Notation	Meaning
$k(x, x')$	kernel (covariance) function evaluated at x and x'
$\mathcal{GP}$	Gaussian process
Z	inducing-point locations
$\mathbb{X}$	second order process w.r.t X
$\mathbf{X}$	A rough path as a pair $(X, \mathbb{X})$
$\|\cdot\|_\alpha$	α-Hölder continuous norm
$\|\cdot\|_{p;[0,T]}$	p-varation norm over $[0, T]$
$\varrho_\alpha(\cdot, \cdot)$	α-Hölder metric distance

Table 1: List of notations

Pseudo-code for our approach solving SDEs Here, we provide our approach in Algorithm 1 in contrast to the SDEs with Brownian motion in Algorithm 2.

Algorithm 1: Our approach solving SDE for mBm

Input: Drift function $\mu(t, x)$, diffusion function $\sigma(t, x)$, initial value X_{t_0}, Hurst function $h(\cdot)$ inducing-point locations Z
Output: Solution X_{t_1} at t_1

`/* Compute drift and diffusion in Eq. (12) */`
def `f_and_g(`t, x`):`
  `/* Python syntax: global */`
  global inducing points u
  `/* Mean and variance of GP approximation in Eq. (3) */`
  $m(t; u), v(t; u)$ = `GP_approx(`t, u, $h(\cdot)$`)`
  $\nu(t; u) = m(t; u)$
  $\varsigma(t; u) = v(t; u)\Delta t$
  drift = $\mu(t, x) + \sigma(t, x)\nu(t; u)$
  diffusion = $\sigma(t, x)\varsigma(t; u)$
  return drift, diffusion

def `brownian_increment(`t`):`
  return Brownian noise ΔW_t

`/* Use Eq. (9) and (10) */`
inducingpoints u = `sample_inducing(`Z, $h(\cdot)$`)`
`/* call SDE solver */`
X_{t_1} = `sdeint(f_and_g, brownian_increment,` X_{t_0}, t_0, t_1`)`
return X_{t_1}

Algorithm 2: SDE solver for Brownian cases

Input: Drift function $\mu(t, x)$, diffusion function $\sigma(t, x)$, initial value X_{t_0}
Output: Solution X_{t_1} at t_1

`/* Compute drift and diffusion directly */`
def `f_and_g(`t, x`):`
  return $\mu(t, x)$, $\sigma(t, x)$

def `brownian_increment(`t`):`
  return Brownian noise ΔW_t

`/* call SDE solver */`
X_{t_1} = `sdeint(f_and_g, brownian_increment,` X_{t_0}, t_0, t_1`)`
return X_{t_1}

Model training with Latent SDE The training method used in experiments of § 5.2 and § 5.3 is adopted from [54] via a Bayesian approximation method. Consider two SDEs sharing the same

diffusion function

$$d\tilde{X}_t = \mu_\theta(t, \tilde{X}_t) + \sigma(t, \tilde{X}_t)dB_t, \qquad \text{(Prior SDE)}$$
$$dX_t = \mu_\phi(t, X_t) + \sigma(t, X_t)dB_t. \qquad \text{(Approximate posterior SDE)}$$

Applying our approach like in (12), we can obtain new drift and diffusion functions (with conditioned on inducing points) for SDEs with Brownian motion. Let us write in this form

$$d\tilde{X}_t = \bar{\mu}_\theta(t, \tilde{X}_t) + \bar{\sigma}(t, \tilde{X}_t)dW_t, \qquad \text{(Prior SDE)}$$
$$dX_t = \bar{\mu}_\phi(t, X_t) + \bar{\sigma}(t, X_t)dW_t, \qquad \text{(Approximate posterior SDE)}$$

where $\bar{\mu}_\theta, \bar{\mu}_\phi, \bar{\sigma}$ are derived similar to (12).

The variational bound requires computing the KL divergence between the approximate posterior and prior which is related to the *change of measures* between prior paths and posterior paths, therefore, is obtained by Girsanov Theorem. That is, with a well-defined $u(t, x)$ satisfying $\bar{\sigma}(t, x)u(t, x) = \bar{\mu}_\phi(t, x) - \bar{\mu}_\theta(t, x)$, the change of measure is written as

$$M_t = \exp\left(-\frac{1}{2}\int_0^t |u(s, X_s)|^2 ds - \int_0^t u(s, X_s)dW_s\right).$$

Finally, the variational bound consists of the log-likelihood and $\mathbb{E}[\log M_t]$. Details of the learning algorithm can be found in [54].

A Fractional Brownian Motion and Multifractional Brownian Motion

This section contains additional background of fractional Brownian motions (fBm) and multifractional Brownian motions (mBm) including their covariance function and a detail description of long-range dependency in such processes.

A.1 Fractional Brownian Motion

Fractional Brownian motion can be viewed as a Gaussian process with covariance

$$k_H(t, s) = \frac{1}{2}(|t|^{2H} + |s|^{2H} - |t - s|^{2H}). \tag{15}$$

Here, H is called the Hurst exponent. It has some properties: (1) long-range dependence ($H > 1/2$); (2) anti-persistence or irregularity ($H < 1/2$). Such properties are interesting to applications in modeling internet traffic, highly textured images.

A.2 Multifractional Brownian Motion

Multifractional Brownian motion $B^{(h)}$ is a Gaussian process with covariance [52]

$$k_h(t, s) = \frac{c^2_{h_{t,s}}}{c_{h(t)}c_{h(s)}}(|t|^{2h_{t,s}} + |s|^{2h_{t,s}} - |t - s|^{2h_{t,s}}), \tag{16}$$

with $h : [0, T] \to (0, 1)$, $h_{t,s} := \frac{h(t)+h(s)}{2}$, and $c_x = \left(\frac{2\pi}{\Gamma(2x+1)\sin(\pi x)}\right)^{\frac{1}{2}}$. This extension is to model cases that behaviors of signals happen locally. If $h(t)$ is close to 0, we expect irregular local behaviors. If $h(t) > 1/2$ at time t, we expect local long-range dependence.

A.3 Itô Integral for Multifractional Brownian Motion

Traditional Brownian motion case The Itô integral is computed as

$$f(T, B_T) = f(0, 0) + \int_0^T \frac{\partial f}{\partial t}(t, B_t)dt + \int_0^T \frac{\partial f}{\partial x}(t, B_t)dB_t + \frac{1}{2}\int_0^T \frac{\partial^2 f}{\partial x^2}(t, B_t)dt$$

Multifractional Brownian motion case The Itô integral has additional derivative of kernel function

$$f(T, B^{(h)}_T) = f(0, 0) + \int_0^T \frac{\partial f}{\partial t}(t, B^{(h)}_t)dt + \int_0^T \frac{\partial f}{\partial x}(t, B^{(h)}_t)dB^{(h)}_t + \frac{1}{2}\int_0^T \left(\frac{d}{dt}R_h(t, t)\right)\frac{\partial^2 f}{\partial x^2}(t, B^{(h)}_t)dt.$$

The red term $\frac{d}{dt}R_h(t, t)$ is the derivative of variance function of mBm. This can be interpreted as the second derivative term in Itô's formula, $\frac{\partial^2 f}{\partial x^2}(t, B^{(h)}_t)$, is adjusted according to how fast the variance function changes when time goes. Note that $\frac{d}{dt}R_h(t, t) = 2t^{2h(t)-1}(h'(t)t\ln(t) + h(t))$ [52]. This result is used in the main text to get the analytic solution of $dX_t = \alpha X_t dt + \beta dB_t$

A.4 Properties of Fractional Brownian Motion

As discussed in the main text, fBm exhibits distinct properties including self-similarity and long-range dependency. We provide a glimpse of these notions to readers.

Self-similarity Simply speaking, self-similarity in general refers to objects which look exact or approximately similar to a part of itself. This concept is related to fractals. In statistics, consider a discrete stochastic process $\{X_t\}_{t\in\mathbb{N}}$. We say X is self-similar if a finite sum, i.e., $X_{km}+\cdots+X_{(k+1)m-1}$ and a scaled version $a_m X_k$ have the same distribution. In a restricted definition when a_m is just a polynomial of m, we say $X_k^{(m)}=\frac{1}{m}(X_{km}+\cdots+X_{(k+1)m-1})$ and $m^{1-H}X^{(m)}$ are distributionally equal or

$$(X_{k_1},\ldots,X_{k_d}) \quad \text{distributionally equal to} \quad (m^{1-H}X_{k_1}^{(m)},\ldots,m^{1-H}X_{k_d}^{(m)})$$

if X is self-similar.

Now, consider the noise with respect to fBm, defined as

$$X_k = B^H(k+1)-B^H(k).$$

The self-similarity of X is expressed by comparing $mX^{(m)}$ and $m^H X$. Both of them are zero-mean Gaussian, having the same covariance because

$$\begin{aligned}
&\mathrm{Cov}(X_{km}+\cdots+X_{(k+1)m-1}, X_{lm}+\cdots+X_{(l+1)m-1})\\
=&\,\mathrm{Cov}(B^H((k+1)m)-B^H(km), B^H((l+1)m)-B^H(lm))\\
=&\,\mathrm{Cov}(m^H B^H(k+1)-m^H B^H(k), m^H B^H(l+1)-m^H B^H(l))\\
=&\,\mathrm{Cov}(m^H X_k, m^H X_l).
\end{aligned}$$

Long-range dependence Long-range dependence in stochastic processes means the sum of the autocovariance function is unbounded. For fractional Gaussian noise, $X_k=B_{k+1}^H-B_k^H$ has the autocovariance function

$$\gamma(k)=\frac{1}{2}(|k-1|^{2H}-2|k|^{2H}+|k+1|^{2H}).$$

This can be approximated as $\gamma(k)\sim H(2H-1)k^{2H-2}$, leading $\sum_{k=1}^{\infty}\gamma(k)=\infty$ when $1/2<H<1$.

Perhaps, fractional ARIMA is a more intuitive model to understand long-range dependency. While well-known ARIMA model is defined as

$$\phi(L)W_k=\theta(L)\epsilon_k$$

where L is the lag operator as $LW_k=W_{k-1}$, ϕ and θ are polynomials. By introducing

$$(1-L)^d W_k=\sum_{n=0}^{\infty}\binom{d}{n}(-L)^n W_k, \quad \binom{d}{n}=\frac{\Gamma(d+1)}{\Gamma(d-n+1)\Gamma(n+1)},$$

we can construct fractional ARIMA model. The long-range dependence is expressed by the infinite sum in the right hand side. Here, the Hurst parameter is $H=d+1/2$.

B Convergence analysis

In this section, we provide some background on rough path theory. Our proof will investigate the *variation* of covariance of Gaussian processes and sparse Gaussian processes which help us to judge their Hölder continuity. Then, we use results of rough path theory [56, 26] to bound the KL divergence between solutions via the KL divergence fractional Brownian motions and their sparse GP approximations. Finally, we use [8] to justify the convergence of sparse GPs.

B.1 Rough path theory

Here we give a brief review of rough path theory. For a complete introduction, readers can refer to [26].

Notation We denote $X_{s,t} := X_t - X_s$ and $\int_s^t X_{s,r} dX_r := \mathbb{X}_{s,t}$. Also, p-variation seminorm is defined as

$$\|X\|_{p-\text{var};[0,T]} = \left(\sup_{\mathcal{P}} \sum_{[s,t]\in\mathcal{P}} |X_{s,t}|^p \right)^{1/p},$$

where $\mathcal{P}$ is a set of disjoint intervals of some partitioning scheme over $[0,T]$. We will write $x \lesssim y$ to express $x \leq Cy$ for $x, y > 0$.

In this section, we will sometimes write the covariance as $R(s,t)$. A *rectangular increment* of covariance is denoted by

$$R\begin{pmatrix} s,t \\ s',t' \end{pmatrix} := \mathbb{E}[X_{s,t} X_{s',t'}]. \tag{17}$$

The p-variation of a rectangle increment is written as $\|R\|_{p;[0,T^2}$.

The space of rough paths The following will provide the specification of the space of rough paths including its norm and its metric.

Definition 2. Given $\alpha \in \left(\frac{1}{3}, \frac{1}{2}\right]$, the space of α-Hölder rough paths is defined by

- pairs $(X, \mathbb{X}) =: \mathbf{X}$
- norm
$$\|X\|_\alpha = \sup_{s\neq t\in[0,T]} \frac{|X_{s,t}|}{|t-s|^\alpha} < \infty, \quad \|\mathbb{X}\|_{2\alpha} = \sup_{s\neq t\in[0,T]} \frac{|\mathbb{X}_{s,t}|}{|t-s|^{2\alpha}} < \infty$$
- For any $\mathbf{X}$ and $\mathbf{Y}$, (inhomogeneous) α-Hölder rough path metrics is
$$\varrho_\alpha(\mathbf{X}, \mathbf{Y}) := \|X - Y\|_\alpha + \|\mathbb{X} - \mathbb{Y}\|_{2\alpha}.$$

The space of geometric rough paths Many results in rough path theory require rough paths being *geometric* where the following holds

$$\text{Sym}(\mathbb{X}_{s,t}) = \frac{1}{2}(X_t - X_s)^2.$$

For Brownian motions, Stratonovich integration leads to geometric Brownian rough paths. There is a way to convert between Itô and Stratonovich version of Brownian motions (see [39, §3]). A similar technique can be achieved for fBm by [70].

Itô-Lyons map One of interesting results in rough path theory is the Itô-Lyons maps. Suppose we have a solution map

$$B(\omega) \overset{\Psi}{\mapsto} (B, \mathbb{B})(\omega) \overset{S}{\mapsto} X(\omega).$$

Simply speaking, while Ψ is universal as it is just an enhancement of B, the solution map S is a *continuous* map on both the initial conditions and driving noise. This property does not hold for Itô maps from sole Brownian motion paths to solution paths.

B.2 Main proof

Before going to the main proof, we provide the Gaussian process form of sample paths $\widehat{B}_t$ corresponding to B'_t described in (9) and (10). We can understand $\widehat{B}_t$ is the result of applying integration operator on B_t (with small numerical error). Since integration operation is linear, $\widehat{B}_t$ is also a Gaussian process:

$$\widehat{B}_t \sim \mathcal{GP}\left(0, k_{\widehat{B}}(t,s)\right), \quad k_{\widehat{B}}(t,s) = \int_0^t \int_0^s k(u,v) - \alpha(u)^\top k(Z,Z)\alpha(v) du dv. \tag{18}$$

In fact, one can recover $k_H(t,s)$ in (15) which is $\int_0^t \int_0^s k(u,v) du dv = k_H(t,s)$.

For relevant results of rough path theory in the context that driving signals are Gaussian processes, we encourage readers to see [26, Chapter 10] for a complete description.

Proof of Theorem 1. Since $\mathbb{E}[(\widehat{B}_t - \widehat{B}_{t+\tau})^2] \leq L|\tau|^{1/\varrho}$, by [26, Theorem 10.9], the covariance of $\widehat{B}_t$ has finite ϱ-variation which is

$$\left\|R_{\widehat{B}}\right\|_{\varrho;[s,t]^2} \leq M|t-s|^{1/\varrho}.$$

As B_t is a fractional Brownian motion with Hurst exponent H, then the covariance of B_t has finite $\bar{\varrho}$-variation with $\bar{\varrho} = \frac{1}{2H}$ (see [26, Example 10.11]), or,

$$\|R_B\|_{\bar{\varrho};[s,t]^2} \leq M|t-s|^{1/\bar{\varrho}}.$$

We consider $R_{(B,\widehat{B})}(u,v) := \mathbb{E}[B_u \widehat{B}_v]$. Using Cauchy-Schwartz inequality, we have $\mathbb{E}[B_u\widehat{B}_v] \leq \sqrt{\mathbb{E}[B_u^2]}\sqrt{\mathbb{E}[\widehat{B}_v^2]}$. Since $\varrho > \bar{\varrho}$, we use a similar technique in the proof of [26, Corollary 10.6]

$$\left\|R_{(B,\bar{B})}\right\|_{\varrho;[s,t]^2} \leq M|t-s|^{1/\varrho}.$$

Now applying the result in [26, Corollary 10.6], for every $\alpha \in (1/3, 1/2\varrho)$ and every $\theta \in (0, 1/2 - \varrho\alpha)$ and $q < \infty$

$$|\varrho_\alpha(\mathbf{B}, \widehat{\mathbf{B}})|_{L^q} \lesssim \sup_{s,t\in[0,T]} \left[\mathbb{E}[|B_{s,t} - \widehat{B}_{s,t}|^2]\right]^\theta \lesssim \sup_{t\in[0,T]} \left[\mathbb{E}[|B_t - \widehat{B}_t|^2]\right]^\theta. \tag{19}$$

This implies $|\varrho_\alpha(\mathbf{B}, \widehat{\mathbf{B}})|_{L^q} \lesssim \sup_{t\in[0,T]} \mathcal{W}(B_t, \widehat{B}_t)^{2\theta}$. Using Talagrand inequality [81], we have

$$|\varrho_\alpha(\mathbf{B}, \widehat{\mathbf{B}})|_{L^q} \lesssim \sup_{t\in[0,T]} \mathrm{KL}(B_t||\widehat{B}_t)^\theta. \tag{20}$$

We follow closely the proof of [39]. That is, following [26, Theorem 8.15], the solution map $S_t(\mathbf{B}, \cdot)$ is a $\mathcal{C}^1$-diffeomorphism, i.e., its bijective and its inversion are differentiable. This means we can obtain the solution X_t at time t via canonical lift $\mathbf{B}$ and initial state X_0. Let us define the inverse map as $S_{-t}(\mathbf{B}, \cdot)$ which gives us a way to get X_0 given X_t.

One can obtain the probability density distribution of solution at time t by a change of variables:

$$p_t(x) = p_0(S_{-t}(\mathbf{B}, x))|\det \nabla_x S_{-t}(\mathbf{B}, x)|, \tag{21}$$

$$\widehat{p}_t(x) = p_0(S_{-t}(\widehat{\mathbf{B}}, x))|\det \nabla_x S_{-t}(\widehat{\mathbf{B}}, x)|. \tag{22}$$

Then continuity of S_t or S_{-t} is stated in [26, Theorem 8.5]. That is, given rough paths, $\mathbf{X}$, $\mathbf{Y}$, and initial state x, y, we can obtain a local Lipschitz estimate, for any $\gamma \in [1/3, \alpha]$

$$\|S(\mathbf{X}, x) - S(\mathbf{Y}, y))\|_\gamma \leq C_\gamma(\|x-y\| + \varrho_\alpha(\mathbf{X}, \mathbf{Y})).$$

Combining the continuity of p_0 and S_t, one can arrive with

$$|\log \widehat{p}_t(x) - \log p_t(x)| \lesssim \varrho_\alpha(\mathbf{B}, \widehat{\mathbf{B}}). \tag{23}$$

It is easy to show that from (23), one can have $\mathrm{KL}(\widehat{p}_t\|p_t) \lesssim |\varrho_\alpha(\mathbf{B}, \widehat{\mathbf{B}})|_{L^1}$. Combining with (20), $\mathrm{KL}(\widehat{p}_t\|p_t) \lesssim \sup_{t\in[0,T]} \mathrm{KL}(\widehat{B}_t\|B)^\theta$. □

Discussion In Theorem 1, we make an assumption that ϱ is greater than $1/2H$ because of the following observations. Compared to the covariance of B_t, that of $\widehat{B}_t$ in (18) is subtracted with a positive term. Therefore, it is reasonable to assume that $\mathbb{E}[(\widehat{B}_t - \widehat{B}_{t+\tau})^2]$ is smaller than $\mathbb{E}[(B_t - B_{t+\tau})^2] = |\tau|^{2H}$. This also means than we can find ϱ such that $\mathbb{E}[(\widehat{B}_t - \widehat{B}_{t+\tau})^2] \leq L|\tau|^{1/\varrho} < |\tau|^{2H}$. In other words, ϱ is greater than $1/2H$. As $\widehat{B}_t$ tries to approximate B_t, we believe ϱ is *slightly* greater than $1/2H$. Also, we can understand that $\widehat{B}_t$ comes from the covariance with truncated spectral information, therefore it is less varying and ϱ tends to bigger.

B.3 Spectral properties of fBm covariance matrix

We empirically compute the eigenvalues of fBm covariance matrices when $H = 0.3, 0.5, 0.8$. The covariance matrices are computed for finite data points in range $[0, 2]$. In Figure 8, we can see that the eigenvalues rapidly decay.

C Experiment results

C.1 Set up

We run experiment on a server equipped with Nvidia Tesla P40 GPU. The number of inducing points is 70. In practice, this parameter is less sensitive in training GP when the number of data points (steps) are not too big.

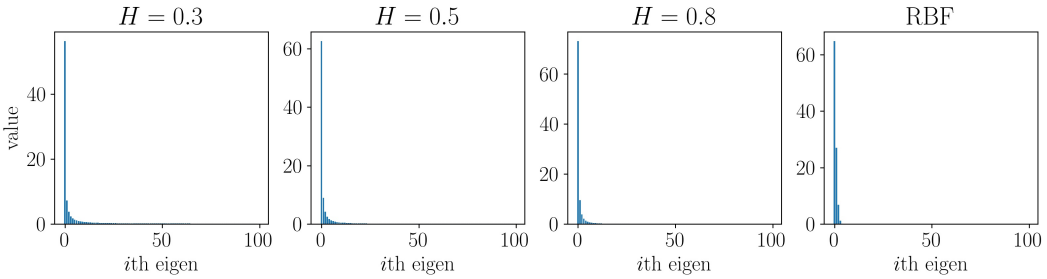


Figure 8: Eigenvalues of covariance matrices. The last column is the plot of eigenvalue of squared exponential function which is one of the studied kernels in [8].

C.2 Sampling mBm

In the experiment in §5.1, the baseline for sampling mBm is based on [65].

To sample mBm given a Hurst function $h(t)$, [65] uses the integral approximation over a discretization of time $t_0 < t_1 < \cdots < t_N$,

$$B_t = \frac{1}{\Gamma(h(t) + 1/2)} \int_0^t (t-s)^{h(t)-1/2} dW_s \approx \frac{1}{\Gamma(h(t) + 1/2)} \sum_{i=0}^{N-1} \int_{t_i}^{t_{i+1}} (t_{i+1} - s)^{h(t_{i+1})+1/2} dW_s.$$

Each component in the sum can be expressed analytically, we have the mBm samples over the time mesh as

$$B_{t_i} = \sum_{j=1}^{i} w_{i-j+1} \xi_j, \tag{24}$$

$$w_k = \frac{1}{\Gamma(h(t_k) + 1/2)} \left[\frac{t_k^{2h(t_k)} - (t_{k-1})^{2h(t_k)}}{2h(t_k)} \right]^{1/2}, \quad k \geq 1. \tag{25}$$

Here, ξ_j are standard Gaussian random variables. Note that our procedure sampling paths for B_t needs only one Gaussian random variable at a time step (see Eq. (11)) while using [65] requires all Gaussian random variables in the sum in Eq. (24). The number of such variables increases as we increase the mesh size. In the experiment, we used the implementation in [25] to get mBm samples.

Estimating Hurst from data Here, we provide a brief description of how to estimate Hurst from data according to [27]. Motivating from the following property

$$\mathbb{E}[|X_t - X_s|^k] = \frac{2^{k/2}\Gamma(\frac{k+1}{2})}{\Gamma(\frac{1}{2})} \sigma^k |t-s|^{kh(t)}, \tag{26}$$

the estimation is based on the ratio of two statistics

$$M_k(t) = \frac{1}{W} \sum_{i=0}^{W-1} |X_{t-i/N} - X_{t-(i+1)/N}|^k, \tag{27}$$

$$M'_k(t) = \frac{2}{W} \sum_{i=0}^{W/2-1} |X_{t-2i/N} - X_{t-2(i+1)/N}|^k, \tag{28}$$

where W is a window size. The second statistic has a halved resolution. From [27], the estimation $\hat{h}(t) = \frac{1}{2} \log_2 \left(\frac{M'_2(t)}{M_2(t)} \right)$ as the coefficient in Eq. (26) vanishes, remaining Hurst exponent values. This converges a.s. to $h(t)$.

In our experiments, we set the windows size, $W = 10$.

C.3 Synthetic data

Figure 9 shows the log-likelihood of the model when the number of data points $N = 100, 200, 300, 400$. As all the settings have similar average log-likelihoods, the model can fit well data in such settings.

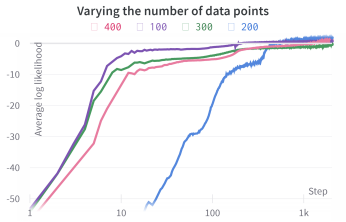
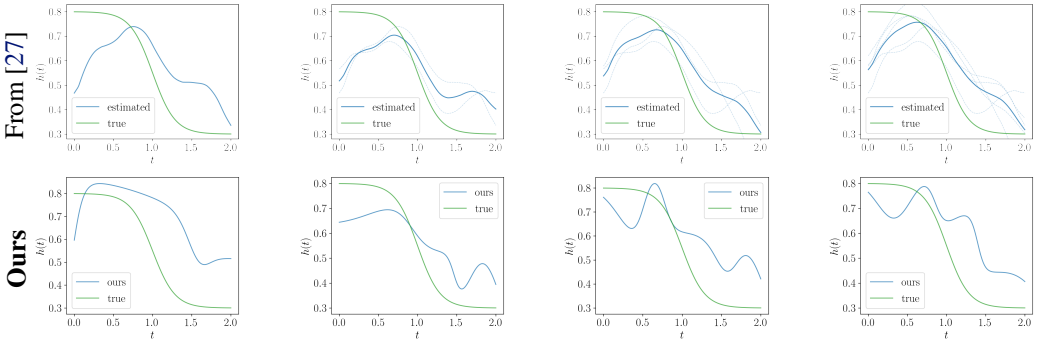


Figure 9: Log likelihood when varying the number of data points in the experiment with synthetic data



(a) 1 path (b) 2 paths (c) 3 paths (d) 5 paths

Figure 10: Learned Hurst exponent vs. true Hurst exponent $h(t) = 0.3 + 0.5\texttt{sigmoid}(7(1-t))$. The first row contains the Hurst estimations computed from [27] (the dashed lines is the estimation for each sample while the solid line is the mean). The second row contain our learned Hurst functions. Our Hurst exponent gets closer to the true one as the sample paths in the training are added. Our approach gives better estimations of Hurst for high values, while slightly overestimates those with small values.

Additional experiments We consider the true Hurst functions including $h(t) = 0.3 + 0.5\texttt{sigmoid}(7(1-t))$ and $h(t) = 0.3 + 0.5\texttt{sigmoid}(7(t-1))$.

Figures 10 and 11 show the comparisons between our learned Hurst functions and the true one when we increase the number sample paths in the training. The more sample paths are added, the closer our learned Hurst functions are to the true one. This observation is similar to the results of [27] of which the plots are placed in the first row of Figure 10 and 11.

We observe that choosing small step sizes helps training better, especially for the case $H < 1/2$.

C.4 Financial data

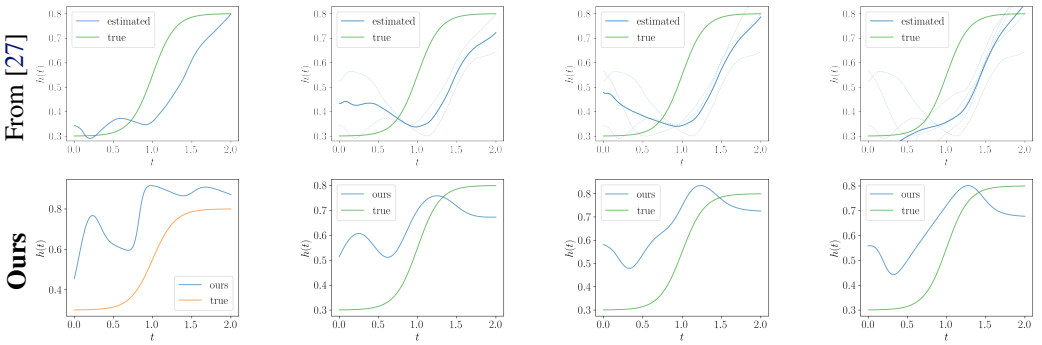
Figure 12 contains the negative log-likelihoods of stock data on the training data. Figures 13 and 14 show the negative log-likelihoods and root mean square errors on the test data. The posterior plot of these stocks are in Figure 16(APPL), 15(AMZN), 17(GOOGL), and 18(MSFT).

C.5 Hurst exponent in score-based generative models

Recent work [77] on generative model based on score functions and SDEs presents impressive results on par with generative adversarial networks (GAN). The main idea is to learn a score function $\nabla \log p(x)$ of data. which is the gradient of log likelihood of data via a process of gradually adding perturbation to data up to the point the result is unrecognizable compared to the original data, similar to random noise. This process models by a SDE

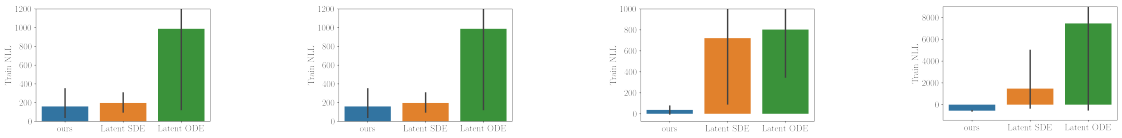
$$dX_t = -\frac{1}{2}\beta(t)X_t dt + \sqrt{\beta(t)}dW_t.$$

As perturbing data along the time interval, it is possible to compute analytically the distribution of perturbed data X_t at time t given input data X_0, $p(X_t|X_0)$. During the sampling phase, such information and the input data X_0 will both not be available. [77] uses a neural network $s(X_t)$ to approximate $\nabla \log p(X_t|X_0)$.



(a) 1 path (b) 2 paths (c) 3 paths (d) 5 paths

Figure 11: Learned Hurst exponent vs. true Hurst exponent $h(t) = 0.3 + 0.5\texttt{sigmoid}(7(t-1))$ (see more description in Figure 11). Our Hurst exponent gets closer to the true one as the sample paths in the training are added. Our model can capture the global dynamic, however, overestimates the Hurst having small value.



APPL AMZN GOOGL MSFT

Figure 12: Train negative log likelihoods of four stocks (smaller is better). Our model consistently has the best performance in this measure as it can flexibly learn neural SDEs parameters as well as noise parameters

Perturbation with noise from B_t We consider the case of perturbation in data under a new SDE, $dX_t = -\frac{1}{2}\beta(t)X_t dt + \sqrt{\beta(t)}dB_t$ where $B_t = \int_0^t f(t,s)dW_s$. Our approach leads to a new form

$$dX_t = [-\frac{1}{2}\beta(t)X_t + \sqrt{\beta(t)}\nu(t;u,Z)]dt + \sqrt{\beta(t)}\varsigma(t;Z)dW_t.$$

The analytic solution of $p(X_t|X_0)$ according to this SDE is rather complicated. Therefore, we resort a similar form of

$$dX_t = [-\frac{1}{2}\beta(t)\varsigma^2(t)X_t + \sqrt{\beta(t)}\nu(t;u,Z)]dt + \sqrt{\beta(t)}\varsigma(t;Z)dW_t.$$

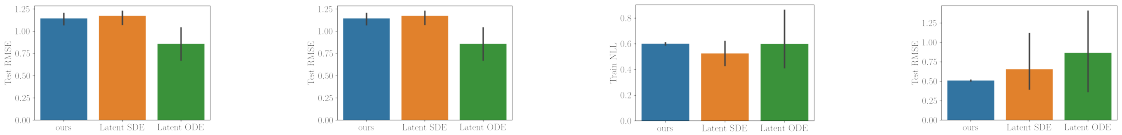
We can get a similar $p(X_t|X_0)$ but depended on $\beta(t)$ and $\varsigma(t;Z)$. Note that $\varsigma(t;Z)$ is the variance produced by sparse GP and $\nu(t;u,Z)$ is vanished because its expectation $\mathbb{E}[\nu(t;u,Z)] = 0$.

Forward vs backward Figures 19 and 20 show how our model adds noise noise during forward phases and how images are reconstructed during backward phases. These figures are of two settings: decreasing Hurst and increasing Hurst. The first case having better negative log-likelihood (better than the baseline $H = 0.5$) demonstrates that noise perturbation with more dependency at the beginning is preferable.

With a similar setup, we can test the performance on the FashionMNIST data set between"decreasing Hurst" and baseline "$H = 0.5$". We included the NLL plot in Figure 21 where "decreasing Hurst" remains having better NLLs.

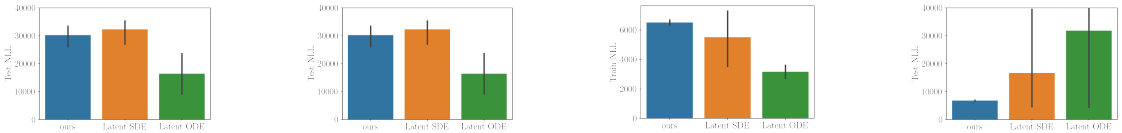
Based on the empirical evaluation on MNIST and FashionMNIST, we found that it is better to inject correlated noise (long-range dependent $H > 1/2$) at the beginning time of perturbation. At the end of perturbation, there is no need for such correlated noise but irregular one ($H < 1/2$).

With t close to 0 (close to real images), $p(X_t)$ is very complex, requiring a careful perturbation in which correlated noise ($H > 1/2$) is a good choice because it retains dependency between two time steps. On the other hand, with t close to the terminal time T, $p(X_t)$ becomes close to a Gaussian distribution so that we can accelerate the perturbation by adding even more irregular noise ($H < 1/2$).



APPL AMZN GOOGL MSFT

Figure 13: Test root mean square error



APPL AMZN GOOGL MSFT

Figure 14: Test negative log likelihood

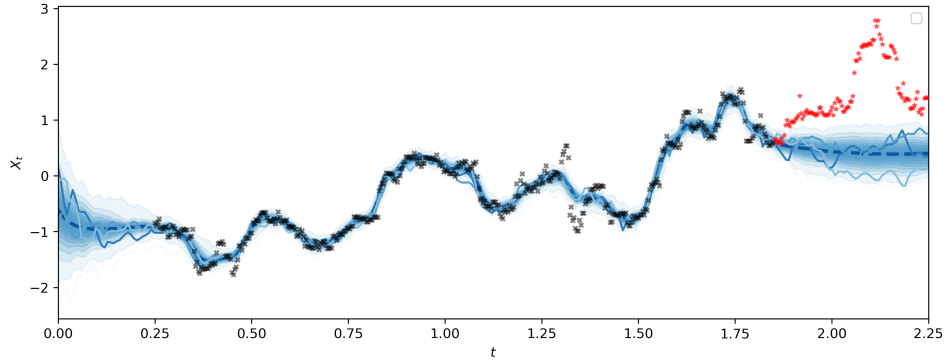


Figure 15: Model fit of AMZN

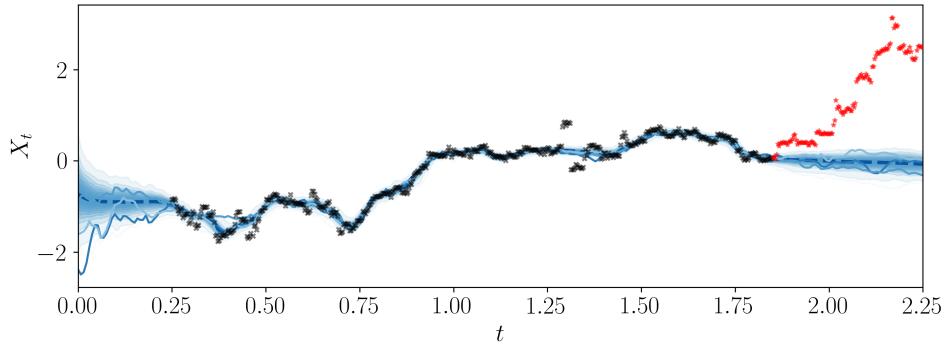


Figure 16: Model fit of APPL

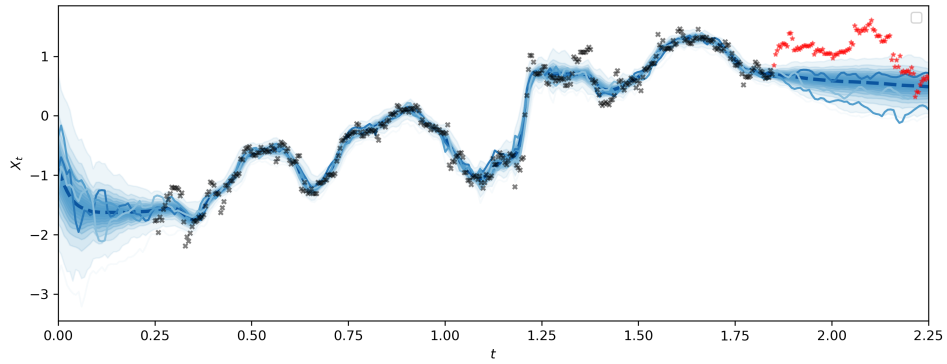


Figure 17: Model fit of GOOGL

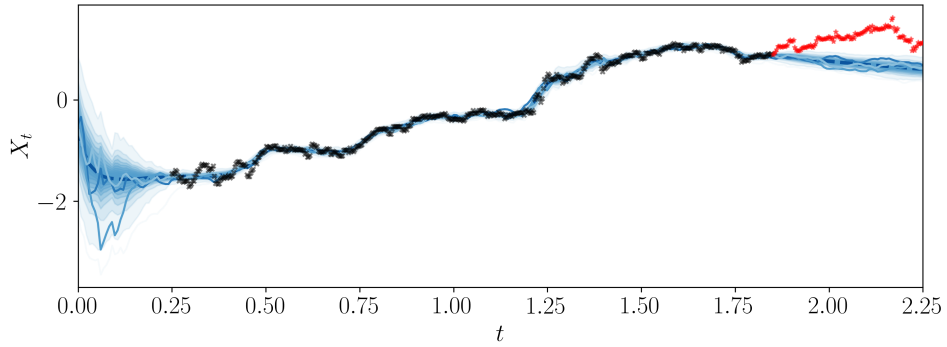


Figure 18: Model fit of MSFT

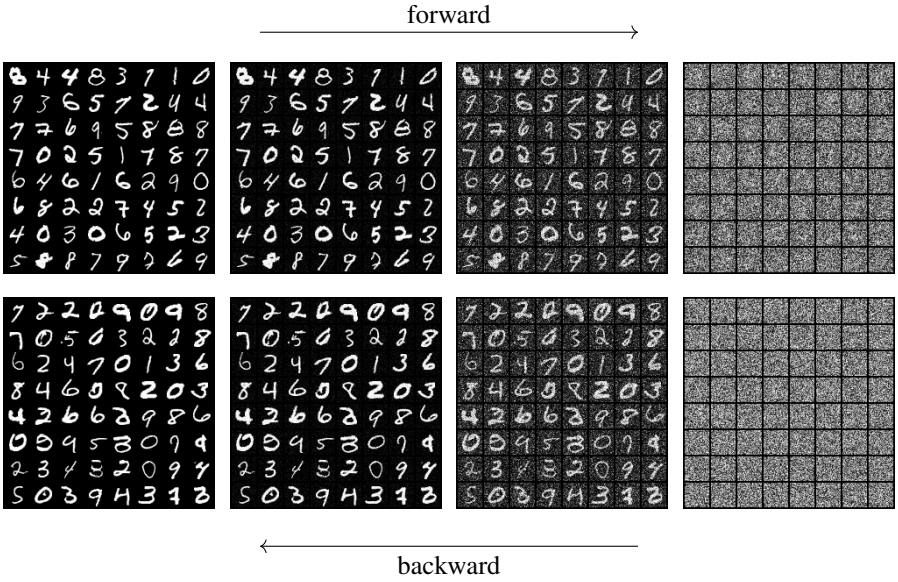


Figure 19: Decreasing Hurst: the Hurst exponent linearly decreases from 0.8 to 0.3.

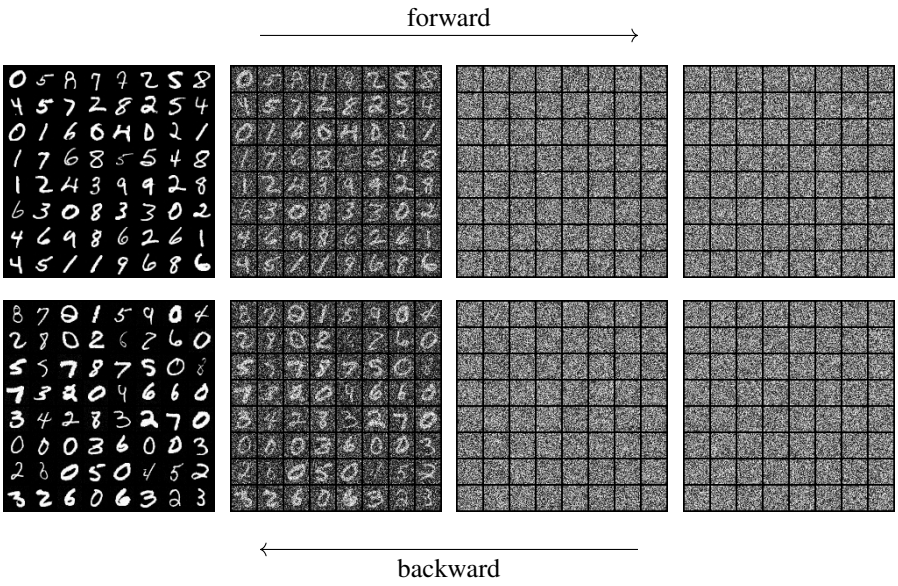


Figure 20: Increasing Hurst: The Hurst exponent linearly increases from 0.3 to 0.5

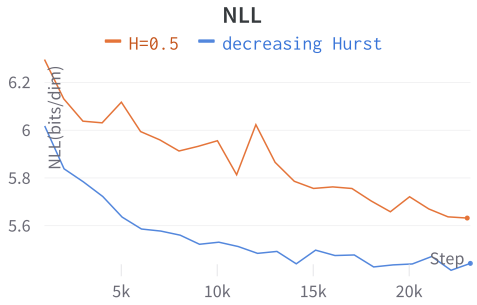


Figure 21: NLL over a batch during training score-based generative models on FashionMNIST data set. The setting decreasing Hurst is better than baseline $H = 1/2$ in terms of NLLs

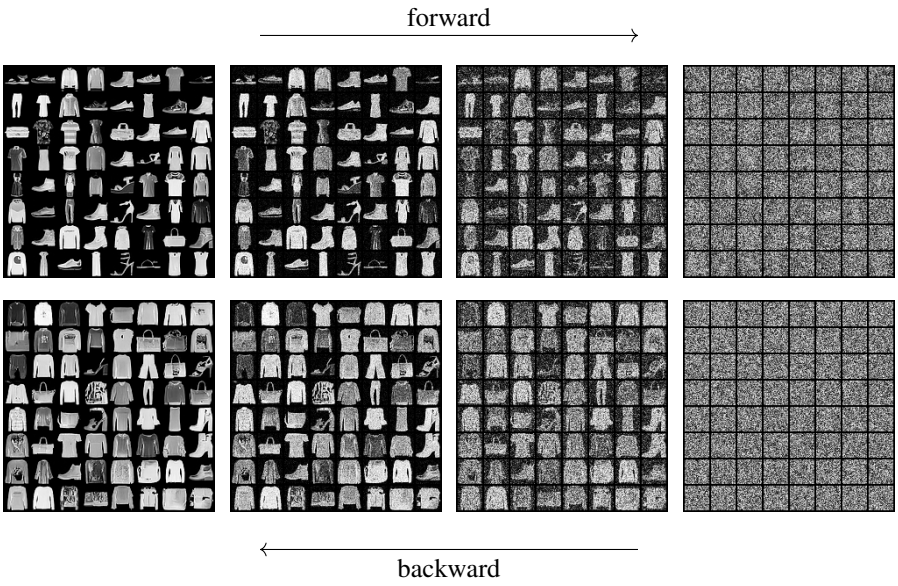


Figure 22: Forward and backward of FashionMNIST dataset for the case of decreasing Hurst.