

Neural Stochastic Volterra Equations: Learning Path-Dependent Dynamics

Martin Bergerhausen *¹, David J. Prömel †¹, and David Scheffels ‡¹

¹University of Mannheim, Mannheim 68161, Germany.

Abstract. Stochastic Volterra equations (SVEs) serve as mathematical models for the time evolutions of random systems with memory effects and irregular behaviour. We introduce neural stochastic Volterra equations as a physics-inspired architecture, generalizing the class of neural stochastic differential equations, and provide some theoretical foundation. Numerical experiments on various SVEs, like the disturbed pendulum equation, the generalized Ornstein-Uhlenbeck process, the rough Heston model and a monetary reserve dynamics, are presented, comparing the performance of neural SVEs, neural stochastic differential equations (SDEs) and Deep Operator Networks (DeepONets).

Keywords:

Feedforward neural network,
Deep operator network,
Neural stochastic differential equation,
Stochastic Volterra equation,
Supervised learning.

Article Info.:

Volume: 4
Number: 4
Pages: 264 - 289
Date: December/2025
doi.org/10.4208/jml.240926

Article History:

Received: 26/09/2024
Accepted: 31/08/2025

Communicated by:

Zhi-Qin Xu

1 Introduction

Stochastic Volterra equations are used as mathematical models for the time evolutions of random systems appearing in various areas like biology, finance or physics. SVEs are a natural generalization of ordinary stochastic differential equations and, in contrast to SDEs, they are capable to represent random dynamics with memory effects and very irregular trajectories. For instance, SVEs are used in the modelling of turbulence [2], of volatility on financial markets [12] and of DNA patterns [36].

Combining differential equations and neural networks into hybrid approaches for statistical learning has been gaining increasing interest in recent years, see e.g. [5,11]. This has led to many very successful data-driven methods to learn solutions of various differential equations. For instance, neural stochastic differential equations are SDEs with coefficients parametrized by neural networks, and serve as continuous-time generative models for irregular time series, see [15, 19, 26, 27]. Models based on neural SDEs are of particular interest in financial engineering, see [7, 9, 13]. Further examples of “neural” differential equations are neural controlled differential equations [21], which led to very successful

*martin.bergerhausen@uni-mannheim.de

†Corresponding author. proemel@uni-mannheim.de

‡david.scheffels@lessing-ffm.net

methods for irregular time series, neural rough differential equations [30], which are especially well-suited for long time series, and neural stochastic partial differential equations [37], which are capable to process data from continuous spatiotemporal dynamics. Loosely speaking, “neural” differential equations and their variants can be considered as continuous-time analogous to various recurrent neural networks.

In the present work, we introduce neural stochastic Volterra equations as stochastic Volterra equations with coefficients parameterized by neural networks. They constitute a natural generalization of neural SDEs with the advantage that they are capable to represent time series with temporal dependency structures, which overcomes a limitation faced by neural SDEs. Hence, neural SVEs are suitable to serve as generative models for random dynamics with memory effects and irregular behaviour, even more irregular than neural SDEs. As theoretical justification for the universality of neural SVEs, we provide a stability result for general SVEs in Proposition 2.1, which can be combined with classical universal approximation theorems for neural networks [10, 14, 21, 25]; cf. Remark 2.1.

Relying on neural stochastic Volterra equations parameterized by feedforward neural networks, we study supervised learning problems for random Volterra type dynamics. More precisely, we consider setups, where the training sets consist of sample paths of the “true” Volterra process together with the associated realizations of the driving noise and the initial condition, and build a neural SVEs based model aiming to reproduce the sample paths as good as possible. A related supervised learning problem in the context of stochastic partial differential equations (SPDEs) was treated in [37] introducing neural SPDEs. For unsupervised learning problems using neural SDEs we refer to [18].

We numerically investigate the supervised learning problem for prototypical Volterra type dynamics such as the disturbed pendulum equation [31], the rough Heston model [12], the generalized Ornstein-Uhlenbeck process [40] and a model for the dynamics of monetary reserves [3]. The performance of the neural SVE based models is compared to Deep Operator Networks and to neural SDEs. Recall DeepONets are a popular class of neural learning algorithms for general operators on function spaces that were introduced in [28]. For the training process of the neural SVE we choose the Adam algorithm, as introduced in [22], which is known to be a well-suited stochastic gradient descent method for stochastic optimization problem.

The numerical study in Section 3 demonstrates that the presented neural SVE based methods significantly outperform DeepONets; see Tables 3.1-3.3. In particular, neural SVE based methods generalize much more effectively, as evidenced by their strong performance on the test sets – neural SVEs are up to 20 times more accurate than DeepONets. Moreover, neural SVEs also outperform neural SDE based models for random dynamics with dependency structures; cf. Section 3.5. These observations highlight the advantages of the physics-informed architecture of neural SVEs for supervised learning problems involving random systems with Volterra-type dynamics.

Organization of the paper. In Section 2 we introduce neural stochastic Volterra equations and their theoretical background. The numerical experiments are presented in Section 3. In Appendix A we present the postponed proofs regarding the stability of stochastic Volterra equations.

2 Neural stochastic Volterra equations

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in [0, T]}, \mathbb{P})$ be a filtered probability space, which satisfies the usual conditions, $T \in (0, \infty)$ and $d, m \in \mathbb{N}$. Given an \mathbb{R}^d -valued random initial condition ξ and an m -dimensional standard Brownian motion $(B_t)_{t \in [0, T]}$, we consider the d -dimensional stochastic Volterra equation

$$X_t = \xi g(t) + \int_0^t K_\mu(t-s)\mu(s, X_s) ds + \int_0^t K_\sigma(t-s)\sigma(s, X_s) dB_s, \quad t \in [0, T], \quad (2.1)$$

where $g: [0, T] \rightarrow \mathbb{R}$ is a deterministic continuous function (where we usually normalize $g(0) = 1$), the coefficients $\mu: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $\sigma: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m}$, and the convolutional kernels $K_\mu, K_\sigma: [0, T] \rightarrow \mathbb{R}$ are measurable functions. Furthermore, $\int_0^t K_\sigma(t-s)\sigma(s, X_s) dB_s$ is defined as an Itô integral. We refer to [17, 31] for introductory textbooks on stochastic integration and to [6, 8, 32] for classical results on SVEs.

To define the notion of a (strong) L^p -solution, let $L^p(\Omega \times [0, T])$ be the space of all real-valued, p -integrable functions on $\Omega \times [0, T]$. We call an $(\mathcal{F}_t)_{t \in [0, T]}$ -progressively measurable stochastic process $(X_t)_{t \in [0, T]}$ in $L^p(\Omega \times [0, T])$, on the given probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \in [0, T]}, \mathbb{P})$, a (strong) L^p -solution of the SVE (2.1) if $\int_0^t (|K_\mu(t-s)\mu(s, X_s)| + |K_\sigma(t-s)\sigma(s, X_s)|^2) ds < \infty$ for all $t \in [0, T]$ and the integral equation (2.1) hold \mathbb{P} -almost surely. As usual, a strong L^1 -solution $(X_t)_{t \in [0, T]}$ of the SVE (2.1) is often just called solution of the SVE (2.1).

2.1 Neural SVEs

To learn the dynamics of the SVE (2.1), that is, the corresponding operators $\xi, g, K_\mu, K_\sigma, \mu$ and σ , we rely on some neural network architecture. To that end, let for some latent dimension $d_h > d$,

$$\begin{aligned} L_\theta: \mathbb{R}^d &\rightarrow \mathbb{R}^{d_h}, & g_\theta: [0, T] &\rightarrow \mathbb{R}, & K_{\mu, \theta}: [0, T] &\rightarrow \mathbb{R}, & K_{\sigma, \theta}: [0, T] &\rightarrow \mathbb{R}, \\ \mu_\theta: [0, T] \times \mathbb{R}^{d_h} &\rightarrow \mathbb{R}^{d_h}, & \sigma_\theta: [0, T] \times \mathbb{R}^{d_h} &\rightarrow \mathbb{R}^{d_h \times m}, & \Pi_\theta: \mathbb{R}^{d_h} &\rightarrow \mathbb{R}^d \end{aligned}$$

be seven feedforward neural networks (see [42, Section 3.6.1]) that are parameterized by some common parameter θ . Note that L_θ lifts the given initial value to the latent space \mathbb{R}^{d_h} , Π_θ is the readout back from the latent space to the space \mathbb{R}^d , and the other networks try to imitate their respective in Eq. (2.1) on the latent d_h -dimensional space.

Given the input data $\xi \in \mathbb{R}^d$ and $(B_t)_{t \in [0, T]} \in C([0, T]; \mathbb{R}^m)$, \mathbb{P} -a.s., we introduce the neural stochastic Volterra equations

$$\begin{aligned} Z_0 &= L_\theta(\xi), \\ Z_t &= Z_0 g_\theta(t) + \int_0^t K_{\mu, \theta}(t-s)\mu_\theta(s, Z_s) ds + \int_0^t K_{\sigma, \theta}(t-s)\sigma_\theta(s, Z_s) dB_s, \\ X_t &= \Pi_\theta(Z_t), \quad t \in [0, T]. \end{aligned} \quad (2.2)$$

The objective is to optimize θ as good as possible such that the generated paths are as close as possible to the given training paths. Therefore, one needs to solve a stochastic optimization problem at each training step. One typically chosen and well-suited stochastic gradient descent method for stochastic optimization problems is the Adam algorithm, introduced in [22]. The Adam algorithm is known to be computationally efficient, requires little memory, is invariant to diagonal rescaling of gradients and is well-suited for high-dimensional problems with regard to data/parameters.

Given a trained supervised model $(L_\theta, K_{\mu,\theta}, K_{\sigma,\theta}, \mu_\theta, \sigma_\theta, \Pi_\theta)$, we can evaluate the neural SVE (2.2) given the input data (ζ, B) by using any numerical scheme for stochastic Volterra equations. For that purpose, we use the Volterra Euler-Maruyama scheme introduced in [43] for the training procedure. Note that Lipschitz conditions on μ_θ and σ_θ can be imposed by using, e.g., LipSwish, ReLU or tanh activation functions.

2.2 Neural network architecture

The structure of the neural SVE model (2.2) is analogously defined to the structure of neural stochastic differential equations, as introduced in [18], and of neural stochastic partial differential equations, as introduced in [37]. The d_h -dimensional process Z represents the hidden state. We impose the readout Π_θ to get back to dimension d . The model has, at least if one considers a setting where the initial condition cannot be observed like an unsupervised setting, some minimal amount of architecture. It is in such a setting necessary to induce the lift L_θ and the randomness by some additional variable $\tilde{\zeta}$ to learn the randomness induced by the initial condition $X_0 = \Pi_\theta(L_\theta(\tilde{\zeta})g_\theta(0))$ (otherwise X_0 would not be random since it does not depend on the Brownian motion B). Moreover, the structure induced by the lift L_θ and the readout Π_θ is the natural choice to lift the d -dimensional SVE (2.1) to the latent dimension $d_h > d$.

We use LipSwish activation functions in any layer of any network. These were introduced in [4] as $\rho(z) = 0.909z\sigma(z)$, where σ is the sigmoid function. Due to the constant 0.909, LipSwish activations are Lipschitz continuous with Lipschitz constant one and smooth. Moreover, there is strong empirical evidence that LipSwish activations are very suitable for a variety of challenging approximation tasks, see [35].

For a given latent dimension $d_h > d$, the lift L_θ is modeled as a linear 1-layer network from dimension d to d_h without any additional hidden layer, and, as its counterpart, the readout Π_θ as a linear 1-layer network from d_h to d . The networks $K_{\mu,\theta}$, $K_{\sigma,\theta}$ and g_θ are all designed as linear networks from dimension 1 to 1 with two hidden layers of size d_K for some additional dimension $d_K > d$. Lastly, the network μ_θ is defined as a linear network from dimension $1 + d_h$ to d_h with one hidden layer of size d_h and the network σ_θ from $1 + d_h$ to $d_h \cdot m$ with one hidden layer of size $d_h \cdot m$.

2.3 Stability for SVEs

The mathematical reason that neural stochastic Volterra equations provide a suitable structure for learning the dynamics of general SVEs is the universal approximation property of neural networks, see e.g. [10, 14, 20, 25], and the stability result for SVEs, presented in this

subsection. More precisely, our stability result yields that if we approximate the kernels and coefficients of an SVE sufficiently well, we get a good approximation of the solution by the respective approximating solutions. To formulate the stability result, we need the following definitions and assumptions.

For $p \geq 1$, the L^p -norm of a function $h: [0, T] \rightarrow \mathbb{R}$ is defined by

$$\|h\|_p := \left(\int_0^T |h(s)|^p ds \right)^{\frac{1}{p}},$$

and the sup-norms for functions $f: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $g: [0, T] \rightarrow \mathbb{R}^d$, respectively, are given by

$$\|f\|_\infty := \sup_{t \in [0, T], x \in \mathbb{R}^d} |f(t, x)|, \quad \|g\|_\infty := \sup_{t \in [0, T]} |g(t)|.$$

As approximation of the SVE (2.1), we consider a sequence of SVEs, given by

$$X_t^n = \xi g_n(t) + \int_0^t K_{\mu,n}(t-s) \mu_n(s, X_s^n) ds + \int_0^t K_{\sigma,n}(t-s) \sigma_n(s, X_s^n) dB_s, \quad (2.3)$$

for $t \in [0, T]$ and $n \in \mathbb{N}$. We make the following assumptions on the kernels $K_{\mu,n}, K_{\sigma,n}$, the coefficients μ_n, σ_n and the initial conditions g_n .

Assumption 2.1. The initial conditions $g, g_n: [0, T] \rightarrow \mathbb{R}^d$ and kernels $K_\mu, K_\sigma: [0, T] \rightarrow \mathbb{R}$ and $K_{\mu,n}, K_{\sigma,n}: [0, T] \rightarrow \mathbb{R}$ for $n \in \mathbb{N}$ satisfy the following conditions: There are constants $\gamma \in (0, 1/2], \varepsilon > 0$ and $L > 0$ such that

(i) For all $n \in \mathbb{N}$ the measurable functions $K_{\mu,n}, K_{\sigma,n}: [0, T] \rightarrow [0, \infty)$ fulfill

$$\begin{aligned} \int_0^{T-h} |K_{\mu,n}(h+r) - K_{\mu,n}(r)|^{1+\varepsilon} dr + \int_0^h |K_{\mu,n}(r)|^{1+\varepsilon} dr &\leq Lh^{\gamma(1+\varepsilon)}, \\ \int_0^{T-h} |K_{\sigma,n}(h+r) - K_{\sigma,n}(r)|^{2+\varepsilon} dr + \int_0^h |K_{\sigma,n}(r)|^{2+\varepsilon} dr &\leq Lh^{\gamma(2+\varepsilon)} \end{aligned}$$

for all $h \in [0, T]$.

(ii) It holds that

$$\begin{aligned} \int_0^T |K_{b,n}(s) - K_b(s)| ds &\rightarrow 0 \quad \text{as } n \rightarrow \infty, \\ \int_0^T |K_{\sigma,n}(s) - K_\sigma(s)|^{2+\varepsilon} ds &\rightarrow 0 \quad \text{as } n \rightarrow \infty. \end{aligned}$$

(iii) g and g_n are γ -Hölder-continuous.

Assumption 2.2. Let $\mu, \mu_n: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $\sigma, \sigma_n: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m}, n \in \mathbb{N}$, be continuous functions such that:

- (i) μ, σ and μ_n, σ_n are (uniformly) of linear growth, i.e. there is a constant $C_{\mu, \sigma} > 0$ such that

$$\sup_{n \in \mathbb{N}} |\mu_n(t, x)| + |\sigma_n(t, x)| + |\mu(t, x)| + |\sigma(t, x)| \leq C_{\mu, \sigma}(1 + |x|)$$

for all $t \in [0, T]$ and $x \in \mathbb{R}^d$.

- (ii) For any compact subset $K \subset \mathbb{R}^d$ we have

$$\lim_{n \rightarrow \infty} \sup_{t \in [0, T]} \sup_{x \in K} |\sigma_n(t, x) - \sigma(t, x)| + |\mu_n(t, x) - \mu(t, x)| + |g_n(t) - g(t)| = 0.$$

Based on the aforementioned assumptions, we obtain the following stability result for stochastic Volterra equations, generalizing the classical stability result for ordinary stochastic differential equations proven in [16].

Theorem 2.1. *Suppose Assumptions 2.1 and 2.2 and let $\xi \in L^p(\Omega)$ with $p > \max\{1/\gamma, (4 + 2\varepsilon)/\varepsilon\}$. Moreover, suppose that there are unique L^p -solutions $(X_t)_{t \in [0, T]}$ and $(X_t^n)_{t \in [0, T]}$ to the SVEs (2.1) and (2.3), for $n \in \mathbb{N}$, respectively. Then, one has*

$$\lim_{n \rightarrow \infty} \mathbb{E} \left[\sup_{t \in [0, T]} |X_t^n - X_t|^2 \right] = 0.$$

Proof. See Appendix A. □

Note that the assumption on the existence of unique solutions can be ensured by postulating the coefficients to be Lipschitz continuous, see e.g. [41]. However, for instance, in a one-dimensional setting a unique solution can also be obtained for Hölder continuous diffusion coefficients, see e.g. [1, 34].

Assuming that the coefficients of a SVE are Lipschitz continuous, one can quantify the stability result of Theorem 2.1, as we shall present below. To that end, we consider, as comparison to the SVE (2.1), the SVE

$$\tilde{X}_t = \xi \tilde{g}(t) + \int_0^t \tilde{K}_\mu(t-s) \tilde{\mu}(s, \tilde{X}_s) ds + \int_0^t \tilde{K}_\sigma(t-s) \tilde{\sigma}(s, \tilde{X}_s) dB_s, \quad t \in [0, T], \quad (2.4)$$

where $\tilde{g}: [0, T] \rightarrow \mathbb{R}$ is a continuous function, and where the coefficients $\tilde{\mu}: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $\tilde{\sigma}: [0, T] \times \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m}$, and the convolutional kernels $\tilde{K}_\mu, \tilde{K}_\sigma: [0, T] \rightarrow \mathbb{R}$ are measurable functions.

For the convolutional kernels, we make the following assumption.

Assumption 2.3. Let $q, \tilde{q} > 1$ and $p \geq 2$ be such that

$$\frac{1}{p} + \frac{1}{q} = 1, \quad \frac{2}{p} + \frac{1}{\tilde{q}} = 1. \quad (2.5)$$

Suppose that $\|K_\mu\|_q + \|\tilde{K}_\mu\|_q < \infty, \|K_\sigma\|_{2\tilde{q}} + \|\tilde{K}_\sigma\|_{2\tilde{q}} < \infty$, and $\|g\|_\infty + \|\tilde{g}\|_\infty < \infty$.

For the coefficients, we require the standard Lipschitz and linear growth conditions.

Assumption 2.4. Let $\mu, \tilde{\mu}: [0, T] \times \mathbb{R} \rightarrow \mathbb{R}^d$ and $\sigma, \tilde{\sigma}: [0, T] \times \mathbb{R} \rightarrow \mathbb{R}^{d \times m}$ be measurable functions such that:

- (i) μ, σ and $\tilde{\mu}, \tilde{\sigma}$ are of linear growth, i.e. there is a constant $C_{\mu, \sigma} > 0$ such that

$$|\tilde{\mu}(t, x)| + |\tilde{\sigma}(t, x)| + |\mu(t, x)| + |\sigma(t, x)| \leq C_{\mu, \sigma}(1 + |x|)$$

for all $t \in [0, T]$ and $x \in \mathbb{R}^d$.

- (ii) μ, σ and $\tilde{\mu}, \tilde{\sigma}$ are Lipschitz continuous in the space variable uniformly in time, i.e. there is a constant $C_{\mu, \sigma} > 0$ such that

$$|\mu(t, x) - \mu(t, y)| + |\sigma(t, x) - \sigma(t, y)| \leq C_{\mu, \sigma}|x - y|,$$

$$|\tilde{\mu}(t, x) - \tilde{\mu}(t, y)| + |\tilde{\sigma}(t, x) - \tilde{\sigma}(t, y)| \leq C_{\mu, \sigma}|x - y|$$

holds for all $t \in [0, T]$ and $x, y \in \mathbb{R}^d$.

Based on these assumptions, we obtain the following stability result for stochastic Volterra equations with Lipschitz continuous coefficients. For related stability results in the context ordinary stochastic differential equations, we refer to [25, Section 3] and the references therein.

Proposition 2.1. *Suppose Assumptions 2.3, 2.4 and assume $\xi \in L^p(\Omega)$. Let $(X_t)_{t \in [0, T]}$ and $(\tilde{X}_t)_{t \in [0, T]}$ be the solutions to the SVEs (2.1) and (2.4), respectively. Then, there is some constant $C > 0$, depending on $\mu, \sigma, \tilde{\mu}, \tilde{\sigma}, K_\mu, K_\sigma, \tilde{K}_\mu, \tilde{K}_\sigma, p, \xi$ such that*

$$\sup_{t \in [0, T]} \mathbb{E}[|X_t - \tilde{X}_t|^p] \leq C \left(\|g - \tilde{g}\|_\infty^p + \|\mu - \tilde{\mu}\|_\infty^p + \|\sigma - \tilde{\sigma}\|_\infty^p + \|K_\mu - \tilde{K}_\mu\|_q^p + \|K_\sigma - \tilde{K}_\sigma\|_{2q}^p \right). \quad (2.6)$$

Proof. See Appendix A. □

Remark 2.1. The stability results, presented in Theorem 2.1 and Proposition 2.1, demonstrate the universality of neural stochastic Volterra equations like (2.2). Indeed, the unique solution of any general stochastic Volterra equations can be approximated arbitrary well by solutions of neural stochastic Volterra equations, assuming that the associated neural network converges in a suitable sense. For the coefficients the required suitable convergence is the local uniform convergence subject to a uniform global linear growth constraint. As shown in [25], many frequently used classes of neural networks do allow for such convergence, in particular, neural networks based on LipSwish activation functions, as we use in the numerical experiments below. For the kernels, the suitable type of convergence is an L^p -convergence on the compact interval $[0, T]$. L^p -type universal approximation theorem can already be found in the classical work of [14]; we also refer to [20] and the references therein.

3 Numerical experiments

In this section, we numerically investigate the supervised learning problem utilizing neural stochastic Volterra equations aiming to learn Volterra type dynamics such as the disturbed pendulum equation, the generalized Ornstein-Uhlenbeck process, a model for the dynamics of monetary reserves, and the rough Heston model. The performance is compared to Deep Operator Networks and neural stochastic differential equations. For all the neural SVEs, we chose the latent dimensions $d_h = d_K = 12$ which experimentally proved to be well-suited. We consider the interval $[0, T]$ for $T = 5$ and discretize it equally-sized using the grid size $\Delta t = 0.1$.

As a benchmark model, we use the Deep Operator Network (DeepONet) algorithm. DeepONet is a popular class of neural learning algorithms for general operators on function spaces that was introduced in [28]. A DeepONet consists of two neural networks: the branch network which operates on the function space $C([0, T]; \mathbb{R}^n)$ (where $[0, T]$ is represented by some fixed discretization), and the so-called trunk network which operates on the evaluation point $t \in [0, T]$. Then, the output of the DeepONet is defined as

$$\text{DeepONet}(f)(t) = \sum_{k=1}^p b_k t_k + b_0,$$

where $(b_k)_{k=1, \dots, p}$ is the output of the branch network operating on the discretization of $f \in C([0, T]; \mathbb{R}^n)$, $(t_k)_{k=1, \dots, p}$ is the output of the trunk network operating on $t \in [0, T]$ and $p \in \mathbb{N}$ is the dimension of the output of both networks. Following [28], we model both networks as feedforward networks. We perform a grid search to optimally determine the depth and width of both networks such as the activation functions, optimizer and learning rate.

We perform experiments on a one-dimensional disturbed pendulum equation, a one- and a two-dimensional Ornstein-Uhlenbeck equation as well as a one-dimensional rough Heston equation. We perform the experiments on low-, mid- and high-data regimes with $n = 100, n = 500$ and $n = 2000$, and use 80% of the data for training and 20% for testing. We compare the results of neural SVEs to those of DeepONet and consider for both algorithms the mean relative L^2 -loss. All experiments are trained for an appropriate number epochs of iterations until there is no improvement anymore. For neural SVEs, we use the Adam stochastic optimization algorithm, which heuristically proved to be well-suited, with learning rate 0.01 and scale the learning rate by a factor 0.8 after every 25% of epochs.

Note that since DeepONet is not able to deal with random initial conditions, we use deterministic initial conditions $\xi = 2$ in the DeepONet experiments. For neural SVEs we use initial conditions $\xi \sim \mathcal{N}(2, 0.2)$ unless stated otherwise.

Remark 3.1. The results in this section show that neural SVEs are able to outperform DeepONet significantly (see Tables 3.1-3.3). Especially, neural SVEs generalize much better which can be seen in the good performance on the test sets where neural SVEs are up to 20 times better than DeepONet. This can be explained by the explicit structure of the Volterra equation that is already part of the model for neural SVEs.

All the code is published on https://github.com/davidscheffels/Neural_SVEs.

3.1 Disturbed pendulum equation

As first example, we study the disturbed pendulum equation resulting from Newton’s second law. Recall, general second-order differential systems (without first-order terms) perturbed by a multiplicative noise are given by

$$y''(t) = \mu(t, y(t)) + \sigma(t, y(t)) \dot{B}_t, \quad t \in [0, T],$$

where $\dot{B}_t = dB_t/dt$ is white noise for some standard Brownian motion $(B_t)_{t \in [0, T]}$. Using the deterministic and the stochastic Fubini theorem, this system can be rewritten as stochastic Volterra equations

$$y(t) = y(0) + t \cdot y'(0) + \int_0^t (t - u) \mu(u, y(u)) du + \int_0^t (t - u) \sigma(u, y(u)) dB_u.$$

A concrete example from physics is the disturbed pendulum equation (see [31, Exercise 5.12]) resulting from Newton’s second law, see e.g. [23, Section 2.4], which describes the motion of an object X with deterministic initial value x_0 under some force F , can be described by the differential equation

$$m \frac{d^2 X(t)}{dt^2} = F(X(t)), \quad X(0) = x_0.$$

Hence, $(X_t)_{t \in [0, T]}$ solves the SVE

$$X(t) = x_0 + tX'(0) + \int_0^t (t - s) \frac{F(X(s))}{m} ds + \int_0^t (t - s) \frac{\varepsilon X_s}{m} dB_s.$$

As prototyping example, we consider the one-dimensional equation

$$y_t = \zeta - \int_0^t (t - s) y_s ds + \int_0^t (t - s) y_s dB_s, \quad t \in [0, T] \tag{3.1}$$

with the target to learn its dynamics by neural SVEs and DeepONet. The results are presented in Table 3.1.

Sample paths of the training and the testing sets together with their learned approximations are shown in Fig. 3.1. It is clearly visible that while DeepONet is not able to generalize properly to the testing set, the learned neural SVE paths are very close to the true paths also for the test set.

Table 3.1: Mean relative L^2 -losses after training for the disturbed pendulum equation (3.1).

Neural SVE	Train set	Test set	DeepONet	Train set	Test set
$n = 100$	0.01	0.013	$n = 100$	0.003	0.2
$n = 500$	0.008	0.008	$n = 500$	0.003	0.06
$n = 2000$	0.006	0.006	$n = 2000$	0.003	0.02

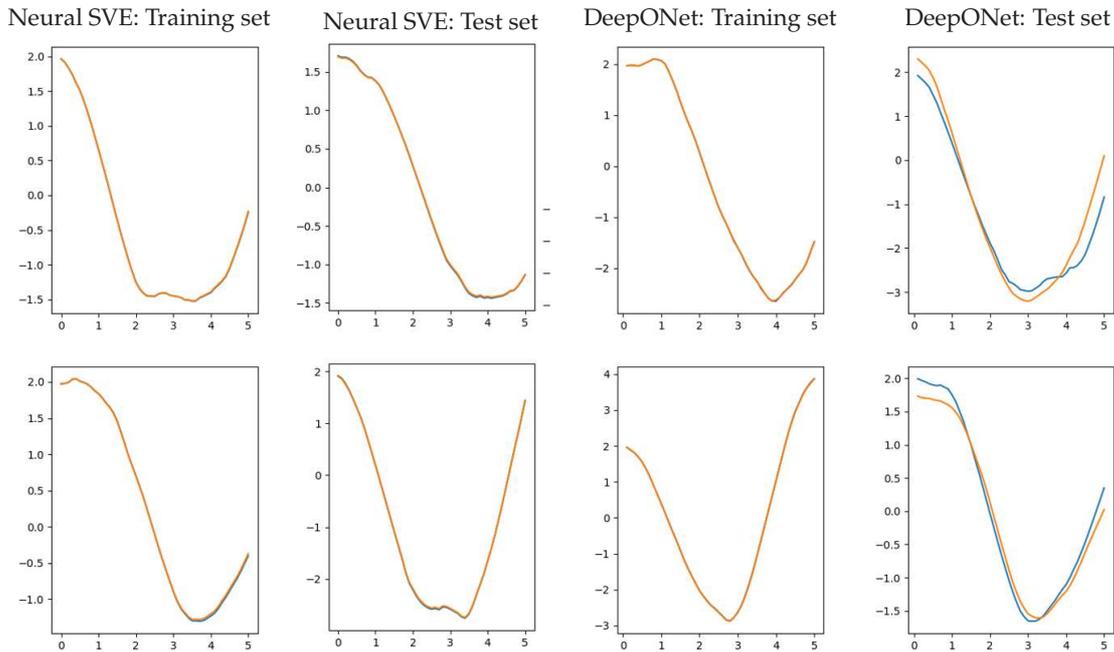


Figure 3.1: Sample neural SVE and DeepONet paths from the training and the test set for the disturbed pendulum equation and $n = 100$. Blue (barely visible) are the original paths and orange the learned approximations.

To highlight the performance of neural SVE in a more complex setting, we take a non-linear coefficient μ and consider

$$y_t = \xi - \int_0^t (t - s) \sin(y_s) ds + \int_0^t 0.4(t - s)y_s dB_s, \quad t \in [0, T]. \tag{3.2}$$

The results are presented in Table 3.2. It stands out immediately that for $n = 100$ the performance in the test set is far worse than in the training set. Most solutions remain in the range of $[-\pi, \pi]$. Yet, in some rare cases, solutions may explode outside this range. An example of this can be seen in Fig. 3.2. In the training set this happens too rarely for the neural network to learn the functions outside of $[-\pi, \pi]$ precisely. Table 3.2 clearly shows that this discrepancy disappears when the training set is sufficiently large. In the case of $n = 2000$ one also sees that the neural network can learn these complex functions as good as in the prior example. Then the plots for the test set look similar to the training set plots of the neural SVE trained with $n = 100$ trajectories.

Table 3.2: Mean relative L^2 -losses after training for the nonlinear disturbed pendulum equation (3.2).

Neural SVE	Train set	Test set	DeepONet	Train set	Test set
$n = 100$	0.007	0.028	$n = 100$	0.004	0.23
$n = 500$	0.012	0.014	$n = 500$	0.006	0.13
$n = 2000$	0.007	0.006	$n = 2000$	0.004	0.06

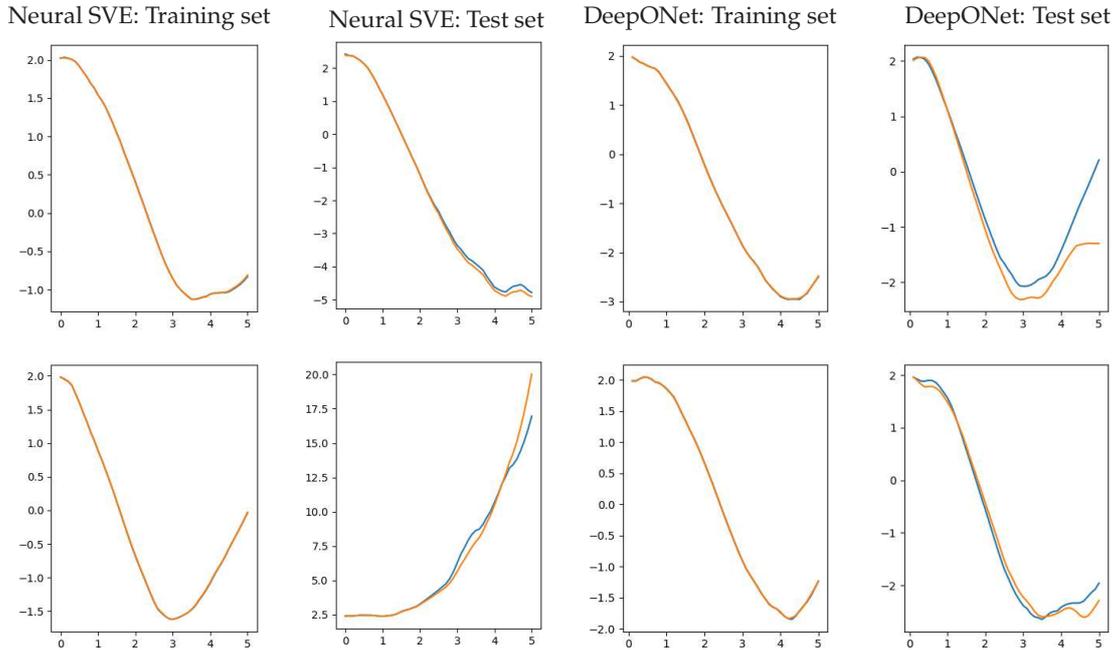


Figure 3.2: Sample neural SVE and DeepONet paths from the training and the test set for the disturbed pendulum equation with nonlinear drift and $n = 100$. Blue are the original paths and orange the learned approximations.

3.2 Rough Heston equation

The rough Heston model is one of the most prominent representatives of rough volatility models in mathematical finance, see e.g. [1, 12], where the volatility process is modeled by SVEs with the singular kernels $(t - s)^{-\alpha}$ for some $\alpha \in (0, 1/2)$, that is

$$V_t = V_0 + \frac{1}{\Gamma(\alpha)} \int_0^t (t - s)^{-\alpha} \lambda (\theta - V_s) ds + \frac{\lambda \nu}{\Gamma(\alpha)} \int_0^t (t - s)^{-\alpha} \sqrt{|V_s|} dB_s, \quad t \in [0, T],$$

where $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$ denotes the real valued Gamma function, and $\lambda, \theta, \nu \in \mathbb{R}$. As specific example, we consider the one-dimensional equation

$$V_t = \xi + \frac{1}{\Gamma(0.4)} \int_0^t (t - s)^{-0.4} (2 - V_s) ds + \frac{1}{\Gamma(0.4)} \int_0^t (t - s)^{-0.4} \sqrt{|V_s|} dB_s, \quad t \in [0, T] \quad (3.3)$$

with the target to learn its dynamics by neural SVEs and DeepONet. The results are presented in Table 3.3. Neural SVEs outperform DeepONet here by far.

Sample paths of the training and the testing sets together with their learned approximations are shown in Fig. 3.3.

3.3 Generalized Ornstein-Uhlenbeck process

The Ornstein-Uhlenbeck process, introduced in [39], is a commonly used stochastic process with applications in finance, physics or biology, see e.g. [29, 38, 40]. We consider the

Table 3.3: Mean relative L^2 -losses after training for the rough Heston equation (3.3).

Neural SVE	Train set	Test set	DeepONet	Train set	Test set
$n = 100$	0.003	0.003	$n = 100$	0.035	0.13
$n = 500$	0.0025	0.0028	$n = 500$	0.004	0.037
$n = 2000$	0.0015	0.0017	$n = 2000$	0.003	0.014

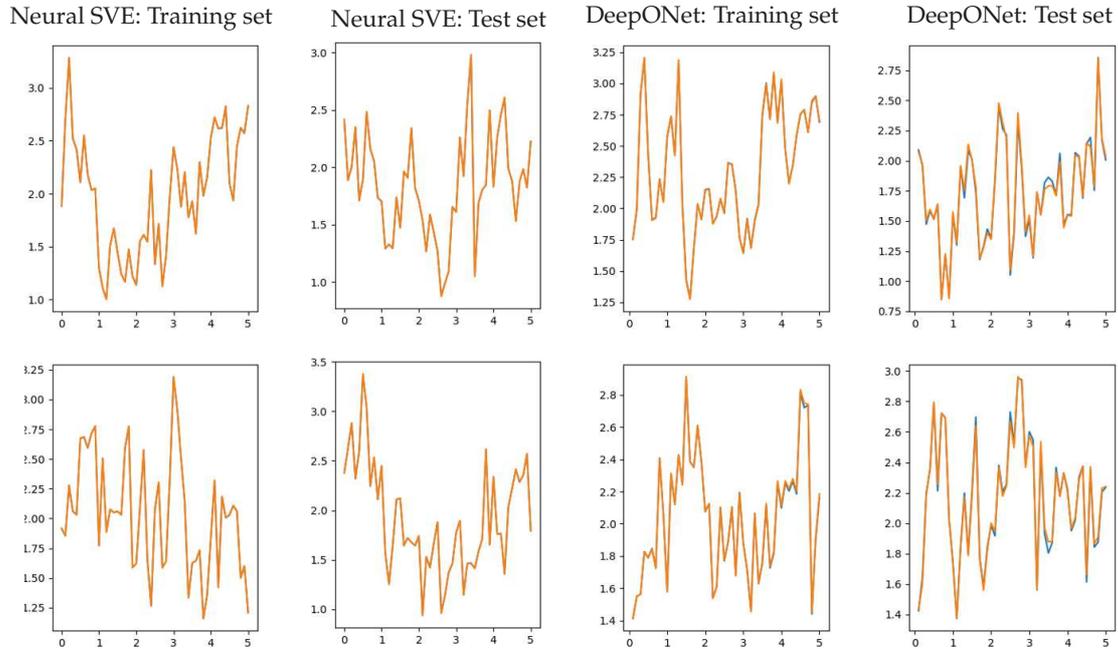


Figure 3.3: Sample neural SVE and DeepONet paths from the training and the test set for the rough Heston equation and $n = 2000$. Blue (barely visible) are the original paths and orange the learned approximations.

generalized Ornstein-Uhlenbeck process that is given by the stochastic differential equation

$$dX_t = \theta(\mu(t, X_t) - X_t) dt + \sigma(t, X_t) dB_t, \quad t \in [0, T],$$

which, using Itô's formula, can be equivalently rewritten as the SVE

$$X_t = X_0 e^{-\theta t} + \theta \int_0^t e^{-\theta(t-s)} \mu(s, X_s) ds + \int_0^t e^{-\theta(t-s)} \sigma(s, X_s) dB_s, \quad t \in [0, T].$$

As prototyping example, we consider the one-dimensional equation

$$X_t = \zeta e^{-t} + \int_0^t e^{-(t-s)} X_s ds + \int_0^t e^{-(t-s)} \sqrt{|X_s|} dB_s, \quad t \in [0, T], \quad (3.4)$$

with the target to learn its dynamics by neural SVEs and DeepONet. The results are presented in Table 3.4.

Sample paths of the training and the testing sets together with their learned approximations are shown in Fig. 3.4.

Table 3.4: Mean relative L^2 -losses after training for the one-dimensional Ornstein-Uhlenbeck equation (3.4).

Neural SVE	Train set	Test set	DeepONet	Train set	Test set
$n = 100$	0.015	0.038	$n = 100$	0.025	0.23
$n = 500$	0.014	0.036	$n = 500$	0.018	0.15
$n = 2000$	0.014	0.02	$n = 2000$	0.028	0.12

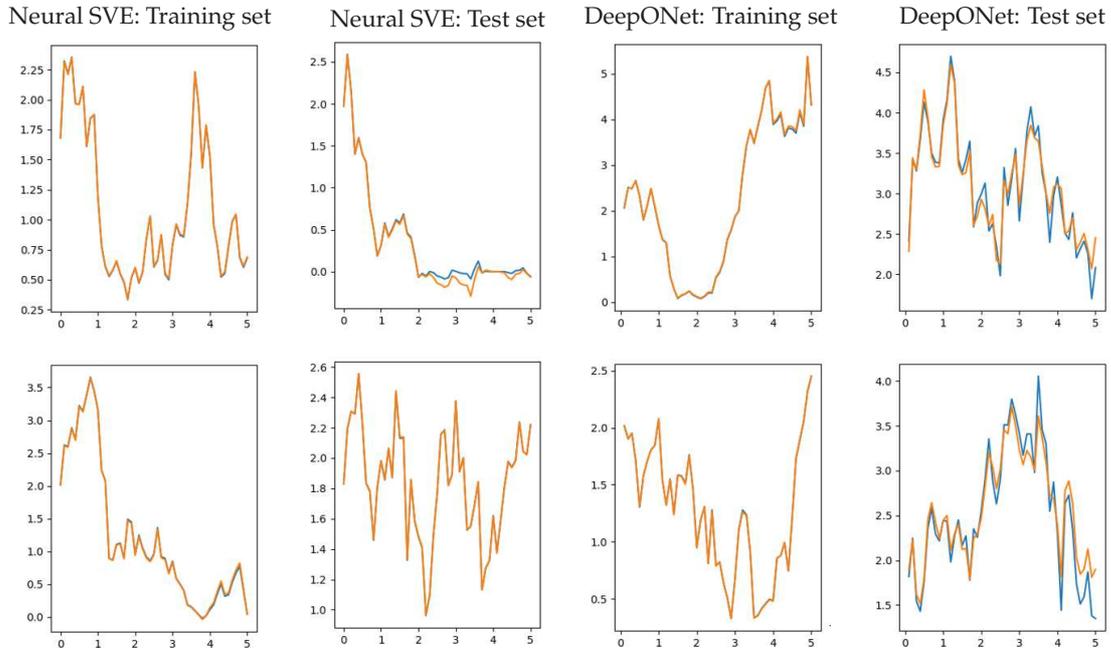


Figure 3.4: Sample neural SVE and DeepONet paths from the training and the test set for the one-dimensional Ornstein-Uhlenbeck equation and $n = 500$. Blue (barely visible) are the original paths and orange the learned approximations.

Moreover, neural SVEs are able to learn multi-dimensional SVEs. As an example, we consider the two-dimensional equation

$$\begin{pmatrix} X_t^1 \\ X_t^2 \end{pmatrix} = \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} e^{-t} + \int_0^t e^{-(t-s)} \begin{pmatrix} X_s^1 \\ X_s^2 \end{pmatrix} ds + \int_0^t e^{-(t-s)} \begin{pmatrix} \sqrt{|X_s^1|}, 0 \\ 0, \sqrt{|X_s^2|} \end{pmatrix} dB_s, \quad t \in [0, T], \quad (3.5)$$

where B is a 2-dimensional Brownian motion, and try to learn its dynamics by neural SVEs. The results are presented in Table 3.5.

Sample paths of the training and the testing sets together with their learned approximations are shown in Fig. 3.5.

3.4 Monetary reserve modelling

For a higher-dimensional model with cross-dependency between the different trajectories we simulate and approximate the bank run model as in [3]. There, the dynamics of the

Table 3.5: Mean relative L^2 -losses after training for the two-dimensional Ornstein-Uhlenbeck equation (3.5).

Neural SVE	Train set	Test set
$n = 100$	0.038	0.095
$n = 500$	0.04	0.085
$n = 2000$	0.038	0.04

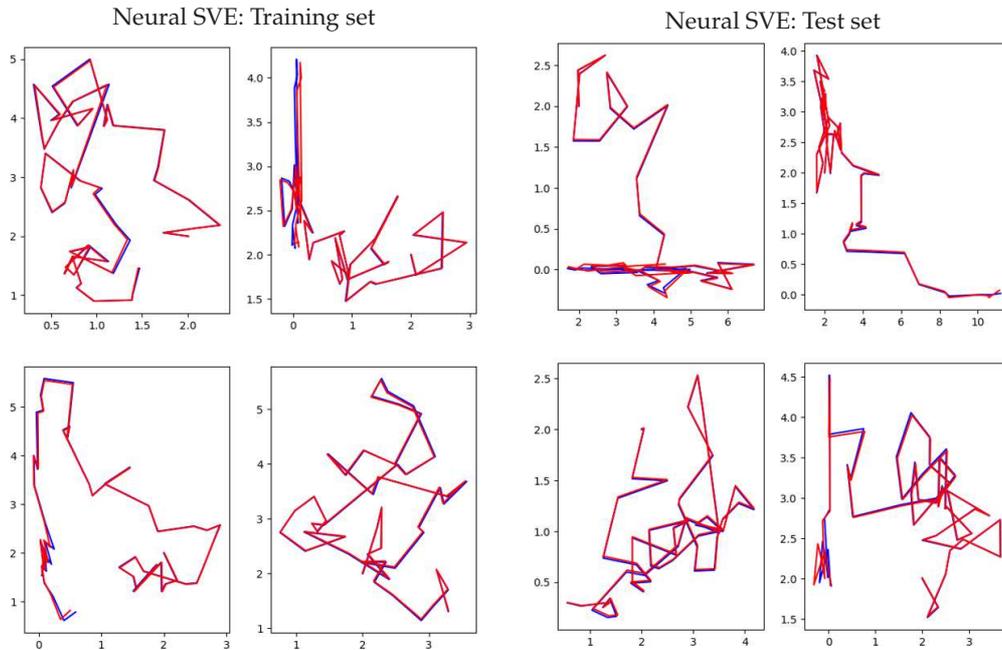


Figure 3.5: Sample neural SVE paths from the training and the test set for the two-dimensional Ornstein-Uhlenbeck equation and $n = 2000$. Blue (barely visible) are the original paths and red the learned approximations.

log-monetary reserves of N banks are modelled by the coupled diffusion processes X^i , $i = 1, \dots, N$,

$$dX_t^i = (\alpha_t^i - \alpha_{t-\tau}^i) dt + \sigma dW_t^i, \quad 0 \leq t \leq T, \tag{3.6}$$

where W^i , $i = 1, \dots, N$, are independent standard Brownian motions, and the rate of borrowing or lending α_t^i represents the control of the bank i on the system. Vice versa, the delayed control $\alpha_{t-\tau}^i$ represents the repayments after a fixed time τ . In the paper mentioned the authors solve the differential game where bank i , $i = 1, \dots, N$, aims to minimize its objective function

$$J^i(\alpha) = \mathbb{E} \left[\int_0^T f_i(X_t, \alpha_t^i) dt + g_i(X_T) \right]$$

with $f_i(x, \alpha^i)$ and $g_i(x, \alpha^i)$ heavily depending on $\sum_{j=1}^N (x_j - x_i)/N$. The optimal control α^* can only be stated in terms of multiple differential equations with no closed-form solution,

Table 3.6: Mean relative L^2 -losses after training for the monetary reserve model (3.6).

Neural SVE	Train set	Test set
$n = 1000$	0.037	0.039

therefore we choose

$$\alpha_t^i = \left(0.1 + 0.5 \sin \left(\frac{\pi t}{2T} \right) \right) (\bar{X}_t - X_t^i).$$

Here, we choose $X_0^i \sim \mathcal{N}(10, 1), \sigma = 0.05, T = 50, \Delta t = 1$ and $\tau = 10$, opening up to the interpretation of each time unit corresponding to one day and borrowing (lending) decisions being made on a daily basis. Note that this problem is highly complex since \bar{X} is not an input to the neural network, so the network has to learn how X^i depends on $X^j, j = 1, \dots, N$, itself. To account for the higher complexity we opted for a network with latent dimension $d_h = d_K = 24$ and used $n = 1000$ datasets with 10 banks each.

As shown in Table 3.6, the neural SVE learned the dynamics quite well also in this example. Again, it can be observed that the neural SVE learned the dynamics quite well.

In Fig. 3.6 one can see the log-monetary reserves of the 10 banks from one data set and its estimation from the test and the training set each. The dips when the first borrowing contracts end after $t = 10$ are not as pronounced in the network’s approximation as in the original data set. This likely stems from neural networks struggling to learn indicator functions such as K_μ here. Still, the order of the banks’ reserves as well as their magnitude are captured very well.

3.5 Comparison to neural SDEs

Introduced in [18], a neural stochastic differential equation (neural SDE) is defined by

$$\begin{aligned} Z_0 &= L_\theta(\xi), \\ Z_t &= Z_0 g_\theta(t) + \int_0^t \mu_\theta(s, Z_s) ds + \int_0^t \sigma_\theta(s, Z_s) dB_s, \\ X_t &= \Pi_\theta(Z_t), \quad t \in [0, T], \end{aligned}$$

where all objects are defined as in the neural SVE (2.2). Since a neural SDE does not possess the kernel functions $K_{\mu,\theta}$ and $K_{\sigma,\theta}$ compared to the neural SVE (2.2), it is not able to fully capture the dynamics induced by SVEs.

Note that due to the need of discretizing the time interval when it comes to computations, some of the properties introduced by the kernels are attenuated. However, the memory structure of an SVE is a property which can be learned by a neural SVE but, in general, not by a neural SDE since SDEs possess the Markov property. Therefore, to see the potential capabilities of neural SVEs compared to neural SDEs, it is best to look at examples where the dependency on the whole path plays a crucial role. To construct such an example, we consider the kernels

$$K_\mu(s, t) := K_\sigma(s, t) := K(t - s) = \begin{cases} 1, & \text{if } (t - s) \leq T/4, \\ -1, & \text{if } (t - s) > T/4, \end{cases}$$

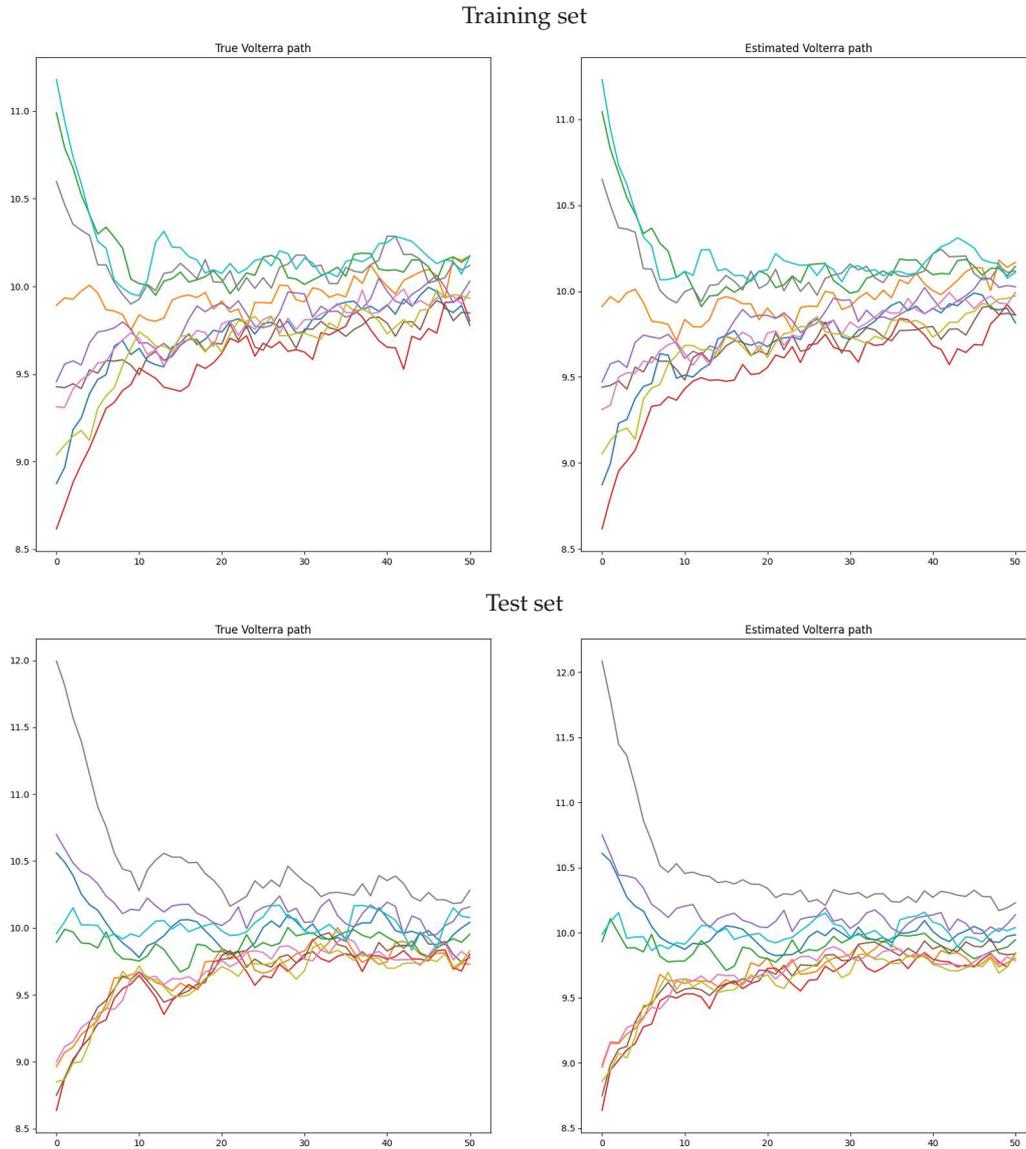


Figure 3.6: Sample paths from the training and test sets for the monetary reserve model. On the left side one can see the paths as from the data set, on the right side the learnt approximation by the SVE. The same color corresponds to the same bank.

and aim to learn the dynamics to the one-dimensional SVE

$$X_t = \zeta + \int_0^t K(t-s)(2 - X_s) ds + \int_0^t K(t-s)\sqrt{|X_s|} dB_s, \quad t \in [0, T], \quad (3.7)$$

where $\zeta \sim \mathcal{N}(5, 0.5)$ and $T = 5$. The process $(X_t)_{t \in [0,5]}$ is expected to decrease in the first quarter of the interval $[0, 5]$ where $K(t-s) = 1$ holds due to the mean-reverting effect of

the drift coefficient $\mu(s, x) = 2 - x$, then something unpredictable will happen and finally in the last part of the interval $t \in [0, 5]$ where the kernels attain -1 for a large proportion of $s \in [0, t]$, the process might become big due to the turning sign in the drift. Hence, it is expected that the path dependency will have a substantial impact.

We learn the dynamics of Eq. (3.7) simulated on an equally-sized grid with grid size $\Delta t = 0.1$ by a neural SDE and by a neural SVE for a dataset of size $n = 500$ and compare the results in Table 3.7. It can be observed that the neural SDE fails to learn the dynamics of (3.7) properly while the neural SVE performs well.

Sample paths of the training and the testing sets together with their learned approximations are shown in Fig. 3.7.

Table 3.7: Mean relative L^2 -losses after training for the SVE (3.7).

Neural SVE	Train set	Test set	Neural SDE	Train set	Test set
$n = 500$	0.008	0.009	$n = 500$	0.19	0.21

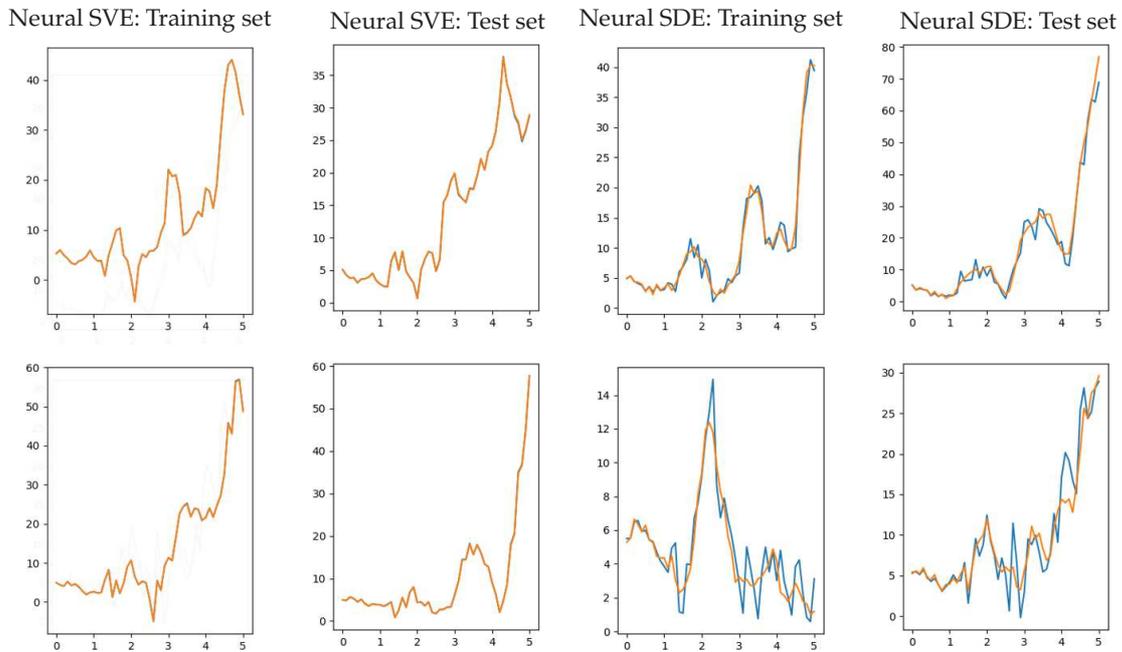


Figure 3.7: Sample neural SVE and neural SDE paths from the training and the test set for the SVE (3.7) and $n = 500$. Blue are the original paths and orange the learned approximations.

3.6 Computational aspects

Next we briefly analyse computational aspects of the neural SVE, most particularly, its runtime and memory usage. For each property we outline the influence of the number of

epochs, the grid size Δt , the terminal time T , the dimension of the SVE, the latent dimension and the sample-size n .

All computations were made using an AMD Ryzen 7 5800X processor. Using CUDA with the NVIDIA GeForce RTX 3060 roughly triples the runtime. This is likely due to the relatively small size, especially the small width, of the neural network. Note that for a one-dimensional SVE there are 1264 parameters to be trained, for a two-dimensional SVE there are 1901 parameters. These parameters are basically split into five different neural networks (one for each kernel and coefficient as well as one for g). Also solving an SVE cannot be parallelized due to the past-dependency of the solution.

As base case let us take the parameters as in the beginning of this section with $n = 500$, a batch size of 50 and 1000 epochs. Runtime and memory usage scale linearly in the sample size. During the training, the runtime per iteration remains roughly constant. Hence, the runtime grows linearly in the number of epochs. The memory usage is basically independent of the number of epochs. We therefore focus on the runtime per epoch. The results are reported in Table 3.8. Doubling the latent dimension from 12 to 24 increases the number of parameters in the one-dimensional model to 4540. The impact to the runtime is relatively low, it increases to barely 17 seconds. A two-dimensional model with latent dimension 24 has 6965 parameters. Only when rapidly increasing the latent dimension one can see a significant effect. The very small dependency of runtime and memory on the number of parameters to be learned indicates that the main effort lies in solving the SVE numerically. In order to improve performance of neural SVE one should focus on finding a more efficient numerical scheme to solve SVEs in the first place.

Changing the grid size Δt has an influence that is counterintuitive at first glance: Halving the grid size quadruples the runtime. Obviously, for a fixed terminal time T , one has to evaluate at (nearly) twice as many evaluation points. But the numerical scheme for

Table 3.8: Computational performance of neural networks trained to model stochastic Volterra equations. Reported are the number of parameters, average training time per iteration (in seconds), and peak memory usage (in MB) for varying input dimensions (dim) and latent dimensions of the network.

Latent dimension = 12			
	Parameters	Training time per iteration	Memory
dim = 1	1264	16,04	1233
dim = 2	1901	15,91	1232
Latent dimension = 24			
	Parameters	Training time per iteration	Memory
dim = 1	4540	16,08	1231
dim = 2	6965	16,35	1233
Latent dimension = 120			
	Parameters	Training time per iteration	Memory
dim = 1	103324	18,70	1234
dim = 2	161525	20,09	1263

solving an SVE requires the whole past, that now also contains twice as many points to consider. Doubling the terminal time T while keeping the grid size Δt fixed has the same impact.

Remark 3.2. Our procedure corresponds to a discretise-then-optimize approach in classical neural SDE. For neural SDEs one may also consider an optimize-then-discretize approach. The optimize-then-discretize approach requires less memory but it is slower and may lead to an inaccurate solution [18]. Since the optimize-then-discretize approach transforms the SDE into an backward SDE for the backpropagation, it is infeasible for neural SVE due to their non-Markovian structure.

Appendix A Proofs of Theorem 2.1 and Proposition 2.1

In this appendix, we present the proofs Theorem 2.1 and of Proposition 2.1.

Proof of Theorem 2.1. We provide a proof by contradiction. To that end, we assume that there are $\delta > 0$ and an increasing sequence $(n_k)_{k \in \mathbb{N}} \subset \mathbb{N}$ satisfying

$$\inf_{k \in \mathbb{N}} \mathbb{E} \left[\sup_{t \in [0, T]} |X_t^{n_k} - X_t|^2 \right] \geq \delta.$$

Moreover, we define

$$A_t^n := \int_0^t \mu_n(s, X_s^n) ds, \quad M_t^n := \int_0^t \sigma_n(s, X_s^n) ds$$

for $t \in [0, T], n \in \mathbb{N}$.

As in the proof of [33, Lemma 3.8], we can obtain the tightness of probability measure

$$\mathbb{P}_{X, X^{n_k}, A^{n_k}, M^{n_k}, B}, \quad k \in \mathbb{N},$$

which denotes the probability distribution of the corresponding random vector

$$(X, X^{n_k}, A^{n_k}, M^{n_k}, B).$$

Using Prokhorov's theorem and the Skorokhod representation theorem, one deduces that there is a probability space $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$ with continuous stochastic processes $\hat{X}^l, \hat{Y}^l, \hat{B}^l, \hat{A}^l, \hat{M}^l, l \in \mathbb{N}$ and $\hat{X}, \hat{Y}, \hat{B}, \hat{A}, \hat{M}$ such that

$$\begin{aligned} (\hat{\xi}^l, \hat{X}^l, \hat{Y}^l, \hat{B}^l, \hat{A}^l, \hat{M}^l) &\stackrel{\mathcal{L}}{\sim} (\xi, X, X^{n_{k_l}}, B, A^{n_{k_l}}, M^{n_{k_l}}), \quad l \in \mathbb{N}, \\ (\hat{X}^l, \hat{Y}^l, \hat{B}^l, \hat{A}^l, \hat{M}^l) &\rightarrow (\hat{X}, \hat{Y}, \hat{B}, \hat{A}, \hat{M}) \end{aligned}$$

in $C([0, T]; \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^d)$ as $l \rightarrow \infty$, $\hat{\mathbb{P}}$ -a.s., and $\hat{\xi}^l \rightarrow \hat{\xi}$ as $l \rightarrow \infty$, $\hat{\mathbb{P}}$ -a.s.¹ With $\stackrel{\mathcal{L}}{\sim}$ we denote equality in law. From here on we identify any space of continuous functions with the supremum norm.

¹One may drop the $\xi, \hat{\xi}^l, l \in \mathbb{N}$, here, as they are uniquely determined by the X, \hat{X}^l , respectively.

Applying Fatou’s lemma, we obtain

$$\begin{aligned} \delta &\leq \liminf_{k \rightarrow \infty} \mathbb{E} \left[\sup_{t \in [0, T]} |X_t^{n_k} - X_t|^2 \right] \\ &\leq \liminf_{l \rightarrow \infty} \mathbb{E}_{\hat{\mathbb{P}}} \left[\sup_{t \in [0, T]} |\hat{Y}_t^l - \hat{X}_t^l|^2 \right] \\ &\leq \mathbb{E}_{\hat{\mathbb{P}}} \left[\limsup_{l \rightarrow \infty} \sup_{t \in [0, T]} |\hat{Y}_t^l - \hat{X}_t^l|^2 \right] \\ &= \mathbb{E}_{\hat{\mathbb{P}}} \left[\sup_{t \in [0, T]} |\hat{Y}_t - \hat{X}_t|^2 \right]. \end{aligned}$$

We check that (\hat{X}, \hat{Z}) with $\hat{Z} := \hat{A} + \hat{M}$, $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$, $(\hat{\mathcal{F}}_t)_{t \in [0, T]}$ solves the Volterra local martingale problem [33, Definition 2.4] given $(\zeta g, \mu, \sigma, K_\mu, K_\sigma)$. Then, by [33, Lemma 2.7] \hat{Y} is a solution of the SVE (2.1). Conditions (i)-(iii) of [33, Definition 2.4] are clear. (iv) of [33, Definition 2.4] follows as in the proof of [33, Lemma 3.9]. We only need to note that the definitions and results of aforementioned paper [33] extend verbatim to the slightly more general present setting.

To show (v) of [33, Definition 2.4] we introduce the processes

$$Z^l := A^{n_{k_l}} + M^{n_{k_l}}, \quad \hat{Z}^l = \hat{A}^l + \hat{M}^l, \quad l \in \mathbb{N}.$$

Since $(\hat{Y}^l, \hat{M}^l) \stackrel{\mathcal{Q}}{\approx} (X^{n_{k_l}}, M^{n_{k_l}})$, for every $l \in \mathbb{N}$, and pathwise uniqueness holds by assumption a general version of the Yamada-Watanabe result (see, e.g., [24]) shows that we may express \hat{Y}^l as solution of

$$\hat{Y}_t^l = \hat{\xi}^l g_{k_l}(t) + \int_0^t K_{\mu, n_{k_l}}(t-s) \mu_{n_{k_l}}(s, \hat{Y}_s^l) ds + \int_0^t K_{\sigma, n_{k_l}}(t-s) d\hat{M}_s^l, \quad t \in [0, T]. \quad (\text{A.1})$$

We know that $\hat{Y}^l \rightarrow \hat{Y}$ and that $\hat{\xi}^l g_{k_l} \rightarrow \hat{\xi} g$ $\hat{\mathbb{P}}$ -a.s. By $\hat{\xi}^l \stackrel{\mathcal{Q}}{\approx} \xi$, $l \in \mathbb{N}$, we can conclude $\hat{\xi} \stackrel{\mathcal{Q}}{\approx} \xi$. Next we show

$$\left(\int_0^t K_{\mu, n_{k_l}}(t-s) d\hat{A}_s^l \right)_{t \in [0, T]} \xrightarrow{\hat{\mathbb{P}}} \left(\int_0^t K_\mu(t-s) d\hat{A}_s \right)_{t \in [0, T]}. \quad (\text{A.2})$$

Therefore, let $\eta, \phi > 0$ be arbitrary but fixed. Denoting $\bar{K} := \int_0^T |K_\mu(s)| ds$, we choose $N_1 \in \mathbb{N}$ and $L_1 \in \mathbb{N}$ sufficiently large, such that

$$\hat{\mathbb{P}} \left(\|\hat{Y}\|_\infty \geq \frac{N_1}{2} \right) \leq \frac{\phi}{3}, \quad \hat{\mathbb{P}} \left(\|\hat{Y}^l - \hat{Y}\|_\infty \geq \max \left(\frac{\eta}{3C_{\mu, \sigma} \bar{K}}, \frac{N_1}{2} \right) \right) \leq \frac{\phi}{3}$$

for all $l \geq L_1$. On $\{\|\hat{Y}\|_\infty \vee \|\hat{Y}^l\|_\infty \leq N_1\}$, we have

$$|G_t^l - G_t| := \left| \int_0^t K_{\mu, n_{k_l}}(t-s) \mu_{n_{k_l}}(s, \hat{Y}_s^l) ds - \int_0^t K_\mu(t-s) \mu_n(s, \hat{Y}_s^l) ds \right|$$

$$\begin{aligned}
 &\leq \left| \int_0^t \left(K_{\mu, n_{k_l}}(t-s) - K_\mu(t-s) \right) \mu_{n_{k_l}}(s, \hat{Y}_s^l) \, ds \right| \\
 &\quad + \int_0^t |K_\mu(s)| \, ds \left(\sup_{s \in [0, T]} \sup_{x \in [-N_1, N_1]} |\mu_{n_{k_l}}(s, x) - \mu(s, x)| \right. \\
 &\quad \quad \left. + \sup_{s \in [0, T]} |\mu(s, \hat{Y}_s^l) - \mu(s, \hat{Y}_s)| \right) \\
 &\leq C_{\mu, \sigma}(1 + N_1) \int_0^t |K_{\mu, n_{k_l}}(s) - K_\mu(s)| \, ds \\
 &\quad + \bar{K} \left(\sup_{s \in [0, T]} \sup_{x \in [-N_1, N_1]} |\mu_{n_{k_l}}(s, x) - \mu(s, x)| + C_{\mu, \sigma} \|\hat{Y}^l - \hat{Y}\|_\infty \right).
 \end{aligned}$$

By the convergence of the kernels and coefficients there is an $L_2 \geq L_1$ such that, for all $l \geq L_2$,

$$\begin{aligned}
 C_{\mu, \sigma}(1 + N_1) \int_0^T |K_{\mu, n_{k_l}}(s) - K_\mu(s)| \, ds &\leq \frac{\eta}{3}, \\
 \bar{K} \sup_{s \in [0, T]} \sup_{x \in [-N_1, N_1]} |\mu_{n_{k_l}}(s, x) - \mu(s, x)| &\leq \frac{\eta}{3}.
 \end{aligned}$$

Note that $\hat{\mathbb{P}}(\bar{K}C_{\mu, \sigma}\|\hat{Y}^l - \hat{Y}\|_\infty \geq \eta/3) \leq \phi/3$ and

$$\begin{aligned}
 &\hat{\mathbb{P}}(\|\hat{Y}\|_\infty \vee \|\hat{Y}^l\|_\infty \geq N_1) \\
 &\leq \hat{\mathbb{P}}(\{\|\hat{Y}\|_\infty \geq N_1\} \cup \{\|\hat{Y}^l - \hat{Y}\|_\infty + \|\hat{Y}\|_\infty \geq N_1\}) \\
 &\leq \hat{\mathbb{P}}\left(\left\{\|\hat{Y}\|_\infty \geq \frac{N_1}{2}\right\}\right) + \hat{\mathbb{P}}\left(\|\hat{Y}^l - \hat{Y}\|_\infty \geq \frac{N_1}{2}\right) \leq \frac{2\phi}{3}.
 \end{aligned}$$

Hence, for all $l \geq L_2$ we have

$$\begin{aligned}
 &\hat{\mathbb{P}}(\|G^l - G\|_\infty \geq \eta) \\
 &\leq \hat{\mathbb{P}}(\{\|G^l - G\|_\infty \geq \eta\} \cap \{\|\hat{Y}\|_\infty \vee \|\hat{Y}^l\|_\infty < N_1\}) \\
 &\quad + \hat{\mathbb{P}}(\|\hat{Y}\|_\infty \vee \|\hat{Y}^l\|_\infty \geq N_1) \leq \phi.
 \end{aligned}$$

It remains to show that

$$\left(\int_0^t K_{\sigma, n_{k_l}}(t-s) \, d\hat{M}_s^l \right)_{t \in [0, T]} \xrightarrow{\hat{\mathbb{P}}} \left(\int_0^t K_\sigma(t-s) \, d\hat{M}_s \right)_{t \in [0, T]}.$$

As $\tilde{p} = p/(p-2) \leq 1 + \varepsilon/2$, using the Burkholder-Davis-Gundy inequality, we get, for any $t \in [0, T]$,

$$\mathbb{E}_{\hat{\mathbb{P}}} \left[\left(\int_0^t K_{\sigma, n_{k_l}}(t-s) \, d\hat{M}_s^l - \int_0^t K_\sigma(t-s) \, d\hat{M}_s \right)^p \right]^{\frac{1}{p}}$$

$$\begin{aligned}
 &\leq \mathbb{E}_{\hat{\mathbb{P}}} \left[\left(\int_0^t K_{\sigma, n_{k_l}}(t-s) d(\hat{M}_s^l - \hat{M}_s) \right)^p \right]^{\frac{1}{p}} \\
 &\quad + \mathbb{E}_{\hat{\mathbb{P}}} \left[\left(\int_0^t (K_{\sigma, n_{k_l}}(t-s) - K_{\sigma}(t-s)) d\hat{M}_s \right)^p \right]^{\frac{1}{p}} \\
 &\leq \mathbb{E}_{\hat{\mathbb{P}}} \left[\left(\int_0^t (K_{\sigma, n_{k_l}}(t-s) (\sigma_{n_{k_l}}(s, \hat{Y}_s^l) - \sigma(s, \hat{Y}_s)))^2 ds \right)^{\frac{p}{2}} \right]^{\frac{1}{p}} \\
 &\quad + \mathbb{E}_{\hat{\mathbb{P}}} \left[\left(\int_0^t ((K_{\sigma}(t-s) - K_{\sigma, n_{k_l}}(t-s)) \sigma(s, \hat{Y}_s))^2 ds \right)^{\frac{p}{2}} \right]^{\frac{1}{p}} \\
 &\leq C_{p,t} \left(\left(\int_0^t |K_{\sigma, n_{k_l}}(s)|^{2\bar{p}} ds \right)^{\frac{p}{2\bar{p}}} \mathbb{E}_{\hat{\mathbb{P}}} \left[\int_0^t |\sigma_{n_{k_l}}(s, \hat{Y}_s^l) - \sigma(s, \hat{Y}_s)|^p ds \right]^{\frac{1}{p}} \right. \\
 &\quad \left. + \left(\int_0^t |K_{\sigma}(s) - K_{\sigma, n_{k_l}}(s)|^{2\bar{p}} ds \right)^{\frac{p}{2\bar{p}}} \mathbb{E}_{\hat{\mathbb{P}}} \left[\int_0^t |\sigma(s, \hat{Y}_s)|^p ds \right]^{\frac{1}{p}} \right).
 \end{aligned}$$

Note that $\int_0^t |K_{\sigma, n_{k_l}}(s)|^{2\bar{p}} ds$ is uniformly bounded in l . We can obtain

$$\int_0^t \sigma_{n_{k_l}}(s, \hat{Y}_s^l) ds \xrightarrow{\hat{\mathbb{P}}} \int_0^t \sigma(s, \hat{Y}_s) ds$$

by the same steps we used to show (A.2). One can mimic the proof of [34, Lemma 3.4] to obtain

$$\sup_{t \in [0, T]} \mathbb{E}_{\hat{\mathbb{P}}} [|\hat{Y}_t^l|^p] \leq C_{p, L, \gamma, \epsilon, T, \mu, \sigma} \left(1 + \mathbb{E}_{\hat{\mathbb{P}}} [|\hat{\xi}^l|^p] \sup_{t \in [0, T]} |g_{k_l}(t)|^p \right),$$

where $C_{p, L, \gamma, \epsilon, T, \mu, \sigma}$ depends only on $p, L, \gamma, \epsilon, T$ and the linear growth constant $C_{\mu, \sigma}$ of the coefficients (see Assumption 2.2). Since the $\hat{\xi}^l, l \in \mathbb{N}$, are identically distributed with finite p -th moment, and since $\sup_{t \in [0, T]} |g_{k_l}(t)| \leq CT^\gamma + 1$ by the γ -Hölder-continuity of the $g^l, l \in \mathbb{N}$, we then get uniform p -integrability of \hat{Y}^l . Together with the uniform linear growth condition on $\sigma_{n_{k_l}}, l \in \mathbb{N}$, one gets

$$\mathbb{E}_{\hat{\mathbb{P}}} \left[\int_0^t |\sigma_{n_{k_l}}(s, \hat{Y}_s^l) - \sigma(s, \hat{Y}_s)|^p ds \right] \rightarrow 0 \text{ as } l \rightarrow \infty.$$

Therefore, with Assumption 2.1 we can conclude that

$$\mathbb{E}_{\hat{\mathbb{P}}} \left[\left(\int_0^t K_{\sigma, n_{k_l}}(t-s) d\hat{M}_s^l - \int_0^t K_{\sigma}(t-s) d\hat{M}_s \right)^p \right]^{\frac{1}{p}} \rightarrow 0 \text{ as } l \rightarrow \infty$$

and it follows that, for all $t \in [0, T]$,

$$\int_0^t K_{\sigma, n_{k_l}}(t-s) d\hat{M}_s^l \xrightarrow{\hat{\mathbb{P}}} \int_0^t K_{\sigma}(t-s) d\hat{M}_s \text{ as } l \rightarrow \infty.$$

By (A.1) we know that there is some continuous process $\hat{V} = (\hat{V}_t)_{t \in [0, T]}$ such that

$$\sup_{r \in [0, T]} \left| \int_0^r K_{\sigma, n_{k_l}}(r-s) d\hat{M}_s^l - \hat{V}_r \right| \xrightarrow{\hat{\mathbb{P}}} 0 \text{ as } l \rightarrow \infty.$$

Using a uniqueness of limits argument, one obtains $\hat{V}_t = \int_0^t K_{\sigma}(t-s) d\hat{M}_s$ for all $t \in [0, T]$ and by the continuity we can conclude that the processes are indistinguishable. Taking a limit in probability in (A.1) or the $\hat{\mathbb{P}}$ -a.s. limit of some subsequence, we obtain that \hat{X} is a solution to the SVE (2.1) such that $\mathbb{E}[\sup_{t \in [0, T]} |\hat{X}_t - X_t|^2] \geq \delta$, which contradicts the assumption that pathwise uniqueness holds for SVE (2.1) and, thus, completes the proof. \square

Proof of Proposition 2.1. First notice that, due to Assumptions 2.3 and 2.4, there exist unique solutions $(X_t)_{t \in [0, T]}$ and $(\tilde{X}_t)_{t \in [0, T]}$ to the SVEs (2.1) and (2.4), see [41, Theorem 1.1].

Let $t \in [0, T]$ and $C > 0$ be a generic constant, which may change from line to line. We get that

$$\begin{aligned} \mathbb{E}[|X_t - \tilde{X}_t|^p] &= \mathbb{E} \left[\left| \zeta(g(t) - \tilde{g}(t)) + \int_0^t K_{\mu}(t-s)\mu(s, X_s) ds - \int_0^t \tilde{K}_{\mu}(t-s)\tilde{\mu}(s, \tilde{X}_s) ds \right. \right. \\ &\quad \left. \left. + \int_0^t K_{\sigma}(t-s)\sigma(s, X_s) dB_s - \int_0^t \tilde{K}_{\sigma}(t-s)\tilde{\sigma}(s, \tilde{X}_s) dB_s \right|^p \right] \\ &\leq C \left(\sup_{s \in [0, T]} |g(s) - \tilde{g}(s)|^p + \mathbb{E} \left[\left| \int_0^t (K_{\mu}(t-s) - \tilde{K}_{\mu}(t-s))\mu(s, X_s) ds \right|^p \right] \right. \\ &\quad + \mathbb{E} \left[\left| \int_0^t \tilde{K}_{\mu}(t-s)(\mu(s, X_s) - \tilde{\mu}(s, \tilde{X}_s)) ds \right|^p \right] \\ &\quad + \mathbb{E} \left[\left| \int_0^t (K_{\sigma}(t-s) - \tilde{K}_{\sigma}(t-s))\sigma(s, X_s) dB_s \right|^p \right] \\ &\quad \left. + \mathbb{E} \left[\left| \int_0^t \tilde{K}_{\sigma}(t-s)(\sigma(s, X_s) - \tilde{\sigma}(s, \tilde{X}_s)) dB_s \right|^p \right] \right). \end{aligned}$$

Applying the Burkholder-Davis-Gundy inequality and Hölder's inequality with (2.5), we deduce that

$$\begin{aligned} \mathbb{E}[|X_t - \tilde{X}_t|^p] &\leq C \left(\|g - \tilde{g}\|_{\infty}^p + \left(\int_0^t |K_{\mu}(t-s) - \tilde{K}_{\mu}(t-s)|^q ds \right)^{\frac{p}{q}} \mathbb{E} \left[\int_0^t |\mu(s, X_s)|^p ds \right] \right. \\ &\quad + \left(\int_0^t |\tilde{K}_{\mu}(t-s)|^q ds \right)^{\frac{p}{q}} \mathbb{E} \left[\int_0^t |\mu(s, X_s) - \tilde{\mu}(s, \tilde{X}_s)|^p ds \right] \\ &\quad \left. + \mathbb{E} \left[\left| \int_0^t (K_{\sigma}(t-s) - \tilde{K}_{\sigma}(t-s))^2 \sigma(s, X_s)^2 ds \right|^{\frac{p}{2}} \right] \right) \end{aligned}$$

$$\begin{aligned}
 & + \mathbb{E} \left[\left| \int_0^t \tilde{K}_\sigma(t-s)^2 (\sigma(s, X_s) - \tilde{\sigma}(s, \tilde{X}_s))^2 ds \right|^{\frac{p}{2}} \right] \\
 \leq & C \left(\|g - \tilde{g}\|_\infty^p + \|K_\mu - \tilde{K}_\mu\|_q^p \int_0^t (1 + \mathbb{E}[|X_s|^p]) ds \right. \\
 & + \left(\int_0^t |\tilde{K}_\mu(t-s)|^q ds \right)^{\frac{p}{q}} \int_0^t \mathbb{E}[|\mu(s, X_s) - \mu(s, \tilde{X}_s)|^p] ds \\
 & + \left(\int_0^t |K_\sigma(t-s) - \tilde{K}_\sigma(t-s)|^{2\tilde{q}} ds \right)^{\frac{p}{2\tilde{q}}} \mathbb{E} \left[\int_0^t |\sigma(s, X_s)|^p ds \right] \\
 & \left. + \left(\int_0^t |\tilde{K}_\sigma(t-s)|^{2\tilde{q}} ds \right)^{\frac{p}{2\tilde{q}}} \mathbb{E} \left[\int_0^t |\sigma(s, X_s) - \tilde{\sigma}(s, \tilde{X}_s)|^p ds \right] \right).
 \end{aligned}$$

Using the regularity assumptions on μ and σ (Assumption 2.4) and the boundedness of all moments of Volterra processes (see [34, Lemma 3.4]), we get

$$\begin{aligned}
 \mathbb{E}[|X_t - \tilde{X}_t|^p] \leq & C \left(\|g - \tilde{g}\|_\infty^p + \|\mu - \tilde{\mu}\|_\infty^p + \|\sigma - \tilde{\sigma}\|_\infty^p + \|K_\mu - \tilde{K}_\mu\|_q^p + \|K_\sigma - \tilde{K}_\sigma\|_{2\tilde{q}}^p \right) \\
 & + C \int_0^t \mathbb{E}[|X_s - \tilde{X}_s|^p] ds.
 \end{aligned}$$

Applying Grönwall's lemma leads to

$$\mathbb{E}[|X_t - \tilde{X}_t|^p] \leq C \left(\|g - \tilde{g}\|_\infty^p + \|\mu - \tilde{\mu}\|_\infty^p + \|\sigma - \tilde{\sigma}\|_\infty^p + \|K_\mu - \tilde{K}_\mu\|_q^p + \|K_\sigma - \tilde{K}_\sigma\|_{2\tilde{q}}^p \right),$$

which implies (2.6) by taking the supremum on the left-hand side. \square

References

- [1] E. Abi Jaber and O. El Euch, Multifactor approximation of rough volatility models, *SIAM J. Financial Math.*, 10(2):309–349, 2019.
- [2] O. E. Barndorff-Nielsen and J. Schmiegel, Time change, volatility, and turbulence, in: *Mathematical Control Theory and Finance*, Springer, 29–53, 2008.
- [3] R. Carmona, J.-P. Fouque, S. M. Mousavi, and L.-H. Sun, Systemic risk and stochastic games with delay, *J. Optim. Theory Appl.*, 179(2):366–399, 2018.
- [4] R. T. Q. Chen, J. Behrmann, D. K. Duvenaud, and J.-H. Jacobsen, Residual flows for invertible generative modeling, in: *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates, Inc., 9884–9894, 2019.
- [5] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, Neural ordinary differential equations, in: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Curran Associates Inc., 6572–6583, 2018.
- [6] W. G. Cochran, J.-S. Lee, and J. Potthoff, Stochastic Volterra equations with singular kernels, *Stochastic Process. Appl.*, 56(2):337–349, 1995.
- [7] S. N. Cohen, C. Reisinger, and S. Wang, Hedging option books using neural-SDE market models, *Appl. Math. Finance*, 29(5):366–401, 2022.

- [8] L. Coutin and L. Decreusefond, Stochastic Volterra equations with singular kernels, in: *Stochastic Analysis and Mathematical Physics. Progress in Probability*, Vol. 50, Birkhäuser, 39–50, 2001.
- [9] C. Cuchiero, W. Khosrawi, and J. Teichmann, A generative adversarial network approach to calibration of local stochastic volatility models, *Risks*, 8(4):101, 2020.
- [10] G. Cybenko, Approximation by superpositions of a sigmoidal function, *Math. Control Signals Systems*, 2(4):303–314, 1989.
- [11] W. E, A proposal on machine learning via dynamical systems, *Commun. Math. Stat.*, 5(1):1–11, 2017.
- [12] O. El Euch and M. Rosenbaum, The characteristic function of rough Heston models, *Math. Finance*, 29(1):3–38, 2019.
- [13] P. Gierjatowicz, M. Sabate-Vidales, D. Siska, L. Szpruch, and Z. Zuric, Robust pricing and hedging via neural stochastic differential equations, *J. Comput. Finance*, 26(3):1–32, 2022.
- [14] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks*, 4(2):251–257, 1991.
- [15] Z. Issa, B. Horvath, M. Lemerrier, and C. Salvi, Non-adversarial training of Neural SDEs with signature kernel scores, in: *Advances in Neural Information Processing Systems*, Vol. 36, Curran Associates, Inc., 11102–11126, 2024.
- [16] H. Kaneko and S. Nakao, A note on approximation for stochastic differential equations, in: *Séminaire de Probabilités, XXII. Lecture Notes in Mathematics*, Vol. 1321, Springer, 155–162, 1988.
- [17] I. Karatzas and S. E. Shreve, *Brownian Motion and Stochastic Calculus*, in: *Graduate Texts in Mathematics*, Vol. 113, Springer, 1991.
- [18] P. Kidger, On neural differential equations, *arXiv:2202.02435*, 2022.
- [19] P. Kidger, J. Foster, X. Li, and T. Lyons, Efficient and accurate gradients for neural SDEs, *arXiv:2105.13493*, 2021.
- [20] P. Kidger and T. Lyons, Universal approximation with deep narrow networks, in: *Proceedings of the 33rd Annual Conference on Learning Theory*, Vol. 125, PMLR, 2306–2327, 2020.
- [21] P. Kidger, J. Morrill, J. Foster, and T. Lyons, Neural controlled differential equations for irregular time series, in: *Advances in Neural Information Processing Systems*, Vol. 33, Curran Associates, Inc., 6696–6707, 2020.
- [22] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980*, 2014.
- [23] E. Kreyszig, *Advanced Engineering Mathematics*, Wiley & Sons, Inc., 1999.
- [24] T. G. Kurtz, Weak and strong solutions of general stochastic models, *Electron. Commun. Probab.*, 19:58, 2014.
- [25] A. P. Kwosek, D. J. Prömel, and J. Teichmann, Universal approximation property of neural stochastic differential equations, *arXiv:2503.16696*, 2025.
- [26] X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud, Scalable gradients for stochastic differential equations, in: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, PMLR, 3870–3882, 2020.
- [27] X. Liu, T. Xiao, S. Si, Q. Cao, S. Kumar, and C.-J. Hsieh, Neural SDE: Stabilizing neural ODE networks with stochastic noise, *arXiv:1906.02355*, 2019.
- [28] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nat. Mach. Intell.*, 3:218–229, 2021.
- [29] E. P. Martins, Estimating the rate of phenotypic evolution from comparative data, *Am. Nat.*, 144(2):193–209, 1994.
- [30] J. Morrill, C. Salvi, P. Kidger, and J. Foster, Neural rough differential equations for long time series, in: *Proceedings of the 38th International Conference on Machine Learning*, PMLR, 7829–7838, 2021.
- [31] B. Øksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer-Verlag, 2003.
- [32] É. Pardoux and P. Protter, Stochastic Volterra equations with anticipating coefficients, *Ann. Probab.*, 18(4):1635–1655, 1990.
- [33] D. J. Prömel and D. Scheffels, On the existence of weak solutions to stochastic Volterra equations, *Electron. Commun. Probab.*, 28:52, 2023.
- [34] D. J. Prömel and D. Scheffels, Stochastic Volterra equations with Hölder diffusion coefficients, *Stochastic Process. Appl.*, 161:291–315, 2023.

- [35] P. Ramachandran, B. Zoph, and Q. V. Le, Searching for activation functions, *arXiv:1710.05941*, 2017.
- [36] P. Reynaud-Bouret and S. Schbath, Adaptive estimation for Hawkes processes; application to genome analysis, *Ann. Statist.*, 38(5): 2781–2822, 2010.
- [37] C. Salvi, M. Lemerrier, and A. Gerasimovics, Neural stochastic PDEs: Resolution-invariant learning of continuous spatiotemporal dynamics, in: *Advances in Neural Information Processing Systems*, Vol. 35, Curran Associates, Inc., 1333–1344, 2022.
- [38] E. Tuzel and A. Erzan, Dissipative dynamics and the statistics of energy states of a Hookean model for protein folding, *J. Stat. Phys.*, 100:405–422, 1999.
- [39] G. E. Uhlenbeck and L. S. Ornstein, On the theory of the Brownian motion, *Phys. Rev.*, 36:823–841, 1930.
- [40] O. Vasicek, An equilibrium characterization of the term structure, *J. Financ. Econ.*, 5(2):177–188, 1977.
- [41] Z. Wang, Existence and uniqueness of solutions to stochastic Volterra equations with singular kernels and non-Lipschitz coefficients, *Statist. Probab. Lett.*, 78(9):1062–1071, 2008.
- [42] N. Yadav, A. Yadav, and M. Kumar, *An Introduction to Neural Network Methods for Differential Equations*, in: *SpringerBriefs in Applied Sciences and Technology*, Springer, 2015.
- [43] X. Zhang, Euler schemes and large deviations for stochastic Volterra equations with singular kernels, *J. Differential Equations*, 244(9):2226–2250, 2008.