Journal of Machine Learning Research 27 (2026) 1-23 Submitted 9/23; Revised 7/25; Published 1/26

Nonparametric generative modeling for time series via Schrödinger bridge

Mohamed HAMDOUCHE MOHAMED.HAMDOUCHE AT OUTLOOK.COM
LPSM, Université Paris Cité and Sorbonne Université, and Qube Research and Technologies

Pierre HENRY-LABORDERE PIERRE.HENRYLABORDERE AT QUBE-RT.COM
Qube Research and Technologies

Huyên PHAM* HUYEN.PHAM AT POLYTECHNIQUE.EDU
CMAP, Ecole Polytechnique

Editor: Jianfeng Lu

Abstract

We propose a novel generative model for time series based on Schrödinger bridge (SB) approach. This consists in the entropic interpolation via optimal transport between a reference probability measure on path space and a target measure consistent with the joint data distribution of the time series. The solution is characterized by a stochastic differential equation on finite horizon with a path-dependent drift function, hence respecting the temporal dynamics of the time series distribution. We estimate the drift function from data samples by nonparametric, e.g. kernel regression methods, and the simulation of the SB diffusion yields new synthetic data samples of the time series.

The performance of our generative model is evaluated through a series of numerical experiments. First, we test with autoregressive models, a GARCH Model, and the example of fractional Brownian motion, and measure the accuracy of our algorithm with marginal, temporal dependencies metrics, and predictive scores. Next, we use our SB generated synthetic samples for the application to deep hedging on real-data sets.

Keywords: Generative models, time series, Schrödinger bridge, kernel estimation, deep hedging.

1. Introduction

Sequential data appear widely in our society like in video and audio data, and simulation of time series models are used in various industrial applications including clinical predictions Lyu et al. (2018), and weather forecasts Chen et al. (2018). In the financial industry, simulations of dynamical scenarios are considered in market stress tests, risk measurement, and risk management, e.g. in deep hedging Buehler et al. (2019). The design of time series model is a delicate issue, requiring expensive calibration task, and subject to error

∗. This author is supported by the BNP-PAR Chair "Futures of Quantitative Finance", the Chair Société Générale "Risques Financiers", and by FiME, Laboratoire de Finance des Marchés de l'Energie, and the "Finance and Sustainable Development" EDF - CACIB Chair

misspecification and model risk. Therefore, the generation for synthetic samples of time series has gained an increasing attention over the last years, and opens the door in the financial sector for a pure data driven approach in risk management.

Generative modeling (GM) has become over the last years a successful machine learning task for data synthesis notably in (static) image processing. Several competing methods have been developed and state-of-the-art includes *Likelihood-based models* like energy-based models (EBM) LeCun et al. (2006), variational auto-encoders (VAE) Kingma and Welling (2014), *Implicit generative models* with the prominent works on generative adversarial network (GAN) Goodfellow et al. (2014) and its extensions Arjovsky et al. (2017), and recently the new *generation of score-based models* using Langevin dynamics, Song and Ermon (2019), Song et al. (2021), Guth et al. (2022), and diffusion models via Schrödinger bridge, see Wang et al. (2021), De Bortoli et al. (2021). We also mention Henry-Labordere (2019) for the application of diffusion Schrödinger bridge to calibration of stochastic volatility models. Generative methods for time series raises challenging issues for learning efficiently the temporal dependencies. Indeed, in order to capture the potentially complex dynamics of variables across time, it is not sufficient to learn the time marginals or even the joint distribution without exploiting the sequential structure. An increasing attention has been paid to these methods in the literature and state-of-the-art generative methods for time series are: *Time series GAN* Yoon et al. (2019) which combines an unsupervised adversarial loss on real/synthetic data and supervised loss for generating sequential data, *Quant GAN* Wiese et al. (2020) with an adversarial generator using temporal convolutional networks, *Causal optimal transport COT-GAN* Xu et al. (2020) with adversarial generator using the adapted Wasserstein distance for processes, and *Signature embedding of time series* Fermanian (2019), Ni et al. (2020), Buehler et al. (2020). We also mention stochastic differential equations representation of the time series with parametric (e.g. Neural networks) coefficients that are trained to fit with real samples, see Remlinger et al. (2021) and Kidger et al. (2021).

In this paper, we develop a novel generative model based on Schrödinger bridge approach that captures the temporal dynamics of the time series. This consists in the entropic interpolation via optimal transport between a reference probability measure on path space and a target measure consistent with the joint data distribution of the time series. The solution is characterized by a stochastic differential equation on finite horizon with a path-dependent drift function, called Schrödinger bridge time series (SBTS) diffusion. The drift function is estimated by nonparametric regression, and the simulation of the SBTS diffusion yields new synthetic data samples of the time series.

While our work builds on Schrödinger bridge framework, it differs in both scope and methodology from recent approaches that focus on transporting between marginal (or static) distributions: In Wang et al. (2021), the authors perform generative models by solving two SB problems. The paper De Bortoli et al. (2021) formulates generative modeling by computing the SB problem between the data and prior distribution. In Chizat et al. (2022), a min-entropy criterion is used to infer a trajectory aiming to recover the law of a stochastic process given samples from its temporal marginals at various time-points. The very recent work Chen et al. (2025) proposes momentum SB by considering an additional

velocity variable for learning multi marginal distributions. Let us mention also the recent paper Ram Somnath et al. (2023) that combines SB with h-transform in order to respect aligned data. In contrast, our model targets the generation of entire time series, and interpolates between joint distributions over temporal trajectories. This leads to different learning objectives: rather than estimating a marginal score, we learn a time-dependent drift that defines the conditional evolution of the process. Moreover, our drift estimation strategy is non-parametric and data-driven, relying on kernel-based methods applied to samples of observed trajectories. While Wang et al. (2021) uses a logistic regression for estimating the density ratio and then the drift function, which requires additional samples from Gaussian noises, and De Bortoli et al. (2021) performs an extension of the Sinkhorn algorithm, we propose a kernel regression method relying solely on data samples, and this turns out to be quite simple, efficient and low-cost computationally. Compared to GAN type methods, the simulation of synthetic samples from SBTS is much faster as it does not require the training of neural networks. As a result, our approach is specifically tailored to modeling the dynamics of time series from limited observations, rather than static data generation.

We validate our methodology with several numerical experiments. We first test on some time series models like autoregressive, GARCH models, and also for the fractional Brownian motion with rough paths. The accuracy is measured by some metrics aiming to capture the temporal dynamics, the correlation structure and the predictive characteristics. We also provide operational metrics of interest for the financial industry by implementing our results on real data-sets, and applying to the deep hedging of call options.

2. Background on Schrödinger bridge and problem formulation

Let μ be the distribution of a time series representing the evolution of some $\mathbb{R}^d$-valued process of interest (e.g. asset price, claim process, audio/video data, etc), and suppose that one can observe samples of this process at given fixed times of a discrete time grid $\mathcal{T} = \{t_i, i = 1, \dots, N\}$ on $(0, \infty)$. We set $T = t_N$ as the terminal observation horizon. Our goal is to construct a model that generates time series samples according to the unknown target distribution $\mu \in \mathcal{P}((\mathbb{R}^d)^N)$ the set of probability measures on $(\mathbb{R}^d)^N$. For that objective, we propose a dynamic modification of the Schrödinger bridge as follows. Let $\Omega = C([0,T];\mathbb{R}^d)$ be the set of $\mathbb{R}^d$-valued continuous functions on $[0,T]$, called *path space*, $X = (X_t)_t$ the canonical process: $X_t(\omega) = \omega(t)$, $0 \le t \le T$, for $\omega \in \Omega$, with initial value $X_0 = 0$, and $\mathbb{F} = (\mathcal{F}_t)_t$ the canonical filtration. Denoting by $\mathcal{P}(\Omega)$ as the space of probability measures on Ω, we search for $\mathbb{P}$ (representing the theoretical generative model) in $\mathcal{P}(\Omega)$, close to the Wiener measure $\mathbb{W}$ (i.e. the probability measure on Ω s.t. X is a Brownian motion) in the sense of Kullback-Leibler (or relative entropy), and consistent with the observation samples. In other words, we look for a probability measure $\mathbb{P}^* \in \mathcal{P}(\Omega)$ solution to:

$$\mathbb{P}^* \in \arg \min_{\mathbb{P} \in \mathcal{P}^\mu_\mathcal{T}(\Omega)} \mathcal{H}(\mathbb{P}|\mathbb{W}), \tag{2.1}$$

where $\mathcal{P}^{\mu}_{\mathcal{T}}(\Omega)$ is the set of probability measures $\mathbb{P}$ on Ω with joint distribution μ at $(t_1,\ldots,t_N)$, i.e., $\mathbb{P}\circ(X_{t_1},\ldots,X_{t_N})^{-1}=\mu$, and $\mathcal{H}(.|.)$ is the relative entropy between two probability measures defined by

$$\mathcal{H}(\mathbb{P}|\mathbb{W}) = \begin{cases} \int \ln \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{W}} \mathrm{d}\mathbb{P}, & \text{if } \mathbb{P} \ll \mathbb{W} \\ \infty, & \text{otherwise .} \end{cases}$$

Denoting by $\mathbb{E}_{\mathbb{P}}$ and $\mathbb{E}_{\mathbb{W}}$ the expectation on Ω under $\mathbb{P}$ and $\mathbb{W}$, we see that $\mathcal{H}(\mathbb{P}|\mathbb{W}) = \mathbb{E}_{\mathbb{P}}[\ln \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{W}}] = \mathbb{E}_{\mathbb{W}}[\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{W}} \ln \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{W}}]$ when $\mathbb{P} \ll \mathbb{W}$. Compared to the classical Schrödinger bridge (SB) (see Leonard (2014), and the application to generative modeling in Wang et al. (2021)), which looks for a probability measure $\mathbb{P}$ that interpolates between an initial probability measure and a target probability distribution at terminal time T, here, we take into account via the constraint in $\mathcal{P}^{\mu}_{\mathcal{T}}(\Omega)$ the temporal dependence of the process observed at sequential times $t_1 < \ldots < t_N$, and look for an entropic interpolation of the time series distribution. We call (2.1) the Schrödinger bridge for time series (SBTS) problem.

Let us now formulate (SBTS) as a stochastic control problem following the well-known connection established for classical (SB) in Dai Pra (1991). Given $\mathbb{P} \in \mathcal{P}(\Omega)$ with finite relative entropy $\mathcal{H}(\mathbb{P}|\mathbb{W}) < \infty$, it is known by Girsanov's theorem that one can associate to $\mathbb{P}$ an $\mathbb{F}$-adapted $\mathbb{R}^d$-valued process $\alpha = (\alpha_t)$ with finite energy: $\mathbb{E}_{\mathbb{P}}[\int_0^T |\alpha_t|^2 \mathrm{d}t] < \infty$ such that

$$\ln \frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{W}} = \int_0^T \alpha_t . \mathrm{d}X_t - \frac{1}{2}\int_0^T |\alpha_t|^2 \mathrm{d}t, \tag{2.2}$$

and $X_t - \int_0^t \alpha_s \mathrm{d}s$, $0 \le t \le T$, is a Brownian motion under $\mathbb{P}$. We then have

$$\mathcal{H}(\mathbb{P}|\mathbb{W}) = \mathbb{E}_{\mathbb{P}}\Big[\frac{1}{2}\int_0^T |\alpha_t|^2 \mathrm{d}t\Big].$$

Therefore, (SBTS) is reformulated equivalently in the language of stochastic control as:

$$\begin{cases} \text{Minimize over } \alpha \in \mathcal{A}, \quad J(\alpha) \;=\; \mathbb{E}_{\mathbb{P}}\Big[\frac{1}{2}\int_0^T |\alpha_t|^2 \mathrm{d}t\Big] \\ \text{subject to } \; \mathrm{d}X_t \;=\; \alpha_t \mathrm{d}t + \mathrm{d}W_t, \; X_0 = 0, \quad (X_{t_1},\ldots,X_{t_N}) \overset{\mathbb{P}}{\sim} \mu, \end{cases}$$

where W is a Brownian motion under $\mathbb{P}$, $\mathcal{A}$ is the set of $\mathbb{R}^d$-valued $\mathbb{F}$-adapted processes s.t. $\mathbb{E}_{\mathbb{P}}[\int_0^T |\alpha_t|^2 \mathrm{d}t] < \infty$, and $(X_{t_1},\ldots,X_{t_N}) \overset{\mathbb{P}}{\sim} \mu$ is the usual notation for $\mathbb{P}\circ(X_{t_1},\ldots,X_{t_N})^{-1} = \mu$. In the sequel, when there is no ambiguity, we omit the reference on $\mathbb{P}$ in $\mathbb{E} = \mathbb{E}_{\mathbb{P}}$ and $\sim = \overset{\mathbb{P}}{\sim}$. We denote by V_{SBTS} the infimum of this stochastic control problem under joint distribution constraint:

$$V_{SBTS} := \inf_{\alpha \in \mathcal{A}^{\mu}_{\mathcal{T}}} J(\alpha),$$

where $\mathcal{A}^{\mu}_{\mathcal{T}}$ is the set of controls α in $\mathcal{A}$ satisfying $(X_{t_1},\ldots,X_{t_N}) \sim \mu$ with $X_t = \int_0^t \alpha_s \mathrm{d}s + W_t$. Our goal is to prove the existence of an optimal control α^* that can be explicitly derived, and then used to generate samples of the time series distribution μ via the probability measure $\mathbb{P}^*$ on Ω, i.e., the simulation of the optimal diffusion process X controlled by the drift α^*.

3. Analytic solution to Schrödinger bridge for time series

Similarly as for the classical Schrödinger bridge problem, we assume that the target distribution μ admits a density with respect to the Lebesgue measure on $(\mathbb{R}^d)^N$, and by misuse of notation, we denote by $\mu(x_1, \ldots, x_N)$ this density function. Denote by $\mu_{\mathcal{T}}^W$ the distribution of the Brownian motion on $\mathcal{T}$, i.e. of $(W_{t_1}, \ldots, W_{t_N})$, which admits a density given by (by abuse of language, we use the same notation for the measure and its density)

$$\mu_{\mathcal{T}}^W(x_1, \ldots, x_N) = \prod_{i=0}^{N-1} \frac{1}{\sqrt{2\pi(t_{i+1} - t_i)}} \exp\Big(- \frac{|x_{i+1} - x_i|^2}{2(t_{i+1} - t_i)}\Big), \tag{3.1}$$

for $(x_1, \ldots, x_N) \in (\mathbb{R}^d)^N$ (with the convention that $t_0 = 0$, $x_0 = 0$). The measure μ is absolutely continuous with respect to $\mu_{\mathcal{T}}^W$, and we shall assume that its relative entropy is finite, i.e.,

$$\mathcal{H}(\mu|\mu_{\mathcal{T}}^W) \;=\; \int \ln \frac{\mu}{\mu_{\mathcal{T}}^W} \mathrm{d}\mu < \infty.$$

The explicit solution to the (SBTS) problem is provided in the following theorem.

Theorem 3.1 *The diffusion process $X_t = \int_0^t \alpha_s^* \mathrm{d}s + W_t$, $0 \le t \le T$, with α^* defined as*

$$\alpha_t^* = \; \mathrm{a}^*(t, X_t; (X_{t_i})_{t_i \le t}), \quad 0 \le t < T,$$

with $\mathrm{a}^*(t, x; \boldsymbol{x}_i)$, for $t \in [t_i, t_{i+1})$, $\boldsymbol{x}_i = (x_1, \ldots, x_i) \in (\mathbb{R}^d)^i$, $x \in \mathbb{R}^d$, given by

$$\mathrm{a}^*(t, x; \boldsymbol{x}_i) = \nabla_x \ln \mathbb{E}_{\mathbb{W}}\Big[\frac{\mu}{\mu_{\mathcal{T}}^W}(X_{t_1}, \ldots, X_{t_N}) \Big| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i, X_t = x \Big],$$

where we set $\boldsymbol{X}_{t_i} = (X_{t_1}, \ldots, X_{t_i})$, induces a probability measure $\mathbb{P}^* = \frac{\mu}{\mu_{\mathcal{T}}^W}(X_{t_1}, \ldots, X_{t_N})\mathbb{W}$, which solves the Schrödinger bridge time series problem. Moreover, we have

$$V_{SBTS} = \; \mathcal{H}(\mathbb{P}^*|\mathbb{W}) \;=\; \mathcal{H}(\mu|\mu_{\mathcal{T}}^W).$$

Proof. First, observe that $\mathbb{E}_{\mathbb{W}}[\frac{\mu}{\mu_{\mathcal{T}}^W}(X_{t_1}, \ldots, X_{t_N})] = 1$, and thus one can define a probability measure $\mathbb{P}^* \ll \mathbb{W}$ with density process

$$Z_t = \; \mathbb{E}_{\mathbb{W}}\Big[\frac{\mathrm{d}\mathbb{P}^*}{\mathrm{d}\mathbb{W}} \Big| \mathcal{F}_t \Big] \;=\; \mathbb{E}_{\mathbb{W}}\Big[\frac{\mu}{\mu_{\mathcal{T}}^W}(X_{t_1}, \ldots, X_{t_N}) \Big| \mathcal{F}_t \Big], \quad 0 \le t \le T.$$

Notice from the Markov and Gaussian properties of the Brownian motion that for $t \in [t_i, t_{i+1})$, $i = 0, \ldots, N-1$, we have $Z_t = h_i(t, X_t; \boldsymbol{X}_{t_i})$, where for a path $\boldsymbol{x}_i = (x_1, \ldots, x_i) \in (\mathbb{R}^d)^i$, $h_i(; \boldsymbol{x}_i)$ is defined on $[t_i, t_{i+1}) \times \mathbb{R}^d$ by

$$\begin{aligned} h_i(t, x; \boldsymbol{x}_i) = \; \mathbb{E}_{\boldsymbol{Y}^i \sim \mathcal{N}(0, I_{d\times(N-i)})} \Big[& \frac{\mu}{\mu_{\mathcal{T}}^W}(\boldsymbol{x}_i, x + \sqrt{t_{i+1} - t} Y_{i+1}, \ldots, \\ & x + \sqrt{t_{i+1} - t} Y_{i+1} + \sum_{j=i+1}^{N-1} \sqrt{t_{j+1} - t_j} Y_{j+1}) \Big] \end{aligned}$$

for $t \in [t_i, t_{i+1})$, $x \in \mathbb{R}^d$, and $\mathbb{E}_{\boldsymbol{Y}^i \sim \mathcal{N}(0, I_{d\times(N-i)})}[.]$ is the expectation when $\boldsymbol{Y}^i = (Y_{i+1}, \ldots, Y_N)$ is distributed according to the Gaussian law $\mathcal{N}(0, I_{d\times(N-i)})$. Moreover, by the law of conditional expectations, we have

$$h_i(t, x; \boldsymbol{x}_i) = \mathbb{E}_{\mathbb{W}}\Big[h_{i+1}(t_{i+1}, X_{t_{i+1}}; \boldsymbol{x}_i, X_{t_{i+1}})\big|X_t = x\Big],$$

with the convention that $h_N(t_N, x; \boldsymbol{x}_{N-1}, x) = \frac{\mu}{\mu^W_{\mathcal{T}}}(x_1, \ldots, x_{N-1}, x)$. Therefore, for $i = 0, \ldots, N-1$, and $\boldsymbol{x} \in (\mathbb{R}^d)^i$, the function $(t, x) \mapsto h_i(t, x; \boldsymbol{x}_i)$ is a strictly positive $C^{1,2}([t_i, t_{i+1}) \times \mathbb{R}^d) \cap C^0([t_i, t_{i+1}] \times \mathbb{R}^d)$ classical solution to the heat equation

$$\frac{\partial h_i(.; \boldsymbol{x}_i)}{\partial t} + \frac{1}{2}\Delta_x h_i(.; \boldsymbol{x}_i) = 0, \quad \text{on } [t_i, t_{i+1}) \times \mathbb{R}^d,$$

with the terminal condition: $h_i(t_{i+1}, x; \boldsymbol{x}_i) = h_{i+1}(t_{i+1}, x; \boldsymbol{x}_i, x)$ (here Δ_x is the Laplacian operator). By applying Itô's formula to the martingale density process Z of $\mathbb{P}^*$ under the Wiener measure $\mathbb{W}$, we derive

$$\begin{aligned}
\mathrm{d}Z_t &= \nabla_x h_i(t, X_t; \boldsymbol{X}_{t_i})\mathrm{d}X_t, \\
&= Z_t \nabla_x \ln h_i(t, X_t; \boldsymbol{X}_{t_i})\mathrm{d}X_t, \quad t_i \le t < t_{i+1},
\end{aligned}$$

for $i = 0, \ldots, N-1$. Thus, by defining the process α^* by $\alpha^*_t = \nabla_x \ln h_i(t, X_t; \boldsymbol{X}_{t_i})$, for $t \in [t_i, t_{i+1})$, $i = 0, \ldots, N-1$, we have

$$\frac{\mathrm{d}\mathbb{P}^*}{\mathrm{d}\mathbb{W}} = \exp\Big(\int_0^T \alpha^*_t \mathrm{d}X_t - \frac{1}{2}\int_0^T |\alpha^*_t|^2 \mathrm{d}t\Big),$$

and by Girsanov's theorem, $X_t - \int_0^t \alpha^*_s \mathrm{d}s$ is a Brownian motion under $\mathbb{P}^*$. On the other hand, by definition of $\mathbb{P}^*$, and Bayes formula, we have for any bounded measurable function φ on $(\mathbb{R}^d)^N$:

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}^*}\big[\varphi(X_{t_1}, \ldots, X_{t_N})\big] &= \mathbb{E}_{\mathbb{W}}\Big[\frac{\mu}{\mu^W_{\mathcal{T}}}(X_{t_1}, \ldots, X_{t_N})\varphi(X_{t_1}, \ldots, X_{t_N})\Big] \\
&= \int \frac{\mu}{\mu^W_{\mathcal{T}}}(x_1, \ldots, x_N)\varphi(x_1, \ldots, X_N)\mu^W_{\mathcal{T}}(x_1, \ldots, x_N)\mathrm{d}x_1 \ldots \mathrm{d}x_N \\
&= \int \varphi(x_1, \ldots, X_N)\mu(x_1, \ldots, x_N)\mathrm{d}x_1 \ldots \mathrm{d}x_N,
\end{aligned}$$

which shows that $(X_{t_1}, \ldots, X_{t_N}) \overset{\mathbb{P}^*}{\sim} \mu$. Moreover, by noting that

$$J(\alpha^*) = \mathbb{E}_{\mathbb{P}^*}\Big[\int_0^T \frac{1}{2}|\alpha^*_t|^2 \mathrm{d}t\Big] = \mathbb{E}_{\mathbb{P}^*}\Big[\ln \frac{\mathrm{d}\mathbb{P}^*}{\mathrm{d}\mathbb{W}}\Big] = \mathcal{H}(\mathbb{P}^*|\mathbb{W}) = \mathcal{H}(\mu|\mu^W_{\mathcal{T}}) < \infty,$$

where we used in the last inequality the fact that $(X_{t_1}, \ldots, X_{t_N}) \overset{\mathbb{P}^*}{\sim} \mu$, this shows in particular that $\alpha^* \in \mathcal{A}^\mu_{\mathcal{T}}$.

It remains to show that for any $\alpha \in \mathcal{A}^{\mu}_{\mathcal{T}}$ associated to a probability measure $\mathbb{P} \ll \mathbb{W}$ with density given by (2.2), i.e. $J(\alpha) = \mathcal{H}(\mathbb{P}|\mathbb{W})$, we have

$$J(\alpha) \geq \ \mathcal{H}(\mu|\mu^W_{\mathcal{T}}). \tag{3.2}$$

For this, we write from Bayes formula and since $W_t = X_t - \int_0^t \alpha_s \mathrm{d}s$ is a Brownian motion under $\mathbb{P}$ by Girsanov theorem:

$$\begin{aligned}
1 =& \ \mathbb{E}_{\mathbb{W}}\Big[\frac{\mu}{\mu^W_{\mathcal{T}}}(X_{t_1}, \ldots, X_{t_N})\Big] \\
=& \ \mathbb{E}_{\mathbb{P}}\Big[\exp\Big(\ln \frac{\mu}{\mu^W_{\mathcal{T}}}(X_{t_1}, \ldots, X_{t_N}) - \int_0^T \alpha_t \mathrm{d}W_t - \frac{1}{2}\int_0^T |\alpha_t|^2 \mathrm{d}t\Big)\Big] \\
\geq& \ \exp\Big(\mathbb{E}_{\mathbb{P}}\Big[\ln \frac{\mu}{\mu^W_{\mathcal{T}}}(X_{t_1}, \ldots, X_{t_N}) - \int_0^T \alpha_t \mathrm{d}W_t - \frac{1}{2}\int_0^T |\alpha_t|^2 \mathrm{d}t\Big]\Big) \\
=& \ \exp\Big(\mathcal{H}(\mu|\mu^W_{\mathcal{T}}) - J(\alpha)\Big),
\end{aligned}$$

where we use Jensen's inequality, and the fact that $(X_{t_1}, \ldots, X_{t_N}) \overset{\mathbb{P}}{\sim} \mu$ in the last equality. This proves the required inequality (3.2), and ends the proof. □

Remark 3.2 *The optimal drift of the Schrödinger bridge time series diffusion is in general path-dependent: it depends at given time t not only on its current state X_t, but also on the past values $\boldsymbol{X}_{\eta(t)} = (X_{t_1}, \ldots, X_{\eta(t)})$, where $\eta(t) = \max\{t_i : t_i \leq t\}$, and we have:*

$$\mathrm{d}X_t = \ \mathrm{a}^*(t, X_t; \boldsymbol{X}_{\eta(t)})\mathrm{d}t + \mathrm{d}W_t, \quad 0 \leq t \leq T, \ \ X_0 = 0. \tag{3.3}$$

Moreover, the proof of the above theorem shows that this drift function is explicitly given by

$$\mathrm{a}^*(t, x; \boldsymbol{x}_i) = \ \frac{\nabla_x h_i(t, x; \boldsymbol{x}_i)}{h_i(t, x; \boldsymbol{x}_i)}, \quad t \in [t_i, t_{i+1}), \boldsymbol{x}_i \in (\mathbb{R}^d)^i, x \in \mathbb{R}^d, \tag{3.4}$$

for $i = 0, \ldots, N-1$, where

$$\begin{aligned}
h_i(t, x; \boldsymbol{x}_i) = \ \mathbb{E}_{\boldsymbol{Y}^i \sim \mathcal{N}(0, I_{d\times(N-i)})}\Big[&\rho(\boldsymbol{x}_i, x + \sqrt{t_{i+1} - t}Y_{i+1}, \ldots, \\
&x + \sqrt{t_{i+1} - t}Y_{i+1} + \sum_{j=i+1}^{N-1} \sqrt{t_{j+1} - t_j}Y_{j+1})\Big],
\end{aligned} \tag{3.5}$$

with $\rho := \frac{\mu}{\mu^W_{\mathcal{T}}}$ the density ratio, and $\boldsymbol{Y}^i = (Y_{i+1}, \ldots, Y_N)$.

The following result states an alternate representation of the drift function that will be useful in the next section for estimation.

Proposition 3.3 *For $i = 0, \dots, N-1$, $t \in [t_i, t_{i+1})$, $\boldsymbol{x}_i = (x_1, \dots, x_i) \in (\mathbb{R}^d)^i$, $x \in \mathbb{R}^d$, we have*

$$\mathrm{a}^*(t, x; \boldsymbol{x}_i) = \frac{1}{t_{i+1} - t} \frac{\mathbb{E}_\mu \left[(X_{t_{i+1}} - x) F_i(t, x_i, x, X_{t_{i+1}}) \big| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \right]}{\mathbb{E}_\mu \left[F_i(t, x_i, x, X_{t_{i+1}}) \big| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \right]}, \tag{3.6}$$

where

$$F_i(t, x_i, x, x_{i+1}) = \exp\left(-\frac{|x_{i+1} - x|^2}{2(t_{i+1} - t)} + \frac{|x_{i+1} - x_i|^2}{2(t_{i+1} - t_i)} \right),$$

and $\mathbb{E}_\mu[\cdot]$ denotes the expectation under μ.

Proof. Fix $i \in [\![0, N-1]\!]$, and $t \in [t_i, t_{i+1})$. From the expression of $\mu_{\mathcal{T}}^W$ in (3.1), we have

$$\begin{aligned}
&\mathbb{E}_{\mathbb{W}} \left[\frac{\mu}{\mu_{\mathcal{T}}^W}(X_{t_1}, \cdots, X_{t_N}) \Big| (X_{t_1}, \cdots, X_{t_i}) = (x_1, \cdots, x_i),\ X_t = x \right] \qquad (3.7) \\
&= C \int \frac{\mu}{\mu_{\mathcal{T}}^W}(x_1, \cdots, x_N) \exp\left(-\frac{|x_{i+1} - x|^2}{2(t_{i+1} - t)} \right) \prod_{j=i+1}^{N-1} \exp\left(-\frac{|x_{j+1} - x_j|^2}{2\Delta t_j} \right) \mathrm{d}x_{i+1} \cdots \mathrm{d}x_N \\
&= C \int F_i(t, x_i, x, x_{i+1}) \frac{\mu(x_1, \dots, x_N)}{\mu_i(x_1, \dots, x_i)} \mathrm{d}x_{i+1} \cdots \mathrm{d}x_N \ = \ C \mathbb{E}_\mu \Big[F_i(t, x_i, x, X_{t_{i+1}}) \big| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \Big],
\end{aligned}$$

where C is a constant varying from line to line and depending only on t and $\boldsymbol{x}_i = (x_1, \cdots, x_i)$, but not on x, and μ_i is the density of $(X_{t_1}, \cdots, X_{t_i})$ under μ, *i.e.*,

$$\mu_i(x_1, \cdots, x_i) = \int \mu(x_1, \cdots, x_N) \mathrm{d}x_{i+1} \cdots \mathrm{d}x_N.$$

By plugging the new expression (3.7) into a^* and differentiating with respect to x, we then get

$$\mathrm{a}^*(t, x; \boldsymbol{x}_i) = \frac{1}{t_{i+1} - t} \frac{\mathbb{E}_\mu \left[(X_{t_{i+1}} - x) F_i(t, x_i, x, X_{t_{i+1}}) \big| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \right]}{\mathbb{E}_\mu \left[F_i(t, x_i, x, X_{t_{i+1}}) \big| \boldsymbol{X}_{t_i} = \boldsymbol{x}_i \right]}.$$

□

Remark 3.4 *In the case where μ is the distribution arising from a Markov chain, i.e., in the form $\mu(\mathrm{d}x_1, \dots, x_N) = \prod_{i=1}^{N-1} \nu_i(x_i, \mathrm{d}x_{i+1})$, for some transition kernels ν_i on $\mathbb{R}^d$, then the conditional expectations in* (3.6) *will depend on the past values $\boldsymbol{X}_{t_i} = (X_{t_1}, \dots, X_{t_i})$ only via the last value X_{t_i}.*

4. Generative learning of time series

From Theorem 3.1, we can run an Euler scheme for simulating the Schrödinger bridge diffusion, and then samples of the target distribution μ. For that purpose, we need an accurate estimation of the drift terms, i.e., of the functions a_i^*, for $i = 0, \dots, N-1$. We propose several estimation methods. In the sequel, for a probability measure ν on $(\mathbb{R}^d)^N$, we denote by $\mathbb{E}_\nu[\cdot]$ the expectation of random variables under the distribution ν.

4.1 Drift estimation

Estimation of the density ratio. This method follows the idea in Wang et al. (2021). Denote by $\rho = \frac{\mu}{\mu_{\mathcal{T}}^W}$ the density ratio, and observe that the log-density ratio $\ln \rho$ minimizes over functions r on $(\mathbb{R}^d)^N$ the logistic regression function

$$L_{logistic}(r) = \mathbb{E}_{\mu}\Big[\ln\big(1+\exp(-r(\boldsymbol{X})\big)\Big] + \mathbb{E}_{\mu_{\mathcal{T}}^W}\Big[\ln\big(1+\exp(r(\boldsymbol{X})\big)\Big].$$

Therefore, given data samples $\boldsymbol{X}^{(m)} = (X_{t_1}^{(m)}, \dots, X_{t_N}^{m})$, $m = 1, \dots, M$ from μ, and using samples $\boldsymbol{Y}^{(m)}$ from $\mu_{\mathcal{T}}^W$, we estimate the density ratio ρ by

$$\hat{\rho} = \exp(r_{\hat{\theta}})$$

where $r_{\hat{\theta}}$ is the neural network that minimizes the empirical logistic loss function:

$$\theta \mapsto \frac{1}{M}\sum_{m=1}^{M} \ln\big(1+\exp(-r(\boldsymbol{X}^{(m)})\big) + \ln\big(1+\exp(r(\boldsymbol{Y}^{(m)})\big).$$

Recalling the expression (3.4) of the Schrödinger drift function, we obtain an estimator of a^* by substituting into (3.5) the above estimate $\hat{\rho}$ of ρ, and then computing the expectation with Monte-Carlo approximations from samples in $\mathcal{N}(0, I_{d\times(N-i)})$. Notice that this method is very costly as it requires in addition to the training of the neural networks for estimating the density ratio, another Monte-Carlo sampling for estimating finally the drift.

Kernel regression estimation of the drift. In order to overcome the computational issue of the above estimation method, we propose an alternative approach that relies on the representation of the drift term in Proposition 3.3. Indeed, the key feature of the formula (3.6) is that it involves (conditional) expectations under the target distribution μ, which is amenable to direct estimation using data samples. For the approximation of the conditional expectation, we can then use nonparametric regression, like e.g. the classical kernel methods.

From data samples $\boldsymbol{X}_{t_N}^{(m)} = (X_{t_1}^{(m)}, \dots, X_{t_N}^{(m)})$, $m = 1, \dots, M$ from μ, the Nadaraya-Watson estimator of the drift function is given by

$$\hat{\mathrm{a}}(t, x; \boldsymbol{x}_i) = \frac{1}{t_{i+1}-t} \frac{\displaystyle\sum_{m=1}^{M} (X_{t_{i+1}}^{(m)} - x) F_i(t, X_{t_i}^{(m)}, x, X_{t_{i+1}}^{(m)}) \boldsymbol{K}_{\boldsymbol{h}_i}^i(\boldsymbol{x}_i - \boldsymbol{X}_{t_i}^{(m)})}{\displaystyle\sum_{m=1}^{M} F_i(t, X_{t_i}^{(m)}, x, X_{t_{i+1}}^{(m)}) \boldsymbol{K}_{\boldsymbol{h}_i}^i(\boldsymbol{x}_i - \boldsymbol{X}_{t_i}^{(m)})}, \tag{4.1}$$

for $t \in [t_i, t_{i+1})$, $\boldsymbol{x}_i \in (\mathbb{R}^d)^i$, $x \in \mathbb{R}^d$, $i = 0, \dots, N-1$, where $\boldsymbol{K}^i$ is a multivariate kernel function operating on i arguments, each of dimension d, i.e., a symmetric density function

on $(\mathbb{R}^d)^i$, and $\boldsymbol{K}^i_{\boldsymbol{h}_i}(\boldsymbol{x}_i) := \frac{1}{h_1 \ldots h_i} \boldsymbol{K}^i(x_1/h_1, \ldots, x_i/h_i)$ with a vector of bandwidth $\boldsymbol{h}_i = (h_1, \ldots, h_i) \in (0, \infty)^i$. We shall use the standard multiplicative kernel

$$\boldsymbol{K}^i(\boldsymbol{x}_i) = \; K(x_1) \cdot \ldots \cdot K(x_i),$$

where K is a kernel function on $\mathbb{R}^d$, like e.g. the Gaussian kernel, or the quartic kernel: $K(x) \propto (1-|x|^2)^2 1_{|x|\leq 1}$. We point out the limitations of kernel regression methods for very high dimensional data (in d and/or N), as it is well-known they suffer from the curse of dimensionality. Indeed, in this case, the bandwidth has to be chosen large, hence inducing a high bias. In such instances, a more suitable course of action would involve employing LSTM neural networks to approximate the conditional expectation inherent in the drift function.

LSTM network approximation of the path-dependent drift. The conditional expectations in the numerator and denominator of the drift terms can alternately be approximated by neural networks. In order to achieve this, we need a neural network architecture that fits well with the path-dependency of the drift term, i.e., the *a priori* non Markov feature of the data time series distribution μ. We shall then consider a combination of feed-forward and LSTM (Long Short Term Memory) neural network.

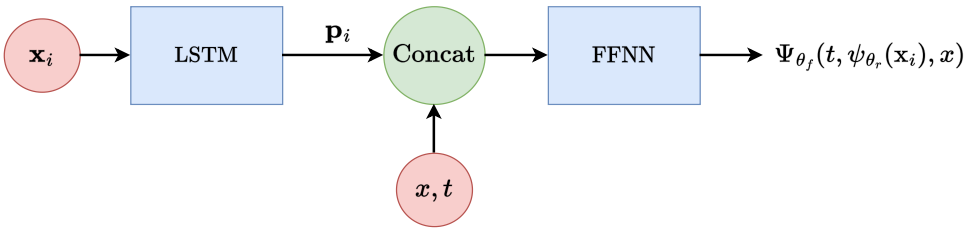

Figure 1: Architecture of the neural network

For $i = 0, \ldots, N-1$, the conditional expectation function in the numerator:

$$\begin{aligned} & (t, \boldsymbol{x}_i, x) \in [t_i, t_{i+1}) \times (\mathbb{R}^d)^i \times \mathbb{R}^d \\ \longmapsto \quad & \mathbb{E}_\mu \Big[(X_{t_{i+1}} - X_t) F(t, X_{t_i}, X_t, X_{t_{i+1}}) \big| (\boldsymbol{X}_{t_i}, X_t) = (\boldsymbol{x}_i, x) \Big], \end{aligned}$$

is approximated by $\Psi_{\theta_f}(t, p_{t_i}, x)$, where $\Psi_{\theta_f} \in NN_{d+k+1,d}$ is a feed-forward neural network with input dimension $1 + k + d$, and output dimension d, and p_{t_i} is an output vector of dimension k from an LSTM network, i.e., $p_{t_i} = \psi_{\theta_r}(\boldsymbol{x}_i)$ with $\psi_{\theta_r} \in \mathrm{LSTM}_{i,d,k}$ at time t_i. This neural network is trained by minimizing over the parameter $\theta = (\theta_f, \theta_r)$ the quadratic loss function

$$L(\theta) = \sum_{i=0}^{N-1} \hat{\mathbb{E}} \Big| (X_{t_{i+1}} - X) F(\tau, X_{t_i}, X, X_{t_{i+1}}) - \Psi_{\theta_f}(\tau, \psi_{\theta_r}(\boldsymbol{X}_{t_i}), X) \Big|^2.$$

Here $\hat{\mathbb{E}}$ is the empirical loss expectation where $(X_{t_1}, \ldots, X_{t_N})$ are sampled from the data distribution μ, τ is sampled according to an uniform law on $[t_i, t_{i+1})$, and X is sampled e.g. from a Gaussian law with mean X_{t_i} for $i = 0, \ldots, N-1$. The conditional expectation function in the denominator is similarly approximated.

The output of this neural network training yields an approximation:

$$t \in [0,T), \boldsymbol{x} \in (\mathbb{R}^d)^{\eta(t)}, x \in \mathbb{R}^d \longmapsto \hat{\mathrm{a}}(t, x; \boldsymbol{x}),$$

of the drift term function, which is then used for generating samples $(X_{t_1}, \ldots, X_{t_N})$ of μ from the simulation of the diffusion:

$$\mathrm{d}X_t = \hat{\mathrm{a}}(t, X_t; \boldsymbol{X}_{\eta(t)})\mathrm{d}t + \mathrm{d}W_t, \; X_0 = 0. \tag{4.2}$$

4.2 Schrödinger bridge time series algorithm

From the estimator $\hat{\mathrm{a}}$ of the drift path-dependent function, we can now simulate the SB SDE (4.2) by an Euler scheme. Let N_π be the number of uniform time steps between two consecutive observation dates t_i and t_{i+1}, for $i = 0, \ldots, N-1$, and $t^\pi_{k,i} = t_i + \frac{k}{N_\pi}$, $k = 0, \ldots, N_\pi - 1$, the associated time grid. The pseudo-code of the Schrödinger bridge time series (SBTS) algorithm is described in Algorithm 1.

Algorithm 1: SBTS Simulation

Input: data samples of time series $(X^{(\mathrm{m})}_{t_1}, \cdots, X^{(\mathrm{m})}_{t_N})$, $m = 1, \ldots, M$, and N_π.
Initialization: initial state $x_0 = 0$;
for $i = 0, \ldots, N-1$ **do**
  Initialize state $y_0 = x_i$;
  for $k = 0, \ldots, N_\pi - 1$ **do**
    Compute $\hat{\mathrm{a}}(t^\pi_{k,i}, y_k; \boldsymbol{x}_i)$ e.g. by kernel estimator (4.1);
    Sample $\varepsilon_k \in \mathcal{N}(0,1)$ and compute

$$y_{k+1} = y_k + \frac{1}{N_\pi}\hat{\mathrm{a}}(t^\pi_{k,i}, y_k; \boldsymbol{x}_i) + \frac{1}{\sqrt{N_\pi}}\varepsilon_k,$$

  end
  Set $x_{i+1} = y_{N_\pi}$.
end
Return: $x_1, \cdots, x_N$

The complexity of our algorithm with kernel drift estimation is of order $O(MNN_\pi)$. Indeed, in the computation of the drift from t_i to t_{i+1}, $i = 0, \ldots, N-1$, at time $t^\pi_{k,i}$, $k = 0, N_\pi - 1$, we have to evaluate $\prod_{j=1}^i K_h(X^{(\mathrm{m})}_{t_j} - x_j)$ (resp. $\sum_{j=1}^i K_h(X^{(\mathrm{m})}_{t_j} - x_j)$ for the multiplicative (resp. additive) kernel, for all paths $m = 1, \cdots, M$. These values are then stored, and therefore, when computing the kernel for the next interval, i.e., $[t_{i+1}, t_{i+2}]$, there is no necessity to recompute the entire product (resp. sum), and we only need to compute $K_h(X^{(\mathrm{m})}_{t_{i+1}} - x_{i+1})$, which reduces the computational process by a factor N.

5. Numerical experiments

In this section, we demonstrate the effectiveness of our SBTS algorithm on several examples of time series models, as well as on real data sets for an application to deep hedging. The algorithms are performed on a computer with the following characteristics: Intel(R) i7-7500U CPU @ 2.7GHz, 2 Core(s). Code and data are available at GitHub-SBTS

We shall take the quartic kernel as it tends to produce similar estimates than the Gaussian kernel, and it is less costly to compute power functions than exponential in Gaussian density. The choice of the bandwidth is much more critical, and it is a matter of tradeoff between bias and variance as largely documented in the statistical literature, see e.g. the textbooks Wand and Jones (1995), Härdle et al. (2004), Scott (2014). The value of the bandwidth can be chosen according to the *rule-of-thumb*, *Direct Plug-In* (DPI), or *cross validation*. In practice, we propose a simple approach to select the bandwidth, by considering it as a hyper-parameter to fine-tune. Given a train set $X = (X^m_{t_1}, \cdots, X^m_{t_N})_{m=1,\cdots,M}$, a test set $Y = (Y^q_{t_1}, \cdots, Y^q_{t_N})_{q=1,\cdots,Q}$, both from real data, and a list of bandwidths $H = \{h_1, \cdots, h_K\}$, we generate L realizations of $\hat{Y}^q_{t_N}$ given the first real values of the series $(Y^q_{t_1}, \cdots, Y^q_{t_{N-1}})$ for each q, using (4.1). Then, we choose $h^* \in H$ such that it minimizes

$$MSE_h = \frac{1}{Q}\sum_{q=1}^{Q}\left|\frac{1}{L}\sum_{l=1}^{L}\hat{Y}^{q,l}_{t_N} - Y^q_{t_N}\right|^2$$

with $\hat{Y}^{q,l}_{t_N}$ the l-th generated realizations of $\hat{Y}^q_{t_N}$ given $(Y^q_{t_1}, \cdots, Y^q_{t_{N-1}})$. Discussion and example about the sensitivity to the choice of the bandwidth is provided in Alouadi et al. (2025).

5.1 Evaluation metrics

In addition to visual plot of data vs generator samples path, we use some metrics to evaluate the accuracy of our generators:

- *Marginal metrics* for quantifying how well are the marginal distributions from the generated samples compared to the data ones. These include
  - Classical statistics like mean, 95% and 5% percentiles
  - Kolmogorov-Smirnov test: we compute the p-valued, and when $p > \alpha$ (usually 5%), we do not reject the null-hypothesis (generator came from data of reference distribution)
- *Temporal dynamics* metrics for quantifying the ability of the generator to capture the time structure of the time series data. We compute the empirical distribution of the quadratic variation along the time grid $\mathcal{T}$: $\sum_{i=0}^{N-1}|X_{t_{i+1}} - X_{t_i}|^2$.
- *Correlation structure* for evaluating the ability of the generator to capture the multi-dimensional structure of the time series. We shall compare the empirical covariance

or correlation matrix induced by the generator SBTS and the ones from the data samples on the time grid $\mathcal{T}$.

- *Predictive score*: We train a post-hoc sequence model to predict the latter part of a time series given the first part, using synthetic data. Then, we evaluate the trained model on real dataset. Performance is measured in terms of mean absolute error (MAE) that we shall compare with the TimeGAN method in Yoon et al. (2019).

5.2 Toy autoregressive model of time series

We consider the following toy autoregressive (AR) model:

$$\begin{cases} X_{t_1} &= b + \varepsilon_1, \\ X_{t_2} &= \beta_1 X_{t_1} + \varepsilon_2, \\ X_{t_3} &= \beta_2 X_{t_2} + \sqrt{|X_{t_1}|} + \varepsilon_3, \end{cases}$$

where the noises $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, $i = 1, \dots, 3$ are mutually independent. The model parameters are $b = 0.7$, $\sigma_1 = 0.1$, $\sigma_2 = \sigma_3 = 0.05$ and $\beta_1 = \beta_2 = -1$.

We use samples of size $M = 1000$ for simulated data of the AR model. The drift of the SBTS diffusion is estimated with a kernel of bandwidth $h = 0.05$, and simulated from euler scheme with $N_\pi = 100$. The runtime for generating 500 paths of SBTS is 8 seconds.

In Figure 2, we plot the empirical distribution of each pair (X_{t_i}, X_{t_j}) from the AR model, and from the generated SBTS. We also show the marginal empirical distributions. Table 1 presents the marginal metrics for the AR model and generator (p-value and percentiles at level 5% and 95%). In Table 2, we give the difference between the empirical correlation from generated samples and AR data samples. This shows that our generated SBTS model has accurately recovered the true marginal distributions, and captured well the correlation.

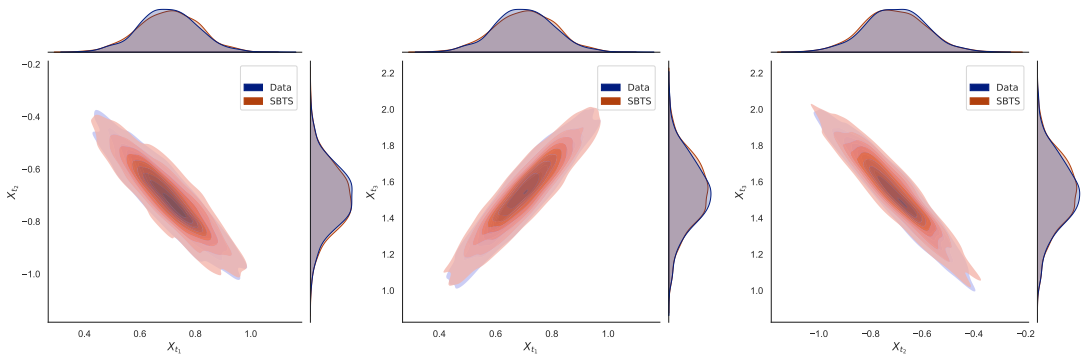

Figure 2: Comparison between the true and generated distribution for each couple (X_{t_i}, X_{t_j}) with $i, j \in [\![1, 3]\!]$ with $i \neq j$

	p-value	q_5	$\tilde{q}_5$	q_{95}	$\tilde{q}_{95}$
X_{t_1}	0.98	0.535	0.528	0.855	0.861
X_{t_2}	0.74	-0.873	-0.861	-0.516	-0.514
X_{t_3}	0.90	1.243	1.251	1.808	1.793

Table 1: Marginal metrics for AR model and generator ($\tilde{q}$ for percentile)

	X_{t_1}	X_{t_2}	X_{t_3}
X_{t_1}	0	0.014	-0.01
X_{t_2}	0.014	0	0.013
X_{t_3}	-0.01	0.013	0

Table 2: Difference between empirical correlation from generated samples and reference samples

5.3 A multivariate autoregressive Gaussian model

We consider an illustrative example as in Yoon et al. (2019):

$$X_{t_{i+1}} = \phi X_{t_i} + \varepsilon_{t_{i+1}}, \quad \text{with} \quad \varepsilon_{t_i} \sim \mathcal{N}(0, \sigma \mathbf{1}_d + (1-\sigma)\mathbb{I}_d),$$

where $\mathbf{1}_d$ is the $d \times d$ matrix with unit elements, and $\mathbb{I}_d$ is the identity matrix. The coefficient $\phi \in [0,1]$ controls the correlation across time steps while $\sigma \in [-1,1]$ controls the correlation across the components (features).

We want to illustrate the predictive characteristics of our SBTS method, i.e. its ability to capture the conditional distribution over time. We simulate the conditional mean in the SBTS model, and compute the predictive score as the mean absolute error between this conditional mean and the true value given in the AR model by ϕX_{t_i}. It is compared with the predictive score of the TimeGAN method Yoon et al. (2019), and the results are reported in Table 3 in the two-dimensional case $d = 2$. We note that the scores are comparable, but the SBTS has a better prediction than TimeGAN

when varying temporal correlations.

	Temporal correlation (fixing $\sigma = 0.8$)			Feature correlation (fixing $\phi = 0.8$)		
Settings	$\phi = 0.2$	$\phi = 0.5$	$\phi = 0.8$	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 0.8$
Predictive score (lower the better)						
SBTS	$\mathbf{.161 \pm .016}$	$\mathbf{.180 \pm .026}$	$\mathbf{.244 \pm .014}$	$.325 \pm .052$	$.295 \pm .038$	$\mathbf{.244 \pm .014}$
	$.640 \pm 0.003$	0.412 ± 0.002	$.251 \pm .002$	$\mathbf{.282 \pm .005}$	$\mathbf{.261 \pm .002}$	$.251 \pm .002$

Table 3: Predictive score for SBTS vs TimeGAN

5.4 GARCH Model

We consider a GARCH model:

$$\begin{cases} X_{t_{i+1}} &= \sigma_{t_{i+1}} \varepsilon_{t_{i+1}} \\ \sigma^2_{t_{i+1}} &= \alpha_0 + \alpha_1 X^2_{t_i} + \alpha_2 X^2_{t_{i-1}}, \quad i = 1, \dots, N, \end{cases}$$

with $\alpha_0 = 5$, $\alpha_1 = 0.4$, $\alpha_2 = 0.1$, and the noises $\varepsilon_{t_i} \sim \mathcal{N}(0, 0.1)$, $i = 1, \dots, N$, are i.i.d. The size of the time series is $N = 60$.

The hyperparameters for the training and generation of SBTS are $M = 1000$, $N_\pi = 100$, and a bandwidth for the kernel estimation $h = 0.2$ (larger than for the AR model since by nature the GARCH process is more "volatile"). The runtime for generating 1000 paths is 120 seconds.

In Figure 3, we plot four sample paths of the GARCH process to be compared with four sample paths of the SBTS. Figure 4 represents samples plot of the joint distribution between (X_{t_1} and the terminal value X_{t_N}) of the time series. Figure 5 provides some metrics to evaluate the performance of SBTS. On the left, we represent the p-value for each of the marginals of the generated SBTS. This confirms that our model has accurately recovered the true marginal distributions. On the right, we compute for each marginal index $i = 1, \dots, N = 60$, the difference between ρ_i and $\hat{\rho}_i$ where ρ_i (resp. $\hat{\rho}_i$) is the sum over j of the empirical correlation between X_i and X_{t_j} from GARCH (resp. generated SBTS), and plot its mean and standard deviation. This shows that generated SBTS captures accurately the correlation in the time series.

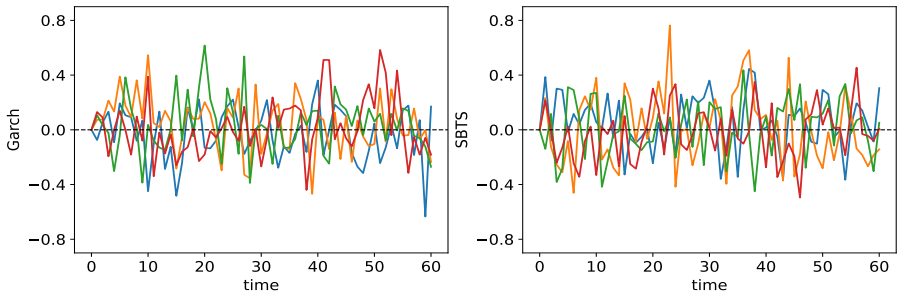

Figure 3: Samples path of reference GARCH (left) and generator SBTS (right)

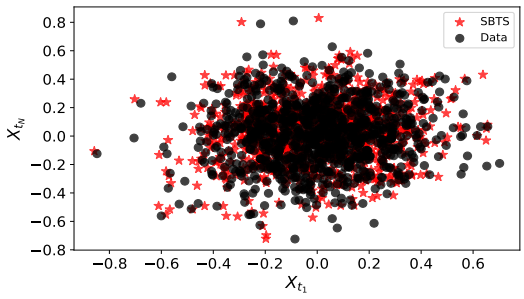

Figure 4: Samples plot of the joint distribution (X_{t_1}, X_{t_N})

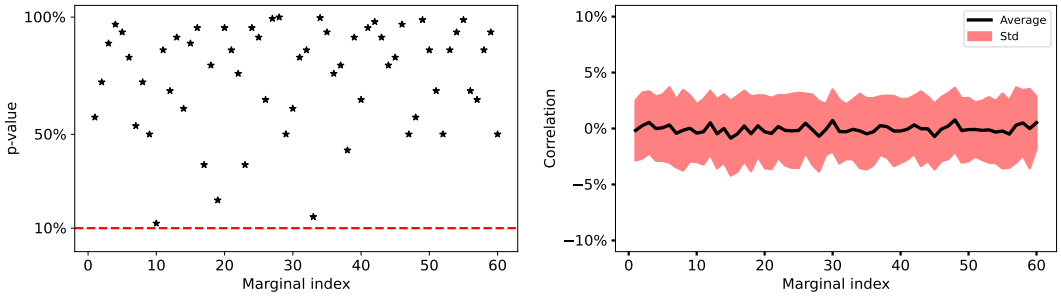

Figure 5: *Left*: p-value for the marginals X_{t_i}. *Right*: Difference between the term-by-term empirical correlation from generated samples and reference samples.

5.5 Fractional Brownian Motion

We consider a fractional Brownian motion (FBM) with Hurst index H that measures the roughness of this Gaussian process. We plot in Figure 6 four samples path of FBM with $H = 0.1$, and samples paths generated by SBTS. The generator is trained with $M = 1000$ sample paths, and the hyperparameters used for the simulation are $N_\pi = 100$, with bandwidth $h = 0.05$ for the kernel estimation of the Schrödinger drift. The runtime for 1000 paths is 100 seconds.

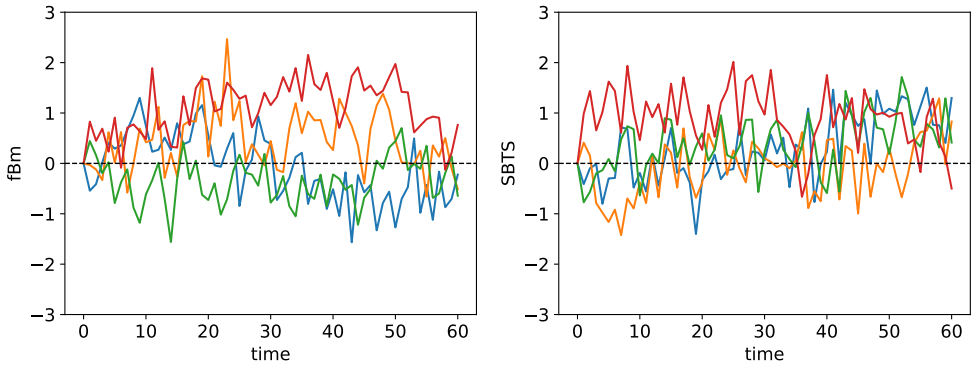

Figure 6: Samples path of reference FBM (left) and generator SBTS (right)

Figure 7 represents the covariance matrix of $(X_{t_1}, \ldots, X_{t_N})$ for $N = 60$ of the FBM and of the generated SBTS, where each entry corresponds to the covariance between X_{t_i} and X_{t_j}), $i, j \in [\![1, 60]\!]$. The values are visualized through a color-coded heatmap. We also plot in Figure 8 the empirical distribution of the quadratic variation $\sum_{i=0}^{N-1} |X_{t_{i+1}} - X_{t_i}|^2$ for the FBM and the SBTS. These figures allow for a direct comparison of second-order temporal dependencies between real and generated data, and the similarity suggests that the generative model accurately captures the time dependent covariance and quadratic variation structure of the observed time series.

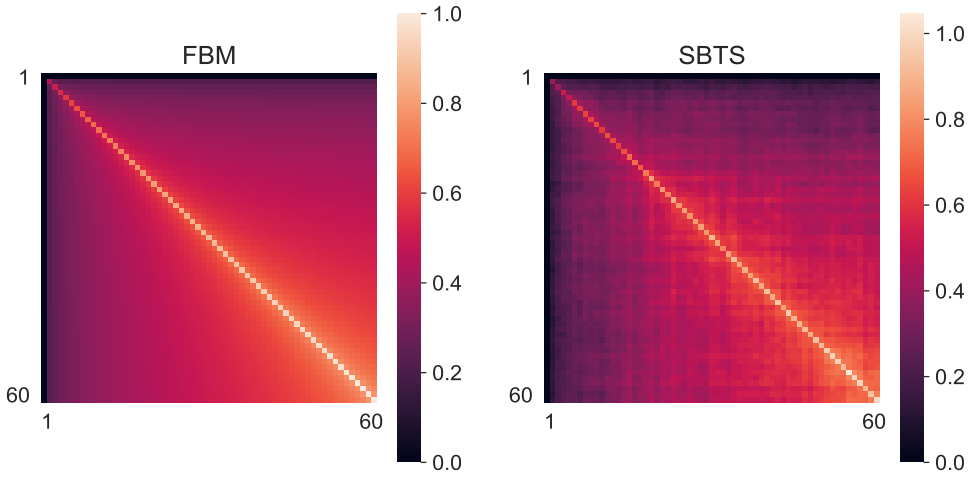

Figure 7: Covariance matrix for reference FBM and SBTS

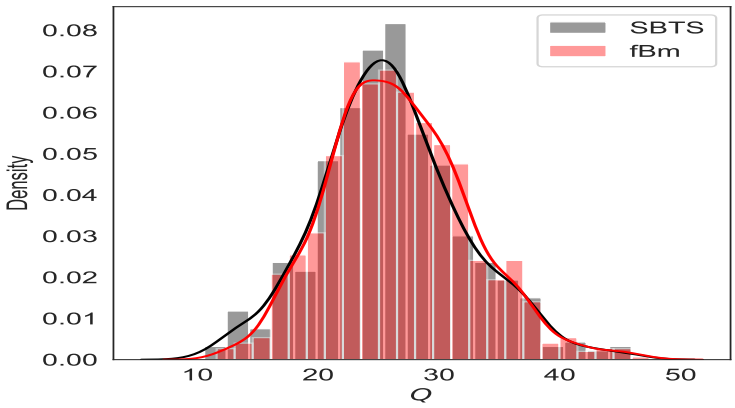

Figure 8: Quadratic variation distribution

Finally, we provide estimate of the Hurst index from our generated SBTS with the standard estimator (see e.g. Gairing et al. (2020)) given by:

$$\hat{H} = \frac{1}{2}\left[1 - \frac{\log\left(\sum_{i=0}^{N-1} |X_{t_{i+1}} - X_{t_i}|^2\right)}{\log N}\right].$$

for $N = 60$, we get: $\hat{H} = 0.102$, Std $= 0.003$.

5.6 Application to deep hedging on real-data sets

In this paragraph, we use generated time series for applications to risk management, and notably the pricing of derivatives and the computation of associated hedging strategies via deep hedging approach. We use samples of historical data for generating by SBTS new synthetic time series samples. We then compute deep hedging strategies that are trained from these synthetic samples, and we compare with the PnL and the replication error based on historical dataset. This experiment can be considered as a reinforcement learning task. The general backtest procedure is illustrated in Figure 9.

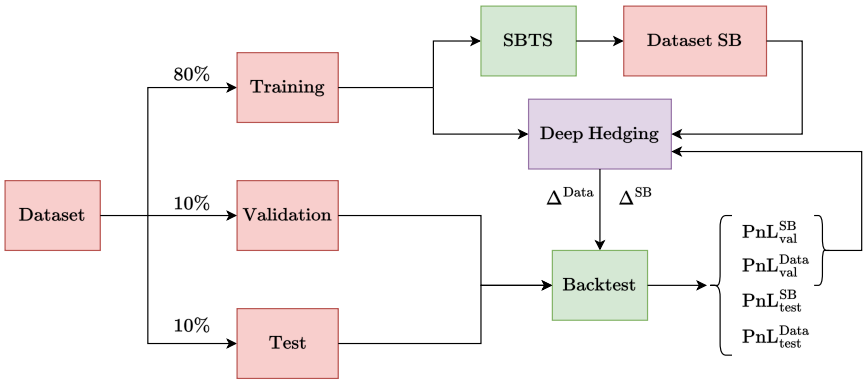

Figure 9: Procedure of backtest for deep hedging

We consider stock price S from the company *Apple* with data (ticker is AAPL) from january 1, 2010 to january 30, 2020, and produce $M = 2500$ samples of $N = 60$ successive days, with a sliding window. The hyperparameters for the generation of SBTS synthetic samples are $N_\pi = 100$, bandwidth $h = 0.05$.

Remark 5.1 *Since we only observe a single realization of the time series, generating training samples via sliding windows implicitly relies on an assumption of (local) stationarity. This assumption is necessary to justify treating overlapping segments as approximately drawn from the same underlying distribution. In real-world applications such as finance, raw asset prices are typically non-stationary due to trends and volatility regimes. However, it is standard practice to apply preprocessing transformations—such as computing log-returns—to induce stationarity. In our deep hedging experiment with Apple stock data, we follow this approach by modeling log-returns instead of prices. This transformation ensures that the stationarity assumption underlying our learning framework is more appropriate. A more detailed discussion of this preprocessing and its impact on generative modeling is provided in our companion paper Alouadi et al. (2025).*

We plot in Figure 10 four sample paths of the SBTS diffusion to be compared with the real ones from *Apple*. We illustrate the excess of kurtosis of the real data by plotting in Figure 11 the tail distribution for the return $R_{t_i} = \frac{S_{t_{i+1}}}{S_{t_i}} - 1$: x in log-scale $\mapsto \mathbb{P}[|R| \geq x]$,

and found that the excess of kurtosis of real data is 1.96, to be compared with the one from SBTS, and equal to 2.34. Figure 12 represents the empirical distribution of the *Apple* time series data vs SBTS, while Figure 13 shows their covariance matrices.

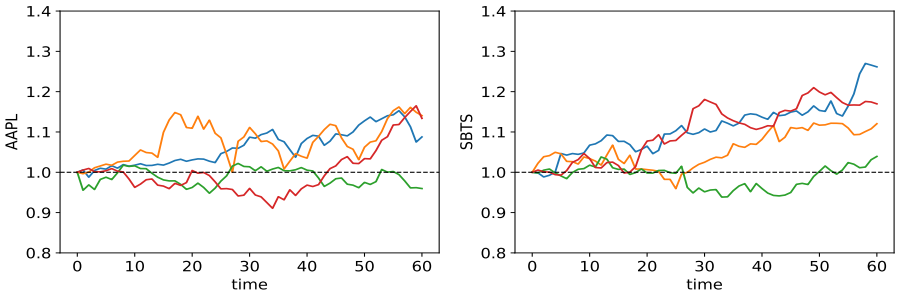

Figure 10: Four paths generated by Schrodinger bridge(*Right*) vs real ones (*Left*)

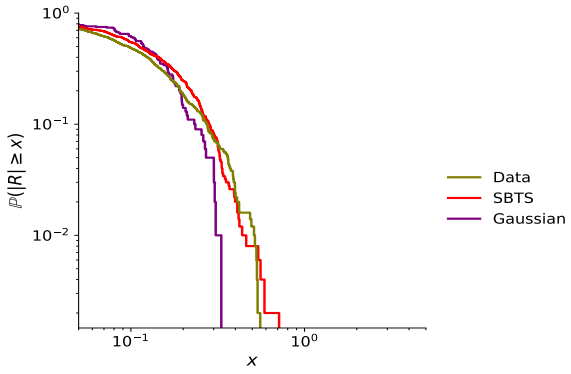

Figure 11: Plot of tail distribution for the return: x in log-scale $\mapsto \mathbb{P}[|R| \geq x]$. Excess of kurtosis for real-data = 1.96, for generated SBTS = 2.34

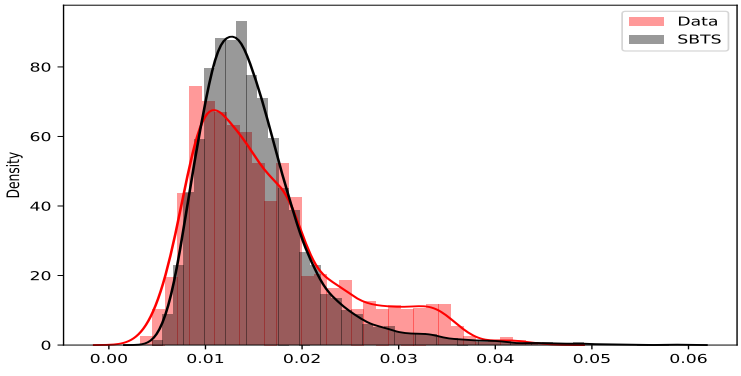

Figure 12: Comparison of quadratic variation distribution

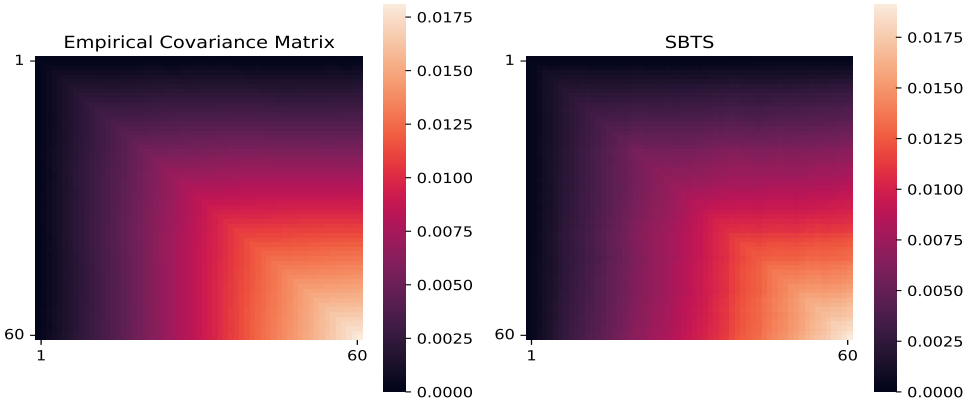

Figure 13: Covariance matrix for real-data and generative SBTS

The synthetic time series generated by SBTS is now used for the deep hedging of ATM call option $g(S_T) = (S_T - S_0)_+$, i.e., by minimizing over the initial capital p (premium) and the parameters of the neural network Δ the (empirical) loss function, called replication error:

$$\mathbb{E}\big|\mathrm{PnL}^{\mathrm{p},\Delta}\big|^2, \quad \text{with} \quad \mathrm{PnL}^{\mathrm{p},\Delta} \;=\; p + \sum_{i=0}^{N-1} \Delta(t_i, S_{t_i})(S_{t_{i+1}} - S_{t_i}) \;-\; g(S_T).$$

We then compare with the deep hedging on historical data by looking at the PnL and replication errors. The historical data set of Apple is split in the chronological order, namely training data set from 01/01/2007 to 31/12/2017, validation data set from 01/01/2018 to 31/12/2028, and test set from 01/01/2019 to 30/01/2020. As pointed out in Wang and Ruf (2022), it is important not to break the time structure as it may lead to an overestimation of the model performance.

In Figure 14, we plot the empirical distribution of the PnL with deep hedging obtained from real data vs SBTS, and backtested on the validation and test sets. It appears that the PnL from SBTS has a smaller variance (hence smaller replication error), and yields less extreme values, i.e. outside the zero value, than the PnL from real data. This is also quantified in Table 4 where we note that the premium obtained from SBST is higher than the one from real data, which means that one is more conservative with SBTS by charging a higher premium.

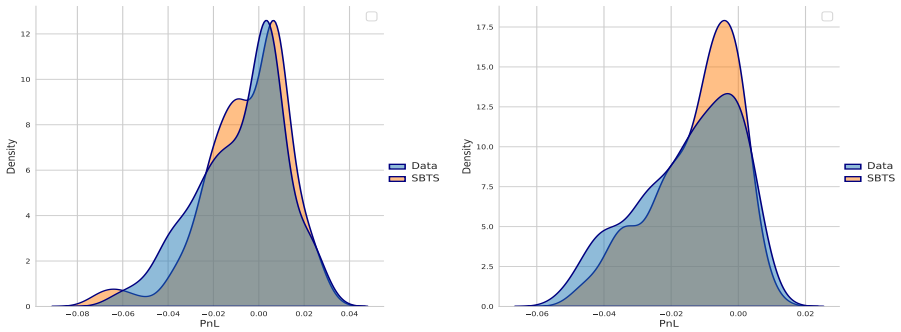

Figure 14: Deep hedging PnL distribution with backtest from validation set (*left*) and test set (*right*).

		Training Set		Validation Set		Test Set	
	Premium	Mean	Std	Mean	Std	Mean	Std
Data	0.0488	0.000165	0.011	-0.00754	0.0193	-0.015	0.015
SBTS	0.0506	0.0023	0.011	-0.0051	0.0186	-0.012	0.013

Table 4: Mean of PnL and its Std (replication error).

6. Conclusion

We proposed a novel generative model based on the dynamics of the Schrödinger bridge formulation that captures the temporal structure of time series. We use a nonparametric kernel estimator of the drift, offering an interpretable and computationally efficient alternative to neural network-based models.

In a companion study Alouadi et al. (2025), we further evaluate our method (SBTS) on diverse synthetic and real-world datasets, benchmarking it against a broad range of baselines including GANs, COT-GAN, and diffusion-based models like TSGM. We also conduct robustness checks (e.g., bandwidth and Markovian order sensitivity), parameter recovery experiments, and runtime comparisons. These results confirm the practical effectiveness of SBTS, particularly for low-to moderately high-dimensional time series (e.g., datasets with 6 to 28 features such as Google stock, UCI Energy, and Air Quality data).

While kernel methods may suffer in very high-dimensional settings due to the curse of dimensionality, we address this by proposing an alternative drift learning strategy based on LSTM networks (Section 4.1), better suited to such contexts. Overall, SBTS offers a compelling, flexible, and efficient solution for generative modeling of time series data.

References

A. Alouadi, B. Barreau, L. Carlier, and H. Pham. Robust time series generation via Schrödinger bridge: a comprehensive evaluation. In *ICAIF '25: Proceedings of the 6th ACM International Conference on AI in Finance*, pages 906–914, 2025.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.

H. Buehler, B. Horvath, T. Lyons, I. Perez-Arribas, and B. Wood. A data-driven market simulator for small data environments. *SSRN 3632431*, 2020.

T. Chen, G-H. Liu, M. Tao, and E. Theodorou. Deep momentum multi-marginal Schrödinger bridge. In *NeurIPS2025*, 2025.

Y. Chen, Y. Wang, D. Kirschen, and B. Zhang. Model-free renewable scenario generation using generative adversarial networks. *IEEE Transactions on power systems*, 33(3): 3265–3275, 2018.

L. Chizat, S. Zhang, M. Heitz, and G. Schiebinger. Trajectory inference via mean-field Langevin in path space. *Advances in Neural Information Processing Systems*, 35:16731–16742, 2022.

P. Dai Pra. A stochastic control approach to reciprocal diffusion process. *Applied Mathematics and Optimization*, 23(1):313–329, 1991.

V. De Bortoli, J. Thornton, J. Heng, and A. Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *NeurIPS2021*, 2021.

A. Fermanian. Embedding and learning with signatures. *arXiv:1911.13211*, 2019.

J. Gairing, P. Imkeller, R. Shevchenko, and C. Tudor. Hurst index estimation in stochastic differential equations driven by fractional brownian motion. *Journal of Theoretical Probability*, 33:1691–1714, 2020.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Faley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

F. Guth, S. Coste, V. De Bortoli, and S. Mallat. Wawelet score-based generative modeling. *arXiv: 2208.05003*, 2022.

W. Härdle, M. Müller, S. Sperlich, and A. Werwatz. *Nonparametric and semiparametric models*. Springer, 2004.

P. Henry-Labordere. From (martingale) Schrodinger bridges to a new class of stochastic volatility model. *SSRN.3353270*, 2019.

P. Kidger, J. Foster, X. Li, H. Oberhauser, and T. Lyons. Neural SDEs as infinite-dimensional GANs. In *Proceedings of ICML*, 2021.

D. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.

Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based modeling. *Predicting structured data*, 1, 2006.

C. Leonard. A survey of the schrodinger problem and some of its connections with optimal transport. *Dynamical Systems*, 34(4):1533–1574, 2014.

X. Lyu, S. Hueser, S.L. Hyland, G. Zerveas, and G. Raetsch. Improving clinical predictions through unsupervised time series representation learning. *arXiv:1812.00490*, 2018.

H. Ni, L. Szpruch, M. Wiese, S. Liao, and B. Xiao. Conditional Sig-Wasserstein GANs for time series generation. *arXiv:2006.05421*, 2020.

V. Ram Somnath, M. Pariset, Y-P. Hsieh, M.R. Martinez, A. Krause, and C. Bunne. Aligned diffusion Schrödinger bridges. *Arxiv:2302.11419*, 2023.

C. Remlinger, J. Mikael, and R. Elie. Conditional versus adversarial Euler-based generators for time series. *arXiv:2102.05313*, 2021.

D. Scott. *Multivariate density estimation*. Probability and statistics. Wiley, 2014.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the distribution. In *NIPS*, pages 11918–11930, 2019.

Y. Song, J. Sohl-Dickstein, D. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on learning Representation*, 2021.

M.P. Wand and M.C. Jones. *Kernel smoothing*, volume 60 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, 1995.

G. Wang, Y. Jiao, Q. Xu, Y. Wang, and C. Yang. Deep generative learning via schrödinger bridge. *Proceedings of Machine Learning Research 2021*, 2021.

W. Wang and J. Ruf. A note on spurious model selection. *Quantitative Finance*, 22(10): 1797–2000, 2022.

M. Wiese, R. Knobloch, R. Korn, and Kretschmer. Quant gans: deep generation of financial time series. *Quantitative Finance*, 20(9):1419–1440, 2020.

T. Xu, W. Li, M. Munn, and B. Acciaio. COT-GAN: Generative Sequential Data via Causal Optimal Transport. In *NeurIPS*, 2020.

J. Yoon, D. Jarrett, and Van der Schaar. Time-series generative adversarial networks. In *NeurIPS*, 2019.