Modeling Unknown Stochastic Dynamical Systems via Autoencoder

Zhongshu Xu, Yuan Chen, Qifan Chen, and Dongbin Xiu

Department of Mathematics
The Ohio State University

December, 2024

1. Introduction
2. Method
3. Numerical results
4. Explanation

Setting

Consider an Itô type stochastic differential equations (SDEs)

$$dx_t = a(x_t)\,dt + b(x_t)\,dW_t \tag{1}$$

- W_t: m-dimensional Brownian motion. **Unobservable**
- $a : \mathbb{R}^d \to \mathbb{R}^d$: the drift function. **Unknown**
- $b : \mathbb{R}^d \to \mathbb{R}^{d\times m}$: the diffusion function. **Unknown**

Known fact: x is time-homogeneous, for any $\Delta \geq 0$, $s \geq 0$,

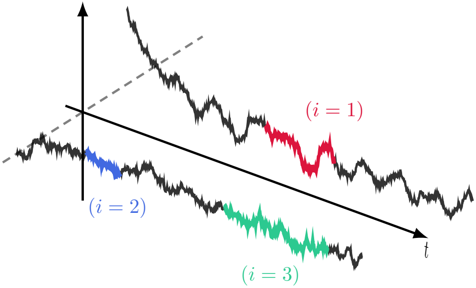
$$\mathbb{P}(x_{s+\Delta}|x_s) = \mathbb{P}(x_\Delta|x_0). \tag{2}$$

Data

We have the solution trajectory data of (1):

$$\mathrm{X}^{(i)} = \left(\mathrm{x}_0^{(i)}, \mathrm{x}_1^{(i)}, \ldots, \mathrm{x}_L^{(i)}\right), \qquad i = 1, \ldots, N_T, \tag{3}$$

where $\mathrm{x}_k^{(i)} = \mathrm{x}(t_k^{(i)})$, with $t_k^{(i)} - t_{k-1}^{(i)} \equiv \Delta, \forall k, i$.



Objective

Based on time-homogeneity, we know there is an exact **"stochastic flow map"** G_Δ from x_t to $\mathrm{x}_{t+\Delta}$.

$$\mathrm{x}_{t+\Delta} = \mathrm{G}_\Delta(\mathrm{x}_t; w) \tag{4}$$

where $w \in \mathbb{R}^m$ is a sample from $\mathrm{W} \sim N(0, I_m)$.

<u>**Goal**</u>: based on the trajectory data (3), construct a numerical model $\widetilde{\mathrm{G}}_\Delta$, such that for all x,

$$\widetilde{\mathrm{G}}_\Delta(\mathrm{x}; \mathrm{Z}) \overset{d}{\approx} \mathrm{G}_\Delta(\mathrm{x}; \mathrm{W}). \tag{5}$$

where $\mathrm{Z} \sim N(0, I_{n_z})$ with $n_z \geq 1$.

Example

Let's consider a simple example:

$$\mathrm{x}_{t+\Delta} = \mathrm{G}_\Delta(\mathrm{x}_t; w) = \mathrm{x}_t + \mathrm{w} \tag{6}$$

Once w is sampled, $\mathrm{x}_{t+\Delta}$ can be uniquely determined by x_t and w.

Example

Let's consider a simple example:

$$\mathrm{x}_{t+\Delta} = \mathrm{G}_\Delta(\mathrm{x}_t; w) = \mathrm{x}_t + \mathrm{w} \tag{6}$$

Once w is sampled, $\mathrm{x}_{t+\Delta}$ can be uniquely determined by x_t and w.

$$\mathrm{w} = \mathrm{x}_{t+\Delta} - \mathrm{x}_t \tag{7}$$

Adjacent data pair $(\mathrm{x}_t, \mathrm{x}_{t+\Delta})$ can also uniquely determine w.

Auto-encoder

Train a Neural Network Encoder $\widetilde{\mathrm{E}}$ to identify the unobserved stochastic component:

$$(x_0, x_1) \xrightarrow{\widetilde{\mathrm{E}}} z$$

and a Neural Network Decoder $\widetilde{\mathrm{D}}$ to reconstruct the trajectory

$$(x_0, z) \xrightarrow{\widetilde{\mathrm{D}}} \widetilde{x}_1$$

such that $\widetilde{x}_1 \approx x_1$.

Counter-Example 1

For $x_1 = G_\Delta(x_0; w) = x_0 + w$.
We have a counter-example:

$$\widetilde{E}\left(x_0, x_1\right) = x_1 - 0.1x_0 := z, \quad \widetilde{D}\left(x_0, z\right) = 0.1x_0 + z$$

- $\widetilde{D}\left(x_0, \widetilde{E}\left(x_0, x_1\right)\right) = x_1$
- $\widetilde{D}(x_0, z) = 0.1x_0 + z \neq G_\Delta(x_0; z)$

Counter-Example 1

For $x_1 = G_\Delta(x_0; w) = x_0 + w$.
We have a counter-example:

$$\widetilde{E}\left(x_0, x_1\right) = x_1 - 0.1x_0 := z, \quad \widetilde{D}\left(x_0, z\right) = 0.1x_0 + z$$

- $\widetilde{D}\left(x_0, \widetilde{E}\left(x_0, x_1\right)\right) = x_1$
- $\widetilde{D}(x_0, z) = 0.1x_0 + z \neq G_\Delta(x_0; z)$

Therefore, we need to further enforce

$$z \sim \mathcal{N}\left(0, I\right)$$

Enforce Density

To enforce the distribution of z, we need a batch of samples,

1. Input: $B_x = \left\{ \left(\mathrm{x}_0^{(i)}, \mathrm{x}_1^{(i)} \right), i = 1, \ldots, N \right\}$
2. Encoder: $\mathrm{z}^{(i)} = \mathrm{E}_\Delta \left(\mathrm{x}_0^{(i)}, \mathrm{x}_1^{(i)} \right), \quad i = 1, \ldots, N.$
3. Output: $B_z = \left\{ \mathrm{z}^{(i)}, i = 1, \ldots, N \right\}$
4. Distributional loss function:

$$\mathcal{L}_D(B_z) = \mathcal{L}_{\text{distance}} \left(B_z, \mathcal{N}(0, \mathrm{I}) \right) + \tau \cdot \mathcal{L}_{\text{moment}} \left(B_z, \mathcal{N}(0, \mathrm{I}) \right)$$

Network Structure

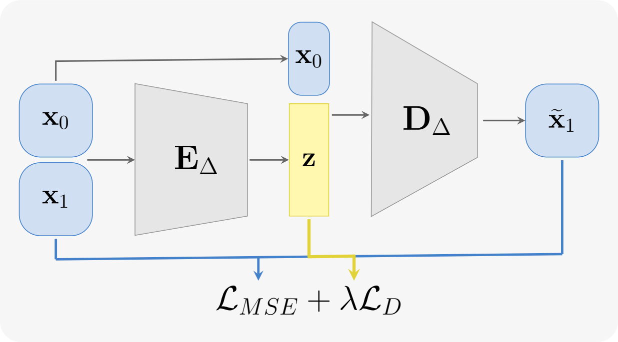


Figure 1: An illustration of the network structure and training loss for the proposed autoencoder sFML method.

Counter-Example 2

For $x_1 = G_\Delta(x_0; w) = x_0 + w$.
We have another counter-example:

$$\widetilde{E}(x_0, x_1) = \frac{1}{3}x_1 + \frac{1}{3}x_0 := z, \quad \widetilde{D}(x_0, z) = 3z - x_0$$

- $\widetilde{E}(x_0, x_1) \sim \mathcal{N}(0, I)$ when $x_0 \sim \mathcal{N}(0, I)$
- $\widetilde{D}\left(x_0, \widetilde{E}(x_0, x_1)\right) = x_1$
- $\widetilde{D}(x_0, z) = -x_0 + 3z \neq x_0 + z$

Counter-Example 2

For $x_1 = G_\Delta(x_0; w) = x_0 + w$.
We have another counter-example:

$$\widetilde{E}(x_0, x_1) = \frac{1}{3}x_1 + \frac{1}{3}x_0 := z, \quad \widetilde{D}(x_0, z) = 3z - x_0$$

- $\widetilde{E}(x_0, x_1) \sim \mathcal{N}(0, I)$ when $x_0 \sim \mathcal{N}(0, I)$
- $\widetilde{D}\left(x_0, \widetilde{E}(x_0, x_1)\right) = x_1$
- $\widetilde{D}(x_0, z) = -x_0 + 3z \neq x_0 + z$

Therefore, we need to enforce further z is independent of x_0.

Theorem (Weak convergence)

Assume random variable $X_0 \in \mathbb{R}^d$ *and* $Z \sim N(0, I_m)$. *We define* $X_1 = G_\Delta(X_0, Z)$. x_0, x_1 *and* z *are sampled from* X_0, X_1, Z. *If continuous functions* D_Δ, E_Δ *satisfied the following conditions:*

1. *For any* x_0, $x_1 \in \mathbb{R}^d$, $D_\Delta\left(x_0, E_\Delta\left(x_0, x_1\right)\right) = x_1$;
2. $E_\Delta(X_0, X_1) \sim N(0, I_m)$ *holds for* $X_0 \sim \delta(x), \forall x \in \mathbb{R}^d$. *Here* $\delta(x)$ *is Dirac delta distribution at* x.

Then for any $x_0 \in \mathbb{R}^d$ *we have:*

$$D_\Delta(x_0, Z) \stackrel{d}{=} G_\Delta(x_0, Z).$$

Sampling Strategy

- Separate initial trajectory into data pair:

$$\left(\mathrm{x}_0^{(i)}, \mathrm{x}_1^{(i)}\right), \left(\mathrm{x}_1^{(i)}, \mathrm{x}_2^{(i)}\right), \ldots, \left(\mathrm{x}_{L-1}^{(i)}, \mathrm{x}_L^{(i)}\right), \quad i = 1, \ldots, N_T \tag{8}$$

- Removal of time:

$$\left(\mathrm{x}_0^{(i)}, \mathrm{x}_1^{(i)}\right), \quad i = 1, \ldots, M, \quad M = N_T L \tag{9}$$

- Resample:
  1. Randomly choose n_B samples of x_0 from (9) using uniform distribution;
  2. For each of the chosen x_0 samples, find its $(N-1)$ nearest neighbor points to form a batch with N samples.

Sampling Result

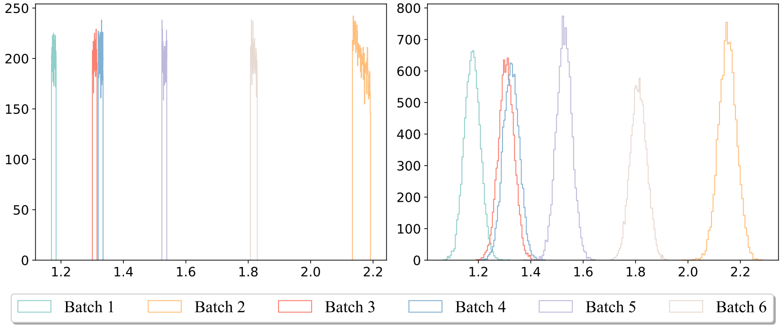


Figure 2: An illustration of the proposed batch sampling: Left: histograms of 6 batches of sampled x_0's; Right: histograms of the corresponding x_1's.

Configuration

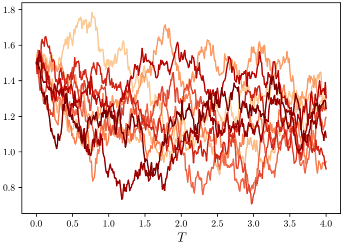
- **Data Generation**: Uniformly sample $N_T = 10^4$ initial conditions within a region and solve the SDE with a time step $\Delta = 0.01$ up to $T = 1.0$.
- **Sampling**: Cut these trajectories into $M = 10^6$ data pairs, then sub-sample into $n_B = 1,000$ batches, each of which contains $N = 10,000$ data pairs.
- **Network Architecture**: Both encoder and decoder are 4 layer 20 nodes per layer DNNs with eLU activation function for the first three layers.
- **Training**: With learning rate $1e-4$, we train our model for 1000 epochs.

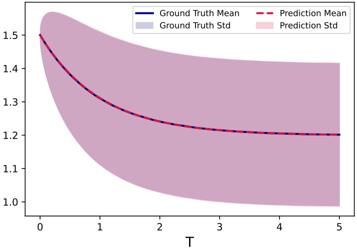
Ornstein–Uhlenbeck process

We consider the 1D OU process, in the following form

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t, \tag{10}$$

for $\theta = 1.0$, $\mu = 1.2$, and $\sigma = 0.3$:





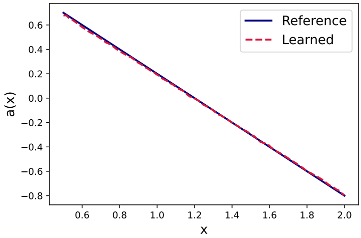
Ornstein–Uhlenbeck process

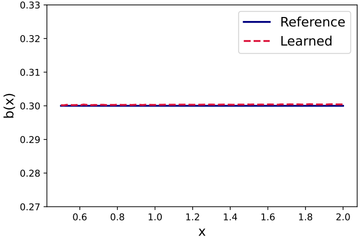
For discrete Itô SDEs, we have:

$$x_{n+1} = \mathrm{G}_\Delta(x_n) \approx x_n + a(x_n)\Delta + \sqrt{\Delta}\, b(x_n) Z. \tag{11}$$

To examine the accuracy of sFM learning, we propose the following effective drift and diffusion estimator:

$$\tilde{a}(x) = \frac{\mathbb{E}_\omega(\widetilde{\mathrm{D}}_\Delta(x) - x)}{\Delta}, \quad \tilde{b}(x) = \frac{\mathrm{Std}_\omega(\widetilde{\mathrm{D}}_\Delta(x))}{\sqrt{\Delta}} \tag{12}$$

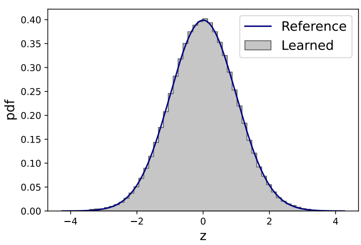


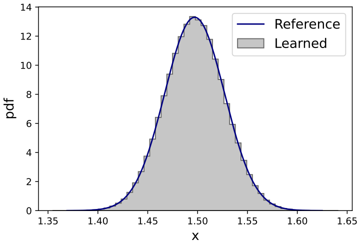


Ornstein–Uhlenbeck process

We can also compare the distribution of:

- The latent variable from the trained encoder $\widetilde{\mathrm{E}}_{\Delta}\left(x_0, x_1\right), x_0 = 1.5$, against standard normal $\mathcal{N}(0, 1)$
- One-step conditional distribution of the trained decoder $\widetilde{\mathrm{D}}_{\Delta}(\mathrm{x}_0, Z), Z \sim \mathcal{N}(0, 1)$ with the true one-step conditional distribution $P(\mathrm{x}_1|\mathrm{x}_0)$ at $\mathrm{x}_0 = 1.5$.



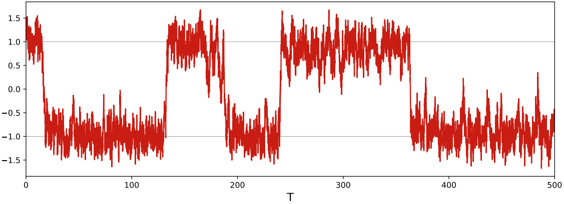


Double Well Potential

We consider the classical double well potential system, in the following form

$$dx_t = (x_t - x_t^3)dt + 0.5dW_t, \tag{13}$$

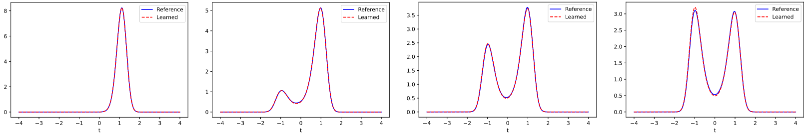
The trajectory may contain random switching between the two stable states x = ±1:



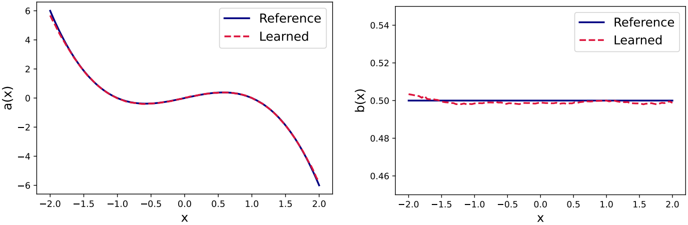
However, there is no transition in the training data (T = 1).

Double Well Potential

The evolution of solution PDF at time $T = 0.5$, 10.0, 30.0 and 100.0 given $x_0 = 1.5$:



Estimation of drift, diffusion functions:



Exponential Noise

We consider an SDE with exponentially distributed noise:

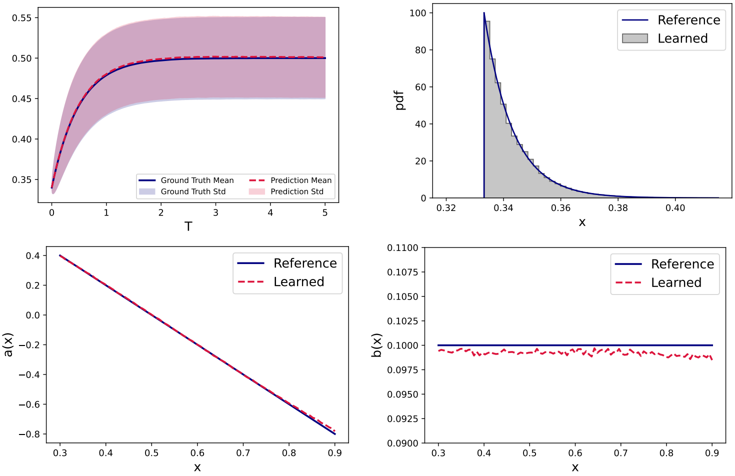
$$dx_t = \mu x_t dt + \sigma \sqrt{dt} \eta_t, \quad \eta_t \sim \mathrm{Exp}(1) \tag{14}$$

where η_t has an exponential pdf $f_\eta(x) = e^{-x}, x \geq 0$, and the constants are set as $\mu = -2.0$ and $\sigma = 0.1$.

Even with non-Gaussian noise, we still force the latent variable z to be standard normal and approximate an equivalent SDE:

$$dx_t = \mu x_t dt + \sigma \sqrt{dt} f(\mathrm{W}), \quad \mathrm{W} \sim \mathcal{N}(0,1) \tag{15}$$

where f transforms standard normal distribution to exponential distribution.


Ground Truth Mean
Prediction Mean
Ground Truth Std
Prediction Std
T
Reference
Learned
pdf
x
a(x)
b(x)

5D Ornstein–Uhlenbeck

We now consider a 5-dimensional OU process:

$$dx_t = Bx_t dt + \Sigma \, dW_t, \tag{16}$$

we choose the following 5 different cases for Σ, whose ranks vary from 1 to 5: $\mathrm{rank}(\Sigma_k) = k$, $k = 1, \ldots, 5$.
For each case, we progressively increase the values of n_z. During this process, we monitor the models' MSE losses

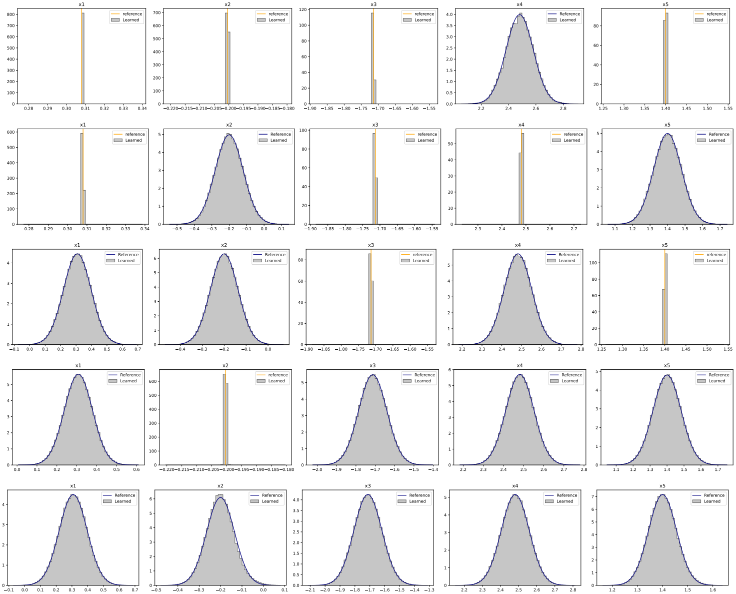
n_z	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5
1	3.5×10^{-7}	1.1×10^{-3}	1.8×10^{-3}	2.3×10^{-3}	3.3×10^{-3}
2	5.1×10^{-7}	6.4×10^{-7}	7.2×10^{-4}	1.4×10^{-3}	1.9×10^{-3}
3	3.3×10^{-7}	5.0×10^{-7}	7.1×10^{-7}	5.3×10^{-4}	8.8×10^{-4}
4	3.8×10^{-7}	5.4×10^{-7}	4.1×10^{-7}	7.2×10^{-7}	4.5×10^{-4}
5	7.2×10^{-7}	9.4×10^{-7}	6.3×10^{-7}	6.1×10^{-7}	6.2×10^{-7}

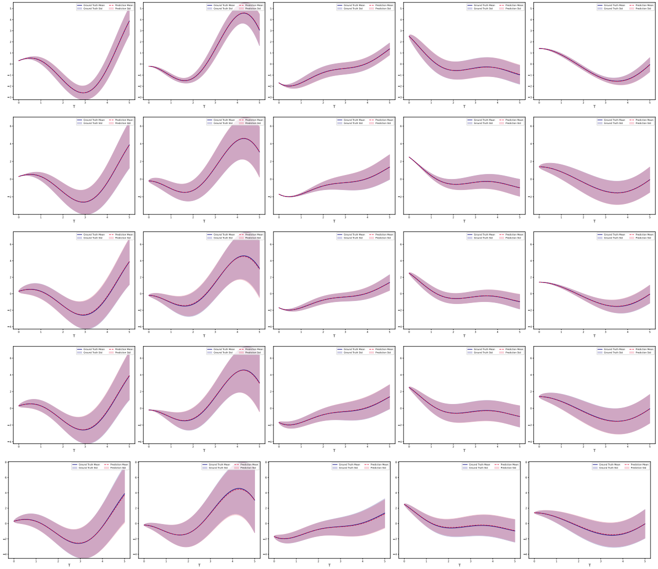
n_z	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5
1	**1.04E-02**	1.83E-02	6.27E-02	8.32E-02	7.60E-02
2	1.60E-02	1.40E-02	4.98E-02	**2.61E-02**	1.78E-01
3	1.74E-02	**1.26E-02**	3.65E-02	3.80E-02	2.43E-01
4	1.74E-02	3.57E-02	**3.40E-02**	3.45E-02	**4.55E-02**
5	6.67E-02	4.23E-02	4.99E-02	4.46E-02	4.86E-02

Table 1: The maximum relative l_2 error between the mean of model predictions and that of the reference solution.

n_z	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5
1	9.47E-03	5.85E-01	6.39E-01	5.04E-01	4.95E-01
2	**8.69E-03**	**1.83E-02**	3.52E-01	2.60E-01	4.10E-01
3	1.81E-02	1.88E-02	1.12E-02	9.48E-02	1.42E-01
4	2.00E-02	2.70E-02	**8.41E-03**	**9.30E-03**	1.31E-01
5	1.99E-01	3.21E-02	3.05E-02	2.89E-02	**9.48E-03**

Table 2: The maximum relative l_2 error between the STD of model predictions and that of the reference solution.


x1
x2
x3
x4
x5
reference
Reference
Learned


Ground Truth Mean
Prediction Mean
Ground Truth Std
Prediction Std
T

Paper and Code

Details and more examples can be found:

- Z. Xu, Y. Chen, Q. Chen, D. Xiu. *Modeling Unknown Stochastic Dynamical System via Autoencoder*, *Journal of Machine Learning for Modeling and Computing*.
- https://github.com/AtticusXu/Modeling-Unknown-Stochastic-Dynamical-System-via-Autoencoder.

Related Papers

- Yuan Chen, Dongbin Xiu. Learning Stochastic Dynamical System via Flow Map Operator, Journal of Computational Physics, 508 (2024), 112984.
- Yuan Chen, Dongbin Xiu. Data-driven Effective Modeling of Multiscale Stochastic Dynamical Systems, submitted.
- Yanfang Liu, Yuan Chen, Dongbin Xiu, Guannan Zhang. A Training-Free Conditional Diffusion Model for Learning Stochastic Dynamical Systems, submitted.