

MODELING UNKNOWN STOCHASTIC DYNAMICAL SYSTEM SUBJECT TO EXTERNAL EXCITATION*

YUAN CHEN[†] AND DONGBIN XIU[†]

Abstract. We present a numerical method for learning an unknown nonautonomous stochastic dynamical systems, i.e., stochastic systems subject to time dependent excitation or control signals. Our basic assumption is that the governing equations for the stochastic system are unavailable. However, short bursts of signal-to-state data consisting of certain known excitation signals and their corresponding system responses are available. When a sufficient amount of such signal-to-state data are available, our method is capable of learning the unknown dynamics and producing an accurate predictive model for the stochastic responses of the system subject to arbitrary excitation signals not in the training data. Our method has two key components: (1) a local approximation of the training signal-to-state data to transfer the learning into a parameterized form and (2) a generative model to approximate the underlying unknown stochastic flow map in distribution. After presenting the method in detail, we present a comprehensive set of numerical examples to demonstrate the performance of the proposed method, especially for long-term system predictions.

Key words. data driven modeling, stochastic dynamical system, deep neural network, nonautonomous system

MSC codes. 60H10, 60H35, 62M45, 65C30

DOI. 10.1137/24M1671207

1. Introduction. There has been a growing interest in recovering/discovering unknown dynamical systems from observational data. Most of the existing studies focus on deterministic systems, with methods such as physics-informed neural networks (PINNs) [48, 49], SINDy [7], Fourier neural operator (FNO) [30], computational graph completion [39], sparsity promoting methods [50, 51, 24], flow map learning (FML) [47, 13], to name a few.

Learning unknown stochastic systems is notably more challenging, as the stochastic noises in the systems usually cannot be directly observed. The existing work utilizes Gaussian process [58, 2, 15, 37], polynomial approximations [54, 28], deep neural networks (DNNs) [10, 57, 11, 60, 17, 61], Koopman operators [14, 34], etc. More recently, a stochastic extension of the deterministic flow map learning (FML) approach [47, 13] was proposed. It employs generative models such as GANs (generative adversarial networks) [12] or autoencoders [56] to model the underlying stochasticity. However, most, if not all, of these methods are developed for autonomous systems, where time-invariance (in distribution) holds true and is critical to the method development.

The focus and contribution of this paper is on the learning and modeling of unknown nonautonomous stochastic systems. More specifically, we consider SDEs with unknown governing equations and subject to time dependent external excitation or control signals. Our goal is to develop a method that can capture the stochastic dynamics of the unknown systems by using short-term data consisting of signal-to-state

*Submitted to the journal's Machine Learning Methods for Scientific Computing section June 21, 2024; accepted for publication (in revised form) December 5, 2025; published electronically April 6, 2026.

<https://doi.org/10.1137/24M1671207>

Funding: This work was partially supported by AFOSR grant FA9550-24-1-0237.

[†]Department of Mathematics, The Ohio State University, Columbus, OH 43210 USA (chen.11050@osu.edu, xiu.16@osu.edu).

relations between the excitation signals and their corresponding system responses in the state variables. We remark that there exist some studies on modeling deterministic nonautonomous systems, using methodology such as Dynamic Mode Decomposition (DMD) [31, 43, 26, 23], SINDy [8], Koopman operator [44, 33, 38], FML [45], etc. These methods are not applicable for stochastic nonautonomous systems.

The proposed method in this paper has two key components. First, our method utilizes the observed signal-to-state data to construct an accurate representation of unknown stochastic dynamics of the system. This is accomplished by a generative model that learns the stochastic mapping of the system between two consecutive discrete time steps. The learning of this stochastic flow map is similar to the work of [12, 56], which extended the deterministic FML to stochastic systems. While [12, 56] utilized GANs and autoencoder as the generative model, in this paper we employ conditional normalizing flow (cf. [40]). Normalizing flow has been widely adopted as a probabilistic model for generating data with desired distributions. Its applications include image and video generation [27], statistical inference and sampling [35, 52], reinforcement learning [22], as well as scientific computing [32, 29, 21, 16]. It has also been used to model temporal evolution of density function for a stochastic system [59]. In this paper, we demonstrate that normalizing flow is applicable to modeling unknown stochastic systems and can serve as an effective alternative. The second key component of the proposed method is local parameterization of the excitation signal in the training signal-to-state data. The local parameterization approach was first introduced in [45] for a deterministic nonautonomous system. We adopt the similar idea and extend it to a stochastic system. The approach seeks to parameterize the excitation signals in the training data via a local polynomial over one time step, and subsequently transform the nonautonomous stochastic system into a local Markovian system with homogeneous increments, provided that the unknown stochastic input has stationary and independent increments. This, in turn, transforms the learning problem into a parametric learning between the coefficients of the local polynomials and the system responses. This is a critical component, as it allows the learned system to conduct long-term system predictions under arbitrary excitation signals that are never seen in the training data. Although the proposed method requires a large number of short bursts data, the overall demand for long trajectory data may not be as large, because each burst of the training signal-to-state data consists of merely two time steps of data entries. Once trained, the learned model is able to simulate the unknown stochastic systems for very long time and subject to arbitrary excitation/control signals. A prominent feature of the method, perhaps unique to most of the existing methods, is that it does not require the unknown system to be a classical SDE (with drift and diffusion) and is applicable to learning general non-Gaussian stochastic systems.

2. Setup. Let Ω be an event space, and let T a finite time horizon. We consider a d -dimensional ($d \geq 1$) stochastic process $\mathbf{x}(\omega, t) : \Omega \times [0, T] \mapsto \mathbb{R}^d$ driven by an unknown nonautonomous stochastic dynamical system in the following general form:

$$(2.1) \quad \frac{d\mathbf{x}_t}{dt}(\omega) = \mathbf{f}(\mathbf{x}_t, \mathbf{u}(t), \omega),$$

where $\omega \in \Omega$ represents the random input and $\mathbf{u}(t)$ are external excitation or control signals. Our basic assumption in this paper is that the governing equations of the stochastic system are not available. That is, \mathbf{f} is unknown. Consequently, the solution \mathbf{x}_t cannot be obtained by solving (2.1). However, we assume that the unknown function \mathbf{f} does not explicitly depend on the time variable t .

We assume observation data are available. More specifically, we have signal-to-state time history data between the excitations \mathbf{u} , i.e., the signals, and system state \mathbf{x} , i.e., the state,

$$(2.2) \quad \text{Signal-to-state training data:} \quad \mathbf{u}(t) \rightarrow \mathbf{x}(t).$$

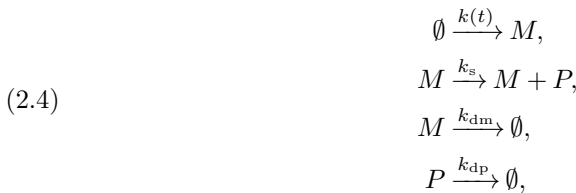
Our goal is to construct a numerical model for the unknown system (2.1) such that it can produce accurate predictions of the system state $\mathbf{x}(t)$ for arbitrarily given excitation signals $\mathbf{u}(t)$ that are not observed in the training data (2.2).

Example 1. As an example of the unknown stochastic system (2.1), let us consider a nonautonomous stochastic differential equation (SDE)

$$(2.3) \quad d\mathbf{x}_t = \mathbf{a}(\mathbf{x}_t, \mu(t))dt + \mathbf{b}(\mathbf{x}_t, \nu(t))d\mathbf{W}_t,$$

where \mathbf{W}_t is m -dimensional ($m \geq 1$) Brownian motion, $\mathbf{a} : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ is drift function, $\mathbf{b} : \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^{d \times m}$ is diffusion function, and $\mu(t)$ and $\nu(t)$ are time-dependent external inputs into the stochastic system. In our setting, both the drift \mathbf{a} and the diffusion \mathbf{b} are unknown, and the excitation signal is $\mathbf{u}(t) = (\mu(t), \nu(t))^T$. Also, the driving Brownian motion \mathbf{W}_t cannot be observed. We emphasize that even though many stochastic systems can be modeled in the form of (2.3), our proposed method does not require such a structure. In fact, the proposed method works for unknown stochastic systems of the general form (2.1), where the random inputs do not need to be Gaussian or Brownian motion.

Example 2. Another example is a reaction network, which can be modeled as a stochastic jump process. For example, consider a gene expression model in the form of biochemical reaction network [53, 6]:



where M is mRNA population and P is protein population, and parameters k 's are reaction rates. Due to external excitation, one of the reaction rates becomes time dependent $k(t)$, whereas the other reaction rates remain constant. The dynamical behavior of M and P can be modeled by an exact Monte Carlo method known as the Gillespie algorithm, or stochastic simulation algorithm (SSA); cf. [20]. In our setting, we assume that the reaction network (2.4) is not known. Subsequently, it is not possible to conduct SSA to predict the system behavior. We assume we can collect time history data of (M, P) subject to certain known external signal $k(t)$. That is, in the form of the signal-to-state data (2.2), the state variables are $\mathbf{x}(t) = (M, P)^T$ and the excitation signal is $\mathbf{u}(t) = k(t)$. Our goal is to model the long-term dynamics of $\mathbf{x}(t)$ as a function of $k(t)$. Note that the stochastic process $\mathbf{x}(t)$ cannot be modeled by the SDE (2.3). It is not driven by Brownian motion and does not follow Gaussian distribution. This will be one of the numerical examples in this paper.

2.1. Problem statement. The method presented in this paper is based on discrete time setting. Let $t_0 < t_1 < \dots$ be discrete time points. For simplicity, we assume the time steps are of uniform length $\Delta = t_{i+1} - t_i, \forall i \geq 0$. Suppose we

observe $N_T \geq 1$ S/S sequences of solution responses subject to input excitations: for $i = 1, \dots, N_T$,

$$(2.5) \quad \left(\mathbf{u} \left(t_0^{(i)} \right), \mathbf{u} \left(t_1^{(i)} \right), \dots, \mathbf{u} \left(t_{L_i}^{(i)} \right) \right) \rightarrow \left(\mathbf{x} \left(t_0^{(i)} \right), \mathbf{x} \left(t_1^{(i)} \right), \dots, \mathbf{x} \left(t_{L_i}^{(i)} \right) \right),$$

where $(L_i + 1)$ is the length of the i th observation sequence. Note that each sequence of the signal-to-state data can cover different time spans. Also, one may have more information about the excitation $\mathbf{u}(t)$ beyond its point values. For example, the analytical form of $\mathbf{u}(t)$ may be known within the time interval $[t_0^{(i)}, t_{L_i}^{(i)}]$ for some sequences.

The objective is to construct a numerical model to predict the system state of (2.3) subject to arbitrary excitation signals. More specifically, given an initial condition \mathbf{x}_0 and excitation signal $\mathbf{u}(t)$ that is not in the training signal-to-state data (2.5), we require the model prediction $\hat{\mathbf{x}}$ to approximate the true system response \mathbf{x} , i.e.,

$$(2.6) \quad \hat{\mathbf{x}}(t_n; \mathbf{x}_0, \mathbf{u}(t)) \stackrel{d}{\approx} \mathbf{x}(t_n; \mathbf{x}_0, \mathbf{u}(t)), \quad n = 1, \dots,$$

where $\stackrel{d}{\approx}$ stands for approximation in distribution. Note that since, in general, the stochastic driving term \mathbf{W}_t cannot be directly observed, a weak approximation, such as approximation in distribution, is typically the most one can achieve from a mathematical point of view.

2.2. Related work and contribution. This method developed in this paper has its foundation in two recent works: flow map learning (FML) for modeling deterministic unknown dynamical systems and its extension to modeling stochastic dynamical systems.

For an unknown deterministic autonomous system, $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$, where $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is unknown, the FML method seeks to approximate the unknown flow map $\mathbf{x}_n = \Phi_{t_n - t_s}(\mathbf{x}_s)$ by using observation data. More specifically, by using data on \mathbf{x} over one time step Δt , the FML method constructs a model

$$\mathbf{x}_{n+1} = \tilde{\Phi}_{\Delta t}(\mathbf{x}_n),$$

where $\tilde{\Phi}_{\Delta t} \approx \Phi_{\Delta t}$ is a numerical approximation of the true flow map over one time step Δt . Once constructed, the FML model can be used as a time marching scheme to predict the system response under a given initial condition. This framework was proposed in [47], with extensions to partially observed systems [19], parametric systems [46], as well as nonautonomous deterministic systems [45].

For learning the autonomous stochastic system, $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \omega(t))$, where $\omega(t)$ represents an unknown stochastic process driving the system. The work of [12] developed stochastic flow map learning (sFML). Assuming the system satisfies time-homogeneous property [36] $\mathbb{P}(\mathbf{x}_{s+\Delta t} | \mathbf{x}_s) = \mathbb{P}(\mathbf{x}_{\Delta t} | \mathbf{x}_0)$, $s \geq 0$, the method uses the observation data on the state variable \mathbf{x} to construct a one-step generative model

$$\mathbf{x}_{n+1} = \mathbf{G}_{\Delta t}(\mathbf{x}_n; \mathbf{z}),$$

where \mathbf{z} is a random variable with known distribution (e.g., standard Gaussian). The function \mathbf{G} , termed stochastic flow map, approximates the conditional distribution $\mathbf{G}_{\Delta t}(\mathbf{x}_s; \mathbf{z}) \approx \mathbb{P}(\mathbf{x}_{s+\Delta t} | \mathbf{x}_s)$. Subsequently, the sFML model becomes a weak approximation, in distribution, to the true stochastic dynamics. Different generative models can be employed under the sFML framework. For example, generative adversarial networks (GANs) are used in [12], and an autoencoder is employed in [56].

The primary contribution of this paper is on the development of data driven modeling for unknown stochastic systems subject to external excitations. To accomplish this, we extend the sFML framework [12], which was developed for a autonomous system, to a nonautonomous stochastic system. To learn the system signal-to-state responses, we employ the local parameterization technique developed for a nonautonomous deterministic system [45]. The method parameterizes the input excitations in the data and transforms the learning problem into learning a parametric dynamical system. For stochastic nonautonomous system considered in this paper, we incorporate the method into a generative model in the sFML framework. In particular, we use normalizing flow as the generative model, which has not been considered in stochastic flow map learning. We shall demonstrate that the newly developed method is highly effective in modeling unknown stochastic systems, when excitations are not present in the training data.

3. Methodology. In this section, we describe the proposed learning method in detail. Since it is difficult to present the mathematical content in the general form (2.1), without knowledge of the equation and the stochastic inputs, our exposition in this section is primarily bases on the more concrete SDE setting (2.3). It shall be clear that the proposed method does not rely on the special structure provided by (2.3).

3.1. Parameterization of inputs. Again, for the convenience of mathematical exposition, let us consider the unknown SDE (2.3) over time interval $[t_n, t_{n+1}]$, $n \geq 0$,

$$(3.1) \quad \mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{a}(\mathbf{x}(s), \mu(s)) ds + \int_{t_n}^{t_{n+1}} \mathbf{b}(\mathbf{x}(s), \nu(s)) d\mathbf{W}(s),$$

which can be written equivalently as

$$(3.2) \quad \begin{aligned} \mathbf{x}(t_{n+1}) = \mathbf{x}(t_n) &+ \int_0^\Delta \mathbf{a}(\mathbf{x}(t_n + \tau), \mu(t_n + \tau)) d\tau \\ &+ \int_0^\Delta \mathbf{b}(\mathbf{x}(t_n + \tau), \nu(t_n + \tau)) d\mathbf{W}(t_n + \tau). \end{aligned}$$

By using the compact notation $\mathbf{u}(t) = (\mu(t), \nu(t))^T$, we now consider the excitation $\mathbf{u}(t)$ in the time interval $[t_n, t_{n+1}]$. Given the information of the excitation in the training data (2.5), we construct a parameterized form

$$(3.3) \quad \mathbf{u}(t)|_{[t_n, t_{n+1}]} \approx \tilde{\mathbf{u}}(\tau; \mathbf{\Gamma}_n) = \sum_{k=1}^m \alpha_n^k p_k(\tau), \quad \tau \in [0, \Delta),$$

where $\{p_k, k = 1, \dots, m\}$ is a set of prescribed analytical basis functions and

$$(3.4) \quad \mathbf{\Gamma}_n = \{\alpha_n^1, \dots, \alpha_n^m\} \in \mathbb{R}^{m\Gamma}$$

are the expansion coefficients. In principle, one can choose any suitable basis functions. Since the time interval $[t_n, t_{n+1}]$ usually has a (very) small step size Δ , it suffices to use low-order polynomials. In fact, low-degree monomials bases, $p_k(\tau) = \tau^{k-1}$, $k \geq 1$, would be sufficient for most problems. When $k = 0$, the parameterization takes form of piecewise constant function; when $k = 1$, piecewise linear function. If higher degree polynomial approximation is needed, one should employ orthogonal polynomials as the basis for numerical stability.

The local parameterization of \mathbf{u} is carried out based on the information one has about the excitations. If the excitations are only known at discrete time instances, as shown in (2.5), then it is natural to utilize piecewise linear function,

$$\tilde{\mathbf{u}}(\tau; \mathbf{\Gamma}_n) = \mathbf{u}(t_n) + \frac{\tau}{\Delta}(\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n)).$$

If more information about $\mathbf{u}(t)$ is available, one can construct a higher degree polynomial. We remark that in the representation, only the values of the excitations \mathbf{u} at t_n and t_{n+1} are needed. The values of the time t_n and t_{n+1} are not required.

It will become clear in the following sections that the critical approximation in the proposed method is the approximation in distribution (2.6), which is accomplished by a stochastic generative model via the observation data (2.5). Here, the numerical errors induced by randomness and finite sampling are dominant and overwhelm the small deterministic error of the local polynomial approximation. We have found no numerical advantage of using polynomials beyond second order.

3.2. Parametric stochastic flow map. By replacing \mathbf{u} by the local polynomial $\tilde{\mathbf{u}}$ (3.3), we transform the system (3.2) into

$$(3.5) \quad \begin{aligned} \tilde{\mathbf{x}}(t_{n+1}) = \tilde{\mathbf{x}}(t_n) &+ \int_0^\Delta \mathbf{a}(\tilde{\mathbf{x}}(t_n + \tau), \tilde{\mu}(\tau; \mathbf{\Gamma}_n)) d\tau \\ &+ \int_0^\Delta \mathbf{b}(\tilde{\mathbf{x}}(t_n + \tau), \tilde{\nu}(\tau; \mathbf{\Gamma}_n)) d\mathbf{W}(t_n + \tau), \end{aligned}$$

where the excitation signals $\mathbf{u} = (\mu, \nu)^T$ has been parameterized by $\tilde{\mathbf{u}}$ via a set of parameters $\mathbf{\Gamma}_n$. Compared to (3.2), the transformed system (3.5) contains possible numerical error introduced by the parameterization of the excitations over the time domain $[0, \Delta)$. The error can be made arbitrarily small for smooth excitations if one uses higher degree polynomials when Δ is sufficiently small.

By using subscripts to denote the time level and letting

$$\tilde{\mathbf{x}}(t_n) = \tilde{\mathbf{x}}_n, \quad d\mathbf{W}_n(\tau) = d\mathbf{W}(t_n + \tau),$$

the parameterized system (3.5) indicates that there exists a mapping

$$(3.6) \quad \tilde{\mathbf{x}}_{n+1} = \mathbf{G}_\Delta(\tilde{\mathbf{x}}_n, d\mathbf{W}_n(\Delta); \mathbf{\Gamma}_n),$$

where \mathbf{G}_Δ is what we shall call *parametric stochastic flow map*, which is parameterized by $\mathbf{\Gamma}_n$. Note that \mathbf{G}_Δ is a general expression stating that the future state $\tilde{\mathbf{x}}_{n+1}$ depends only on the current state $\tilde{\mathbf{x}}_n$ and the incremental noise $d\mathbf{W}_n(\Delta)$. Such a Markovian property holds true for many stochastic systems and is not restricted to SDE (2.3). That is, the existence of \mathbf{G}_Δ does not rely on the existence of the drift \mathbf{a} and diffusion \mathbf{b} . Here, we use (2.3) because it offers a concrete formulation for derivation.

Remark 3.1. It is important to recognize that for the Brownian motion $\mathbf{W}(t)$, or, in general, for Lévy processes (càdlàg stochastic processes with stationary independent increments), the process $d\mathbf{W}_n(\tau)$ is stationary and independent of $\mathbf{W}(t_n)$. Therefore, only the time difference $\Delta = t_{n+1} - t_n$ matters, and the values of t_n and t_{n+1} do not. Consequently, we have suppressed the time variable t_n and t_{n+1} in (3.6).

Remark 3.2. The derivation of (3.6), via the suppression of the time variable t_n , represents a subtle and yet significant step. The parametric stochastic flow map (3.6)

is Markovian, in the sense that $\tilde{\mathbf{x}}_{n+1}$ depends only on the system information at t_n , i.e., on $\tilde{\mathbf{x}}_n$, the local parameters for the excitation signal $\mathbf{\Gamma}_n$, and the stationary local stochastic increment $d\mathbf{W}_n$.

Again, it is clear that the derivation of (3.6) does not rely on the special structure of (2.3), so long as the stochastic input has stationary and independent increments. Therefore, we conclude that the parametric stochastic flow map (3.6) is applicable to the general system (2.1) when the stochastic inputs are Lévy processes.

3.3. Stochastic flow map learning. In this section, we describe our main method of stochastic flow map learning (sFML), which constructs a generative model to approximate the stochastic flow map (3.6) by using the trajectory data (2.5).

3.3.1. Training data. To construct the training data set, we reorganize the original training data set (2.5) into pairs of consecutive time instances, often referred to as snapshots. Since for each of the i th trajectory, $i = 1, \dots, N_T$, we can extract L_i such pairs, there are a total number of $M = L_1 + \dots + L_{N_T}$ signal-to-state data pairs from the data set (2.5):

$$(3.7) \quad \left(\mathbf{u} \left(t_k^{(j)} \right), \mathbf{u} \left(t_{k+1}^{(j)} \right) \right) \rightarrow \left(\mathbf{x} \left(t_k^{(j)} \right), \mathbf{x} \left(t_{k+1}^{(j)} \right) \right), \quad j = 1, \dots, M.$$

Next, we perform the local parameterization procedure from section 3.1 to each pair of the input data $(\mathbf{u}(t_k^{(j)}), \mathbf{u}(t_{k+1}^{(j)}))$ and obtain its parameterization $\mathbf{\Gamma}_k^{(j)}$, $j = 1, \dots, M$, in the form of (3.3). We now have

$$(3.8) \quad \mathbf{\Gamma}_k^{(j)} \rightarrow \left(\mathbf{x} \left(t_k^{(j)} \right), \mathbf{x} \left(t_{k+1}^{(j)} \right) \right), \quad j = 1, \dots, M.$$

Since the values of the time variables do not matter (see Remark 3.1), we again suppress the time variables and write our training data set as

$$(3.9) \quad \mathcal{S}_M = \left\{ \mathbf{\Gamma}_0^{(j)} ; \left(\mathbf{x}_0^{(j)}, \mathbf{x}_1^{(j)} \right) \right\}_{j=1}^M,$$

where M is the total number of the parametric data pairs. In this way, each j th entry of the data set is a trajectory of length two over one time step Δ , starting with its “initial condition” at $\mathbf{x}_0^{(j)}$, ending one step later at $\mathbf{x}_1^{(j)}$, and driven by a known excitation parameterized by $\mathbf{\Gamma}_0^{(j)}$ and an unknown stationary stochastic process $d\mathbf{W}_n(\Delta)$.

3.3.2. Generative model. In ssFML, we seek to approximate the parametric stochastic flow map (3.6) via a recursive generative model in the form of

$$(3.10) \quad \hat{\mathbf{x}}_{n+1} = \hat{\mathbf{G}}_\Delta(\hat{\mathbf{x}}_n, \mathbf{z}_n; \mathbf{\Gamma}_n),$$

where $\mathbf{z} \in \mathbb{R}^{n_z}$ is a random variable of known distribution. Again, since the stationary stochastic process $d\mathbf{W}_n(\Delta)$ in (3.6) is not observed, the constructed sFML model (3.10) is expected to be a weak approximation of (3.6), and in this particular case, approximation in distribution.

In order to construct the sFML model (3.10), we execute the model for one time step over Δ ,

$$(3.11) \quad \hat{\mathbf{x}}_1 = \hat{\mathbf{G}}_\Delta(\hat{\mathbf{x}}_0, \mathbf{z}_0; \mathbf{\Gamma}_0),$$

and utilize the training data set (3.9) to learn the unknown operator $\hat{\mathbf{G}}$. Note that the random variable \mathbf{z}_0 is not in the training date set. In practice, one chooses \mathbf{z}_0

with a known distribution, typically a standard Gaussian, and a specified dimension $n_z \geq 1$. The presence of the random variable \mathbf{z}_0 enables (3.11) to be a stochastic generative model that can produce random realizations. Several methods exist to construct stochastic generative models, e.g., GANs, diffusion model, normalizing flow, autoencoder-decoder, etc. In this paper, we adopt normalizing flow for (3.11). We remark that the use of normalizing flow is not critical to the applicability of the proposed method. Our numerical experiments have indicated that other generative models such as GANs and diffusion model work generally well. We employ normalizing flow primarily because it has not been used in the FML framework before. The property, applicability, and performance comparison of different generative models is a rather complex issue and out of the scope of this paper.

3.3.3. Normalizing flow model. Normalizing flows are generative models that produce tractable distributions to enable efficient and accurate sampling and density evaluation. A normalizing flow is a transformation of a simple probability distribution, e.g., a standard normal, into a more complex distribution by a sequence of diffeomorphism. Let $\mathbf{Z} \in \mathbb{R}^D$ be a random variable with a known and tractable distribution $p_{\mathbf{Z}}$. Let \mathbf{g} be a diffeomorphism, whose inverse is $\mathbf{f} = \mathbf{g}^{-1}$, and $\mathbf{Y} = \mathbf{g}(\mathbf{Z})$. Then using the change of variable formula, one obtain the probability of \mathbf{Y} :

$$p_{\mathbf{Y}}(\mathbf{y}) = p_{\mathbf{Z}}(\mathbf{f}(\mathbf{y})) |\det \mathbf{Df}(\mathbf{y})| = p_{\mathbf{Z}}(\mathbf{f}(\mathbf{y})) |\det \mathbf{Dg}(\mathbf{f}(\mathbf{y}))|^{-1},$$

where $\mathbf{Df}(\mathbf{y}) = \partial \mathbf{f} / \partial \mathbf{y}$ is the Jacobian of \mathbf{f} and $\mathbf{Dg}(\mathbf{z}) = \partial \mathbf{g} / \partial \mathbf{z}$ is the Jacobian of \mathbf{g} . When the target complex distribution $p_{\mathbf{Y}}$ is given, usually as a set of samples of \mathbf{Y} , one chooses \mathbf{g} from a parameterized family \mathbf{g}_{θ} , where the parameter θ is optimized to match the target distribution. Also, to circumvent the difficulty of constructing a complicated nonlinear function \mathbf{g} , one utilizes a composition of (much) simpler diffeomorphisms: $\mathbf{g} = \mathbf{g}_m \circ \mathbf{g}_{m-1} \circ \dots \circ \mathbf{g}_1$. It can be shown that \mathbf{g} remains a diffeomorphism with its inverse $\mathbf{f} = \mathbf{f}_1 \circ \dots \circ \mathbf{f}_{m-1} \circ \mathbf{f}_m$. There exist a large amount of literature on normalizing flows. We refer the interested reader to review articles [25, 41].

In our setting, we seek to construct the one-step generative model (3.11) by using the training data (3.9). Let $\mathbf{z}_0 \in \mathbb{R}^d$ be a random variable with a known distribution. In our approach, we choose \mathbf{z}_0 to be d -dimensional standard normal. Let \mathbf{T}_{θ} be a diffeomorphism with a set of parameters $\theta \in \mathbb{R}^{n_{\theta}}$. Our objective is to find θ such that $\mathbf{T}_{\theta}(\mathbf{z}_0)$ follows the distribution of $\{\mathbf{x}_1^{(j)}\}_{j=1}^M$ in (3.9).

Since the distribution of \mathbf{x}_1 clearly depends on \mathbf{x}_0 and $\mathbf{\Gamma}_0$, we constraint the choice of θ to be a function of \mathbf{x}_0 and $\mathbf{\Gamma}_0$. We define

$$(3.12) \quad \theta = \mathbf{N}(\mathbf{x}_0, \mathbf{\Gamma}_0; \Theta),$$

where \mathbf{N} is a DNN mapping with trainable hyperparameters Θ . No special DNN structure required, and we adopt the straightforward fully connected feedforward DNN for \mathbf{N} . This effectively defines

$$(3.13) \quad \mathbf{x}_1 = \mathbf{T}_{\mathbf{N}(\mathbf{x}_0, \mathbf{\Gamma}_0; \Theta)}(\mathbf{z}_0),$$

where the diffeomorphism \mathbf{T} is effectively parameterized by the trainable hyperparameters Θ of the DNN. Let $\mathbf{S} = \mathbf{T}^{-1}$ be the inverse of \mathbf{T} . We have $\mathbf{z}_0 = \mathbf{S}_{\mathbf{N}_{\Delta}(\mathbf{x}_0, \mathbf{\Gamma}_0; \Theta)}(\mathbf{x}_1)$.

The invertibility of \mathbf{T} allows us to compute

$$(3.14) \quad p(\mathbf{x}_1 | \mathbf{x}_0, \mathbf{\Gamma}_0; \Theta) = p_{\mathbf{z}_0}(\mathbf{S}_{\mathbf{N}(\mathbf{x}_0, \mathbf{\Gamma}_0; \Theta)}(\mathbf{x}_1)) |\det \mathbf{DT}_{\mathbf{N}(\mathbf{x}_0, \mathbf{\Gamma}_0; \Theta)}(\mathbf{S}_{\mathbf{N}(\mathbf{x}_0, \mathbf{\Gamma}_0; \Theta)}(\mathbf{x}_1))|^{-1}.$$

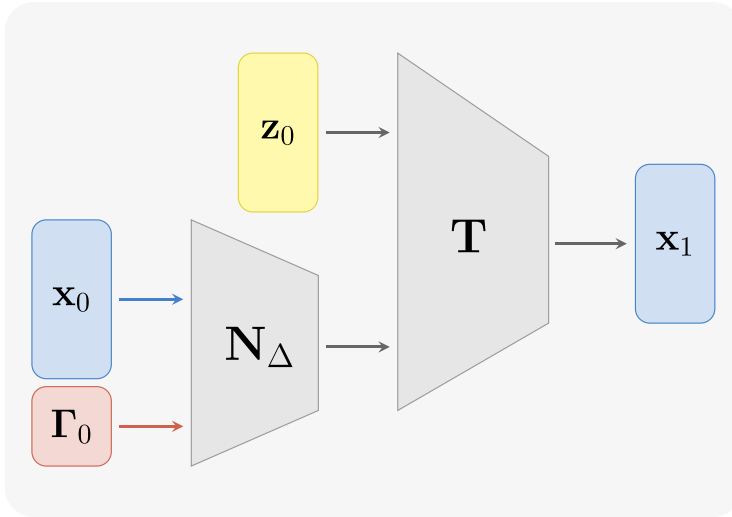


FIG. 1. The DNN model structure for the proposed normalizing flow sFML method (3.13).

The hyperparameters Θ are determined by maximizing the expected log-likelihood, which is accomplished by minimizing its negative as the loss function,

$$\mathcal{L}(\Theta) := -\mathbb{E}_{(\Gamma_0, \mathbf{x}_0, \mathbf{x}_1) \sim p_{\text{data}}} \log(p(\mathbf{x}_1 | \mathbf{x}_0, \Gamma_0; \Theta)),$$

where p_{data} is the distribution from the training data set (3.9) and computed as

$$(3.15) \quad \mathcal{L}(\Theta) = -\sum_{j=1}^M \log \left(p(\mathbf{x}_1^{(j)} | \mathbf{x}_0^{(j)}, \Gamma_0^{(j)}; \Theta) \right).$$

Several designs for the invertible map \mathbf{T} have been developed and studied extensively in the literature. These include, for example, masked autoregressive flow (MAF) [42], real-valued nonvolume preserving (RealNVP) [18], neural ordinary differential equations (Neural ODE) [9], etc. In this paper, we adopt the MAF approach, where the dimension of the parameter θ in (3.12) is set to be $n_\theta = 2d$, where d is the dimension of the dynamical system. For the technical detail of MAF, see [42].

3.4. DNN model structure and system prediction. An illustration of the proposed sFML model structure can be found in Figure 1. This is in direct correspondence of (3.13). Minimization of the loss function (3.15), using the data set (3.9), results in the training of the DNN hyperparameters Θ . Once the training is completed and Θ fixed, (3.13) effectively defines the one-step sFML model (3.11):

$$\mathbf{x}_1 = \mathbf{T}_{\mathbf{N}(\mathbf{x}_0, \Gamma_0)}(\mathbf{z}_0) = \widehat{\mathbf{G}}_\Delta(\mathbf{x}_0, \mathbf{z}_0; \Gamma_0),$$

where we have suppressed the fixed parameter Θ .

Iterative execution of the one-step sFML model allows one to conduct system predictions under excitations that are not in the training data. For a given (new) excitation signal $\mathbf{u}(t) = (\mu(t), \nu(t))^T$, we first conduct its parameterization in the form of (3.3), to obtain its local parameter Γ_n for $[t_n, t_{n+1}]$, for any $n \geq 0$. The sFML system then produces the system prediction, for a given initial condition \mathbf{x}_0 ,

$$(3.16) \quad \mathbf{x}_{n+1} = \widehat{\mathbf{G}}_\Delta(\mathbf{x}_n, \mathbf{z}_n; \Gamma_n), \quad n \geq 0,$$

where \mathbf{z}_n are i.i.d. d -dimensional standard normal random variables.

4. Numerical examples. In this section, we present several numerical tests to demonstrate the performance of our proposed method. After presenting results for learning an Ornstein–Uhlenbeck (OU) process and a nonlinear SDE, we focus on nonlinear SDE systems for long-term predictions. These include a stochastic predator–prey model and a stochastic oscillator with double well potential. In both cases, we study *very* long-term predictions of the learned sFML models. In particular, for the stochastic oscillator, we utilize a periodic excitation signal that is known to generate the well-known “stochastic resonance” phenomenon. At the end of this section, we present learning of a stochastic reaction network and a stochastic partial differential equation (SPDE). The stochastic reaction network is a non-Gaussian stochastic jump process driven by its inherent stochasticity that is not Brownian motion. It demonstrates the applicability of the proposed method to general stochastic system (2.1) beyond the classical SDE (2.3). The SPDE example demonstrates the applicability of the method to a modestly high dimension $d = 30$.

In all the examples, the true stochastic systems are known. However, the true stochastic systems are used only to generate the training data set (3.9). When the true systems follow the SDE form (2.3), we solve them by Euler–Maruyama method with a time step $\Delta = 0.01$. The “initial conditions” \mathbf{x}_0 in (3.9) are sampled uniformly in a domain $I_{\mathbf{x}}$, specified in each example, and the excitations are local polynomials whose coefficients $\mathbf{\Gamma}_0$ are sampled in a domain specified for each example.

In our sFML model, Figure 1, the DNN \mathbf{N} has three layers, each of which with 20 nodes, and utilizes tanh activation function. We employ cyclic learning rate with a base rate 3×10^{-4} and a maximum rate 5×10^{-4} , $\gamma = 0.99999$, and step size 10,000. The cycle is set for every 40,000 training epochs and with a decay scale 0.5. A small weight decay of 0.01 on the gradient updates is also used to help stabilize the training. In our examples, the DNN training is usually conducted for 200,000 ~ 300,000 epochs.

For validation tests, we conduct system predictions by the learned FML for time domains well beyond that of the training data set. We employ mostly periodic excitation signals to ensure that the long-term system behavior remains “interesting”, that is, it does not either decay to zero or blow up. We remark that periodicity of the excitation is not required in the method, as it was never assumed in the derivation of the method.

4.1. Linear SDE with control. We first consider the OU process with control/excitation. Two cases are considered: when the control is in the drift and when the control is in both the drift and the diffusion. Note that since the true equations are not known, one has no information on “where” the excitations operate onto the system. The sFML approach also does not seek to recover the drift or diffusion terms.

4.1.1. OU with drift control. We first consider an OU process,

$$(4.1) \quad dx_t = [-\mu x_t + \alpha(t)] dt + \sigma dW_t,$$

where μ and σ are set as $\mu = 1.0$ and $\sigma = 0.2$, and the control signal $\alpha(t)$ is applied to the drift. The training data set (3.9) is generated by sampling x_0 in $(-2, 2)$ and using Taylor polynomial of degree 2 for the control $\alpha(t)$. This introduces three parameters for $\mathbf{\Gamma}_0$, which are sampled from $(-9, 9)^3$. A total of $M = 120,000$ trajectory pairs are used in the training data set (3.9), where the time step $\Delta = 0.01$.

Once the sFML model (3.10) is trained, we conduct system prediction for up to $T = 10.0$, which requires 1,000 time steps. In Figure 2, we compare some sample trajectory paths produced by the ground truth (left) and the learned sFML model

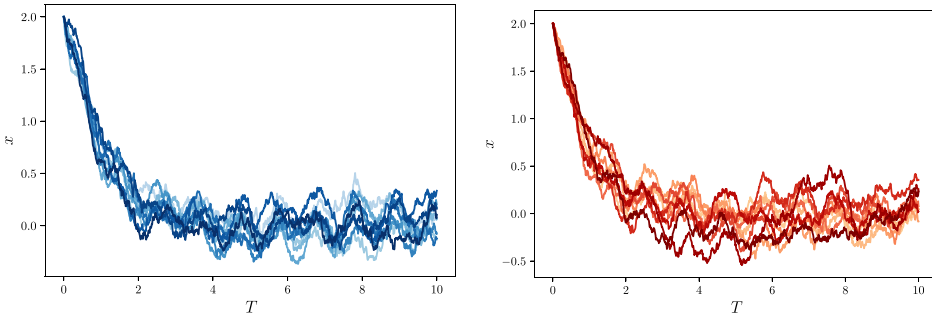


FIG. 2. Sample trajectories of Example 4.1.1 with an initial condition $x_0 = 2.0$ and $\alpha(t) = \frac{1}{2} \sin(6t)$. Left: ground truth. Right: Simulation using the trained sFML model.

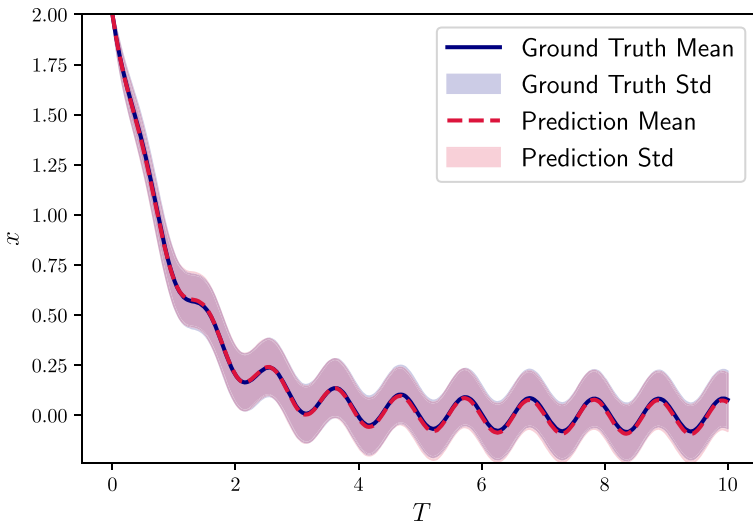


FIG. 3. Mean and standard deviation (STD) of Example 4.1.1 with an initial condition $x_0 = 2.0$ and $\alpha(t) = \frac{1}{2} \sin(6t)$.

(right), with an initial condition $x_0 = 2.0$ and a “new” control signal $\alpha(t) = \frac{1}{2} \sin(6t)$. We observe that the two sets appear visually similar to each other. To further validate the sFML model prediction, we compute the mean and standard deviation of the solution averaged over 10,000 trajectories. The sFML model predictions are shown in Figure 3, along with the reference ground truth. In this figure, lines represent the mean of sample trajectories, while shadow bands depict one standard deviation above and below the mean curve. In Figure 4, we also show the comparison of the solution probability distributions at time $T = 2, 4, 8$. We observe good agreement between the learned sFML model and the true model. This verifies that the sFML model indeed provides an accurate approximation in distribution.

We now present the results under a different setting: the initial condition $x_0 = -1.0$, and the excitation $\alpha(t) = \frac{1}{2} \sin(5t) + \frac{1}{5} \sin(1.5t)$. The sample solution trajectories are shown in Figure 5 and the solution mean and standard deviation averaged over 10,000 trajectories are shown in Figure 6. Again, we observe good agreement between the sFML model prediction and the ground truth.

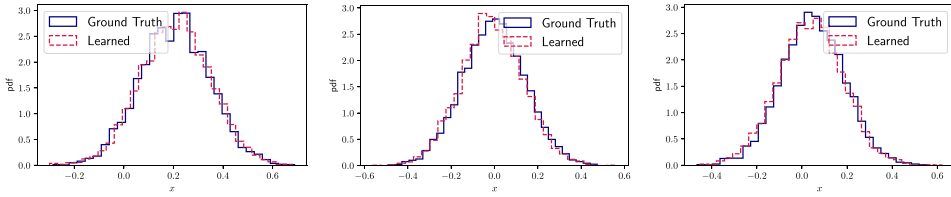


FIG. 4. Comparison of distribution of Example 4.1.1 at $T = 2, 4, 8$ with an initial condition $x_0 = 2.0$ and $\alpha(t) = 0.5 \sin(6t)$.

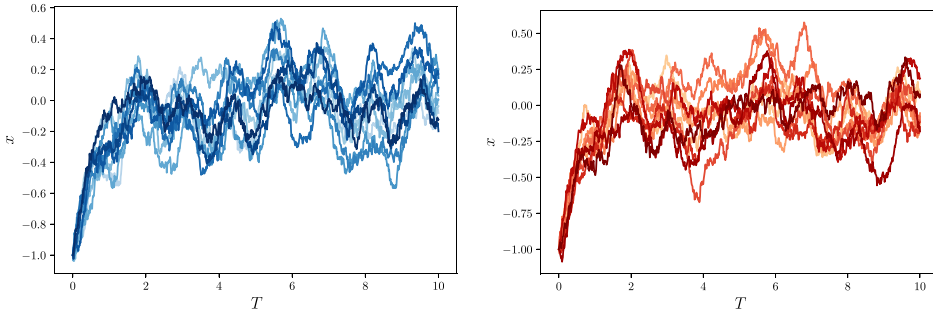


FIG. 5. Sample trajectories of Example 4.1.1 with an initial condition $x_0 = -1.0$ and $\alpha(t) = \frac{1}{2} \sin(5t) + \frac{1}{5} \sin(1.5t)$. Left: ground truth. Right: Simulation using the trained sFML model.

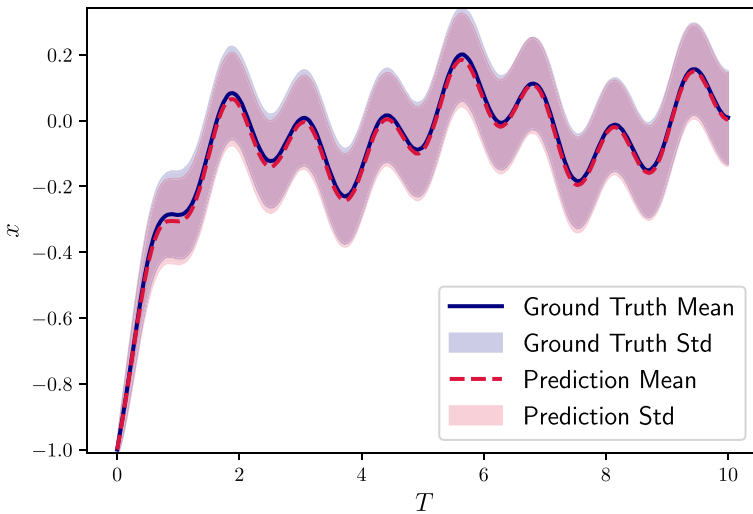


FIG. 6. Mean and standard deviation (STD) of Example 4.1.1 with an initial condition $x_0 = -1.0$ and $\alpha(t) = \frac{1}{2} \sin(5t) + \frac{1}{5} \sin(1.5t)$.

4.1.2. Fully control. We then consider the following OU process with control on both drift and diffusion terms:

$$(4.2) \quad dx_t = [-\mu x_t + \alpha(t)] dt + \beta(t) dW_t,$$

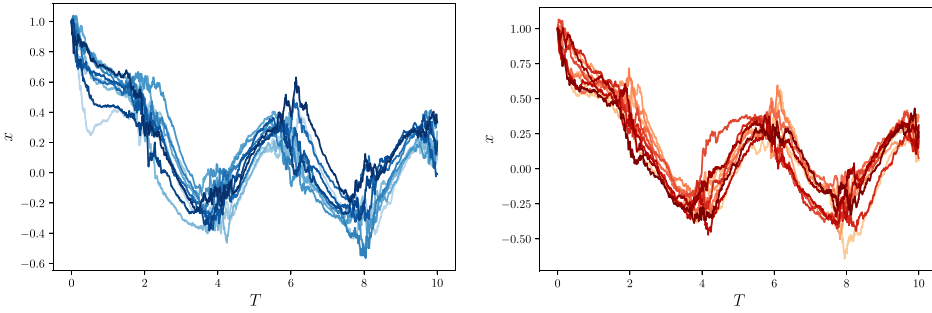


FIG. 7. Sample trajectories of Example 4.1.2 with an initial condition $x_0 = 1.0$, $\alpha(t) = \frac{1}{2} \sin(\frac{\pi}{2}t)$, and $\beta(t) = \frac{1}{10} e^{\cos(\pi t)}$. Left: ground truth. Right: Simulation using the trained sFML model.

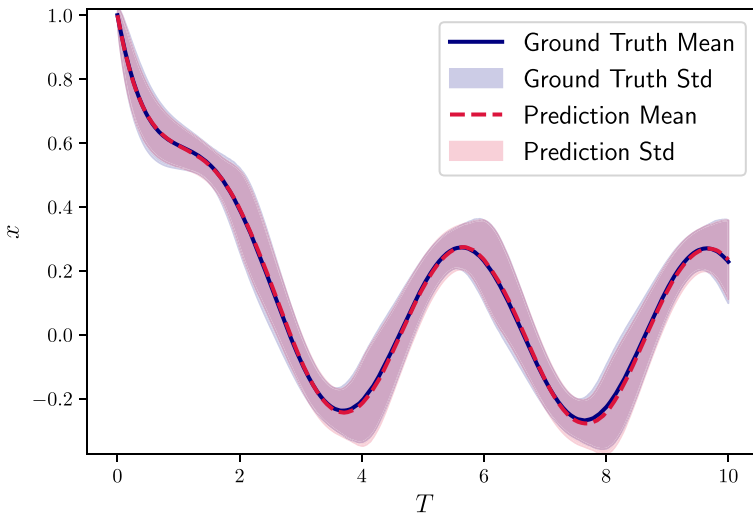


FIG. 8. Mean and standard deviation (STD) of Example 4.1.2 with an initial condition $x_0 = 1.0$, $\alpha(t) = \frac{1}{2} \sin(\frac{\pi}{2}t)$, and $\beta(t) = \frac{1}{10} e^{\cos(\pi t)}$.

where $\mu = 1.0$, and $\alpha(t)$ and $\beta(t)$ are the excitation/control signals. To generate training data, we conduct the local parameterization of $\alpha(t)$ and $\beta(t)$ with 2nd degree Taylor polynomials, resulting in $\Gamma_n \in \mathbb{R}^{n_\Gamma}$, $n_\Gamma = 3 + 3 = 6$. Moreover, we generate 120,000 training data pairs with initial conditions uniformly sampled from $I_x = [-0.8, 1.5]$ and $I_\Gamma = [-0.6, 0.6] \times [-0.8, 0.8] \times [-0.7, 0.7] \times [0.01, 0.35] \times [-0.5, 0.5] \times [-1.55, 0.55]$.

To examine the performance of the learned sFML model, we conduct a simulation with an initial condition $\mathbf{x}_0 = 1.0$ and excitations $\alpha(t) = \frac{1}{2} \sin(\frac{\pi}{2}t)$ and $\beta(t) = \frac{1}{10} e^{\cos(\pi t)}$. (Note that the excitations are not the Taylor polynomials in the training data set.) Some sample solution trajectories are shown in Figure 7. The mean and STD of the solution are shown in Figure 8. And in Figure 9, we also show the comparison of the probability distribution of the solution at $T = 2, 6, 8$. We observe good agreement between the sFML model prediction and the ground truth.

4.2. Nonlinear SDEs with control. We now consider a nonlinear system of SDEs, inspired by an exmple in section 2.3.2 of [55]:

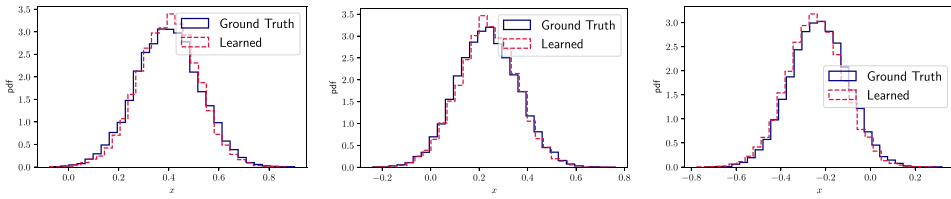


FIG. 9. Comparison of distribution of Example 4.1.2 at $T = 2, 6, 8$ with an initial condition $x_0 = 1.0$, $\alpha(t) = \frac{1}{2} \sin(\frac{\pi}{2}t)$, and $\beta(t) = \frac{1}{10} e^{\cos(\pi t)}$.

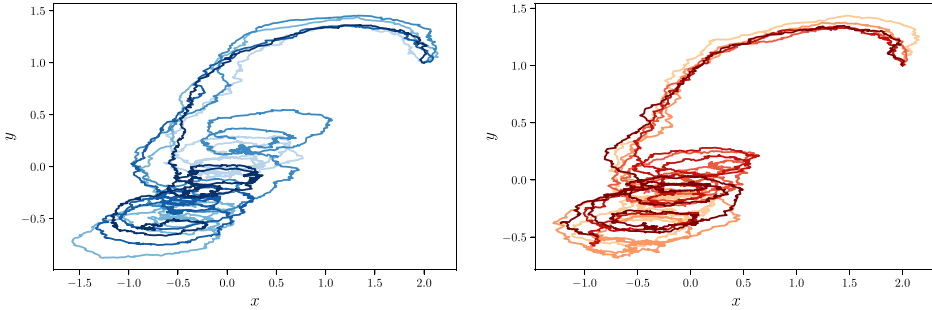


FIG. 10. Sample phase portraits of Example 4.2 with an initial condition $x_0 = 2.0$, $y_0 = 1.0$ for time up to $T = 10$. Left: reference solution. Right: sFML model prediction.

$$(4.3) \quad \begin{cases} dx_t = f(x_t, y_t, t)dt + \sigma_1 dW_1, \\ dy_t = -\mu(y_t - x_t)dt + \sigma_2 dW_2, \end{cases}$$

where W_1 and W_2 are independent Brownian motions, $\mu = 1.0$, $\sigma_1 = 0.2$, $\sigma_2 = 0.05$, and the function f contains a control signal $u(t)$:

$$f(x, y, t) = -y^3 + u(t), \quad u(t) = \sin(\pi t) + \cos(\sqrt{2}\pi t).$$

To generate the training data, we simulate the system with 120,000 sample paths over one time step $\Delta = 0.01$ from initial conditions uniformly in $I_{\mathbf{x}} = [-1.5, 2.0] \times [-1.0, 1.6]$ and under controls by 2nd-degree Taylor polynomials with coefficients sampled from $[-2, 2] \times [-8, 8] \times [-15, 15]$.

For the learned sFML model, we conduct system predictions with an initial condition $x_0 = 2.0$ and $y_0 = 1.0$. In Figure 10, we plot a few sample phase portraits from ground truth (left), as well as from the sFML model prediction (right). They appear to be visually in agreement. The mean and standard deviation of the system prediction by the sFML model are shown in Figure 11, along with those of the true solution. In Figure 12, we also show the comparison of reference and learned density functions of the test trajectory at time $T = 4, 6, 7, 9$. We observe that the sFML model exhibits good accuracy in these predictions.

4.3. Stochastic predator-prey model. We then consider a stochastic Lotka-Volterra system with a time-dependent excitation $u(t)$:

$$(4.4) \quad \begin{cases} dx_t = [x_t - x_t y_t + u(t)] dt + \sigma_1 x_t dW_1, \\ dy_t = (-y_t + x_t y_t) dt + \sigma_2 y_t dW_2, \end{cases}$$

where W_1 and W_2 are independent Brownian motions, and $\sigma_1 = \sigma_2 = 0.05$. The training data are generated by simulating 120,000 solution samples for one step

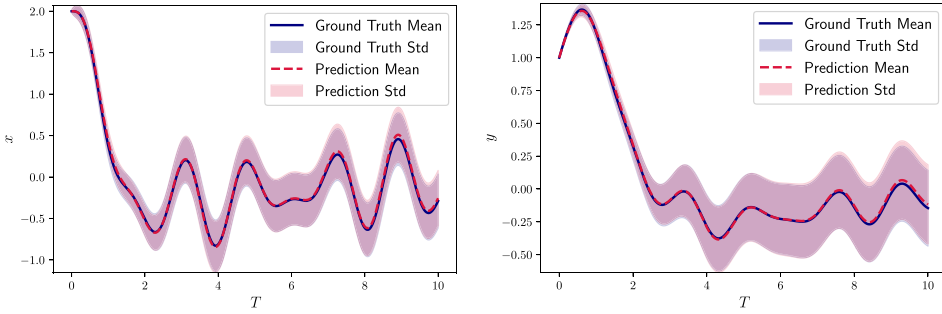


FIG. 11. Mean and standard deviation (STD) of Example 4.2 with an initial condition $x_0 = 2.0$, $y_0 = 1.0$. Left: x . Right: y .

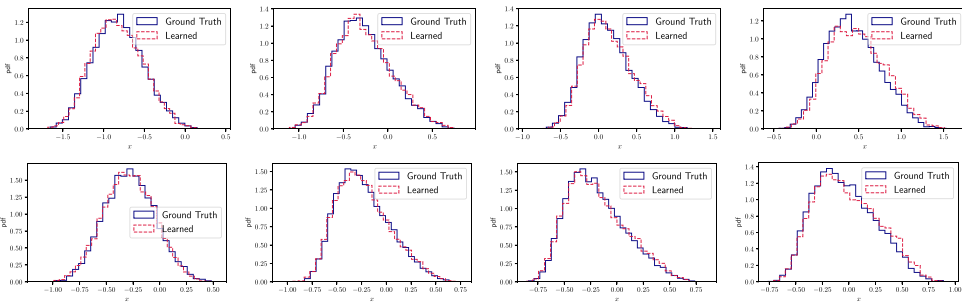


FIG. 12. Comparison of probability distributions of Example 4.2 at $T = 4, 6, 7, 9$ with an initial condition $x_0 = 2.0$, $y_0 = 1.0$. Top row: x . Bottom row: y .

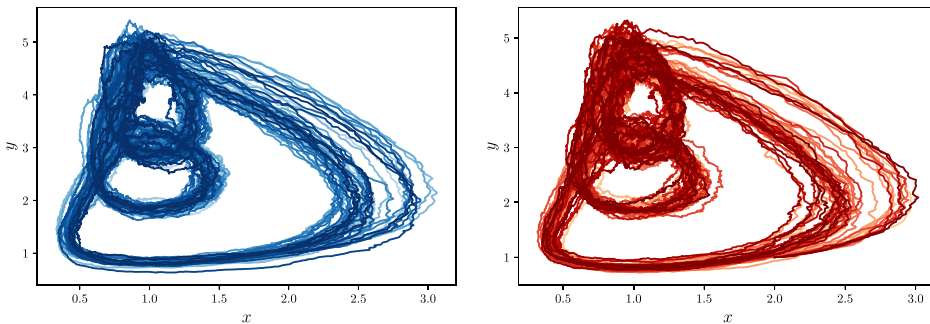


FIG. 13. Phase portrait samples of the Lotka–Volterra system 4.3 with an initial condition $x_0 = 2.0$, $y_0 = 1.0$, and $u(t) = \sin(\frac{t}{3}) + \cos(t) + 2$.

$\Delta = 0.01$, from initial conditions in $I_{\mathbf{x}} = [0.1, 0.35] \times [0.2, 5.5]$ and under excitations of 2nd-degree Taylor polynomials whose coefficients are from $[0.01, 4.2] \times [-1.5, 1.5] \times [-0.7, 0.7]$.

Once we have the trained model, we conduct system prediction with an initial condition $x_0 = 2.0$, $y_0 = 1.0$ and excitation $u(t) = \sin(\frac{t}{3}) + \cos(t) + 2$. We conduct relatively long-term prediction for time up to $T = 80$. (Note that the training data are of length 0.01.) In Figure 13, we plot a few samples of the phase portrait of the system. Good visual agreement between the sFML prediction and the ground truth

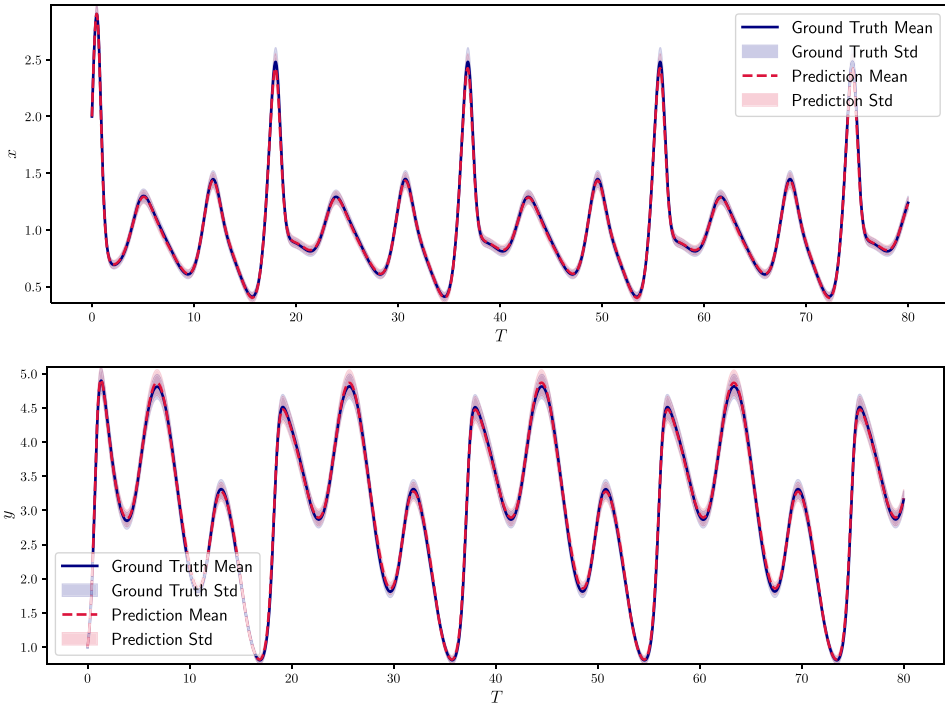


FIG. 14. Mean and standard deviation (STD) of Example 4.3 with an initial condition $x_0 = 2.0$, $y_0 = 1.0$, and $u(t) = \sin(\frac{t}{3}) + \cos(t) + 2$. Upper: x . Lower: y .

can be observed. To examine the accuracy more closely, we present the mean and standard deviation of the system in Figure 14. We observe good predictive accuracy of the sFML model for up to $T = 80$.

4.4. Stochastic resonance. In this section, we consider the following SDE with a double-well potential and excitation,

$$(4.5) \quad dx_t = [x_t - x_t^3 + u(t)] dt + \sigma dW_t,$$

where $\sigma = 0.25$ is a parameter, and $u(t)$ is the excitation. When $V = 0$, there is no excitation to the system. The solution would exhibit random transition between two metastable states $x = -1$ and $x = 1$. The transition probability depends on the parameters σ . When $V \neq 0$, an excitation is exerted to the system. If the excitation is periodic, under the right circumstance the random transition between the two metastable states becomes synchronized with the periodicity of the excitation, resulting in the so-called stochastic resonance; cf. [5, 3, 4].

Here, we demonstrate that the proposed sFML method can accurately model and predict the long-term system behavior using only very a short burst of measurement data. Our data are 30,000 trajectories of one step ($\Delta = 0.01$) length, with initial conditions sampled from $I_{\mathbf{x}} = [-1.6, 1.6]$ and under piecewise constant excitations sampled from $[-0.13, 0.13]$.

Once the sFML model is trained, we conduct system prediction under various excitations. In particular, we choose $u(t) = V \cos(\omega t)$, with $V = 0.12$ and $\omega = 0.001$. These parameters are chosen according to [5], to ensure the occurrence of stochastic resonance. An exceptionally long-term system prediction is conducted by the sFML

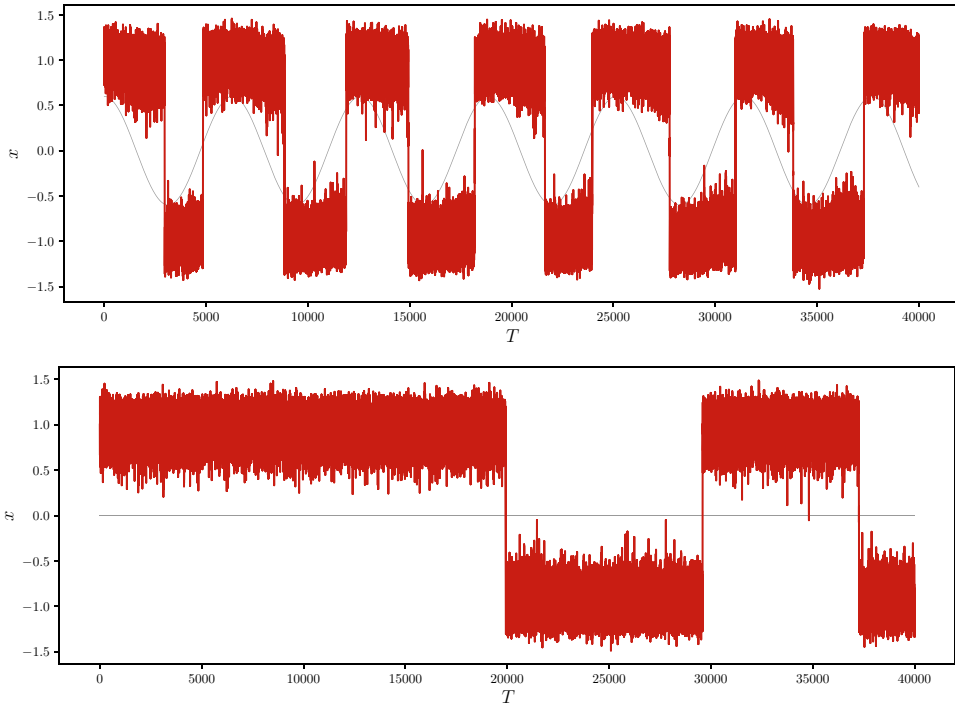


FIG. 15. Sample trajectories of Example 4.4 with an initial condition $x_0 = 1.0$. Top: With the periodic excitation $u(t) = V \cos(\omega t)$, $V = 0.12$, and $\omega = 0.001$. The system exhibits stochastic resonance. Bottom: No excitation case with $u(t) = 0$. The system exhibits random transitions with very small probability $O(10^{-5})$.

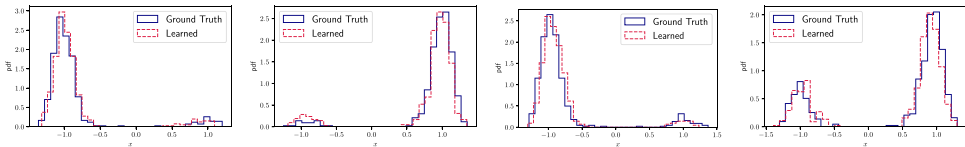


FIG. 16. Comparison of probability distributions of Example 4.4 at $T = 10,000, 20,000, 30,000, 40,000$ with the periodic excitation $u(t) = V \cos(\omega t)$, $V = 0.12$, and $\omega = 0.001$.

model, for time up to $T = 40,000$. The result is shown in the top of Figure 15, where we also plotted the (rescaled) periodic excitation in light grey line in the background. We can clearly observe the synchronization between the random transition and the periodic excitation—the stochastic resonance. For reference, we also conduct the sFML system prediction with $V = 0$, i.e., no excitation. The solution, shown in the bottom of Figure 15, exhibits the expected random transition between the two metastable states. We shall emphasize that in this case the transition probability is very small, $O(10^{-6})$. The learned sFML model is capable of capturing such a small probability event. We shall remark again that the training data are pairwise data separated by one time step. Thus, none of the (long-term) system behaviors can be observed in the training data. In Figure 16, we also show the comparison of reference and learned density functions of the test trajectory at time $T = 10,000, 20,000, 30,000, 40,000$. We observe that the sFML model exhibits good accuracy in these predictions.

4.5. A gene expression model. We consider a gene expression model in the form of biochemical reaction network [53, 6], described as Example 2 in section 2,



where there are two species: M represents mRNA and P represents protein. In this two-stage reaction network model, mRNAs are transcribed in a time-dependent rate $k(t)$ and proteins are translated at a rate of k_s off each mRNA molecule. Meanwhile, mRNAs and proteins degrade at rate k_{dm} and k_{dp} . The dynamical behavior of state variables (M, P) is usually modeled as a stochastic jump process due to the stochasticity caused by a low number of molecules. Stochastic Simulation Algorithm (SSA; cf. [20]) is an exact Monte Carlo method designed for producing realizations of such processes. Though the transitional probability of these processes is non-Gaussian and nearly impossible to be explicitly written, we demonstrate in this example that the proposed sFML method is capable of learning and predicting such systems.

We simulate the system using Modified Next Reaction method [1], which is a variant of the original SSA. The training data are generated via simulating the system up to a short time $\Delta = 0.1$ with randomly sampled initial conditions for $M \leq 10$ and $P \leq 400$. Taylor polynomials of degree 2 are used to conduct local approximation of the excitation signal $k(t)$ within the Δ time interval. The three parameters for $\mathbf{\Gamma}_0$ are sampled uniformly from $[0, 40] \times [-5.3, 5.3] \times [-0.7, 0.7]$. For other parameters, we fix $\Delta = 0.1$, $k_s = 500$, $k_{dm} = 20$, $k_{dp} = 5$. (Note that these parameters are only used to generate the training data. They are not included in the FML model.)

For the learned sFML model, we conduct a system prediction with an initial condition $(M_0, P_0) = (1, 1)$ and $k(t) = 20 + 20\sin(\frac{\pi x}{12})$ up to time $T = 240$. To test the learning accuracy, we fix $(\mathbf{x}_0, \mathbf{\Gamma}_0)$, generate 10,000 samples of \mathbf{z} for $\mathbf{G}_\Delta(\mathbf{x}_0, \mathbf{z}; \mathbf{\Gamma}_0)$ to compare the conditional distribution. In Figure 17, we show the two-dimensional histogram for two scenarios: $\mathbf{x}_0 = (2, 133)$, $\mathbf{\Gamma}_0 = (30, -4.535, -0.335)$, and $\mathbf{x}_0 = (5, 287)$, $\mathbf{\Gamma}_0 = (39.319, 1.355, -0.664)$. The sFML model achieves high accuracy in this non-Gaussian transitional probability distribution case.

In Figure 18, we show one trajectory produced by SSA (left) and the learned sFML model (right). These trajectories reveal similar patterns. To further validate the long time accuracy, we compute the mean and standard deviation with 10,000 trajectories and show the results in Figure 19. Moreover, the probability distributions at time $T = 50, 60, 78, 110$ are shown in Figure 20. We observe good agreement between the sFML model predictions and the ground truth.

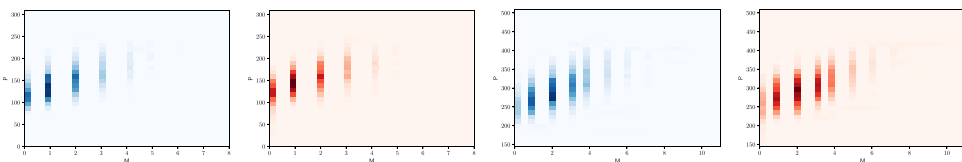


FIG. 17. Two-dimensional histogram of (non-Gaussian) conditional distribution $\mathbf{G}_\Delta(\mathbf{x}_0, \mathbf{z}; \mathbf{\Gamma}_0)$ with fixed $(\mathbf{x}_0, \mathbf{\Gamma}_0)$ of Example 4.5. Left two figures: reference and learned distribution for $\mathbf{x}_0 = (2, 133)$ and $\mathbf{\Gamma}_0 = (30, -4.535, -0.335)$; Right two figures: reference and learned distribution for $\mathbf{x}_0 = (5, 287)$ and $\mathbf{\Gamma}_0 = (39.319, 1.355, -0.664)$.

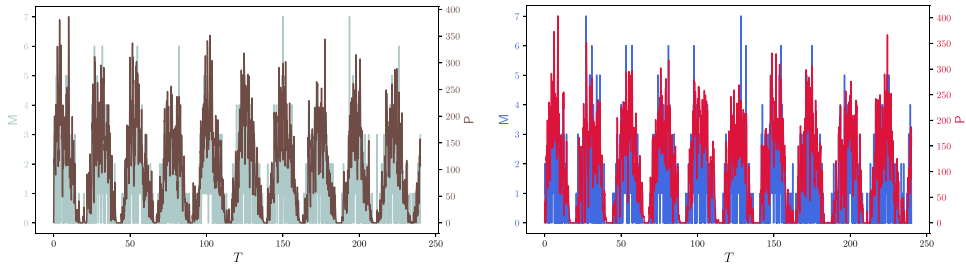


FIG. 18. Sample trajectory for Example 4.5 with an initial condition $(M_0, P_0) = (1, 1)$ and $k(t) = 20 + 20 \sin(\frac{\pi x}{12})$. Left: a reference sample trajectory. Right: a learned sample trajectory.

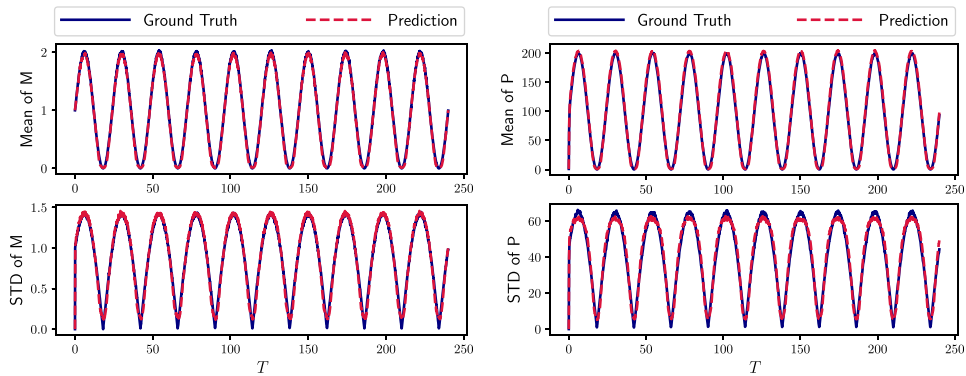


FIG. 19. Mean and standard deviation (STD) for Example 4.5 with an initial condition $(M_0, P_0) = (1, 1)$ and $k(t) = 20 + 20 \sin(\frac{\pi x}{12})$. Left: M . Right: P .

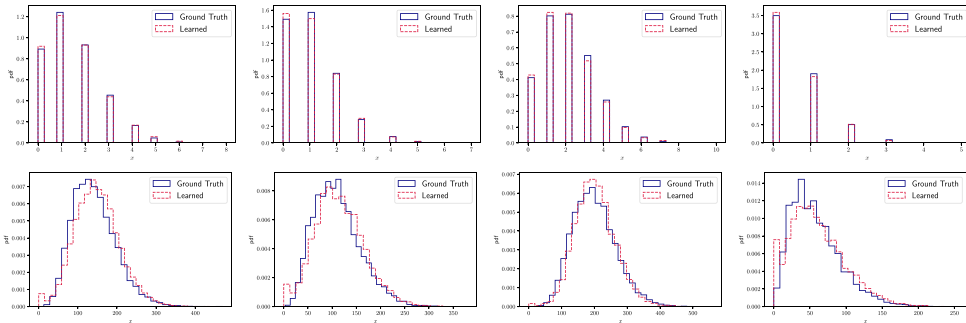


FIG. 20. Comparison of probability distributions of Example 4.5 at $T = 50, 60, 78, 110$ with an initial condition $(M_0, P_0) = (1, 1)$ and $k(t) = 20 + 20 \sin(\frac{\pi x}{12})$. Top row: M . Bottom row: P .

4.6. A stochastic heat equation with source. In this section, we consider learning a one-dimensional stochastic heat equation driven by space-time noise

$$(4.7) \quad \partial_t u = \varepsilon \Delta u + f(x, t) + \sigma \xi(x, t), \quad (x, t) \in (0, 2\pi) \times (0, T],$$

where $\xi(x, t)$ is a space-time white noise, $f(x, t) = \alpha(t)e^{-\frac{(x-p)^2}{q^2}}$, ε, p, q , and σ are model parameters, $\alpha(t)$ is the excitation. Periodic boundary condition is prescribed and $\varepsilon = 0.1$.

We conduct the sFML modeling in modal space, i.e., in Fourier space. Assume the trajectory data of (4.7) are observed at grid points $x_j = jh, j = 0, 1, \dots, N - 1$, where $h = 2\pi/N$. These observational data are then projected onto a finite-dimensional space $\mathbb{V}_N = \text{span}\{e_k\}_{k=1}^N$, where $\{e_k\}_{k=1}^N$ is an orthogonal basis of $\mathbb{V} = L^2_{\text{per}}(0, 2\pi)$. Thus, the solution $u(x, t)$ can be approximated by

$$(4.8) \quad u_N(x, t) = \sum_{i=0}^N c_k(t)e_k(x),$$

where we use the Fourier basis for e_k 's and c_k 's are the Fourier coefficients. We solve (4.7) with Fourier collocation method with $N = 30$. We then apply the proposed sFML method to learn the dynamical behavior of the coefficients $\mathbf{c} = [c_0, \dots, c_N]^T$, where the excitation signal $\alpha(t)$ is represented as 2nd degree Taylor polynomial with the coefficient $\Gamma_0 \in (-1.2, 1.2) \times (-3.5, 3.5) \times (-5, 5)$. A total of 120,000 trajectory pairs are used in the training data set (3.9), where the time step $\Delta = 0.05$.

For validation, we conduct the sFML simulation for $p = 1.0, q = 1.0, \sigma = 0.05$, and with $\alpha(t) = \sin t$. The initial condition is set as $u(x, 0) = \exp(-\sin^2 x) - 1$. In the top of Figure 21, we compute the temporal evolution of the mean and standard deviation of the solution as several locations, averaged over 10,000 trajectories. In the bottom of Figure 21, we present the solution at several different time instances $T = 0.5, 2.5, 5, 8$. In Figure 22, we also show the comparison of the probability distribution of the solution at a few arbitrarily chosen spatial-temporal locations,

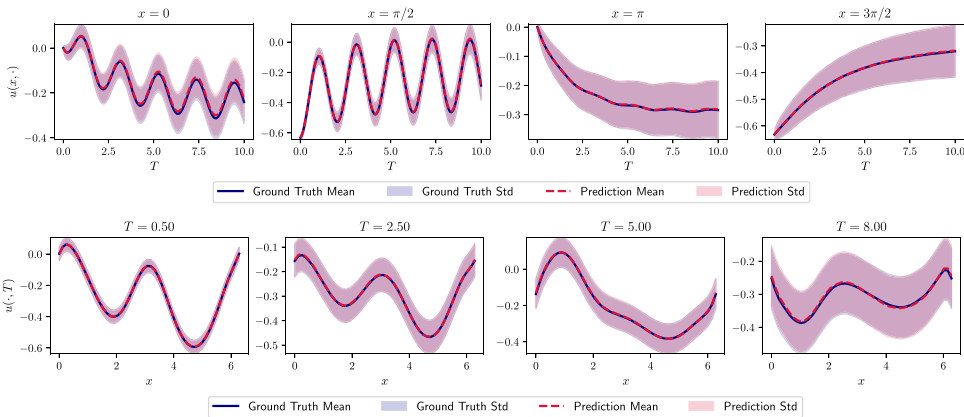


FIG. 21. Mean and standard deviation (STD) of Example 4.7 with an initial condition $u(x, 0) = \exp(-\sin^2 x) - 1$ and $\alpha(t) = \sin t$. Upper: Comparison for $u(x, \cdot)$ with $x = 0, \pi/2, \pi, 3\pi/2$. Lower: Comparison for $u(\cdot, T)$ with $T = 0.5, 2.5, 5, 8$.

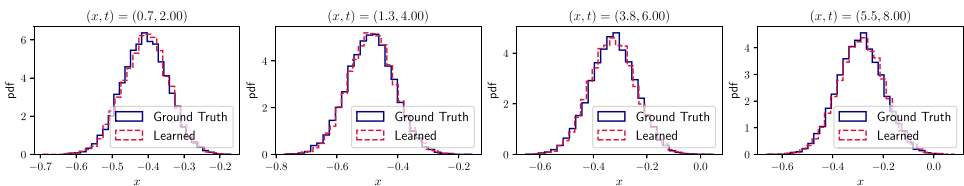


FIG. 22. Comparison of probability distributions of Example 4.7 at $(x, t) = (0.7, 2), (1.3, 4), (3.8, 6), (5.5, 8)$ with an initial condition $u(x, 0) = \exp(-\sin^2 x) - 1$ and $\alpha(t) = \sin t$.

$(x, t) = (0.7, 2), (1.3, 4), (3.8, 6), (5.5, 8)$. We observe good agreement between the sFML model prediction and the ground truth.

5. Conclusion. In this paper, we presented a general numerical framework for modeling unknown nonautonomous stochastic systems by using observed trajectory data. To overcome the difficulties brought by the external time-dependent inputs, we transfer the original system into a local parametric stochastic system. We accomplished this by locally parameterizing the time-dependent external signals on several discrete time points. The resulting stochastic system is then driven by a stationary parametric stochastic flow map. A normalizing flow model is employed to approximate the parametric stochastic flow map. By using a comprehensive set of numerical examples, we demonstrated that the proposed approach is effective and accurate in modeling a variety of unknown stochastic systems. One of the key features of the method is that it is able to produce learned models that can conduct accurate system predictions under external excitations not in the training data and for exceptionally long-term well beyond the time domain of the training data. Although the method works remarkably well for the variety of examples in the paper, we caution that its rigorous numerical analysis such as error estimate is still lacking, largely due to the lack of fundamental results on DNNs. Also, the performance for high-dimensional stochastic systems is unclear. These issues are being pursued at this moment and the results will be reported in future studies.

REFERENCES

- [1] D. F. ANDERSON, *A modified next reaction method for simulating chemical systems with time dependent propensities and delays*, J. Chem. Phys., 127 (2007), <https://doi.org/10.1063/1.2799998>.
- [2] C. ARCHAMBEAU, D. CORNFORD, M. OPPER, AND J. SHAWE-TAYLOR, *Gaussian process approximations of stochastic differential equations*, in Gaussian Processes in Practice, Proceedings of Machine Learning Research 1, N. D. Lawrence, A. Schwaighofer, and J. Quiñonero Candela, eds., PMLR, Bletchley Park, UK, 2007, pp. 1–16, <https://proceedings.mlr.press/v1/archambeau07a.html>.
- [3] R. BENZI, G. PARISI, A. SUTERA, AND A. VULPIANI, *Stochastic resonance in climatic change*, Tellus, 34 (1982), pp. 10–16, <https://doi.org/10.3402/tellusa.v34i1.10782>.
- [4] R. BENZI, G. PARISI, A. SUTERA, AND A. VULPIANI, *A theory of stochastic resonance in climatic change*, SIAM J. Appl. Math., 43 (1983), pp. 565–578, <https://doi.org/10.1137/0143037>.
- [5] R. BENZI, A. SUTERA, AND A. VULPIANI, *The mechanism of stochastic resonance*, J. Phys. A, 14 (1981), pp. L453–L457, <https://doi.org/10.1088/0305-4470/14/11/006>.
- [6] C. G. BOWSER, M. VOLIOTIS, AND P. S. SWAIN, *The fidelity of dynamic signaling by noisy biomolecular networks*, Plos Comput. Biol., 9 (2013), e1002965, <https://doi.org/10.1371/journal.pcbi.1002965>.
- [7] S. L. BRUNTON, J. L. PROCTOR, AND J. N. KUTZ, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proc. Natl. Acad. Sci. USA, 113 (2016), pp. 3932–3937, <https://doi.org/10.1073/pnas.1517384113>.
- [8] S. L. BRUNTON, J. L. PROCTOR, AND J. N. KUTZ, *Sparse identification of nonlinear dynamics with control (sindyc)*, IFAC-PapersOnLine, 49 (2016), pp. 710–715, <https://doi.org/10.1016/j.ifacol.2016.10.249>.
- [9] R. T. Q. CHEN, Y. RUBANOVA, J. BETTENCOURT, AND D. K. DUVENAUD, *Neural Ordinary Differential Equations*, in Advances in Neural Information Processing Systems, Vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds., Curran Associates, 2018, https://proceedings.neurips.cc/paper_files/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf.
- [10] X. CHEN, J. DUAN, J. HU, AND D. LI, *Data-driven method to learn the most probable transition pathway and stochastic differential equation*, Phys. D, 443 (2023), 133559, <https://doi.org/10.1016/j.physd.2022.133559>.
- [11] X. CHEN, L. YANG, J. DUAN, AND G. E. KARNIADAKIS, *Solving inverse stochastic problems from discrete particle observations using the Fokker–Planck equation and physics-informed*

- neural networks*, SIAM J. Sci. Comput., 43 (2021), pp. B811–B830, <https://doi.org/10.1137/20M1360153>.
- [12] Y. CHEN AND D. XIU, *Learning stochastic dynamical system via flow map operator*, J. Comput. Phys., 508 (2024), 112984, <https://doi.org/10.1016/j.jcp.2024.112984>.
- [13] V. CHURCHILL AND D. XIU, *Flow map learning for unknown dynamical systems: Overview, implementation, and benchmarks*, J. Mach. Learn. Model. Comput., 4 (2023), pp. 173–201, <https://doi.org/10.1615/JMachLearnModelComput.2023049717>.
- [14] M. J. COLBROOK, Q. LI, R. V. RAUT, AND A. TOWNSEND, *Beyond expectations: Residual dynamic mode decomposition and variance for stochastic dynamical systems*, Nonlinear Dynam., 112 (2024), pp. 2037–2061, <https://doi.org/10.1007/s11071-023-09135-w>.
- [15] M. DARCY, B. HAMZI, G. LIVIERI, H. OWHADI, AND P. TAVALLALI, *One-shot learning of stochastic differential equations with data adapted kernels*, Phys. D, 444 (2023), 133583, <https://doi.org/10.1016/j.physd.2022.133583>.
- [16] R. DENG, B. CHANG, M. A. BRUBAKER, G. MORI, AND A. LEHRMANN, *Modeling continuous stochastic processes with dynamic normalizing flows*, in Advances in Neural Information Processing Systems, Vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds., Curran Associates, 2020, pp. 7805–7815, https://proceedings.neurips.cc/paper_files/paper/2020/file/58c54802a9fb9526cd0923353a34a7ae-Paper.pdf.
- [17] F. DIETRICH, A. MAKEEV, G. KEVREKIDIS, N. EVANGELOU, T. BERTALAN, S. REICH, AND I. G. KEVREKIDIS, *Learning effective stochastic differential equations from microscopic simulations: Linking stochastic numerics to deep learning*, Chaos, 33 (2023), 023121, <https://doi.org/10.1063/5.0113632>.
- [18] L. DINH, J. SOHL-DICKSTEIN, AND S. BENGIO, *Density estimation using real NVP*, in International Conference on Learning Representations, 2017, <https://openreview.net/forum?id=HkpbH9lx>.
- [19] X. FU, L.-B. CHANG, AND D. XIU, *Learning reduced systems via deep neural networks with memory*, J. Mach. Learn. Model. Comput., 1 (2020), pp. 97–118, <https://doi.org/10.1615/2020034232>.
- [20] D. T. GILLESPIE, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem., 81 (1977), pp. 2340–2361, <https://doi.org/10.1021/j100540a008>.
- [21] L. GUO, H. WU, AND T. ZHOU, *Normalizing field flows: Solving forward and inverse stochastic differential equations using physics-informed flow models*, J. Comput. Phys., 461 (2022), 111202, <https://doi.org/10.1016/j.jcp.2022.111202>.
- [22] T. HAARNOJA, K. HARTIKAINEN, P. ABBEEL, AND S. LEVINE, *Latent space policies for hierarchical reinforcement learning*, in Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research 80, PMLR, J. Dy and A. Krause, eds., 2018, pp. 1851–1860, <https://proceedings.mlr.press/v80/haarnoja18a.html>.
- [23] E. KAISER, J. N. KUTZ, AND S. L. BRUNTON, *Sparse identification of nonlinear dynamics for model predictive control in the low-data limit*, Proc. Roy. Soc. A, 474 (2018), 20180335, <https://doi.org/10.1098/rspa.2018.0335>.
- [24] S. H. KANG, W. LIAO, AND Y. LIU, *IDENT: Identifying differential equations with numerical time evolution*, J. Sci. Comput., 87 (2021), 1, <https://doi.org/10.1007/s10915-020-01404-9>.
- [25] I. KOBYZEV, S. PRINCE, AND M. BRUBAKER, *Normalizing flows: An introduction and review of current methods*, IEEE Trans. Pattern Anal. Mach. Intell., 43 (2021), pp. 3964–3979, <https://doi.org/10.1109/TPAMI.2020.2992934>.
- [26] M. KORDA AND I. MEZIĆ, *Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control*, Automatica, 93 (2018), pp. 149–160, <https://doi.org/10.1016/j.automatica.2018.03.046>.
- [27] V. LAPARRA, G. CAMPS-VALLS, AND J. MALO, *Iterative Gaussianization: From ICA to random rotations*, IEEE Trans. Neural Networks, 22 (2011), pp. 537–549, <https://doi.org/10.1109/TNN.2011.2106511>.
- [28] Y. LI AND J. DUAN, *A data-driven approach for discovering stochastic dynamical systems with non-Gaussian Lévy noise*, Phys. D, 417 (2021), 132830, <https://doi.org/10.1016/j.physd.2020.132830>.
- [29] Y. LI, Y. LU, S. XU, AND J. DUAN, *Extracting stochastic dynamical systems with α -stable Lévy noise from data*, J. Stat. Mech. Theory Exp., 2022 (2022), 023405, <https://doi.org/10.1088/1742-5468/ac4e87>.
- [30] Z. LI, N. B. KOVACHKI, K. AZIZZADENESHELI, B. LIU, K. BHATTACHARYA, A. STUART, AND A. ANANDKUMAR, *Fourier neural operator for parametric partial differential equations*, in International Conference on Learning Representations, 2021, <https://openreview.net/forum?id=c8P9NQVtmmO>.

- [31] H. LU AND D. M. TARTAKOVSKY, *Data-driven models of nonautonomous systems*, J. Comput. Phys., 507 (2024), 112976, <https://doi.org/10.1016/j.jcp.2024.112976>.
- [32] Y. LU, R. MAULIK, T. GAO, F. DIETRICH, I. G. KEVREKIDIS, AND J. DUAN, *Learning the temporal evolution of multivariate densities via normalizing flows*, Chaos, 32 (2022), 033121, <https://doi.org/10.1063/5.0065093>.
- [33] A. MAUROY AND J. GONCALVES, *Koopman-based lifting techniques for nonlinear systems identification*, IEEE Trans. Automat. Control, 65 (2019), pp. 2550–2565, <https://doi.org/10.1109/TAC.2019.2941433>.
- [34] I. MEZIĆ AND A. BANASZUK, *Comparison of systems with complex behavior*, Phys. D, 197 (2004), pp. 101–133, <https://doi.org/10.1016/j.physd.2004.06.015>.
- [35] T. MÜLLER, B. MCWILLIAMS, F. ROUSSELLE, M. GROSS, AND J. NOVÁK, *Neural importance sampling*, ACM Trans. Graph., 38 (2019), pp. 1–19, <https://doi.org/10.1145/3341156>.
- [36] B. ØKSENDAL, *Stochastic differential equations*, in Stochastic Differential Equations, Springer, 2003, pp. 65–84.
- [37] M. OPPER, *Variational inference for stochastic differential equations*, Ann. Phys., 531 (2019), 1800233, <https://doi.org/10.1002/andp.201800233>.
- [38] S. E. OTTO AND C. W. ROWLEY, *Koopman operators for estimation and control of dynamical systems*, Ann. Rev. Control Robotics Autonomous Syst., 4 (2021), pp. 59–87, <https://doi.org/10.1146/annurev-control-071020-010108>.
- [39] H. OWHADI, *Computational graph completion*, Res. Math. Sci., 9 (2022), 27, <https://doi.org/10.1007/s40687-022-00320-8>.
- [40] G. PAPAMAKARIOS, E. NALISNICK, D. J. REZENDE, S. MOHAMED, AND B. LAKSHMINARAYANAN, *Normalizing flows for probabilistic modeling and inference*, J. Mach. Learn. Res., 22 (2021), 57.
- [41] G. PAPAMAKARIOS, E. NALISNICK, D. J. REZENDE, S. MOHAMED, AND B. LAKSHMINARAYANAN, *Normalizing flows for probabilistic modeling and inference*, J. Mach. Learn. Res., 22 (2021), pp. 1–64.
- [42] G. PAPAMAKARIOS, T. PAVLAKOU, AND I. MURRAY, *Masked Autoregressive Flow for Density Estimation*, in Advances in Neural Information Processing Systems, Vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Curran Associates, 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/6c1da886822c67822bcf3679d04369fa-Paper.pdf.
- [43] J. L. PROCTOR, S. L. BRUNTON, AND J. N. KUTZ, *Dynamic mode decomposition with control*, SIAM J. Appl. Dyn. Syst., 15 (2016), pp. 142–161, <https://doi.org/10.1137/15M1013857>.
- [44] J. L. PROCTOR, S. L. BRUNTON, AND J. N. KUTZ, *Generalizing Koopman theory to allow for inputs and control*, SIAM J. Appl. Dyn. Syst., 17 (2018), pp. 909–930, <https://doi.org/10.1137/16M1062296>.
- [45] T. QIN, Z. CHEN, J. D. JAKEMAN, AND D. XIU, *Data-driven learning of nonautonomous systems*, SIAM J. Sci. Comput., 43 (2021), pp. A1607–A1624, <https://doi.org/10.1137/20M1342859>.
- [46] T. QIN, Z. CHEN, J. D. JAKEMAN, AND D. XIU, *Deep learning of parameterized equations with applications to uncertainty quantification*, Int. J. Uncertain. Quantif., 11 (2021), pp. 63–82, <https://doi.org/10.1615/Int.J.UncertaintyQuantification.2020034123>.
- [47] T. QIN, K. WU, AND D. XIU, *Data driven governing equations approximation using deep neural networks*, J. Comput. Phys., 395 (2019), pp. 620–635, <https://doi.org/10.1016/j.jcp.2019.06.042>.
- [48] M. RAISSI, P. PERDIKARIS, AND G. KARNIADAKIS, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., 378 (2019), pp. 686–707, <https://doi.org/10.1016/j.jcp.2018.10.045>.
- [49] M. RAISSI, P. PERDIKARIS, AND G. E. KARNIADAKIS, *Multistep Neural Networks for Data-Driven Discovery of Nonlinear Dynamical Systems*, preprint, <https://arxiv.org/abs/1801.01236>, 2018.
- [50] H. SCHAEFFER AND S. G. MCCALLA, *Sparse model selection via integral terms*, Phys. Rev. E, 96 (2017), 023302, <https://doi.org/10.1103/PhysRevE.96.023302>.
- [51] H. SCHAEFFER, G. TRAN, AND R. WARD, *Extracting sparse high-dimensional dynamics from limited data*, SIAM J. Appl. Math., 78 (2018), pp. 3279–3295, <https://doi.org/10.1137/18M116798X>.
- [52] J. SONG, S. ZHAO, AND S. ERMON, *A-nice-mc: Adversarial Training for MCMC*, in Advances in Neural Information Processing Systems, Vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.,

- Curran Associates, 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/2417dc8af8570f274e6775d4d60496da-Paper.pdf.
- [53] M. VOLIOTIS, P. THOMAS, R. GRIMA, AND C. G. BOWSER, *Stochastic simulation of biomolecular networks in dynamic environments*, Plos Comput. Biol., 12 (2016), e1004923, <https://doi.org/10.1371/journal.pcbi.1004923>.
- [54] Y. WANG, H. FANG, J. JIN, G. MA, X. HE, X. DAI, Z. YUE, C. CHENG, H.-T. ZHANG, D. PU, D. WU, Y. YUAN, J. GONÇALVES, J. KURTHS, AND H. DING, *Data-driven discovery of stochastic differential equations*, Engineering, 17 (2022), pp. 244–252, <https://doi.org/10.1016/j.eng.2022.02.007>.
- [55] E. WEINAN, *Principles of Multiscale Modeling*, Cambridge University Press, 2011.
- [56] Z. XU, Y. CHEN, Q. CHEN, AND D. XIU, *Modeling Unknown Stochastic Dynamical System Via Autoencoder*, preprint, <https://arxiv.org/abs/2312.10001>, 2023.
- [57] L. YANG, C. DASKALAKIS, AND G. E. KARNIADAKIS, *Generative ensemble regression: Learning particle dynamics from observations of ensembles with physics-informed deep generative models*, SIAM J. Sci. Comput., 44 (2022), pp. B80–B99, <https://doi.org/10.1137/21M1413018>.
- [58] C. YILDIZ, M. HEINONEN, J. INTOSALMI, H. MANNERSTROM, AND H. LAHDESMAKI, *Learning stochastic differential equations with gaussian processes without gradient matching*, in Proceedings of the 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2018, pp. 1–6.
- [59] L. YU, R. MAULIK, T. GAO, F. DIETRICH, I. G. KEVREKIDIS, AND J. DUAN, *Learning the temporal evolution of multivariate densities via normalizing flows*, Chaos, 32 (2022), 033121.
- [60] J. ZHANG, S. ZHANG, AND G. LIN, *Multiauto-DeepONet: A Multi-Resolution Autoencoder DeepONet for Nonlinear Dimension Reduction, Uncertainty Quantification and Operator Learning of Forward and Inverse Stochastic Problems*, preprint, <https://arxiv.org/abs/2204.03193>, 2022.
- [61] A. ZHU AND Q. LI, *DynGMA: A Robust Approach for Learning Stochastic Differential Equations From Data*, preprint, <https://arxiv.org/abs/2402.14475>, 2024.