THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

APS March Meeting

AI in Science
A program of SCHMIDT FUTURES

Training Machine Learning Emulators

to Preserve Invariant Measures of Chaotic Attractors



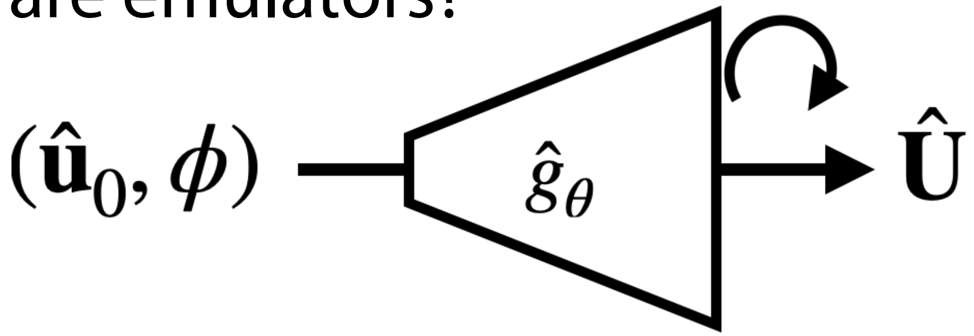

Ruoxi Jiang*, ***Peter Y. Lu****, Elena Orlova, Rebecca Willett

APS March Meeting 2024

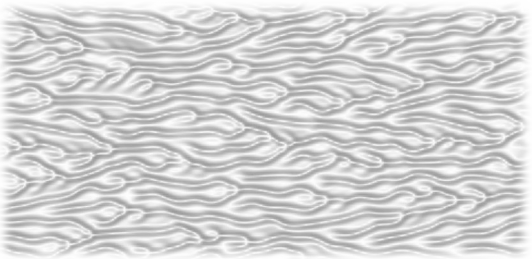
March 7, 2024

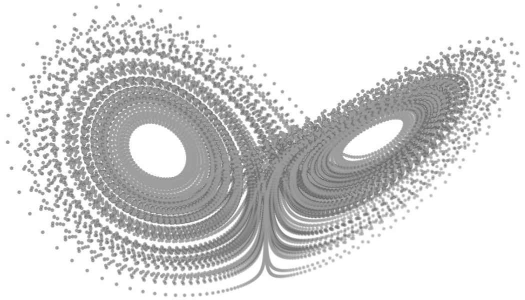
NeurIPS 2023

Introduction: What are emulators?

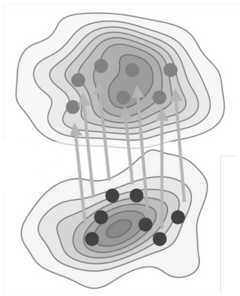


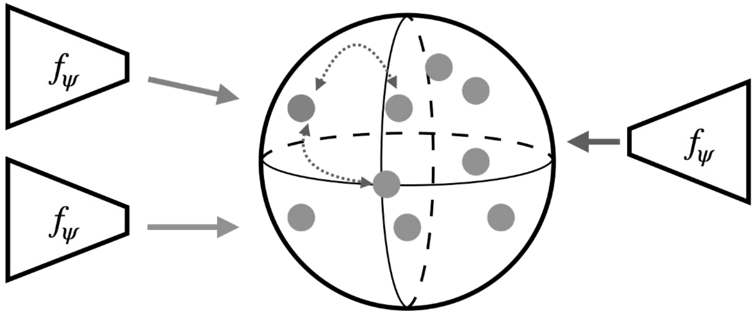
Problem: Why is it difficult to train emulators for chaos?





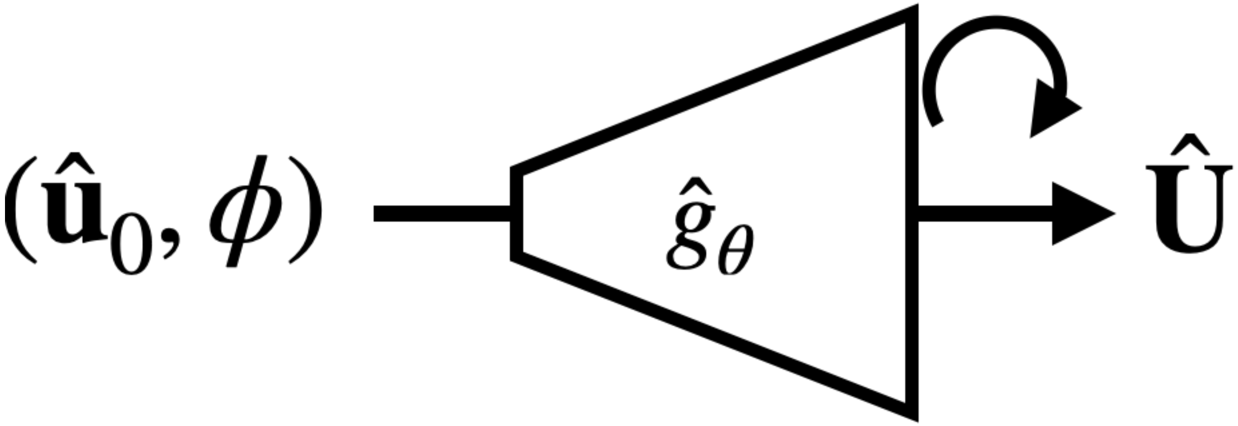
New Methods: How do we train emulators to capture chaos?





NeurIPS 2023

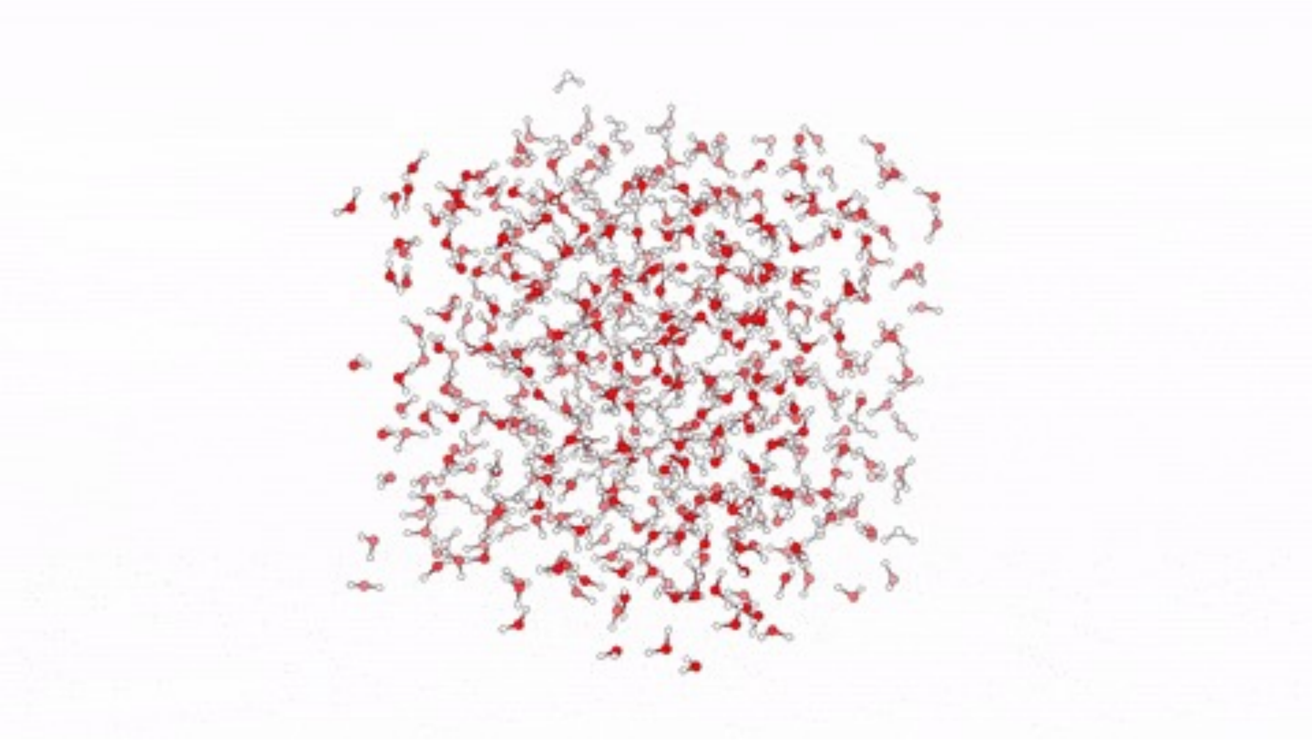
What are emulators?

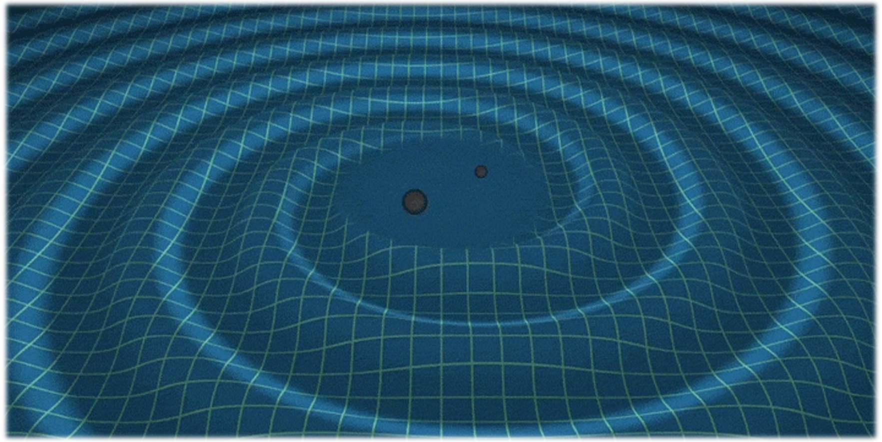


NeurIPS 2023

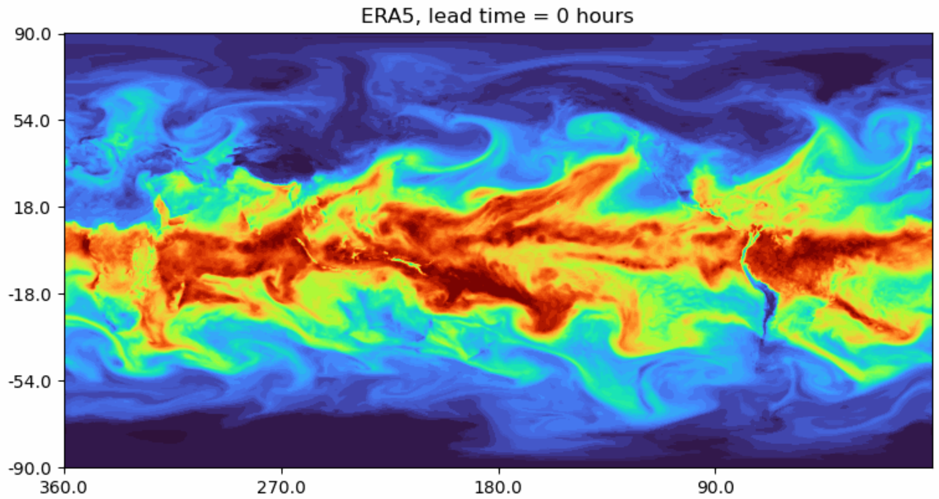
Dynamical Systems

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = G(\mathbf{u}, \phi)$$





LIGO/T. Pyle



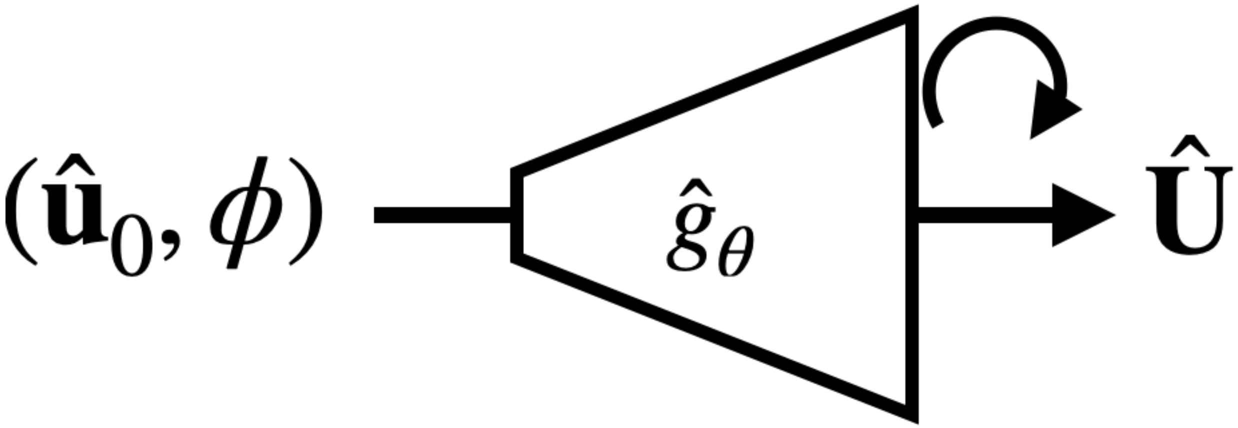
https://github.com/NVlabs/FourCastNet

NeurIPS 2023

Machine Learning Emulators

Emulators are machine learning models trained to **simulate physical systems**.

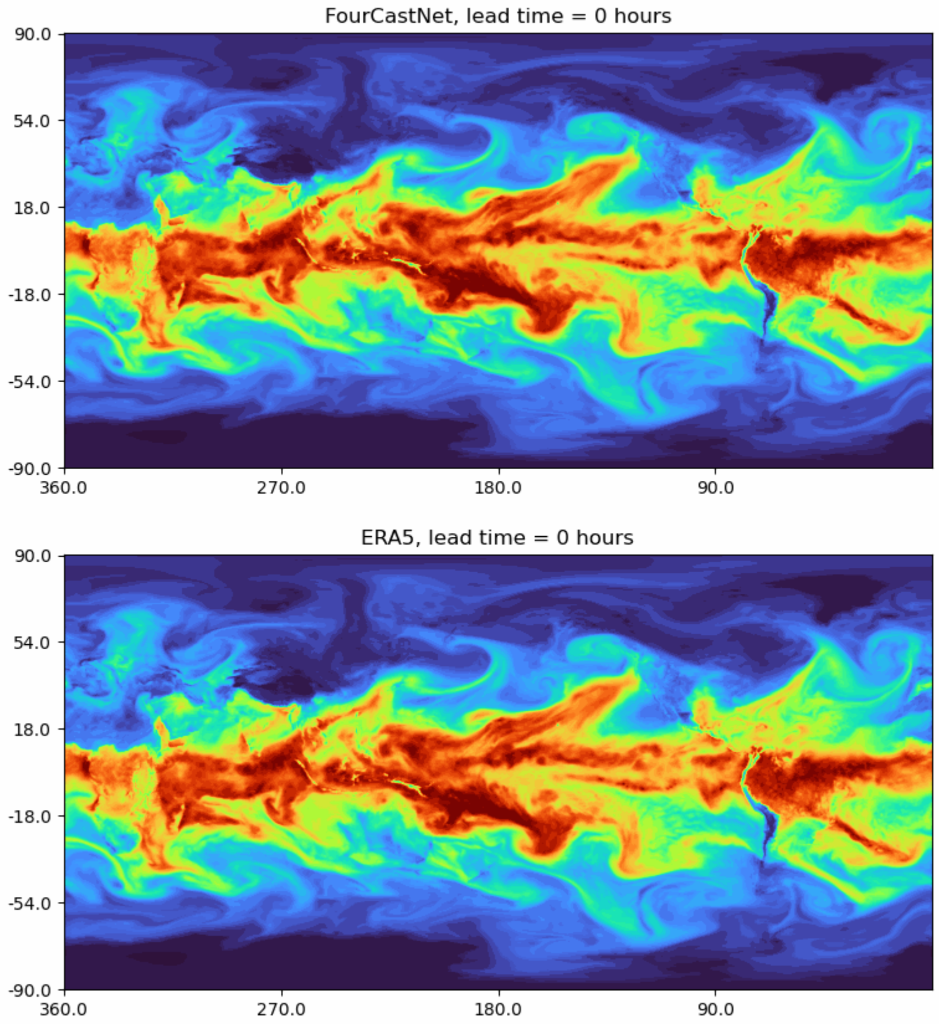
Emulators often consist of a **physics-informed architecture** trained on **observed or simulated data**.



Machine Learning Emulators

Emulators are machine learning models trained to **simulate physical systems**.

Emulators often consist of a **physics-informed architecture** trained on **observed or simulated data**.



Pathak et al. arXiv:2202.11214

NeurIPS 2023

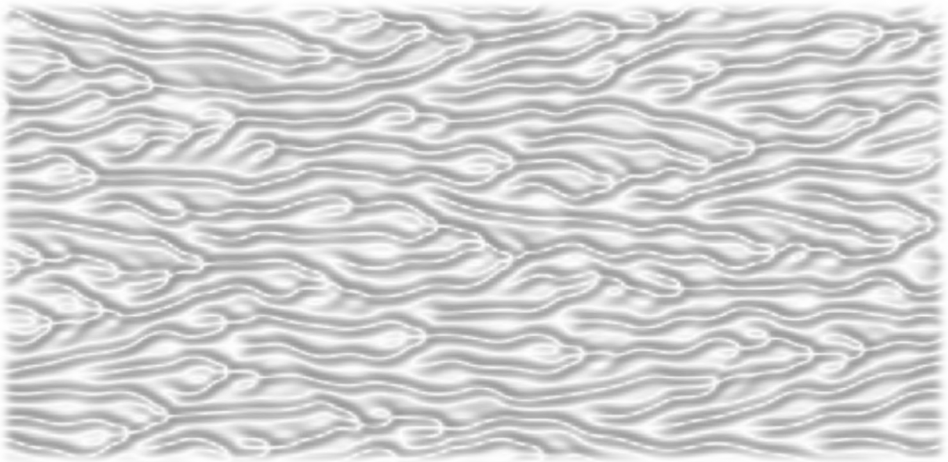
Machine Learning Emulators

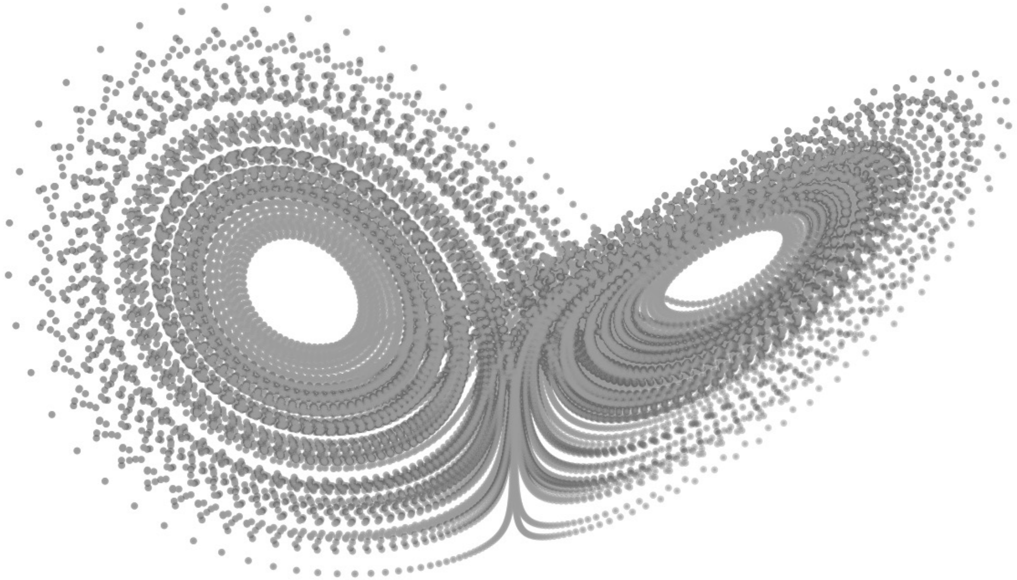
Accelerate expensive numerical simulations:
fast predictions, fast sampling for uncertainty quantification

Improve existing physical models using data:
more accurate predictions, better scientific understanding

Solve inverse problems (simulation-based inference):
parameter estimation, state reconstruction

Why is it difficult to train emulators for chaos?



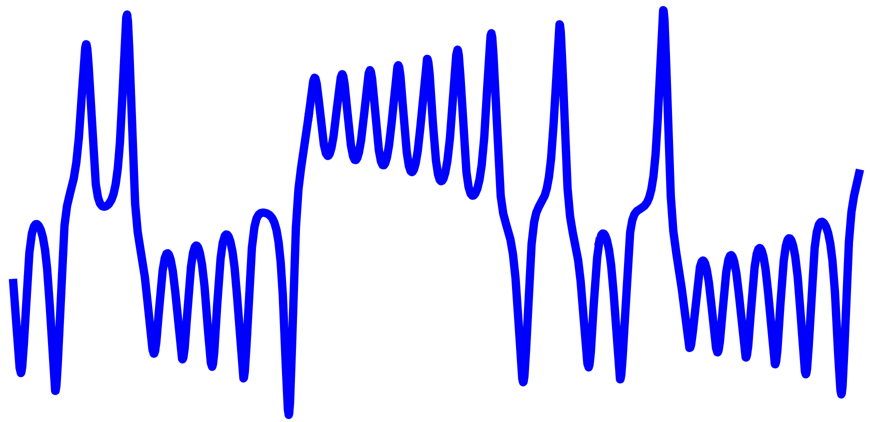


NeurIPS 2023

Chaotic Dynamics

A key feature of chaos is **sensitivity to initial conditions**.

Simple example: Lorenz-63

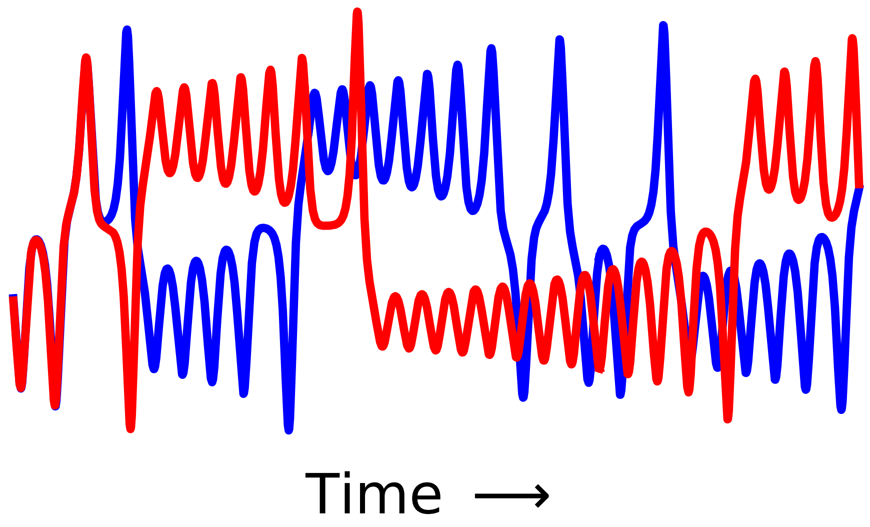


Time ⟶

Chaotic Dynamics

A key feature of chaos is **sensitivity to initial conditions**.

Simple example: Lorenz-63

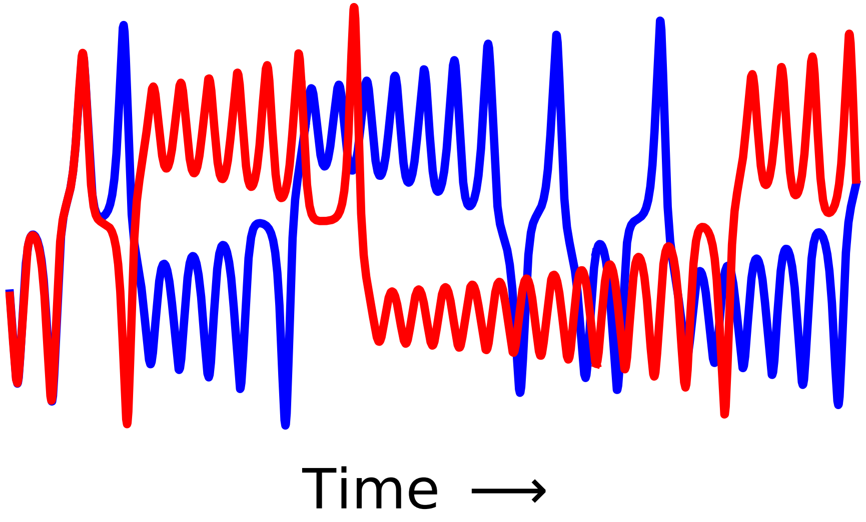


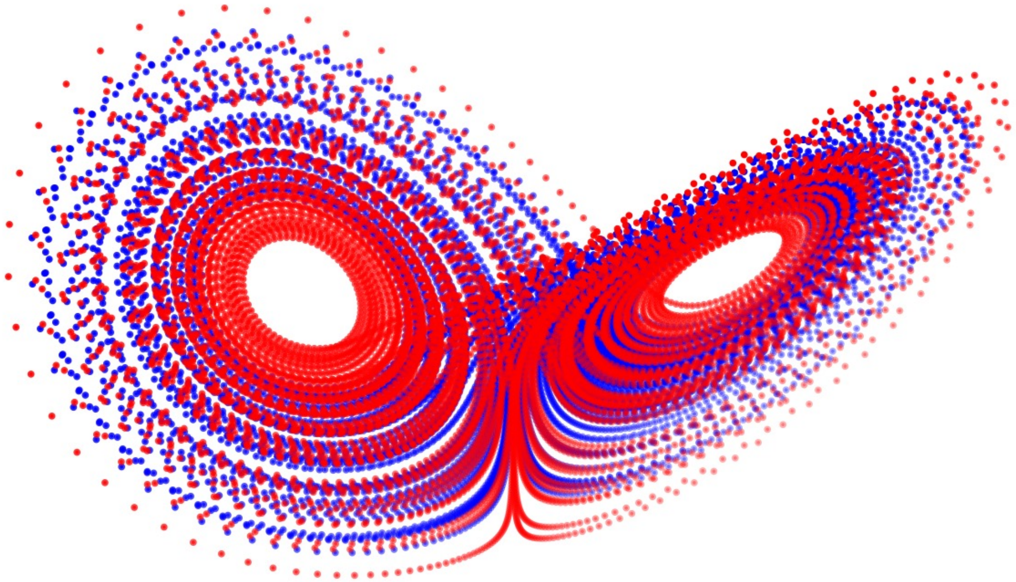
NeurIPS 2023

Chaotic Dynamics

A key feature of chaos is **sensitivity to initial conditions**.

Simple example: Lorenz-63





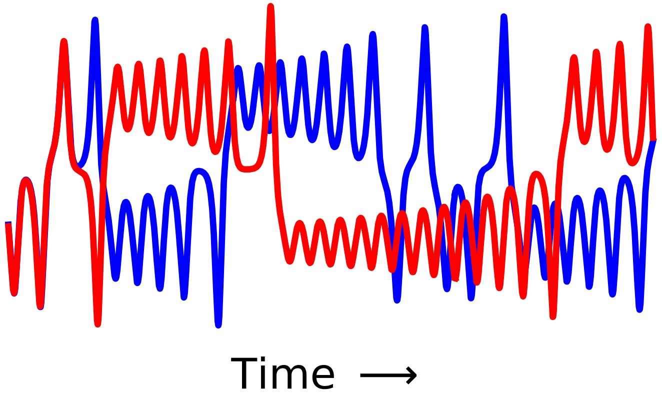
NeurIPS 2023

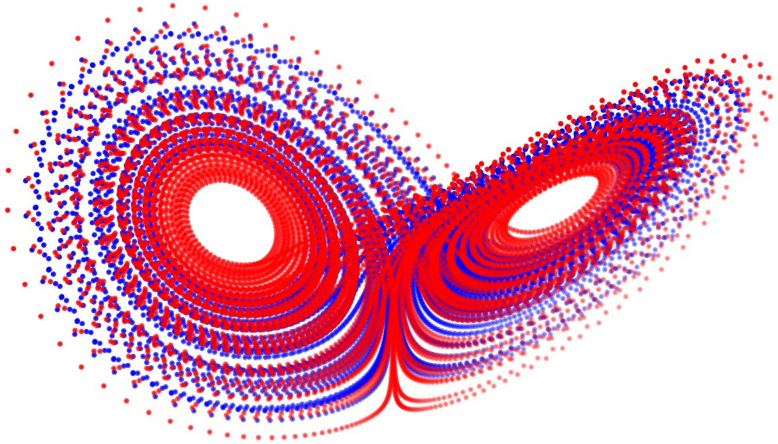
Chaotic Dynamics & Noise

Training emulators for chaotic dynamics is hard because **chaos is fundamentally unpredictable**.

Noise makes this worse due to **sensitivity to initial conditions**.

→ Instead of short-term forecasts, focus on **long-term statistics**.



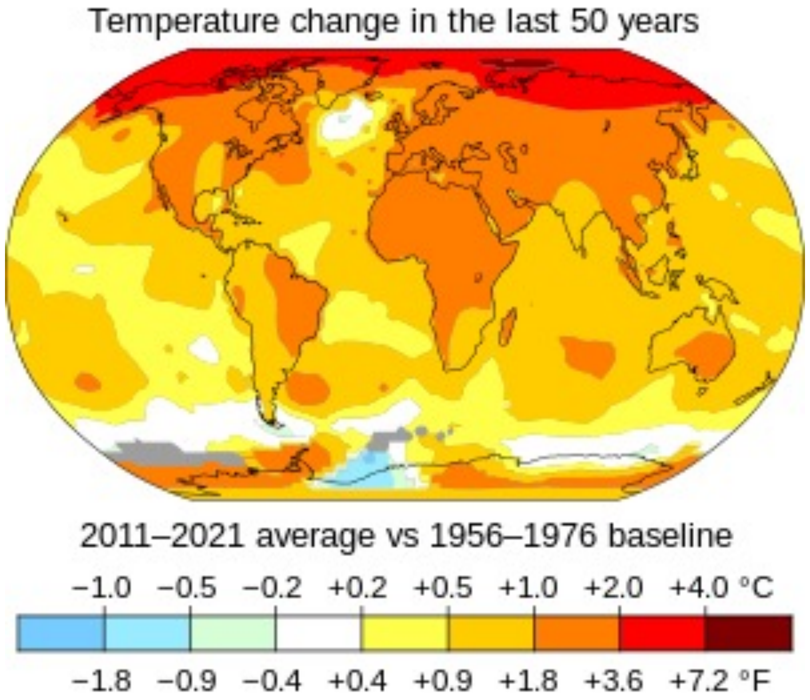


NeurIPS 2023

Modeling Chaos: Forecasting vs. Statistics

Example:
Weather vs. Climate

https://www.bbc.com/news/world-us-canada-47071683



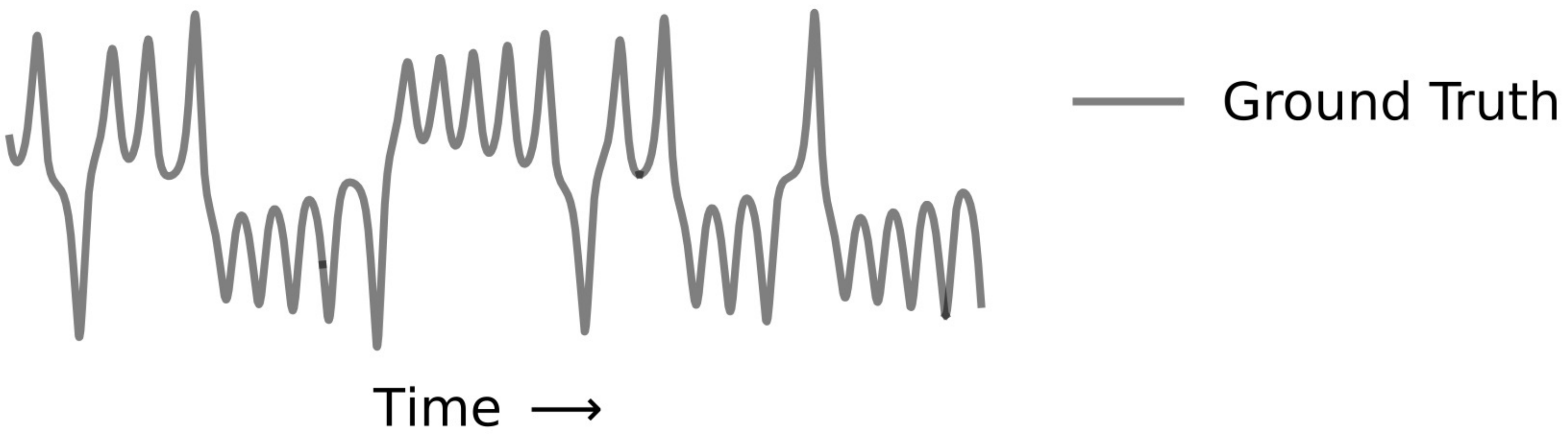
NeurIPS 2023

Modeling Chaos: Forecasting vs. Statistics

Accurate short-term forecasting

≠

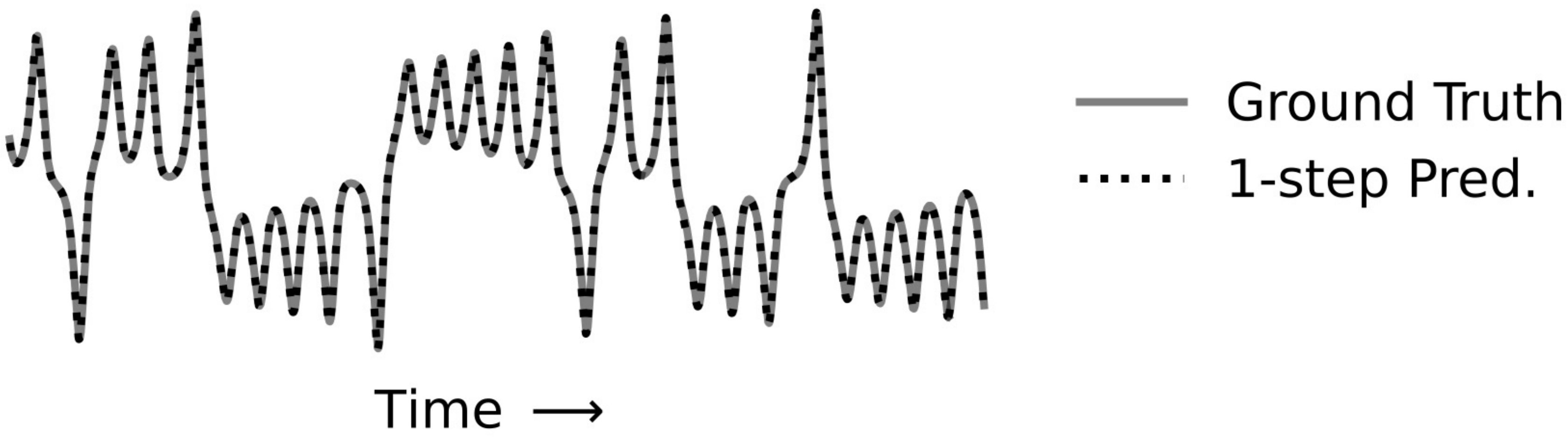
Correct long-term statistics!



NeurIPS 2023

Modeling Chaos: Forecasting vs. Statistics

Accurate short-term forecasting
≠
Correct long-term statistics!



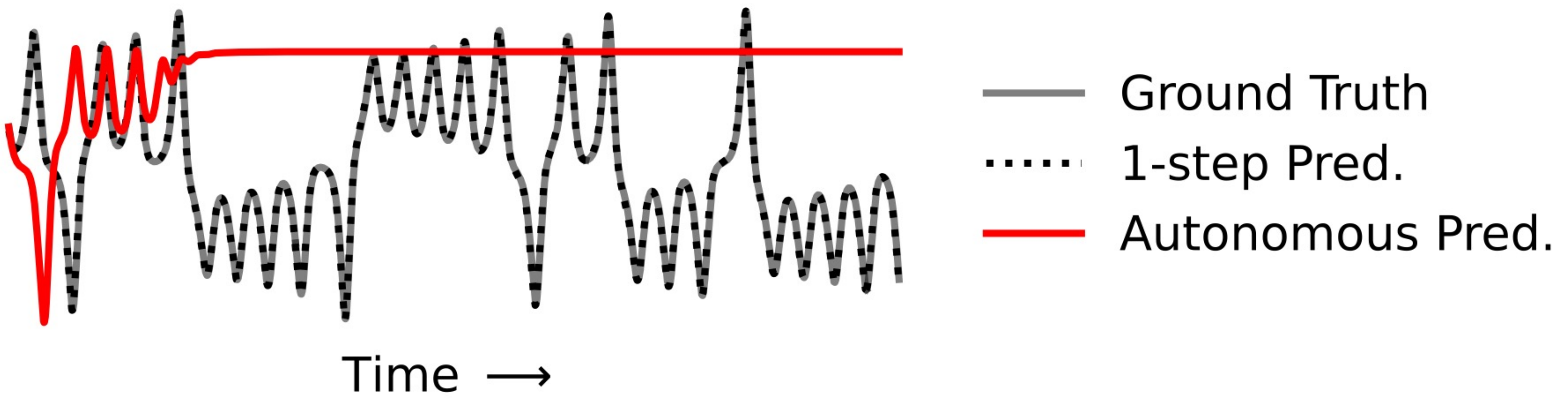
NeurIPS 2023

Modeling Chaos: Forecasting vs. Statistics

Accurate short-term forecasting

≠

Correct long-term statistics!



Modeling Chaos: Forecasting vs. Statistics

Accurate short-term forecasting

≠

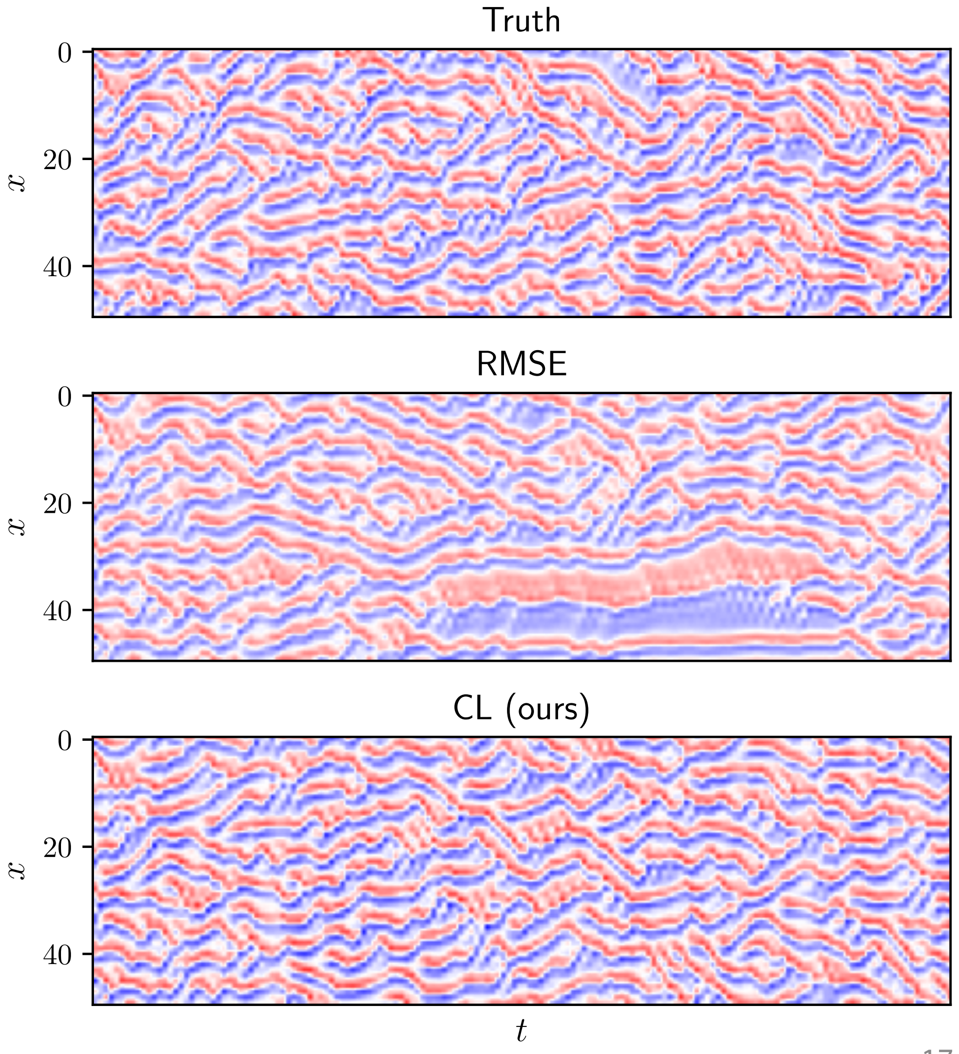
Correct long-term statistics!

Preview of results on noisy data:

Training on short-term *RMSE*

vs.

Training using *invariant statistics*

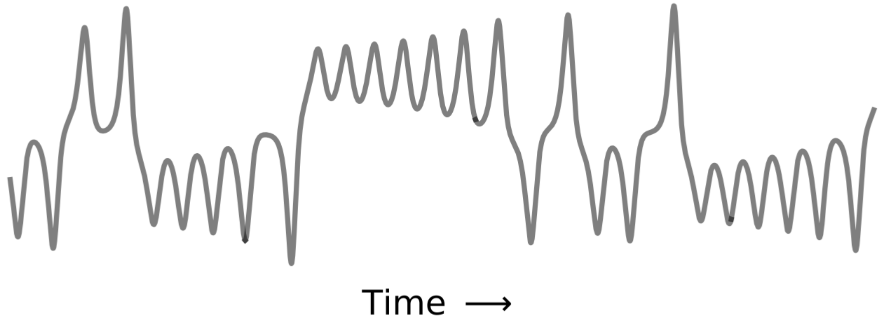


NeurIPS 2023

Effect of Noise on Error Metrics

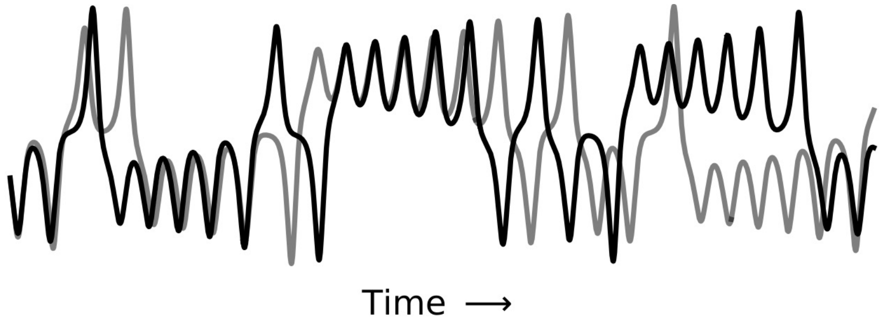
Original trajectory

$$U_G(u_0)$$



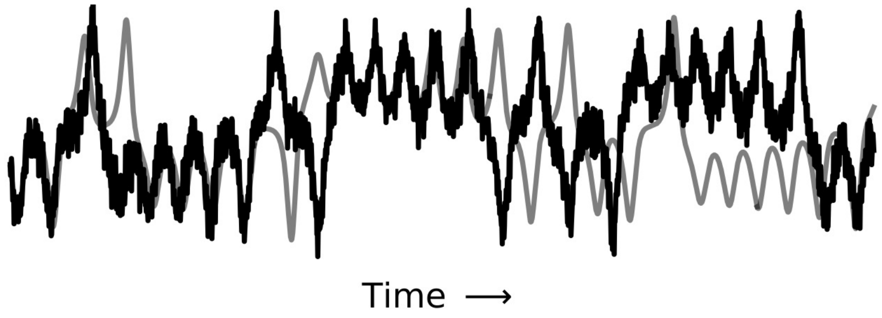
Noisy initial conditions

$$U_G(u_0 + \eta)$$



Noisy IC & measurements

$$U_G(u_0 + \eta) + \eta$$



NeurIPS 2023

Effect of Noise on Error Metrics

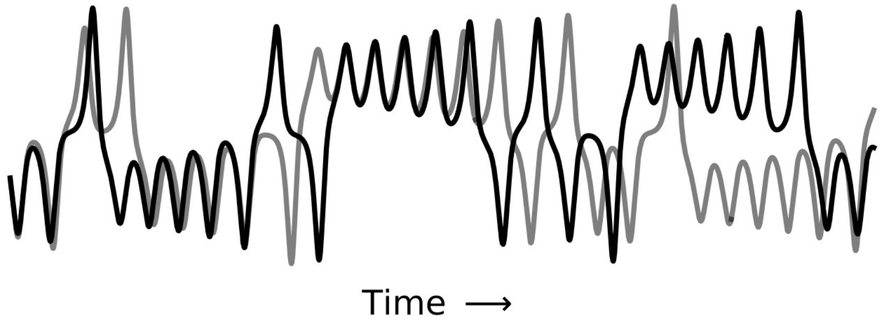
Original trajectory

$$U_G(u_0)$$

How does increasing noise η affect error metrics?

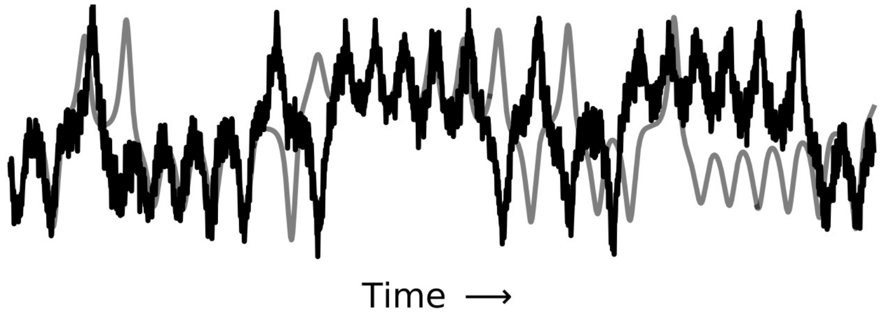
Noisy initial conditions

$$U_G(u_0 + \eta)$$



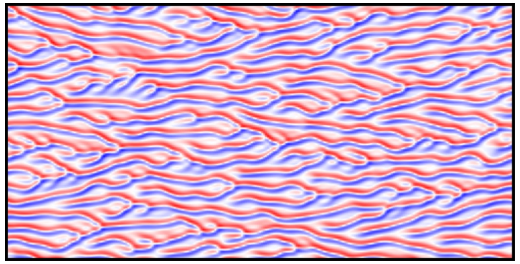
Noisy IC & measurements

$$U_G(u_0 + \eta) + \eta$$



NeurIPS 2023

Effect of Noise on Error Metrics



Original trajectory

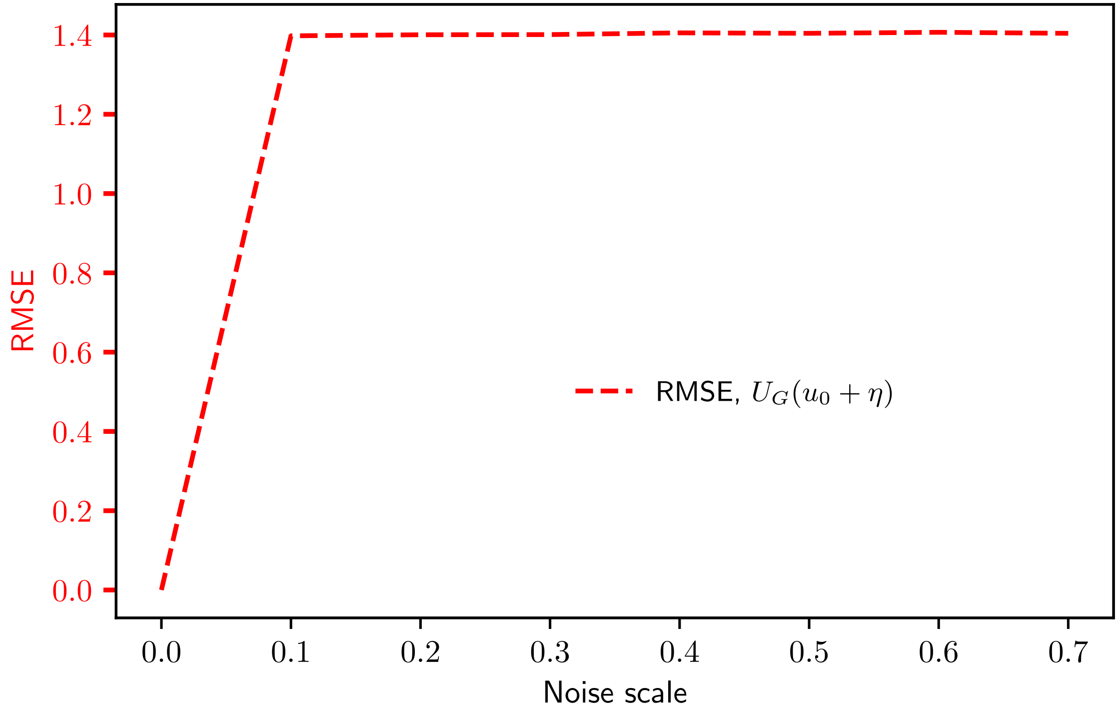
$$U_G(u_0)$$

Noisy initial conditions

$$U_G(u_0 + \eta)$$

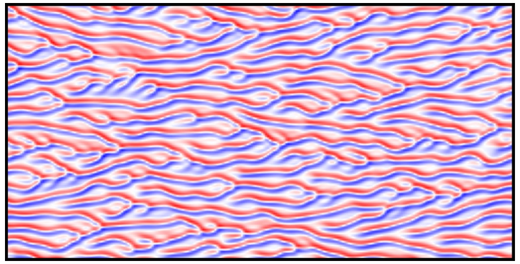
Noisy IC & measurements

$$U_G(u_0 + \eta) + \eta$$



NeurIPS 2023

Effect of Noise on Error Metrics



Original trajectory

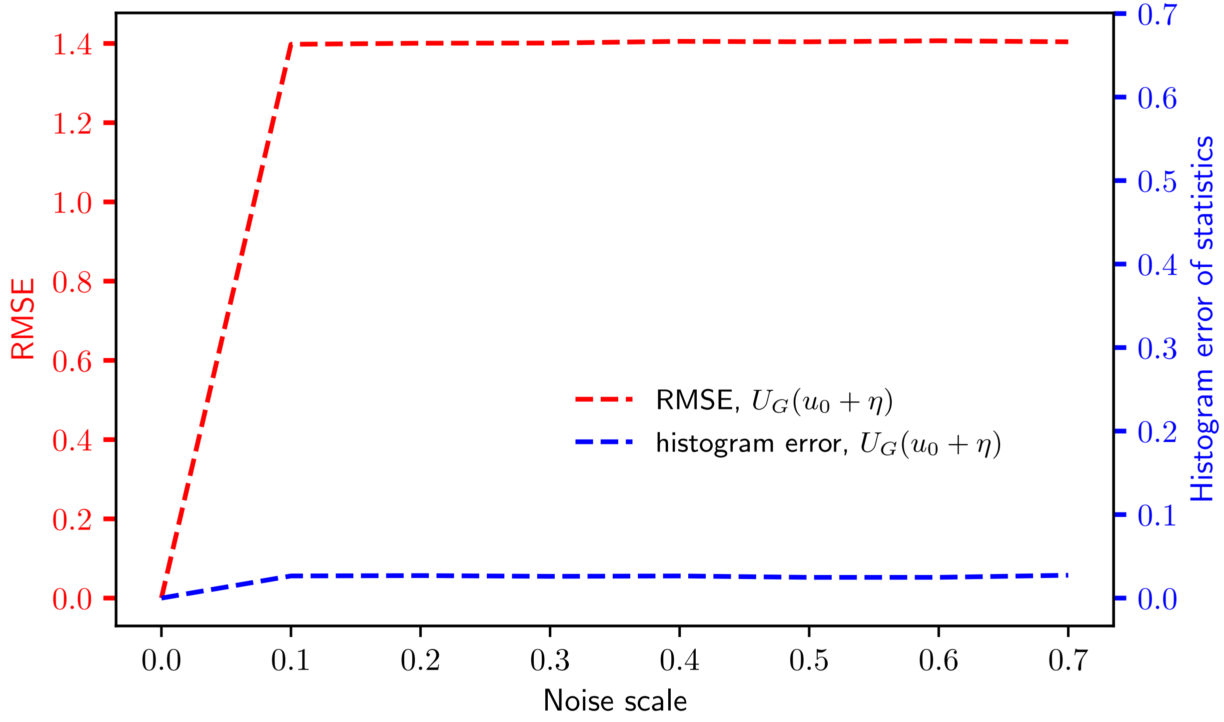
$$U_G(u_0)$$

Noisy initial conditions

$$U_G(u_0 + \eta)$$

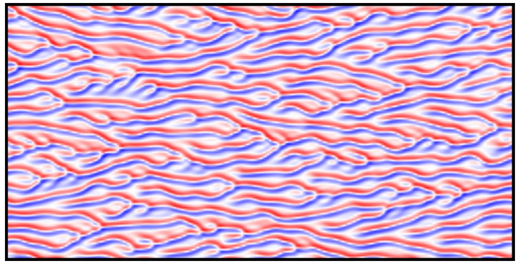
Noisy IC & measurements

$$U_G(u_0 + \eta) + \eta$$



NeurIPS 2023

Effect of Noise on Error Metrics



Original trajectory

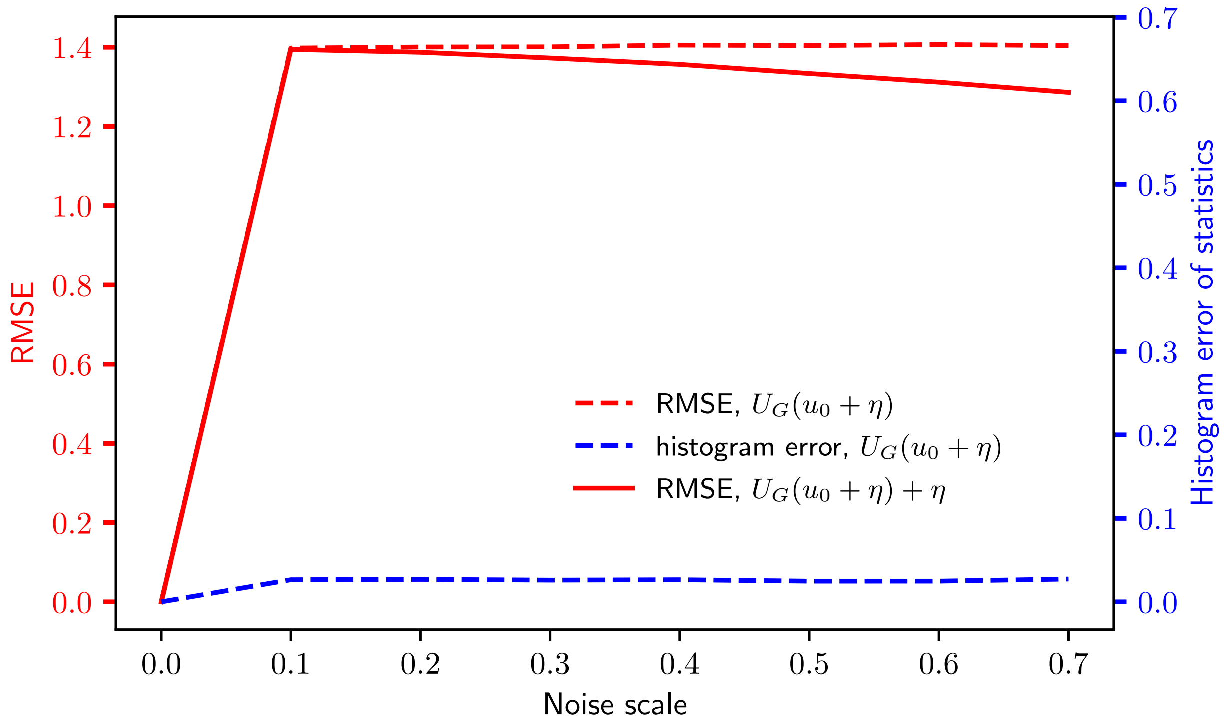
$$U_G(u_0)$$

Noisy initial conditions

$$U_G(u_0 + \eta)$$

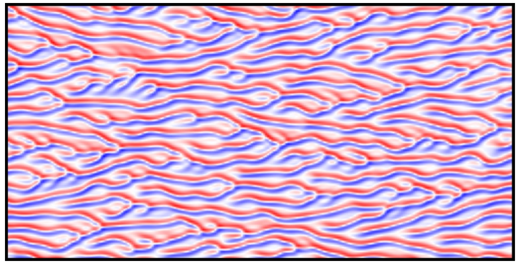
Noisy IC & measurements

$$U_G(u_0 + \eta) + \eta$$



NeurIPS 2023

Effect of Noise on Error Metrics



Original trajectory

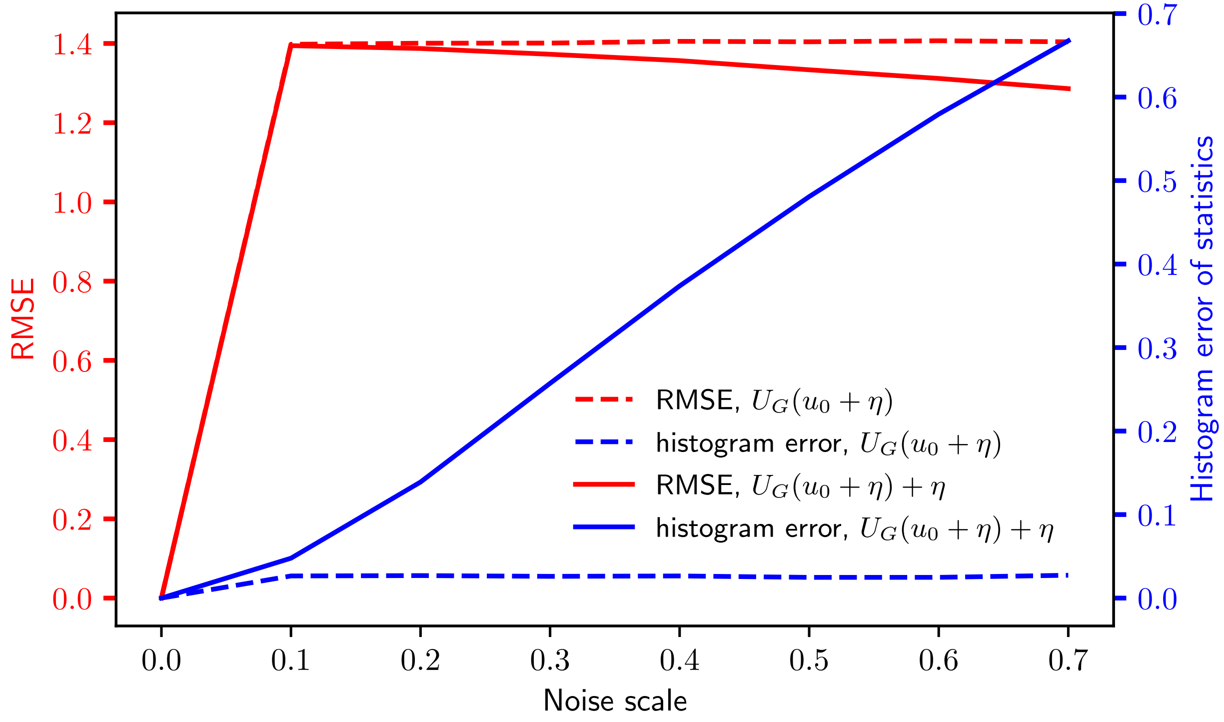
$$U_G(u_0)$$

Noisy initial conditions

$$U_G(u_0 + \eta)$$

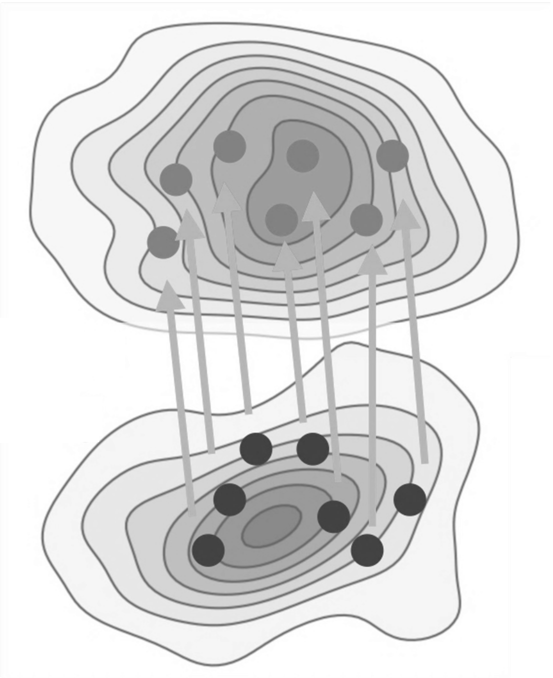
Noisy IC & measurements

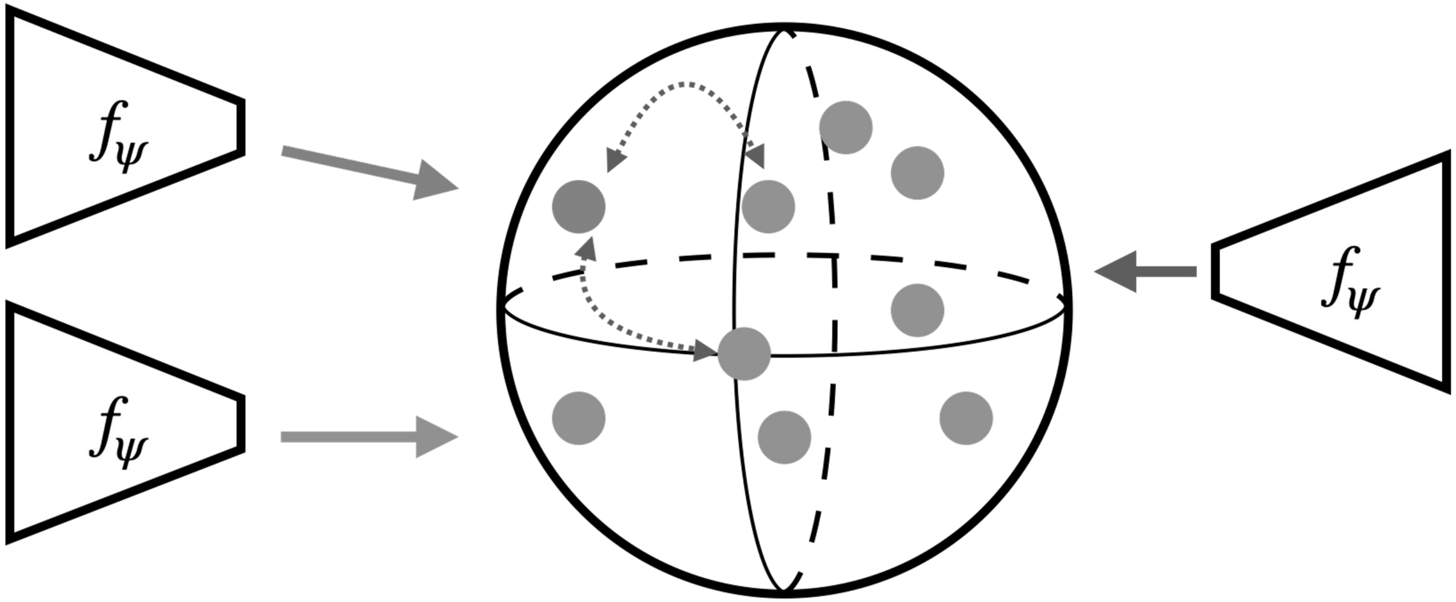
$$U_G(u_0 + \eta) + \eta$$



NeurIPS 2023

How do we train emulators to capture chaos?



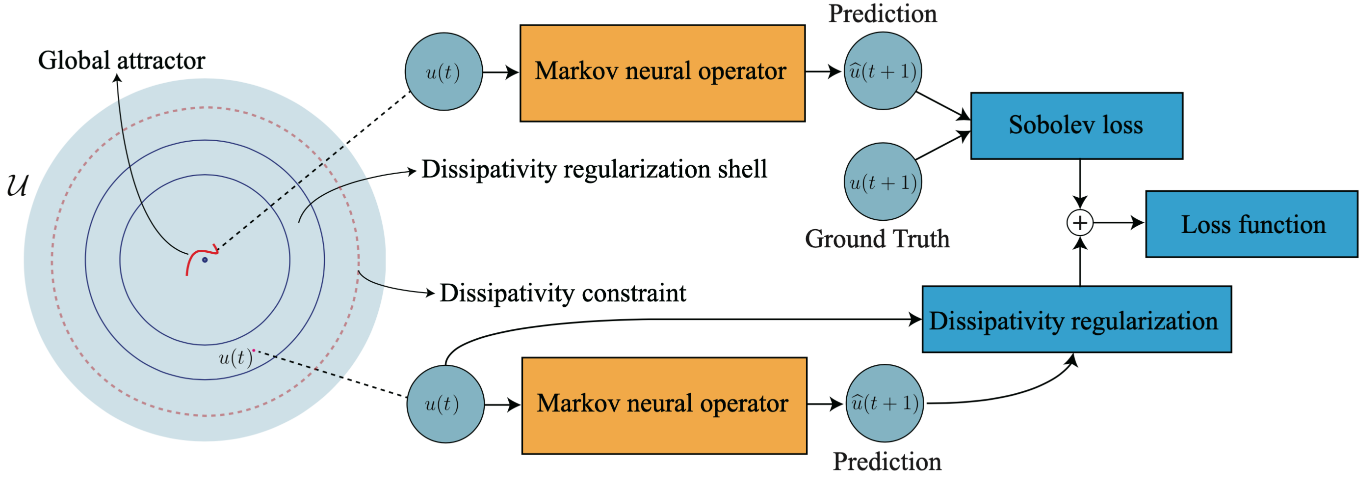


NeurIPS 2023

Recent Work

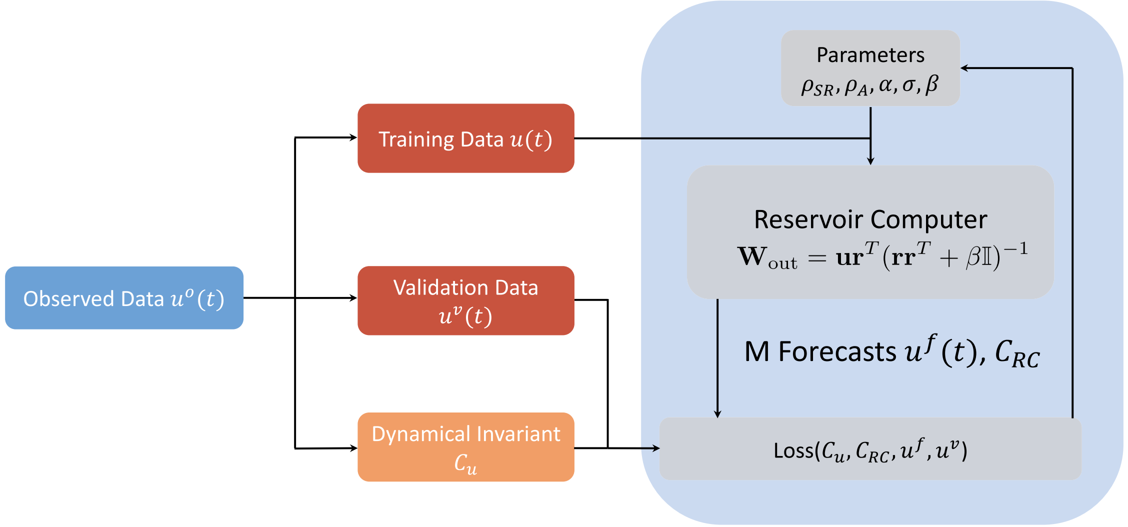
Zongyi Li et al. NeurIPS 2022

proposed Sobolev norm and dissipative regularization.



Jason A. Platt et al. Chaos (2023)

proposed enforcing dynamical invariants in reservoir computers.



NeurIPS 2023

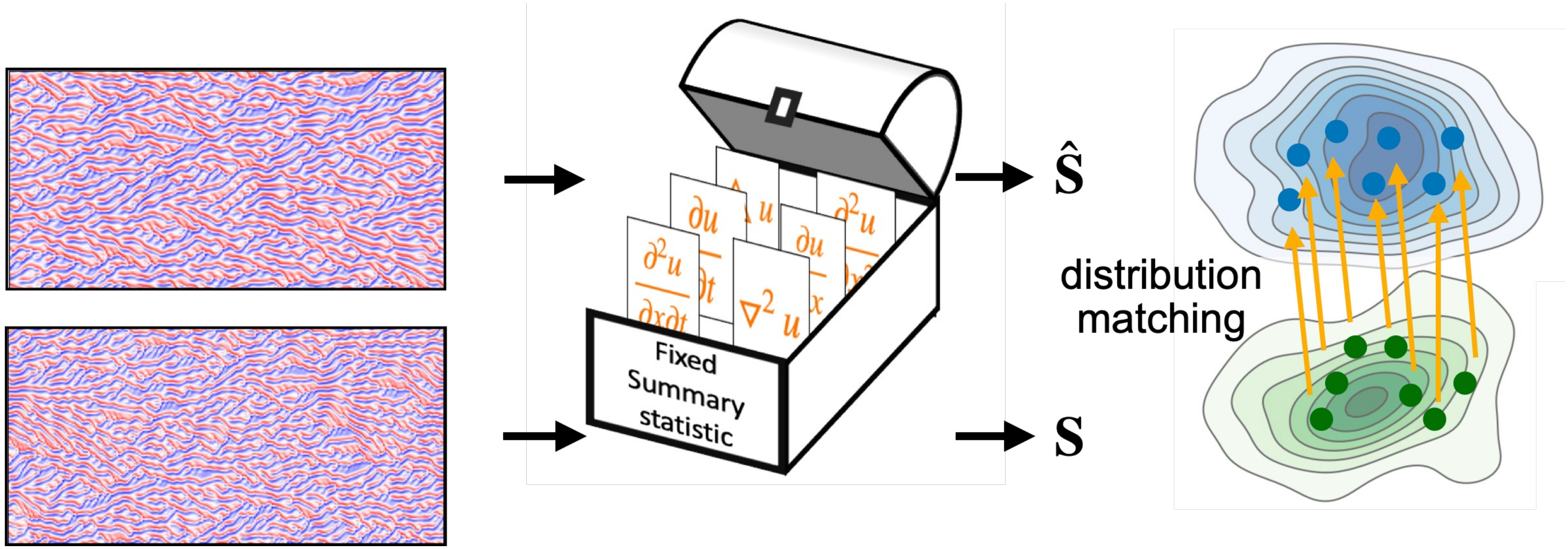
Using Known Summary Statistics

For **high-dimensional problems**, we cannot explicitly match the attractor measure.

Using Known Summary Statistics

For **high-dimensional problems**, we cannot explicitly match the attractor measure.

Instead, we can use **Sinkhorn divergence** (~ Wasserstein **optimal transport** distance) to match **known summary statistics**.



NeurIPS 2023

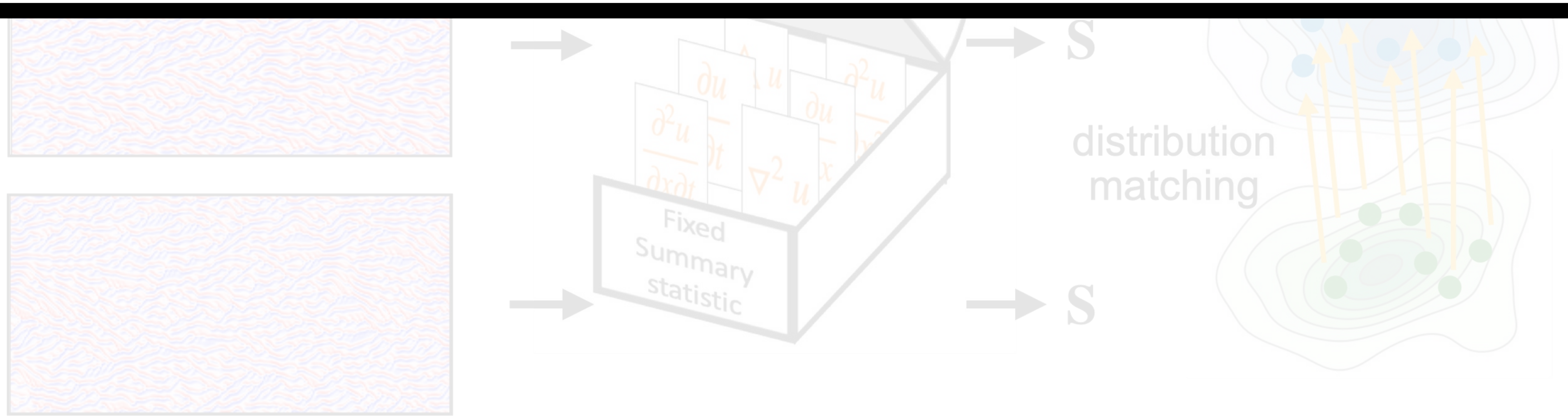
1. Using Known Summary Statistics

For **high-dimensional problems**, we cannot explicitly match the attract

Instead **nal**

transp

What if we don't know the relevant statistics?



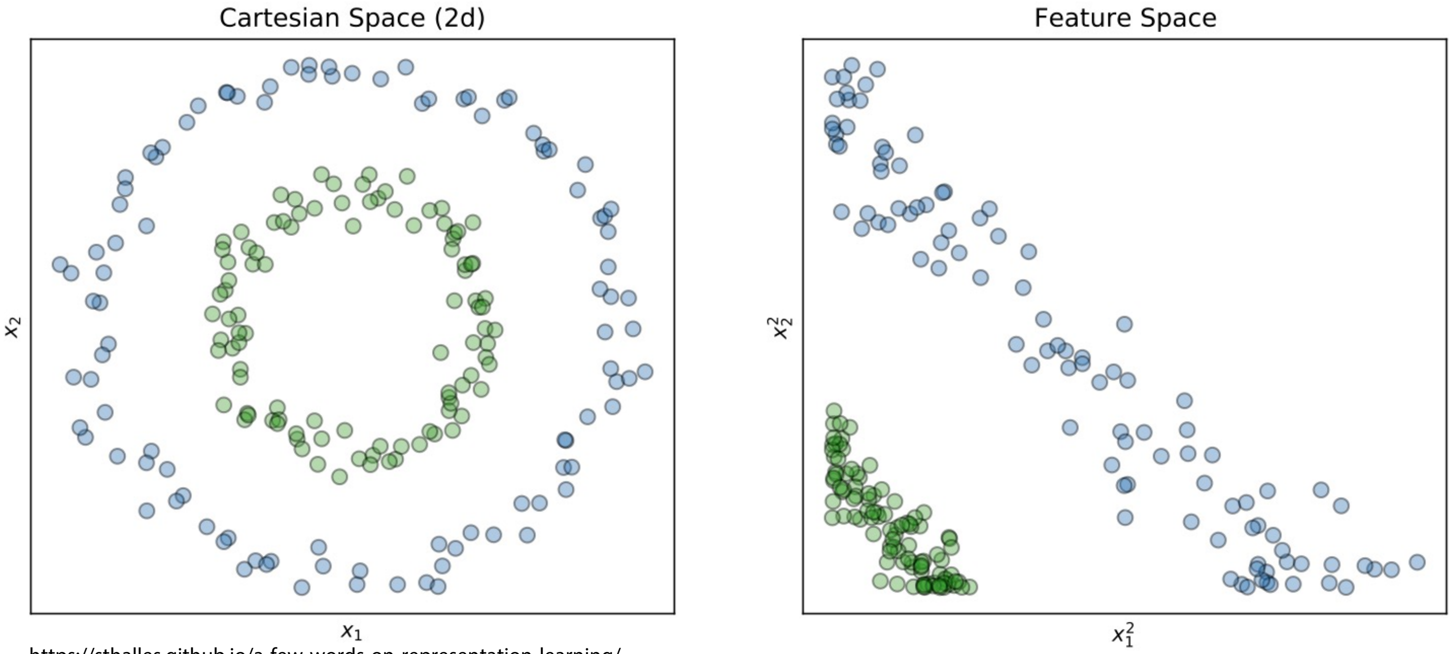
NeurIPS 2023

Representation Learning

What if we don't know the relevant statistics?

Representation learning aims to automatically discover **representations** or **features** that characterize a dataset.

- Useful for **dimensionality reduction**, **downstream tasks**, and sometimes **interpretability**.



https://sthalles.github.io/a-few-words-on-representation-learning/

NeurIPS 2023

Representation Learning

What if we don't know the relevant statistics?

Representation learning aims to automatically discover **representations** or **features** that characterize a dataset.

- Useful for **dimensionality reduction**, **downstream tasks**, and sometimes **interpretability**.

Scientists have always been doing representation learning!

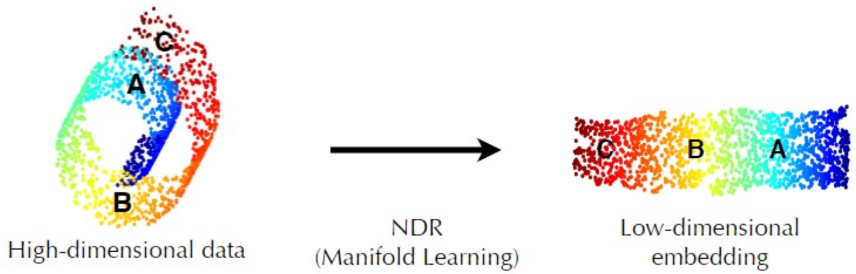
- The world is inherently high-dimensional.
- What we call **"scientific understanding"** is a low-dimensional description.

Representation Learning

What if we don't know the relevant statistics?

Classical Manifold Learning (*popular for data visualization*)

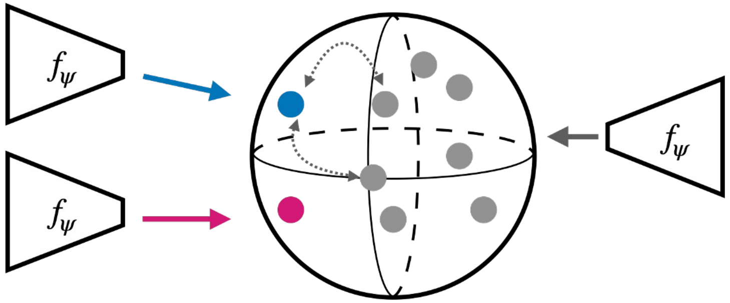
- PCA, MDS, Isomap, t-SNE, UMAP, LLE
- Diffusion maps, Laplacian eigenmaps



Unsupervised/Self-Supervised Learning

(*popular in computer vision & natural language processing*)

- Autoencoders, variational autoencoders
- Contrastive learning, CLIP



NeurIPS 2023

Representation Learning

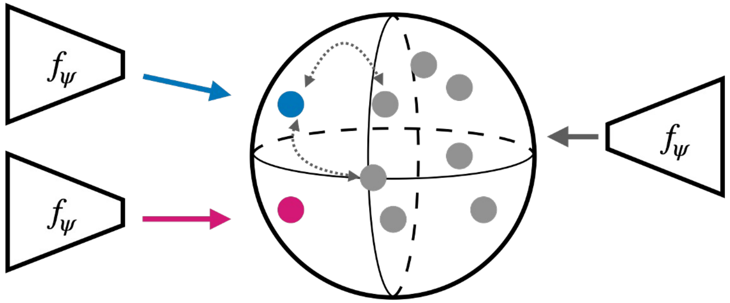
What if we don't know the relevant statistics?

Classical Manifold Learning (*popular for data visualization*)

- PCA, MDS, Isomap, t-SNE, UMAP, LLE
- Diffusion maps, Laplacian eigenmaps

Unsupervised/Self-Supervised Learning
(*popular in computer vision & natural language processing*)

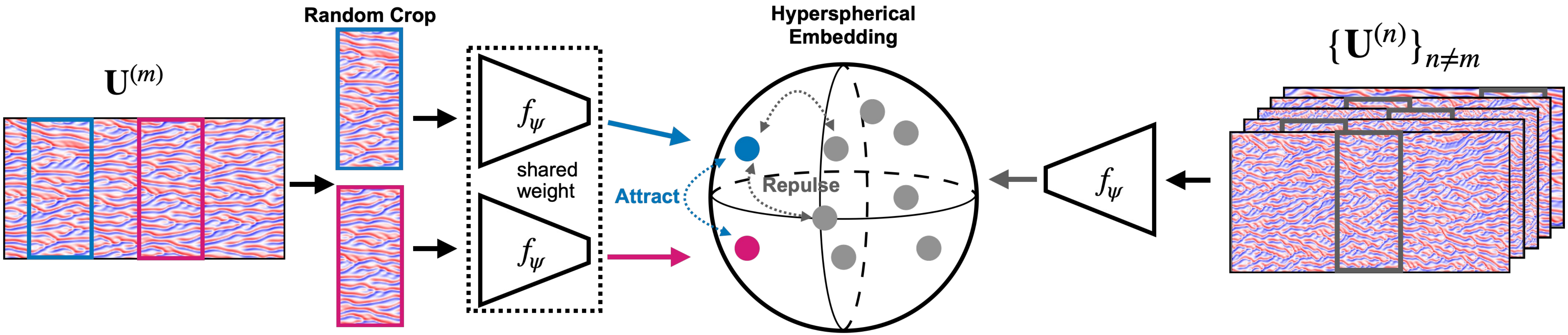
- Autoencoders, variational autoencoders
- Contrastive learning, CLIP



NeurIPS 2023

Learning Invariant Statistics from Data

Without prior knowledge, we can use **contrastive learning** in a multi-environment setting to learn **relevant invariant statistics**.



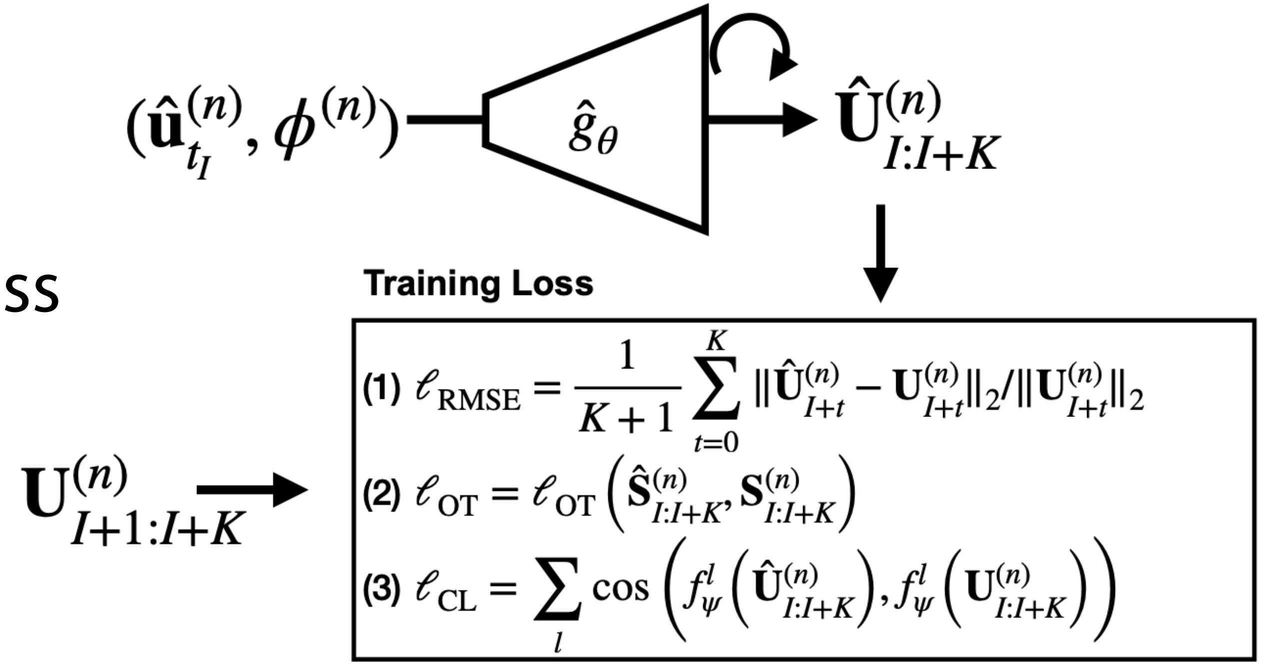
Training Emulators on Chaotic Dynamics

Optimal Transport (OT):

1. Choose summary statistics via expert knowledge
2. Train w/ RMSE + OT Sinkhorn loss

Contrastive Learning (CL):

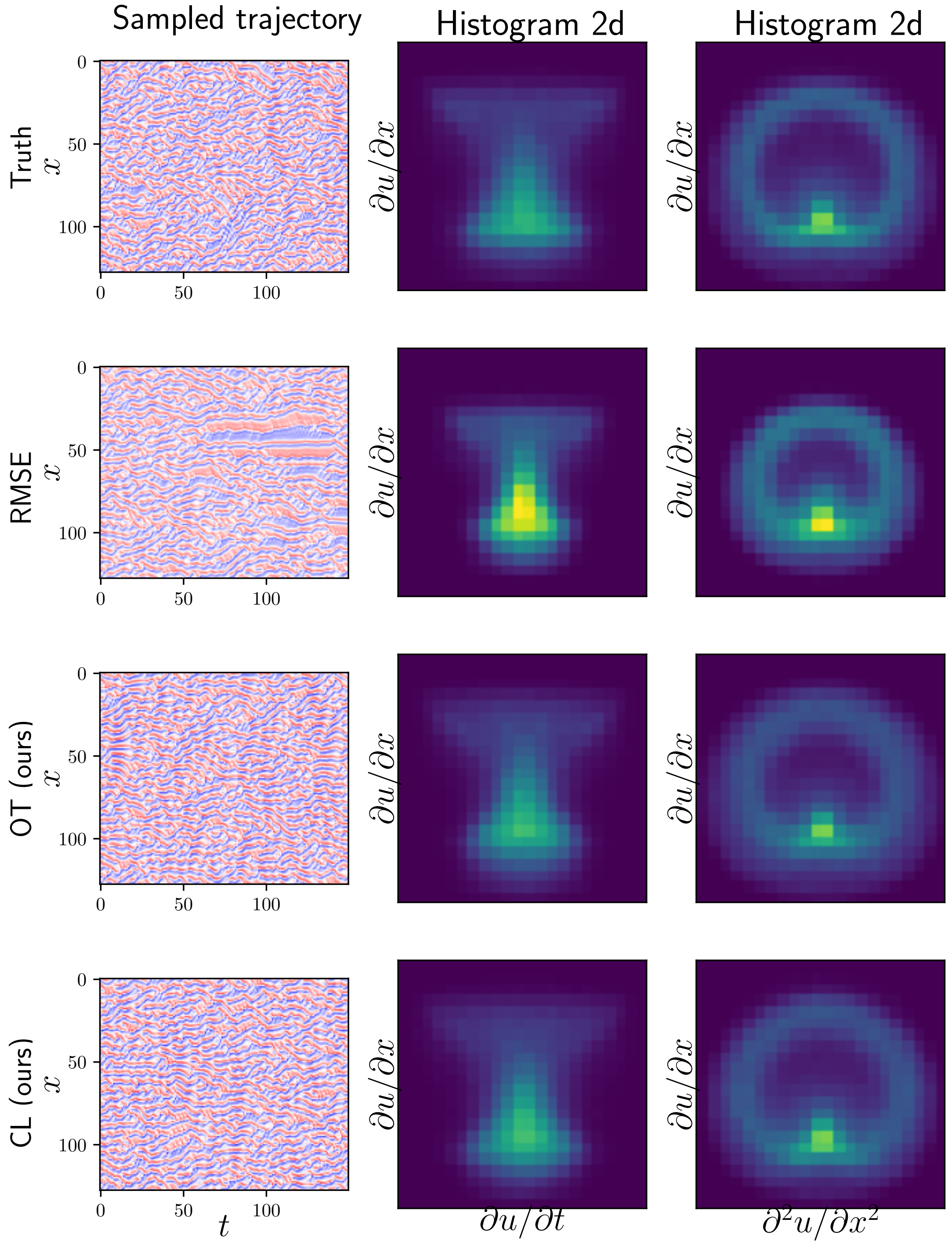
1. Learn invariant statistics via contrastive learning
2. Train w/ RMSE + CL feature loss



NeurIPS 2023

Emulator Evaluation: Kuramoto–Sivashinsky

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \phi\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

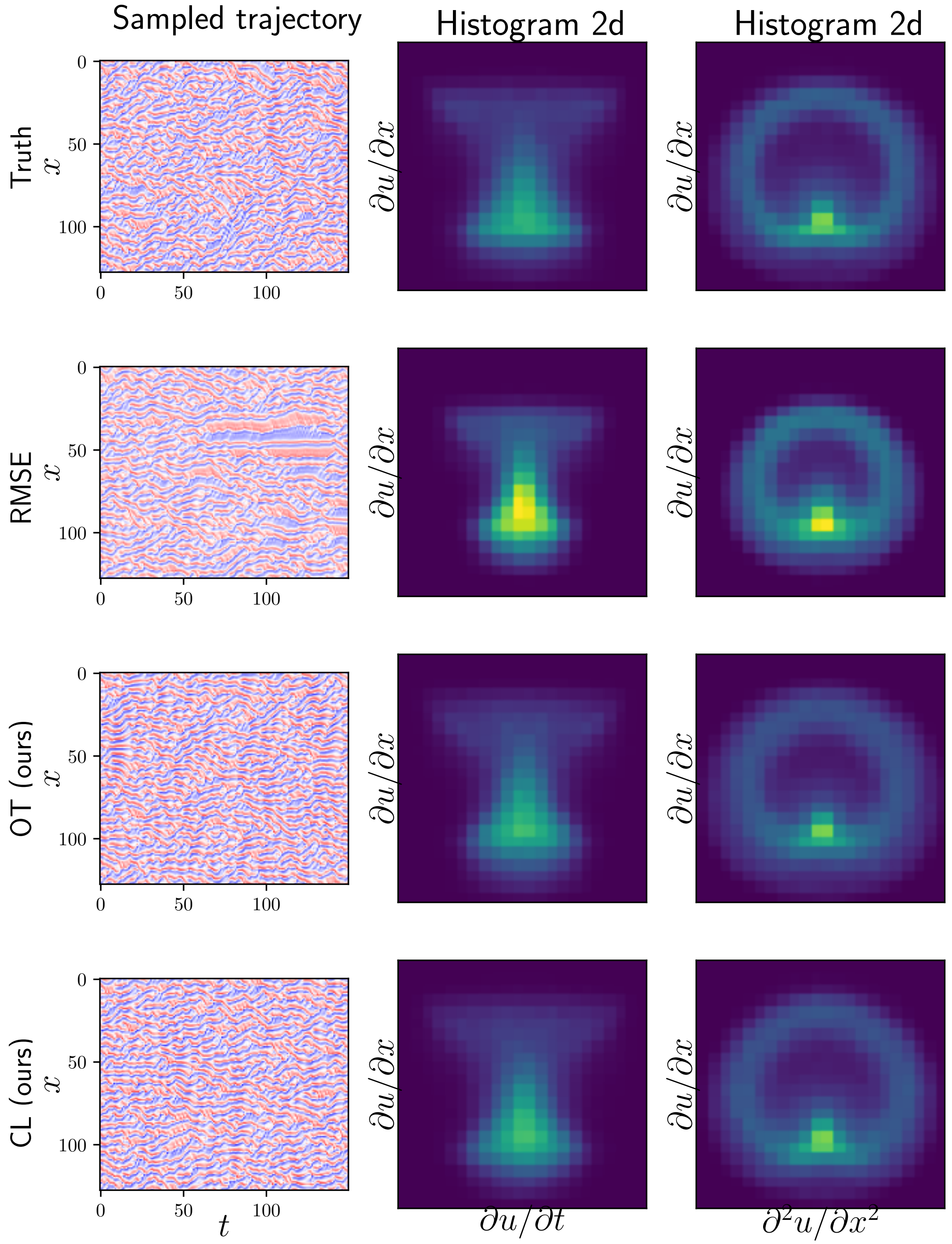


NeurIPS 2023

Emulator Evaluation: Kuramoto–Sivashinsky

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \phi\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

Training	Histogram Error ↓
ℓ_{RMSE}	0.390 (0.325, 0.556)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.172** (0.146, 0.197)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.193 (0.148, 0.247)



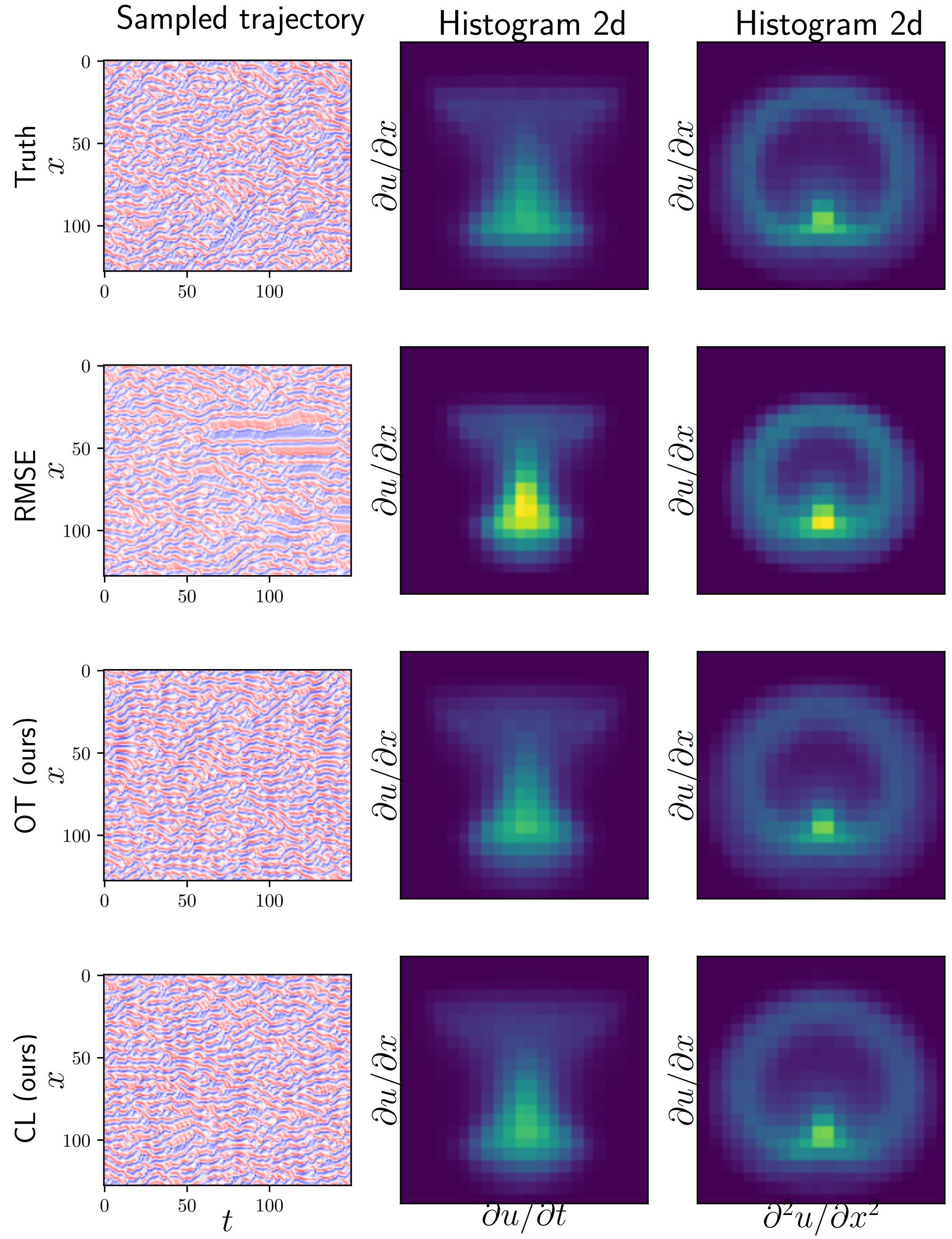
NeurIPS 2023

Emulator Evaluation: Kuramoto–Sivashinsky

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \phi\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

Training	Histogram Error ↓
ℓ_{RMSE}	0.390 (0.325, 0.556)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.172** (0.146, 0.197)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.193 (0.148, 0.247)

- We see significant improvements on statistical properties for noisy data.
- Again, contrastive learning requires **no prior knowledge**!



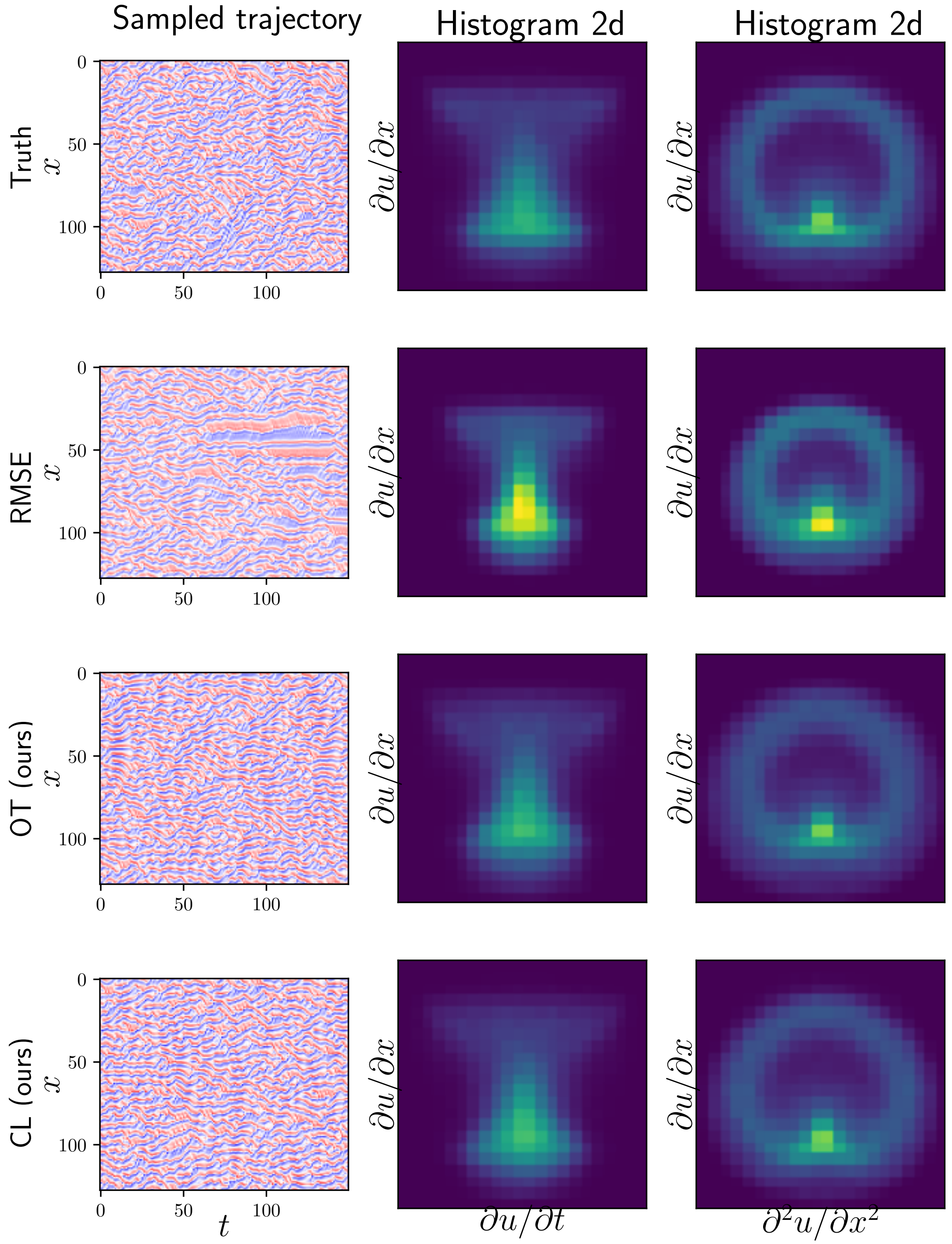
NeurIPS 2023

Emulator Evaluation: Kuramoto–Sivashinsky

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \phi\frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}$$

Training	Energy Spec. Error ↓
ℓ_{RMSE}	0.290 (0.225, 0.402)
$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	0.211 (0.188, 0.250)
$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	**0.176** (0.130, 0.245)

- We see significant improvements on statistical properties for noisy data.
- Again, contrastive learning requires **no prior knowledge**!

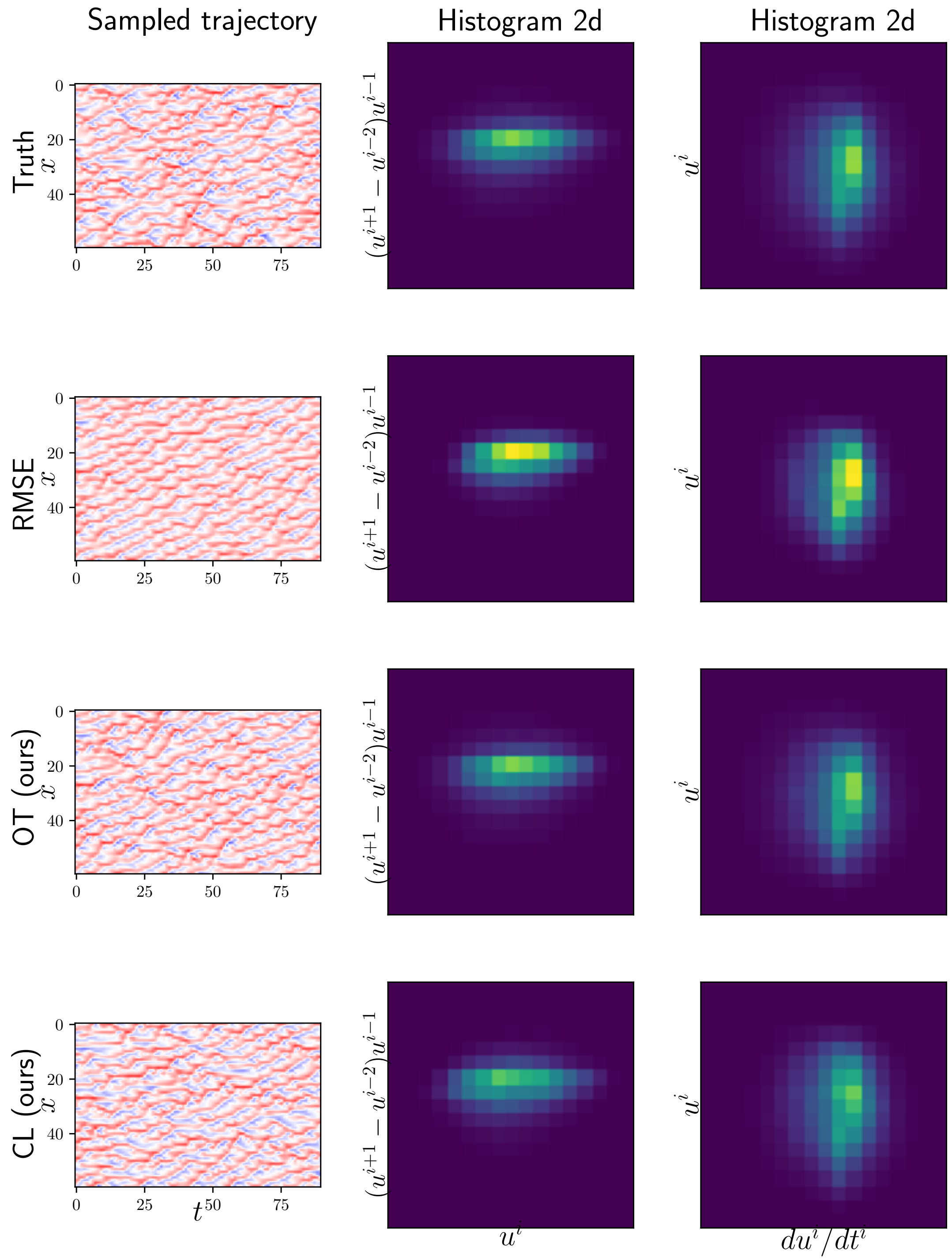


NeurIPS 2023

Emulator Evaluation: Lorenz 96

$$\frac{du_i}{dt} = (u_{i+1} - u_{i-2})u_{i-1} - u_i + F$$

r	Training	Histogram Error ↓
0.1	ℓ_{RMSE}	0.056 (0.051, 0.062)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.029** (0.027, 0.032)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.033 (0.029, 0.037)
0.2	ℓ_{RMSE}	0.130 (0.118, 0.142)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.039** (0.035, 0.042)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.073 (0.066, 0.080)
0.3	ℓ_{RMSE}	0.215 (0.204, 0.234)
	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.057** (0.052, 0.064)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.132 (0.111, 0.151)



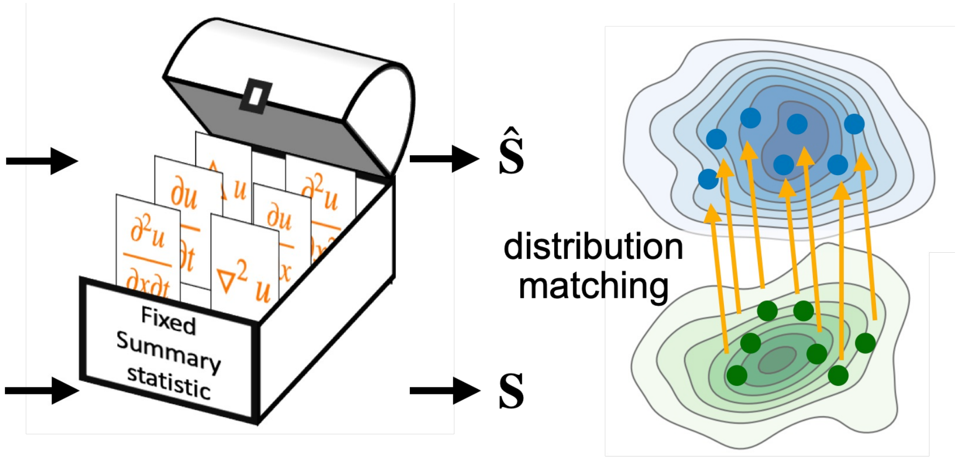
NeurIPS 2023

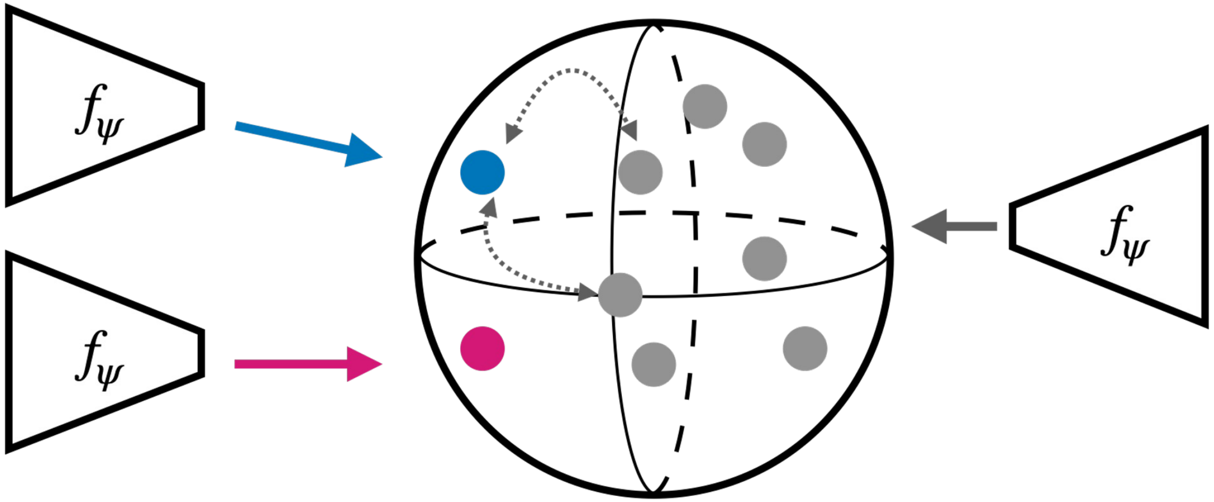
Training Emulators for Chaos:
Summary

We train emulators to match the statistics of **high-dimensional, spatiotemporal chaotic attractors**.

Our approach performs significantly better in the **high noise regime**.

Contrastive representation learning can identify relevant statistics **without prior knowledge**.





Training Emulators for Chaos:
Future Directions

Applications:

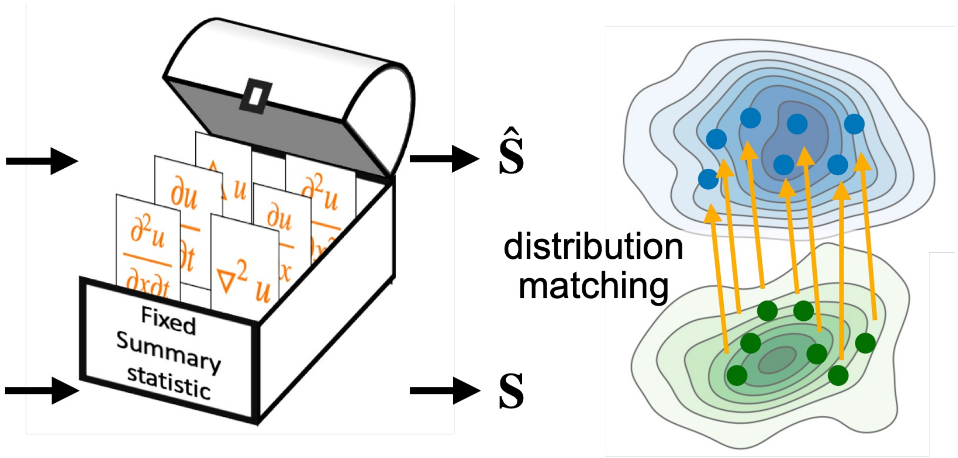
- Fluid turbulence
- Weather/climate modeling

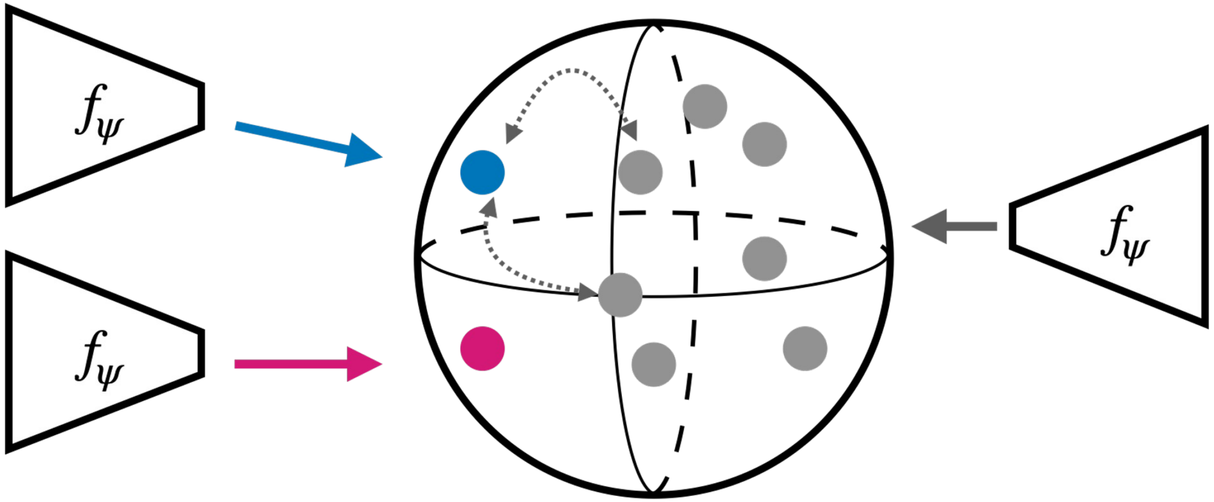
Extensions:

- Transient dynamics
- Stochastic dynamics

Interpretability:

- What are the learned statistics?
- Can we improve interpretability?





Acknowledgments

Ruoxi (Roxie) Jiang

Elena Orlova

Rebecca Willett

Vincenzo Vitelli

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

SCHMIDT FUTURES

NeurIPS 2023

UChicago AI + Science Summer School

July 15–19, 2024

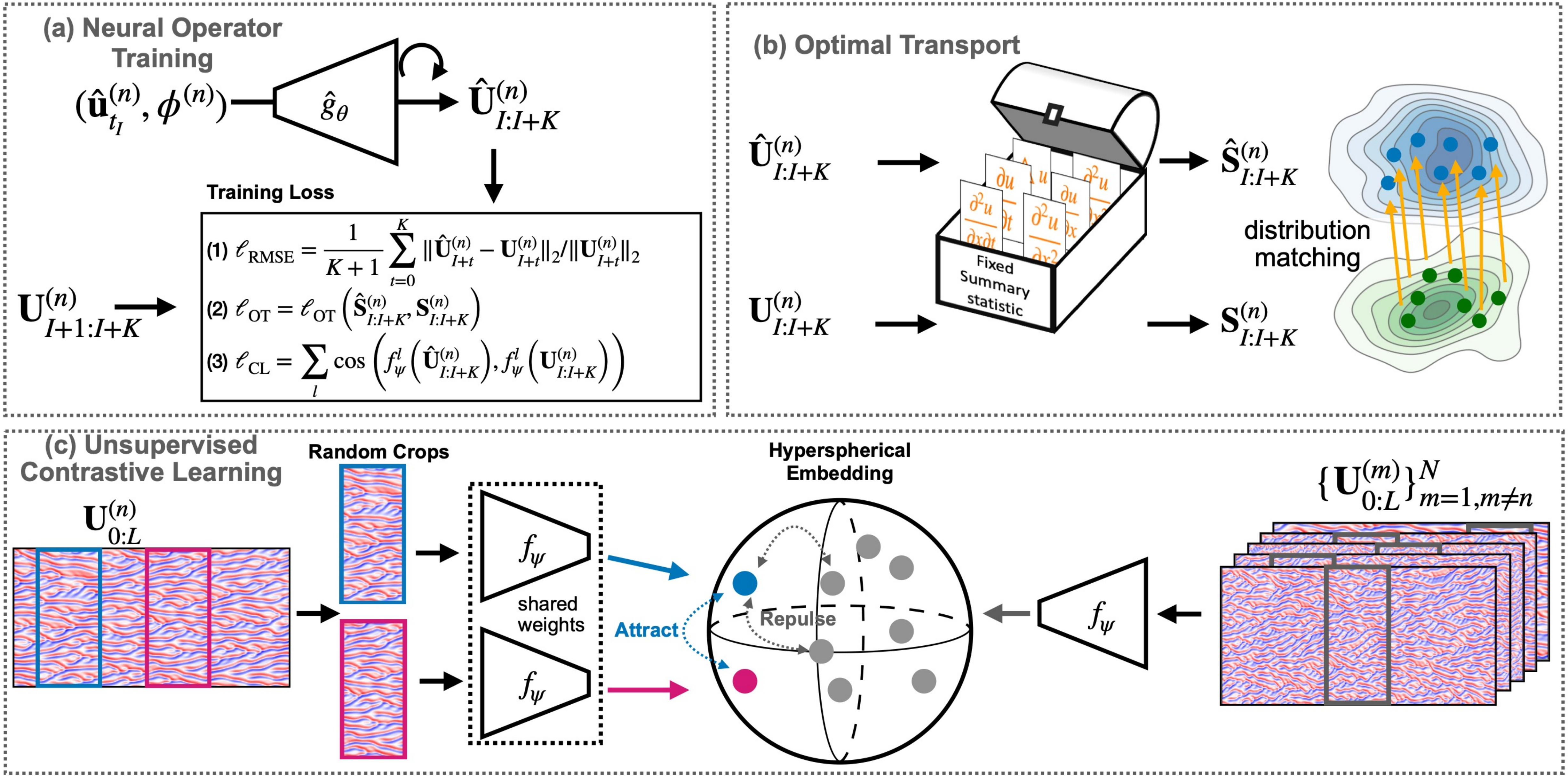
Applications due **tomorrow (March 8)**!

https://datascience.uchicago.edu/events/aiscience-summer-school-2024

NeurIPS 2023

THE UNIVERSITY OF CHICAGO
DATA SCIENCE INSTITUTE

AI in Science
A program of SCHMIDT FUTURES



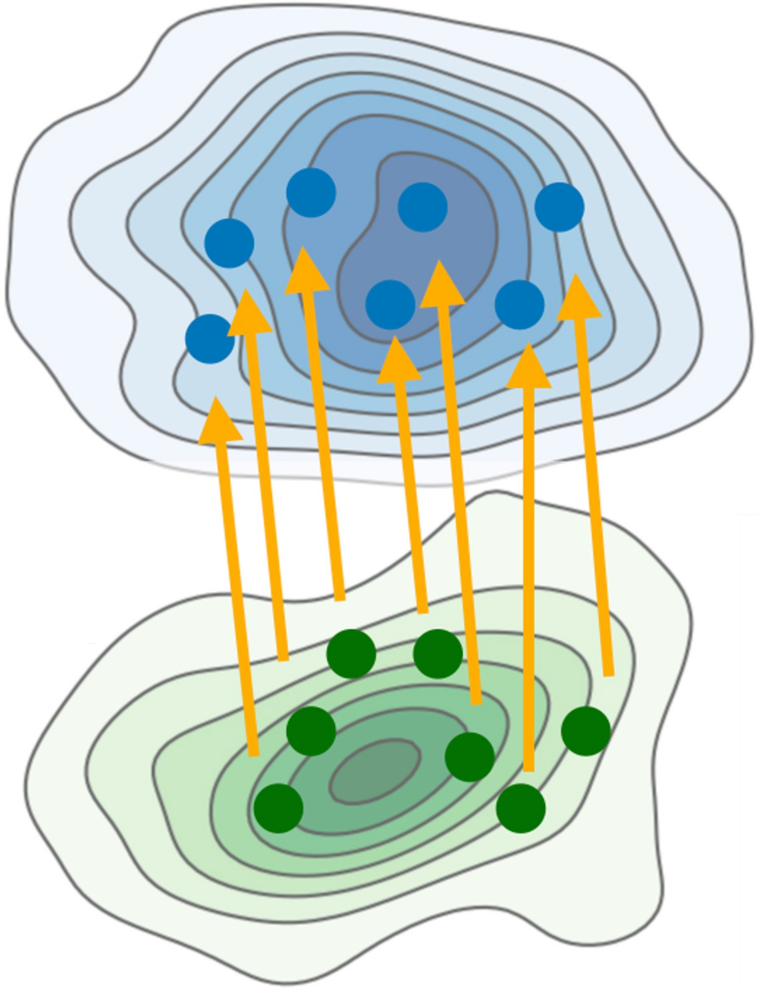
NeurIPS 2023

Wasserstein Distance

$$\frac{1}{2}W(\mathbf{S},\hat{\mathbf{S}})^2 := \min_{T\in\Pi}\sum_{i,j} T_{ij}C_{ij}$$

$$C_{ij} = \tfrac{1}{2}\|\mathbf{s}_i - \hat{\mathbf{s}}_j\|^2$$

$$\forall i,j, T_{ij} \geq 0, \textstyle\sum_j T_{ij} = 1, \text{ and } \textstyle\sum_i T_{ij} = 1$$

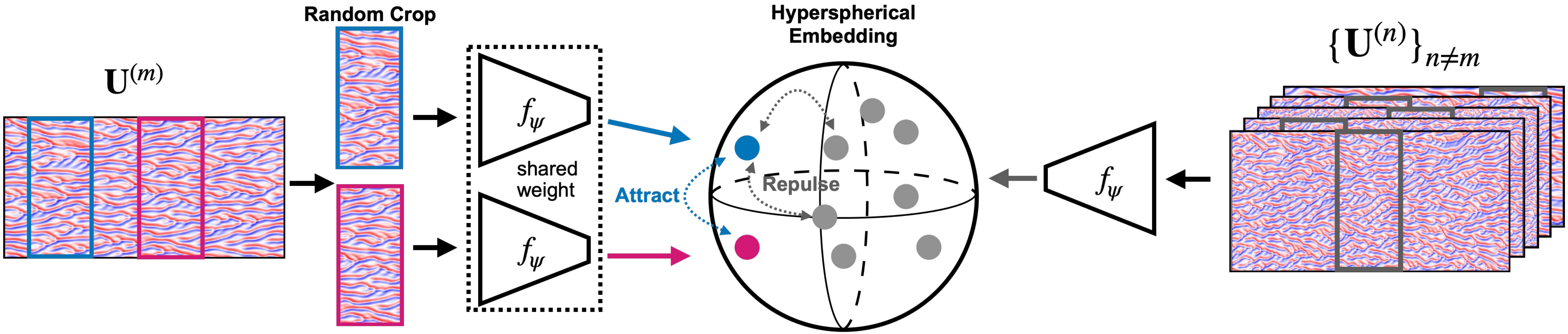


Entropy-regularized Wasserstein distance:

$$\frac{1}{2}W^{\gamma}(\mathbf{S},\hat{\mathbf{S}})^2 := \min_{T\in\Pi}\sum_{i,j} T_{ij}C_{ij} - \gamma\, h(T)$$

$$h(T) = -\textstyle\sum_{i,j} T_{ij}\log T_{ij}$$

Contrastive Learning: InfoNCE Loss



$$\ell_{\mathrm{InfoNCE}}(\psi;\tau) := \mathop{\mathbb{E}}_{\substack{n\in\{1,\dots,N\}\\ I,J\in\{0,\dots,L-K\}}}\left[-\log\left(\frac{\exp\left(\langle f_\psi(\mathbf{U}^{(n)}_{I:I+K}), f_\psi(\mathbf{U}^{(n)}_{J:J+K})\rangle/\tau\right)}{\mathop{\mathbb{E}}_{\substack{m\neq n\\ H\in\{0,\dots,L-K\}}}\left[\exp\left(\langle f_\psi(\mathbf{U}^{(n)}_{I:I+K}), f_\psi(\mathbf{U}^{(m)}_{H:H+K})\rangle/\tau\right)\right]}\right)\right]$$

NeurIPS 2023

Full Results: Kuramoto–Sivashinsky

Training	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓
ℓ_{RMSE}	0.390 (0.325, 0.556)	0.290 (0.225, 0.402)	0.101 (0.069, 0.122)
$\ell_{\text{Sobolev}} + \ell_{\text{dissipative}}$	0.427 (0.289, 0.616)	0.237 (0.204, 0.315)	**0.023** (0.012, 0.047)
$\ell_{\text{MMD}} + \ell_{\text{RMSE}}$	0.245 (0.218, 0.334)	0.216 (0.186, 0.272)	0.101 (0.058, 0.125)
$\ell_{\text{OT}} + \ell_{\text{RMSE}}$	**0.172** (0.146, 0.197)	0.211 (0.188, 0.250)	**0.094** (0.041, 0.127)
$\ell_{\text{CL}} + \ell_{\text{RMSE}}$	0.193 (0.148, 0.247)	**0.176** (0.130, 0.245)	0.108 (0.068, 0.132)

Training	Histogram Error ↓	Energy Spec. Error↓	Leading LE Error ↓
ℓ_{RMSE} ($\sigma_b = 0.1$)	**0.390** (0.326, 0.556)	**0.290** (0.226, 0.402)	**0.098** (0.069, 0.127)
ℓ_{RMSE} ($\sigma_b = 0.5$)	1.011 (0.788, 1.264)	0.493 (0.379, 0.623)	**0.098** (0.041, 0.427)
ℓ_{RMSE}	**0.390** (0.325, 0.556)	**0.290** (0.225, 0.402)	0.101 (0.069, 0.122)

Full Results: Lorenz 96

r	Training	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓	FD Error ↓
	ℓ_{RMSE}	0.056 (0.051, 0.062)	0.083 (0.078, 0.090)	**0.013** (0.006, 0.021)	1.566 (0.797, 2.309)
0.1	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.029** (0.027, 0.032)	**0.058** (0.052, 0.064)	0.050 (0.040, 0.059)	1.424 (0.646, 2.315)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.033 (0.029, 0.037)	**0.058** (0.049, 0.065)	0.065 (0.058, 0.073)	**1.042** (0.522, 1.685)
	ℓ_{RMSE}	0.130 (0.118, 0.142)	0.182 (0.172, 0.188)	0.170 (0.156, 0.191)	2.481 (1.428, 3.807)
0.2	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.039** (0.035, 0.042)	**0.086** (0.079, 0.095)	0.016 (0.006, 0.030)	2.403 (1.433, 3.768)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.073 (0.066, 0.080)	0.131 (0.117, 0.149)	**0.012** (0.006, 0.018)	**1.681** (0.656, 2.682)
	ℓ_{RMSE}	0.215 (0.204, 0.234)	0.291 (0.280, 0.305)	0.440 (0.425, 0.463)	3.580 (2.333, 4.866)
0.3	$\ell_{\mathrm{OT}} + \ell_{\mathrm{RMSE}}$	**0.057** (0.052, 0.064)	**0.123** (0.116, 0.135)	0.084 (0.062, 0.134)	3.453 (2.457, 4.782)
	$\ell_{\mathrm{CL}} + \ell_{\mathrm{RMSE}}$	0.132 (0.111, 0.151)	0.241 (0.208, 0.285)	**0.064** (0.045, 0.091)	**1.894** (0.942, 3.108)

Training statistics	Histogram Error ↓	Energy Spec. Error ↓	Leading LE Error ↓	FD Error ↓
$\mathbf{S}$ (full)	**0.057** (0.052, 0.064)	**0.123** (0.116, 0.135)	0.084 (0.062, 0.134)	3.453 (2.457, 4.782)
$\mathbf{S}_1$ (partial)	0.090 (0.084, 0.098)	0.198 (0.189, 0.208)	0.263 (0.217, 0.323)	3.992 (2.543, 5.440)
$\mathbf{S}_2$ (minimum)	0.221 (0.210, 0.234)	0.221 (0.210, 0.230)	0.276 (0.258, 0.291)	**3.204** (2.037, 4.679)

NeurIPS 2023