

# Detecting Stochastic Governing Laws with Observation on Stationary Distributions \*

Xiaoli Chen<sup>a,b,†</sup>, Hui Wang<sup>c,‡</sup>, Jinqiao Duan<sup>d,§</sup>

February 17, 2023

## Abstract

Mathematical models for complex systems are often accompanied with uncertainties. The goal of this paper is to extract a stochastic differential equation governing model with observation on stationary probability distributions. We develop a neural network method to learn the drift and diffusion terms of the stochastic differential equation. We introduce a new loss function containing the Hellinger distance between the observation data and the learned stationary probability density function. We discover that the learnt stochastic differential equation provides a fair approximation of the data-driven dynamical system after minimizing this loss function during the training method. The effectiveness of our method is demonstrated in numerical experiments.

Key words: Stochastic dynamical systems; Fokker-Planck equations; Machine learning; Neural network.

## 1 Introduction

Mathematical models for scientific and engineering systems often involve uncertainties and thus are often in the form of stochastic differential equations (SDE). These stochastic dynamical systems are ubiquitous in biology, physics, geosciences and other fields. Stochastic dynamical systems provide an appropriate framework to investigate random phenomena [1–5]. Hence, the determination of SDE models is crucial for quantifying and predicting dynamical

---

\*This work is supported by the National Natural Science Foundation of China (NSFC) (Grant No.11901536). Xiaoli Chen is supported by the Ministry of Education, Singapore, under its Research Centre of Excellence award to the Institute for Functional Intelligent Materials (I-FIM, project No. EDUNC-33-18-279-V12)

†xlchen@nus.edu.sg

‡Corresponding author: huiwang2018@zzu.edu.cn

§duan@iit.edu

behaviors of the nonlinear system under random fluctuations. An SDE is characterized by the drift term and diffusion terms. In this paper, we aim to detect appropriate drift and diffusion terms with stationary probability distribution data.

The stationary probability distribution of a stochastic dynamical system does not change with time and it is the stationary solution of the corresponding Fokker-Planck equation [6–12]. The stationary probability distribution carries the information of the underlying stochastic system [13, 14]. The Hellinger distance is the distance between probability distributions which characterizes how close two different distributions are. In this paper, we use Hellinger distance to identify whether the constructed SDE is an appropriate approximation of a data-driven stochastic dynamical system.

Neural networks can be represented as compositions of simple functions with parameters, and such functional representations can be used for parameter estimation of time-series data and kernel estimation [15]. There has been some progress in learning stochastic differential equation models from noisy data. A variation estimation method was used to learn the drift term with the observation trajectory data [16–19]. There was also an RNN-based variational method [20], a sparse learning method [21], and a Kramers-Moyal formulae [22] for learning stochastic dynamical systems. A stochastic adjoint sensitivity method was proposed to learn stochastic differential equations [23] or stochastic differential equations with jumps [24]. In [25], they used small samples from just a few snapshots of unpaired data to infer the drift and diffusion terms of stochastic differential equations. Moreover, in [26, 27], they learned Lévy noise parameters by deep neural networks. In [28], they solved the steady-state Fokker-Planck equation with a small amount of data through combining the deep KD-tree.

We have recently developed a data-driven approach [29, 30] to discover stochastic differential equations with non-Gaussian Lévy noise using the nonlocal Kramers-Moyal formulas, and further learned the stochastic differential equations from discrete particle samples at different time snapshots using the Fokker-Planck equation and physics-informed neural networks [31].

However, in addition to sample path observation data, there are recent advances in observing or measuring stationary probability distributions [7, 9, 11, 32]. To take advantage of these new types of data, we devise a neural network method to extract stochastic dynamical system models with stationary probability distribution or a long time trajectory as observation data. This motivates our research reported in this paper. Specifically, we develop a neural network method to extract stochastic governing laws based on probability measures. Given observation data, we learn the drift and diffusion terms which are approximated by two neural networks. Since if we learn the drift and diffusion together, the results would not be unique. So in this work we proposed two approaches. The first approach entails simply learning the drift or diffusion terms. The second technique involves learning the drift and diffusion terms simultaneously with one drift term observational data. We compare our learned results in three-dimensional settings with Hellinger distance substituted by Jensen-Shannon divergence and mean-square distance which demonstrate the efficacy of our proposed approaches.

This paper is organized as follows. In Section 2, we present two methods for learning stochastic governing laws based on physics informed neural networks and Hellinger distance of probability distributions. In Section 3, we present examples to learn the drift terms and the diffusion terms. Finally, we end with some discussions in Section 4.

## 2 Methodology

### 2.1 Problem setup

Consider the following stochastic differential equation (SDE)

$$dX_t = b(X_t)dt + \sigma(X_t)dB_t, \quad X_0 = x_0, \quad (1)$$

where the  $n$ -dimensional vector function  $b(\cdot)$  is the drift term, the  $n \times n$  matrix function  $\sigma(\cdot)$  is the diffusion term, and  $B_t$  is an  $n$ -dimensional Brownian motion.

The generator of the SDE (1) is [33]:

$$Au = \sum_{i=1}^n b_i \frac{\partial u}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^n (\sigma\sigma^T)_{i,j} \frac{\partial^2 u}{\partial x_i \partial x_j}.$$

The probability density function (PDF) is a quantity that carries information of the stochastic system. The time evolving probability density function of the solution process  $X_t$  is governed by the Fokker-Planck equation, which is written as follows:

$$\begin{aligned} \partial_t p(x, t) &= A^* p(x, t), \quad x \in \mathbb{R}^n, t > 0, \\ p(x, 0) &= p_0(x), \end{aligned} \quad (2)$$

where  $p_0(x)$  is the initial probability density function,  $A^*$  is the adjoint operator of the generator  $A$  and has the following form:

$$A^* p = - \sum_{i=1}^n \frac{\partial}{\partial x_i} (b_i p) + \frac{1}{2} \sum_{i,j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} ((\sigma\sigma^T)_{i,j} p). \quad (3)$$

Note that the Fokker-Planck equation is a deterministic linear partial differential equation with an initial condition.

The stationary Fokker-Planck equation is:

$$A^* p(x) = 0, \quad (4)$$

with a condition  $\int_{\mathbb{R}^n} p(x) dx = 1$ . We assume that there exists a unique stationary probability density function (still denote it by  $p(x)$ ) in this paper.

We consider the scenario when available data is time series observation data or probability density function. Our objective is to infer the drift and diffusion terms. Because of the stochasticity of the dynamical system, we could not use the mean square error to get the loss function of the SDE (1). The main issue is how to quantify the stochastic dynamics with use the deterministic indexes. For example, we can use the maximal likelihood estimation [34] or the most transition pathway [35] to extract or learn the SDE model. Here we will use the stationary probability density function as the deterministic index to learn the SDE. If the available data is long time trajectory data of  $X(t)$ , we may first use kernel density estimation to learn the probability density function. We will propose a machine learning method to learn the drift and the diffusion terms of the SDE, with different measures for the distance of the observed probability density function.

## 2.2 Machine Learning

As the drift  $b$  and diffusion  $\sigma$  characterize the uncertainty of the SDE, we will estimate them based on observations of probability distributions (i.e., probability measures) of the system paths  $X_t$ . Now we introduce the Hellinger distance [36, 37] between two probability distributions. It is used to quantify the distance between two probability distributions in the space of probability measures. For our purpose here, the Hellinger distance  $H(p, q)$  between two probability density functions  $p(x)$  and  $q(x)$  is defined as follows,

$$H(p, q) \triangleq \sqrt{\frac{1}{2} \int (\sqrt{p(x)} - \sqrt{q(x)})^2 dx}, \quad (5)$$

which satisfies the property:  $0 \leq H(p, q) \leq 1$ .

With the observed stationary probability density  $q(x)$ , we determine or estimate the drift term  $b(x)$  by minimizing the Hellinger distance between the true stationary probability density  $p(x)$  for the solution process  $X(t)$  and the observation probability density  $q(x)$ .

Note that Hellinger distance is a measure to describe the distance of two probability density functions. Other distance also can describe the distance. Such as, given the probability density function  $p(x)$  and  $q(x)$ , respectively, the Kullback-Leibler (KL) divergence is defined as

$$H_{KL}(p||q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx. \quad (6)$$

While the Kullback-Leibler divergence is asymmetry, there also exists a symmetric measure between two probability density, which is Jensen-Shannon divergence, introduced as follows:

$$H_{JS}(p||q) = \frac{1}{2} H_{KL}(p||\frac{q+p}{2}) + \frac{1}{2} H_{KL}(\frac{q+p}{2}||p). \quad (7)$$

Later we will also use the Jensen-Shannon divergence to measure the distance.

Given the noise intensity  $\sigma(x)$  and observation of the stationary probability density  $q(x)$ , we will learn the drift term  $b$ . We devise two neural networks to approximate the drift term and stationary probability density  $p(x)$ , where the input is the space domain  $x$  and the output is the  $b_{NN}(x)$  and  $p_{NN}(x)$ .

On the one hand, the output of  $p_{NN}(x)$  should satisfy the functional (21). We define the loss function as:

$$Loss_H = \frac{1}{2N_H} \sum_{i=1}^{N_H} (\sqrt{p_{NN}(x_i)} - \sqrt{q(x_i)})^2, \quad (8)$$

where  $\{x_i\}_{i=1}^{N_H}$  are the points in the spatial domain to compute the integral and  $N_H$  is the number of the observation data.

On the other hand, the neural networks of the drift term  $b_{NN}(x)$  and the stationary probability density  $p_{NN}(x)$  should satisfy the steady Fokker-Planck equation (4).

Similar to the physics informed neural network [39, 40], we define the residual neural network as

$$f(x) = - \sum_{i=1}^n \frac{\partial}{\partial x_i} (b_{iNN} p_{NN}) + \frac{1}{2} \sum_{i,j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} ((\sigma \sigma^T)_{i,j} p_{NN}). \quad (9)$$

Then the loss function of the residual neural network is defined as:

$$Loss_f = \frac{1}{N_f} \sum_{i=1}^{N_f} (f(x_i))^2, \quad (10)$$

where  $\{x_i\}_{i=1}^{N_f}$  is the residual points in the spatial domain and  $N_f$  is the number of the residual points. Here we randomly choose the residual points at each iteration step. The sketch of the method is shown in Figure 1.

The total loss function is

$$Loss = Loss_H + Loss_f. \quad (11)$$

For the unknown drift term and diffusion term, because  $b(x) = 0$  and  $\sigma(x) = 0$  is also the minimization solution of loss function, thus we could not learn the terms uniquely if we train the loss function (11). The observation data [40] of drift term at some points need to know. To avoid the zeros solution, the observation data of drift term at few points is given, i.e.  $\{x_i, b(x_i)\}_{i=1}^{N_b}$ . The loss function of the drift term is:

$$Loss_b = \frac{1}{N_b} \sum_{i=1}^{N_b} (b_{NN}(x_i) - b(x_i))^2, \quad (12)$$

so the loss function is defined as:

$$Loss_2 = Loss_H + Loss_f + Loss_b. \quad (13)$$

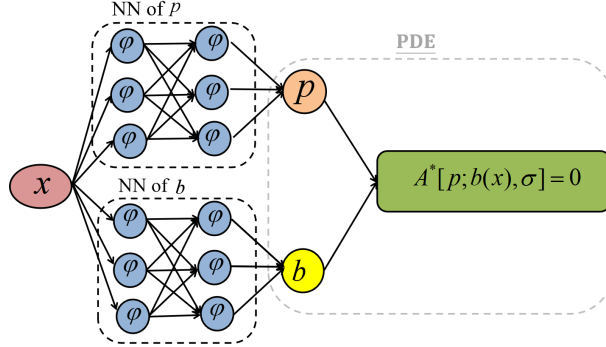


Figure 1: **Schematic of the neural network for solving PDEs:** two neural networks for approximate the probability  $p$  and the drift term  $b$ , where the input is  $x$  and  $\varphi$  is the activation function.

If we use the Jensen-Shannon divergence to measure the distance, we just replace the  $loss_H$  to

$$Loss_{JS} = H_{JS}(p_{NN}||q). \quad (14)$$

We also compare our method (11) with the traditional physics informed neural network (PINN) method [39] for solving the inverse problem of the Fokker-Planck equation. With the observation data of the stationary probability density function  $q(x)$ , the loss function is written as mean square error:

$$Loss_{ob} = \frac{1}{N_{ob}} \sum_{i=1}^{N_{ob}} (p_{NN}(x_i) - q(x_i))^2. \quad (15)$$

The total loss of PINN method is defined as:

$$Loss_{PINN} = Loss_{ob} + Loss_f. \quad (16)$$

If the observed data is the trajectory of stochastic differential equation, i.e.,  $X = (X_{t_0}, X_{t_1}, \dots, X_{t_N})$ , we can use the kernel density estimation to obtain the probability density function, denoting as  $q_{KD}$ . Similarly, we replace the probability density function loss  $loss_H$  in Eq. 8 or  $loss_{JS}$  in Eq. 14 with another loss from the estimated density, as  $q_{KD}$ .

Remark: For an SDE with non-Gaussian Lévy case

$$dX_t = b(X_t)dt + \varepsilon dL_t^\alpha, \quad X_t \in \mathbb{R}^n, \quad (17)$$

where  $b(\cdot)$  is the vector drift term, and  $L_t^\alpha$  is a symmetric  $\alpha$ -stable Lévy process in  $\mathbb{R}^n$ . The generating triplet of the Lévy process is  $(0, 0, \nu_\alpha)$ .

The corresponding nonlocal Fokker-Planck operator is [33]

$$A^*p = - \sum_{i=1}^n \frac{\partial}{\partial x_i} (a_i p) + \varepsilon^\alpha \int_{\mathbb{R}^n \setminus \{0\}} [p(x+y) - p(x)] \nu_\alpha(dy), \quad (18)$$

where  $\nu_\alpha(dy)$  is the  $\alpha$ -stable Lévy measure and  $\nu_\alpha(dy) = C_{n,\alpha}||y||^{-n-\alpha}dy$ ,  $C_{n,\alpha} = \frac{\alpha\Gamma((n+\alpha)/2)}{2^{1-\alpha}\pi^{n/2}\Gamma(1-\alpha/2)}$ .

The stationary probability density function is the solution for the nonlocal equation  $A^*p = 0$ . To use our method to learn the SDE (17) driven by Lévy noise, we only need to change the loss function of residual neural network (10) to (18). As for the nonlocal integral term, we discretize it with a scheme in our earlier work [45], and while for the first order derivative term, we evaluate with automatic differentiation [44].

### 3 Numerical Experiments

We first present an analytical example to learn a simple stochastic system, with quite involved calculations. For more complex stochastic systems, we will have to use our proposed machine learning method as demonstrated in the following numerical experiments.

#### 3.1 Analytical method for learning stochastic dynamical systems

Consider a scalar stochastic differential equation

$$dX_t = b(X_t)dt + \sigma dB_t, \tag{19}$$

with appropriate conditions on drift  $b$  and diffusion  $\sigma$  (see [38, p.170]), such as,  $b \leq 0$  and  $\sigma \neq 0$  as well as some smoothness requirements, there exists a unique stationary probability density  $p(x)$  for the SDE (19), as a solution of the steady Fokker-Planck equation,

$$p(x) = \frac{C}{\sigma^2(x)} e^{\int_{x^*}^x \frac{2b(y)}{\sigma^2(y)} dy}, \tag{20}$$

where the positive normalization constant  $C$  is chosen so that  $p > 0$  and  $\int_{\mathbb{R}} p(x)dx = 1$ , i.e.,

$$C \triangleq 1 / \int_{-\infty}^{\infty} \frac{e^{\int_{x^*}^x \frac{2b(y)}{\sigma^2(y)} dy}}{\sigma^2(x)} dx.$$

Note that  $x^*$  here may be an arbitrary reference point so that the integral  $\int_{x^*}^x \frac{2b(y)}{\sigma^2(y)} dy$  exists. Different choice of  $x^*$  only affects the normalization constant  $C$ . (Say, take  $x^* = 0$  if that is possible).

Given the observed stationary probability density  $q$ , we like to find out the true stationary probability density  $p$ . Consider Hellinger distance  $H$  between probability densities  $p(x)$  and  $q(x)$ .

$$H(b, \sigma) \triangleq \sqrt{\frac{1}{2} \int_{\mathbb{R}} (\sqrt{p(x)} - \sqrt{q(x)})^2 dx}, \tag{21}$$

where  $p(x)$  is in (20). The corresponding Euler-Lagrange equation for  $H^2$  is

$$\frac{d}{dt}H_\sigma^2 = H_b^2. \quad (22)$$

Since the Euler-Lagrange equation is the necessary condition for the functional to obtain the minimum. So we can solve the corresponding Euler-Lagrange equation to get the minimum value of the functional.

**Example 1.** Consider a specific scalar stochastic model

$$dX = b(X)dt + dB_t,$$

with unknown drift  $b(x)$ , and given diffusion  $\sigma = 1$ . Given an “observation” of the stationary probability density  $q(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$  (the Gaussian distribution). Find a function  $b(x)$  so that the Hellinger distance  $H^2(b(x)) = \frac{1}{2} \int_{\mathbb{R}} [\sqrt{p(x)} - \sqrt{q(x)}]^2 dx$  is minimized.

The true stationary probability density for the solution process  $X_t$  is

$$p(x) = \frac{e^{2 \int_0^x b(y) dy}}{\int_{-\infty}^{\infty} e^{2 \int_0^x b(y) dy} dx}. \quad (23)$$

The Euler-Lagrange equation is a necessary condition for functional minima

$$I(b) = \frac{1}{2} \int_{\mathbb{R}} (p(x) + q(x) - 2\sqrt{p(x)}\sqrt{q(x)}) dx. \quad (24)$$

Submitting Eq. (23) and  $q(x)$  into Eq. (24), we get

$$I(b) = \frac{1}{2\sqrt{2\pi}} \int_{\mathbb{R}} e^{-\frac{1}{2}x^2} dx + \frac{1}{2} \int_{\mathbb{R}} \frac{e^{2 \int_0^x b(y) dy}}{\int_{\mathbb{R}} e^{2 \int_0^x b(y) dy} dx} dx - \int_{\mathbb{R}} \sqrt{\frac{e^{-\frac{1}{2}x^2 + 2 \int_0^x b(y) dy}}{\sqrt{2\pi} \int_{\mathbb{R}} e^{2 \int_0^x b(y) dy} dx}} dx. \quad (25)$$

In order to get the minima of  $I(b)$ , we can obtain  $I'(b) = 0$  and  $b(x) = -kx, k \geq 0$ .

Submitting  $b(x)$  into  $p(x)$ , then  $p(x) = \sqrt{\frac{k}{\pi}} e^{-kx^2}$ , which satisfies  $\int_{-\infty}^{\infty} p(x) dx = 1$ .

The error  $Err = \|p(x) - q(x)\|_H$ , submitting  $p(x)$  and  $q(x)$  into  $Err$ :

$$\begin{aligned} Err &= \|p(x) - q(x)\|_H = \frac{1}{2} \int_{\mathbb{R}} (\sqrt{p(x)} - \sqrt{q(x)})^2 dx \\ &= \frac{1}{4\sqrt{\pi}} + \sqrt{\frac{k}{2\pi}} - \sqrt{\frac{4k}{2\pi(k + \frac{1}{2})}}. \end{aligned} \quad (26)$$



This is a function  $f(k)$  about variable  $k$ .  $f_{min}$  attains when  $k = \frac{1}{2}$ . So,  $b(x) = -\frac{1}{2}x$ .

In this example, we can luckily find the optimal drift term  $b$  analytical. While it is exceedingly difficult to compute the true drift term by hand in many problems. So, in the cases below, we use our proposed machine learning method to learn the drift and diffusion terms.

### 3.2 Machine learning for learning stochastic dynamical systems

The neural networks in our numerical experiments below have 4 hidden layers and 20 neurons per layer, with tanh activation function. The weights are initialized with truncated normal distributions. The biases are initialized as zero. We use the Adam optimizer with a learning rate  $10^{-4}$  to train the loss function.

**Example 2.** Consider a scalar stochastic model

$$dX = b(X)dt + \sigma dB_t,$$

with drift function  $b(x) = x - x^3$ . Given an “observation” of the stationary probability density  $q(x) = \frac{1}{A} e^{\frac{1}{\sigma^2}x^2 - \frac{x^4}{2}}$ , where  $A = \int_{\mathbb{R}} e^{\frac{1}{\sigma^2}x^2 - \frac{x^4}{2}} dx$ . Find a drift function  $b(x)$  so that the Hellinger distance  $I(b(x)) = \frac{1}{2} \int_{\mathbb{R}} [\sqrt{p(x)} - \sqrt{q(x)}]^2 dx$  is minimized.

Given the noise intensity (diffusion)  $\sigma = 1$ , we use two fully connected neural networks to approximate the drift term and stationary probability density respectively. We choose  $N_H = 1001$ ,  $N_f = 10000$  to train the loss function (11). The results we learned are shown in Figure 2. In Figure 2 (a), we plot the true drift term (black line) and the learned drift term (red line). The neural network can approximate the drift very well. In Figure 2 (b), the given  $q(x)$  and neural network result of  $p_{NN}(x)$  can approximate well too. We also plot the loss function evolves with the number of iterative steps. The loss is less than  $10^{-4}$ .

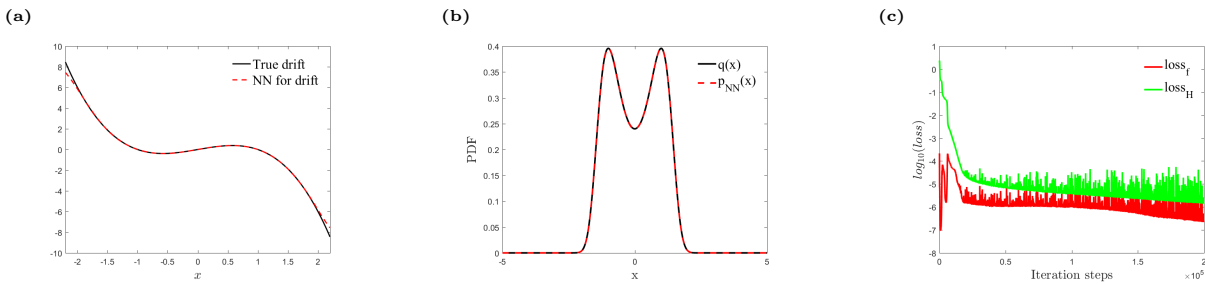


Figure 2: **Unknown drift of Example 2:** (a) learned drift term; (b) learned probability; (c) loss function.

For the unknown drift term and diffusion term, we train the loss function (13) to get the optimal results. Only one observation data of the drift term at  $x = -2$  is given. The results

are shown in Figure 3. We present learned drift result in Figure 3 (a), and the diffusion term evolution predictions as the iteration of the optimiser progresses in Figure 3 (b). The neural network of probability density is shown in Figure 3 (c). We can see for one observation data of the drift term, the drift and diffusion term can be learned well. When  $|x| > 2$ , the error of the learned drift term becomes larger than that for the other  $x$ . The fundamental reason for this is that the probability in bigger  $x$  is almost zero, providing very little information there.

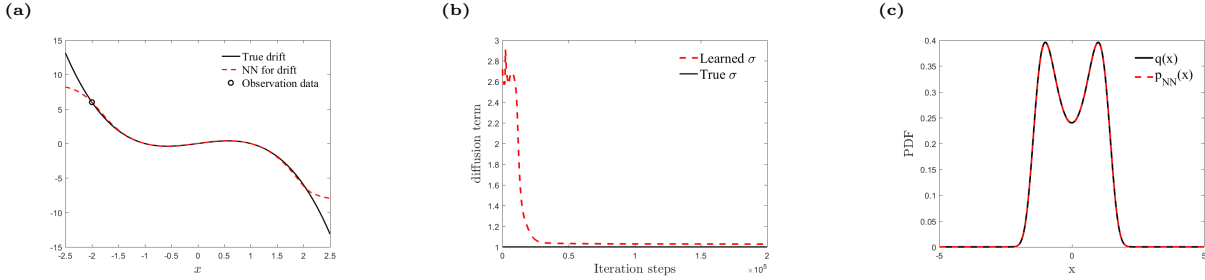


Figure 3: **Unknown drift and diffusion of Example 2:** (a) the learned drift term with 1 observation data at  $x = -2$ ; (b) the learned diffusion term; (c) the learned probability density.

We also use our method to learn the SDE model, with only one trajectory observation data  $X(t)$ . The observation data is the long time trajectory of  $X(t)$  and is shown in Figure 4 (a). We use kernel density estimation to get the probability density and then use our method to learn the drift and diffusion terms. The results of the drift term are shown in Figure 4 (b) and the probability density is shown in Figure 4 (c). The results validate that our method also works with long time trajectory observation data, while the error of the probability density function is larger than using the stationary probability density function observation data. If we have more trajectory data of the  $X(t)$ , we can learn the probability density function better.

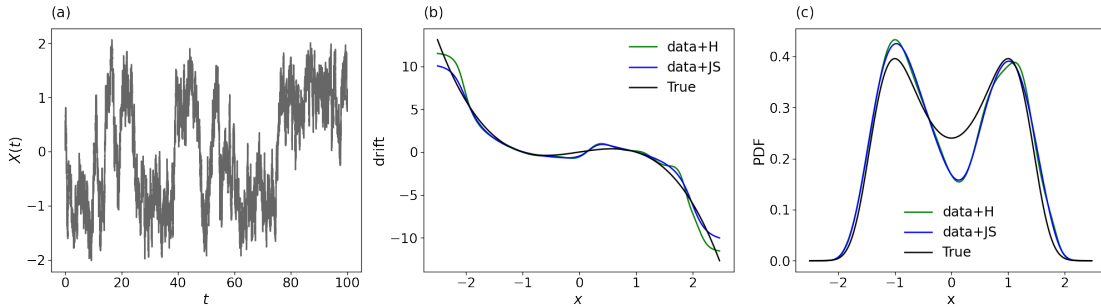


Figure 4: **Unknown drift of Example 2 with one trajectory observation data:** (a) trajectory observation data  $X(t)$ ; (b) the learned drift term; (c) the learned probability density.

**Example 3.** Consider a scalar stochastic model

$$dX = b(X)dt + \sigma dB_t,$$

with drift function  $b(x)$ . Given an “observation” of the stationary probability density  $q(x) = \frac{1}{\pi} \frac{1}{1+x^2}$ . Find a function  $b(x)$  so that the Hellinger distance  $I(b(x)) = \frac{1}{2} \int_{\mathbb{R}} [\sqrt{p(x)} - \sqrt{q(x)}]^2 dx$  is minimized.

Similar to the last example, we fix the noise intensity (diffusion)  $\sigma = 1$ , and use two neural networks to approximate the drift term and stationary probability density respectively. Here we also choose  $N_H = 1001$ ,  $N_f = 10000$ .

$$I(b(x)) = \frac{1}{2} \int_{\mathbb{R}} (\sqrt{p(x)} - \sqrt{q(x)})^2 dx, \quad (27)$$

where  $q(x) = \frac{1}{\pi} \frac{1}{1+x^2}$ .

The results are shown in Figure 5. In this case, the true drift term  $b(x)$  is unknown, so we could not compare the drift term. By comparing the learned stationary distribution and the observation data, the result shows that they match well. What is more, the loss function is sufficiently small, and the probability density distribution is concentrated around zero. So zero could be the stable point of this system. Our learned drift term  $b(x)$  has one stable point zero, as in Figure 5. This is in line with our expectations.

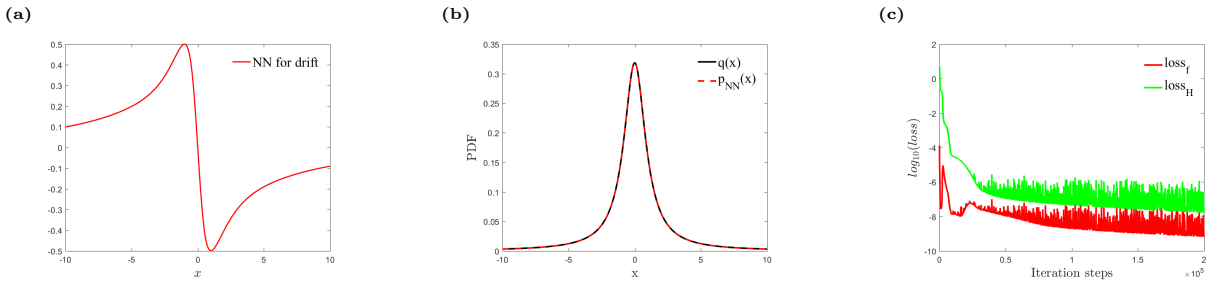


Figure 5: **Unknown drift of Example 3:** (a) learned drift term; (b) learned probability; (c) loss function of OM part  $loss_H$  and equation part  $loss_f$ .

Molecular and cell biology is playing an increasingly important role in life sciences. For example, many research findings are about stochastic fluctuations inducing phenotypic diversity in gene expression. Here we consider a stochastic gene regulation model [41, 42].

**Example 4.** This is a stochastic model for a transcription factor (i.e., a protein) concentration evolution in a certain gene regulation network

$$dX = b(X)dt + \sigma dB_t,$$

with drift function  $b(x) = \frac{k_f x^2}{x^2 + K_d} - k_d x + R_{bas}$ . Here the parameters are  $K_d = 10$ ,  $k_d = 1 \text{min}^{-1}$ ,  $k_f = 6 \text{min}^{-1}$ , and  $R_{bas} = 0.4 \text{min}^{-1}$ . We will find a drift function  $b(x)$  and diffusion  $\sigma$  so that the Hellinger distance  $I(b(x), \sigma) = \frac{1}{2} \int_{\mathbb{R}} [\sqrt{p(x)} - \sqrt{q(x)}]^2 dx$  is minimized.

Given the noise intensity  $\sigma = 1$ , two neural networks are used to approximate the drift term and stationary probability density respectively. Here we still choose  $N_H = 1001$ ,  $N_f = 10000$ . The result of the drift is shown in Figure 6 (a), where the black line is the true drift term and the red line is the neural network result. We find that the learned drift term can fit the true result very well. And the learned probability density function is shown in Figure 6 (b), which fits very well with the observation probability  $q(x)$ . We also show the loss function of  $loss_H$  and  $loss_f$  in Figure 6 (c). We see that the loss function decreases fast with the iteration steps increasing.

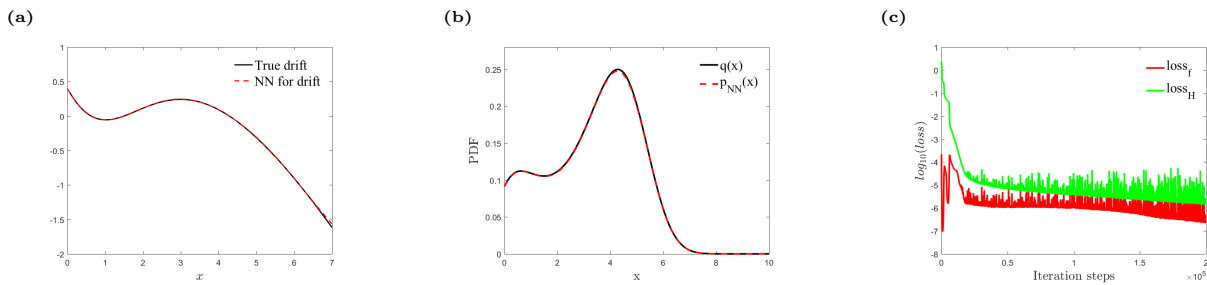


Figure 6: **Unknown drift of Example 4:** (a) The learned drift term; (b) The learned probability; (c) The loss function.

For the unknown drift term and diffusion term case, we train the loss function (13) to get the optimal result. Only one observation data of drift term at  $x = 5$  is given. The results are shown in Figure 7. We present learned drift result in Figure 7 (a), and the diffusion term evolution predictions as the iteration of the optimiser progresses in Figure 7 (b). The neural network result of probability density is shown in Figure 7 (c). From the figures, we see that the drift and diffusion terms can be learned well.

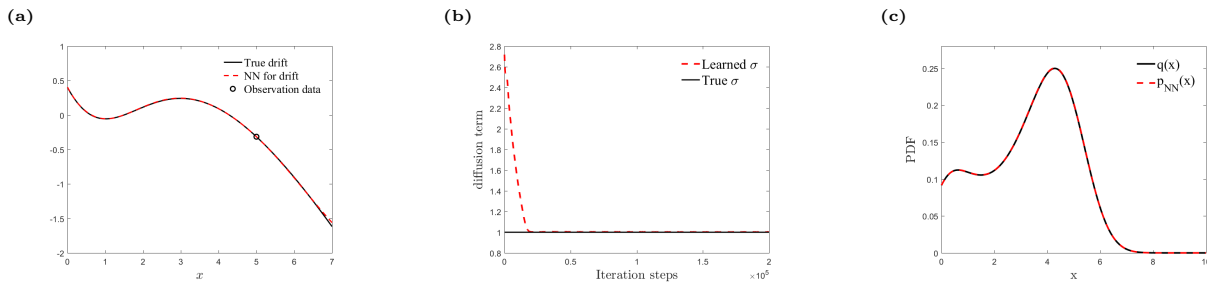


Figure 7: **Unknown drift of Example 4:** (a) learned drift term with 1 observation data at  $x = 5$ ; (b) learned diffusion term; (c) learned probability density function.

**Example 5.** We now consider the following three dimensional stochastic dynamical systems with non-polynomial drift [31]:

$$d \begin{pmatrix} X_t \\ Y_t \\ Z_t \end{pmatrix} = \begin{pmatrix} -\partial_{X_t} \Phi(X_t, Y_t, Z_t) \\ -\partial_{Y_t} \Phi(X_t, Y_t, Z_t) \\ -\partial_{Z_t} \Phi(X_t, Y_t, Z_t) \end{pmatrix} dt + \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} d \begin{pmatrix} B_{1,t} \\ B_{2,t} \\ B_{3,t} \end{pmatrix}, \quad (28)$$

where the potential  $\Phi(x, y, z) = -\frac{1}{2} \log[(2 \exp(\lambda_{01}(x - \lambda_{11})^2 + \lambda_{02}(y - \lambda_{12})^2 + \lambda_{03}(z - \lambda_{13})^2) + \exp(\lambda_{04}(x - \lambda_{14})^2 + \lambda_{05}(y - \lambda_{15})^2 + \lambda_{06}(z - \lambda_{16})^2)]$ ,  $\lambda_{0i} = -5, -2.5, -5, -1, -1, -1$ ,  $\lambda_{1i} = 1, 1, 1, -2, -1, -1$  and  $\sigma_j = 1$ , where  $i = 1, 2, \dots, 6$  and  $j = 1, 2, 3$ . The “observation” of the stationary probability density is  $q(x, y, z) = 1/Z \exp(-2\Phi(x, y, z))$ , where  $Z$  is the normalization parameter such that the integral of  $q(x, y, z)$  on domain  $\mathbb{R}^3$  is equal to 1. Find the parameters in drift term and diffusion term so that the Hellinger distance  $I = \frac{1}{2} \int_{\mathbb{R}^3} [\sqrt{p(x, y, z)} - \sqrt{q(x, y, z)}]^2 dx dy dz$  is minimized.

We use neural network to approximate the stationary probability density. And here we choose  $N_H = 50000$ ,  $N_f = 5000$ .

First, we learn all the parameters  $\lambda_{0i}$ ,  $\lambda_{1i}$  and  $\sigma_j$ , for  $i = 1, 2, \dots, 6$  and  $j = 1, 2, 3$ . The results are shown in Figure 8. In Figure 8(a) and (c), the parameters of  $\lambda_{0i}$  and drift term are learned not well. While the parameters  $\lambda_{2i}$  can be learned well, see Figure 8 (b). So we will learn the parameter in the drift term and diffusion term respectively.

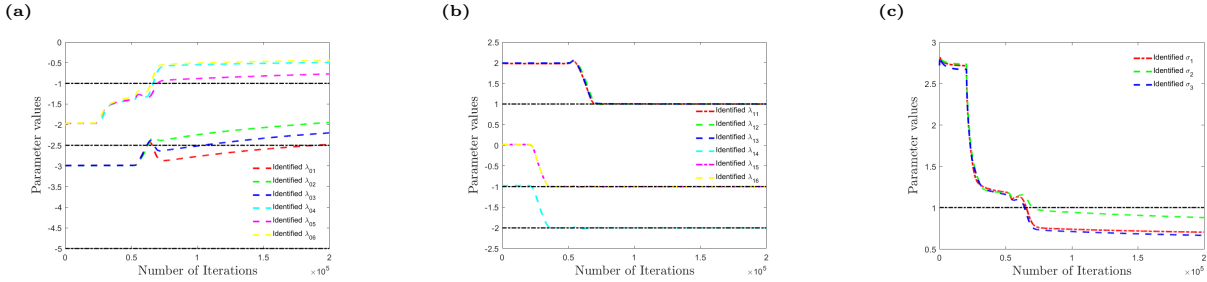


Figure 8: **Three dimensional results of Example 5:** (a) learned  $\lambda_{0i}$ ; (b) learned  $\lambda_{1i}$ ; (c) learned  $\sigma_j$ , where  $i = 1, 2, \dots, 6$  and  $j = 1, 2, 3$ .

On the one hand, we just learn the diffusion term given the drift term. The results are shown in Figure 9. The parameters in the diffusion term approach to the true parameter as the number of iterations increases.

On the other hand, we learn the parameters  $\lambda_{0i}$  and  $\lambda_{1i}$  in the drift term given the diffusion term. The results are shown in Figure 10. We learn the results for three cases. For case I: the observation data is clean, i.e.  $q(x, y, z)$ . For case II: the observation data is given with 5% noise, i.e.  $q(x, y, z) * (1 + 0.05N(0, I))$ , and for case III: the observation data is given with 10% noise, i.e.  $q(x, y, z) * (1 + 0.1N(0, I))$ . Here  $N(0, I)$  mean the standard normal distribution. The parameters we learned are well even the observation data has 10% noise.

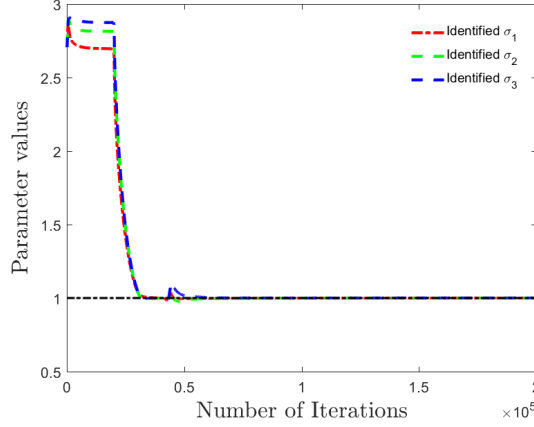


Figure 9: **3D result** the learned drift term of Example 5.

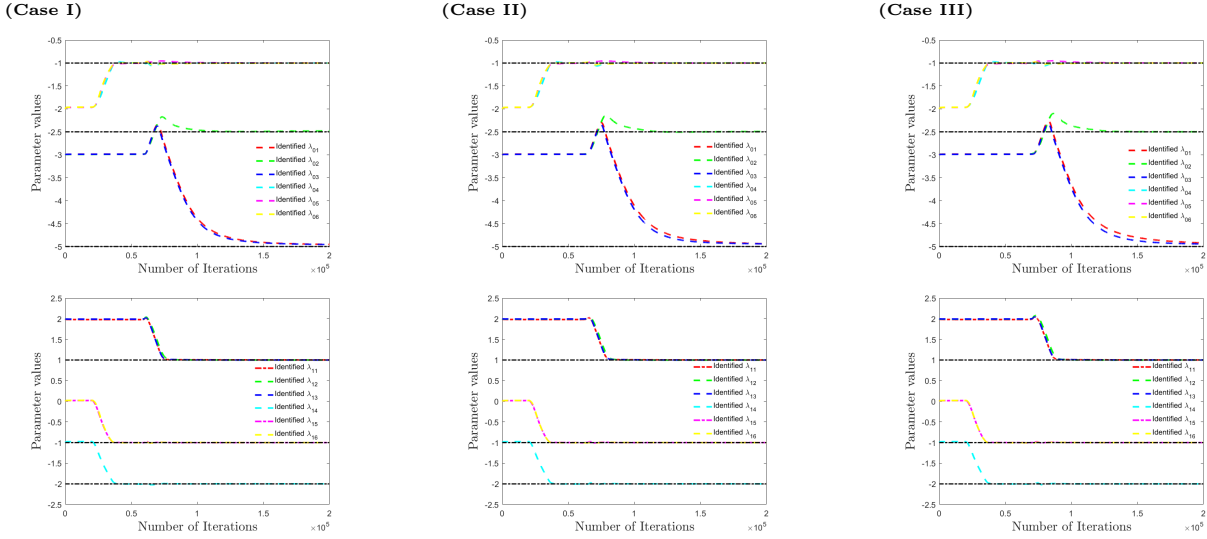


Figure 10: **Three dimensional results of Example 5.** Learn all parameters in the drift terms with perturbation. Case I: clean observation data of the PDF; Case II: 5% noise observation data of the PDF; Case III: 10% noise observation data of the PDF. Left: learned  $\lambda_{0i}$ ; right: learned  $\lambda_{1i}$ , where  $i = 1, 2, \dots, 6$  and  $j = 1, 2, 3$ .

In the following, We change the Hellinger distance to the Jensen-Shannon divergence in the loss function. The results of learned parameters in the drift term are shown in Figure (11). The unknown parameters can be learned well. While compared with the Hellinger distance, this method needs more iteration steps to train.

Here we also compare our results with the traditional physics informed neural network (PINN) with the case of learning the parameter in the drift term. The results are shown in Figure 12. Only several parameters in the drift term can be learned well using PINN method. Compared with PINN method using mean square error (16), our loss with Hellinger

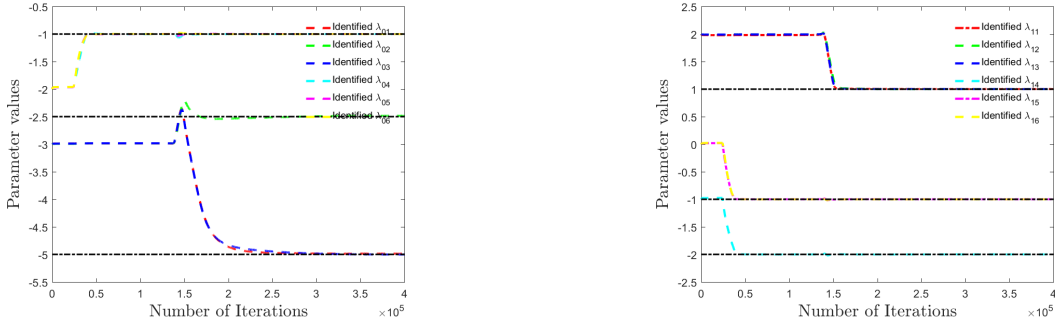


Figure 11: **Three dimensional results of Example 5.** Learn parameters in the drift terms using Jensen-Shannon distance. Left: learned  $\lambda_{0i}$ ; right: learned  $\lambda_{1i}$ , where  $i = 1, 2, \dots, 6$  and  $j = 1, 2, 3$ .

distance (11) would get better results. We use the neural network to approximate the probability density function and plot the learned probability density function when  $z = -1, 0.5, 1$  using different methods. The results are shown in Figure 13. Compared with the true probability density function, the proposed method with Hillinger distance and Jensen-Shannon divergence works better than the mean square distance.

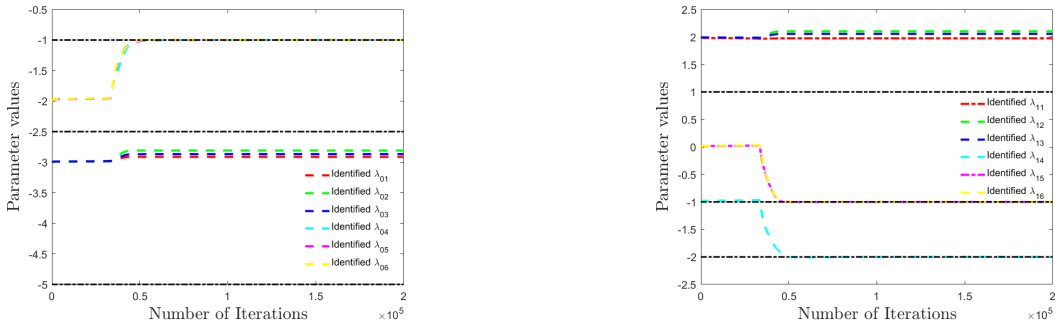


Figure 12: **Three dimensional results with PINN loss of Example 5.** Learn all the parameters in the drift term with clean observation data of the PDF. Left: learned  $\lambda_{0i}$ ; right: learned  $\lambda_{1i}$ , where  $i = 1, 2, \dots, 6$  and  $j = 1, 2, 3$ .

For the unknown drift term, we use our proposed method to recover the drift term. The results are shown in Figure 14. The first row of the Figure 14 (a1,b1,c1,d1) are the the projection of  $\Phi(x, y, z)$  var minimization of  $z$ , i.e.  $\min_z \Phi(x, y, z)$ . The second and third rows are projected on  $(x, z)$  domain and  $(y, z)$  domain separately. The true projections are shown in Figure 14(a). The results with Hellinger distance are shown in Figure 14(b). Comparing with the true potential, this example illustrates our method with Hellinger distance works well. However, it is difficult to recover the SDE with the Jensen-Shannon divergence (see Figure 14(c)) and PINN losses (see Figure 14(d)). This indicates the efficiency of our proposed method using the Hellinger distance in this example.

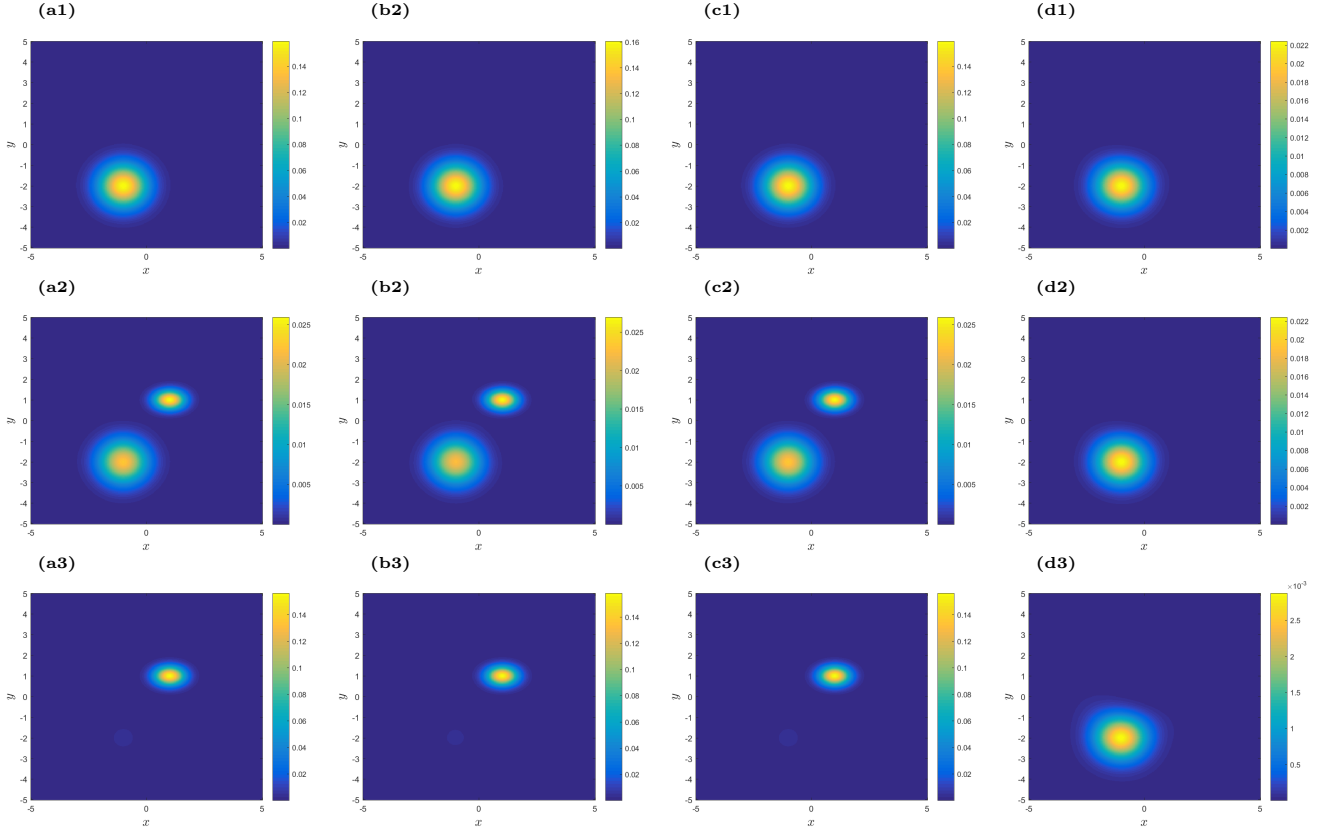


Figure 13: **The probability density function of Example 5.** (a1)-(a3): the true PDF with  $z = -1, 0.5, 1$ ; (b1)-(b3): the learned PDF using Hellinger distance (11); (c1)-(c3): the learned PDF using Jensen-Shannon divergence loss (14); (d1)-(d3): the learned PDF using PINN loss (16).

**Example 6.** Finally, We now consider the following five dimensional stochastic dynamical systems with non-polynomial drift:

$$d \begin{pmatrix} X_t \\ Y_t \\ Z_t \\ V_t \\ W_t \end{pmatrix} = \begin{pmatrix} -\partial_{X_t} \Phi(X_t, Y_t, Z_t, V_t, W_t) \\ -\partial_{Y_t} \Phi(X_t, Y_t, Z_t, V_t, W_t) \\ -\partial_{Z_t} \Phi(X_t, Y_t, Z_t, V_t, W_t) \\ -\partial_{V_t} \Phi(X_t, Y_t, Z_t, V_t, W_t) \\ -\partial_{W_t} \Phi(X_t, Y_t, Z_t, V_t, W_t) \end{pmatrix} dt + \begin{bmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \\ 0 & 0 & 0 & \sigma_4 & 0 \\ 0 & 0 & 0 & 0 & \sigma_5 \end{bmatrix} d \begin{pmatrix} B_{1,t} \\ B_{2,t} \\ B_{3,t} \\ B_{4,t} \\ B_{5,t} \end{pmatrix},$$

where the potential  $\Phi(x, y, z, v, w) = -\frac{1}{2} \log[\exp((\lambda_{01}(x - \lambda_{11}) + \lambda_{02}(y - \lambda_{12}) + \lambda_{03}(z - \lambda_{13}) + \lambda_{04}(v - \lambda_{14}) + \lambda_{05}(w - \lambda_{15}))) + \exp((\lambda_{06}(x - \lambda_{16}) + \lambda_{07}(y - \lambda_{17})) + \lambda_{08}(z - \lambda_{18}) + \lambda_{09}(v - \lambda_{19}) + \lambda_{10}(w - \lambda_{110}))]$ ,  $\lambda_{0i} = -1$ ,  $\lambda_{1i} = (1, 1, 1, 1.5, 1.5, -2, -1, -1, -1, -2)$  where  $i = 1, 2, \dots, 10$ , and  $\sigma_j = 1$ ,  $j = 1, 2, 3, 4, 5$ . The “observation” of the stationary probability density is  $q(x, y, z, v, w) = 1/Z \exp(-2\Phi(x, y, z, v, w))$ , where  $Z$  is the normalization parameter such that the integral of  $q(x, y, z, v, w)$  on domain  $\mathbb{R}^5$  is equal to 1. Find the parameters in drift



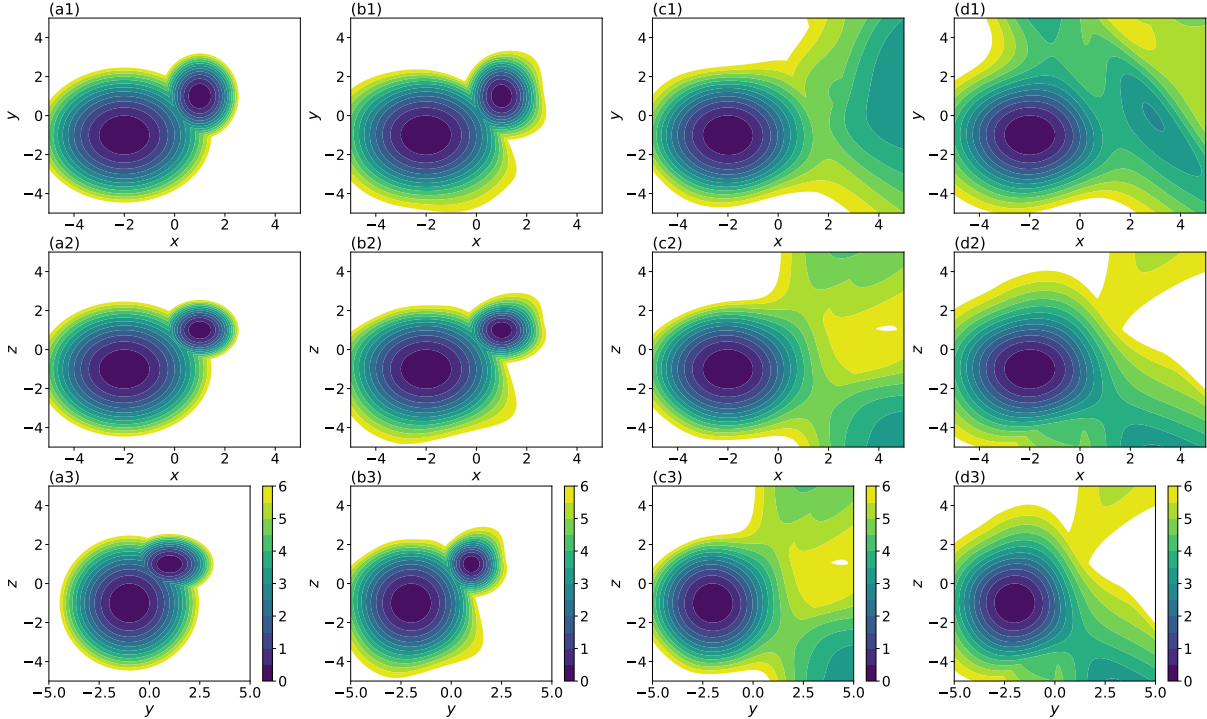


Figure 14: **The learned potential of Example 5.** (a1)-(a3): the true potential; (b1)-(b3): the learned potential using Hellinger distance (11); (c1)-(c3): the learned potential using Jensen-Shannon divergence loss (14); (d1)-(d3): the learned potential using PINN loss (16).

term so that the Hellinger distance  $I = \frac{1}{2} \int_{\mathbb{R}^5} [\sqrt{p(x, y, z, v, w)} - \sqrt{q(x, y, z, v, w)}]^2 dx dy dz$  is minimized.

We use a neural network to approximate the stationary probability density. And here we choose  $N_H = 50000$ ,  $N_f = 5000$ . We learn the parameters  $\lambda_{0i}$  and  $\lambda_{1i}$  with  $i = 1, 2, \dots, 10$  in the drift term given the diffusion term. The results are shown in Figure 15, indicating that our method also works for five dimensional case.

Remark: With only one observation trajectory data, we first use kernel density estimation to approximate the stationary probability density function, and then learn the SDE model. This works well for one dimension, while for high dimensional cases, we need data on multiple trajectories.

## 4 Discussion

Based on minimizing Hellinger distance between two probability distributions, we have devised a data-driven method to extract stochastic dynamical systems models from observation

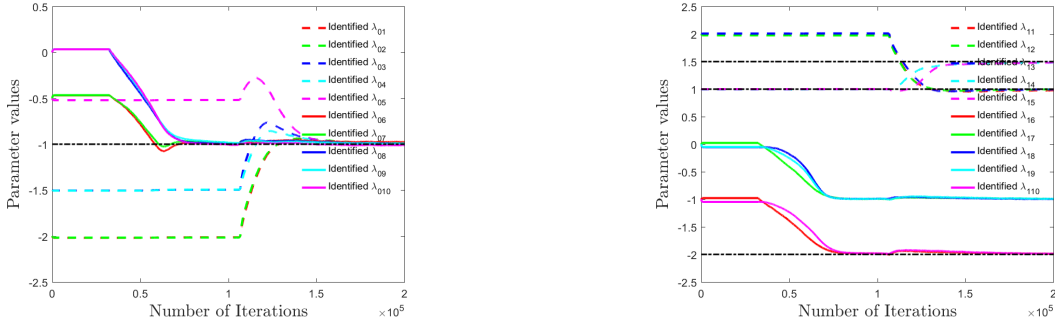


Figure 15: **Five dimensional results of Example 6.** Learn all the parameters in the drift term with clean observation data of the PDF. Left: learned  $\lambda_{0i}$ ; right: learned  $\lambda_{1i}$ , where  $i = 1, 2, \dots, 10$ .

data of either long time trajectories or stationary probability distributions. Our numerical results in one, three dimensional and five dimensional examples have verified that this method is feasible. We may also take other distances, in the space of probability distributions, in our method. Indeed, we have also tried our method using Jensen-Shannon divergence and mean square distance.

In principle, we may extend our method to learn high dimensional stochastic dynamical systems. But when dealing with a high dimensional case, the larger search space for the Fokker-Planck equation makes it difficult to train the neural network. We will try to use parallel computing [43] or active sampling [46] to train the neural network. Moreover, our method is more stringent on data requirements for higher dimensions cases. Therefore, in the future, we are going to explore other method to recover stochastic differential equation models. We will also try to use our method to learn the stochastic differential equations with non-Gaussian Lévy noise.

This approach leads to a data-driven stochastic dynamical systems study of random phenomena, as we can further examine dynamical behaviors of the leaned stochastic governing models [33].

## Acknowledgements

We would like to thank Xi Chen for helpful discussions. This work is supported by the National Natural Science Foundation of China (NSFC) (Grant No.11901536, 12141107). Xiaoli Chen is supported by the Ministry of Education, Singapore, under its Research Centre of Excellence award to the Institute for Functional Intelligent Materials (I-FIM, project No. EDUNC-33-18-279-V12)

## CRediT authorship contribution statement

X. Chen: Conceptualization, Software, Formal analysis, Writing-original draft. H. Wang: Conceptualization, Formal analysis, Writing-original draft. J. Duan: Discussion and Suggestion.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

- [1] L. Arnold, *Random Dynamical Systems*. New York, Springer,(2003) Corrected 2nd printing.
- [2] C. Pasquero, E. Tziperman, Statistical parameterization of heterogeneous oceanic convection, *Journal of Physical Oceanography*. (2007) 37: 214-229.
- [3] C. Penland, P. Sura, Sensitivity of an ocean model to “details” of stochastic forcing. In *Proc. ECMWF Workshop on Representation of Subscale Processes using Stochastic-Dynamic Models*. Reading, England, (2005)6-8 June.
- [4] T. Gao, J. Duan, Quantifying model uncertainty in dynamical systems driven by non-gaussian Lévy stable noise with observations on mean exit time or escape probability. *Communications in Nonlinear Science and Numerical Simulation*. 39(2016) 1-6.
- [5] D. Wu, M. Fu, J. Duan, Discovering mean residence time and escape probability from data of stochastic dynamical systems. *Chaos*. 29(9)(2019)093122.
- [6] C. L. Hung, X. Zhang, N. Gemelke, C. Chin, Observation of scale invariance and universality in two-dimensional Bose gases. *Nature*. 470(7333)(2011) 236-239.
- [7] G. Hairapetian, R. Stenzel, Observation of a stationary, current-free double layer in a plasma. *Physical Review Letters*. 65(2)(1990)175.
- [8] E. J. Yarmchuk, M. J. V. Gordon, R. E. Packard, Observation of Stationary Vortex Arrays in Rotating Superfluid Helium. *Physical Review Letters*. 43(3)(1979)214-217.
- [9] O. Gefen, O. Fridman, I. Ronin, N. Q. Balaban, Direct observation of single stationary-phase bacteria reveals a surprisingly long period of constant protein production activity. *Proceedings of the National Academy of Sciences*. 111(1)(2014)556-561.

- [10] L. Arnold, V. Wishtutz, Stationary solutions of linear systems with additive and multiplicative noise. *Stochastics-an International Journal of Probability & Stochastic Processes*. 7(1-2)(1982)133-155.
- [11] D. Liberzon, R. W. Brockett, Nonlinear feedback systems perturbed by noise: steady-state probability distributions and optimal control. *Automatic Control IEEE Transactions on*. 45(6)(2000)1116-1130.
- [12] A. H. Gray, Uniqueness of Steady-State Solutions to the Fokker-Planck Equation. *Journal of Mathematical Physics*. 6(4)(1965)644-647.
- [13] R. Khasminskii, *Stochastic stability of differential equations*,(2012) Springer.
- [14] B. Schmalfuss, Lyapunov functions and non-trivial stationary solutions of stochastic differential equations. *Dynamical Systems: An International Journal*. 16(4)(2001) 303-317.
- [15] F. Gerber, D. W. Nychka, Fast covariance parameter estimation of spatial Gaussian process models using neural networks. *Statistics and Probability*.(2021) 10(1).
- [16] P. Batz, A. Ruttor, M. Opper, Variational estimation of the drift for stochastic differential equations from the empirical density. *Journal of Statistical Mechanics- Theory and Experiment*. (8)(2016) 083404.
- [17] P. Batz, A. Ruttor, M. Opper, Approximate Bayes learning of stochastic differential equations. *Physical Review E*. 98(2)(2018) 022109.
- [18] M. Opper, An estimator for the relative entropy rate of path measures for stochastic differential equations. *Journal of Computational Physics*. 330(2017) 127-133.
- [19] M. Opper, Variational inference for stochastic differential equations. *Annals of Physics*. 531(3)(2019) 1800233.
- [20] T. Ryder, A. Golightly, A. S. McGough, D. Prangle, Black-box variational inference for stochastic differential equations. *ICML*.(2018) 4423-4432.
- [21] L. Boninsegna, F. Nüske, C. Clementi, Sparse learning of stochastic dynamical equations. *Journal of chemical physics*. 148(24)(2018) 241723.
- [22] R. Tabar, *Analysis and data-based reconstruction of complex nonlinear dynamical systems*, Berlin/Heidelberg, (2019)Germany: Springer.
- [23] X. Li, T. K. L. Wong, R. T. Chen, D. Duvenaud, Scalable gradients for stochastic differential equations. *AISTATS*. (2020) 3870-3882.
- [24] J. Jia, A. R. Benson, Neural jump stochastic differential equations, *NIPS*. (2019)32.

- [25] L. Yang, C. Daskalakis, 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(1)(2022) B80-B99.
- [26] H. Zhang, Y. Xu, Y. Li, J. Kurths, Statistical solution to SDEs with  $\alpha$ -stable Lévy noise via deep neural network. *International Journal of Dynamics and Control*. 8(4)(2020) 1129-1140.
- [27] Y. Xu, H. Zhang, Y. Li, K. Zhou, Q. Liu, J. Kurths, Solving Fokker-Planck equation using deep learning. *Chaos*. 30(1)(2020) 013133.
- [28] H. Zhang, Y. Xu, Q. Liu, X. Wang, Y. Li, Solving Fokker-Planck equations using deep kd-tree with a small amount of data. *Nonlinear Dynamics*. (2022) <https://doi.org/10.1007/s11071-022-07361-2>.
- [29] Y. Li, J. Duan, A data-driven approach for discovering stochastic dynamical systems with non-Gaussian Lévy noise. *Physica D*. 417(2021) 132830.
- [30] Y. Lu, Y. Li, J. Duan, Extracting stochastic governing laws by non-local Kramers-CMoyal formulae. *Phil. Trans. R. Soc. A*. 380, (2022) 20210195.
- [31] X. Chen, L. Yang, J. Duan, G. E. Karniadakis, Solving Inverse Stochastic Problems from Discrete Particle Observations Using the Fokker-Planck Equation and Physics-Informed Neural Networks. *SIAM Journal on Scientific Computing*. 43(3)(2021) B811-30.
- [32] Y. Yang, L. Nurbekyan, E. Negrini, R. Martin, M. Pasha. Optimal transport for parameter identification of chaotic dynamics via invariant measures. arXiv preprint arXiv:2104.15138 (2021).
- [33] J. Duan, *An Introduction to Stochastic Dynamics*, New York(2015) Cambridge University Press.
- [34] F. Dietrich, A. Makeev, G. Kevrekidis, N. Evangelou, T. Bertalan, S. Reich, I. Kevrekidis. Learning effective stochastic differential equations from microscopic simulations: combining stochastic numerics and deep learning. arXiv preprint arXiv:2106.09004 (2021).
- [35] X. Chen, J. Duan, J. Hu, D. Li. Data-driven method to learn the most probable transition pathway and stochastic differential equation. *Physica D: Nonlinear Phenomena* 443 (2023) 133559.
- [36] S. Cha, Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*. 1(4)(2007) 300-307.

- [37] R. Beran, Minimum hellinger distance estimates for parametric models. *Annals of Statistics*. 5(3)(1977)445-463.
- [38] F. C. Klebaner, *Introduction to Stochastic Calculus with Applications*. Imperial College Press, London, (2005)2nd edition.
- [39] M. Raissi, P. Perdikaris, 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.
- [40] X. Chen, J. Duan, G. E. Karniadakis, Learning and meta-learning of stochastic advection-diffusion-reaction systems from sparse measurements. *European Journal of Applied Mathematics*. 32(3) (2020)397-420.
- [41] H. Wang, X. Cheng, J. Duan, J. Kurths, X. Li, Likelihood for transcriptions in a genetic regulatory system under asymmetric stable Lévy noise. *Chaos*. 28 (2018) 013121.
- [42] X. Cheng, H. Wang, X. Wang, J. Duan, X. Li, Most probable transition pathways and maximal likely trajectories in a genetic regulatory system. *Physica A*. 531 (2019) 121779.
- [43] K. Shukla, D. Ameya, G. Karniadakis. Parallel physics-informed neural networks via domain decomposition. *Journal of Computational Physics* 447 (2021): 110683.
- [44] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18 (2018): 1-43.
- [45] X. Chen, F. Wu, J. Duan, J. Kurths, X. Li. Most probable dynamics of a genetic regulatory network under stable Lévy noise. *Applied Mathematics and Computation*, 348 (2019): 425-436.
- [46] X. Yang, Y. Liu, C. Mi, X. Wang. Active learning Kriging model combining with kernel-density-estimation-based importance sampling method for the estimation of low failure probability. *Journal of Mechanical Design*, 140 (2018): 051402.