over factors. Row-best in bold.

	FaDiff ↓	Resample ↓	Gaussian ↓
Mean	**0.0001** (0.0001)	**0.0001** (0.0001)	**0.0001** (0.0001)
Std	**0.0005** (0.0007)	0.0008 (0.0009)	0.0007 (0.0008)
Skew	**0.5687** (0.2966)	0.7504 (0.4658)	0.7006 (0.4994)
Kurt	**3.2859** (2.5528)	3.5267 (3.7421)	5.7692 (4.6579)

value. An multivariate Gaussian is symmetric and thin-tailed by construction, so it cannot represent a non-zero skewness or a positive excess kurtosis at all. The generated sample instead rises with the OOS value across factors, recovering both the sign of the asymmetry and the fattening of the tails; this is the direct payoff of the heavy-tailed forward process, which lets the model place mass in regions an elliptical-Gaussian baseline can never reach.

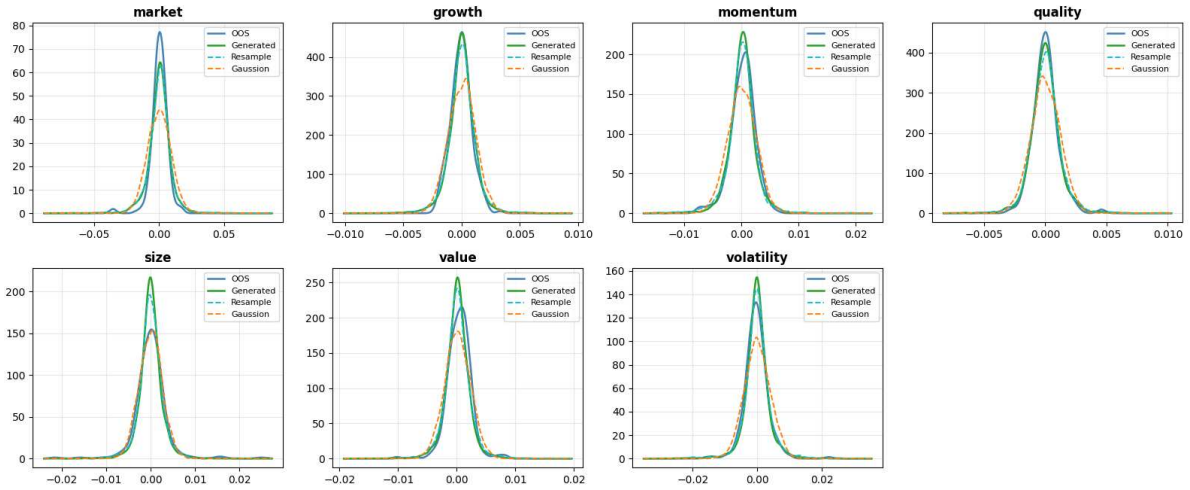
FaDiff wins on all four metrics. The across-factor standard deviations for skewness and excess kurtosis are of the same order as the errors themselves, so the higher-moment fit is far less uniform across the seven factors than the location and scale fit. Relative to the mean and the standard deviation, which every method recovers almost exactly, the third and fourth moments are only approximately matched, reflecting how much harder the tail of the cross-section is to model than its centre.

5.1.2. Marginal Distributions

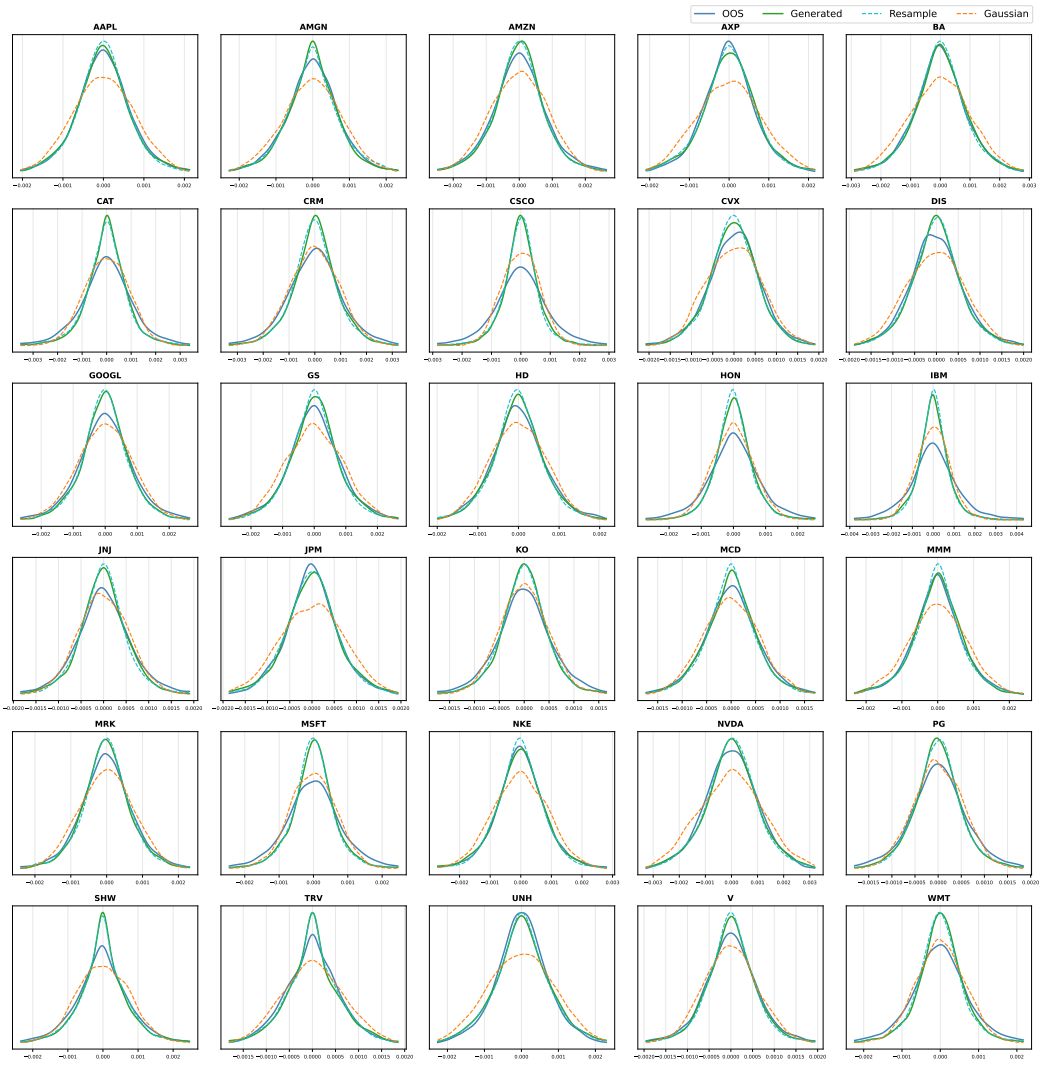
Figure 5.2 shows for every factor, a Gaussian kernel-density estimate of the OOS sample against the generated and the two baseline samples, so that the location, the peak and the tails of each marginal can be compared at once. Table 5.2 then reduces each overlay to a single number, the Wasserstein-1 (W_1) distance to the OOS marginal.

Across the seven factors the generated density sits almost on top of the OOS density. It matches the mode, the width and the asymmetry of the shoulders, which is the marginal counterpart of the higher-moment fit above. The Gaussian baseline, by contrast, is systematically over-concentrated, it places too much mass at the centre and too little in the shoulders, exactly the failure expected from a thin-tailed elliptical fit. The clearest case is the market factor, whose OOS marginal is the most heavy-shouldered of the seven. There the Gaussian peak overshoots the OOS density while the generated and Resample curves stay close to it.

The Wasserstein-1 distances make this comparison quantitative. The generated sample attains the lowest distance on five of the seven factors and ties on the rest, giving the best across-factor average. The gap is widest precisely where the marginals are hardest. On market the generated distance (0.0017) is roughly half the Gaussian one (0.0032), confirming that the heavy-tailed forward process pays off most on the heavy-shouldered factor, while on the near-Gaussian factors (growth and quality) all three methods are



(a) MSCI World, one panel per factor.



(b) Dow 30, one panel per ticker, drawn over the OOS 1 to 99% range.

Figure 5.2: Marginal densities on both panels. Each subplot shows a Gaussian kernel-density estimate of the OOS (blue), FaDiff (*Generated*, green), Resample (dashed) and Gaussian (dashed) samples for one cross-sectional unit.

Table 5.2: Wasserstein-1 distance (W_1, ↓ lower is better) between each method's marginal and the OOS marginal, per factor and averaged across factors. Row-best in bold.

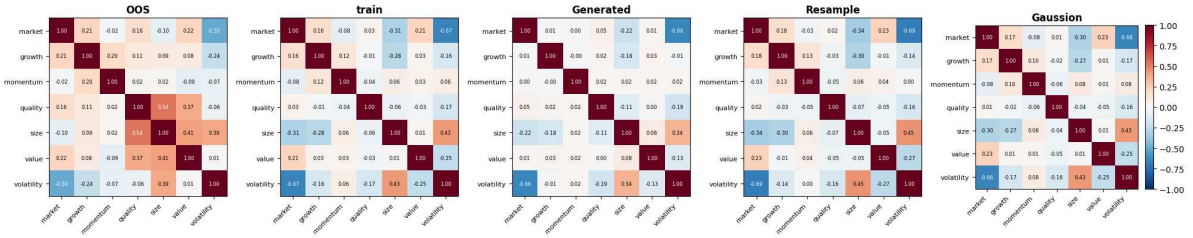
Panel	Series	FaDiff ↓	Resample ↓	Gaussian ↓
MSCI World	market	**0.0017**	0.0019	0.0032
	growth	**0.0002**	**0.0002**	**0.0002**
	momentum	**0.0003**	**0.0003**	0.0005
	quality	**0.0001**	0.0002	0.0002
	size	0.0005	**0.0004**	0.0005
	value	**0.0003**	0.0004	0.0005
	volatility	**0.0005**	**0.0005**	0.0008
	Average	**0.0005**	0.0006	0.0008
Dow 30	Average over 30 assets	$\mathbf{9.35 \times 10^{-5}}$	9.75×10^{-5}	1.030×10^{-4}

essentially indistinguishable.

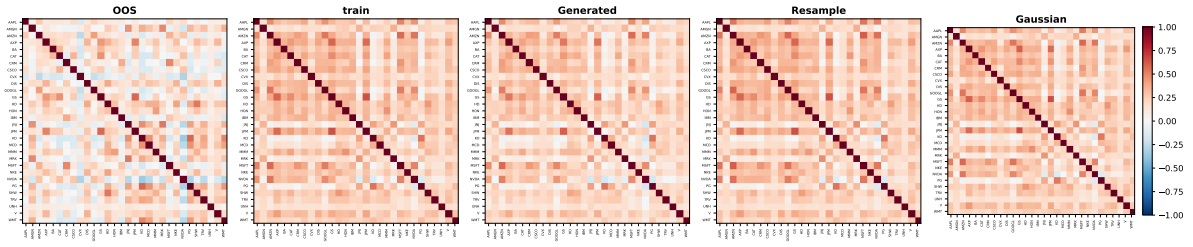
On the dow 30 dataset the three methods are much closer together. FaDiff has the lowest average distance, but it is best on twelve of the thirty tickers against twelve for the Gaussian baseline. This is because One-minute returns are close to symmetric, with a median skewness of 0.02, so there is little asymmetry for a heavy-tailed generator to recover that an elliptical fit would miss.

5.1.3. Factor Correlations

We now turn to the cross-sectional correlation structure to show the linear dependence between factors. Figure 5.3 shows the correlation heatmaps for the OOS and training samples next to those of the three methods, and Figure 5.4 splits each matrix into its 21 off-diagonal entries and plots them against the corresponding OOS correlations.



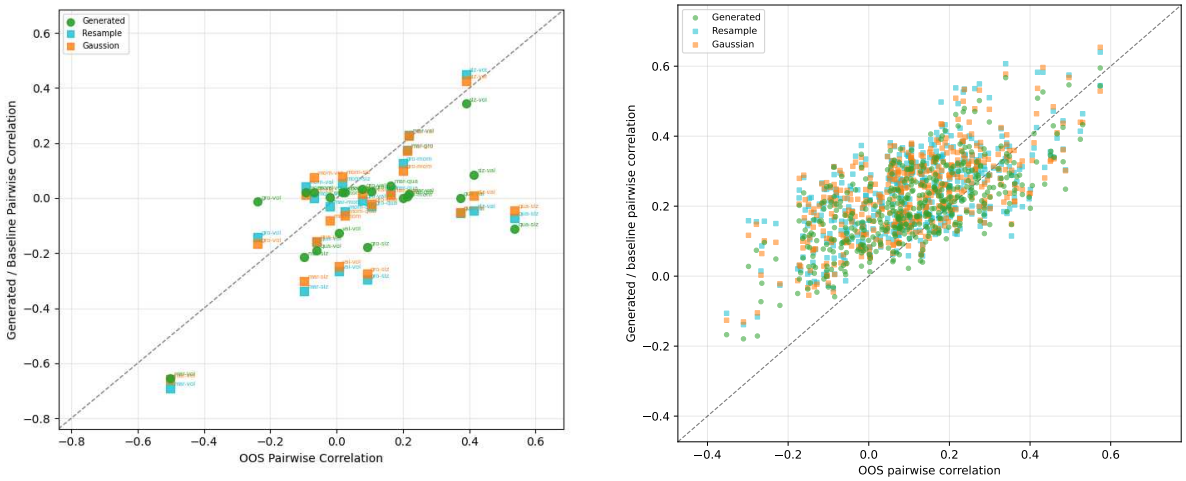
(a) MSCI World, seven factors.



(b) Dow 30, thirty tickers.

Figure 5.3: Correlation matrices for the OOS and training samples and for the FaDiff (*Generated*), Resample and Gaussian samples on MSCI World and Dow 30.

In relative MAE against the OOS correlation matrix, FaDiff is the best of the three, at 0.1674 (generated) versus 0.1738 (Resample) and 0.1687 (Gaussian), though the margin is small. More striking in the two figures is the visible gap to the OOS structure. The scatter points sit mostly below the diagonal, meaning the generated correlations are weaker than the OOS ones. The reason is a distribution shift between the two periods. Several factor correlations, in particular within the quality-size-value block, are markedly stronger in the OOS sample than in the training sample (e.g. quality-size rises from 0.54 in OOS while it is near zero in training). Every method is fitted on the training period and can only reproduce its weaker dependence, and all three sit below the diagonal on the same pairs.



(a) MSCI World, 21 factor pairs.

(b) Dow 30, 435 ticker pairs.

Figure 5.4: Off-diagonal correlations of each method (vertical axis) against the OOS correlations (horizontal axis), one point per pair; points on the dashed diagonal match the OOS correlation exactly. *Generated* denotes FaDiff.

Table 5.3: Correlation-matrix discrepancy against the OOS correlation matrix (↓ lower is better). Best per panel in bold.

Panel	Method	Correlation MAE ↓
MSCI World	FaDiff	**0.1674**
	Resample	0.1738
	Gaussian	0.1687
Dow 30	FaDiff	**0.1221**
	Resample	0.1490
	Gaussian	0.1468
	Train	0.1435

The result on dow 30 dataset separates the methods more clearly, at 0.1221 for FaDiff against 0.1490 for Resample and 0.1468 for the Gaussian (Table 5.3). We also note that FaDiff is closer to the OOS matrix than the training sample itself is, at 0.1221 against 0.1435.

5.1.4. Stock-Level Evaluation

We now push the generated factor draws through the fixed factor loadings and ask whether the stock-level distribution survives that mapping. The comparison runs on the 1924 securities present in both the training and the OOS panel, with 4096 scenarios per method.

Per-stock marginal moments

Figure 5.5 plots, for every stock, each of the first four moments of the generated series against the corresponding OOS value, and Table 5.4 aggregates the results into a mean absolute error across stocks.

Table 5.4: Cross-stock mean absolute error (↓) of the per-stock moments against the OOS panel. Row-best in bold.

Panel	Moment	FaDiff ↓	Resample ↓	Gaussian ↓
MSCI World	Mean	0.00103	0.00103	**0.00100**
	Std	**0.00400**	0.00413	0.00410
	Skew	1.176	**0.974**	1.179
	Kurt	19.27	**9.09**	17.89
Dow 30	Mean	$\mathbf{1.16 \times 10^{-5}}$	1.43×10^{-5}	1.24×10^{-5}
	Std	$\mathbf{1.074 \times 10^{-4}}$	1.075×10^{-4}	1.075×10^{-4}
	Skew	0.287	1.096	**0.173**
	Kurt	**2.51**	22.05	3.24

Mean and standard deviation are recovered by all three methods, and on the standard deviation the generated sample is in fact the most accurate, so per-stock volatility survives the factor-to-stock mapping intact. On the higher moments, however, Resample beats both the generated sample and the Gaussian baseline by a wide margin, since it copies each stock's own historical returns and carries their shape by construction.

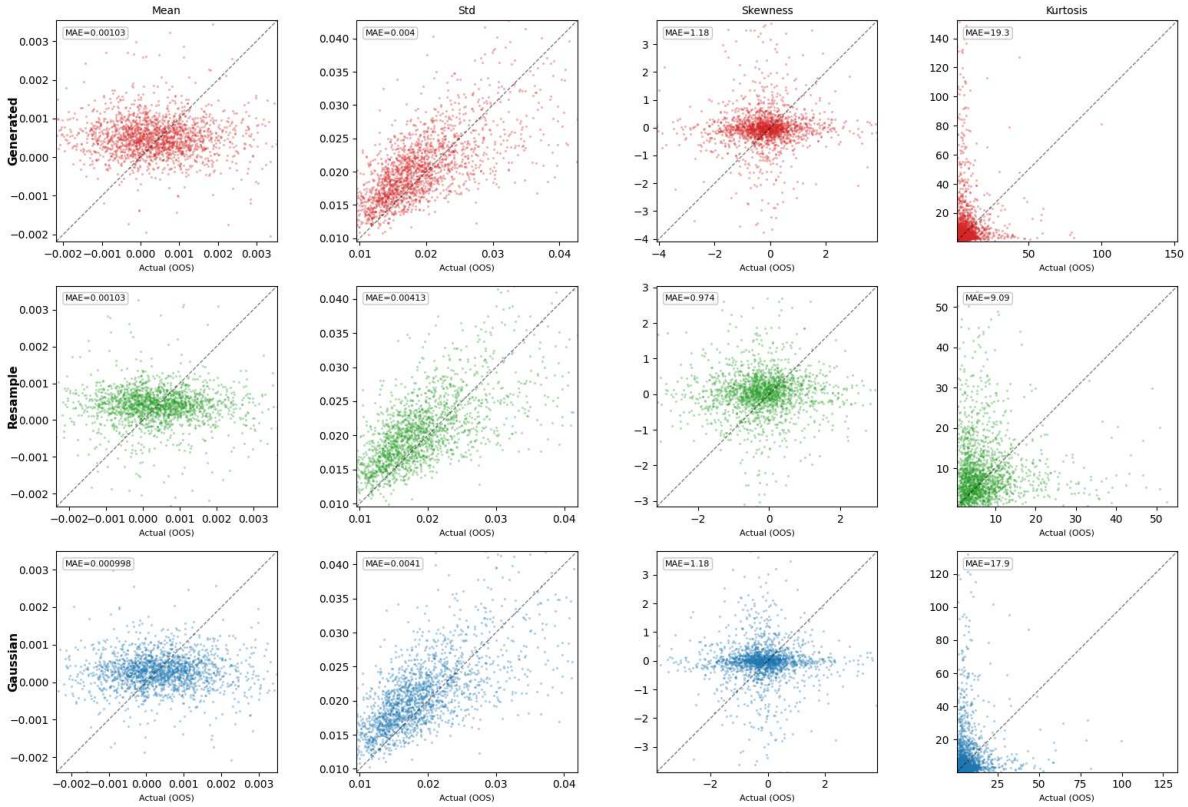
The generated and the Gaussian samples are statistically indistinguishable on both higher moments. At the factor level the same two methods are far apart, at 0.5687 against 0.7006 on skewness and 3.2859 against 5.7692 on excess kurtosis. Since the two differ only in the factor sampler and share the Student-t residual draw, the residual model is what erases an advantage that is decisive in factor space. A well-specified factor generator cannot rescue a mis-specified residual, because the residual contributes almost nothing to the mean of a stock return but dominates its third and fourth moments.

The kurtosis rewards a closer look. Each point is one stock, positioned by its realised OOS kurtosis horizontally and by the kurtosis the method generates for it vertically, so a point above the diagonal means the method gives that stock fatter tails than it turned out to have. So an interesting finding is that all three methods sit above the diagonal on the left of the panel and below it on the right. They overstate the tails of stocks that turned out calm and understate those of the most extreme ones.

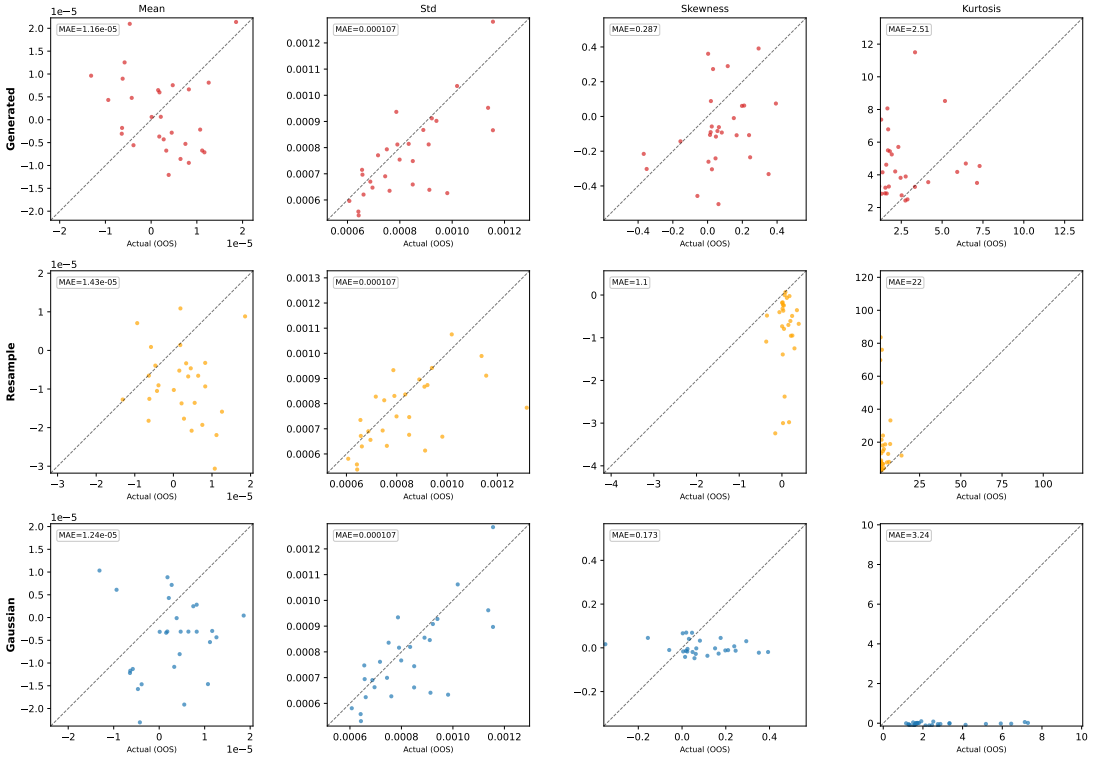
The addition result on dow 30 is informative here because it has no residual model. Its series are generated directly, so the mapping that erases the factor-level advantage on the MSCI panel is simply absent. The excess-kurtosis error is 2.51 for FaDiff against 22.05 for Resample, where on the MSCI panel the same comparison ran the other way. Resample carries each series' training moments by construction, which serves it well when the two windows agree and less well when they do not, and on this panel they do not. The median per-ticker excess kurtosis falls from about 15 in the training window to about 2 in the OOS 50 days. The same distribution shift explains why the Gaussian baseline is competitive on skewness and kurtosis here, at 0.173 and 3.24, since an OOS window that is close to Gaussian is one an elliptical fit describes well. Therefore we read the intraday result as evidence that the stock-level gap of this section is attributable to the inherited residual model rather than to the factor sampler, not as a general claim about higher moments.

Per-stock Kolmogorov-Smirnov test

We continue the distribution comparison with a two-sample Kolmogorov-Smirnov test per stock against the OOS series. Under the null of identical distributions the resulting p-values would be uniform on



(a) MSCI World, one point per stock.



(b) Dow 30, one point per ticker.

Figure 5.5: Per-series moments against the OOS value on both datasets, for FaDiff (*Generated*, red), Resample (green) and Gaussian (blue) samples; points on the dashed diagonal match the OOS moment exactly. The per-panel mean absolute error is printed inside each panel. results are scaled individually and do not share axis limits.

[0,1]; Figure 5.6 shows their histogram across stocks.

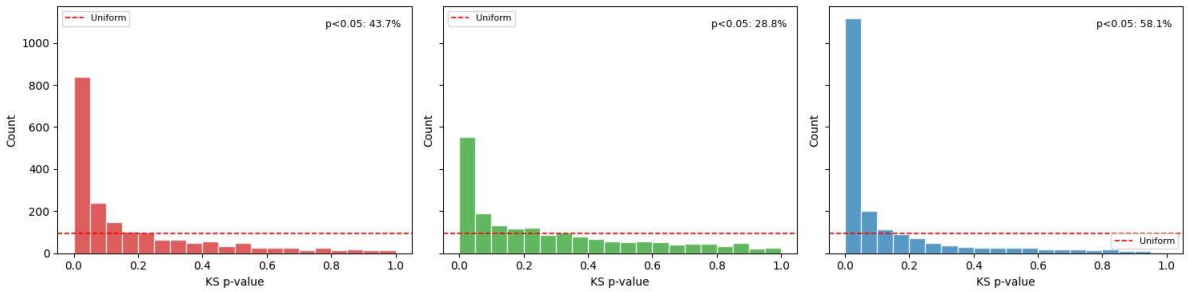


Figure 5.6: Distribution across stocks of the two-sample Kolmogorov-Smirnov p-value against the OOS series, one panel per method, with FaDiff on the left, Resample in the middle and Gaussian on the right. The red dashed line is the uniform reference implied by the null hypothesis. The fraction of stocks with $p < 0.05$ is annotated in each panel.

None of the three histograms is flat, so no method passes as a per-stock generator in absolute terms. Resample is rejected for 28.8% of the stocks, the generated sample for 43.7% and the Gaussian for 58.1%. Resample leads for the same reason it led on the higher moments: it copies each stock's own historical returns, so its per-stock distribution starts from the right shape.

The informative comparison is the other one. FaDiff and the Gaussian baseline share the same fitted Student-t residual draw and differ only in the factor sampler, so the gap between 43.7% and 58.1% cannot come from the residual model. It has to come from the factor sampler. Unlike the moments, which are dominated by the residual-driven tails, the Kolmogorov-Smirnov statistic responds to the whole distribution and therefore also to the systematic component, where the two generators genuinely differ. This is the first indication that the factor generator is still contributing at the stock level, and it motivates measuring the two components separately.

Cross-sectional dispersion

The last measurement asks where in a stock return that contribution sits. A return splits into a systematic part, which depends only on the generated factors and loadings, and an idiosyncratic residual, which FaDiff and the Gaussian baseline draw from the same fitted Student-t. Measuring the two parts separately shows which one carries the stock-level error. The idiosyncratic dispersion of the two pipeline methods must coincide, because it is literally the same draw, so it serves as a control. Any difference between the two methods must come from the systematic part, and the systematic part is also where any surviving factor-level advantage should appear. Figure 5.7 shows the distribution over time of the cross-sectional standard deviation of returns under this split, and Table 5.5 reduces each overlay to the Wasserstein-1 distance between the generated and the OOS dispersion series.

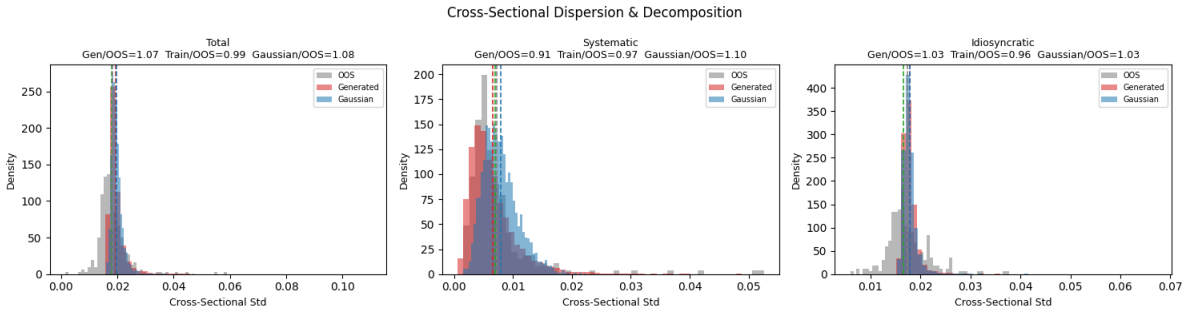


Figure 5.7: Distribution over time of the cross-sectional standard deviation of returns for the OOS (grey), FaDiff (*Generated*, red) and Gaussian (blue) samples, decomposed into total, systematic and idiosyncratic components. Dashed lines mark the means.

On the systematic component the generated sample ties the Resample baseline in Wasserstein distance, at 0.0007 each, and improves on the Gaussian baseline by a factor of three. The distributional shape, not only the average level, carries this result. The OOS systematic dispersion is strongly right-skewed, with a long tail of high-dispersion days, and the generated distribution reproduces that skew, whereas

Table 5.5: Wasserstein-1 distance (↓ lower is better) between each method's cross-sectional dispersion series and the OOS one, by component. Row-best in bold.

Component	FaDiff ↓	Resample ↓	Gaussian ↓
Total	0.0023	**0.0013**	0.0029
Systematic	**0.0007**	**0.0007**	0.0021
Idiosyncratic	0.0022	**0.0015**	0.0022

the Gaussian sits shifted to the right and remains too concentrated to reach the tail. This is the one stock-level statistic on which the factor generator, rather than the residual model, is doing the work, and it is the direct counterpart of the factor-level results above.

The idiosyncratic panel closes the argument from the other side. There the generated and Gaussian distributions coincide almost exactly, as they must, since both draw from the same fitted Student-t. Both also sit at about the right average level, so what separates them from the OOS series is not location but the spread of the dispersion series *over time*: the simulated one is nearly flat from day to day, while the realised one swings between calm and stressed days, so the two histograms in the idiosyncratic panel have almost the same centre and very different widths. The cause is structural. The fitted residual has a fixed per-stock scale and shape, so the idiosyncratic dispersion it produces cannot vary with the market state at all. The residual model is therefore missing not only the right tail shape but any time variation in it, a conclusion reached without using moments or the Kolmogorov-Smirnov test.

Taken together, the stock-level evidence is that the factor-space gains do not survive the mapping to individual equities intact. They are absorbed by the shared residual model, and remain visible only where the measurement isolates the systematic component. The natural remedy is a richer factor model that shrinks the residual's share of a stock's variance and makes the Student-t assumption less load-bearing.

5.2. Conditional Cross-Sectional Generation

Having established that FaDiff reproduces the unconditional cross-section, we now ask whether it can be applied into a prescribed stress regime. We investigate not only whether the targeted factor moves, but whether the joint seven-factor response, the induced shifts in the other factors and in the dependence structure, matches what the regime looks like in reality.

We read every conditional quantity against four references. The unconditional sample is sampled without any condition. Soft rejection is rejection sampling on the conditional loss. It serves as a reference for what the conditional distribution should look like rather than as a method we would deploy, because its cost grows as the acceptance rate falls, so the tighter the condition the slower it becomes, and it is least usable exactly in the extreme regimes a stress test is about. We therefore use it as a sanity check on the guidance approximation and not as a competing generator. The Gaussian baseline applies the same condition to a multivariate Gaussian sample, and the historical stress days, the real days that satisfy the constraint, are the ground-truth regime that all generators are ultimately trying to reproduce. Throughout we generate 4096 conditional samples per method, as in the unconditional evaluation.

5.2.1. Volatility Conditioning

We take as the worked example a volatility crash, defined as the volatility factor falling below the 1% quantile of its unconditional distribution, and trace its effect from the targeted marginal outward to the full joint distribution. The historical analogue of this regime is the days on which the volatility factor fell below the same threshold.

Figure 5.8 shows a sanity check of the guidance method. By construction only 1% of the unconditional mass lies below the threshold; under guidance 72% of the conditional sample clears it. The decisive comparison is against soft rejection, which concentrates 66% of its mass in the same region, and against the resampled historical analogue at 67%. The similar shape between the guidance and soft-rejection KDEs confirms that the guidance approximation is accurate.

Figure 5.9 shows the marginal densities of all seven factors as violin plots. The rejection baseline exhibits a structured cross-factor spillover rather than an isolated shift. Pushing the volatility factor down to a

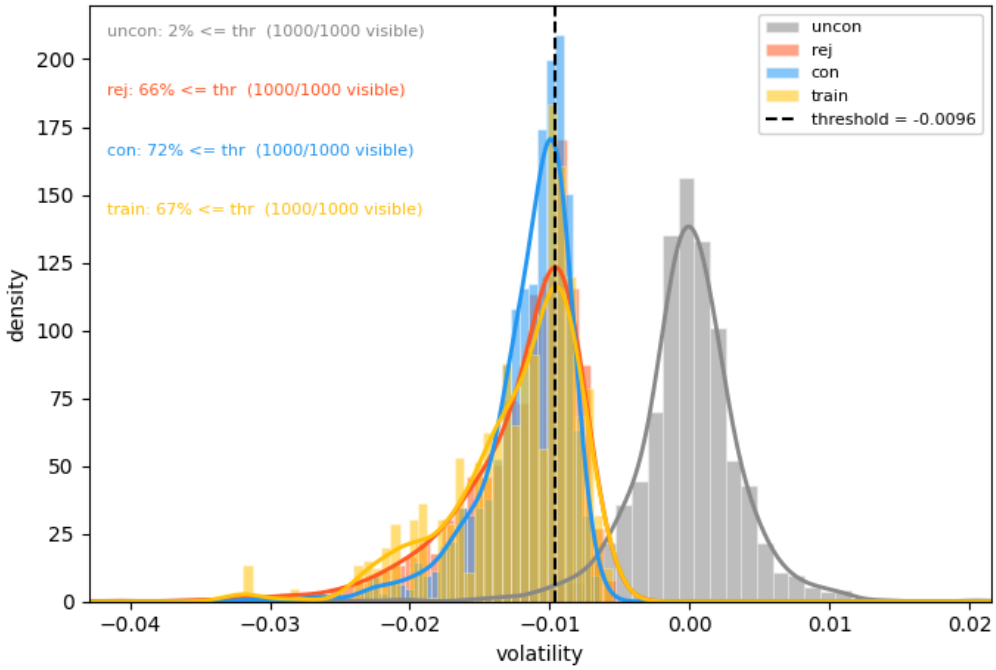


Figure 5.8: Marginal density of the targeted volatility factor under the condition volatility < -0.0096 (dashed line). The unconditional sample (grey) places 1% of its mass in the target tail by construction; guidance (blue) lifts this to 72%, matching soft rejection (66%) and the resampled historical analogue (67%). The legend keys are *uncon* for the unconditional FaDiff sample, *rej* for soft rejection, *con* for the FaDiff guidance sample and *train* for the training-day analogue.

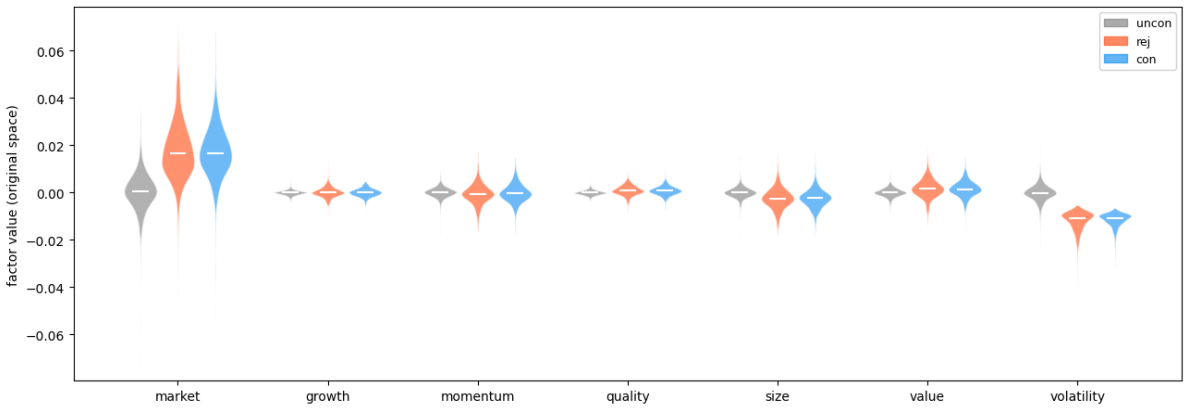
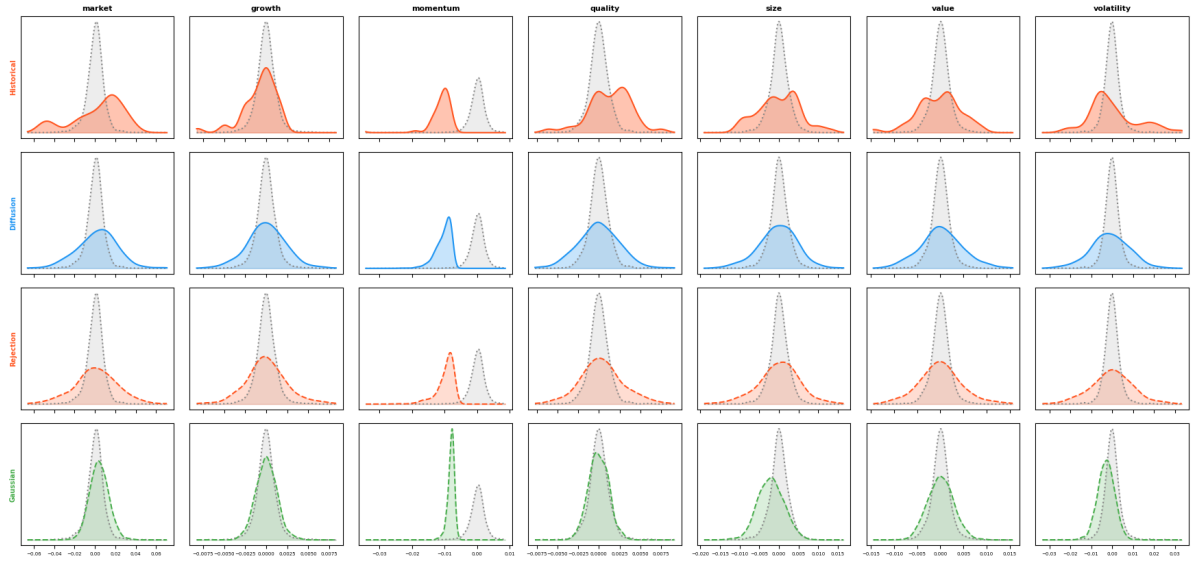


Figure 5.9: Joint response of all seven factors to the volatility condition. Violins show the unconditional (grey, *uncon*), soft-rejection (orange, *rej*) and guidance (blue, *con*) samples, all three drawn from FaDiff; the white bar marks the median. Conditioning the volatility factor alone shifts the market factor upward and tilts size and value, and the guidance violins coincide almost exactly with the soft-rejection violins on every factor.

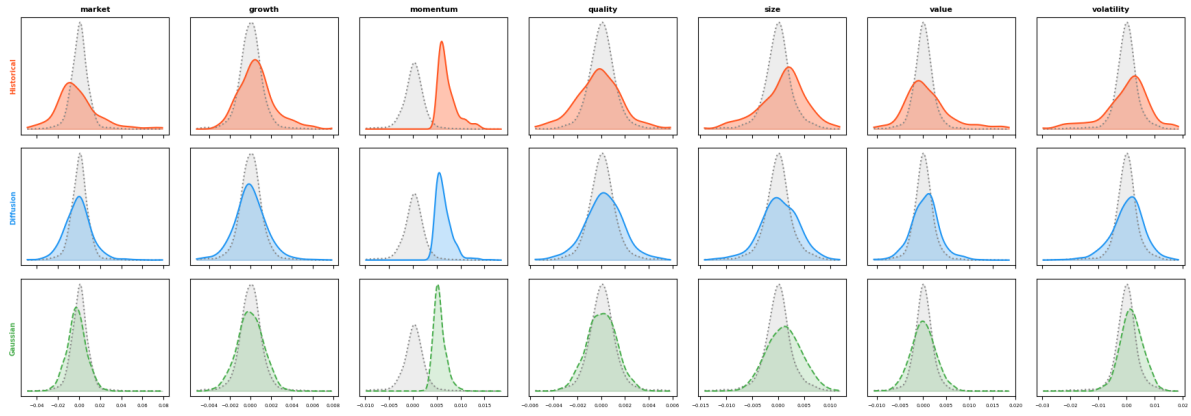
median near -0.011 pulls the market factor up to roughly +0.017 and stretches the size and value factors. Since the guidance violins sit almost exactly on top of the soft-rejection violins, our sample inherits this dependence-driven response.

5.2.2. Momentum Conditioning

The same machinery should work on any factor and in either direction. We test both with momentum, using a downside momentum-crash condition at its 1% quantile and an upside momentum-rally condition at its 99% quantile. Figures 5.10a and 5.10b show the per-factor conditional densities for the two regimes; the remaining diagnostics, which repeat the volatility analysis, are collected in Appendix A.2.



(a) Momentum-crash regime



(b) Momentum-rally regime

Figure 5.10: Per-factor conditional densities for the momentum regimes: the historical stress days, FaDiff under guidance (row *Diffusion*) and the Gaussian baseline (rows), against the unconditional baseline (dotted), one column per factor.

Under the crash condition guidance concentrates 81% of its mass below the threshold, bracketing soft rejection 64% and the historical analogue 59%, and the per-factor densities show the guidance row tracking the historical-stress row, with the momentum marginal sharply displaced into its lower tail while the other factors move only as their dependence with momentum dictates. The rally condition is the mirror image, with 69% of the guided mass now above the threshold, against 55% historically, and the same displacement is reflected into the upper tail. Guidance is thus direction-agnostic, it steers a factor up as readily as down, and the mechanism transfers unchanged from the volatility factor to momentum.

5.2.3. Stock-Level Propagation

The conditions above are imposed at the factor level, but a scenario generator is ultimately used on individual stocks. Reconstructing stock returns from the conditional factors through the factor model lets us check that a factor stress propagates correctly to the cross-section. The mechanism is a loading effect. A stock's conditional expected return should move in proportion to its loading on the stressed factor, so a regression of conditional expected stock return on the momentum loading should have a slope whose sign and magnitude encode how hard the regime hits momentum-exposed stocks.

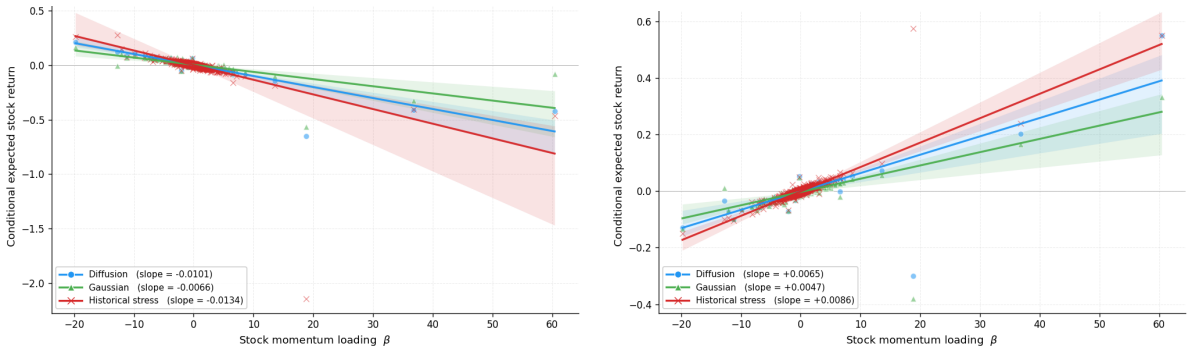


Figure 5.11: Conditional expected stock return against the stock's momentum loading β, under the momentum-crash (left) and momentum-rally (right) regimes, for FaDiff under guidance (*Diffusion*), the Gaussian baseline and the historical stress days. The fitted slope measures how strongly the factor stress propagates to momentum-exposed stocks.

Figure 5.11 confirms the loading effect and its direction. Under the momentum crash the relationship is downward sloping, high-momentum stocks lose the most, with a guidance slope of -0.0101 that sits between the Gaussian baseline (-0.0066) and the historical stress days (-0.0134); under the rally the slope flips positive, +0.0065 for guidance against +0.0047 (Gaussian) and +0.0086 (historical). In both directions guidance recovers the sign of the cross-sectional propagation and lands closer to the historical slope than the Gaussian baseline, which systematically understates how strongly the stress reaches the most exposed stocks. The factor-level conditioning therefore carries through to a coherent, correctly signed response at the stock level.

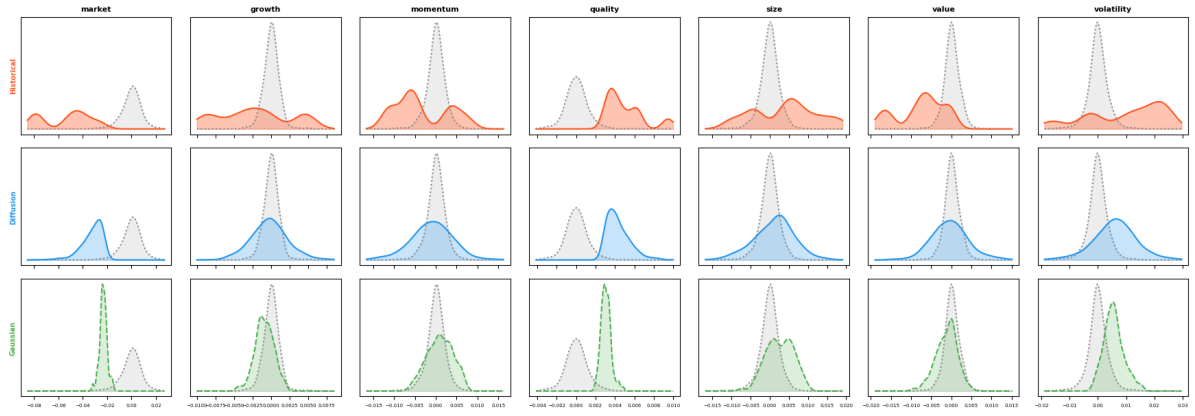
5.2.4. Joint Conditioning

The conditions so far constrain a single factor. A more demanding test is to impose multiple extreme factor constraints at once, since the conditional cross-section is then governed not by one marginal but by the full comovement between the constrained factors and their joint spillover onto the remaining five. We consider two such joint regimes, a quality rally combined with a market drawdown and a volatility spike combined with the same market drawdown. Both are deep in the historical tail, with only nine and twenty-seven historical days respectively satisfying the constraints.

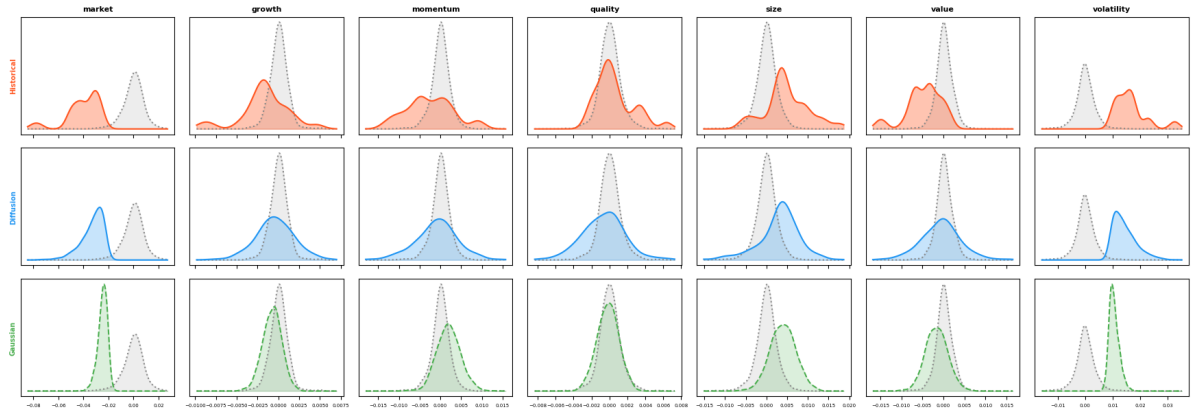
Figure 5.12 shows the per-factor densities for both joint regimes. The historical rows make the difficulty explicit. Under a double constraint the comovement is genuinely complex, the unconstrained factors develop heavy shoulders, asymmetry and multiple modes rather than simply shifting, because so few days satisfy both conditions at once. The Gaussian baseline cannot follow this. Its conditional densities collapse to very thin, over-concentrated spikes, and on several factors they land in visibly wrong places, mode in the wrong tail or no overlap with the historical mass at all, precisely because a single fixed covariance forces an elliptical response that has no freedom to bend with the regime. Guidance, by contrast, reproduces the historical rows closely on both regimes. It keeps the heavier, multi-modal shape of the unconstrained factors and places its mass where the historical days do, rather than degenerating to a spike. Joint conditioning is therefore the regime where the gap between FaDiff and the Gaussian baseline is widest, the guidance model captures a dependence-driven joint response that the Gaussian fundamentally cannot represent.

5.3. Temporal Generation

The evaluations so far concern a single cross-section. The autoregressive extension instead unrolls the first-order Markov chain into a multi-day path. A path can reproduce the correct daily marginal at every



(a) Quality rally and market drawdown (quality $> 0.0032 \wedge$ market < -0.0247, $n = 9$).



(b) Volatility spike and market drawdown (volatility $> 0.0103 \wedge$ market < -0.0247, $n = 27$).

Figure 5.12: Per-factor conditional densities under two simultaneous factor constraints. Rows are the historical stress days, FaDiff under guidance (row *Diffusion*) and the Gaussian baseline; columns are the seven factors; the dotted line is the unconditional baseline.

step and still be wrong along the time axis, by carrying no volatility memory, by compounding into a drifting or exploding cumulative return, or by understating the peak-to-trough losses a risk manager actually faces. We therefore judge the generated paths by whether the temporal stylized facts of return dynamics emerge along the way, none of which is visible to a per-day metric.

We generate 500 FaDiff paths of 22 trading days for all seven factors, and read them against the two baselines on the same test year. The OOS ground truth is the set of 239 overlapping 22-day windows of the OOS factor series. The Resample baseline is a block bootstrap that stitches together real historical days, and the Gaussian baseline draws each day from the fitted normal with its covariance evolving through the RiskMetrics EWMA recursion, so that it carries a time-varying second moment but a Gaussian conditional distribution at every step. Both are specified in full in Section 4.2, including the block-length parameter of the bootstrap and the decay factor of the EWMA recursion.

The substance of the temporal evaluation is whether the path reproduces the stylized facts of return dynamics catalogued by Cont [6]. We report them with the suite of Kubiak et al. [19], defined in Table 4.3 and collected in Table 5.6.

On the return autocorrelation, which should sit near zero, all methods succeed. The generated paths land nearest the OOS level and the per-factor curves of Figure 5.13 fluctuate tightly around zero.

The coarse-fine statistic separates them. The generated paths land almost on the OOS value, 0.0126 against 0.0132, while the Gaussian baseline collapses the cascade to less than half its size at 0.0062 and Resample overshoots it nearly threefold at 0.0360. The Gaussian result follows from its construction: a symmetric variance recursion carries volatility memory but no lead-lag asymmetry, so the cascade has nothing to come from. Only the autoregressive rollout reproduces it at close to its true strength.

Volatility clustering, in contrast, is not measurable at this horizon. Every column of the table is slightly negative and near-identical, the Gaussian baseline included despite its EWMA variance dynamics. This is a small-sample artifact. On windows of only 22 days the mean-subtracted autocorrelation carries a negative bias of about 0.05 in magnitude, inversely proportional to the window length, which swamps the small positive signal that a ten-lag average of daily factor returns would otherwise show. The columns agree because they share this bias, not because any method reproduces clustering, and we therefore read nothing into this row.

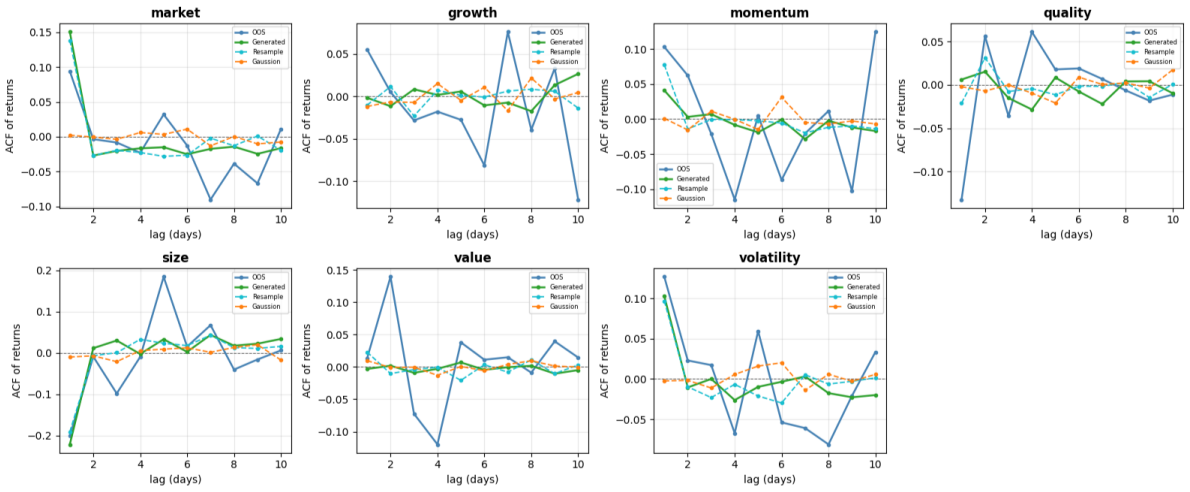


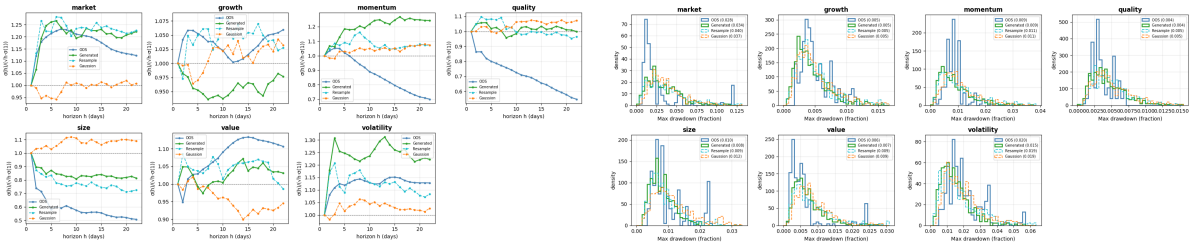
Figure 5.13: Per-factor return autocorrelation against lag days. The FaDiff (*Generated*) and OOS curves both fluctuate around zero across the seven factors, confirming the linear unpredictability of returns; the Resample and Gaussian baselines are shown dashed.

The variance ratio exposes the one limitation of the rollout. The OOS series is mildly mean-reverting, its ratio drifting from 0.97 at $h = 5$ down to 0.88 at the full 22-day horizon, whereas the generated paths sit just above one and essentially flat at roughly 1.07, a near-random-walk term structure with a slight momentum tilt. No method reproduces that mean reversion well, and Resample comes closest at the longest horizon precisely because it reuses real multi-day blocks. This is the expected failure of a

Table 5.6: Temporal stylized-fact statistics for the FaDiff paths and the two baselines. The OOS column is the target, for each generated and baseline entry the value closest to OOS is in bold. ACF, VolClust and Coarse-fine are lag-averaged over $K = 10$; the variance ratio is reported in standardized form $\sigma(h)/(\sqrt{h}\,\sigma(1))$ at horizons $h = 5, 11, 22$.

Statistic	OOS	Generation	Resample	Gaussian
ACF (returns)	-0.0056	**−0.0023**	-0.0015	-0.0000
Volatility clustering	-0.0440	**−0.0440**	-0.0353	-0.0457
Coarse-fine	0.0132	**0.0126**	0.0360	0.0062
Variance ratio $h = 5$	0.9733	1.0757	1.0577	**0.9964**
Variance ratio $h = 11$	0.9411	1.0764	**1.0414**	1.0313
Variance ratio $h = 22$	0.8815	1.0724	**1.0119**	1.0352

first-order Markov transition. Conditioning only on the previous day gives the chain no mechanism to encode reversion that plays out over weeks, so variance accumulates as if increments were independent.



(a) Variance ratio $\sigma(h)/(\sqrt{h}\,\sigma(1))$ against horizon.

(b) Maximum-drawdown distribution.

Figure 5.14: Per-factor variance-ratio term structure (left) and maximum-drawdown distributions (right); *Generated* denotes FaDiff.

Finally we turn to the path-dependent tail measure. Figure 5.14b overlays the per-factor drawdown distributions, and summarising each by its Wasserstein-1 distance to the OOS drawdown distribution, the generated paths give the lowest average error, 0.0036 against 0.0039 for Resample and 0.0037 for the Gaussian, and are closest to OOS on four of the seven factors. Despite the flat variance ratio, then, the model reproduces the distribution of extreme path losses better than either baseline, which is the property that carries directly into the downstream risk application of the next section.

Taken together, the temporal evaluation shows that making the denoiser autoregressive yields paths whose distribution stays intact under compounding and whose nonlinear stylized facts, the coarse-fine cascade and the drawdown profile, match the OOS dynamics more closely than a Resample, which copies real data, or the Gaussian baseline, whose volatility memory is symmetric in time. The single feature it does not capture, the weak multi-week mean reversion of the variance ratio, is the structural limit of a first-order Markov rollout and marks the natural direction for a higher-order temporal extension.

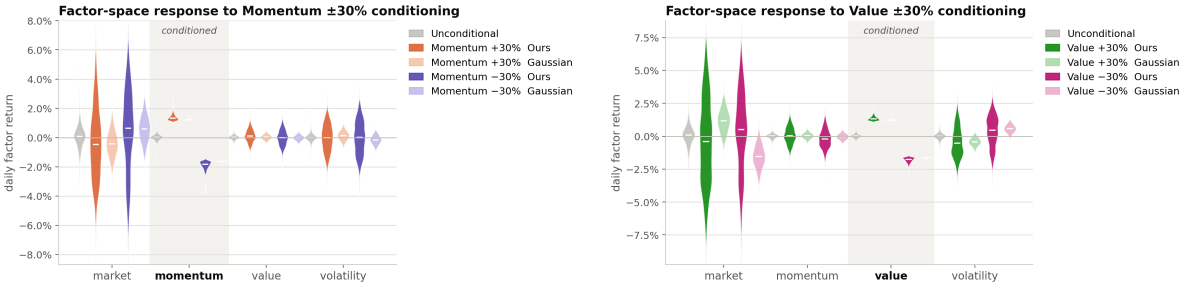
5.4. Downstream Risk Management

The evaluations so far measure how closely the generated distribution resembles the OOS one. We now turn to the practical application of stress testing, which is concerned with regimes that have not occurred. What happens to a portfolio if momentum sells off sharply, or if value rallies? A realised history answers with a single path and a bootstrap can only re-shuffle that path, so neither can evaluate a configuration outside the observed record. This section applies FaDiff's conditional generator to that task and reports what the resulting scenarios show.

We define four scenarios, a 30% move up or down in the monthly return of the momentum factor and of the value factor, and generate 50 scenarios per trading day of 2025 under each. Factor draws are mapped to individual stock returns and then aggregated into portfolio profit and loss. Two portfolio families are considered, a capitalisation-weighted benchmark, which isolates the effect of the scenario itself, and three live strategy portfolios evaluated on their active weights against that benchmark. The Gaussian baseline is conditioned on the same four scenarios and carried through the same pipeline. Throughout, the realised 2025 path serves as the reference for what actually happened, not as a benchmark either generator is trying to match.

5.4.1. Conditional Generation on Factors

Figure 5.15 shows the effect of the conditioning on all factors, not only the targeted one. The conditioned factor reaches the requested level. A 30% monthly shift corresponds to roughly +1.20% per day upward and -1.61% per day downward, and both sets of scenarios centre there, several standard deviations away from the unconditional mode of that factor. Every remaining factor moves off its unconditional position even though none of them was constrained, which is the correlation structure of the factor panel transmitting the shock. What differs between the two generators is the direction in which the shock is transmitted.



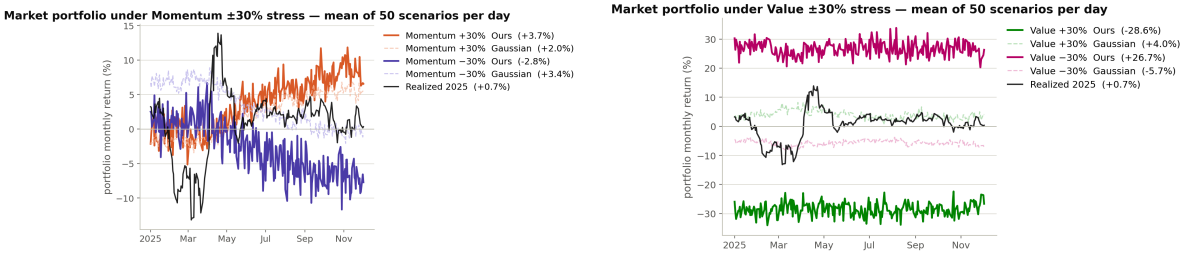
(a) Momentum ±30%. **(b)** Value ±30%.

Figure 5.15: Daily factor returns under each stress scenario for the unconditional sample (grey), FaDiff (labelled *Ours*) and the Gaussian baseline; the conditioned factor is shaded. Both generators reach the target on the conditioned factor and both transmit the shock to the remaining ones; they differ in the direction of the transmission.

The spillover onto the market factor matters most. The benchmark portfolio in the next subsection is capitalisation-weighted, so its profit and loss is driven almost entirely by the overall market return, and whatever a scenario does to the market factor will dominate what it does to the benchmark. Here the two generators disagree. When value rallies by 30%, FaDiff pulls the market factor down by roughly 0.4% per day, whereas the Gaussian baseline pushes it up by about 1.2% per day; when value sells off, both effects flip sign. On the momentum scenarios the two generators agree: both move the market down when momentum rallies and up when it sells off.

5.4.2. Benchmark Portfolio

The benchmark portfolio is the capitalisation-weighted portfolio of the year 2025: on each trading day every stock is held at its share of the total market capitalisation of that day's cross-section. Its monthly profit and loss under a scenario is the weighted sum of the stock monthly returns generated under that scenario, so the weights come from the realised market caps and only the equity returns come from the generator. The realised line is the same weighted sum applied to the stocks' actual returns over the following month. Figure 5.16 tracks this quantity under each scenario across 2025, averaged over the 50 daily draws, and Table 5.7 collects the period means.



(a) Momentum ±30%. **(b)** Value ±30%.

Figure 5.16: Monthly profit and loss of the capitalisation-weighted benchmark under each stress scenario, as the mean of 50 scenarios per day across 2025. Solid lines are FaDiff, dashed lines the Gaussian baseline, and the black line is the realised 2025 path.

The benchmark portfolio returned +0.7% per month on average over 2025. That is the one number history supplies, and it describes the regime that happened to occur; it says nothing about what the

Table 5.7: Mean monthly profit and loss of the benchmark portfolio under each stress scenario, with the realised 2025 value for reference.

Scenario	FaDiff	Gaussian	Realised 2025
Momentum +30%	+3.7%	+2.0%	+0.7%
Momentum -30%	-2.8%	+3.4%	
Value +30%	-28.6%	+4.0%	
Value -30%	+26.7%	-5.7%	

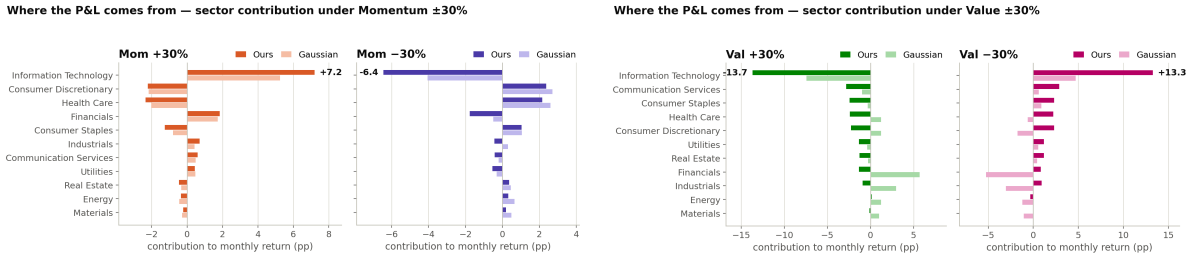
benchmark would have returned had value rallied by 30% in a month. The four scenarios supply that missing number, one per row of the table, and the realised +0.7% sits inside the range they span.

Our scenarios put that cost at -28.6% for a value rally and a gain of +26.7% for a value sell-off, with the momentum scenarios an order of magnitude smaller at +3.7% and -2.8%. The asymmetry between the two directions is expected rather than anomalous, because the shock is defined on a compounded monthly return. A +30% month corresponds to +1.20% per day and a -30% month to -1.61% per day, so the two directions are not symmetric in the quantity the model actually generates.

The Gaussian baseline runs through the same pipeline and returns a different answer on all four scenarios, reversing the sign on both value scenarios and returning a gain in both momentum directions, +2.0% and +3.4%. Both patterns follow from the market-factor spillover described above.

Figure 5.17 decomposes the benchmark's profit and loss by GICS sector. Under our scenarios Information Technology dominates in both cases and with opposite signs, contributing +7.2 pp under a momentum rally and -13.7 pp under a value rally, with Financials on the other side of the momentum trade. This matches prior expectations about these sectors.

The Gaussian attribution agrees on the momentum scenarios and differs on value. Under a value rally our scenarios move every sector the same way. All eleven contribute a loss, Information Technology the largest at -13.7 pp, so the whole benchmark falls together. The Gaussian sectors split instead. Information Technology still falls, by about half as much, but Financials contributes +5.7 pp and Industrials +3.0 pp, and the gains and losses largely cancel. What the Gaussian scenario describes is therefore money rotating between sectors, whereas ours describes the market as a whole moving down.



(a) Momentum ±30%.

(b) Value ±30%.

Figure 5.17: Contribution of each GICS sector to the benchmark's monthly return under each stress scenario, for FaDiff (*Ours*) and the Gaussian baseline.

5.4.3. Strategy Portfolios

The benchmark isolates the scenario. The practical use of a scenario generator, however, is to interrogate a portfolio one actually holds. Figure 5.18 and Table 5.8 report the mean active monthly return of three live strategy portfolios under the same four scenarios.

Judged on realised performance the three portfolios are barely distinguishable. Over 2025 they returned +0.3%, +0.3% and +2.2% per month against the benchmark, and a year of realised performance reveals little about what they are exposed to. Under stress they separate immediately and unambiguously. Strategy 2 moves by about 33 pp in either direction on the value scenarios and only by about 6 pp, with opposite sign, on momentum; Strategy 3 moves by about 39 pp on momentum and 9 pp on value; and Strategy 1 stays within 3 pp of flat on all four. The stress test recovers the design of each portfolio, a

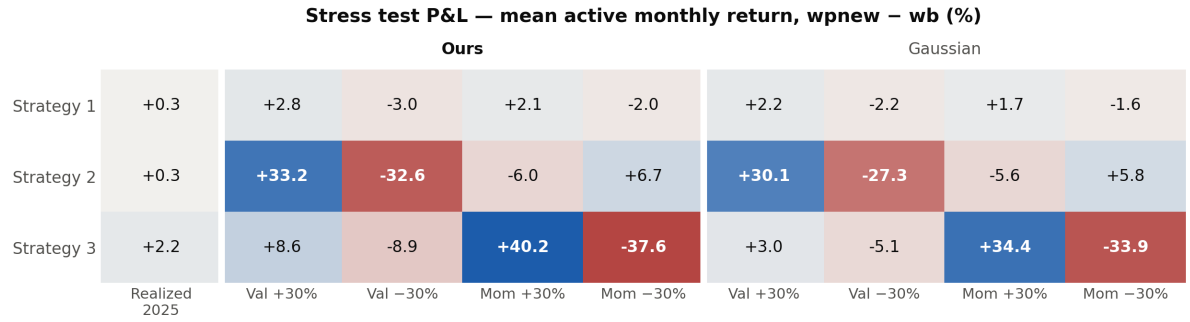


Figure 5.18: Mean active monthly return of three strategy portfolios under each stress scenario, for FaDiff (*Ours*) and the Gaussian baseline, with the realised 2025 active return in the leftmost column.

Table 5.8: Mean active monthly return (%) of three strategy portfolios under each stress scenario. The realised 2025 column is the same for both generators.

		FaDiff				Gaussian			
	Realised 2025	Val +30	Val -30	Mom +30	Mom -30	Val +30	Val -30	Mom +30	Mom -30
Strategy 1	+0.3	+2.8	-3.0	+2.1	-2.0	+2.2	-2.2	+1.7	-1.6
Strategy 2	+0.3	+33.2	-32.6	-6.0	+6.7	+30.1	-27.3	-5.6	+5.8
Strategy 3	+2.2	+8.6	-8.9	+40.2	-37.6	+3.0	-5.1	+34.4	-33.9

value tilt, a momentum tilt and a near factor-neutral construction, from its profit and loss alone. Every exposure is close to antisymmetric between the two directions of a scenario.

The Gaussian baseline recovers the same ranking, so the identification of which portfolio carries which tilt does not require FaDiff. It does, however, separate the portfolios less sharply. On each portfolio's main exposure its response is ten to fifteen per cent smaller, and on the secondary exposure nearly three times smaller; Strategy 3's response to a value rally falls from +8.6 pp to +3.0 pp. Secondary exposures are the ones a manager is least likely to know about in advance, which is exactly where a stress test adds the most.

Overall, this section shows a real-world application of our scenario generator. The generator synthesises a scenario a risk manager would define, applies it to a real benchmark and to live portfolios, attributes the result by sector, and recovers factor exposures that a full year of realised performance leaves invisible. The Gaussian baseline supports the same workflow, and on the strategy portfolios it reaches the same conclusion. The main differences lie in the sign of the benchmark response to a value shock, the sector pattern behind it, and the separation between primary and secondary exposures.

6

Ablation Studies

The evaluation of the previous chapter measures how well FaDiff reproduces the stylised facts of the MSCI World panel. This chapter steps back and asks a more basic question about the approach itself. How well are diffusion models suited to financial factor data in the first place? Equity factor returns are an unusual target for a diffusion model. The panel is short by deep-learning standards, the marginals are heavy-tailed, and the cross-sectional signal is weak relative to the idiosyncratic noise. Any limitation observed in the main results could therefore stem either from the model's representational capacity or from the data regime it is trained in, and the two are easily confused. The ablations below are designed to separate them.

We probe two axes. The first ablation isolates the dependence geometry with a synthetic study in which the ground-truth covariance is prescribed exactly. It asks whether the model's tendency to wash out cross-factor correlation is intrinsic or an artefact of the near-isotropic spectrum of real factor returns. The second isolates the data regime. It varies the length of the training panel to test whether the model's performance on MSCI is limited by sample size and whether training has in fact converged. Together they bound the question from both sides, what the model can represent given clean signal against what it can achieve given the data actually available.

6.1. Diffusion Model on Learning Correlation Structure

In this section we ask which correlation structures a diffusion model can learn well.

We start from the empirical correlation matrix C of the MSCI World factor panel and construct three ground-truth regimes that share the same eigenvector basis but differ in their eigenvalue spectrum:

- **synth**, the historical correlation C used as is;
- **amplified**, which scales every off-diagonal entry of C by a common factor while leaving the diagonal at one. We take the factor as large as positive-definiteness allows, so the pairwise correlations are uniformly strengthened;
- **anisotropic**, in which the eigenvalues of C are raised to the fifth power and renormalised to preserve their mean (eigenvectors unchanged), then the matrix is rescaled back to unit diagonal. This stretches the spectrum, making the leading eigenvalues larger and the trailing ones smaller.

To test how well a diffusion model can learn cross-factor dependence, we turn each of these three correlation matrices into a dataset and fit a separate model to it. For each regime we draw a sample of the same size as the real training panel from a multivariate Gaussian with the prescribed correlation, so that all three datasets share identical standard-normal marginals and differ only in their dependence geometry. Any difference in how well the model performs can therefore be attributed to the correlation structure alone. We then fit a separate diffusion model to each dataset, sample 4096 factor vectors from it, and ask how faithfully the generated samples reproduce the prescribed correlation. We measure this with three statistics that compare the empirical correlation matrix of the generated samples against that of the ground-truth draw. The first two are the Frobenius norm of their difference and the off-diagonal

mean absolute error, both in unnormalised form here since the ground-truth matrix is prescribed rather than estimated; the third is the Pearson correlation between the upper-triangular entries of the two matrices, which captures how well the pattern of pairwise dependence is preserved. We also compare their eigenvalue spectra directly (Figure 6.1), which shows where along the spectrum the model gains or loses fidelity.

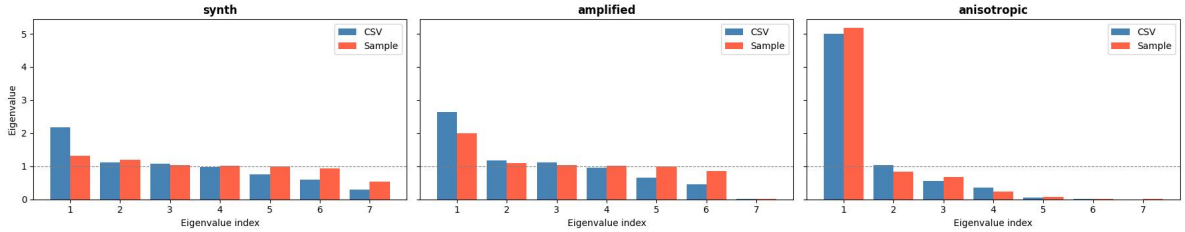


Figure 6.1: Eigenvalue spectra of the correlation matrices for the three regimes, comparing the ground truth (blue) with the FADIFF samples (red). Starting from the empirical factor covariance, we construct three datasets whose correlation structure is progressively more anisotropic, that is with an increasingly uneven eigenvalue spectrum, from synth to anisotropic. The blue bars show these prescribed ground-truth spectra, and the dashed line marks the isotropic value of one. The red bars show that the model reproduces the ground-truth spectrum well in the anisotropic regime but flattens it toward one in the near-isotropic synth regime.

Table 6.1: Recovery of the cross-factor correlation structure by the diffusion model across the three synthetic regimes, ordered by increasing spectral anisotropy. Lower is better for the Frobenius norm and off-diagonal MAE; higher is better for the upper-triangular Pearson r.

Regime	Frobenius ↓	off-diag MAE ↓	upper-tri Pearson r ↑
synth	1.2528	0.1377	0.4934
amplified	1.2006	0.1454	0.8072
anisotropic	**0.5214**	**0.0607**	**0.9938**

Table 6.1 shows that recovery depends on the geometry of the spectrum, but not in the naive way. The synth regime, which uses the real factor correlation, is recovered poorly. The Frobenius distance to the ground truth is 1.25 and the upper-triangular Pearson correlation only 0.49. Simply strengthening every pairwise correlation, the amplified regime, does not fix this. The Frobenius distance (1.20) and the off-diagonal MAE (0.145 against 0.138) are essentially unchanged, and only the pattern of dependence aligns somewhat better (Pearson 0.81). It is only when the data is made genuinely anisotropic that recovery improves sharply, with the Frobenius distance more than halving to 0.52, the off-diagonal MAE dropping to 0.061, and the Pearson correlation reaching 0.99. The heatmaps in Figure 6.2 tell the same story visually. In both the synth and amplified regimes the generated off-diagonal entries are compressed toward zero and stay well below the ground-truth values, whereas in the anisotropic regime the generated matrix reproduces both the sign and the magnitude of every pairwise correlation.

The result shows that our diffusion model recovers the joint dependence structure better, when the correlation spectrum is more anisotropic. The mechanism is consistent with the isotropic mean-squared-error objective of the denoiser. When the spectrum is flat, the weak off-diagonal signal is of the same order as the per-coordinate noise the objective is indifferent to, and the generated correlation matrix collapses toward the identity; when a few large eigenvalues dominate, the dependence becomes a high-variance direction that the objective is forced to fit, and the off-diagonal structure is recovered almost exactly.

6.2. Data convergence on MSCI

This ablation study investigates whether the model's performance can be improved by providing more data. If more history kept improving the generated statistics, the shortfall would be a sample-size limitation.

We retrain the model from scratch on training windows of increasing length and track how the evaluation statistics respond. Concretely we run an expanding-window walk-forward (rolling-origin evaluation)

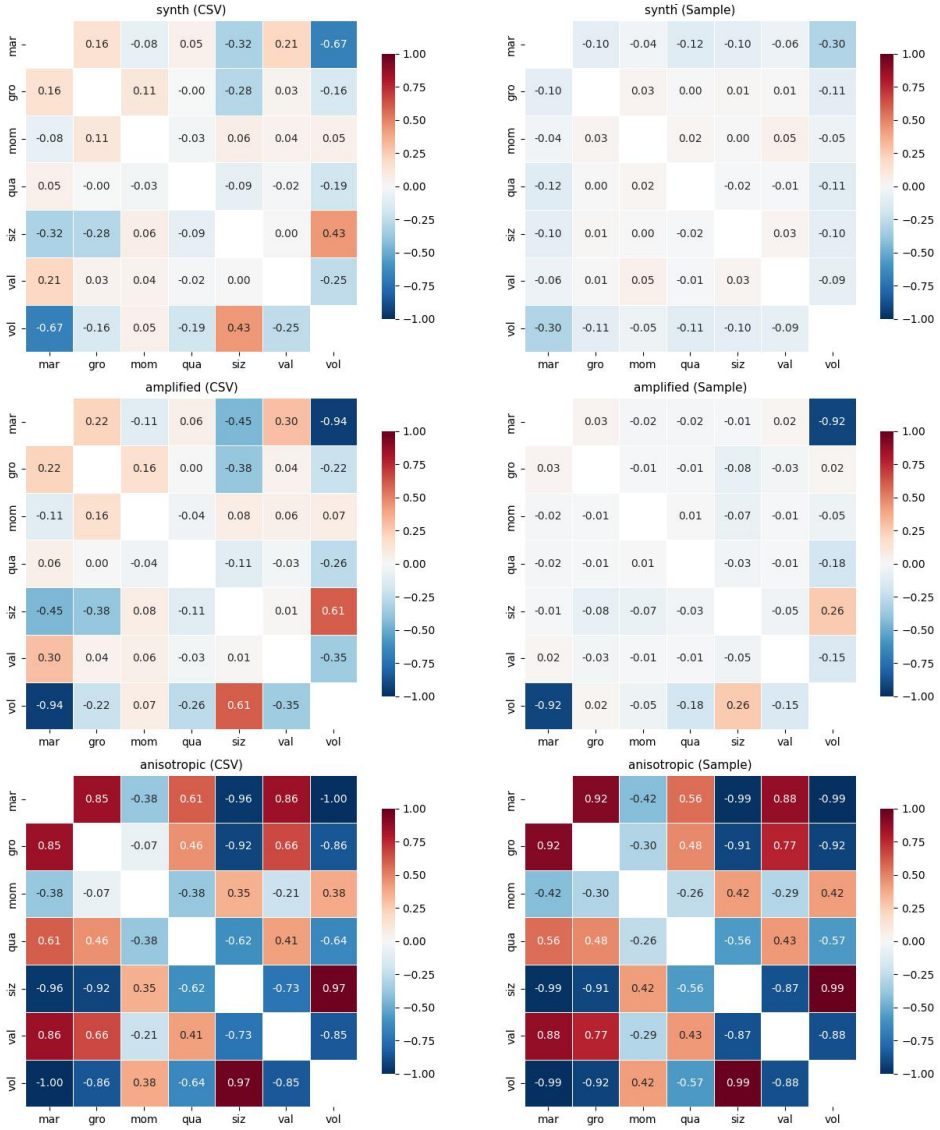


Figure 6.2: Empirical correlation matrices for the three synthetic regimes (rows: synth, amplified, anisotropic). The left column is the ground-truth draw and the right column the FaDiff samples. In the synth regime, which uses the real factor correlation, the generated off-diagonal structure (right) is visibly washed out toward zero. Strengthening every pairwise correlation in the amplified regime does not restore it; the generated entries are still far weaker than the ground-truth values. Only in the anisotropic regime does the generated matrix recover the ground truth, with the two columns nearly indistinguishable.

over the full 25 years. The training window ends in year Y and the test year is the immediately following year $Y+1$. Sweeping Y gives 24 folds whose training length grows from a single year to 24 years each retrained to convergence from random initialisation under an identical configuration. For each fold we compare the 4096 generated vectors against that fold's OOS year on five statistics, the mean absolute error (MAE) across the seven factors of the first four marginal moments (mean, standard deviation, skewness, kurtosis) and the relative Frobenius error between the generated and OOS correlation matrices.

Figure 6.3 and Table 6.2 reveal three distinct behaviours across the five statistics. The mean MAE is already small at the shortest windows and shows no systematic trend with training length, indicating that the first moment has effectively converged regardless of sample size. The standard-deviation MAE is relatively stable across folds, with pronounced spikes confined to crisis test years (the 2008 and 2020 test years), suggesting that more data helps the second moment only modestly and that its largest errors are driven by the difficulty of the test year rather than by a shortage of training data. The remaining three statistics (skewness, kurtosis, and the correlation Frobenius error) behave erratically: they are highly noisy from fold to fold and, if anything, fail to improve or even worsen as more data is added. These higher-order and cross-sectional properties are evidently the hardest for the model to learn, and additional history does not resolve them.

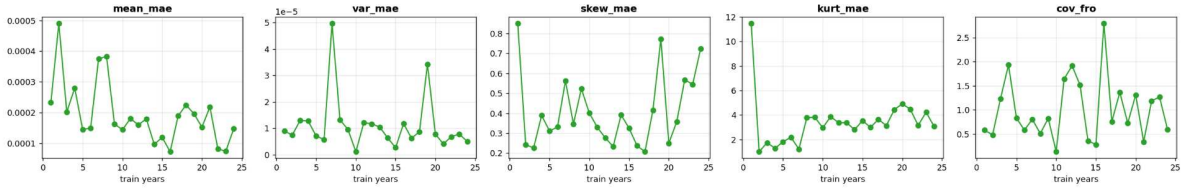


Figure 6.3: Generation quality versus training-set length under the expanding-window walk-forward. Each panel shows one evaluation statistic against the number of training years, lower is better.

Table 6.2: Data-scaling sweep: generation statistics by training-window length, each fold evaluated against its own test year. n is the number of training days. Mean and standard-deviation MAE are reported in scaled units (columns headed $\times 10^{-4}$ and $\times 10^{-5}$); skewness MAE, kurtosis MAE, and the relative correlation Frobenius error are unscaled. Lower is better throughout.

Train yrs	test yr	n	Mean MAE ($\times 10^{-4}$)	Var MAE ($\times 10^{-5}$)	Skew MAE	Kurt MAE	Corr Frob
1	2002	261	2.33	0.90	0.851	11.484	0.581
2	2003	522	4.91	0.74	0.242	1.001	0.483
3	2004	783	2.02	1.30	0.227	1.740	1.235
4	2005	1045	2.80	1.27	0.389	1.282	1.934
5	2006	1305	1.45	0.71	0.311	1.804	0.834
6	2007	1565	1.50	0.57	0.332	2.189	0.582
7	2008	1826	3.76	4.97	0.563	1.207	0.807
8	2009	2088	3.83	1.32	0.347	3.791	0.510
9	2010	2349	1.63	0.95	0.523	3.811	0.826
10	2011	2610	1.45	0.11	0.401	2.967	0.135
11	2012	2870	1.81	1.21	0.330	3.867	1.643
12	2013	3131	1.60	1.17	0.277	3.367	1.914
13	2014	3392	1.80	1.04	0.233	3.372	1.516
14	2015	3653	0.97	0.63	0.392	2.801	0.356
15	2016	3914	1.20	0.26	0.325	3.544	0.284
16	2017	4175	0.73	1.17	0.237	2.991	2.793
17	2018	4435	1.89	0.62	0.207	3.646	0.758
18	2019	4696	2.25	0.87	0.416	3.129	1.364
19	2020	4957	1.96	3.42	0.773	4.442	0.728
20	2021	5219	1.53	0.78	0.250	4.911	1.312
21	2022	5480	2.19	0.40	0.358	4.475	0.340
22	2023	5740	0.82	0.68	0.568	3.154	1.188
23	2024	6000	0.75	0.78	0.545	4.232	1.267
24	2025	6262	1.48	0.50	0.724	3.063	0.596

In summary, on our panel the mean has essentially converged, enlarging the training set improves the variance only to a limited extent, and the behaviour of the remaining three statistics is inconclusive, they are too noisy from fold to fold to attribute any clear gain to a larger training set.

7

Conclusion

This chapter closes the thesis in two steps. Section 7.1 answers the four research questions. Section 7.2 then locates the limitations of the thesis, and the future todos.

7.1. Summary

We propose FaDiff that separates the generation process into two stages. A parametric factor model reduces the high-dimensional cross-section to a small vector of factor returns plus a per-stock idiosyncratic residual, and a Diffusion Transformer generates the factor returns with a forward process driven by anisotropically coloured stable Lévy noise, so that heavy tails and cross-factor co-movement are built into the corruption process rather than being asked of the denoiser alone. Regime conditioning is applied at inference time as signal-to-noise-weighted gradient guidance against a differentiable energy on the clean factor vector, which requires no labelled regimes and no retraining per scenario. A first-order autoregressive rollout extends the generator from a single cross-section to multi-day paths. Every comparison in the thesis holds this pipeline fixed and replaces only the factor sampler, so that differences are attributable to the generative component.

The first research question is answered at the factor level, where the model is the strongest of the three on every unconditional criterion that involves the tail. The Mahalanobis radius of the generated cross-sections curves above the chi-squared reference exactly as the out-of-sample data does, while the Gaussian baseline stays on the reference line and cannot reach the extreme distances at all. The generated marginals attain the lowest Wasserstein distance on five of the seven factors and the best average, with the widest margin on the heavy-shouldered market factor, and they recover both the sign of the skewness and the magnitude of the excess kurtosis, which an gaussion model cannot represent by construction. Cross-factor dependence is reproduced marginally better than by either baseline, although all three sit below the out-of-sample correlations because of the distribution shift between the training and test periods. The same unconditional comparison is repeated on the self-built intraday panel, which varies both the sampling frequency and the composition of the cross-section. The ordering carries over, and the margin on cross-factor dependence is wider there, in the direction the anisotropy ablation would predict.

Pushing the factor draws through to individual securities gives the thesis its main negative result. Per-stock volatility survives the mapping and is in fact reproduced most accurately by FaDiff, but the third and fourth moments of individual stock returns are matched substantially better by direct historical resampling, and the same ordering appears in a per-stock Kolmogorov-Smirnov test. The evidence, from the moment comparison and from the decomposition of cross-sectional dispersion alike, points at the shared Student-t residual draw rather than at the factor sampler.

The second research question is answered by the conditional results about the guidance mechanism. On a single-factor stress the guidance sample agrees closely with soft rejection sampling, which confirms that the guidance approximation targets the intended tilted distribution, and it reproduces the structured cross-factor spillover of the historical analogue rather than merely shifting the targeted marginal. Under

two simultaneous constraints, where only a handful of historical days qualify, the unconstrained factors develop heavy shoulders, asymmetry and multiple modes; guidance follows that shape while the Gaussian conditional collapses into over-concentrated spikes that on several factors sit in the wrong tail entirely. All of conditional samples at inference time so only 1 training is needed.

On the third research question, the temporal extension shows that the generated paths keep their distribution intact under compounding and reproduce the coarse-fine volatility cascade and the maximum-drawdown distribution more closely than either baseline, the latter being the property that carries directly into risk applications. The one stylized fact the first-order rollout misses is the weak multi-week mean reversion of the variance ratio, a horizon that lies beyond what a Markov-1 chain can represent.

The fourth question is answered by the stress-testing application, where the full pipeline runs on a capitalisation-weighted benchmark and three live strategy portfolios. Conditioning on factor moves that have no historical precedent produces coherent, sector-attributable profit and loss, and recovers the factor exposures of the live portfolios that a full year of realised performance does not reveal.

Two ablations give some side findings. Prescribing the ground-truth covariance in a synthetic study shows that the model's tendency to wash out cross-factor correlation is not intrinsic but a property of the data regime. When the correlation spectrum is made genuinely anisotropic the Frobenius error more than halves and the Pearson agreement reaches 0.99, whereas amplifying every correlation while keeping the spectrum flat changes nothing. Varying the length of the training panel shows that the first moment has converged on this dataset and that further data improves the variance only marginally, so the residual gap is not primarily a sample-size problem.

7.2. Discussion

Limitations. The main constraint at the stock level is the inherited residual model. Most of a stock's return variance is idiosyncratic, so a Student-t with a fixed per-stock scale carries most of the distributional weight, and that scale cannot respond to the market state at all. This is a missing mechanism rather than a mis-estimated parameter, and it is what the stock-level results reflect. Our resitual modeling also ignores the lower bound at -1 that limited liability imposes on a simple return, so a small fraction of generated returns fall below it.

The denoising objective is a second structural limitation, and one likely to generalise beyond this application. It is an isotropic mean-squared error, so it weights every coordinate equally. When the correlation spectrum of the target is close to flat, as it is for near-orthogonal style factors, the off-diagonal signal is of the same order as the per-coordinate noise the objective is indifferent to, and the generated dependence collapses toward the identity. Colouring the forward noise mitigates this but does not remove it.

As for the downstream tasks. The stress scenarios are counterfactual by design and have no realised outcome to score against. The disagreement between the two generators on the value scenarios is a concrete case, in which ours moves the market down when value rallies and the Gaussian baseline moves it up, the two answers imply opposite actions for the same portfolio, and no metric in this thesis settles which is right. Someone has to judge which transmission is economically plausible. Finally, the MSCI results cannot be verified without a commercial subscription, which is the reason the Dow 30 panel was built and released.

Future work. The limitations above point at specific and largely independent next steps. The most direct is a residual model with a stochastic or GARCH-type scale, which addresses the missing time variation where it actually lives; giving that residual bounded support would close the -1 violation at the same time. A richer factor set would shrink the residual's share of a stock's variance and make the parametric assumption less load-bearing. On the objective, a loss that weights directions by the spectrum of the target covariance, or a whitened parameterisation of the factor space, is the natural response to the isotropy problem. A higher-order or attention-based rollout would address the multi-week structure the Markov-1 chain cannot represent. Beyond these, the conditioning mechanism currently acts only on endogenous factor coordinates; extending it to the exogenous indicators of Section 1.2, such as the VIX or macroeconomic releases, would widen the range of scenarios a risk manager can pose without

retraining.

References

[1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. "Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models". In: *Transactions on Machine Learning Research*. 2022.

[2] Giovanni Barone-Adesi, Kostas Giannopoulos, and Les Vosper. "VaR without correlations for portfolios of derivative securities". In: *Journal of Futures Markets: Futures, Options, and Other Derivative Assets* 19.5 (1999), pp. 583–602.

[3] Minshuo Chen et al. "Diffusion Factor Models: Generating High-Dimensional Returns with Factor Structure". In: *arXiv preprint arXiv:2504.06566* (2025).

[4] Jooyoung Choi et al. "ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 14347–14356.

[5] Hyungjin Chung et al. "Diffusion Posterior Sampling for General Noisy Inverse Problems". In: *The Eleventh International Conference on Learning Representations (ICLR)*. 2023.

[6] Rama Cont. "Empirical properties of asset returns: stylized facts and statistical issues". In: *Quantitative Finance* 1.2 (2001), pp. 223–236.

[7] Prafulla Dhariwal and Alexander Nichol. "Diffusion Models Beat GANs on Image Synthesis". In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 8780–8794.

[8] Yitong Duan et al. "FactorVAE: A Probabilistic Dynamic Factor Model Based on Variational Autoencoder for Predicting Cross-Sectional Stock Returns". In: *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence*. 2022, pp. 4468–4476.

[9] Paul Embrechts, Alexander McNeil, and Daniel Straumann. "Correlation and dependence in risk management: properties and pitfalls". In: *Risk Management: Value at Risk and Beyond*. Cambridge University Press, 2002, pp. 176–223.

[10] Eugene F Fama and Kenneth R French. "Common risk factors in the returns on stocks and bonds". In: *Journal of Financial Economics* 33.1 (1993), pp. 3–56.

[11] Eugene F Fama and James D MacBeth. "Risk, return, and equilibrium: Empirical tests". In: *Journal of Political Economy* 81.3 (1973), pp. 607–636.

[12] Xuefeng Gao, Mengying He, and Xuedong He. "Factor-Based Conditional Diffusion Model for Portfolio Optimization". In: *arXiv preprint arXiv:2509.22088* (2025).

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 6840–6851.

[14] Jonathan Ho and Tim Salimans. "Classifier-Free Diffusion Guidance". In: *arXiv preprint arXiv:2207.12598* (2022).

[15] Xingchang Huang et al. "Blue Noise for Diffusion Models". In: *ACM SIGGRAPH 2024 Conference Papers*. 2024.

[16] J.P. Morgan and Reuters. *RiskMetrics – Technical Document*. Tech. rep. New York: J.P. Morgan/Reuters, 1996.

[17] Tero Karras et al. "Elucidating the Design Space of Diffusion-Based Generative Models". In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022, pp. 26549–26561.

[18] Zhifeng Kong et al. "DiffWave: A Versatile Diffusion Model for Audio Synthesis". In: *International Conference on Learning Representations*. 2021.

[19] Szymon Kubiak et al. "MacroVAE: Counterfactual Financial Scenario Generation via Macroeconomic Conditioning". In: *Proceedings of the 6th ACM International Conference on AI in Finance*. 2025, pp. 797–805. DOI: 10.1145/3768292.3770360.

[20] Andrew W Lo and A Craig MacKinlay. “Stock market prices do not follow random walks: Evidence from a simple specification test”. In: *The Review of Financial Studies* 1.1 (1988), pp. 41–66.

[21] Andreas Lugmayr et al. “RePaint: Inpainting Using Denoising Diffusion Probabilistic Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11496–11505.

[22] Gábor Lugosi and Shahar Mendelson. “Mean estimation and regression under heavy-tailed distributions: A survey”. In: *Foundations of Computational Mathematics* 19.5 (2019), pp. 1145–1190.

[23] Chenlin Meng et al. “SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations”. In: *International Conference on Learning Representations (ICLR)*. 2022.

[24] Alexander Quinn Nichol and Prafulla Dhariwal. “Improved Denoising Diffusion Probabilistic Models”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8162–8171.

[25] William Peebles and Saining Xie. “Scalable Diffusion Models with Transformers”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 4195–4205.

[26] Dimitris N Politis and Joseph P Romano. “The stationary bootstrap”. In: *Journal of the American Statistical Association* 89.428 (1994), pp. 1303–1313.

[27] Kashif Rasul et al. “Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting”. In: *Proceedings of the 38th International Conference on Machine Learning*. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8857–8868.

[28] Dario Shariatian, Umut Simsekli, and Alain Durmus. “Denoising Lévy Probabilistic Models”. In: *Advances in Neural Information Processing Systems*. 2024.

[29] William F Sharpe. “Capital asset prices: A theory of market equilibrium under conditions of risk”. In: *The Journal of Finance* 19.3 (1964), pp. 425–442.

[30] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”. In: *Proceedings of the 32nd International Conference on Machine Learning*. Vol. 37. Proceedings of Machine Learning Research. PMLR, 2015, pp. 2256–2265.

[31] Jiaming Song et al. “Loss-Guided Diffusion Models for Plug-and-Play Controllable Generation”. In: *Proceedings of the 40th International Conference on Machine Learning*. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 32483–32498.

[32] Yang Song and Stefano Ermon. “Generative Modeling by Estimating Gradients of the Data Distribution”. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019, pp. 11918–11930.

[33] Yang Song and Stefano Ermon. “Improved Techniques for Training Score-Based Generative Models”. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 12438–12448.

[34] Yang Song et al. “Score-Based Generative Modeling through Stochastic Differential Equations”. In: *International Conference on Learning Representations*. 2021.

[35] Yusuke Tashiro et al. “CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation”. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021, pp. 24804–24816.

[36] Ling Yang et al. “Diffusion Models: A Comprehensive Survey of Methods and Applications”. In: *ACM Computing Surveys* (2023).

A

Additional Generation Results

A.1. Unconditional Generation

Two quantile diagnostics support the unconditional evaluation of the main text, and both are reported on each panel. On the MSCI World data the cross-sectional units are the seven estimated factors; on the Dow 30 data they are the thirty tickers themselves, so the same diagnostic is computed in thirty dimensions rather than seven and the degrees of freedom of the reference law change accordingly.

Joint tail against the Gaussian reference

Figures A.1 and A.1b collapse each cross-section into the squared Mahalanobis distance and plot it against a theoretical chi-squared law, under which a Gaussian sample would lie on a straight line. On the MSCI panel the OOS distances peel away from roughly the tenth quantile onward, reaching squared distances near 60 where the reference predicts under 20, so the joint tail is materially heavier than an elliptical Gaussian permits. This is the premise the higher-moment comparison of the main text rests on. The generated sample inherits the same curvature alongside Resample, while the Gaussian baseline stays on the reference line and tops out near 25.

The intraday panel repeats the pattern at its own scale. The OOS distances reach roughly 283 at the maximum, where the chi-squared reference places the far tail an order of magnitude lower, and FaDiff follows them across the upper quantiles. The Gaussian baseline again stops well short, near 69. This is a property of the model class rather than of the fit. An elliptical Gaussian has no mechanism for placing mass that far from the centre, on either panel.

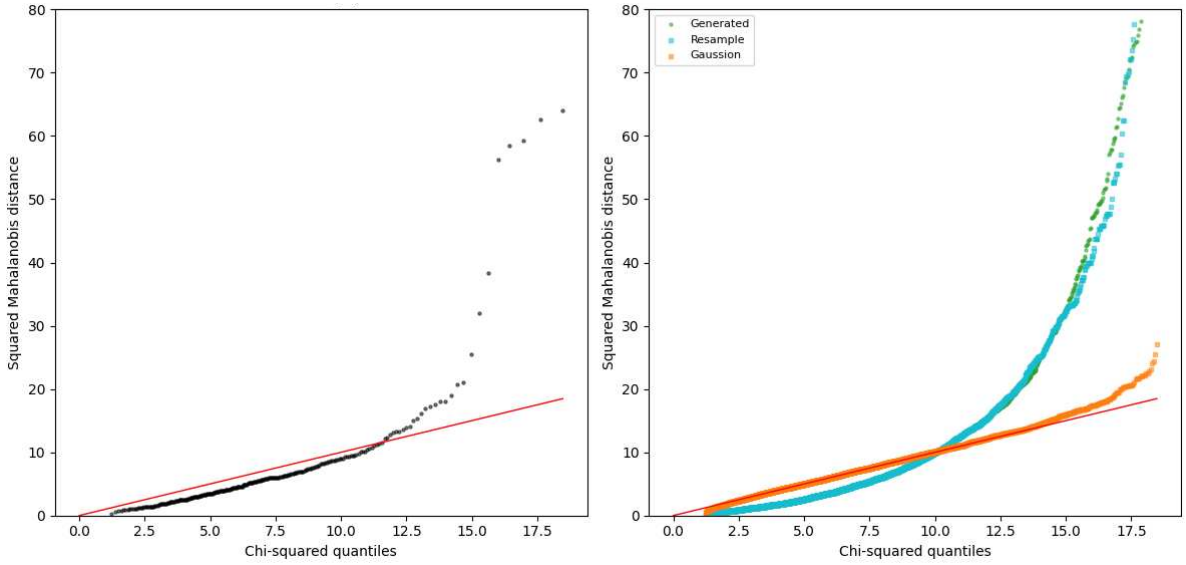
Marginals against the out-of-sample sample

Figures A.2 and A.2b keep the cross-sectional units apart and replace the theoretical reference by the realised OOS sample. On the MSCI panel the generated quantiles stay on the diagonal into the extreme tails, where a single Wasserstein summary is least informative and where the Gaussian baseline bends away from the line; the ordering agrees with the main-text results. On the intraday panel the three methods are much harder to separate, which the Wasserstein averages also show, and the visible deviations are confined to the outermost quantiles of a few tickers.

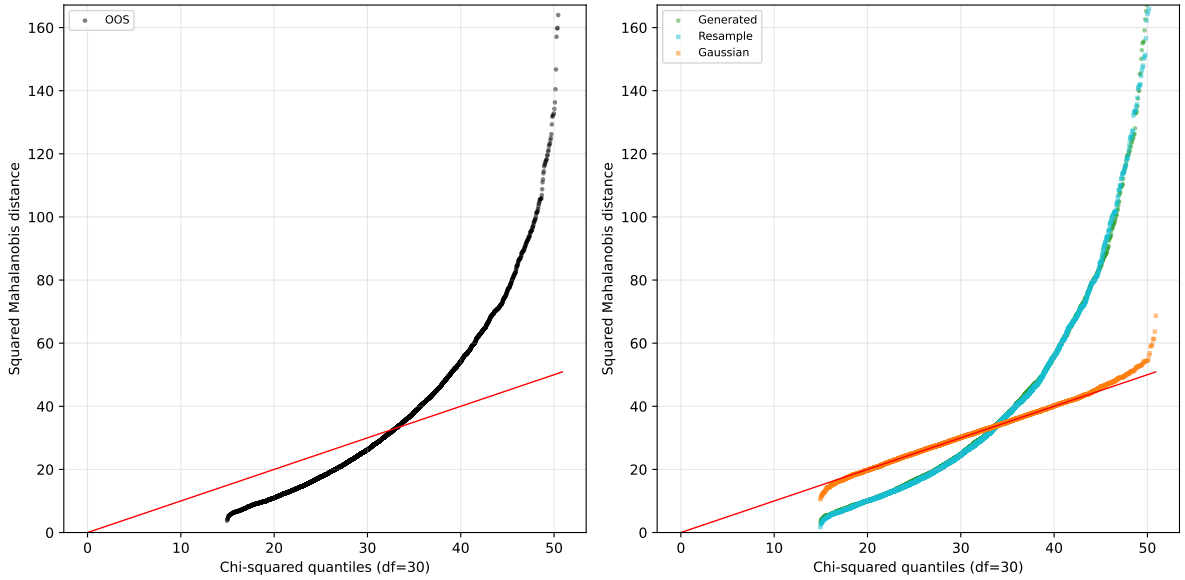
Table A.1 gives the per-asset Wasserstein distances behind the Dow 30 average reported in the main text.

A.2. Conditional Generation

This section collects the full diagnostic suite for the conditional experiments of the main text, covering the volatility stress (volatility < -0.0096) and the two momentum regimes (crash, momentum < -0.0083; rally, momentum > 0.0054). For each regime we report the per-factor conditional densities, factor correlation matrices, joint factor responses, and global-geometry (PCA / scatter) diagnostics, of which only a subset appears in the main text. The momentum crash is the one regime where the Gaussian baseline edges out guidance on dependence. The whole-matrix relative MAE between the two correlation

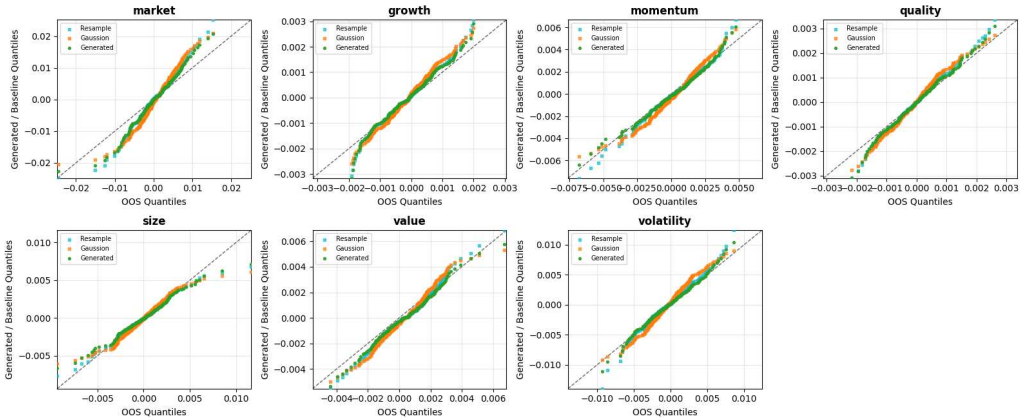


(a) MSCI World, seven factors, against χ^2_7.

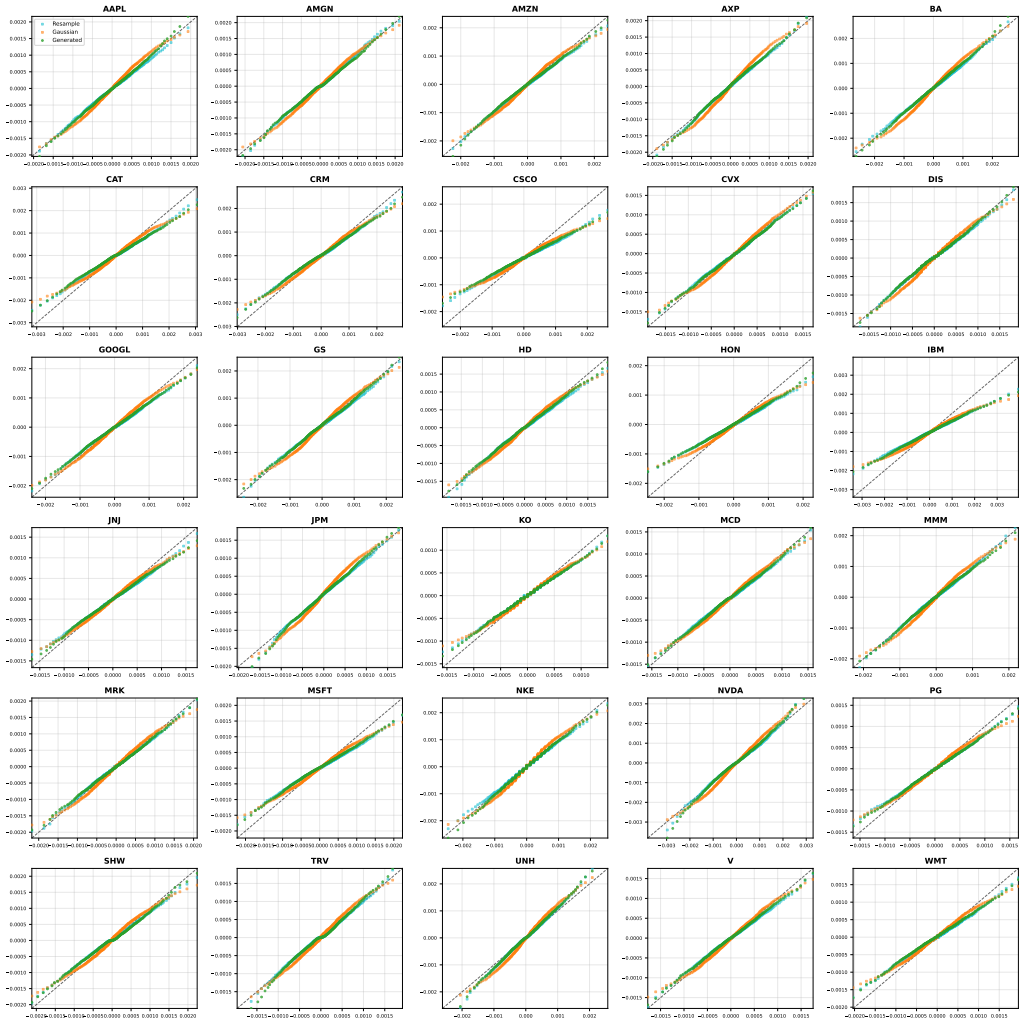


(b) Dow 30, thirty tickers, against χ^2_{30}.

Figure A.1: Mahalanobis Q-Q plots of the squared distance d^2 against the χ^2 reference quantiles on both panels; the red line is the Gaussian reference. Within each row, *left* is the OOS sample and *right* overlays FADIFF (labelled *Generated*, green), Resample (cyan) and the Gaussian baseline (orange).



(a) MSCI World, one panel per factor.



(b) Dow 30, one panel per ticker, over the 1 to 99% quantile range.

Figure A.2: Quantile-quantile plots against the OOS sample on both panels. Each subplot plots 200 evenly spaced quantiles of FaDiff (*Generated*, green), Resample and Gaussian samples against the corresponding OOS quantiles; points on the dashed diagonal indicate an exact quantile match.

Table A.1: Wasserstein-1 distance (W_1, ↓ lower is better) between each method's marginal and the OOS marginal, per ticker on the Dow 30 intraday panel. All values in units of 10^{-5}; row-best in bold.

Ticker	FaDiff	Resample	Gaussian	Ticker	FaDiff	Resample	Gaussian
AAPL	**3.06**	4.75	8.75	JNJ	7.48	8.04	**4.78**
AMGN	4.03	**3.19**	7.75	JPM	3.60	**3.10**	12.03
AMZN	**5.92**	6.52	7.94	KO	8.30	8.31	**6.47**
AXP	4.11	**3.01**	13.72	MCD	**3.50**	5.00	3.66
BA	**3.63**	4.01	14.57	MMM	**3.04**	3.61	10.64
CAT	23.11	20.95	**13.43**	MRK	5.52	**5.35**	6.68
CRM	16.69	16.26	**9.32**	MSFT	16.93	18.85	**11.31**
CSCO	27.15	26.57	**21.09**	NKE	**4.51**	4.97	11.89
CVX	**4.53**	5.34	5.88	NVDA	**7.50**	7.66	15.85
DIS	**3.82**	4.50	6.74	PG	8.87	8.86	**6.16**
GOOGL	10.19	10.96	**5.96**	SHW	9.61	9.72	**8.18**
GS	**4.45**	6.95	8.85	TRV	4.85	**3.69**	7.22
HD	**4.31**	6.43	5.89	UNH	7.73	**7.35**	16.43
HON	19.89	21.63	**14.88**	V	**4.70**	6.26	4.73
IBM	36.89	38.47	**30.37**	WMT	12.51	12.29	**8.00**
Average	FaDiff **9.35**	Resample 9.75	Gaussian 10.30				
Row-best count	FaDiff 12/30	Resample 6/30	Gaussian 12/30				

matrices, against the historical stress days, is 0.1909 for guidance, 0.1704 for the Gaussian baseline and 0.1510 for the unconditional cross-section, reflecting the near-Gaussian shape of the momentum factor. Figures below are grouped by regime (volatility stress, then crash, then rally); captions identify each panel.

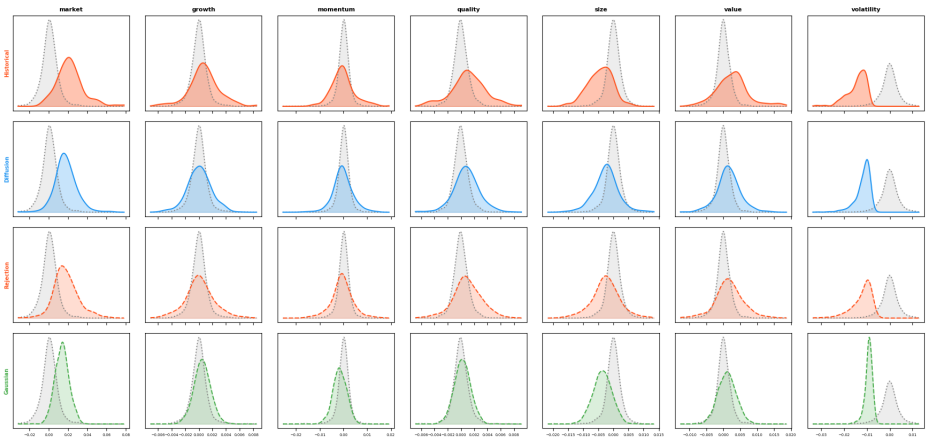


Figure A.3: Volatility stress: per-factor conditional densities for the historical stress days, guidance, soft rejection and the Gaussian baseline (rows), against the unconditional baseline (dotted), one column per factor.

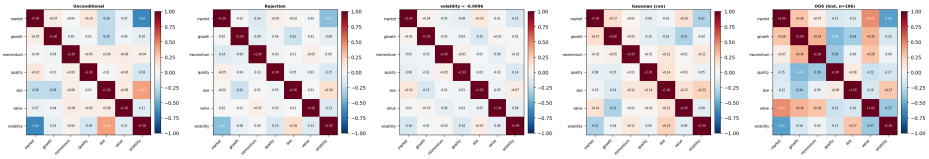


Figure A.4: Volatility stress: factor correlation matrices for the unconditional, soft-rejection, guidance and Gaussian samples and for the historical stress days.

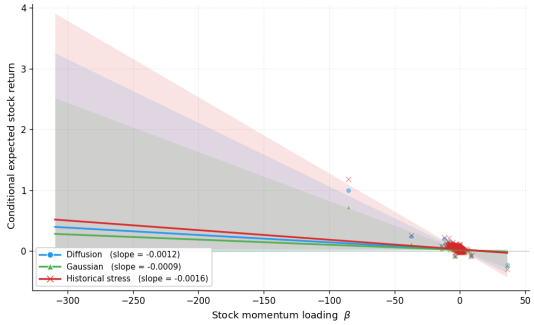


Figure A.5: Volatility stress: conditional expected stock return against the stock's momentum loading β, for guidance, the Gaussian baseline and the historical stress days.

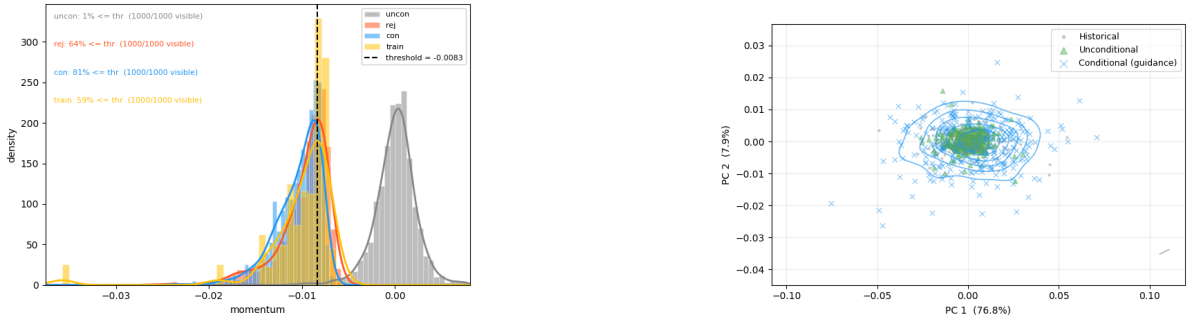


Figure A.6: Momentum crash: targeted-factor marginal under the condition (left) and PCA projection of the conditional and unconditional scenarios (right).

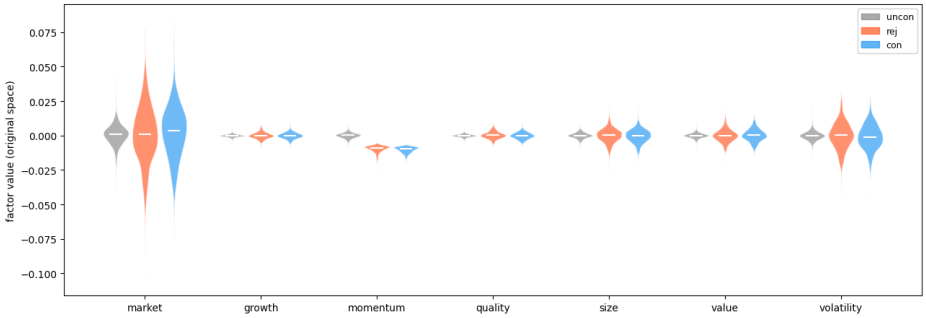


Figure A.7: Momentum crash: joint response of all seven factors for the unconditional, soft-rejection and guidance samples.

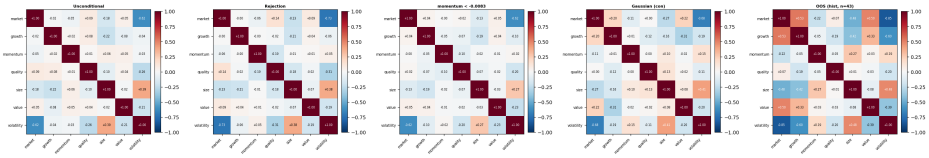


Figure A.8: Momentum crash: factor correlation matrices for the unconditional, soft-rejection, guidance and Gaussian samples and for the historical stress days.

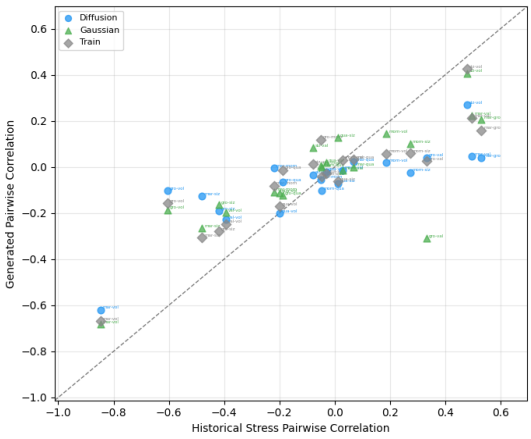


Figure A.9: Momentum crash: pairwise factor correlations of each method against the historical-stress correlations.

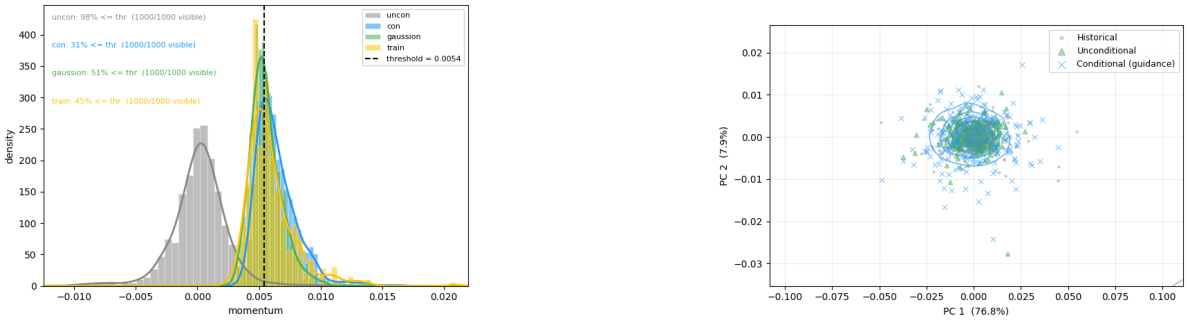


Figure A.10: Momentum rally: targeted-factor marginal under the condition (left) and PCA projection of the conditional and unconditional scenarios (right).

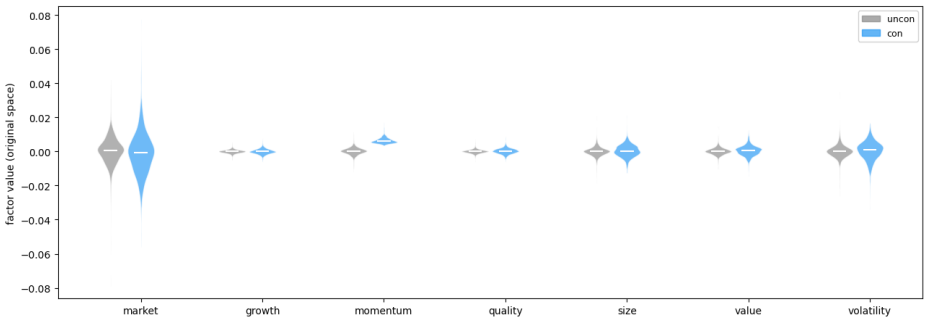


Figure A.11: Momentum rally: joint response of all seven factors for the unconditional, soft-rejection and guidance samples.

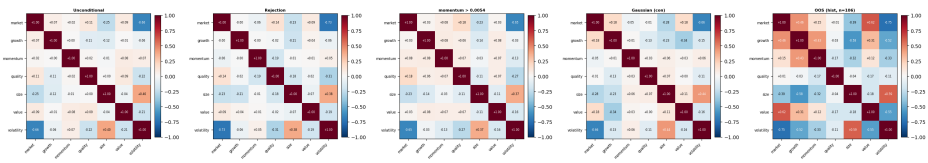


Figure A.12: Momentum rally: factor correlation matrices for the unconditional, soft-rejection, guidance and Gaussian samples and for the historical stress days.

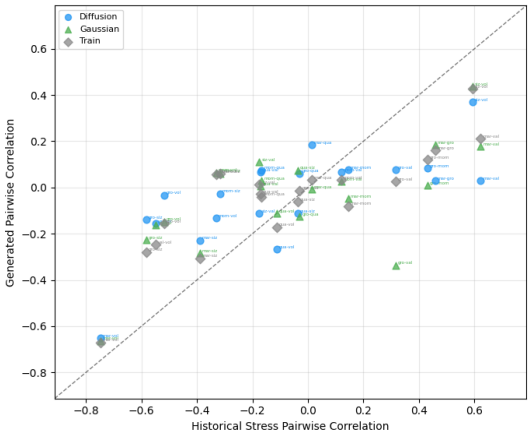


Figure A.13: Momentum rally: pairwise factor correlations of each method against the historical-stress correlations.

B

Notation

Symbols introduced in Chapter 3, grouped by topic.

Symbol	Meaning
Indexing and dimensions	
i	Stock index, $1 \le i \le N$
N	Number of stocks in the cross-section
K	Number of factors (= input dimension of the diffusion model)
T, t	Number of dates / date index in §3.1; **diffusion steps / step index** from §3.2 onwards
B	Minibatch size
M, K_z	Outer / inner Monte Carlo counts (`mc_outer`, `mc_inner`)
s	Synthesis-time sample subscript (§3.1); guidance strength (§3.2.2)
Factor model (§3.1)	
$R_{i,t}$	Stock i return at date t
R_t^f	Risk-free rate
$\mathbf{F}_t \in \mathbb{R}^K$	Factor return vector at date t, eq. (3.3)
a_i	Per-stock intercept (Jensen's alpha), eq. (3.2)
$\boldsymbol{\beta}_i \in \mathbb{R}^K$	Per-stock factor loadings, eq. (3.2)
$\varepsilon_{i,t}$	Per-stock idiosyncratic residual, eq. (3.2)
σ_i, ν_i	Residual standard deviation and Student-t degrees of freedom of stock i, eq. (3.4)
$\mathbf{b}_{i,t-1}$	Lagged firm characteristics of stock i, eq. (3.3)
$f_{0,t}$	Cross-sectional regression intercept (= market factor), eq. (3.3)
$u_{i,t}$	Cross-sectional regression residual, eq. (3.3)
Diffusion forward process (§3.2.1)	
$\mathbf{x}_0$, $\mathbf{x}_t \in \mathbb{R}^K$	Clean / noised factor vector at diffusion step t, eq. (3.5)
$\alpha \in (0, 2]$	Lévy stability index ($\alpha = 2$ recovers Gaussian/DDPM)
$S(\alpha, 0)$	Symmetric α-stable distribution
γ_t, σ_t, $\bar{\gamma}_t$, $\bar{\sigma}_t$	Single-step / cumulative DLPM schedule coefficients, eq. (3.6)
α_t, β_t	Cosine schedule scalars ($\beta_t = 1 - \alpha_t$, DDPM convention)
$A_t \sim S(\alpha/2, 1)$	Positive subordinator (per factor), eq. (3.7)
$\mathbf{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$	Gaussian draw
$\boldsymbol{\varepsilon}_t = \sqrt{A_t}\mathbf{L}\mathbf{z}_t$	Forward-step noise increment, eq. (3.9)
$\mathbf{C}$, $\mathbf{L}$	Empirical factor correlation and its Cholesky factor, $\mathbf{C} = \mathbf{L}\mathbf{L}^\top$
Σ_t	Accumulated forward variance given $A_{1:t}$, eq. (3.8)
Diffusion reverse process and conditional sampling (§3.2.2)	
$\hat{\boldsymbol{\varepsilon}}_\theta(\mathbf{x}_t, t)$	Learned noise-prediction network, eq. (3.12)
θ	Denoiser parameters
Γ_t	Posterior contraction factor, eq. (3.11)

Symbol	Meaning
$\boldsymbol{\mu}_t$	Reverse-step posterior mean, eq. (3.11)
$\hat{\mathbf{x}}_0$	Tweedie estimate of $\mathbf{x}_0$, eq. (3.14)
$\mathbf{c}$, $\mathcal{L}(\mathbf{x}_0, \mathbf{c})$	condition and condition loss eq. (3.13)
w_t, SNR_t, p	SNR-based decay weight, ratio and exponent, eq. (3.16)

C

Generated Samples

This appendix shows what one run of FaDiff produces. The run is the Markov-1 rollout of the temporal extension, unrolled for 22 trading days and repeated 500 times, which is the same sample the temporal evaluation of the main text is computed on. Each path carries all seven factors. Pushing every path through the frozen factor model, with a fresh residual draw per stock and per day, expands it to the 3887 stocks of the generated universe, stored in long format as `(path_id, csecid, date, returns)`.

Figure C.1 shows the factor level and Table C.1 the same sample as numbers at the end of the window. Figure C.2 shows the stock level, for the fifteen largest stocks by average market capitalisation over the window.

Table C.1: Cumulative factor return (%) over the 22 trading days of the window, showing the 5th percentile, median and 95th percentile across the 500 generated paths, and the realised value over the same window.

	Generated (500 paths)			
Factor	5%	Median	95%	Realised
Market	-4.0	+1.5	+6.0	+1.5
Growth	-0.6	0.0	+0.7	-0.9
Momentum	-1.1	+0.2	+1.6	+0.5
Quality	-0.7	0.0	+0.6	+0.3
Size	-0.9	+0.2	+1.2	-0.2
Value	-0.8	+0.2	+1.3	+0.2
Volatility	-1.7	+0.1	+2.3	-3.4

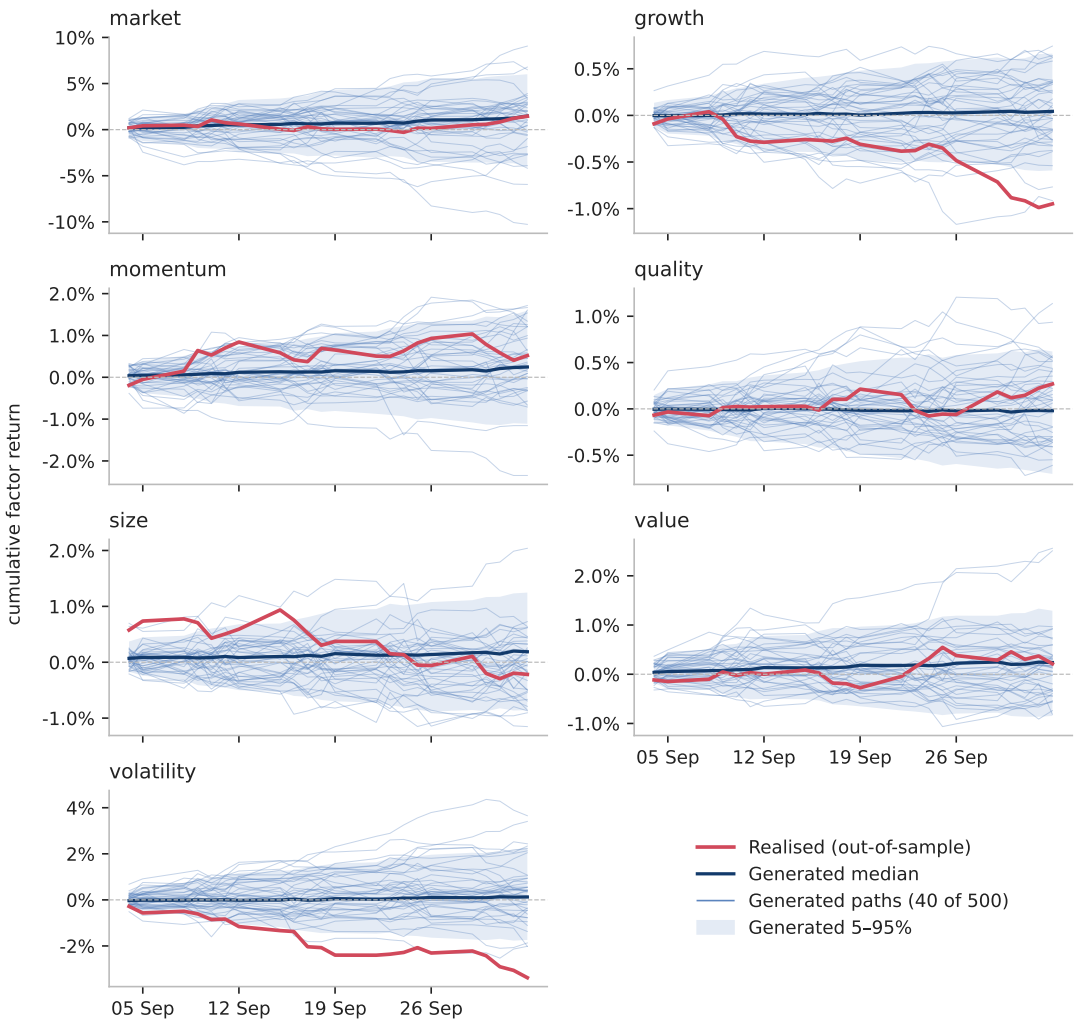


Figure C.1: Cumulative factor returns over the 22 trading days from 4 September to 3 October 2025. Forty of the 500 generated paths are drawn per factor, with the median and the 5 to 95% envelope taken over all 500; the realised out-of-sample path is in red. Vertical scales differ by panel.

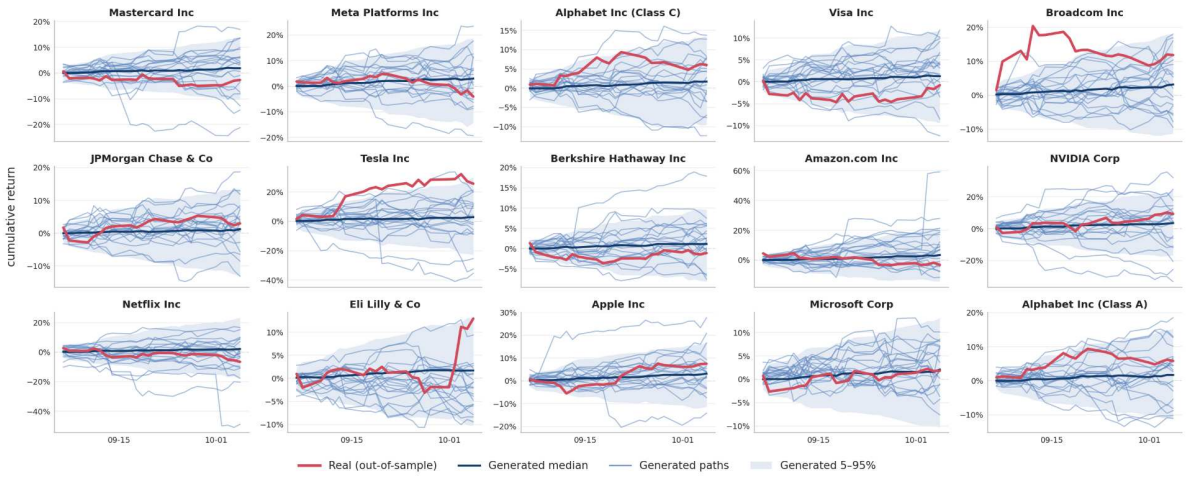


Figure C.2: Generated cumulative return paths over the 22 trading days from 4 September to 3 October 2025, for the fifteen largest stocks by average market capitalisation over that window. Twenty of the 500 generated paths are drawn per stock, with the median and the 5 to 95% envelope taken over all 500; the realised out-of-sample path is in red. Vertical scales differ by panel.

D

Use of Generative AI

Writing. I used Claude, an assistant based on a large language model, as a writing aid on drafts that I had already written. Its use covered proofreading for spelling and grammar, making the terminology consistent across chapters, shortening passages that repeated material stated elsewhere, rewriting sentences that were hard to follow, and enforcing formatting conventions in the LaTeX source.

Literature. The tool was also used to check bibliographic details of cited works, such as publication venues and author lists. Every reference in the bibliography was verified by me against the original publication. No citation was included on the basis of a tool's output alone, and no summary produced by a tool was used in place of reading the source.

Code. Generative AI tools were used for routine programming support in the implementation, such as boilerplate, plotting code and debugging hints.