Optimal Transport Augmented Weak Collocation Regression Parameter Identification For Chaotic Dynamics Via Invariant Mesure

BY ZHIJUN ZENG

2023年9月13日

1 Dynamic System Setting

,

Noisy observations are denoted as

$$\mathbf{X}^* = (\mathbf{x}^*(t_0) + \eta_0, \mathbf{x}^*(t_1) + \eta_1, \cdots, \mathbf{x}^*(t_n) + \eta_n) \tag{1}$$

where $x^*$is the solution of the autonomous dynamical system of $\dot{x} = v(x, \theta^*)$, and $\{\eta_0, \eta_1, \ldots, \eta_n\}$are the measurement errors or uncertainties.

We suppress the time variable and consider the state-space distribution of data

$$\rho^* = \frac{\sum_{i=0}^{n} \delta_{x^*(t_i)}}{n+1} \tag{2}$$

The dynamic system admits a physical measure $\rho(\theta)$ if for a Lebesgue positive set of initial condition $x(0) = x$, one has that

$$\rho(\theta) = \lim_{T \to \infty} \frac{1}{T} \int_0^T \delta_{x(t)} \mathrm{dt} \tag{3}$$

Mathematically, statistical properties of can be characterized by the occupation measure $\rho_{x,T}$ defined as

$$\rho_{x,T}(B) = \frac{1}{T} \int_0^T \mathbb{1}_B(\mathbf{x}(s))\, d\, s = \frac{\int_0^T \mathbb{1}_B(\mathbf{x}(s))\, d\, s}{\int_0^T \mathbb{1}_{\mathbb{R}^d}(\mathbf{x}(s))\, d\, s} \tag{4}$$

where B is any Borel Measurable set. If there exist an invariant measure ρ such that aull $\rho_{x,T}$ weakly converges to ρ for all initial condition, then ρ is an physical measure .

By definition of physical measures μ^*, we have

$$\lim_{T\to\infty} \frac{1}{T} \int_0^T f(x(t))\, d\,t = \int_{\mathbb{R}^d} f(x)\, d\,\mu^*(x), \quad f \in C_c^\infty(\mathbb{R}^d) \tag{5}$$

By taking $f(x) = \nabla \phi(x) \cdot v(x)$ for some $\phi \in C_c^\infty(\mathbb{R}^d)$

$$\begin{aligned}\int_{\mathbb{R}^d} \nabla\, \phi(x) \cdot v(x)\, d\,\mu^*(x) = \lim_{T\to\infty} \frac{1}{T} \int_0^T \nabla\, \phi(x(t)) \cdot v(x(t))\, d\,t &= \\ \lim_{T\to\infty} \frac{1}{T} \int_0^T \nabla\, \phi(x(t)) \cdot \dot{x}(t)\, d\,t & \\ = \lim_{T\to\infty} \frac{1}{T} (\phi(x(T)) - \phi(x(0))) = 0 &\end{aligned} \tag{6}$$

This shows that μ^* is the stationary distributional solution

$$\nabla \cdot (v(x, \theta)\, \rho(x)) = 0. \tag{7}$$

2 Weak Collocation Regression

Consider the observation $\{x_i\}_{i=1}^N$ sampled from the physical measure $\rho(x,\theta)$, Weak Collocation method is an effective direct way to perform parameter identification utilizing the weak form of PDE and monte-carlo integration.

The observation data is aligned in matrix way

$$\boldsymbol{X} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^d & x_2^d & \dots & x_N^d \end{bmatrix} \in \mathbb{R}^{d\times N}$$

For a test function $\phi(x)$, by integration by part

$$\mathbb{E}[v(x,\theta)\cdot\nabla\phi(x)] = (v(x,\theta)\rho(x), \nabla\phi(x)) = 0$$

The velocity field is estimated by a large library representation $\{f_j(x) \in \mathbb{R}^d\}_{j=1}^L$, the representative matrix is

$$\Theta(x) = \begin{bmatrix} f_1^1(x) & \dots & f_L^1(x) \\ \vdots & \vdots & \vdots \\ f_1^d(x) & \dots & f_L^d(x) \end{bmatrix} \in \mathbb{M}^{d \times L}$$

Given an ensemble of test function $\{\phi_k\}_{k=1}^M$, the gradient matrix is given by

$$\boldsymbol{G}(x) = \begin{bmatrix} \phi_1^{x_1}(x) & \dots & \phi_M^{x_1}(x) \\ \vdots & \vdots & \vdots \\ \phi_1^{x_d}(x) & \dots & \phi_M^{x^d}(x) \end{bmatrix} \in \mathbb{M}^{d \times M}$$

The WCR minimize the following target:

$$\mathbb{E}_{\rho(x)}[\|w\Theta(x)^T\boldsymbol{G}(x)\|_2^2] + \mathrm{Reg}(w)$$

where $w \in \mathbb{R}^{1 \times L}$.

3 Regularization Method

3.1 Polynomial regularization

$$\mathrm{Reg}(w) = \| \sum_{j=1}^{L} w_j - C \|_2^2$$

3.2 Diffusion regularization

Consider the pertubation of stationary transport equation

$$\nabla \cdot (v(x,\theta)\rho(x)) + \sigma \Delta \rho(x,\theta) = 0$$

The weak form of stationary Fokker-planck equation is

$$(v(x,\theta)\rho(x), \nabla \phi(x)) = \sigma(\rho(x,\theta), \Delta \phi(x))$$

4 Optimal Transport Model Refinement

Obtaining the sparse identification of parameters in library, we use a forward based parameter identification method to refine the estimation. Indeed, we perform the following steps:

1. Sample batch data from noisy observation $\{x_i\}_{i=1}^{\text{batch}} \sim \mathbf{X}$ to construct emperical density estimation $\rho_{\text{batch},0} = \sum_{i=1}^{\text{batch}} \frac{\delta_{x_i}}{\text{batch}}$ for initial time
2. Compute $\rho_{\text{batch},T} = \text{ODESolve}(f(x,\theta), \rho_{\text{batch},0}, T)$ by odesolver like RK4
3. Compute Sliced Wasserstein Loss $\textsc{Loss} = \text{SWD}(\rho_{\text{batch},T}, \rho_{\text{batch},0})$ and obtain gradient
4. Normalized the source term $f(x,\theta)$

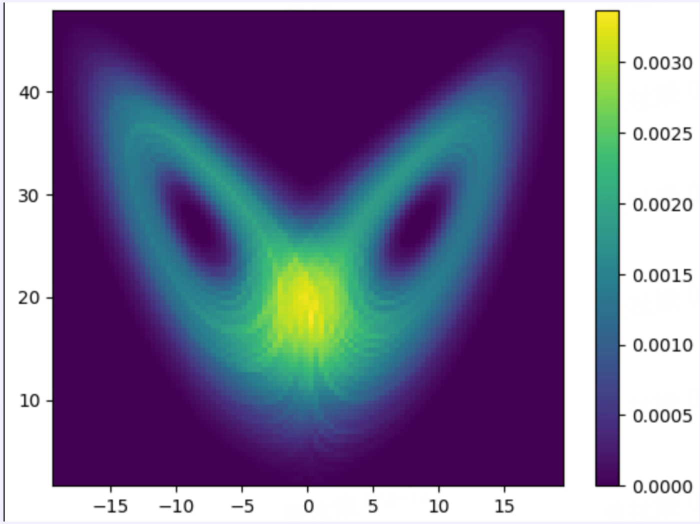
5 Experiment

5.1 ODE System

Lorentz

$$\begin{cases} \dot{x} &= \sigma(y-x) \\ \dot{y} &= x(\rho-z)-y \\ \dot{z} &= \mathrm{xy}-\beta z \end{cases},$$

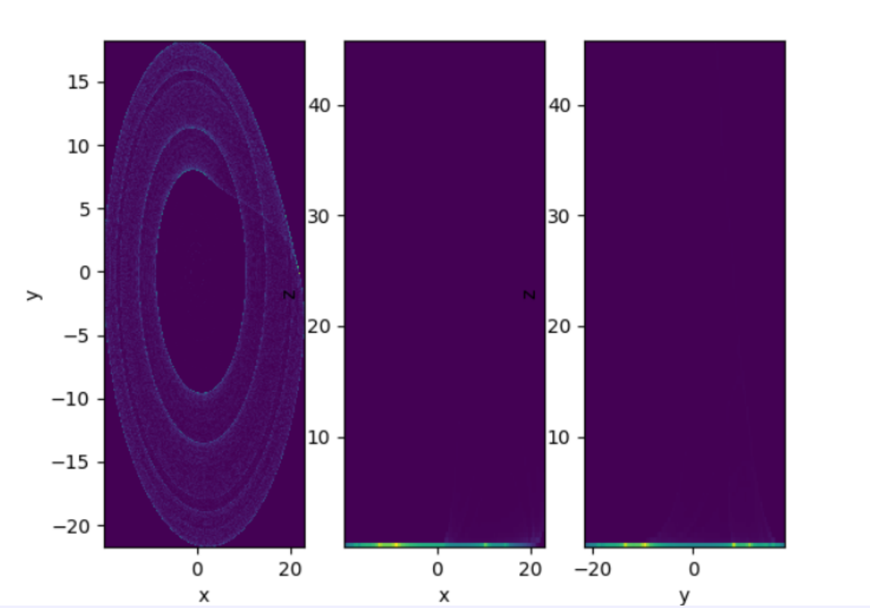
Here $\sigma=10, \rho=28, \sigma=8/3$, initial condition is [1,1,5],$T=20000, \mathrm{dt}=0.01$.the projection of density map wrt xy,xz,yz direction is below.



Rossler

$$\begin{cases} \dot{x} &= -y - z \\ \dot{y} &= x + \mathrm{ay} \\ \dot{z} &= b + z(x - c) \end{cases},$$

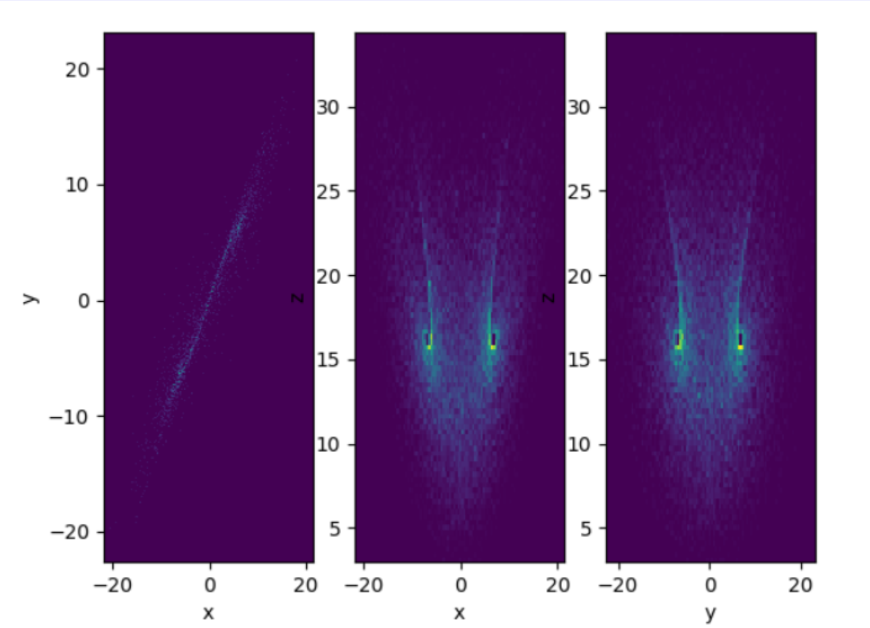
Here $a = 0.1, b = 0.1, c = 14$. initial condition is [2,2,5],$T = 20000, \mathrm{dt} = 0.01$..the projection of density map wrt xy,xz,yz direction is below.



Chen

$$\begin{cases} \dot{x} &= a(y-x) \\ \dot{y} &= (c-a)x - \mathrm{xz} + \mathrm{cy} \\ \dot{z} &= \mathrm{xy} - \mathrm{bz} \end{cases},$$

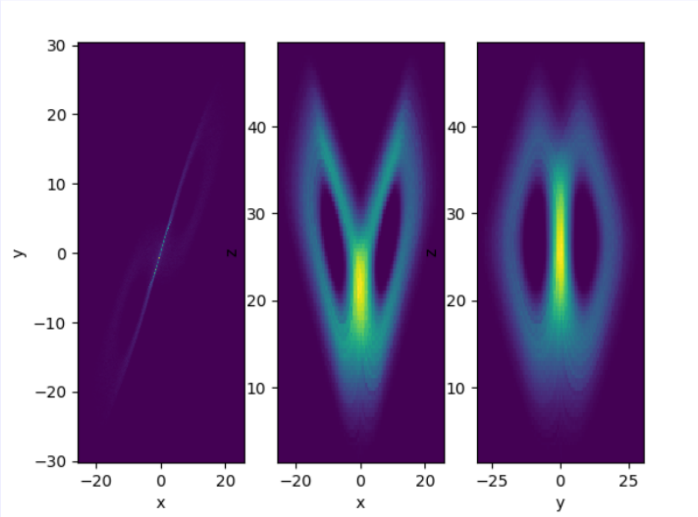
Here $a = 40, b = 3, c = 28$. initial condition is [2,2,5],$T = 20000, \mathrm{dt} = 0.01$.the projection of density map wrt xy,xz,yz direction is below.



Arctan Lorentz

$$\begin{cases} \dot{x} &= 50\arctan\left(\dfrac{\sigma(y-x)}{50}\right) \\ \dot{y} &= 50\arctan\left(\dfrac{x(\rho-z)-y}{50}\right) \\ \dot{z} &= 50\arctan\left(\dfrac{(\mathrm{xy}-\beta z)}{50}\right) \end{cases},$$

Here $\sigma = 10, \rho = 28, \sigma = 8/3$, initial condition is [1,1,5],$T = 20000, \mathrm{dt} = 0.01$.the projection of density map wrt xy,xz,yz direction is below.



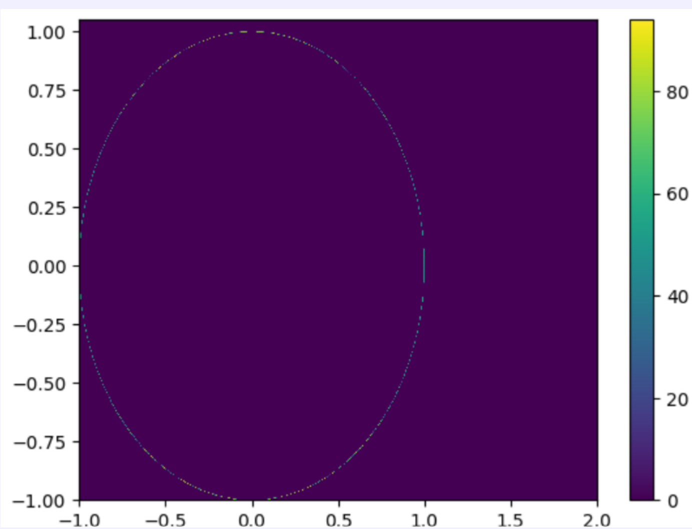
Periodic Limit Cycle

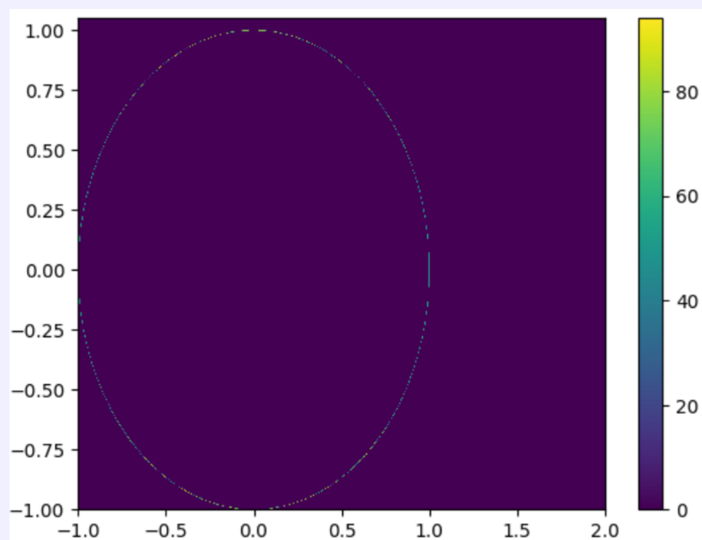
$$\begin{cases} \dot{x} &= x - y - (x^2 + y^2)x \\ \dot{y} &= x + y - (x^2 + y^2)y \end{cases},$$

This is a planar system which has a unique closed orbit γ and is a periodic attractor.

Note that the phase chart of such equation is almost the same as

$$\begin{cases} \dot{x} &= x - 0.3\,y - (x^2 + y^2)x \\ \dot{y} &= 0.3x + y - (x^2 + y^2)y \end{cases},$$





5.2 Parameter Identification by WCR

Lorenz:

表格 1. Lorenz

Gauss	Sigma	sample	Threshold	Result	Error
200	4	lhs	0.2	(10.058,28.029,2.666)	0.217%

Rossler:

表格 2. Rossler

Gauss	Sigma	sample	Threshold	Result	Error
200	4	lhs	0.01	(0.1039,0.0798,13.93))	0.66%

Chen:

表格 3. Chen

Gauss	Sigma	sample	Threshold	Result	Error
200	4	lhs	0.01	39.99,3.0,27.99	3.8e-5

Arctan lorenz:

表格 **4.** Arctan Lorenz

Gauss	Sigma	sample	Threshold	Result	Error
2000	1	lhs	0.2	(9.995,28.001,2.663)	2e-4

Limit cycle：

表格 **5.** Limit cycle

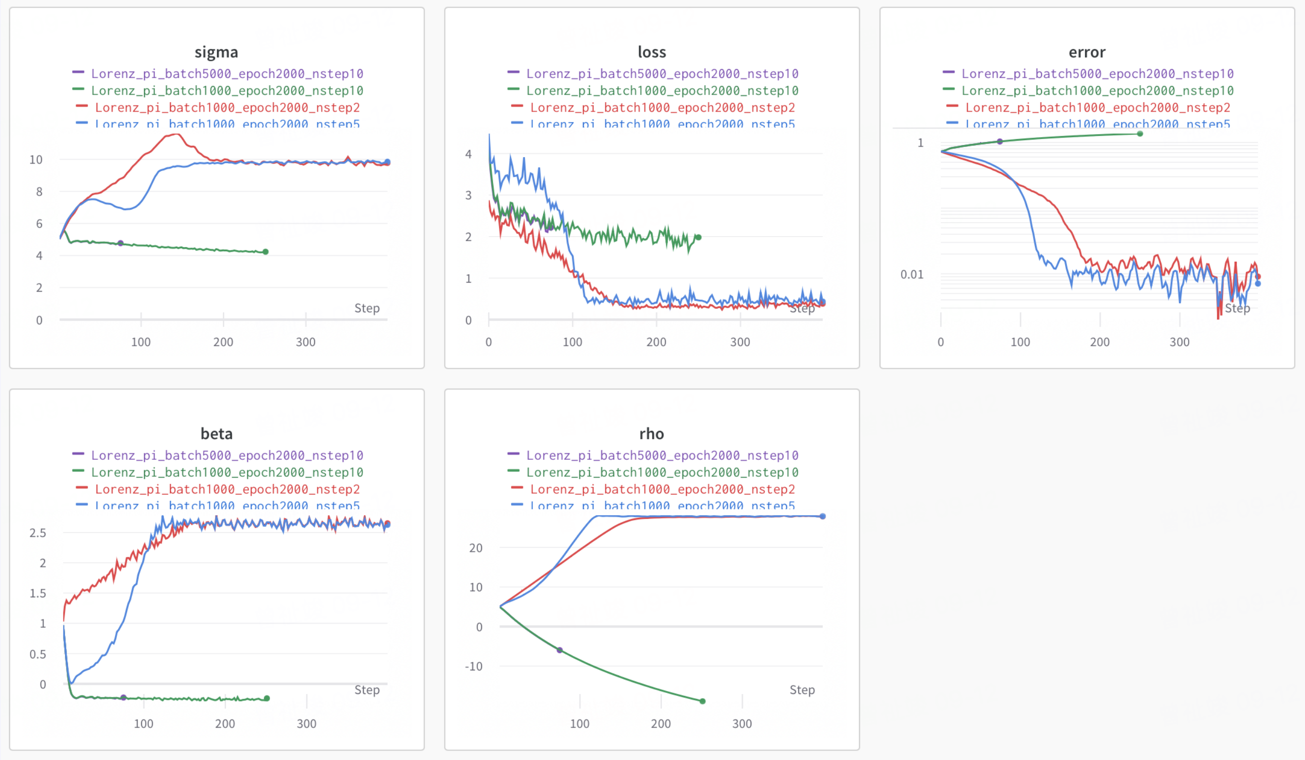
Gauss	Sigma	sample	Threshold	Result	Error
50	1	lhs	0.01	0.997,1.006,0.9972,1.0015	0.33%

5.3 Parameter Identification by OT

Lorenz:

表格 6. Lorenz

Batch	nstep	dt	lr	Result	Error
2000	5	0.02	3e-2	(9.848,27.90,2.628)	0.7%

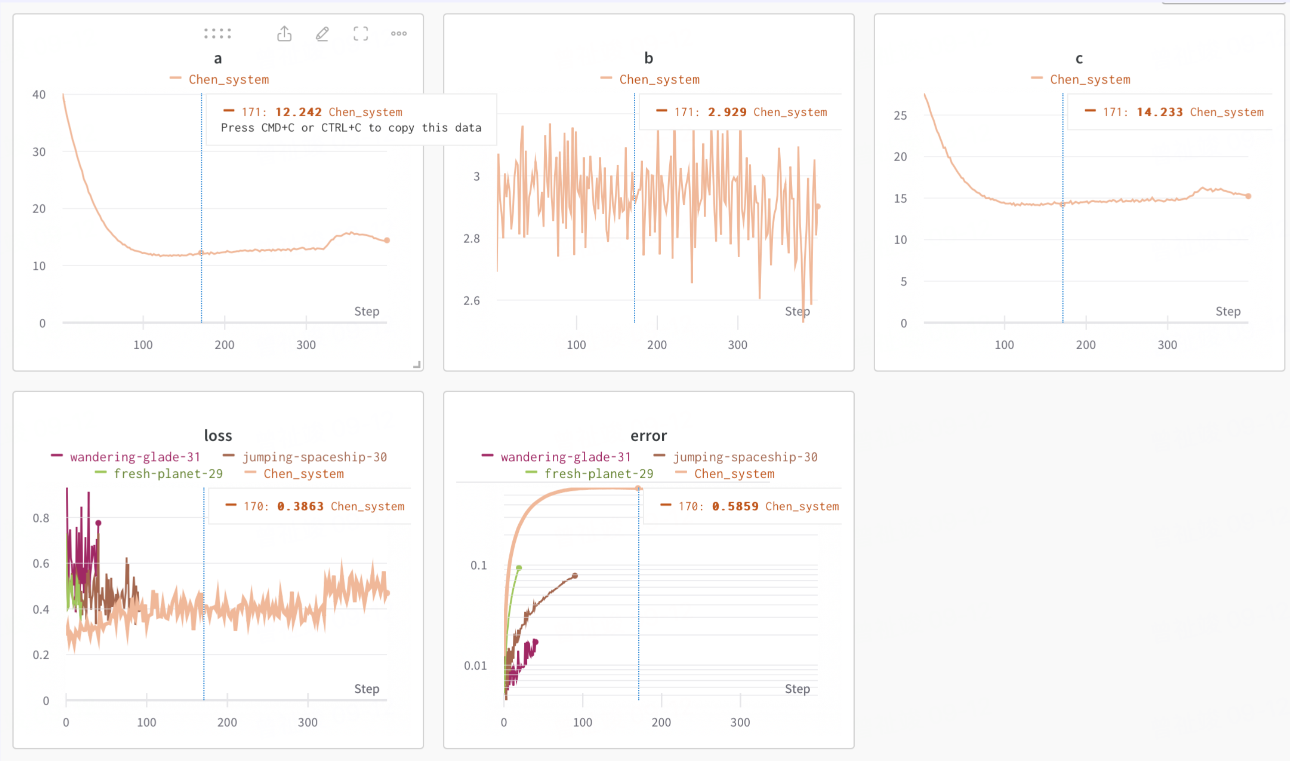


Rossler:

表格 7. Rossler

Batch	nstep	dt	lr	Result	Error
1000	5	0.02	3e-2	(0.122,-0.05,13.827)	3.6%

Chen:

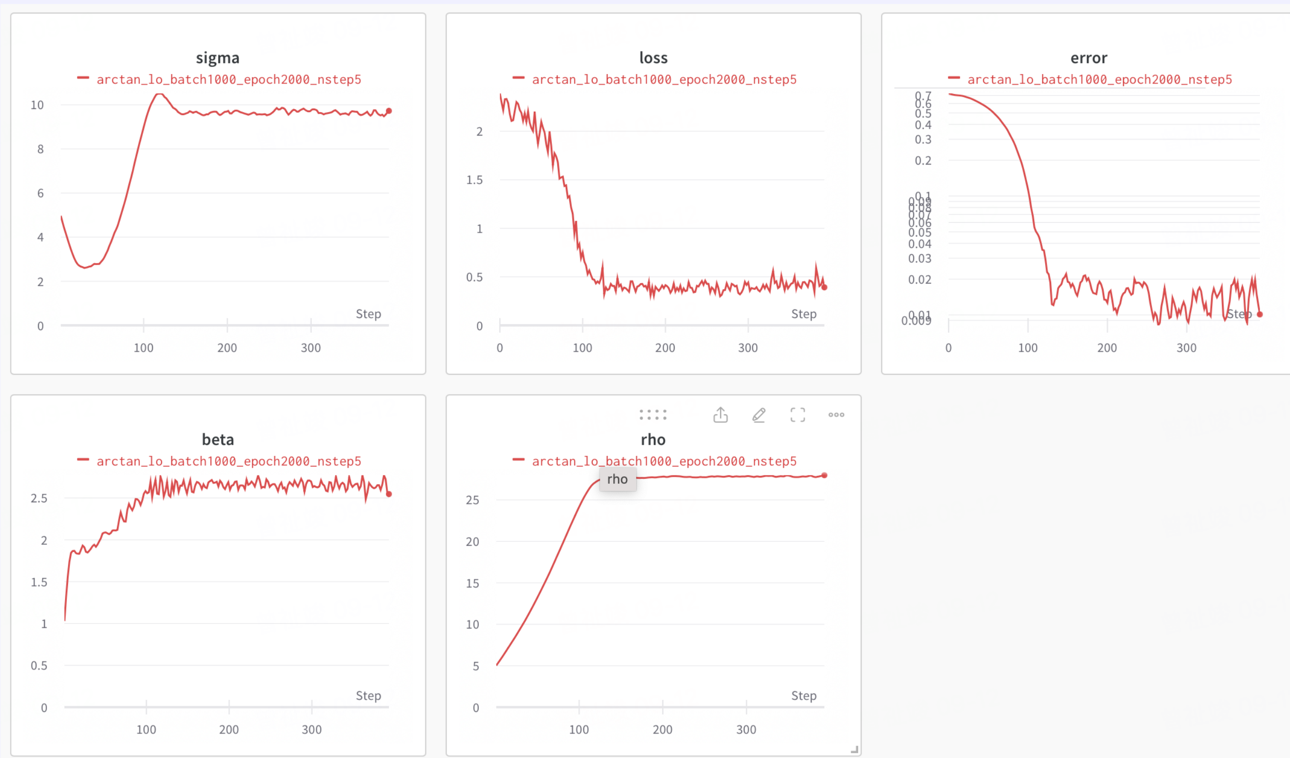


a
Chen_system
171: 12.242 Chen_system
Press CMD+C or CTRL+C to copy this data
Step
b
Chen_system
171: 2.929 Chen_system
Step
c
Chen_system
171: 14.233 Chen_system
Step
loss
wandering-glade-31
jumping-spaceship-30
fresh-planet-29
Chen_system
170: 0.3863 Chen_system
Step
error
wandering-glade-31
jumping-spaceship-30
fresh-planet-29
Chen_system
170: 0.5859 Chen_system
Step

Arctan lorentz:

表格 8. Arctan_lo

Batch	nstep	dt	lr	Result	Error
1000	5	0.02	3e-2	(9.726,27.98,2.54)	1.01%



Periodic:

表格 **9.** Periodic

Batch	nstep	dt	lr	Result	Error
1000	2	0.02	3e-2	(0.994,1.007,0.991,0.996)	0.6%

5.4 WCR diffusion regularization

Lorenz:

表格 10. Lorenz

Gauss	Sigma	sample	Threshold	epsilon	Error
300	4	lhs	0.2	0.1	4.2%

```
[2023-09-12 17:43:35,185][model][INFO] - param theta0tensor([[-0., -0., -0.]], device='cuda:0')
[2023-09-12 17:43:35,189][model][INFO] - param theta1tensor([[-10.5504,  28.8662,  -0.0000],
        [ 10.2398,  -1.4116,  -0.0000],
        [ -0.0000,  -0.0000,  -2.8063]], device='cuda:0')
[2023-09-12 17:43:35,191][model][INFO] - param theta2tensor([[-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000,  1.0226],
        [-0.0000, -1.0269, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000]], device='cuda:0')
[2023-09-12 17:43:35,194][model][INFO] - param theta3tensor([[-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.]], device='cuda:0')
```

Rossler:

```
[2023-09-12 17:46:11,659][model][INFO] - param theta0tensor([[-0.0000, -0.0000, 0.0839]], device='cuda:0')
[2023-09-12 17:46:11,660][model][INFO] - param theta1tensor([[ -0.0000,   1.0265,  -0.0000],
        [ -1.0247,   0.1089,  -0.0000],
        [ -0.9934,  -0.0000, -14.0044]], device='cuda:0')
[2023-09-12 17:46:11,663][model][INFO] - param theta2tensor([[-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, 1.0032],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000]], device='cuda:0')
[2023-09-12 17:46:11,666][model][INFO] - param theta3tensor([[-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.]], device='cuda:0')
```

表格 11. Rossler

Gauss	Sigma	sample	Threshold	epsilon	Error
300	4	lhs	0.05	0.1	1.04%

Chen:

```
[2023-09-12 17:50:55,639][model][INFO] - param theta0tensor([[-0., -0., -0.]], device='cuda:0')
[2023-09-12 17:50:55,643][model][INFO] - param theta1tensor([[-39.9967, -11.9988,  -0.0000],
        [ 39.9971,  27.9983,  -0.0000],
        [ -0.0000,  -0.0000,  -2.9999]], device='cuda:0')
[2023-09-12 17:50:55,644][model][INFO] - param theta2tensor([[-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000,  1.0000],
        [-0.0000, -0.9999, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000]], device='cuda:0')
[2023-09-12 17:50:55,645][model][INFO] - param theta3tensor([[-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.]], device='cuda:0')
```

表格 12. Chen

Gauss	Sigma	sample	Threshold	epsilon	Error
300	4	lhs	0.2	0.1	3.2%

5.5 WCR diffusion regularization

Lorenz:

表格 **13.** Lorenz

Gauss	Sigma	sample	Threshold	epsilon	Error
300	4	lhs	0.2	0.1	4.2%

```
[2023-09-12 17:43:35,185][model][INFO] - param theta0tensor([[-0., -0., -0.]], device='cuda:0')
[2023-09-12 17:43:35,189][model][INFO] - param theta1tensor([[-10.5504,  28.8662,  -0.0000],
        [ 10.2398,  -1.4116,  -0.0000],
        [ -0.0000,  -0.0000,  -2.8063]], device='cuda:0')
[2023-09-12 17:43:35,191][model][INFO] - param theta2tensor([[-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000,  1.0226],
        [-0.0000, -1.0269, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000]], device='cuda:0')
[2023-09-12 17:43:35,194][model][INFO] - param theta3tensor([[-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.],
        [-0., -0., -0.]], device='cuda:0')
```

Rossler:

表格 14. Rossler

Gauss	Sigma	sample	Threshold	epsilon	Error
300	6	lhs	0.04	2	6.95%

```
[2023-09-12 17:48:21,823][model][INFO] - param theta0tensor([[-0.0933, -0.0000,  0.1108]], device='cuda:0')
[2023-09-12 17:48:21,824][model][INFO] - param theta1tensor([[ -0.0000,   1.1160,  -0.0000],
        [ -1.1142,   0.1136,  -0.0000],
        [ -1.1049,   0.2458, -13.8090]], device='cuda:0')
[2023-09-12 17:48:21,826][model][INFO] - param theta2tensor([[-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0162,  1.0003],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.2664],
        [-0.0000, -0.0000, -0.0000]], device='cuda:0')
[2023-09-12 17:48:21,829][model][INFO] - param theta3tensor([[-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, 0.0174],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000],
        [-0.0000, -0.0000, -0.0000]], device='cuda:0')
[2023-09-12 17:48:21,833][model][INFO] - error 0.06954115629196167
```

Chen:

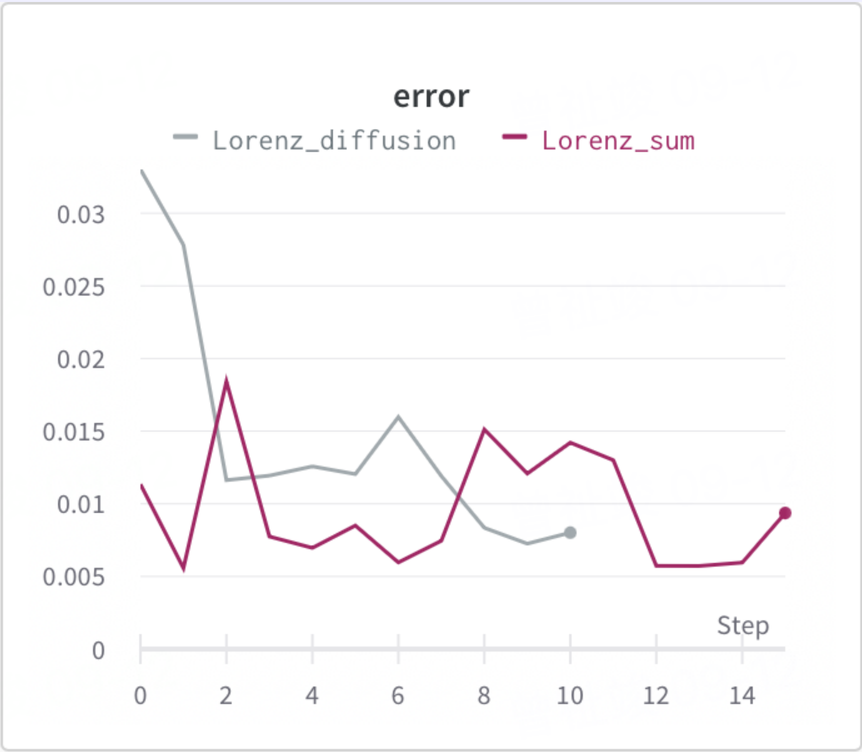
表格 15. Chen

Gauss	Sigma	sample	Threshold	epsilon	Error
300	5	lhs	0.5	2	6.83%

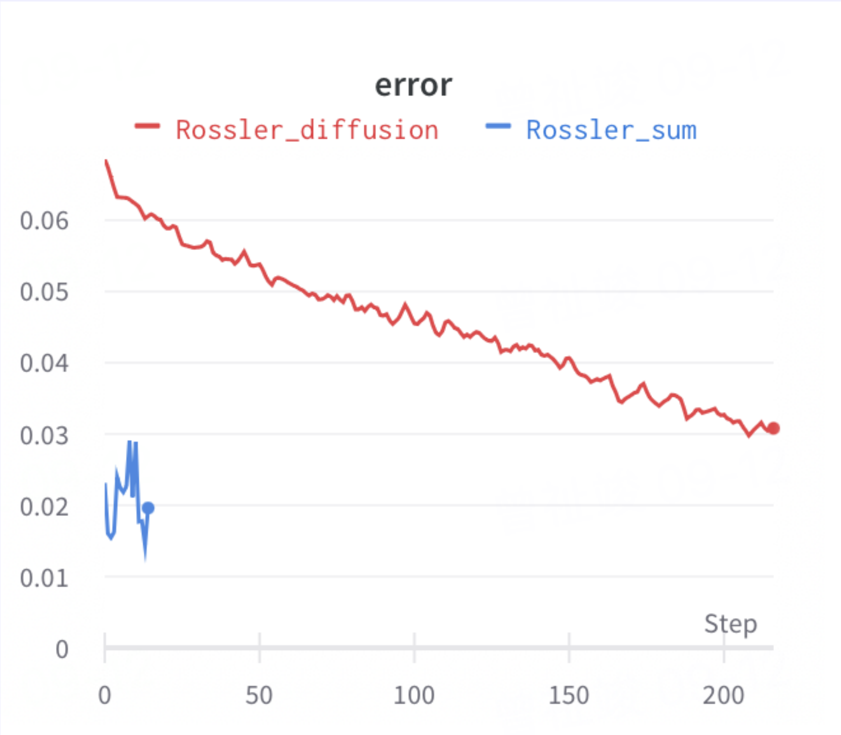
```
[2023-09-12 17:52:57,156][model][INFO] - param theta0tensor([[ 0.0000,  0.0000, -0.3609]], device='cuda:0')
[2023-09-12 17:52:57,158][model][INFO] - param theta1tensor([[-41.2060, -12.6115,   0.0000],
        [ 41.2131,  29.0277,   0.0000],
        [  0.0000,   0.0000,  -3.0660]], device='cuda:0')
[2023-09-12 17:52:57,159][model][INFO] - param theta2tensor([[ 0.0000,  0.0000,  0.0000],
        [ 0.0000,  0.0000,  1.0299],
        [ 0.0000, -1.0263,  0.0000],
        [ 0.0000,  0.0000,  0.0000],
        [ 0.0000,  0.0000,  0.0000],
        [ 0.0000,  0.0000,  0.0000]], device='cuda:0')
[2023-09-12 17:52:57,162][model][INFO] - param theta3tensor([[0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.],
        [0., 0., 0.]], device='cuda:0')
```

5.6 WCR+OT refinement

Lorenz



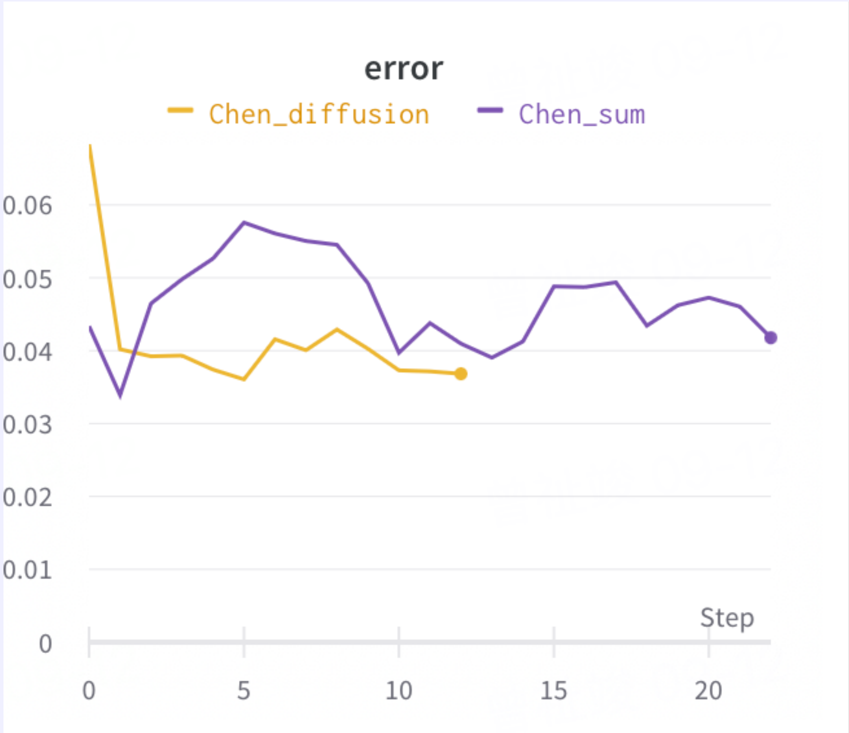
Rossler



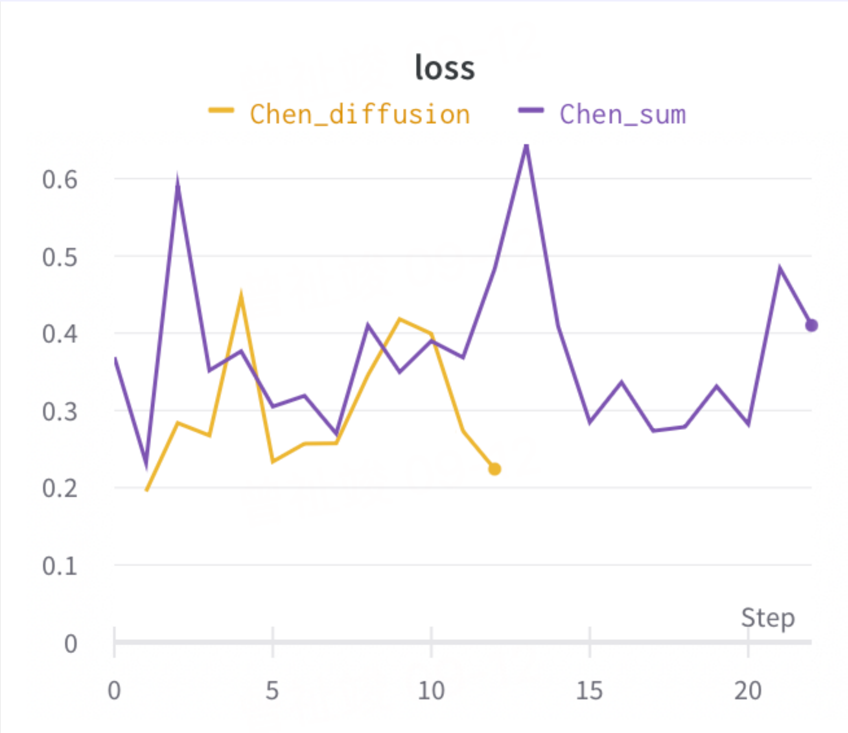
error
Rossler_diffusion
Rossler_sum
0.06
0.05
0.04
0.03
0.02
0.01
0
Step
0
50
100
150
200

loss
Rossler_diffusion
Rossler_sum
0.4
0.3
0.2
0.1
0
Step
0
50
100
150
200

Chen



error
Chen_diffusion
Chen_sum
0.06
0.05
0.04
0.03
0.02
0.01
0
Step
0
5
10
15
20



loss
Chen_diffusion
Chen_sum
0.6
0.5
0.4
0.3
0.2
0.1
0
Step
0
5
10
15
20