Applied Mathematics for Modern Challenges
Vol. 2, No. 4, December 2024, pp. 348-364
doi:10.3934/ammc.2024018

DATA-DRIVEN EFFECTIVE MODELING OF MULTISCALE STOCHASTIC DYNAMICAL SYSTEMS

Yuan Chen ✉*1 and Dongbin Xiu ✉1

[1]Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA.

Abstract. We present a numerical method for learning the dynamics of slow components of unknown multiscale stochastic dynamical systems. While the governing equations of the systems are unknown, bursts of observation data of the slow variables are available. By utilizing the observation data, our proposed method is capable of constructing a generative stochastic model that can accurately capture the effective dynamics of the slow variables in distribution. We present a comprehensive set of numerical examples to demonstrate the performance of the proposed method.

1. Introduction.

Many mathematical models are established based on complex dynamics with multiple time scales. Understanding and simulating such systems are challenging due to the separation of the scales. One common approach is to establish effective equations, also known as averaged equations, for the slow processes by averaging out the fast variables with respect to their invariant measure, cf. [33, 23]. Another popular way resorts to identifying and modeling certain low-dimensional quantities to represent the dynamics, for example, invariant manifolds [18, 39], transition pathways [20], reaction coordinates [12], etc. More recently, efforts have been made to directly model the observables using data driven approaches. See, for example, [17, 21, 43, 3, 46].

The focus of this paper is on numerical modeling of the effective dynamics of unknown slow-fast multiscale stochastic dynamical systems. More specifically, we assume that observation data of the slow variables are available, while the governing equations for the systems may not. The goal is to construct a numerical model that can accurately capture the slow dynamics, which in turn enables system analysis and prediction of the stochastic dynamics. The proposed method falls into a broader area of discovery of dynamics from observational data, which has attracted growing attention in recent years. Most of the methods focus on deterministic systems, cf. [37, 38, 2, 24, 30, 36, 11]. For stochastic systems, the presence of noises poses new challenges to the design of effective data-driven methods. Several methods have been developed by using Gaussian processes ([44, 1, 13, 29]), or deep neural networks (DNNs) [6, 42, 7, 45, 15], etc. More recently, a framework of stochastic flow map learning ([8]) was proposed. By using generative machine learning models,

2020 *Mathematics Subject Classification.* 60H10, 60H35, 62M45, 65C30.

Key words and phrases. Effective model, multiscale stochastic dynamical system, generative model.

This work was partially supported by AFOSR FA9550-22-1-0011.

*Corresponding author: Yuan Chen.

the approach was shown to be highly effective in modeling unknown stochastic dynamical systems.

The framework of sFML is an extension of FML (flow map learning, cf. [36]) for modeling deterministic systems. The FML approach utilizes data to learn the flow map between state variables recorded on consecutive time steps. Originally developed to learn unknown autonomous dynamical systems, it was extended to learning nonautonomous dynamical systems [34], parametric systems [35], PDEs [10, 40], as well as systems when only a subset of state variables are observed [22]. For unknown stochastic systems, the sFML approach incorporates a generative model in the learned flow map operator, such that the operator becomes stochastic and a weak approximation (in distribution) to the true stochastic flow map. Different generative models can be incorporated, for example, Generative Adversarial Networks (GANs) [6], auto-encoders [41], and normalizing flows [9].

In this work, we adopt the sFML approach for learning the dynamics of the learning slow variables in multiscale stochastic dynamical systems. Once successfully constructed, the sFML model generates stochastic trajectories of the slow state variables for arbitrarily given initial conditions. The learned sFML model is a weak approximation (in distribution) to the true and unknown slow dynamics. Consequently, our model serves as an effective model for the slow dynamics, constructed by data of the slow variables but not the fast variables. Unlike [22, 25] for deterministic dynamical systems with missing variables, our sFML model does not require memory terms. This owes to the distinctive feature of multiscale stochastic dynamics — fast variables are quickly incorporated into ("slaved to") the dynamics of the slow variables, in the form of statistics, which independently contributes to the distribution of slow variables. The reduced system formed by only the slow variables thus becomes approximately Markovain. The impact of the missing dynamics of the fast variable is diminishingly small and presents itself, at best, as noises. For the generative model of the sFML model, we utilize conditional normalizing flow (cf. [31]) model to approximate the stochasticity of the slow system. Other kind of generative models can also be considered in the sFML model, as shown the [8, 41, 9].

2. Setup.

Let Ω be an event space and T be a finite time horizon, we consider a d-dimensional stochastic system $\mathbf{u}^\varepsilon : \Omega \times [0,T] \mapsto \mathbb{R}^d$. The state variables $\mathbf{u}$ can be separated into slow variables $\mathbf{x} \in \mathbb{R}^\ell$ and fast variable $\mathbf{y} \in \mathbb{R}^{d-\ell}$, and are driven by an unknown stochastic differential equations:

$$\begin{cases} d\mathbf{x}_t^\varepsilon = \mathbf{f}\left(\mathbf{x}_t^\varepsilon, \mathbf{y}_t^\varepsilon\right) + \boldsymbol{\sigma}(\mathbf{x}_t^\varepsilon, \mathbf{y}_t^\varepsilon) d\mathbf{W}_t^1, \\ d\mathbf{y}_t^\varepsilon = \dfrac{1}{\varepsilon}\mathbf{g}\left(\mathbf{x}_t^\varepsilon, \mathbf{y}_t^\varepsilon\right) + \dfrac{1}{\sqrt{\varepsilon}}\boldsymbol{\beta}\left(\mathbf{x}_t^\varepsilon, \mathbf{y}_t^\varepsilon\right) d\mathbf{W}_t^2, \end{cases} \tag{1}$$

where $\mathbf{f}$ and $\boldsymbol{\sigma}$, $\mathbf{g}$ and $\boldsymbol{\beta}$ are drift and diffusion functions, $\mathbf{W}_t^i$ are (independent) m_i-dimensional Brownian motions, $m_1 + m_2 \geq 1$. The parameter $0 < \varepsilon \ll 1$ measures the separation of time-scale in the system. We emphasize that the form of equations as well as the Brownian motion noises are unknown.

2.1. Objective.

Although (1) is *unknown*, in the sense that the functions $\mathbf{f}$, $\boldsymbol{\sigma}$, $\mathbf{g}$ and $\boldsymbol{\beta}$ are *unknown*, we assume that historical data of the slow variable $\mathbf{x}_t^\varepsilon$ are available. More specifically, let $t_0 < t_1 < ...$ be discrete time instants. We observe $N_T \geq 1$ pieces of time history data of the slow variables $\mathbf{x}_t^\varepsilon$:

$$\left(\mathbf{x}^\varepsilon(t_0^{(i)}), \mathbf{x}^\varepsilon(t_1^{(i)}), ..., \mathbf{x}^\varepsilon(t_{L_i}^{(i)})\right), \qquad i = 1, 2, ... \tag{2}$$

where for simplicity we assume a constant time lag $\Delta = t_i - t_{i-1} \sim \mathcal{O}(1)$ and the length of all pieces of data is the same, $L_i = L$, $\forall i$. We remark that data of the fast variables are not required in this work.

Our goal is to construct a numerical model involving *only* the slow variables $\mathbf{x}_t^\varepsilon$ for the dynamics of the slow variables of the multiscale system (1), without the involvement of the fast variables $\mathbf{y}_t^\varepsilon$. The model predictions $\widehat{\mathbf{x}}$ shall be an approximation, in distribution, to the true (and unknown) slow dynamics, i.e.,

$$\widehat{\mathbf{x}}(t_i; \mathbf{x}_0) \overset{d}{\approx} \mathbf{x}^\varepsilon(t_i; \mathbf{x}_0), \qquad i = 1, 2, \ldots \tag{3}$$

for arbitrarily given initial condition $\mathbf{x}_0$.

2.2. **Related work.** The current work is an extension of the FML (flow map learning) methodology for modeling deterministic unknown dynamical systems and its stochastic extension sFML for modeling unknown 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 $\mathbf{f}$ is unknown. The FML framework seeks to approximate the unknown flow map $\mathbf{x}_n = \mathbf{\Phi}_{t_n - t_s}(\mathbf{x}_s)$ by using historical trajectory data. Specifically, FML method fits a model

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

where $\widetilde{\mathbf{\Phi}}_{\Delta t} \approx \mathbf{\Phi}_{\Delta t}$ is a numerical approximation of the true flow map. Once trained, the FML model can be used as a time-marching scheme to predict the system response under new initial conditions. This framework was proposed in [36], with extensions to parametric systems [35], partially observed dynamical systems [22], as well as non-autonomous deterministic system [34].

For learning stochastic dynamics, $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \omega(t))$, where $\omega(t)$ represents an unknown stochastic process driving the system. The work of [8] developed stochastic flow map learning (sFML). Assuming the system satisfies time-homogeneous property $\mathbb{P}(\mathbf{x}_{s+\Delta t}|\mathbf{x}_s) = \mathbb{P}(\mathbf{x}_{\Delta t}|\mathbf{x}_0)$, $s \geq 0$ (c.f. [28]), the method uses 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 $\mathbf{G}_{\Delta t}(\mathbf{x}_s; \mathbf{z}) \sim \mathbb{P}(\mathbf{x}_{s+\Delta t}|\mathbf{x}_s)$. 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, e.g., generative adversarial networks (GANs) [8], autoencoder [41], etc. Recently, it was extended to stochastic systems subjected to external excitation using normalizing flow [9].

2.3. **Contribution.** The contribution of the current work is on the construction of data-driven effective model for the slow variables of the multiscale stochastic system (1). Under the assumption that the fast variables quickly reach their stationary measure and become "slaved" to the slow variables, we develop a sFML model that involves only the slow variables for accurate predictions of the effective dynamics of the original system. Such a modeling approach is highly useful, as in practice one often only has observation data on the slow variables because the fast variables are difficult, if not impossible, to observe. The proposed work can be considered as an extension of the sFML modeling methodology from general stochastic system to multiscale stochastic system.

3. Method description.

In this section, we describe the proposed method in detail.

3.1. Mathematical motivation.

Let us consider the unknown full system (1) over a time interval $[t_n, t_{n+1}]$, $n \geq 0$,

$$\mathbf{x}^{\varepsilon}_{n+1} = \mathbf{x}^{\varepsilon}_{n} + \int_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{x}^{\varepsilon}(s), \mathbf{y}^{\varepsilon}(s))ds + \int_{t_n}^{t_{n+1}} \boldsymbol{\sigma}(\mathbf{x}^{\varepsilon}(s), \mathbf{y}^{\varepsilon}(s))d\mathbf{W}^1_n(s). \tag{4}$$

For the multiscale system with $\varepsilon \ll 1$, there exists a time scale $\tau \sim O(\varepsilon)$ such that for $t > \tau$ the fast variable $\mathbf{y}$ becomes slaved to the slow variable $\mathbf{x}$, i.e.,

$$\mathbf{y}^{\varepsilon}(t) \approx \mathbf{y}(\mathbf{x}^{\varepsilon}(t)) \sim \mathcal{P}_{\mathbf{x}}, \qquad t > \tau, \tag{5}$$

where $\mathcal{P}_{\mathbf{x}}$ is its (conditioned) invariant measure that is a function of $\mathbf{x}$ only. Therefore, on any time scale larger than $\tau \sim O(\varepsilon)$, there exists an unknown mapping $\mathbf{G}$ such that

$$\mathbf{x}^{\varepsilon}_{n+1} \approx \mathbf{G}(\mathbf{x}^{\varepsilon}_{n}, d\mathbf{W}_n(\Delta)), \tag{6}$$

where $d\mathbf{W}_n(s) = (d\mathbf{W}^1(t_n + s); d\mathbf{W}^2(t_n + s))$ is the independent increment of the Wiener process. The above argument is illustrated in Figure 1.

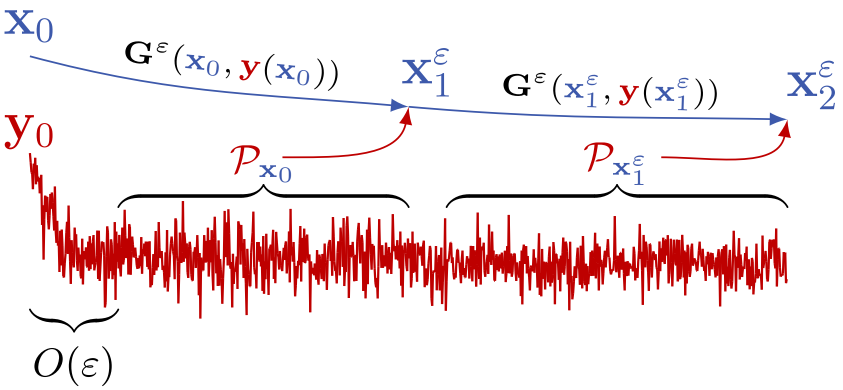
$\mathbf{x}_0$
$\mathbf{G}^{\varepsilon}(\mathbf{x}_0, \mathbf{y}(\mathbf{x}_0))$
$\mathbf{x}^{\varepsilon}_1$
$\mathbf{G}^{\varepsilon}(\mathbf{x}^{\varepsilon}_1, \mathbf{y}(\mathbf{x}^{\varepsilon}_1))$
$\mathbf{x}^{\varepsilon}_2$
$\mathbf{y}_0$
$\mathcal{P}_{\mathbf{x}_0}$
$\mathcal{P}_{\mathbf{x}^{\varepsilon}_1}$
$O(\varepsilon)$

FIGURE 1. Illustration of how the fast variables $\mathbf{y}$ quickly become slaved to the slow variables $\mathbf{x}$ in multiscale system.

3.2. Stochastic flow map learning.

Since the slow variables $\mathbf{x}$ follow the unknown mapping (6), we now seek to construct a stochastic flow map learning (sFML) model in the form of a generative model

$$\widehat{\mathbf{x}}_{n+1} = \widehat{\mathbf{G}}(\widehat{\mathbf{x}}_n, \mathbf{z}_n), \tag{7}$$

where $\mathbf{z}_n \in \mathbb{R}^{n_z}$ is a random variable of known distribution, e.g., a standard Gaussian, n_z is the dimension of the random input. Our objective is to ensure this is a weak approximation, in distribution, to the unknown mapping (6). That is, given an exact slow variable $\mathbf{x}^{\varepsilon}_n$, the sFML model shall produce an approximation of the slow variable t_{n+1},

$$\widetilde{\mathbf{x}}^{\varepsilon}_{n+1} = \widehat{\mathbf{G}}(\mathbf{x}^{\varepsilon}_n, \mathbf{z}_n) \overset{d}{\approx} \mathbf{x}^{\varepsilon}_{n+1}. \tag{8}$$

Note that the approximation is in distribution, a weak form of approximation.

3.2.1. *Learning method and training data.* In order to construct the sFML model (7) satisfying (8), we execute the sFML model for one time step over Δ,

$$\widehat{\mathbf{x}}_1 = \widehat{\mathbf{G}}(\widehat{\mathbf{x}}_0, \mathbf{z}_0), \tag{9}$$

and utilize the data set (2) to learn the unknown operator $\widehat{\mathbf{G}}$.

To construct the training data, we reorganize the trajectory data (2) into pairs, separated by the constant time step Δ,

$$\left(\mathbf{x}^\varepsilon(t_{k-1}^{(i)}), \mathbf{x}^\varepsilon(t_k^{(i)})\right), \qquad k = 1, ...L, \quad i = 1, \dots, N_T, \tag{10}$$

which contain a total of $M = N_T L$ such data pairs. We rename each pair using indices 0 and 1, and re-organize the training data set into a set of such pairwise data entries,

$$\left(\mathbf{x}_0^{\varepsilon\,(i)}, \mathbf{x}_1^{\varepsilon\,(i)}\right), \qquad i = 1, \dots, M. \tag{11}$$

In other words, the training data set is comprised of M numbers of very short trajectories of only two entries. Each of the i-th trajectory, $i = 1, \dots, M$, starts with an "initial condition" $\mathbf{x}_0^{\varepsilon\,(i)}$ and ends a single time step Δ later at $\mathbf{x}_1^{\varepsilon\,(i)}$. The one-step sFML model (9) is then applied to this training data set to accomplish

$$\widetilde{\mathbf{x}}_1^\varepsilon = \widehat{\mathbf{G}}(\mathbf{x}_0^\varepsilon, \mathbf{z}_0) \overset{d}{\approx} \mathbf{x}_1^\varepsilon. \tag{12}$$

The presence of the random variable $\mathbf{z}_0$ enables (9) to be a generative model that can produce random realizations. Several methods exist to construct stochastic generative models, e.g., GANs, diffusion model, normalizing flow, autoencoder, etc. In this paper, we adopt normalizing flow for (9).

3.2.2. *Normalizing flow model.* Normalizing flows are generative models that can produce efficient and accurate sampling and density evaluation. A normalizing flow is designed to transform a simple probability distribution, e.g., a standard Gaussian, into a more complex distribution by a sequence of diffeomorphisms. 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 $\mathbf{Y} = \mathbf{g}(\mathbf{Z})$. By using the change of variable formula, one obtains the probability of $\mathbf{Y}$:

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

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 to find $\mathbf{g}$ from a parameterized family $\mathbf{g}_\theta$, 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: $\mathbf{g} = \mathbf{g}_m \circ \mathbf{g}_{m-1} \circ \cdots \circ \mathbf{g}_1$. It can be shown that $\mathbf{g}$ remains a diffeomorphism with its inverse $\mathbf{f} = \mathbf{f}_1 \circ \cdots \circ \mathbf{f}_{m-1} \circ \mathbf{f}_m$. There exists a large amount of literature on normalizing flows. We refer interested readers to review articles such as [31].

In our setting, we seek to construct the one-step generative model (9) by using the training data (11) to accomplish (12). We choose $\mathbf{z}_0$ to be a ℓ-dimensional standard Gaussian since the slow variables are in ℓ dimension. Let $\mathbf{T}_\theta$ 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 $\mathbf{x}_1^\varepsilon$ given by the samples $\{\mathbf{x}_1^{\varepsilon\,(i)}\}_{i=1}^M$ in (11).

Since the distribution of $\mathbf{x}_1^\varepsilon$ clearly depends on the unknown dynamics starting at $\mathbf{x}_0^\varepsilon$, we constraint the choice of θ to be a function of $\mathbf{x}_0^\varepsilon$, i.e.,

$$\theta = \mathbf{N}(\mathbf{x}_0^\varepsilon; \Theta), \tag{13}$$

where $\mathbf{N}$ is a DNN mapping with trainable hyper-parameters Θ. This effectively defines

$$\mathbf{x}_1^\varepsilon = \mathbf{T}_{\mathbf{N}(\mathbf{x}_0^\varepsilon;\Theta)}(\mathbf{z}_0), \tag{14}$$

where the diffeomorphism $\mathbf{T}$ is parameterized by the trainable hyper-parameters Θ of the DNN. Let $\mathbf{S} = \mathbf{T}^{-1}$ be the inverse of $\mathbf{T}$. We have $\mathbf{z}_0 = \mathbf{S}_{\mathbf{N}(\mathbf{x}_0^\varepsilon;\boldsymbol{\Theta})}(\mathbf{x}_1^\varepsilon)$. The invertibility of $\mathbf{T}$ allows us to compute:

$$p(\mathbf{x}_1^\varepsilon|\mathbf{x}_0^\varepsilon;\boldsymbol{\Theta}) = p_{\mathbf{z}_0}\left(\mathbf{S}_{\mathbf{N}(\mathbf{x}_0^\varepsilon;\Theta)}(\mathbf{x}_1^\varepsilon)\right)\left|\det \mathbf{D}\mathbf{T}_{\mathbf{N}(\mathbf{x}_0^\varepsilon;\Theta)}(\mathbf{S}_{\mathbf{N}(\mathbf{x}_0^\varepsilon;\Theta)}(\mathbf{x}_1^\varepsilon))\right|^{-1}. \tag{15}$$

The hyperparameters Θ are determined by maximizing the expected log-likelihood (15), which is accomplished by minimizing its negative as the loss function,

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

where $p_{\mathtt{data}}$ is the distribution from the training data set (11) and computed as

$$\mathcal{L}(\Theta) = -\sum_{j=1}^{M} \log\left(p(\mathbf{x}_1^{\varepsilon\,(j)}|\mathbf{x}_0^{\varepsilon\,(j)};\Theta)\right). \tag{16}$$

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) [32], real-valued non-volume preserving (RealNVP) [16], neural ordinary differential equations (Neural ODE) [5], etc. In this paper, we adopt the MAF approach, where the dimension of the parameter θ in (13) is set to be $n_\theta = 2\ell$, where ℓ is the dimension of the slow variables. For the technical detail of MAF, see [32].

3.3. **DNN model structure and system prediction.** An illustration of the proposed sFML model structure can be found in Figure 2. This is in direct correspondence of (14). Minimization of the loss function (16), using the data set (11), results in the training of the DNN hyperparameters Θ. Once the training is completed and Θ fixed, (14) effectively defines the one-step sFML model (9):

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

where we have suppressed the fixed parameter Θ.

The sFML system then produces the system prediction iteratively, for a given initial condition of the slow variable $\mathbf{x}_0^\varepsilon$,

$$\begin{cases} \widehat{\mathbf{x}}_0^\varepsilon = \mathbf{x}_0^\varepsilon, \\ \widehat{\mathbf{x}}_{n+1}^\varepsilon = \widehat{\mathbf{G}}(\widehat{\mathbf{x}}_n^\varepsilon, \mathbf{z}_n), \qquad n \geq 0, \end{cases} \tag{17}$$

where $\mathbf{z}_n$ are i.i.d. ℓ-dimensional standard normal random variables. Courtesy of the construction requirement (8), the sFML prediction is expected to be a weak approximation in distribution to the true stochastic dynamics, i.e., $\widehat{\mathbf{x}}_n^\varepsilon \overset{d}{\approx} \mathbf{x}_n^\varepsilon$, $n \geq 0$. We remark again that the sFML model (17) requires only the slow variables. The fast variables are not required during both the training and the prediction of the model. Therefore, the sFML model (17) is an effective model for the evolution of the slow variables of the original (unknown) multiscale stochastic dynamical system (1).

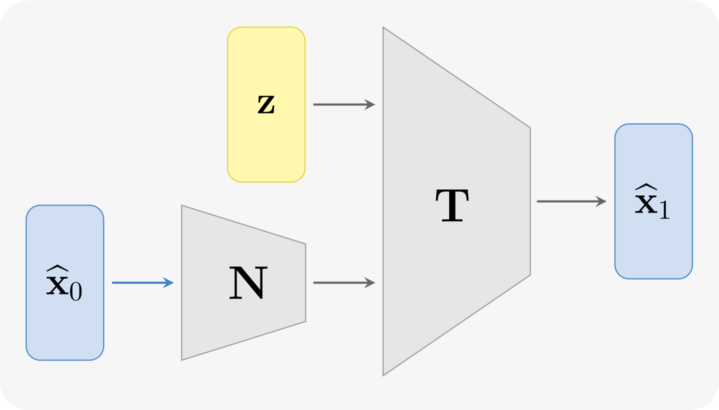

FIGURE 2. The DNN model structure for the proposed normalizing flow sFML method (14).

4. **Numerical examples.**

In this section, we present several numerical tests to demonstrate the performance of our proposed method. The examples include a skew product SDE, an exponential mean OU process, followed by 3 3-dimensional nonlinear SDEs. In all the examples, the true SDE systems are known. However, the known SDEs are used only to generate the training data set (11). We solve the true systems by Euler-Maruyama method with a time step 10^{-4}. The "initial conditions" $\mathbf{x}_0$ in (11) are sampled uniformly in a given domain, specified in each example.

In our sFML model, Figure 2, the DNN has 3 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. In our examples, the DNN training is usually conducted for $200,000 \sim 300,000$ epochs and batch size is taken $30,000 \sim 40,000$.

4.1. **A skew product SDE.**

We first consider the following 2-dimensional stochastic dynamics mentioned in Section 10 of [33],

$$\begin{cases} \dfrac{dx}{dt} = \left(1 - y^2\right) x, \\ \dfrac{dy}{dt} = -\dfrac{\alpha}{\varepsilon} y + \sqrt{\dfrac{2\lambda}{\varepsilon}} \dfrac{dW_t}{dt}, \end{cases} \tag{18}$$

where the parameters are set as $\alpha = 1.0$, $\lambda = 2.4$, and $\varepsilon = 0.005$. The fast variable y is an Ornstein-Uhlenbeck (OU) process, which has a Gaussian stationary distribution with mean 0 and variance λ/α. Notably, this distribution does not depend on the slow variable x. The training data set (11) is generated by uniformly sampling (x_0, y_0) in $(-1.5, 2) \times (-1, 1.6)$ and solved for up to $T = 1.0$. A total of $M = 40,000$ trajectory pairs are used in the training data set (11), where the time step Δ is taken 0.01. A simulation of the full system is illustrated on the left of Figure 4.1.

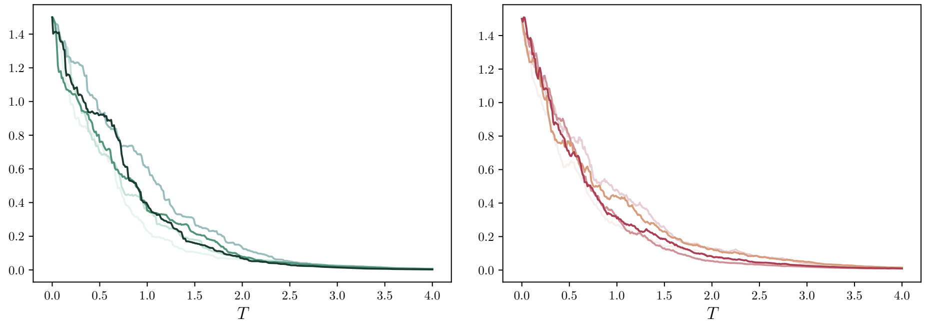

FIGURE 3. Sample trajectories of Example 4.1 slow variable with initial condition $x_0 = 1.5$ and y_0 sampling from stationary distribution. Left: ground truth; Right: Simulation using the trained sFML model.

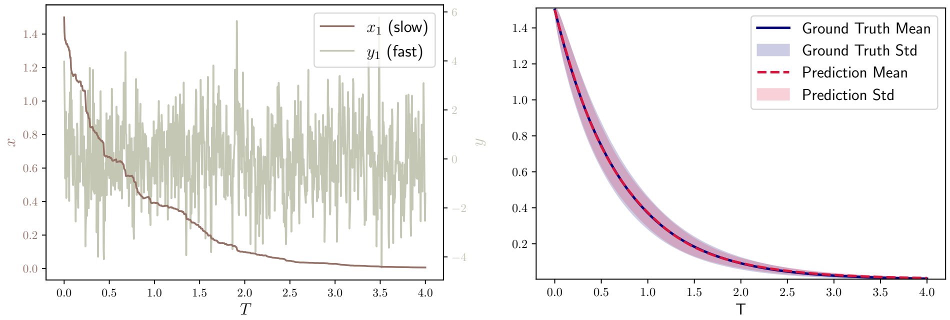

FIGURE 4. Left: One simulated sample of full system of Example 4.1 with initial condition $x_0 = 1.5$ and y_0 sampling from stationary distribution. Right: Comparison for mean and standard deviation (STD) of ground truth and learned sFL model for the slow variable.

Once the sFML model (7) is trained, we conduct system predictions up to $T = 4.0$, which requires 400 time steps. In Figure 4.1, we compare some sample trajectory paths produced by the ground truth (left) and the learned sFML model (right), with an initial condition $x_0 = 1.5$. (For the ground truth, y_0 is sampled from the stationary distribution. For the sFML model, y_0 is not required.) We observe the two sets appear visually similar to each other. To further validate the accuracy of sFML method, we compare the mean and standard deviation of the solution averaged over $10,000$ trajectories. The sFML model predictions are shown in the right of Figure 4.1, along with the reference ground truth. Good agreement is observed.

4.2. **An exponential mean OU process.** We consider the 2-dimensional stochastic dynamics

$$\begin{cases} \dfrac{dx}{dt} = 1 - x + y, \\ \dfrac{dy}{dt} = \dfrac{1}{\varepsilon}(e^{-x} - y) + \sqrt{\dfrac{2}{\varepsilon}}\dfrac{dW_t}{dt}. \end{cases} \tag{19}$$

Conditioned on the slow variable x, the fast variable y is an OU process and eventually converges to a Gaussian stationary distribution with a mean e^{-x} and unit variance.

The training data set is generated with (x_0, y_0) uniformly sampled from $(-1.5, 2) \times (-4, 4)$. We set $\varepsilon = 0.001$ and simulate the full system with termination time $T = 1.0$ with $\Delta = 0.01$. An illustration of the full system can be viewed on the left of Figure 4.2. A total of $M = 40,000$ trajectory pairs are used in the training data set (11).

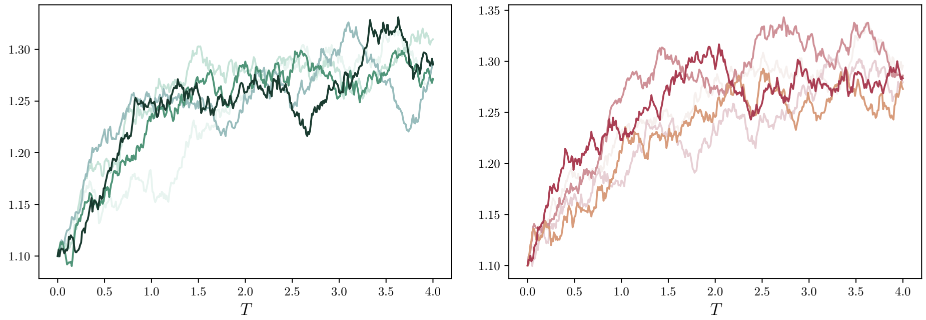

FIGURE 5. Sample trajectories of Example 4.2 slow variable with initial condition $x_0 = 1.5$ and y_0 sampling from stationary distribution. Left: ground truth; Right: Simulation using the trained sFML model.

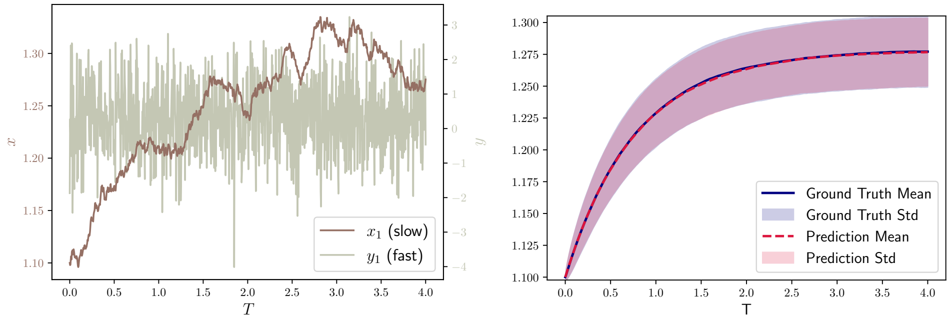

FIGURE 6. Left: One simulated sample of full system of Example 4.2 with initial condition $x_0 = 1.5$ and y_0 sampling from stationary distribution (conditioned on x_0). Right: Comparison for mean and standard deviation (STD) of ground truth and learned sFL model for the slow variable.

With the well-trained sFML model, we compare some sample trajectory paths produced by the ground truth (left) and the learned sFML model (right) in Figure 4.2. The two sets of samples are visually similar. The mean and STD of the solution are shown in Figure 4.2. Good agreement is observed for sFML model and reference ground truth.

4.3. **A triad system.** We consider the following 3-dimensional stochastic dynamics

$$\begin{cases} \dfrac{dx}{dt} = -\dfrac{2}{\varepsilon} y_1 y_2, \\ \dfrac{dy_1}{dt} = -\dfrac{1}{\varepsilon^2} y_1 + \dfrac{1}{\varepsilon} x y_2 + \dfrac{\sigma}{\varepsilon} \dfrac{dW_t^1}{dt}, \\ \dfrac{dy_2}{dt} = -\dfrac{2}{\varepsilon^2} y_2 + \dfrac{1}{\varepsilon} x y_1 + \dfrac{\sigma}{\varepsilon} \dfrac{dW_t^2}{dt}, \end{cases} \tag{20}$$

where the parameters are set as $\sigma = 1$ and $\varepsilon^2 = 0.001$. This system is studied as a stochastic model for mean flow-wave interaction in barotropic-baroclinic turbulence [14, 26]. We are interested in the climate (slow) variable x, instead of the weather (fast) variables $\mathbf{y}$. We plot one trajectory sample of the full system on the left of Figure 4.3. According to the work of [19], the analytically dervied effective equation for x is $d\overline{x}_t = -\frac{1}{2}\overline{x}_t + \frac{1}{\sqrt{3}} dW_t$, which indicated that the limiting process of x is a OU process.

Here we utilize the sFML approach to construct a data-driven effective model for $\mathbf{x}$. We generate the training data set up to $T = 1$ with initial condition (x_0, y_0) uniformly sampling from $(-2, 3) \times (-1, 1)^2$ and time step with $\Delta = 0.01$. A total $60,000$ trajectory pair data are utilized for training the sFML model.

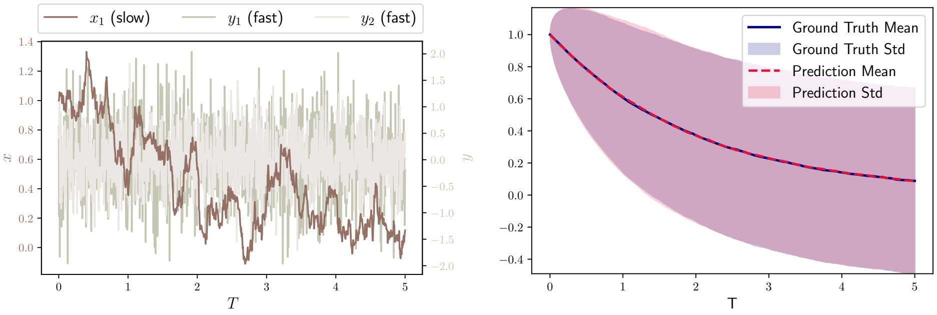

FIGURE 7. Left: One simulated sample of full system of Example 4.4 with initial condition $x_0 = 1$ and $\mathbf{y}$ sampling from stationary distribution (conditioned on x_0). Right: Mean and standard deviation (STD) for slow variable.

Once successfully trained, we generate trajectory samples up to $T = 5$ with initial condition $x_0 = 1$ using the trained model and compare them with ground truth. It takes 500 steps to generate data from sFML model. Visual comparison of the sample paths for $\mathbf{x}$ is presented in Figure 4.3, where we observe qualitatively similar behavior. To further validate the accuracy, we present the comparisons of mean and STD on the right of Figure 4.3. We observe good agreement between the sFML model prediction and the ground truth.

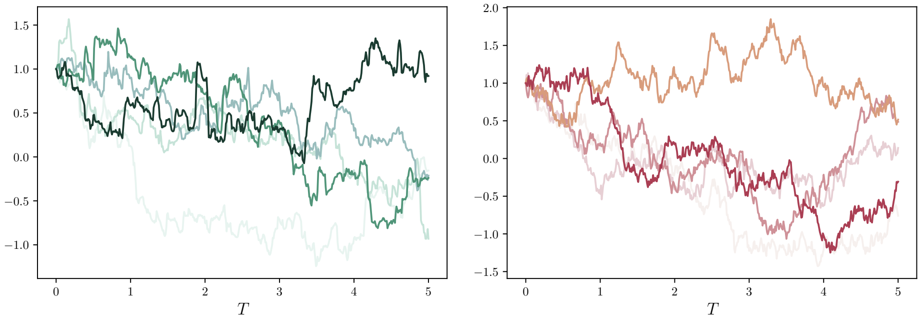

FIGURE 8. Sample trajectories of Example 4.3 slow variable with initial condition $\mathbf{x}_0 = 1$ and $\mathbf{y}_0$ sampling from stationary distribution (conditioned on x_0). Left: ground truth; Right: Simulation of x using the trained sFML model.

4.4. **A 3D nonlinear SDE.**

We consider the following 3-dimensional stochastic dynamics inspired by [21],

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 + \sigma_1 \dfrac{dW_t^1}{dt}, \\ \dfrac{dx_2}{dt} = -x_1 - x_2 + y^2 + \sigma_2 \dfrac{dW_t^2}{dt}, \\ \dfrac{dy}{dt} = -\dfrac{1}{\varepsilon}\left(y - \dfrac{1}{4}x_1\right) + \sigma_3 \sqrt{\dfrac{2}{\varepsilon}} \dfrac{dW_t^3}{dt}, \end{cases} \tag{21}$$

where the parameters are set as $\sigma_1 = 0.3$, $\sigma_2 = 0.3$, $\sigma_3 = 0.1$, and $\varepsilon = 0.001$. The one-dimensional fast variable y is driven by an OU process and evolves to a Gaussian distribution with a mean $\frac{1}{4}x_1$ and variance σ_3^2. We plot one trajectory sample of the full system in Figure 4.4 for a visual illustration.

To train the sFML effective model, we generate the training data set up to $T = 1$ with initial condition $(\mathbf{x}_0, y_0)$ uniformly sampling from $(-1.5, 2.5) \times (-2, 1.5) \times (-0.6, 1)$ and a time step $\Delta = 0.01$. A total of $120{,}000$ trajectory pair data are utilized for training the sFML model. Once successfully trained, we generate trajectory samples up to $T = 8$ with initial condition $\mathbf{x}_0 = (1.5, 1.0)$ using the trained sFML model and compare them with ground truth. Visual comparison of the sample paths of the slow variables are shown 4.4. To further validate the accuracy, we present the comparison of mean and STD in Figure 4.4. And in Figure 4.4, we also show the comparison of the probability distribution of the solution at $T = 2, 4, 6, 8$. We observe good agreement between the sFML model prediction and the ground truth.

4.5. **Multiscale stochastic oscillator.**

We consider the following stochastic oscillator,

$$\begin{cases} \dfrac{dx_1}{dt} = \lambda x_1 - \theta x_2 - \gamma x_1 y + \sigma \dfrac{dW_t^1}{dt}, \\ \dfrac{dx_2}{dt} = \theta x_1 + \lambda x_2 - \gamma x_2 y + \sigma \dfrac{dW_t^2}{dt}, \\ \dfrac{dy}{dt} = -\dfrac{1}{\varepsilon}\left(y - x_1^2 - x_2^2\right) + \sigma \sqrt{\dfrac{2}{\varepsilon}} \dfrac{dW_t^3}{dt}, \end{cases} \tag{22}$$

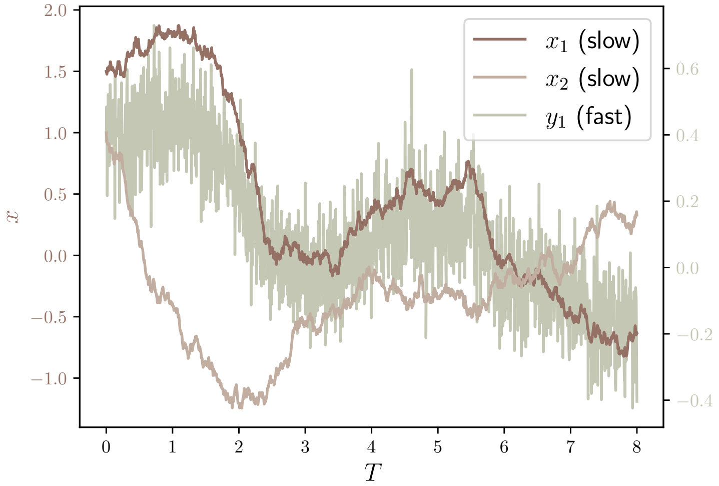

FIGURE 9. One simulated sample of full system of Example 4.4 with initial condition $\mathbf{x}_0 = (1.5, 1.0)$ and y_0 sampling from stationary distribution.

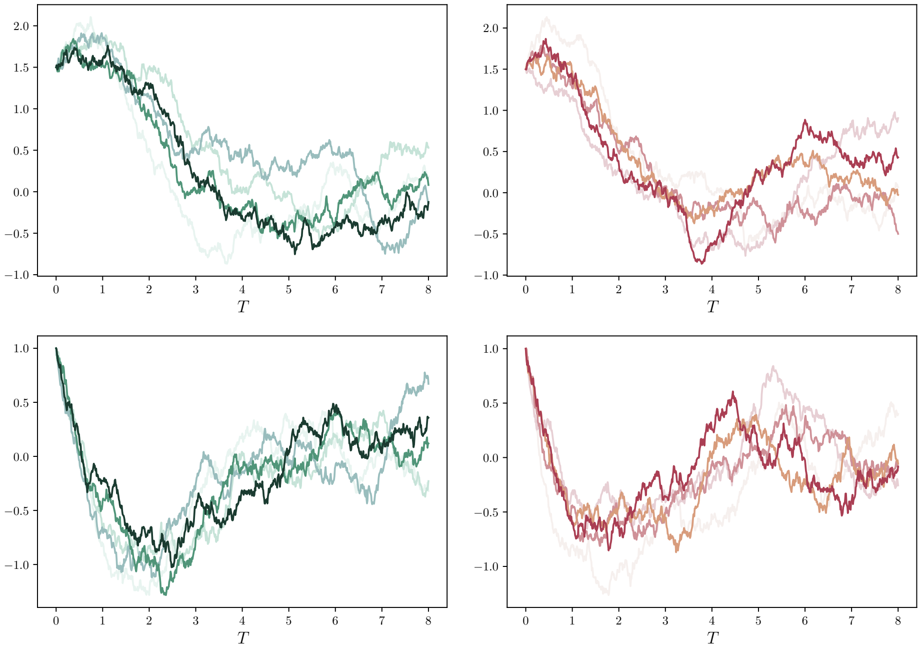

FIGURE 10. Sample trajectories of Example 4.4 slow variable with an initial condition $\mathbf{x}_0 = (1.5, 1.0)$. Left column: ground truth of x_1 (top) and x_2 (bottom), with y_0 sampled from its stationary distribution (conditioned on $\mathbf{x}_0$); Right column: sFML simulation of x_1 (top) and x_2 (bottom).

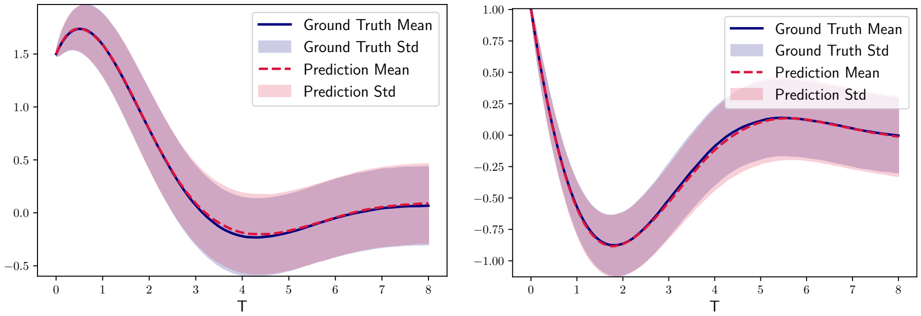

FIGURE 11. Example 4.4: Mean and standard deviation (STD) for the slow variables x_1 (left) and x_2 (right), with an initial condition $\mathbf{x}_0 = (1.5, 1.0)$.

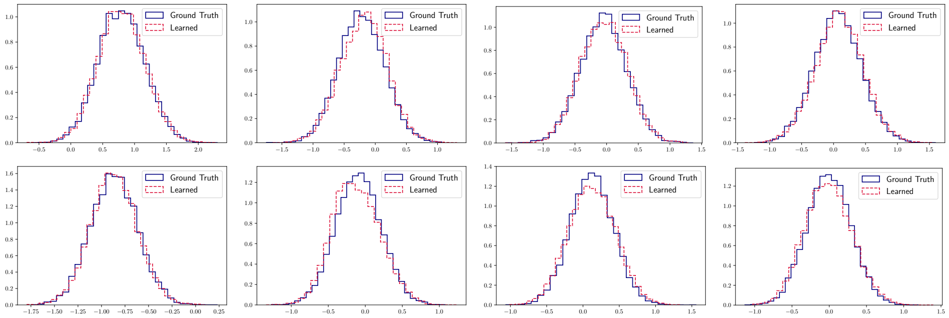

FIGURE 12. Example 4.4: Comparison of the distribution of the slow variables at $T = 2, 4, 6, 8$ with an initial condition $\mathbf{x}_0 = (1.5, 1.0)$. Top row: x_1; Bottom row: x_2.

where the parameters are set as $\lambda = 1.0$, $\theta = 1.0$, $\gamma = 1.0$, $\sigma = 0.1$, and $\varepsilon = 0.001$. This system is studied in [4], whose deterministic version was studied in [27] as a low-dimensional reduced model for a flow past a circular cylinder. The one-dimensional fast process y is an OU process and evolves to a Gaussian distribution with mean $(x_1^2 + x_2^2)$ and variance σ^2. A trajectory sample path of the full system is shown in Figure 4.5 for demonstration purposes.

To train the sFML effective model, we generate the training data set up to $T = 1$ with initial condition $(\mathbf{x}_0, y_0)$ uniformly sampling from $(-1.5, 1.5)^2 \times (0.1, 2.5)$ and time step $\Delta = 0.01$. A total of $120,000$ trajectory data pairs are utilized for training the sFML model. Once the sFML model is obtained, we generate system prediction for up to $T = 20$ with initial condition $\mathbf{x}_0 = (1.0, 1.0)$ and compare them with ground truth. Samples of the phase portraits of the slow variables are shown in Figure 4.5, where we observe good visual comparison. To further validate the accuracy, we present the comparison of mean and STD in Figure 4.5, and the probability distributions at time $T = 4, 8, 12, 16$ in Figure 4.5. We observe good agreement between the sFML model prediction and the ground truth.

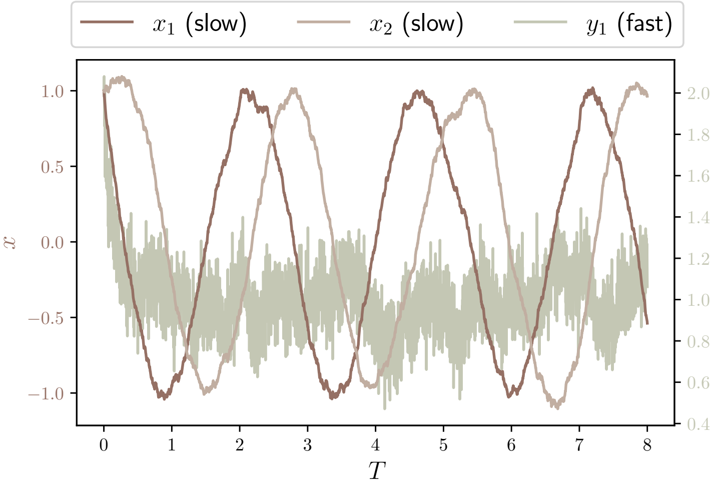

FIGURE 13. One simulated sample of full system of Example 4.5 with initial condition $\mathbf{x}_0 = (1.0, 1.0)$ and y_0 sampling from stationary distribution.

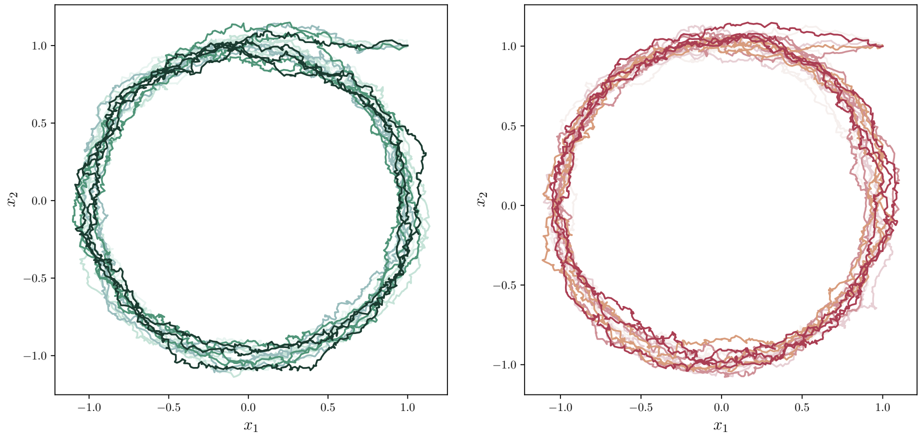

FIGURE 14. Example 4.5: Samples of phase portrait of the slow variables (x_1, x_2) with initial condition $\mathbf{x}_0 = (1.0, 1.0)$. Left: ground truth (with y_0 sampled from its stationary distribution); Right: sFML model.

5. **Conclusions.** In this paper, we presented a numerical method for constructing accurate effective numerical model of slow-fast multiscale stochastic dynamical systems. The method utilizes short bursts of data of the slow variables and constructs the stochastic flow map of the slow dynamics. A conditional normalizing flow model is employed to ensure the learned model is an accurate generative model in distribution. Once successfully trained, the model can serve as a time stepper to produce predictions of the slow variables, with any given initial conditions. By using a comprehensive set of numerical examples, we demonstrated that the proposed approach

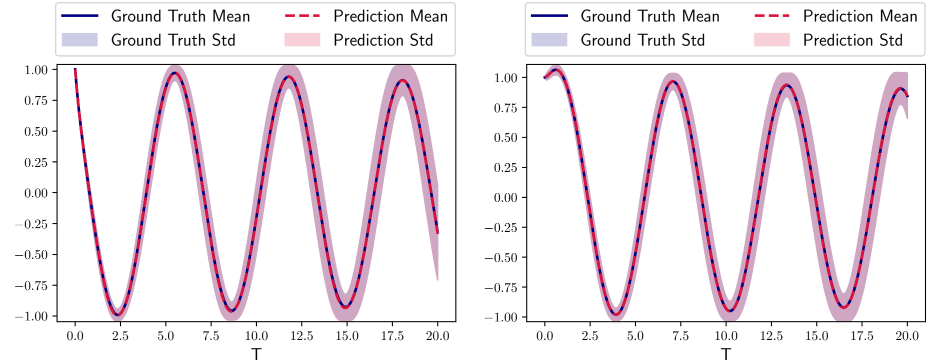

FIGURE 15. Example 4.5: Comparison of the mean and standard deviation (STD) for slow variables with initial condition $\mathbf{x}_0 = (1.0, 1.0)$. Left: x_1; Right: x_2

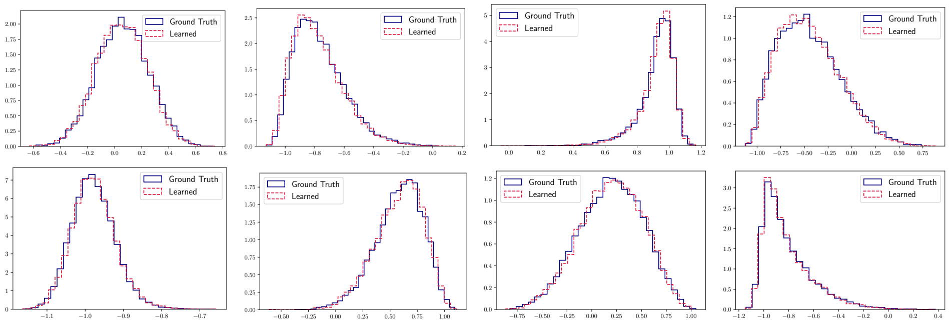

FIGURE 16. Example 4.5: Comparasion of the distribution of the slow variables at $T = 4, 8, 12, 16$ with initial condition $\mathbf{x}_0 = (1.0, 1.0)$. Top row: x_1; Bottom row: x_2.

is effective and accurate in modeling a variety of unknown multiscale stochastic systems.

REFERENCES

[1] C. Archambeau, D. Cornford, M. Opper and J. Shawe-Taylor, Gaussian process approximations of stochastic differential equations. In Neil D. Lawrence, Anton Schwaighofer, and Joaquin Quiñonero Candela, editors, *Gaussian Processes in Practice*, volume 1 of *Proceedings of Machine Learning Research*, Bletchley Park, UK, **1** (2007).

[2] S. L. Brunton, J. L. Proctor and J. Nathan Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proc. Natl. Acad. Sci. USA*, **113** (2016), 3932-3937.

[3] K. P. Champion, S. L. Brunton and J. N. Kutz, Discovery of nonlinear multiscale systems: Sampling strategies and embeddings, *SIAM J. Appl. Dyn. Syst.*, **18** (2019), 312-333.

[4] M. D. Chekroun, A. Tantet, H. A Dijkstra and J. D. Neelin, Ruelle-pollicott resonances of stochastic systems in reduced state space. part i: Theory, *Journal of Statistical Physics*, **179** (2020), 1366-1402.

[5] R. T. Q. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud, Neural ordinary differential equations, In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **31** (2018).

[6] 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.

[7] 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), 811-830.

[8] Y. Chen and D. Xiu, Learning stochastic dynamical system via flow map operator, *J. Comput. Phys.*, **508** (2024), 112984, 23 pp.

[9] Y. Chen and D. Xiu, Modeling unknown stochastic dynamical system subject to external excitation, preprint, arXiv:2406.15747.

[10] Z. Chen, V. Churchill, K. Wu and D. Xiu, Deep neural network modeling of unknown partial differential equations in nodal space, *Journal of Computational Physics*, **449** (2022), 110782, 20 pp.

[11] V. Churchill and D. Xiu, Flow map learning for unknown dynamical systems: Overview, implementation and benchmarks, *Journal of Machine Learning for Modeling and Computing*, **4** (2023), 173-201.

[12] R. R. Coifman, I. G. Kevrekidis, S. Lafon, M. Maggioni and B. Nadler, Diffusion maps, reduction coordinates and low dimensional representation of stochastic systems, *Multiscale Model. Simul.*, **7** (2008), 842-864.

[13] 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, 18 pp.

[14] T. DelSole and B. F. Farrell, The quasi-linear equilibration of a thermally maintained, stochastically excited jet in a quasigeostrophic model, *Journal of Atmospheric Sciences*, **53** (1996), 1781-1797.

[15] 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: An Interdisciplinary Journal of Nonlinear Science*, **33** (2023), 19 pp.

[16] L. Dinh, J. Sohl-Dickstein and S. Bengio, Density estimation using real NVP, In *International Conference on Learning Representations*, 2017.

[17] C. J. Dsilva, R. Talmon, C. W. Gear, R. R. Coifman and I. G. Kevrekidis, Data-driven reduction for a class of multiscale fast-slow stochastic dynamical systems, *SIAM J. Appl. Dyn. Syst.*, **15** (2016), 1327-1351.

[18] J. Duan, K. Lu and B. Schmalfuss, Invariant manifolds for stochastic partial differential equations, *Ann. Probab.*, **31** (2003), 2109-2135.

[19] W. E, D. Liu and E. Vanden-Eijnden, Analysis of multiscale methods for stochastic differential equations, *Comm. Pure Appl. Math.*, **58** (2005), 1544-1585.

[20] W. E and E. Vanden-Eijnden, Metastability, conformation dynamics and transition pathways in complex systems, In *Multiscale modelling and simulation*, *Lect. Notes Comput. Sci. Eng.*, Springer, Berlin, **39** (2004), 35-68.

[21] L. Feng, T. Gao, M. Dai and J. Duan, Learning effective dynamics from data-driven stochastic systems, *Chaos*, **33** (2023), 043131, 15 pp.

[22] X. Fu, L.-B. Chang and D. Xiu, Learning reduced systems via deep neural networks with memory, *J. Machine Learning Model. Comput.*, **1** (2020), 97-118.

[23] R. Khasminskij, On the principle of averaging the itov's stochastic differential equations, *Kybernetika*, **4** (1968), 260-279.

[24] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. B. A. Stuart and A. Anandkumar, Fourier neural operator for parametric partial differential equations, In *International Conference on Learning Representations*, 2021.

[25] C. Ma, J. Wang and W. E, Model reduction with memory and the machine learning of dynamical systems, *Commun. Comput. Phys.*, **25** (2019), 947-962.

[26] A. J. Majda, I. Timofeyev and E. Vanden Eijnden, Models for stochastic climate prediction, *Proceedings of the National Academy of Sciences*, **96** (1999), 14687-14691.

[27] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *Journal of Fluid Mechanics*, **497** (2003), 335-363.

[28] B. Øksendal, *Stochastic Differential Equations*, 6^{th} edition, Springer-Verlag, Berlin, 2003.

[29] M. Opper, Variational inference for stochastic differential equations, *Ann. Phys.*, **531** (2019), 1800233, 9 pp.

[30] H. Owhadi, Computational graph completion, *Research in the Mathematical Sciences*, **9** (2022), 27.

[31] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed and B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, *The Journal of Machine Learning Research*, **22** (2021), 2617-2680.

[32] G. Papamakarios, T. Pavlakou and I. Murray, Masked autoregressive flow for density estimation, In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett, editors, *Advances in Neural Information Processing Systems*, Curran Associates, Inc., **30** (2017).

[33] G. A. Pavliotis and A. Stuart, *Multiscale Methods: Averaging and Homogenization*, Texts Appl. Math., 53 Springer, New York, 2008.

[34] T. Qin, Z. Chen, J. D. Jakeman and D. Xiu, Data-driven learning of nonautonomous systems, *SIAM J. Sci. Comput.*, **43** (2021), 1607-1624.

[35] 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), 63-82.

[36] T. Qin, K. Wu and D. Xiu, Data driven governing equations approximation using deep neural networks, *J. Comput. Phys.*, **395** (2019), 620-635.

[37] M. Raissi, P. Perdikaris and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, **378** (2019), 686-707.

[38] M. Raissi, P. Perdikaris and G. E. Karniadakis, Multistep neural networks for data-driven discovery of nonlinear dynamical systems, preprint, `arXiv:1801.01236`, 2018.

[39] B. Schmalfuss and K. R. Schneider, Invariant manifolds for random dynamical systems with slow and fast variables, *J. Dynam. Differential Equations*, **20** (2008), 133-164.

[40] K. Wu and D. Xiu, Data-driven deep learning of partial differential equations in modal space, *Journal of Computational Physics*, **408** (2020), 109307, 22 pp.

[41] Z. Xu, Y. Chen, Q. Chen and D. Xiu, Modeling unknown stochastic dynamical system via autoencoder, preprint, `arXiv:2312.10001`, 2023.

[42] 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 Journal on Scientific Computing*, **44** (2022), 80-99.

[43] F. X.-F. Ye, S. Yang and M. Maggioni, Nonlinear model reduction for slow-fast stochastic systems near unknown invariant manifolds, *J. Nonlinear Sci.*, **34** (2024), 22, 54 pp.

[44] C. Yildiz, M. Heinonen, J. Intosalmi, H. Mannerstrom and H. Lahdesmaki, Learning stochastic differential equations with gaussian processes without gradient matching, In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, (2018), 1-6.

[45] 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, `arXiv:2204.03193`, 2022.

[46] P. Zieliński and J. S. Hesthaven, Discovery of slow variables in a class of multiscale stochastic systems via neural networks, *J. Nonlinear Sci.*, **32** (2022), 51, 34 pp.

Received August 2024; revised November 2024; early access November 2024.