Beyond Linear Diffusions: Improved Representations for Rare Conditional Generative Modeling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Diffusion models have emerged as powerful generative frameworks with widespread applications across machine learning and artificial intelligence systems. While current research has predominantly focused on linear diffusions, these approaches can face significant challenges when modeling a conditional distribution, $P(Y|X = x)$, when $P(X = x)$ is small. In these regions, few samples, if any, are available for training, thus modeling the corresponding conditional density may be difficult. Recognizing this, we show it is possible to adapt the data representation and forward scheme so that the sample complexity of learning a score-based generative model is small in low probability regions of the conditioning space. Drawing inspiration from conditional extreme value theory we characterize this method precisely in the special case in the tail regions of the conditioning variable, X. We show how diffusion with a data-driven choice of nonlinear drift term is best suited to model tail events under an appropriate representation of the data. Through empirical validation on two synthetic datasets and a real-world financial dataset, we demonstrate that our tail-adaptive approach significantly outperforms standard diffusion models in accurately capturing response distributions at the extreme tail conditions.

1 Introduction

In recent years, diffusion models have emerged as among the most powerful generative modeling techniques for synthesizing data across a diverse set of modalities. From image generation to audio synthesis and time series modeling, these models have demonstrated superior capabilities for capturing intricate data distributions as compared to other generative frameworks. The work [10] introduced denoising diffusion probabilistic models (DDPMs), which frame the generative process by defining a forward process that gradually transforms data into noise, followed by a learned reverse process that reconstructs data from noise. This approach has since been extended to a continuous-time formulation using Langevin diffusions, providing a mathematically elegant framework that connects stochastic processes with generative modeling. The continuous-time formulation views this as a stochastic differential equation (SDE), where the forward process follows a Langevin diffusion that converges to a standard multivariate Gaussian distribution. This perspective has enabled significant theoretical advances while maintaining state-of-the-art empirical performance across applications.

Conditional diffusion models extend this framework by additionally incorporating conditioning information to guide the generation process. However, a fundamental challenge emerges when dealing with extreme values in the sample space of the conditioning, where data is inherently sparse. Traditional diffusion models struggle to accurately capture conditional distributions in these tail regions, particularly when the underlying distributions deviate significantly from Gaussian assumptions. This limitation becomes especially problematic in domains where rare but consequential

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

events drive critical decisions, such as financial risk assessment and climate modeling. To effectively sample from a conditional distribution $P(Y|X = x)$ using score-based diffusion models, we need to estimate a sequence of score functions, $\{\nabla \log p_{\mu_t(.|x)}\}_{t=0}^{T}$. Here, $p_{\mu_t(.|x)}$ refers to the marginal density of the conditional distribution t steps into a (discretized) Langevin diffusion. The bottleneck is estimating these conditional score functions at low-probability, or rare, conditions. When $P(X = x)$ is small, it is unlikely that we see enough samples in our training data to estimate $\nabla \log p_{\mu_t(.|x)}$ accurately. If the score functions in these low probability regions have high sample complexity, sampling from tail conditions seems an improbable task.

We present a data-adaptive methodology for score-based diffusions that addresses this challenge through two key insights: (i) conditional diffusion requires learning complex functions with few samples, and (ii) function complexity can be controlled through the diffusion scheme and data transformation. Our method ensures the conditional denoising functions maintain low sample complexity where $P(X = x)$ is small, using data transformation and a data-driven nonlinear diffusion process. We demonstrate this approach in detail under mild extreme value assumptions. Specifically, our work explores nonlinear conditional diffusion modeling with tail-adaptive drift schemes. We examine the method where data follows extreme value assumptions [9, 8, 13]. Our contributions include:

1. We identify current limitations of standard linear diffusion models with Gaussian equilibrium for conditional generation under extreme tail conditions with limited samples, based on recent neural network sample complexity results.
2. We propose a novel score-based diffusion method that addresses the aforementioned sample complexity issue by utilizing well-designed data transformation and nonlinear Langevin diffusions. We explore this method in detail assuming the data follows some mild extreme value conditions (CEVT); although, we emphasize that our broader modeling philosophy is agnostic to any data assumptions.
3. We validate our method on synthetic and real financial datasets, demonstrating superior conditional distribution modeling at tail extremes compared to standard diffusion variants.

2 Background

2.1 The Difficulty of Conditional Diffusion

Conditional diffusion models frame sampling as the time reversal of a noising process governed by a diffusion SDE. A forward diffusion process $\{Y_t\}_{t=0}^{T}$ is indexed by a continuous time variable $t \in [0, T]$, such that $Y_0 \sim \mu_0(\cdot|X = x)$ is our sampling target, and $Y_T \sim \mu_T \approx \pi$, admits a tractable form to generate samples efficiently. A continuous time evolution, an Ito SDE, governs the forward process $\mu_0 \to \mu_T$. We limit ourselves to Langevin processes. The forward Langevin diffusion process is a stochastic differential equation of the form,

$$\mathrm{d}Y_t = -\nabla f(Y_t)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}B_t, \quad Y_0 \sim \mu_0(\cdot|X = x)\ . \tag{1}$$

where the conditional probability measure of Y_t is denoted $\mu_t(\cdot|x)$ with density $p_{\mu_t(\cdot|x)}$, $\{B_t\}_{t\geq 0}$ denotes a Brownian motion, and $\beta > 0$ is a scale parameter that the determines the noise level of the diffusion. Under mild conditions on f, this evolution admits e^{-f} as equilibrium density as $t \to \infty$. Backward denoising uses the reverse-time SDE [1]:

$$\mathrm{d}Y_t^{\leftarrow} = -(\nabla f(Y_t^{\leftarrow}) + 2\beta^{-1}\nabla \log p_{\mu_t(\cdot|X)}(Y_t^{\leftarrow}))\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\bar{B}_t, \quad Y_T^{\leftarrow} \sim \mu_T, \tag{2}$$

where we use $Y_t^{\leftarrow}$ to denote the time-reversal and $\{\bar{B}_t\}_{t\geq 0}$ denotes another Brownian motion.

2.1.1 Denoising Complexity

Implementing backward denoising process via (2) requires learning the conditional score function $\nabla \log p_{\mu_t(\cdot|x)}$. We instead target the estimation of the function, $\nabla f + \beta^{-1}\nabla \log p_{\mu_t(\cdot|x)}$. The complexity of these functions determines the sample size required for accurately estimation. For neural network predictors, non-asymptotic bounds have been established in [7] that relate target smoothness to estimation accuracy (see Theorem 2 in Appendix B).

Sampling from $P(Y|X=x)$, requires accurately learning the sequence of maps:

$$\{\mathbf{B}_t(y;x)\}_{t=0}^T = \{\nabla f(y) + \beta^{-1}\nabla \log p_{\mu_t(.|x)}(y)\}_{t=0}^T,\ \forall x \in \mathcal{X}$$

Since few training examples exists for rare events where $P(X=x)$ is small, accurate estimation of the denoising maps in these "rare regions" is futile, preventing effective denoising. More rigorously, we can adapt theoretical results from [19] to show that the accuracy of denoising is directly tied to how well the denoising maps are learned. Let μ_θ denote the estimated density resulting from (2) after appropriately estimating the score function sequence, $s_\theta(y;t,x) \approx \mathbf{B}_t(y;x)$. If μ_0 refers to the target, then the Kullback-Leibler (KL) divergence between the two can be upper bounded by the integrated error of score estimation (see Appendix B for a proof for completeness):

$$\begin{aligned} KL(\mu_0(\cdot|x)||\mu_\theta(\cdot|x)) &\lesssim \int_0^T \mathop{\mathbb{E}}_{p_{\mu_t(\cdot|x)}(y)}[\|(\nabla f(y) + \beta^{-1}\nabla \log p_{\mu_t(\cdot|x)}(y)) - s_\theta(y;t,x)\|^2]dt \\ &= \int_0^T \mathop{\mathbb{E}}_{p_{\mu_t(\cdot|x)}(y)}[\|(\mathbf{B}_t(y;x) - s_\theta(y;t,x)\|^2]dt. \end{aligned}$$

Linear Gaussian Dynamics: Standard score-based diffusion models employ the Ornstein-Uhlenbeck process (with $f(x) = \frac{1}{2}x^2$):

$$\mathrm{d}Y_t = -Y_t\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}B_t, \quad Y_0 \sim \mu_0(\cdot|x),$$

which yields at Gaussian stationary distribution. For this case, the denoising sequence in (2) becomes:

$$\{\mathbf{B}_t(y;x)\}_{t=0}^T = \{y + \beta^{-1}\nabla \log p_{\mu_t(\cdot|x)}(y)\}_{t=0}^T$$

As previously mentioned, when $P(X=x)$ is small and $\{\mathbf{B}_t(y;x)\}_{t=0}^T$ complex, this standard paradigm faces sample complexity challenges.

2.2 Extreme Value Theory

Extreme value theory characterizes the tail behavior of random variables. Classical work examines limiting behavior like $P(Y = y|Y > u) \to G(y)$ as $u \to \infty$. [9] extends this to conditional distributions $P(Y|X=x)$ for large x, contrasting with traditional multivariate theory where all variables grow simultaneously. The Heffernan-Tawn model [13] is a flexible approach to model the conditional distribution $P(Y|X=x)$ when x is large. For a broad class of dependency structures between X and Y, this work establishes a semi-parametric relationship that allows one to model a broad range of asymptotic independence/dependence structures at the tail of the condition (see Appendix A for more details).

Assumption 1 (CEVT [9, 13])**.** *Suppose the marginals of X and Y are standard Laplace. Then, as $X = x \to \infty$, we assume X, Y admit the asymptotic dependency,*

$$\lim_{x\to\infty} P\left(\frac{Y - a(X)}{b(X)} < z | X = x\right) = G(z) \tag{3}$$

where G is some distribution independent of X. In other words, for tail values, $X = x \to \infty$:

$$Y = a(X) + b(X)\cdot Z, \quad Z \sim G. \tag{4}$$

3 Proposed Methodology

In this section, we propose a general methodology that aims to ensure that denoising maps discussed in Section 2.1.1 maintain low sample complexity for rare conditions:

$$\{\mathbf{B}_t(y;x)\}_{t=0}^T \text{ is easy to estimate when } P(X=x) \text{ is small.} \tag{5}$$

While we demonstrate this approach under CEVT assumptions for explicit characterization, the framework applies broadly—any transformation yielding favorable tail behavior suffices. The general procedure consists of three steps:

1. Transform $(X,Y) \xrightarrow{T} (X^\star, Z)$ such that $P(Z|X^\star = x) \approx e^{-g}$ for rare x
2. Design forward diffusion with e^{-g} as the stationary density and train using the score-matching objective
3. Sample $Z \sim P(Z|X^\star = x)$ and apply inverse transformation to recover Y.

In the following, we describe an implementation of each of the above steps in the context of CEVT.

3.1 Step 1 – Data Transformation

When CEVT holds, we can obtain explicit transformations $(X, Y) \to (X^\star, Z)$ ensuring (5) for large $X^\star$. Consider the following chain of transformations applied to X and Y:

$$(X, Y) \overset{\text{Laplace Marginals}}{\to} (X^\star, Y^\star) \overset{\text{Normalize}}{\to} (X^\star, Z)\,, \tag{6}$$

where $Z = b(X^\star)^{-1}\left(Y^\star - a(X^\star)\right)$. In the first part of the transformation, we transform (X, Y) to $(X^\star, Y^\star)$ such that marginal distributions of both $X^\star$ and $Y^\star$ are standard Laplace distributions. Since $X^\star$ and $Y^\star$ are both standard Laplace, we can further apply a normalization based on the Heffernan-Tawn model to transform $Y^\star \to Z$, where

$$Z = \frac{Y^\star - a(X^\star)}{b(X^\star)}$$

To apply this normalization, we learn the functions $a(x)$ and $b(x)$ (which often take simple parametric form) using maximum likelihood estimation with samples from the tail of $X^\star$. Despite the small amount of samples available after partitioning the samples of $X^\star$ based on the tail, learning is plausible due to the simple structure of $a(x)$ and $b(x)$ (see Appendix A.1) for details. After applying these sequence of transformation, we have a set of data of the random variables (X_i, Z_i) that satisfy $P(Z|X^\star = x) \approx G$ for large values of x.

3.2 Step 2 – Learning the Conditional of Z

We learn the conditional distribution, $P(Z|X^\star)$, via score-based diffusion models. We provide pseudocode of our training procedure in Algorithm 3 in Appendix C.3. In the following, we describe the design of the forward process of our conditional diffusion as well as our approach to score matching. We also provide an intuitive argument as to why the denoising maps for this diffusion model have low sample complexity at the tails of the condition.

Designing the Forward Process We implement the forward process via a simple Langevin diffusion, but choose the drift term, ∇g, based on extreme value behavior in our observed data. In particular, by Assumption 1, for tail values in the condition, $\{X > x, x \text{ large}\}$, we model,

$$Z \sim G, \quad \text{and} \quad Z \perp X.$$

However, in practice, the distribution G is unknown. To approximate G, we train a lightweight density estimator on tail samples, $\{(X_i, Z_i) : \ X_i > x\}$, to gauge the density of G, e^{-g}. We do so by comparing the smooth estimate to common parametric forms. For example, a wide range of easy-to-sample distributions admit an exponential form, e^{-g}, with convex g, such as Gaussian, Laplace, and Gumbel.

Consider the following Langevin diffusion:

$$dZ_t = -\nabla g(Z_t)dt + \sqrt{2}\mathrm{d}B_t$$

The Langevin diffusion above, for arbitrary convex g does not admit path trajectories that can be expressed in closed form. In practice, we resort to a simple discretization,

$$Z_{t+1} = Z_t - \eta \cdot \nabla g(Z_t) + \sqrt{2\eta} \cdot \mathcal{N}(0, 1).$$

We remark that the convergence of the discretized process, which amounts to unadjusted Langevin dynamics, is sensitive to the curvature of g. We elaborate in Appendix C.1 how we can modify g so that it is appropriately smooth, while still accurately capturing a stationary distribution close to e^{-g}. We also remark that with a nonlinear drift term ∇g, we also lose the ancestral sampling property that score-based diffusions exploit for efficient training. That is, if ∇g is linear, then the t-step ahead distribution $P(Z_t|Z_0, X)$ is readily available. This is not possible for general g. Instead, in Appendix C.2 we show how Taylor-approximations can enable faster sampling in the general case, similar in line to [18].

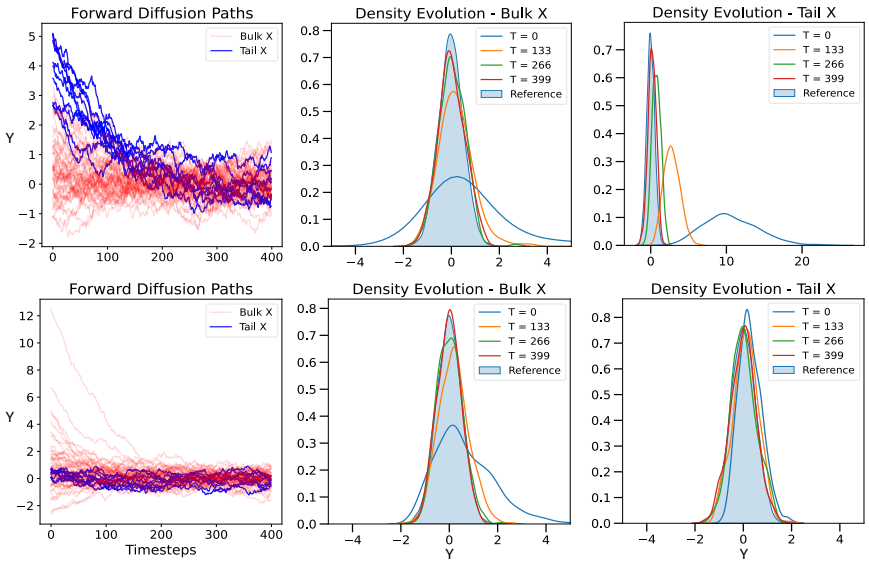


Figure 1: We visualize a forward diffusion before and after the transformation outlined in Section 3.2. Before transformation, the Langevin diffusion induces quite dramatic changes in the conditional density at tail events ($\{X = x\}$, x very large). This can be seen by looking at the blue particle paths (top left) or the evolving density, $p_{\mu_t(\cdot|x)}(y)$, visualized in the top right plot. After taking the steps outlined in Section 3.2, the tail conditional density does not change dramatically in the forward diffusion. Compare the new particle paths in blue (bottom left plot), or the new conditional densities at time t (bottom right plot). For tail, low-probability conditions, after transformation, the conditional density is already (nearly) at stationarity. Details can be found in Appendix A.2

Score Estimation Unlike standard diffusion models, rather than tracking the conditional score function, $\nabla \log p_{\mu_t(.|x)}$, we instead target $(\nabla g + \nabla \log p_{\mu_t(.|x)})$. We train a time-dependent conditional score model $s_\theta(z; x, t)$ based on a slightly modified learning objective.

$$\mathcal{L}(\theta) := \mathop{\mathbb{E}}_{t} \left\{ \lambda(t) \mathop{\mathbb{E}}_{X,Z_0} \mathop{\mathbb{E}}_{Z_t|Z_0} [\| s_\theta(z; x, t) - (\nabla \log p_{\mu_{0t}(.|Z_0,X)}(Z_t)) + \nabla g(Z_t))\|_2^2] \right\}, \tag{7}$$

where $\lambda(t)$ is a weighting function that adjusts the importance of different time steps for the score-matching loss. Recent works explore how to learn the conditional score functions $s_\theta(Z; x, t)$ efficiently. For simplicity, we train based on the standard formulation based on Tweedie's formula.

Why this Works? Since Z is constructed based on Assumption 1, at an event $\{X^\star = x\}$, with x large, the initial density of $\mu_0(Z|X^\star = x)$ will already be (approximately) at equilibria:

$$p_{\mu_0(\cdot|X^\star=x)} \approx e^{-g}$$

Thus, at these extreme values of $X^\star$, the sequence of maps $\{\nabla g + \nabla \log p_{\bar{\mu}_t(.|x^\star)}\}_{t=0}^T \approx 0$, making them much easier to estimate. We display an example of this in Figure 1.

3.3 Step 3 – Sampling

For a desired value of X, we prompt the learned diffusion model to retrieve a sample Z from $P(Z|X = x)$. Sampling is implemented via time-reversal as in (2) and substituting in the learned estimator, $s_\theta(z; x, t)$.

$$\mathrm{d}\bar{Z}_t = -(2s_\theta(Z_t; x, t) - \nabla f(\bar{Z}_t))\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\bar{B}_t, \quad \text{or,} \tag{8}$$

$$\mathrm{d}\bar{Z}_t = -s_\theta(Z_t; x, t)\mathrm{d}t \tag{9}$$

In practice, we use a simple Euler-Maryama discretization of the above to sample. Once we have a sample $Z \sim \hat{P}(Z|X)$, we convert it to a sample from our desired distribution by inverting the

sequence of transformations,

$$Y^\star = a(X^\star) + b(X^\star) \cdot Z$$
$$Y = \hat{F}_Y^{-1}(F_{Lap}(Y^\star))$$

Algorithm 4 in Appendix C.3 provides pseudocode for our sampling procedure.

3.4 Generalization

As previously mentioned, there is potential to adapt **Step 1** of the process to more general circumstances (e.g., if the CEVT assumption is not appropriate for the data of interest). The challenge of adopting the methodology is the finding the appropriate transformation of the data using a data-driven approach, perhaps using an approach similar to [11]. We leave this for future work.

4 Experiments

In this section, we evaluate our proposed approach on two synthetic data examples and a real data example. For baselines, we consider two schemes for denoising. In the standard scheme, we sample $Y_T \sim \mathcal{N}(0, 1)$, and provided a condition $X = x$, we apply the maps,

$$\{\mathbf{B}_t^{Gauss}(y; x)\}_{t=0}^T$$

In the new scheme, we first transform our data, $Y \overset{T}{\to} Z$. Sample $Z_T \sim e^{-g}$ and apply the maps,

$$\{\mathbf{B}_t^{g}(z; x)\}_{t=0}^T$$

Finally, invert the transform, $Z_0 \overset{T^{-1}}{\to} Y_0$. To fairly compare our new scheme to the standard scheme we make the following considerations.

Neural Net Parametrization: Fundamentally, we want to track how well $\mathbf{B}_t^{Gauss}, \mathbf{B}_t^g$ are learned. As a proxy we will look at sample quality. To enable a fair comparison, we deploy the same neural network architectures to learn each score network, which are standard feedforward neural networks.

Forward Chain Length: It is important to recognize that although the sequence of standard denoising maps, $\{\mathbf{B}_t^{Gauss}(y; x)\}_{t=0}^T$, may have high sample complexity, due to the fast convergence of the forward OU process, the number of noise-steps necessary, T, may be smaller. This, in turn, may be beneficial for learning. For example, if one were to train a separate network for each noise-scale $t \in [T]$. We broach this gap by considering smoothed versions of ∇g for the generic scheme. This directly impacts speed of forward convergence, and is detailed in Appendix C. By choosing the smoothing parameters and step-size η appropriately, we are able to use the same number of noise steps for each model. This compromise, between complexity of $\mathbf{B}_t^g(y; x)$, η and size of T needs to be explored more rigorously. We leave this to future work.

4.1 Synthetic Data Examples

We consider two synthetic data experiments: a mean-shifted Laplace distribution; and correlated Gaussian distribution. We provide detailed plots with additional results for both synthetic examples can be found in Appendix D.1.

Mean-Shifted Laplace Target. We consider the following data generating process:

$$X \sim \text{Pareto}(1), \quad Y \sim \frac{10}{X} + \text{Laplace}(0, 1) \tag{10}$$

Without appealing to CEVT, we see that as $X \to \infty$, $Y \sim \text{Laplace}(0, 1)$. This suggests we target standard Laplace as the equilibrium distribution of the forward process, without applying any transformation to the data. We run,

$$Y_{t+\eta} = Y_t - \eta \cdot \nabla g_{Lap*}(Y_t) + \sqrt{2\eta} \cdot \mathcal{N}(0, 1).$$

Refer to Appendix C.1 to see form and justification for ∇g_{Lap*}. We plot a comparison of the new method and standard Gaussian diffusion in Figure 2a. The results of the figure demonstrate that in 90% percentile, a standard diffusion model with Gaussian base distribution does not estimate the target distribution well, while the proposed approach without the CEVT transformation and an appropriately chosen Laplace base distribution more accurately capture the target.

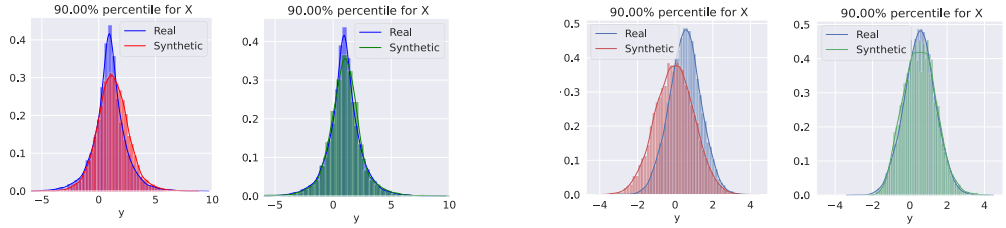
Correlated Gaussian Target. We consider the following data generating process:

$$\begin{bmatrix} X \\ Y \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right), \tag{11}$$

where we set $\rho = 0.4$. First we transform $(X, Y) \to (X^\star, Z)$ as per Algorithm 2. As detailed in the Appendix A.2, we know after this transform, $G \sim \mathcal{N}(0, 2\rho^2(1-\rho^2))$. However, to mimic the data-driven procedure in practice, we instead gauge a form for e^{-g} using tail samples. Based on this, we suggest targeting Gumbel$(0, 0.4)$ and run,

$$Z_{t+\eta} = Z_t - \eta \cdot \nabla g_{Gumb*}(Y_t) + \sqrt{2\eta} \cdot \mathcal{N}(0, 1).$$

Refer to Appendix C.1 to see form and justification for ∇g_{Gumb*}. Once we sample from $P(Z|X^\star = x)$ via the new score-based diffusion, we transform back to the appropriate distribution via inverse CDF. We compare these samples to a traditional (linear) diffusion model that targets sampling from $P(Y|X = x)$. We plot this comparison in Figure 2. From the figure, we can observe that the standard diffusion model fails to capture the target distribution at the tail of the condition, while the proposed method with the Gumbel base distribution almost perfectly captures it.



(a) Mean-Shifted Laplace Target.

(b) Correlated Gaussian Target.

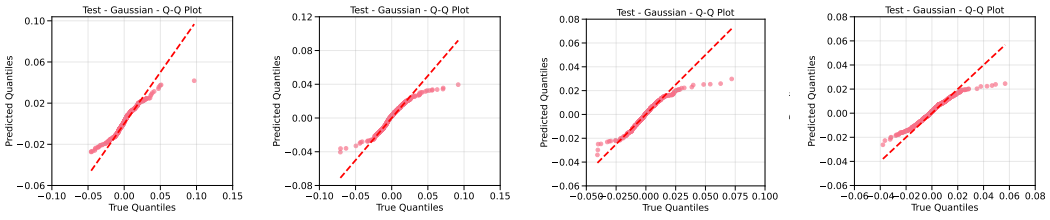
Figure 2: In each subfigure, the left plot shows the standard diffusion with Gaussian base distribution, and the right plot shows our proposed method with a standard Laplace base distribution for the mean-shift example (no transformation) and a Gumbel base distribution for the multivariate Gaussian example (with learned CEVT transformation).

4.2 Stock Returns Conditioned on Volatility Index

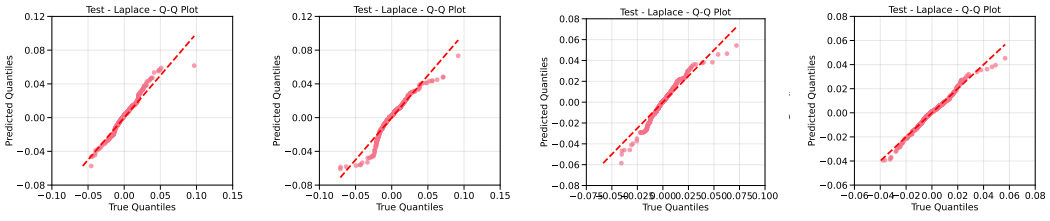
The VIX Indexis a time-series that measures market expectations of near-term volatility conveyed by S&P 500 stock index option prices. A high VIX index typically signals a period of financial stress, as observed during major economic disruptions such as the Global Financial Crisis (GFC) in 2008 and the COVID-19 pandemic in 2020, when the VIX reached elevated levels. In this study, we apply our methodology to real-world data to model the returns of selected financial assets during periods of heightened market volatility. Our objective is to evaluate the proposed method by modeling the returns of financial assets conditioned on a measure of market risk. Specifically, we assess the performance of our approach in generating the marginal returns of a mix of technology and financial stocks during stressed market regimes, using the volatility index VIX as a conditioning factor. The stocks analyzed include AAPL, MSFT, GOOGL, NVDA, AMZN, JPM, WFC, and GS. We focus on two significant periods: the 2008 Global Financial Crisis and the 2020 COVID-19 pandemic. For each period, we establish distinct training and testing phases to evaluate generative performance::

- **GFC**: we use training data from 01/01/2005-12/31/2007 and evaluate on the testing data from 01/01/2008-12/31/2009.
- **COVID**: we use training data from 01/01/2017-12/31/2019 and evaluate on the testing data from 01/01/2020-12/31/2021.

For baselines, we compare a standard linear diffusion (Gaussian base) and our proposed methodology with CEVT-based transformation and a Laplace base distribution. We provide more information on the VIX and plots of it during both periods for both the training and test data in the Appendix D.2, which demonstrate the prevalence of more extreme conditions in the testing dataset for both periods.

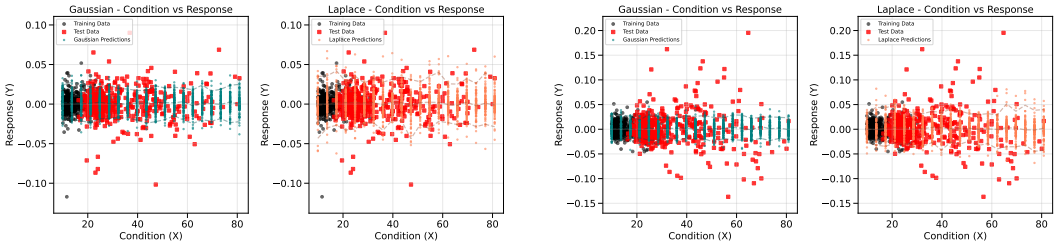


(a) AAPL - Gaussian (b) AMZN - Gaussian (c) GOOGL - Gaussian. (d) MSFT - Gaussian



(e) AAPL - Laplace (f) AMZN - Laplace (g) GOOGL - Laplace (h) MSFT - Laplace

Figure 3: QQ plots on test datasets for COVID period for various technology stocks.



(a) MSFT (b) GS

Figure 4: Performance comparison of Gaussian versus Laplace base distributions based on different values of VIX level for the GFC regime.

Our results demonstrate that, in this example, employing a nonlinear diffusion model offers a clear advantage in capturing the unconditional heavy-tailed behavior of stock returns, while also enhancing the modeling of conditionals for high VIX levels. For instance, as illustrated in the QQ plots in Figure 3, we observe that when capturing the marginal distribution of returns for various technology stocks during the COVID period, utilizing a Laplace base distribution outperforms its Gaussian counterpart in the tails, while maintaining good calibration in the bulk of the distribution. Regarding performance on the conditionals, we observe that during the GFC period, selecting a Laplace base distribution more effectively captures tail behavior as VIX values increase, despite these high VIX levels not being present during training. We offer more detailed plots analyzing the results for each stock across both periods in the Appendix D.2.

5 Conclusions and Future Work

In this work, we propose a methodology for improving rare event sampling in conditional generative modeling based on nonlinear score-based diffusion models. Motivated by conditional extreme value theory, we show that under some transformation of the data, we can choose the equilibrium distribution of the Langevin diffusion that is more advantageous from a sample complexity perspective for our learning problem. We provide numerical simulations on two toy experiments and a practical application of risk modeling for financial returns and demonstrate we can better capture the response distribution for extreme tails of the condition variable. From a practical perspective, challenges pertaining to our work include incorporating data-driven learning of the feature transformation process, extension to high-dimensional conditional variables, and a comprehensive performance comparison across multiple generative models on a larger pool of datasets.

References

[1] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Sinho Chewi. Log-concave sampling. *Book draft available at https://chewisinho. github. io*, 9:17–18, 2023.

[3] Arnak Dalalyan. Further and stronger analogy between sampling and optimization: Langevin monte carlo and gradient descent. In *Conference on Learning Theory*, pages 678–689. PMLR, 2017.

[4] Arnak S Dalalyan and Alexandre B Tsybakov. Sparse regression learning by aggregation and langevin monte-carlo. *Journal of Computer and System Sciences*, 78(5):1423–1443, 2012.

[5] Holger Drees and Anja Jansen. Conditional extreme value models: fallacies and pitfalls. *Extremes*, 20(4):777–805, 2017.

[6] Alain Durmus and Eric Moulines. Nonasymptotic convergence analysis for the unadjusted langevin algorithm. 2017.

[7] Max H Farrell, Tengyuan Liang, and Sanjog Misra. Deep neural networks for estimation and inference. *Econometrica*, 89(1):181–213, 2021.

[8] Janet E Heffernan and Sidney I Resnick. Limit laws for random vectors with an extreme component. 2007.

[9] Janet E. Heffernan and Jonathan A. Tawn. A conditional approach for multivariate extreme values (with discussion). *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(3):497–546, 07 2004.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] Tianyang Hu, Fei Chen, Haonan Wang, Jiawei Li, Wenjia Wang, Jiacheng Sun, and Zhenguo Li. Complexity matters: Rethinking the latent space for generative modeling. *Advances in Neural Information Processing Systems*, 36:29558–29579, 2023.

[12] H. Joe. *Multivariate Models and Multivariate Dependence Concepts*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1997.

[13] Caroline Keef, Ioannis Papastathopoulos, and Jonathan A. Tawn. Estimation of the conditional distribution of a multivariate variable given that one of its components is large: Additional constraints for the heffernan and tawn model. *Journal of Multivariate Analysis*, 115:396–404, 2013.

[14] Tengyuan Liang, Kulunu Dharmakeerthi, and Takuya Koriyama. Denoising diffusions with optimal transport: Localization, curvature, and multi-scale complexity. *arXiv preprint arXiv:2411.01629*, 2024.

[15] Roger B Nelsen. *An introduction to copulas*. Springer, 2006.

[16] Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

[17] Sidney I Resnick and David Zeber. Transition kernels and the conditional extreme value model. *Extremes*, 17(2):263–287, 2014.

[18] Raghav Singhal, Mark Goldstein, and Rajesh Ranganath. What's the score? automated denoising score matching for nonlinear diffusions. *arXiv preprint arXiv:2407.07998*, 2024.

[19] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

A CEVT details

We restate the CEVT modeling assumption for convenience.

Assumption 2 (CEVT [9, 13])**.** *Suppose the marginals of X and Y are standard Laplace. Then, as $X = x \to \infty$, we assume X, Y admit the **asymptotic** dependency,*

$$\lim_{x\to\infty} P\left(\frac{Y-a(X)}{b(X)} < z | X = x\right) = G(z)$$

where G is some distribution independent of X. In other words, for tail values, $X = x \to \infty$, we model,

$$Y = a(X) + b(X) \cdot Z, \quad Z \sim G,$$

In a slightly different formulation, [8] establish that, so long as the conditioning variable X belongs to the domain of attraction of an extreme value distribution, such an assumption about asymptotic behavior is reasonable. More recently, [17] directly related the Heffernan Tawn model to the more general formulation of [8] and found parsimony under some mild conditions. We emphasize that this modeling assumption is theoretically grounded. A growing body of applied statistical methods successfully apply this model, further strengthening its relevance in practice.

Importantly, [9, 13] found that for all standard copula forms of dependence outlined in [12, 15], the functions $a(X), b(X)$ admit simple parametric forms, thus, the limiting form G can be assessed with a relatively small amount of samples. This insight motivates an approach to extrapolating to the tail in conditional score-based diffusion models.

A.1 Normalizing Functions

For a variety of relationships between X and Y, G has a log-concave density and the normalizing functions a and b admit simple forms [9, 13].

Suppose X and Y are marginally Laplace. Then for some suitably high threshold, $x \in \mathbb{R}$ the conditional relationship at the tail values, $X > x$, approximately satisfy,

$$Y = a \cdot X + X^b \cdot Z, \quad Z \sim G, \quad a \in [-1, 1],\ b \in (-\infty, 1).$$

For a detailed examination of this relationship and clear delineation of when this simple form arises a reader should refer to the original work [9] or the follow-up [13]. In particular, Table 1 in [9]. For failure cases a reader can refer to [5]. We assume for our examples that $a(x)$ and $b(x)$ admit this simple structure.

In practice, the scalars a and b are estimated. It is possible to learn these parameters via constrained optimization. The simplest approach , which we implemented, is to assume $Z \sim \mathcal{N}(0, 1)$ and implement maximum likelihood with tail data $\{(X_i, Y_i) : X_i > x\}$.

A.2 Toy Example

As an example, suppose,

$$(X, Y) \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\right)$$

First, transform the variables to have Laplace marginals, $(X, Y) \to (X^\star, Y^\star)$ (e.g., Inverse CDF Transform). For this example, the normalizing functions admit an explicit form,

$$Z = \frac{Y^\star - a(X^\star)}{b(X^\star)}, \quad a(x) = \text{sign}(\rho) \cdot \rho^2 \cdot x,\ b(x) = x^{1/2}$$

In this regime, it is well understood [13] that,

$$\mathbf{P}(Z | X^\star = x^\star) \to \mathcal{N}(0, 2\rho^2(1-\rho^2)), \quad \text{as } x^\star \to \infty.$$

So, setting $g(x) = \frac{1}{2}x^2,\ \beta = (2\rho^2(1-\rho^2))^{-1/2}$, our new forward diffusion is a scaled OU process that admits $G = \mathcal{N}(0, 2\rho^2(1-\rho^2))$ as equilibrium.

We visualize the diffusion process, before and after transformation, in Figure 5. Comparing the plots in the left column,it is clear that the path evolution of particles Y_t that correspond to large, tail values in X (bottom right, depicted in blue) are much more regular after the transformation. We also plot the conditional densities $\bar{\mu}_t(.|x)$, for a collection of timesteps and both bulk and tail events $\{X = x\}$. Before the transformation, $\bar{\mu}_t(y|x)$ changes quite drastically across the forward chain. However, after transformation (see bottom right figure), $\bar{\mu}_t(z|x^\star) \propto G$ for tail values $X^\star = x^\star$. Indeed, we see that at the tail values of the condition, $x^\star \to \infty$, the forward process is already at stationarity. In other words,

$$\nabla g(y) + \beta^{-1}\nabla \log \rho_{\mu_t(.|x^\star)}(y) = y - y = 0, \quad \forall t, \qquad \text{(easy to learn)}$$

And so, where we have few samples, we have a sequence of functions that may be estimated with few samples.

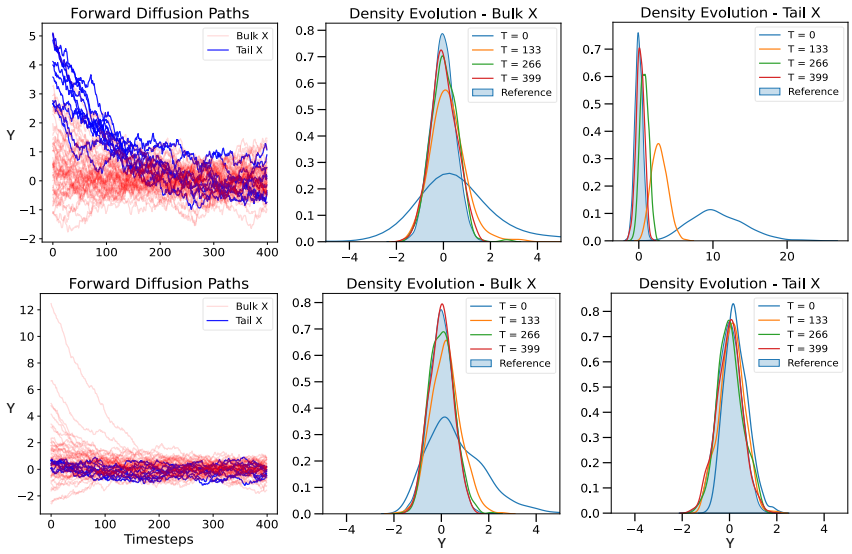


Figure 5: Top row: Before transformation. Bottom row: After transformation.

B Theory

A simple change to Theorem 1 in [19] will reflect our change in target for estimation. For completeness we include the theorem below and detail the small modification to the proof. We state the result for unconditional densities, but the result follows for conditional densities without loss of generality.

Theorem 1. *Denote by $p(y)$ the target density. Let $\{Y_t\}_{t\in[0,T]}$ be the stochastic process defined by the SDE in 1, where $Y_0 \sim p$ and $Y_t \sim p_t$. Suppose $\pi(y)$ is the stationary density of this SDE as $T \to \infty$. Let $\hat{Y}_0^{\leftarrow} \sim p_\theta(y)$ be the result of the approximate reverse-time SDE where we substitute our score model, $s_\theta(y,t)$.*

$$\mathrm{d}\hat{Y}_t^{\leftarrow} = -(2s_\theta(\hat{Y}_t^{\leftarrow}, t) - \nabla f(\hat{Y}_t^{\leftarrow}))\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\bar{B}_t, \quad \hat{Y}_T^{\leftarrow} \sim \pi \tag{12}$$

Under some regularity conditions (see Appendix A [19]),

$$KL(p||p_\theta) \leq \int_0^T \mathop{\mathbb{E}}_{p_t(y)}[\| \left(\nabla f(y) + \beta^{-1}\nabla \log p_t(y)\right) - s_\theta(y,t)\|^2]dt + KL(p_T||\pi).$$

Proof. Denote the path measure of $\{Y_t\}_{t\in[0,T]}$ and $\{\hat{Y}_t^{\leftarrow}\}_{t\in[0,T]}$ by μ and ν. Recall $Y_0 \sim p$ and $Y_T \sim p_T$, whereas $\hat{Y}_0^{\leftarrow} \sim p_\theta$ and $\hat{Y}_T^{\leftarrow} \sim \pi$. Following the line of argumentation in [19], we establish

by data-processing inequality (1), and chain rule (2),

$$KL(p||p_\theta) \overset{1}{\leq} KL(\mu||\nu) \overset{2}{\leq} KL(p_T||\pi) + \mathop{\mathbb{E}}_{z\sim p_T}[KL(\mu(\cdot|Y_t = z)|||\nu(\cdot|\hat{Y}_T^{\leftarrow} = z))].$$

What remains is to tackle the second term on RHS. Due to time-reversal, the path measure $\{Y_t\}_{t\in[0,T]}$ can be equivalently seen as generated by the reverse time SDE,

$$\mathrm{d}Y_t^{\leftarrow} = -(\nabla f(Y_t^{\leftarrow}) + 2\beta^{-1}\nabla \log p_t(Y_t^{\leftarrow}))\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}\bar{B}_t, \quad Y_T^{\leftarrow} \sim \pi \tag{13}$$

Then, $KL(\mu(\cdot|Y_t = z)|||\nu(\cdot|\hat{Y}_T^{\leftarrow} = z))$ can be calculated by comparing the following reverse-time SDEs initialized at the same point:

$$dY_t^{\leftarrow} = -(\nabla f(Y_t^{\leftarrow}) + 2\beta^{-1}\nabla \log p_{\mu_t(\cdot|x)}(Y_t^{\leftarrow}))dt + \sqrt{2\beta^{-1}}dB_t, \quad Y_T^{\leftarrow} = z \tag{14}$$

$$d\hat{Y}_t^{\leftarrow} = -(2s_\theta(\hat{Y}_t^{\leftarrow}; x, t) - \nabla f(\hat{Y}_t^{\leftarrow}))dt + \sqrt{2\beta^{-1}}dB_t, \quad \hat{Y}_T^{\leftarrow} = z \tag{15}$$

Since these SDES share the same diffusion coefficient and starting point, we can appeal to Girsanov's theorem [16] to see,

$$KL(\mu(\cdot|Y_t = z)|||\nu(\cdot|\hat{Y}_T^{\leftarrow} = z)) \leq \int_0^T \mathop{\mathbb{E}}_{p_t(y)}[\| \left(\nabla f(y) + \beta^{-1}\nabla \log p_t(y)\right) - s_\theta(y, t)\|^2]dt$$

□

We adopt the following non-asymptotic bound from [7] with regard to the sample complexity of minimizing the squared error in a multi-layer perceptron neural network.

Theorem 2. *Let $\widehat{f}_{\mathrm{MLP}}$ denote a standard multi-layer perceptron. Under the assumption that the target function $f_\star = \nabla f + \beta^{-1}\nabla \log p_{\mu_t(\cdot|x)}$ lies in the Sobolev ball $\mathcal{W}^{S,\infty}([-1,1]^d)$ with smoothness parameter $S \in \mathbb{N}^+$, then with probability at least $1-\delta$ where $\delta = \exp\left(-n^{\frac{d}{S+d}}\log^8 n\right)$, for large enough n:*

$$\|\widehat{f}_{\mathrm{MLP}} - f_\star\|_{L_2(x)}^2 \leq C\left(n^{-\frac{S}{S+d}}\log^8 n + \frac{\log\log n}{n}\right) \tag{16}$$

Intuitively, the "rougher" the function (the smaller the value of S) and the higher the input dimension d, a larger number of samples are needed to estimate the target function $f_\star$.

C Methodology Details

C.1 Smoothness of f

We implement the Euler-Maryama discretization of the forward diffusion, 1. This amounts to Unadjusted Langvevin Algorithm (ULA). It is well established that the convergence speed of ULA depends on the gradient of our drift term, $\nabla^2 f$ (developed in a sequence of works [4, 3, 6]). We present a result condensed in [2], and for simplicity, specialized to dimension, $d = 1$.

Theorem 3 (Convergence of ULA [2])**.** *Suppose that $\pi \propto e^{-f}$ is the target distribution and f satisfies $\alpha \leq \nabla^2 f \leq \beta$. Define $\kappa = \beta/\alpha$ as the condition number and $\mu_{t\cdot\eta}$ as the $t-$th measure in the sequence. Then, for any $\epsilon \in [0,1]$, with step size $\eta \asymp \epsilon^2/\beta\kappa$, we obtain that after,*

$$T = O\left(\frac{\kappa^2}{\epsilon^2}\log\frac{\alpha W_2^2(\mu_0,\pi)}{\epsilon^2}\right) \quad \textit{iterations,}$$

$$\alpha W_2^2(\mu_{T\cdot\eta}, \pi) \leq \epsilon^2$$

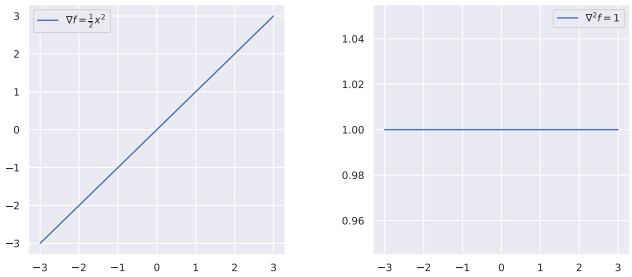
In our methodology, we propose choosing a convex f to target a specific distribution, e^{-f}, that reflects the tail characteristics of our target conditional distribution, $P(Y|X = x)$. However, choosing f with poor curvature directly impacts speed of forward process. This in turn impacts how many noising steps, $[T]$, are necessary to diffuse-then-denoise and can detriment computational efficiency. This is particularly relevant when part of our argument concerns out-performing Gaussian diffusions. However, when $e^{-f} \propto e^{-x^2/2}$, $\kappa = 1$ and convergence is fast.

To overcome this we use appropriately smoothed versions of the new target density, $e^{-f^\star}$. We smooth in such a way that κ is bounded, $f^\star$ is continuously differentiable, but $e^{-f^\star} \approx e^{-f}$. In the backward process, we still initiate samples by drawing from e^{-f}. We emphasize that this does not impact the quality of the method.

- We show below that by appropriately choosing smoothing parameters, the forward process converges to a distribution very similar to the target, e^{-f}.
- Small perturbations between the end of the forward process ($e^{-f^\star}$) and start of the backward process (e^{-f}) is theoretically negligible [14]. Even with standard schemes, owing to the finite time steps $T < \infty$, the end of the forward proccess will not be exactly Gaussian.

Below are examples relevant to this paper.

Gaussian The standard scheme is to target $e^{-f} \propto e^{-x^2/2}$, standard Gaussian density. In this case, $\kappa = 1$.



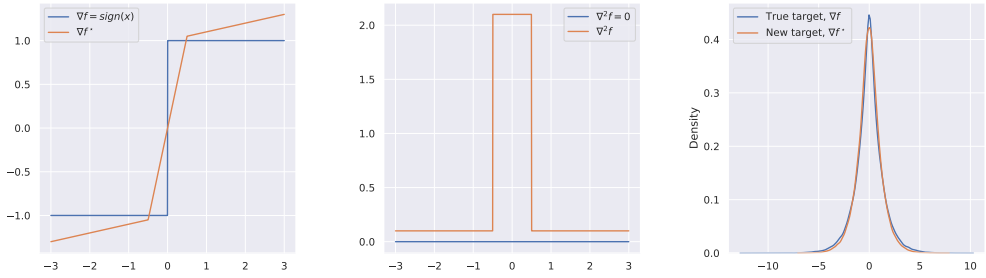
(a) (b)

Figure 6: (a): Plot of ∇f. (b): Plot of $\nabla^2 f$.

Laplace Suppose we want to target $e^{-f} \propto e^{-|x|}$. Then, $\nabla^2 f = 0$ and f is not continuously differentiable (at 0). Convergence theorem for ULA suggests potential problems. Instead, we consider a smooth approximation,

$$\nabla f^\star(x, b, c) = \begin{cases} \frac{1}{b} \cdot x + c \cdot x, & \text{if } x \in (-b, b), \\ \text{sign}(x) + c \cdot x, & \text{otherwise.} \end{cases}$$

Here, $b,\ c \geq 0$ are user specified constants. If $b, c = 0$, then we arrive at ∇f. This is simply the gradient of the Huber function with a linear perturbation by $c \cdot x$. With this smoothing, $\kappa = 1 + \frac{1}{bc}$.



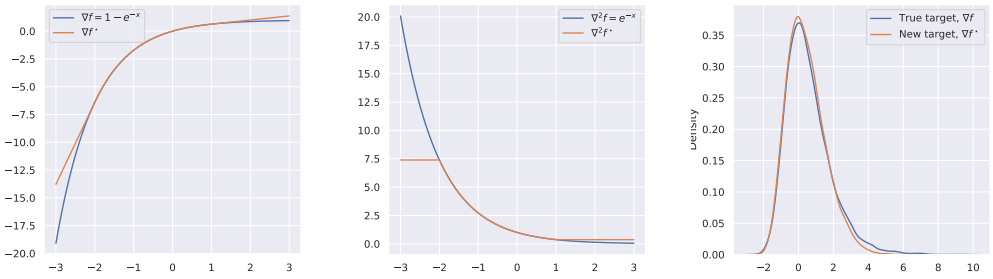
(a) (b) (c)

Figure 7: Set $b = 0.5,\ c = 0.1$. (a): Plot comparing ∇f and $\nabla f^\star$. (b): Plot comparing $\nabla^2 f$ and $\nabla^2 f^\star$. (c): Comparing densities after running ULA ($\eta = 0.01,\ T = 1000$ with ∇f and $\nabla f^\star$.

Huber Suppose we want to target $e^{-f} \propto e^{-(x+e^{-x})}$. Then, $\nabla^2 f = e^{-x}$ which is not bounded above, and approaches $\to 0$ as $x \to \infty$. We consider a smooth approximation,

$$\nabla f^\star(x, b, c) = \begin{cases} e^b, & \text{if } x \leq -b, \\ e^{-x}, & \text{if } -b < x < c, \\ e^{-c}, & \text{if } x \geq c. \end{cases}$$

Here, $b,\ c \geq 0$ are user specified constants. If $b, c = 0$, then we arrive at ∇f. With this smoothing, $\kappa = e^{b+c}$.



(a) (b) (c)

Figure 8: Set $b = 2,\ c = 1$. (a): Plot comparing ∇f and $\nabla f^\star$. (b): Plot comparing $\nabla^2 f$ and $\nabla^2 f^\star$. (c): Comparing densities after running ULA ($\eta = 0.01,\ T = 1000$ with ∇f and $\nabla f^\star$.

C.2 Taylor Accelerated Forward Diffusion

An important practical consideration for our training algorithm is the efficiency in the estimation of the score function. In our work, we utilize the Euler-Maruyama approximation in order to sample Z_t given Z_0. In practice, this can be inefficient, since it requires $\mathcal{O}(t)$ sampling steps to sample. For a given time $t_\star \in \{1, \ldots, T\}$, direct score estimation based on Euler–Maruyama is given by:

$$Z_0 \sim \mathcal{D} \tag{17}$$

$$Z_t = Z_{t-1} - \eta \nabla f(Z_{t-1}) + \sqrt{2\eta} \cdot \mathcal{N}(0, 1), \quad t = 1, \ldots, t_\star \tag{18}$$

A linear SDE can be solved more easily and allows for ancestral sampling, where $Z_t | Z_0$ can be sampled in a single step. As an example, consider the Ornstein–Uhlenbeck (OU) process:

$$dZ_t = \theta(\mu - Z_t)dt + \sigma dW_t \tag{19}$$

and its Euler-Maruyama discretized counterpart:

$$Z_t = Z_{t-1} + \theta(\mu - Z_{t-1}) + \sigma\epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, 1). \tag{20}$$

The discretized process can alternatively be parameterized as:

$$Z_t = (1 - \theta)Z_{t-1} + \theta\mu + \sigma\epsilon_t \tag{21}$$

which allows for straightforward derivation of the conditional $p(Z_t | Z_0)$:

$$p(Z_t | Z_0) = \mathcal{N}\left(Z_t; \alpha^t Z_0 + (1 - \alpha^t)\mu, \sigma^2 \left(\frac{1 - \alpha^{2(t+1)}}{1 - \alpha^2}\right)\right) \tag{22}$$

As we can see above, sampling from a linear SDE like the OU process is easy and does not require multiple rounds of a solver. One idea to make sampling from a nonlinear SDE more efficiently is to consider a first-order Taylor of the score. Particular to this paper, consider a Langevin diffusion with score function $s(Z) = -\nabla_Z f(Z)$, which we know converges to $p(Z_\star) \propto e^{-f(Z)}$ at equilibrium. Consider the first-order Taylor approximation to the score centered around $\tilde{Z}$:

$$s(Z) \approx s(\tilde{Z}) + \nabla_{\tilde{Z}} s(\tilde{Z})(Z - \tilde{Z}) \tag{23}$$

We can see that under this approximation, $s(Z)$ is approximately a linear function in Z. It is straightforward to see that by plugging in this approximation into the Langevin SDE, we can employ ancestor sampling as in (22) to accelerate the forward diffusion for nonlinear SDEs. In particular, we can easily see that under this linear approximation, the Langevin SDE will reduce to an OU process with certain parameterization:

$$
\begin{aligned}
Z_t &= Z_{t-1} - \eta \nabla_{Z_{t-1}} s(Z_{t-1}) + \sqrt{2\eta}\epsilon_t && (24)\\
&\approx Z_{t-1} - \eta(s(\tilde{Z}) + \nabla_{\tilde{Z}} s(\tilde{Z})(Z_{t-1} - \tilde{Z})) + \sqrt{2\eta}\epsilon_t && (25)\\
&= Z_{t-1} - \eta s(\tilde{Z}) - \eta \nabla_{\tilde{Z}} s(\tilde{Z})(Z_{t-1} - \tilde{Z}) + \sqrt{2\eta}\epsilon_t && (26)\\
&= Z_{t-1} - \eta s(\tilde{Z}) - \eta \nabla_{\tilde{Z}} s(\tilde{Z}) Z_{t-1} + \eta \nabla_{\tilde{Z}} s(\tilde{Z})\tilde{Z} + \sqrt{2\eta}\epsilon_t && (27)\\
&= \left(1 - \eta \nabla_{\tilde{Z}} s(\tilde{Z})\right) Z_{t-1} + \eta \nabla_{\tilde{Z}} s(\tilde{Z}) \left(\tilde{Z} - \left(\nabla_{\tilde{Z}} s(\tilde{Z})\right)^{-1} s(\tilde{Z})\right) + \sqrt{2\eta}\epsilon_t && (28)
\end{aligned}
$$

We can see that this is an OU process with the following parameters:

$$
\begin{aligned}
\theta &= \eta \nabla_{\tilde{Z}} s(\tilde{Z}) && (29)\\
\mu &= \tilde{Z} - \left(\nabla_{\tilde{Z}} s(\tilde{Z})\right)^{-1} s(\tilde{Z}) && (30)\\
\sigma &= 2\eta && (31)
\end{aligned}
$$

This means that we can apply ancestral sampling to the Taylor approximation of our Langevin diffusion. We refer the reader to the pseudocode in Algorithm 1 for our specific implementation.

Algorithm 1 Taylor-Accelerated Forward Sampling

Input: Initial residual Z_0, conditioning variable X, target time $t_\star$, step size η, Taylor steps function $K(t)$
Initialize: Current state $Z_{curr} = Z_0$, current time $t_{curr} = 0$
while $t_{curr} < t_\star$ **do**
  Determine number of Taylor horizon: Set $k = \min(K(t_{curr}), t_\star - t_{curr})$
  Compute Taylor approximation:

$$
\begin{aligned}
s_{curr} &= s_\theta(Z_{curr}; X, t_{curr})\\
\nabla s_{curr} &= \nabla_{Z_{curr}} s_\theta(Z_{curr}; X, t_{curr})
\end{aligned}
$$

  Set OU parameters:

$$
\begin{aligned}
\alpha &= 1 - \eta \nabla s_{curr}\\
\mu_{eff} &= Z_{curr} - \frac{s_{curr}}{\nabla s_{curr}}\\
\sigma_{eff} &= \sqrt{2\eta}
\end{aligned}
$$

  Ancestral sampling: Sample directly at time $t_{curr} + k$: $Z_{curr} \sim \mathcal{N}\left(\alpha^k Z_{curr} + (1-\alpha^k)\mu_{eff}, \sigma_{eff}^2 \frac{1-\alpha^{2k}}{1-\alpha^2}\right)$
  Update time: $t_{curr} \leftarrow t_{curr} + k$
end while
Return: Final residual $Z_{t_\star} = Z_{curr}$

C.3 Pseudocode for Methodology

Algorithm 2 CEVT-based Data Preprocessing

Input: $\{(X_i, Y_i)\}_{i=1}^n$, threshold quantile $\alpha > 0$
Estimate empirical CDFs: Compute $\hat{F}_X$ and $\hat{F}_Y$ from data
Transform to Laplace marginals: For all samples $i = 1, \ldots, n$ **do**

$$X_i^\star \leftarrow -\text{sign}(\hat{F}_X(X_i) - 0.5) \cdot \log\Big(1 - 2|\hat{F}_X(X_i) - 0.5|\Big)$$

$$Y_i^\star \leftarrow -\text{sign}(\hat{F}_Y(Y_i) - 0.5) \cdot \log\Big(1 - 2|\hat{F}_Y(Y_i) - 0.5|\Big)$$

Extract tail samples: Find subset $\{(X_i^\star, Y_i^\star)\}_{i=1}^m$ where $\hat{F}_X(X_i) > 1 - \alpha$
Estimate tail parameters: Compute coefficients a, b using tail samples $\{(X_i^\star, Y_i^\star)\}_{i=1}^m$
Compute residuals: Set $Z_i = \frac{Y_i^\star - a \cdot X_i^\star}{(X_i^\star)^b}$ for $i = 1, \ldots, n$
Return: Preprocessed dataset $\{(X_i^\star, Z_i)\}_{i=1}^n$

Algorithm 3 Diffusion Model Training

Input: Preprocessed dataset $\{(X_i^\star, Z_i)\}_{i=1}^n$, learning rate η, epochs E, weighting function $\lambda(t)$
Initialize: Network parameters θ, noise schedule parameters
for epoch $e = 1, \ldots, E$ **do**
    for batch $\{(X_j^\star, Z_j)\}_{j \in \text{batch}}$ **do**
        Sample timestep: Sample t uniformly over time-horizon.
        Generate noisy samples: Sample $Z_t | Z_0 = Z_j$ according to forward process using Euler-Maruyama solver of Taylor-accelerated sampling in Algorithm 1.
        Compute score matching loss: Evaluate $\mathcal{L}(\theta)$ in (7) for each sample in the batch.
        Backward pass: Compute gradients $\nabla_\theta \mathcal{L}(\theta)$
        Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$
    end for
end for
Return: Trained parameters θ

Algorithm 4 Diffusion Model Sampling

Input: Initial noise Z_T, conditioning $X^\star$, trained score $s_{\theta_\star}$, time horizon T
Initialize: Current state $Z_{curr} = Z_T$, current time $t_{curr} = T$
while $t_{curr} > 0$ **do**
    Determine step size: Set $k = \min(K(t_{curr}), t_{curr})$
    Compute score: Evaluate $s_{\theta_\star}(Z_{curr}; X^\star, t_{curr})$
    Reverse step: Apply reverse SDE or Euler-Maruyama:

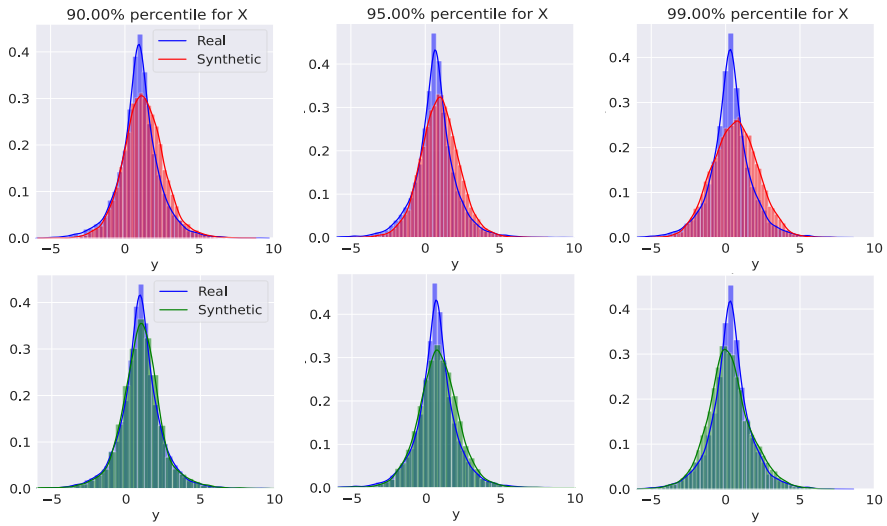
$$Z_{curr} = Z_{curr} + \eta \cdot s_{\theta_\star}(Z_{curr}; X^\star, t_{curr}) + \sqrt{2\eta}\epsilon$$

    where $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    Update time: $t_{curr} \leftarrow t_{curr} - k$
end while
Return: Denoised residual $Z_0 = Z_{curr}$

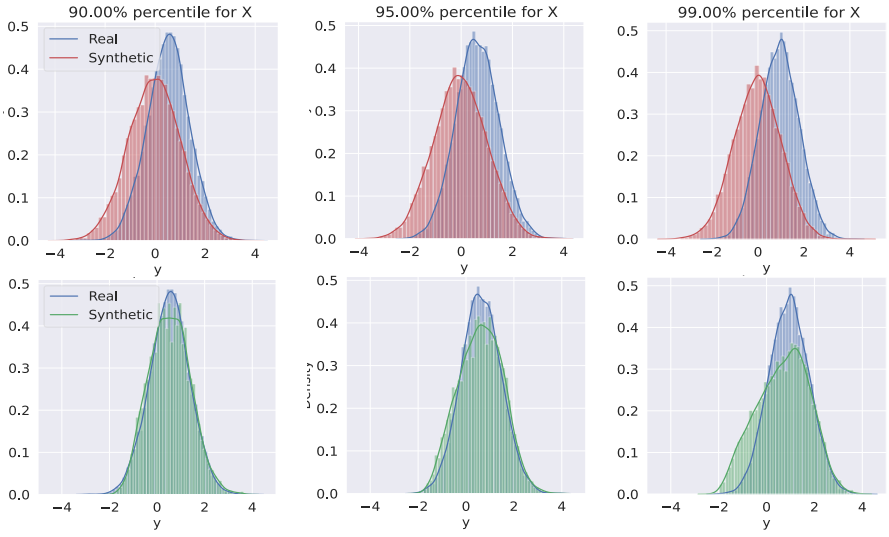
D Additional Experimental Results

Here, we provide additional plots and metrics for the experiments section of our work.

D.1 Synthetic Data



(a) Top Row: Standard method targeting $P(Y|X)$ with linear diffusion. Bottom Row: New method. New method manages to capture the heavy Laplace tails, standard method struggles to do so.



(b) Top Row: Standard method targeting $P(Y|X)$ with linear diffusion. Bottom Row: New method.

Figure 9: (a) Synthetic Example 1 (b) Synthetic Example 2

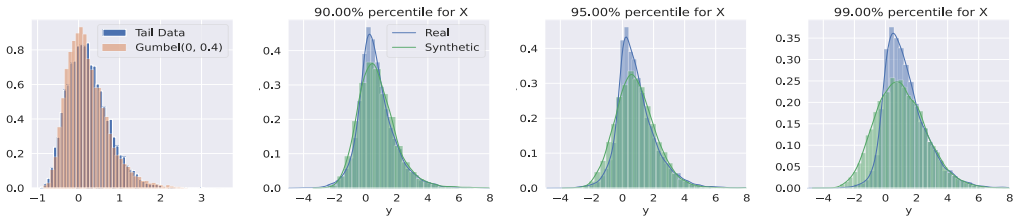


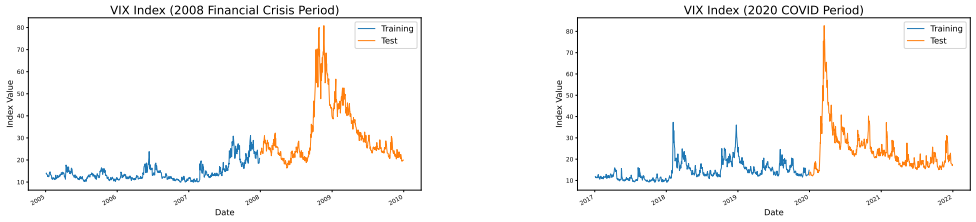
Figure 10: Left Plot: As discussed, we see for extreme (but not infinite) values in the tail, data seem Gumbel distributed. We visualize sampling in the CEVT based representation space ($P(Z|X^\star)$ in the subsequent plots. We capture the one-sided tails.

D.2 Financial Returns Conditioned on VIX

We provide additional and more detailed experimental results for our evaluation on real data.

D.2.1 VIX Time Series

Here, we show a plot of the VIX time series in Figure 11, which serves as the conditional information supplied to the diffusion models for the stock return generation experiment. For both the GFC and COVID periods, the VIX level is relatively lower in the training data (plotted in blue) than the testing data (plotted in orange), indicating that the testing data covers a period of market stress.



(a) GFC (b) COVID

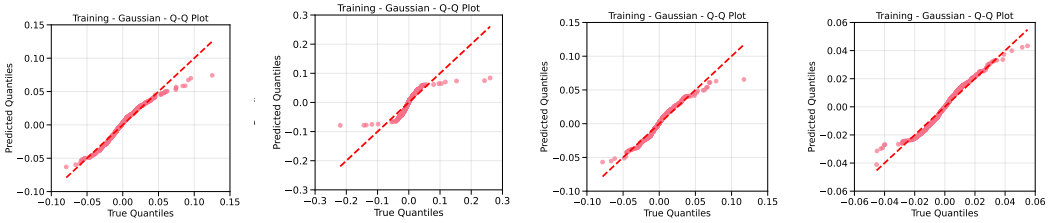
Figure 11: VIX level during the analyzed periods of market stress. VIX level in the training datasets (shown in blue) correspond to more stable market periods, while VIX levels in the testing dataset (shown in orange) correspond to a period of market stress.

D.2.2 Evaluation of Calibration via QQ plots (Unconditional Evaluation)

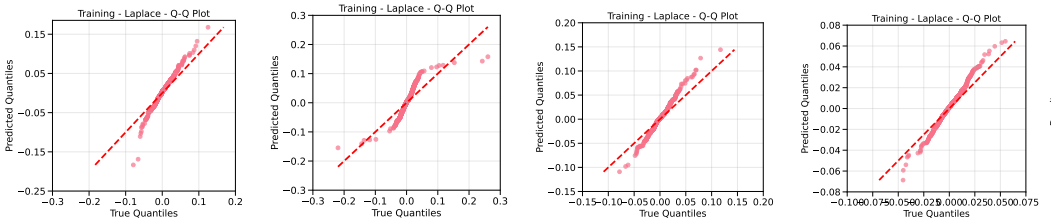
To evaluate the unconditional generative performance (where we marginalize out the conditions) of the proposed conditional diffusion model, we use QQ plots to check for the calibration of the predicted quantiles versus the true quantiles from the empirical dataset. Figure 12 and 13 show the QQ plots for each stock on the training and testing datasets for the GFC period, respectively. Figure 14 and 15 show the QQ plots for each stock on the training and testing datasets for the GFC period, respectively. The results indicate that while the use of a Gaussian base distribution generally leads to better calibration in the training dataset and in the bulk of the distribution (10%-90% quantiles), the use of a Laplace distribution offers a significant advantage in the tail, specifically for the testing datasets, since the testing dataset considers VIX levels (conditions) much larger than what is seen in the training dataset. This showcases the advantages of considering alternative base distributions in the case of generative modeling for heavy-tailed targets.

D.2.3 Scatter Plots of Asset Returns vs. VIX Level (Conditional Evaluation)

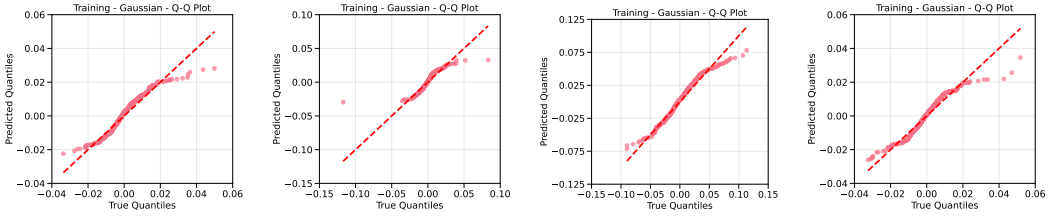
The use of QQ plots makes sense for evaluation of the calibration of the marginal distribution of returns (where we generate samples considering all conditions in the ground truth training and testing datasets); however, it does not provide insight into the performance of the conditional, as we vary the



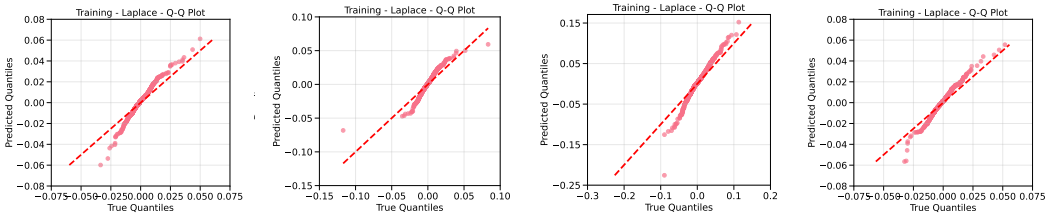
(a) AAPL - Gaussian (b) AMZN - Gaussian (c) GOOGL - Gaussian. (d) GS - Gaussian



(e) AAPL - Laplace (f) AMZN - Laplace (g) GOOGL - Laplace. (h) GS - Laplace

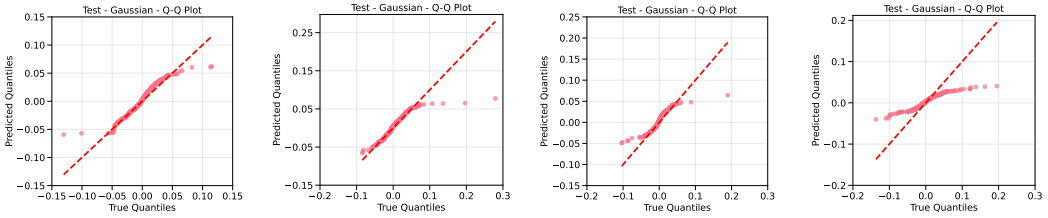


(i) JPM - Gaussian (j) MSFT - Gaussian (k) NVDA - Gaussian. (l) WFC - Gaussian

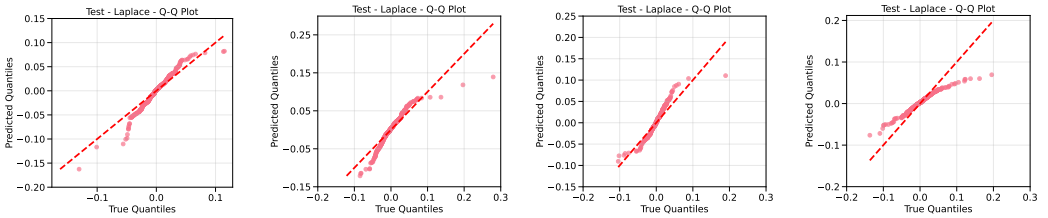


(m) JPM - Laplace (n) MSFT - Laplace (o) NVDA - Laplace. (p) WFC - Laplace

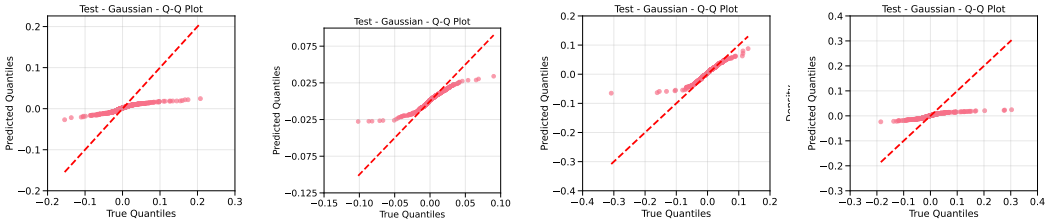
Figure 12: QQ plots on training datasets for the GFC period across all stocks. When comparing the use of a Gaussian base distribution to a Laplace base distribution, we observe that the Gaussian model exhibits superior calibration, particularly in the central mass of the distribution. We hypothesize that this improved fit in the bulk region is attributable to return distributions more closely approximating Gaussian behavior during this period, which coincides with generally lower VIX (volatility index) levels. Another notable observation is that the Laplace base distribution tends to produce overdispersion in the tails, while the Gaussian base leads to underdispersion. This pattern aligns with theoretical expectations, as the Laplace distribution inherently has heavier tails than the Gaussian distribution, making it prone to overestimating tail probabilities when the true data-generating process is closer to Gaussian in nature.



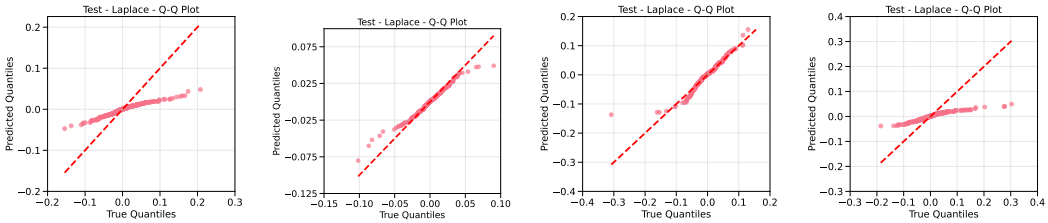
(a) AAPL - Gaussian (b) AMZN - Gaussian (c) GOOGL - Gaussian. (d) GS - Gaussian



(e) AAPL - Laplace (f) AMZN - Laplace (g) GOOGL - Laplace. (h) GS - Laplace

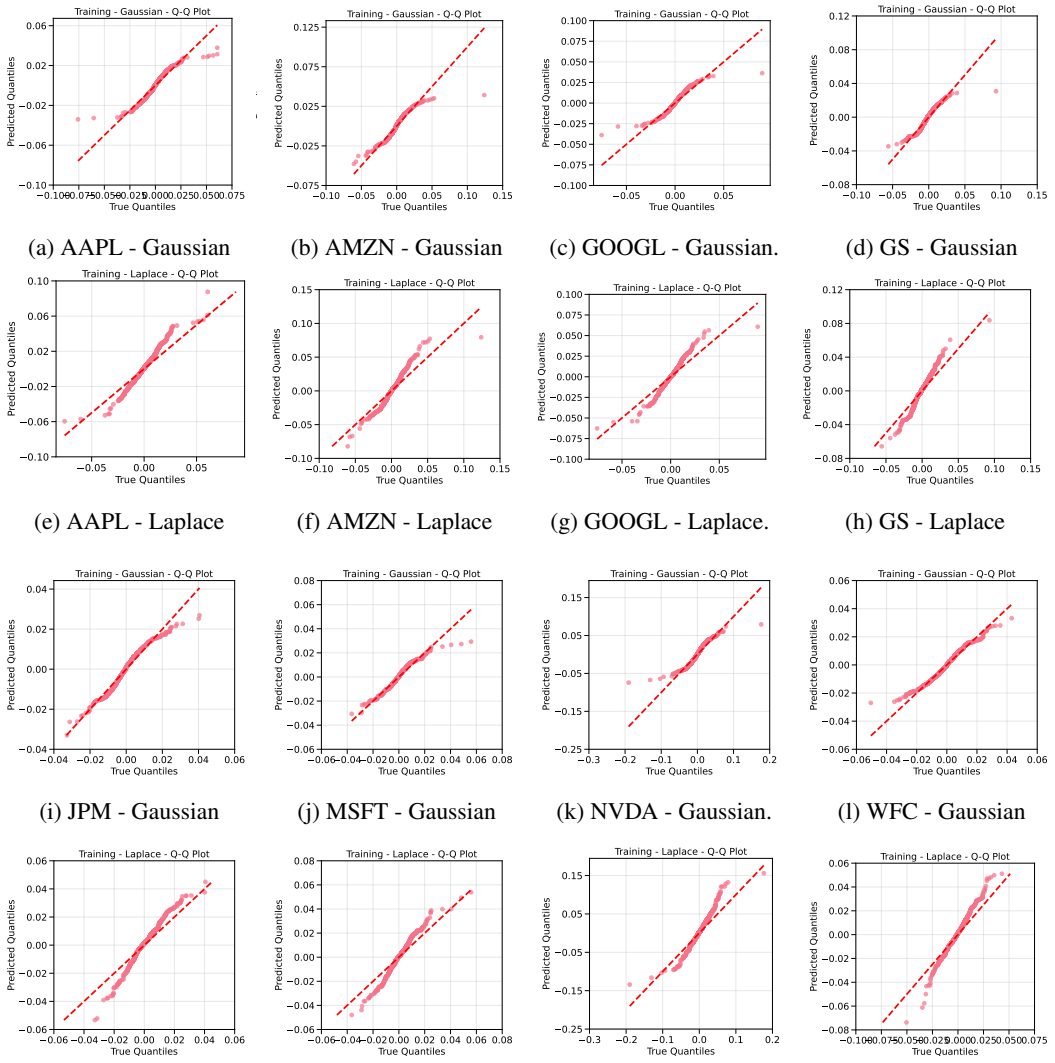


(i) JPM - Gaussian (j) MSFT - Gaussian (k) NVDA - Gaussian. (l) WFC - Gaussian



(m) JPM - Laplace (n) MSFT - Laplace (o) NVDA - Laplace. (p) WFC - Laplace

Figure 13: QQ plots on testing datasets for the GFC period across all stocks. When comparing the use of a Gaussian base distribution to a Laplace base distribution, we observe that the Gaussian model significantly underestimates the tail heaviness of the target distribution (showing extreme underdispersion), while the Laplace distribution provides a much closer approximation to the true tail behavior, particularly in the extreme regions. This pattern is especially pronounced for technology stocks (AAPL, AMZN, GOOGL, NVDA). For financial sector stocks, both distributional models perform inadequately. This can be attributed to the disproportionate impact of the GFC on the financial sector, representing a more comprehensive distribution shift from the training data beyond a covariate shift in market volatility indicators like VIX. Nevertheless, across all stocks, we observe that samples from both base distributions in the conditional generative model exhibit underdispersion relative to the empirical data.



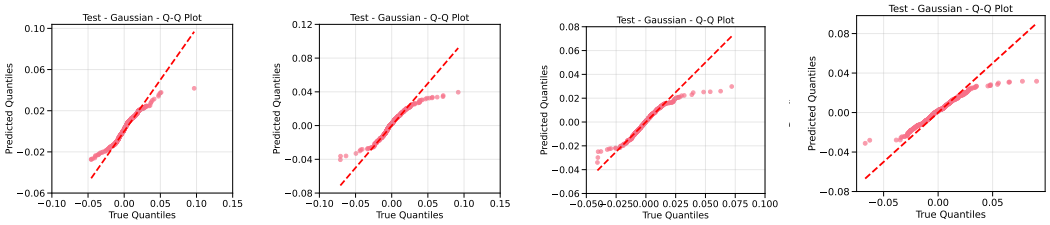
(a) AAPL - Gaussian (b) AMZN - Gaussian (c) GOOGL - Gaussian. (d) GS - Gaussian

(e) AAPL - Laplace (f) AMZN - Laplace (g) GOOGL - Laplace. (h) GS - Laplace

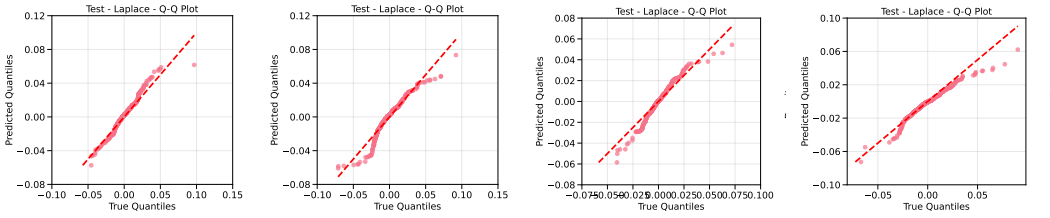
(i) JPM - Gaussian (j) MSFT - Gaussian (k) NVDA - Gaussian. (l) WFC - Gaussian

(m) JPM - Laplace (n) MSFT - Laplace (o) NVDA - Laplace. (p) WFC - Laplace

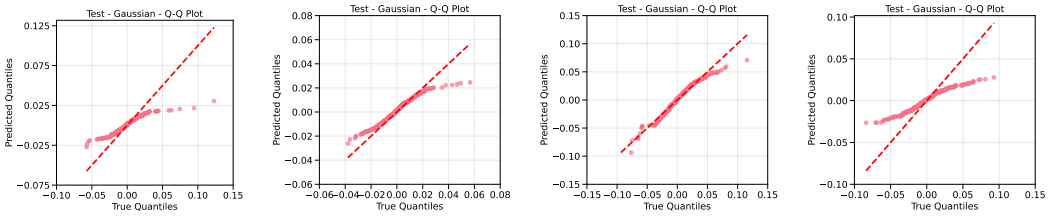
Figure 14: QQ plots on training datasets for the COVID period across all stocks. When comparing the use of a Gaussian base distribution to a Laplace base distribution, we observe that the Gaussian model exhibits better calibration in the bulk of the distribution for most stocks, though with notable deviations in the extremes. Another notable observation is that the Laplace base distribution consistently produces overdispersion in the tails across multiple stocks, while the Gaussian base leads to underdispersion at the extremes; similar to the observation made for the GFC period.



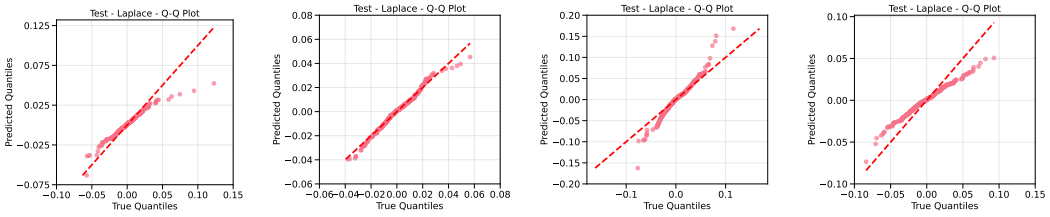
(a) AAPL - Gaussian (b) AMZN - Gaussian (c) GOOGL - Gaussian. (d) GS - Gaussian



(e) AAPL - Laplace (f) AMZN - Laplace (g) GOOGL - Laplace. (h) GS - Laplace



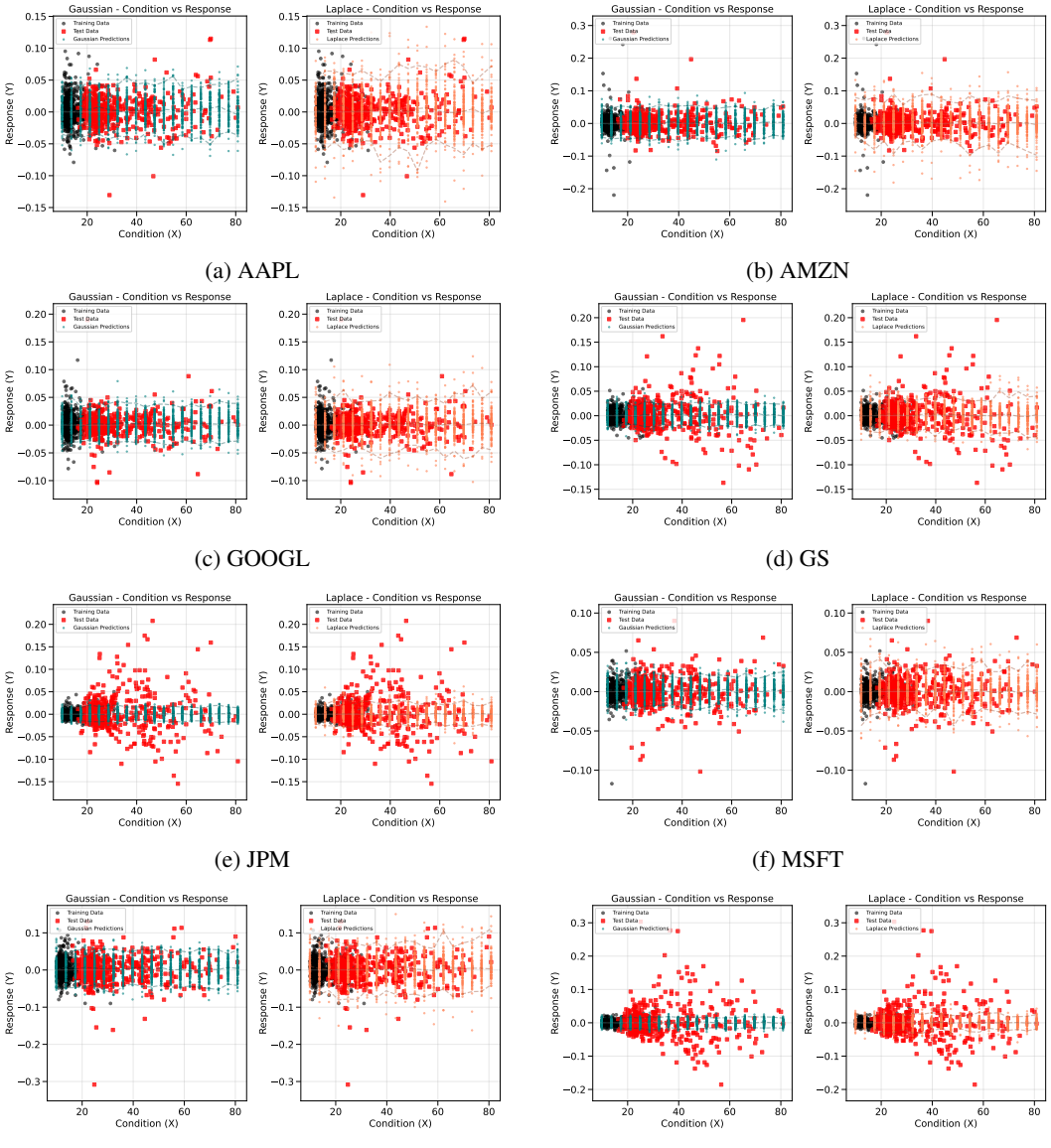
(i) JPM - Gaussian (j) MSFT - Gaussian (k) NVDA - Gaussian. (l) WFC - Gaussian



(m) JPM - Laplace (n) MSFT - Laplace (o) NVDA - Laplace. (p) WFC - Laplace

Figure 15: QQ plots on testing datasets for the COVID period across all stocks. When comparing the use of a Gaussian base distribution to a Laplace base distribution, we observe that the Gaussian model significantly underestimates the tail behavior, particularly evident in technology stocks like JPM (i), MSFT (j), and WFC (l) where predicted quantiles fall below the diagonal reference line at extremes. We hypothesize that this underdispersion reflects the Gaussian distribution's inability to capture the heightened market volatility characteristic of the COVID crisis period. Another notable observation is that the Laplace base distribution provides a markedly better fit to the tail behavior for most stocks, especially visible in AAPL (e), GOOGL (g), and JPM (m), though it still exhibits some deviations from perfect calibration. This pattern aligns with theoretical expectations, as the COVID period featured extreme market movements that are better approximated by distributions with heavier tails, making the Laplace distribution's inherent properties more suitable for modeling the fat-tailed nature of returns during this market stress periods.

condition to extreme values. In the case of VIX, we are interested in the right-tail of the condition; when the VIX level grows to large positive values (around 40-80). To evaluate the conditional performance, we use a scatter plot of the returns and the VIX level, and compare that the empirical percentiles of the conditional diffusion model for both the Gaussian and Laplace base distributions. We show these scatter plots for each ticker in Figure 16 and 17 for the GFC and COVID periods, respectively. For both periods, we can observe that the use of Gaussian base leads to underestimation of the tails across almost all conditions, while the use of a Laplace is much closer.



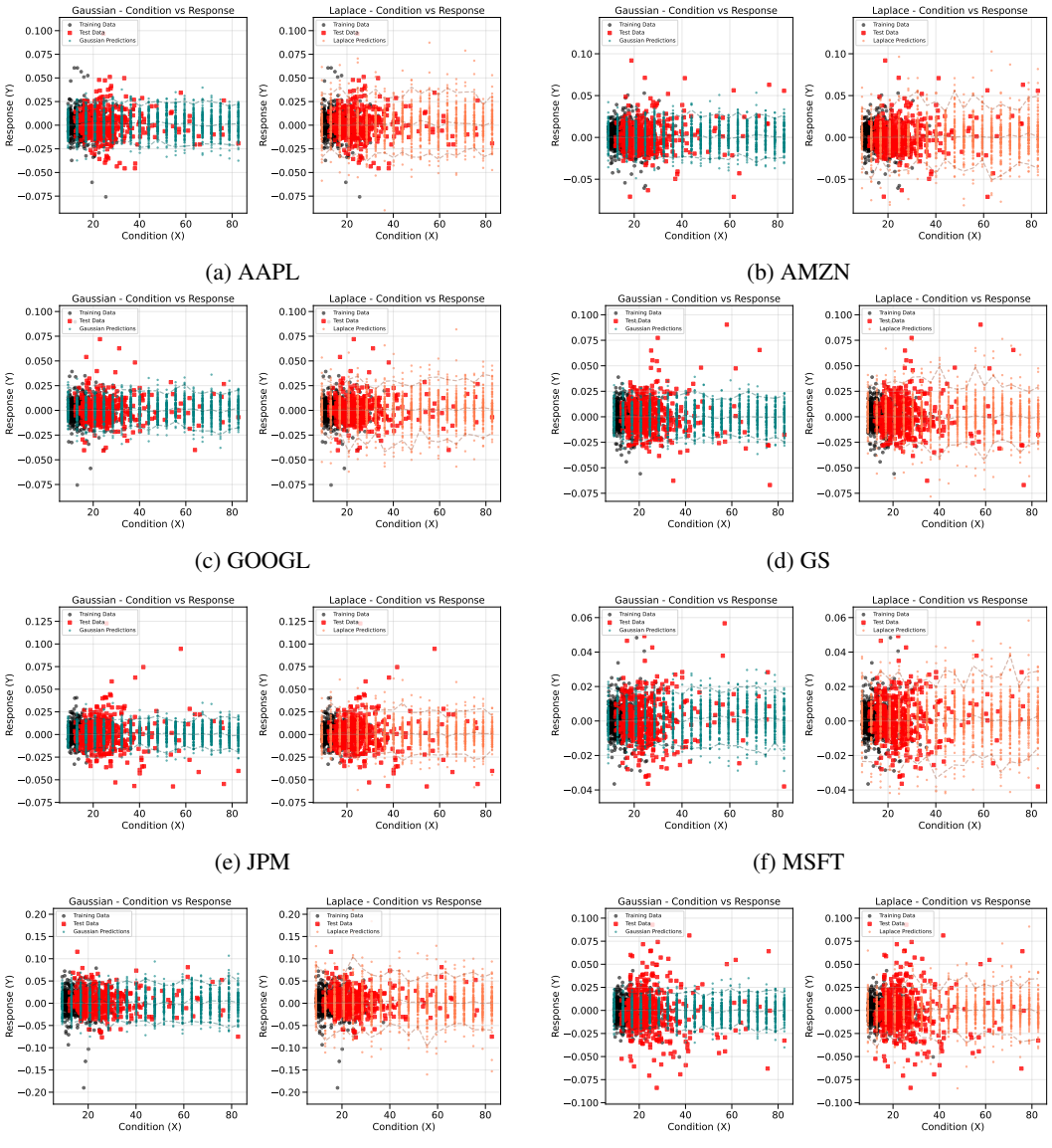
(a) AAPL (b) AMZN

(c) GOOGL (d) GS

(e) JPM (f) MSFT

(g) NVDA (h) WFC

Figure 16: Scatter plots for visualization of conditional generation performance for GFC period.



(a) AAPL (b) AMZN

(c) GOOGL (d) GS

(e) JPM (f) MSFT

(g) NVDA (h) WFC

Figure 17: Scatter plots for visualization of conditional generation performance for COVID period.