PennState

ICLR

Learning vector fields of Differential equations on manifolds with Geometrically constrained operator-valued kernels

Daning Huang[1], Hanyang He[2], John Harlim[3], Yan Li[2]

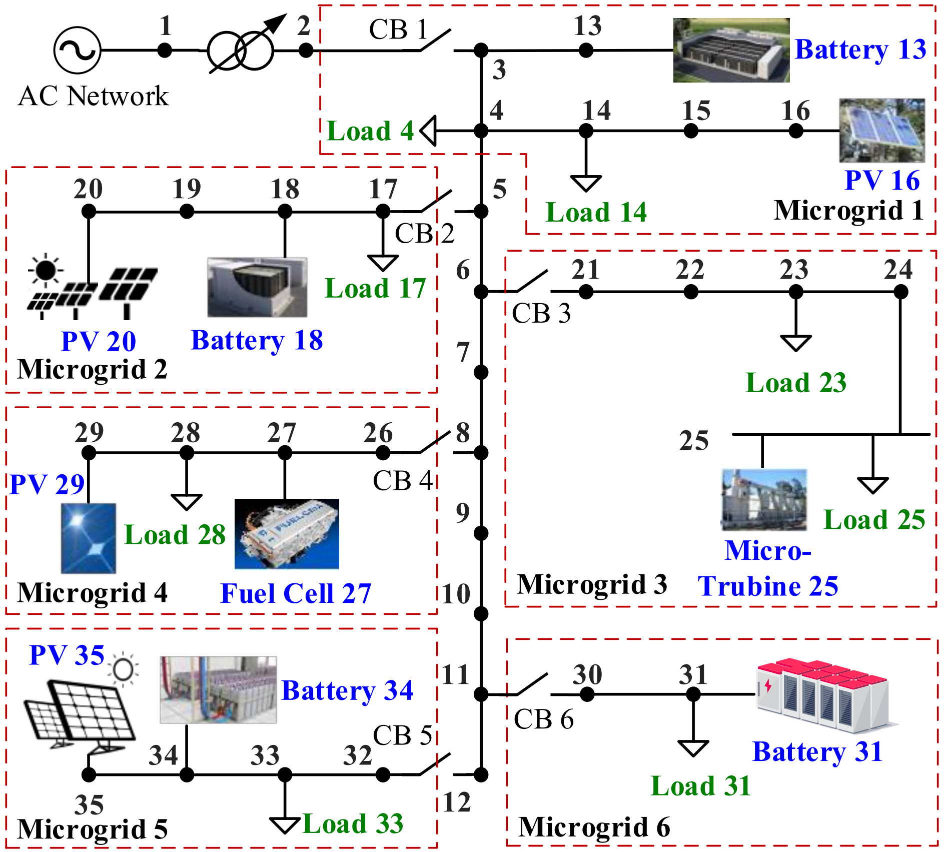
[1]Dept. of Aerospace Engineering, [2]Dept. of Electrical Engineering,

[3]Dept. of Mathematics, Dept. of Meteorology & Atmospheric Science, Inst. for Comp. and Data Sciences

The Pennsylvania State University, University Park, PA 16802, USA

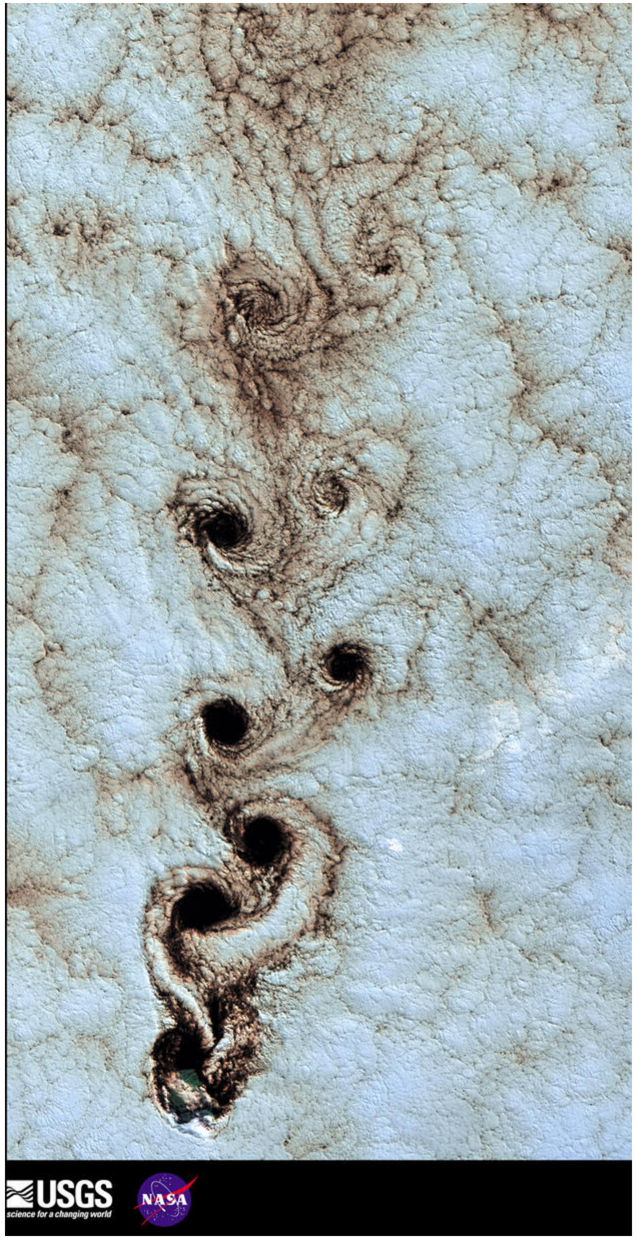
{daning,hfh5310,jharlim,yql5925}@psu.edu

ODEs on manifolds are common



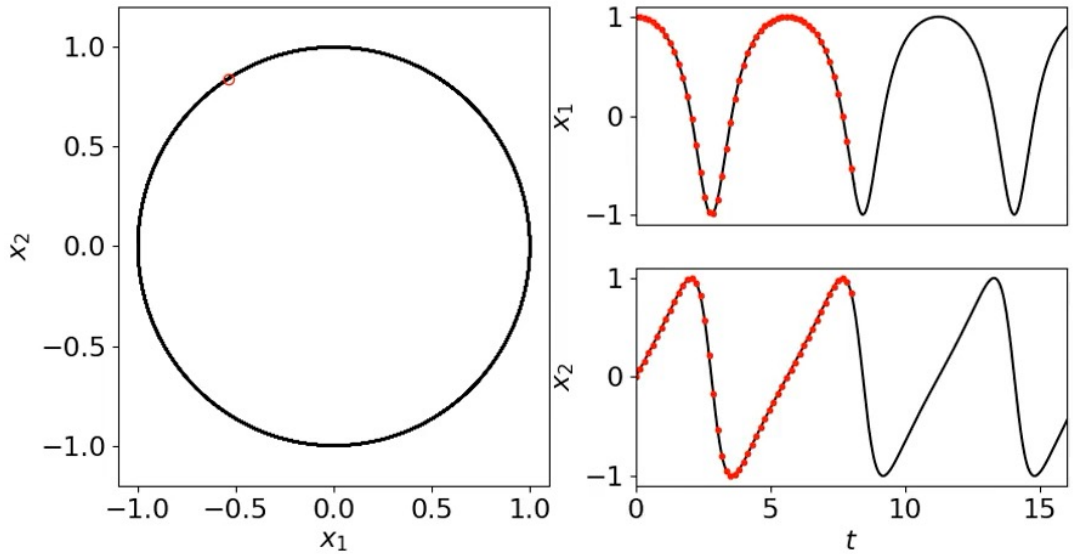
(a) Power grid
(multiple time scales)

(b) Robotic system
(algebraic constraints)



(c) Fluid dynamics
(inertia manifold)

Learning and solving such ODEs is non-trivial



Simple example

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in M$$

Manifold: M

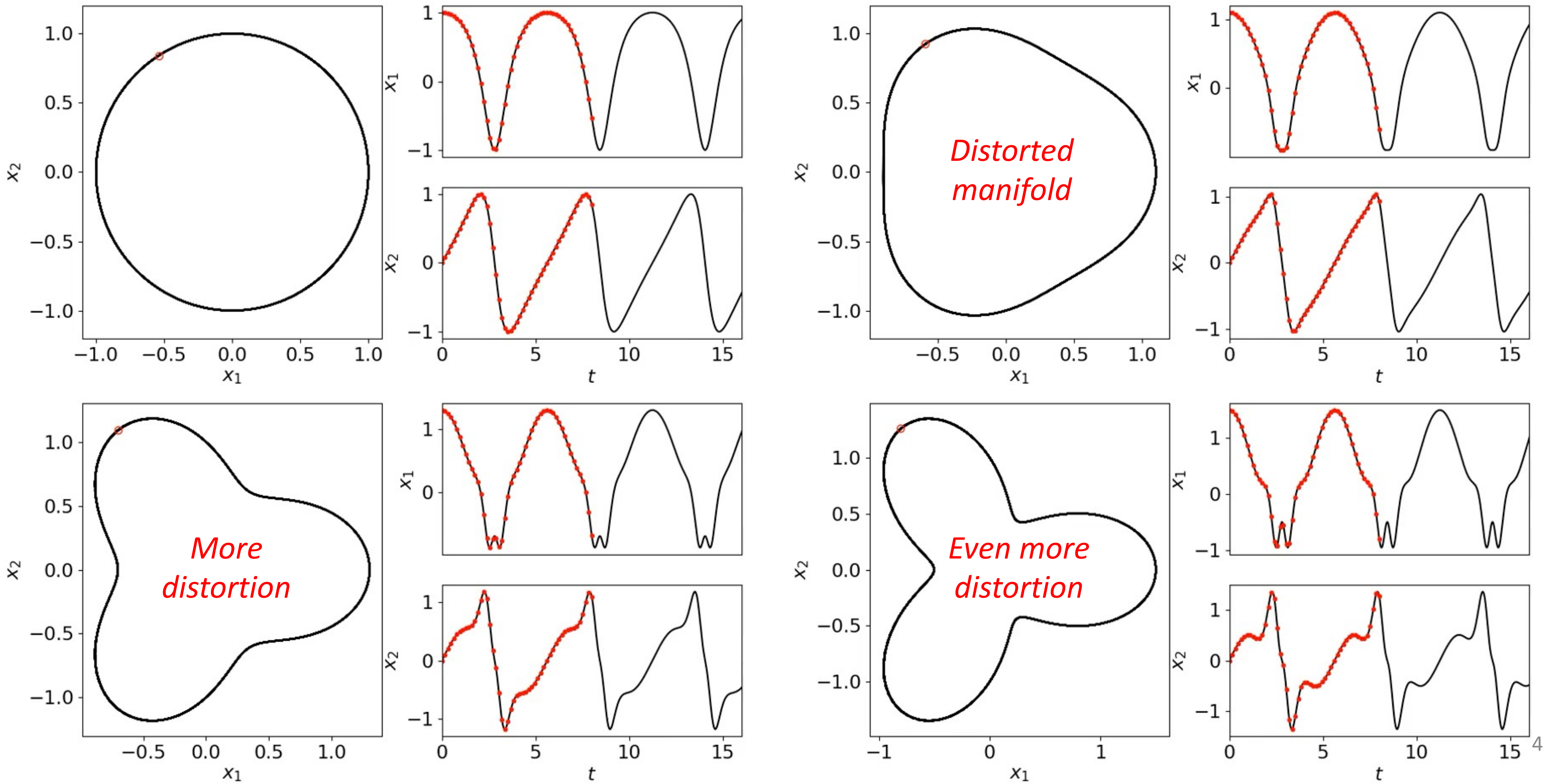
- Intrinsically 1D
- Embedded in 2D

Question:

Given trajectory data $\{(t_i, \mathbf{x}_i)\}_{i=1}^N$

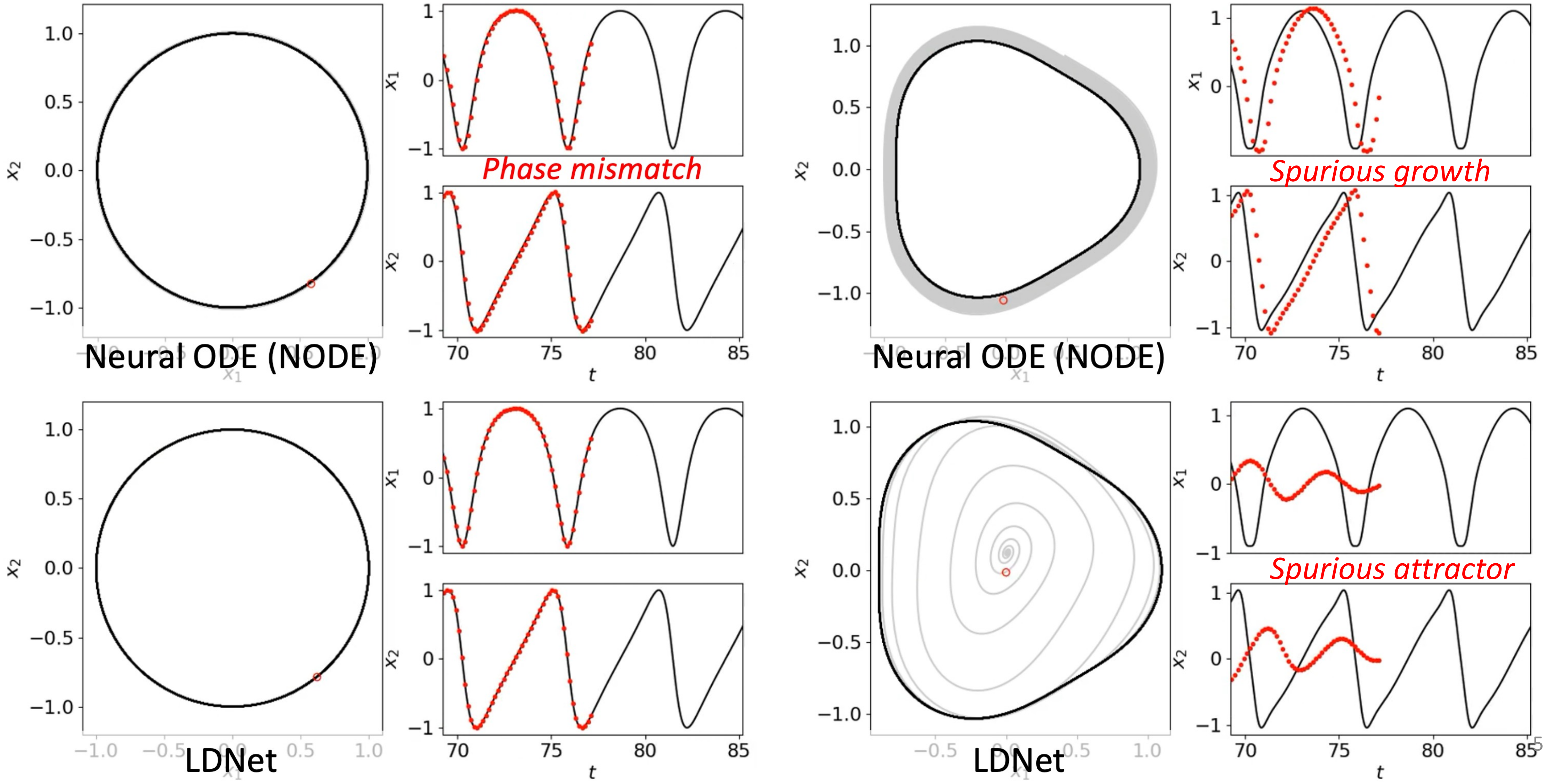
- How to learn $\mathbf{f}(\mathbf{x})$, which should be a vector field on M?
- How to solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, so that the solution stays on M?

Learning and solving such ODEs is non-trivial



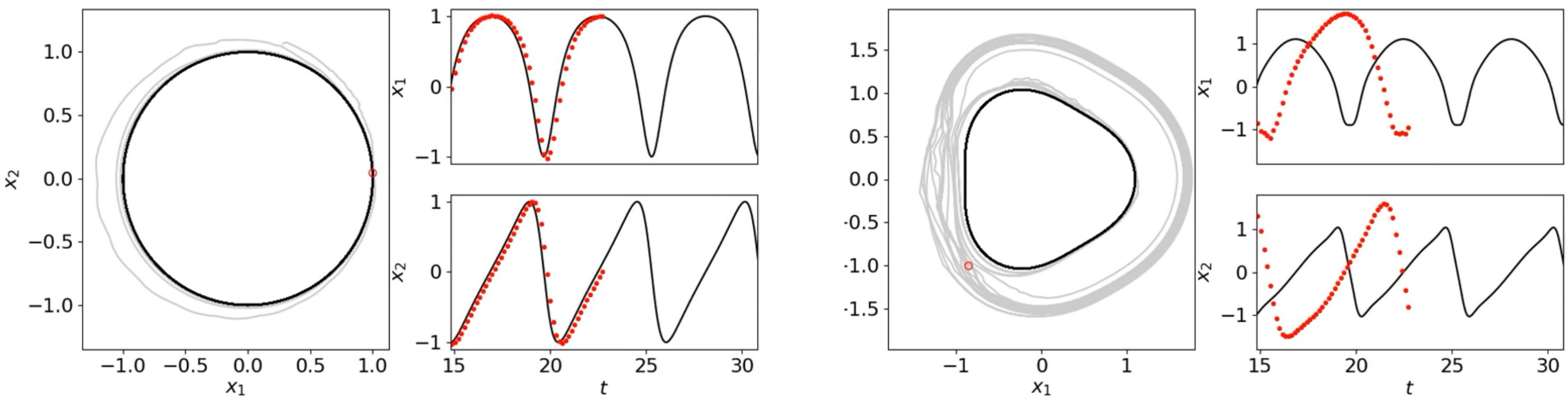
#1 Suboptimal compression
#2 Unconstrained dynamics
#3 Vanilla time integrator
#4 Expensive training

Challenges illustrated – 1/2



Challenges illustrated – 2/2

#1 Suboptimal compression
#2 Unconstrained dynamics
#3 Vanilla time integrator
#4 Expensive training

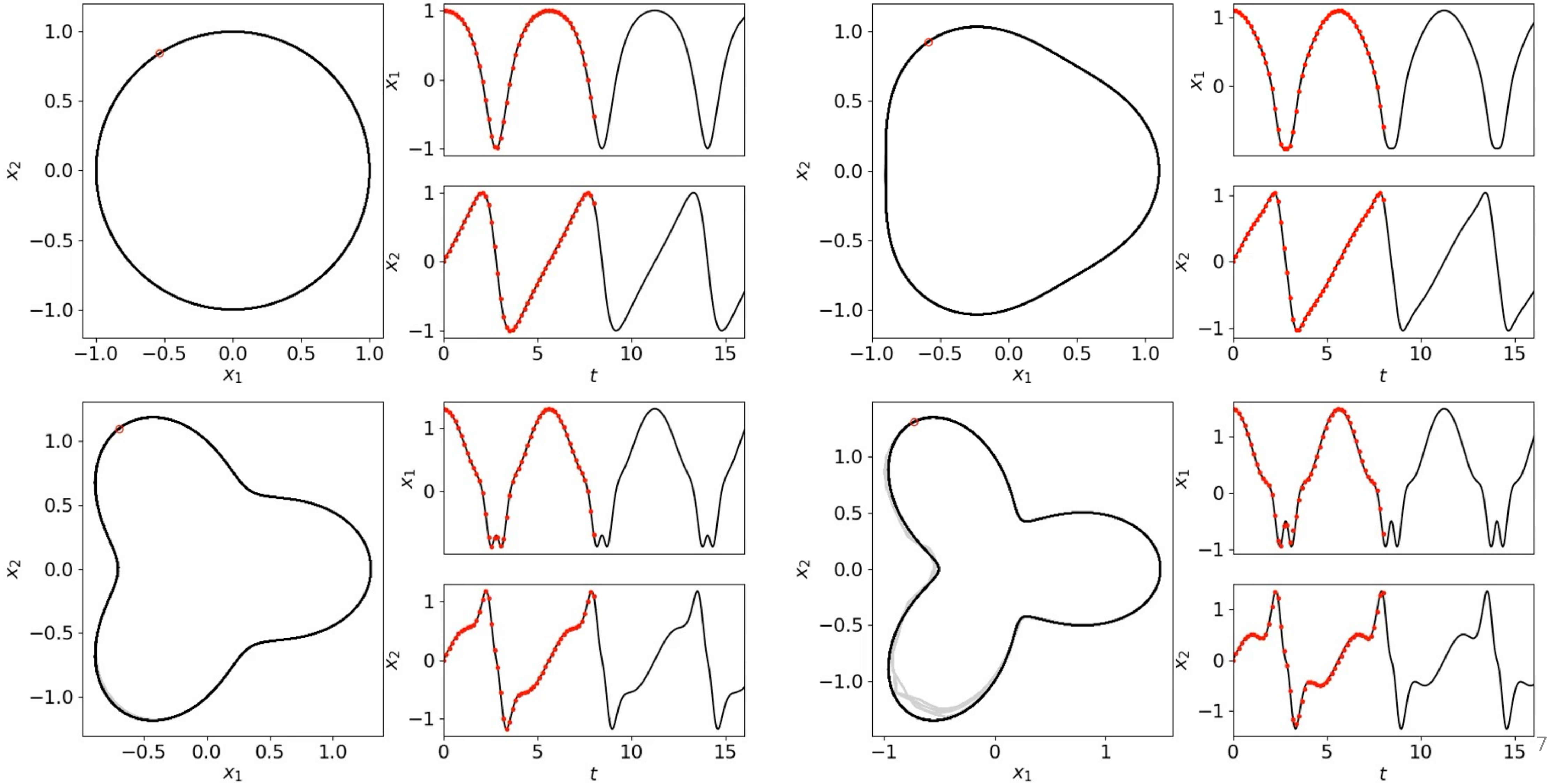


Correct dynamics
but solution by RK2

→ *Violation of geometrical constraints*

Our method to resolve all

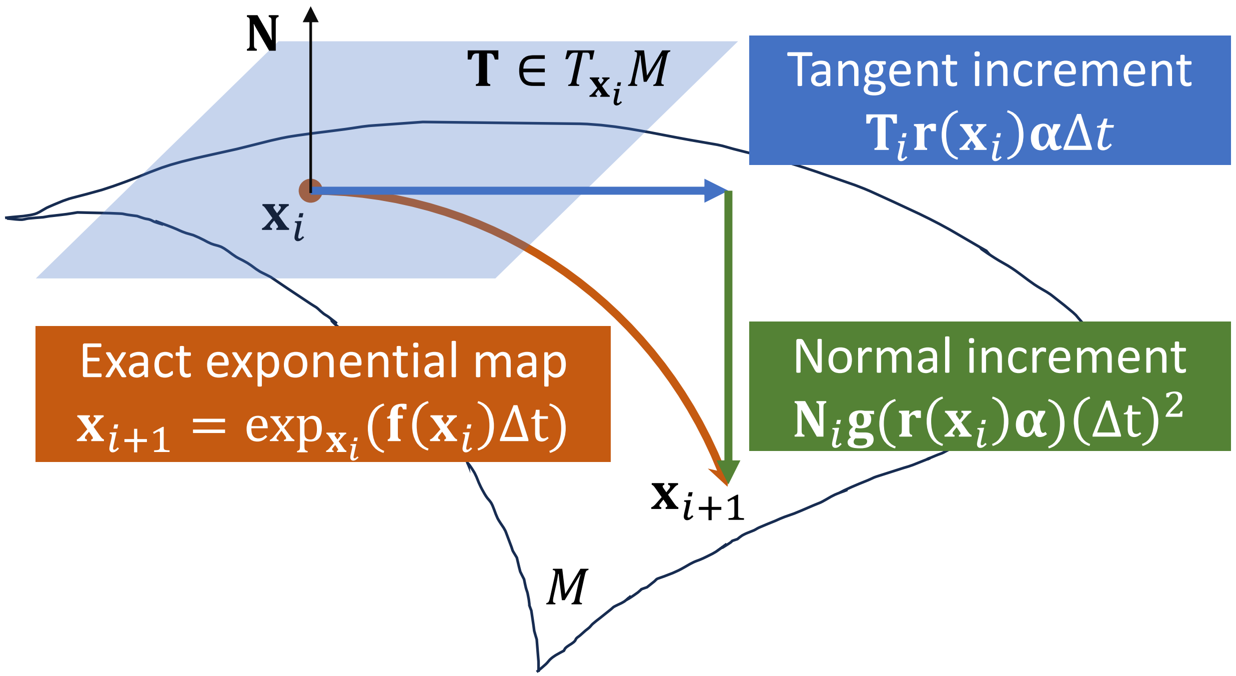
#1 Suboptimal compression
#2 Unconstrained dynamics
#3 Vanilla time integrator
#4 Expensive training



Our solution

Identity of exponential map:

$$\mathbf{x}_{i+1} = \exp_{\mathbf{x}_i}(\mathbf{f}(\mathbf{x}_i)\Delta t) = \mathbf{x}_i + \underbrace{\mathbf{T}_i\mathbf{T}_i^\top(\mathbf{x}_{i+1} - \mathbf{x}_i)}_{\text{Tangent: } \mathbf{f}(\mathbf{x}_i)\Delta t} + \underbrace{\mathbf{N}_i\mathbf{N}_i^\top(\mathbf{x}_{i+1} - \mathbf{x}_i)}_{\text{Normal: } O((\Delta t)^2)}$$

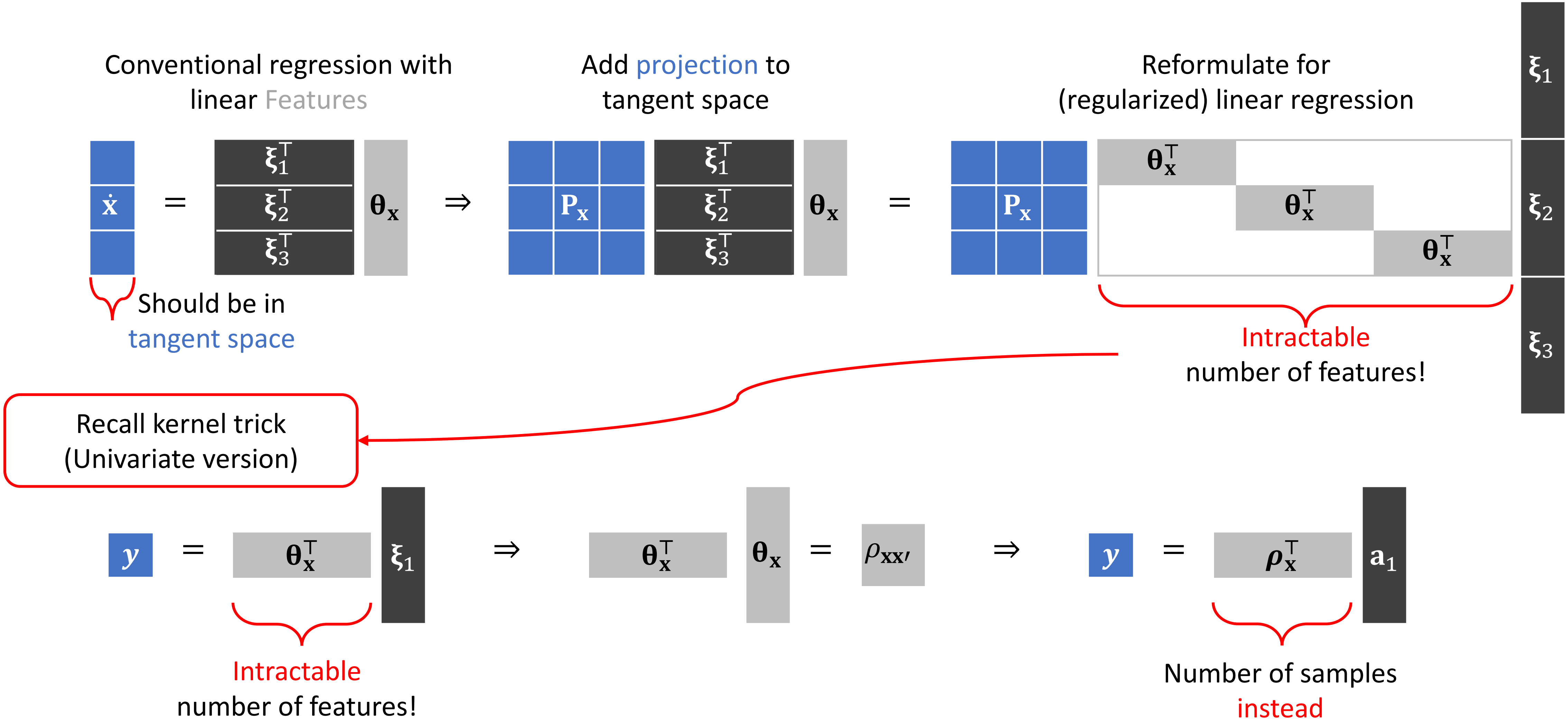


- *Ingredient 1:* Solves #1 and #2
Geometric Multivariate
Kernel Ridge Regression (GMKRR)

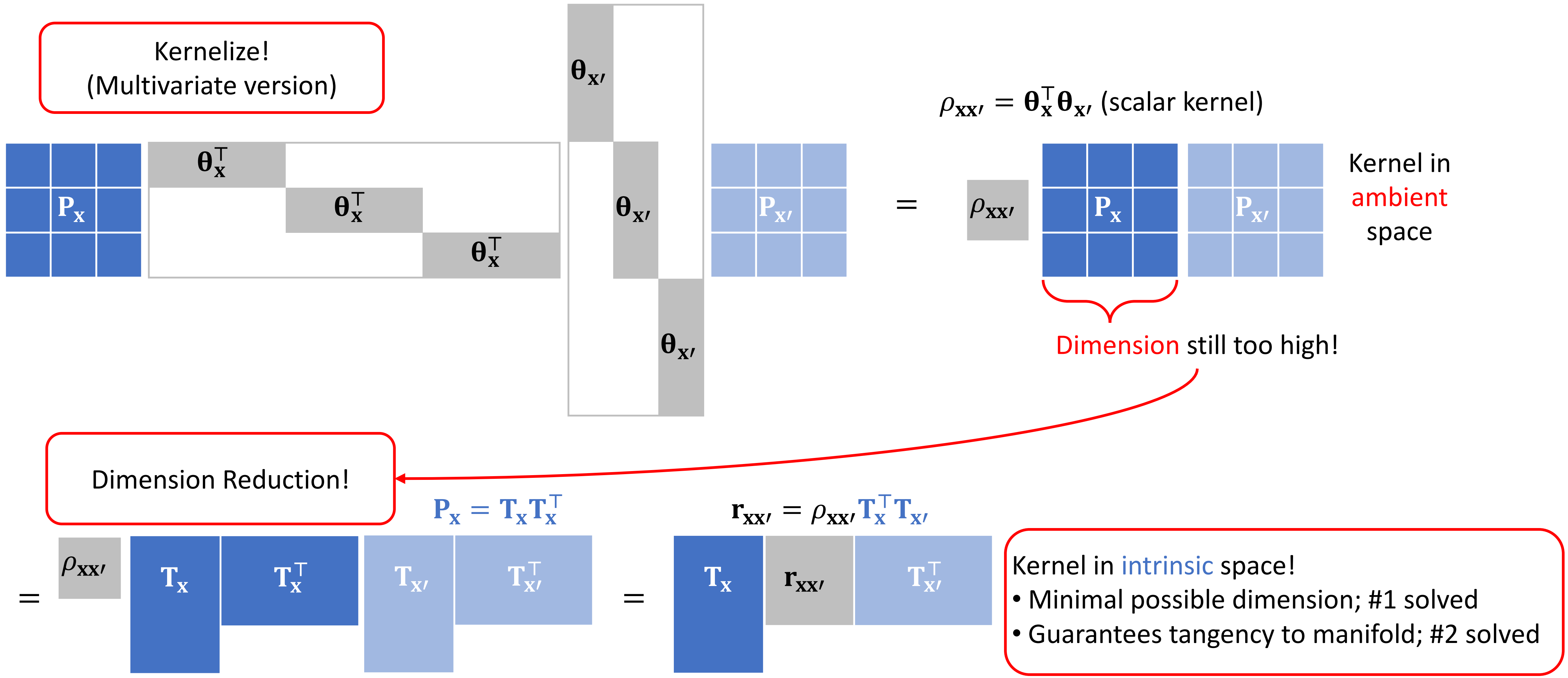
- *Ingredient 2:* Solves #3
Normal Correction (NC) by
Generalized Moving Least Squares (GMLS)

- *Ingredients 1+2:* Solves #4

Ingredient #1: Geometric Multivariate KRR

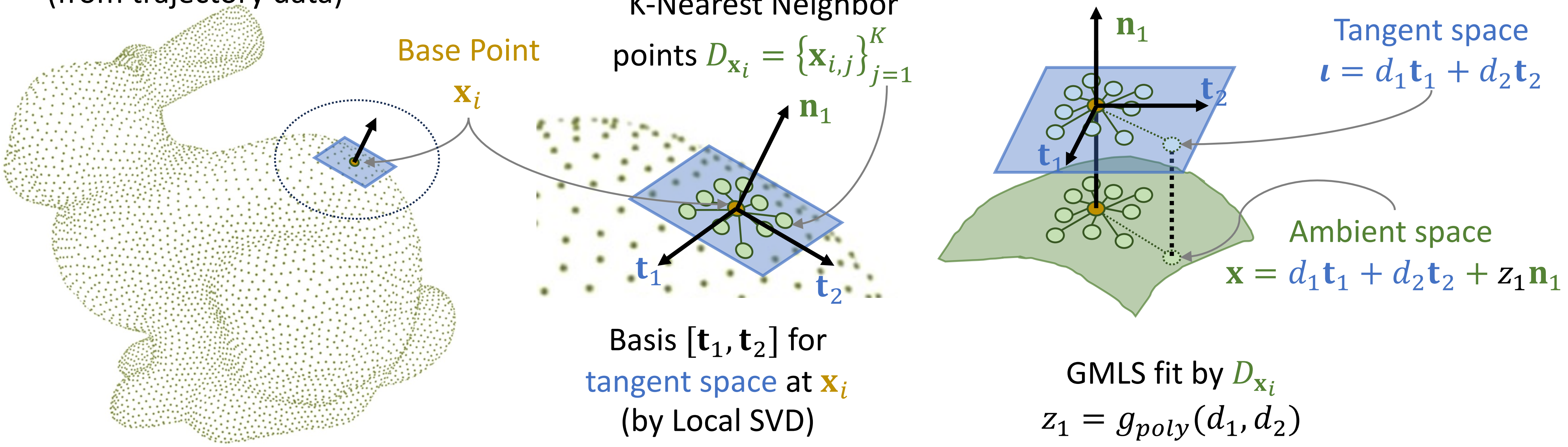


Ingredient #1: Geometric Multivariate KRR



Ingredient #2: Normal Correction

① Point cloud on manifold (from trajectory data)   ② Form local structure   ③ Normal correction by polynomial fit



Summary of algorithm

Theory: $\mathbf{x}_{i+1} = \exp_{\mathbf{x}_i}(\mathbf{f}(\mathbf{x}_i)\Delta t) = \mathbf{x}_i + \underbrace{\mathbf{T}_i\mathbf{T}_i^\top(\mathbf{x}_{i+1} - \mathbf{x}_i)}_{\text{Tangent: } \mathbf{f}(\mathbf{x}_i)\Delta t} + \underbrace{\mathbf{N}_i\mathbf{N}_i^\top(\mathbf{x}_{i+1} - \mathbf{x}_i)}_{\text{Normal: } O((\Delta t)^2)}$

Algorithm: $\hat{\mathbf{x}}_{\epsilon,i+1} = \hat{\mathbf{x}}_{\epsilon,i} + \Delta t \mathbf{f}_\epsilon(\hat{\mathbf{x}}_{\epsilon,i}) + (\Delta t)^2 \hat{\mathbf{N}}_{\epsilon,i}\hat{g}(\hat{\mathbf{T}}_{\epsilon,i}^\top \mathbf{f}_\epsilon(\hat{\mathbf{x}}_{\epsilon,i})))$

Geometric Multivariate Kernel Ridge Regression

Normal Correction by Generalized Moving Least Squares

Convergence: $\sqrt{\mathbb{E}_\mu\left[\|\mathbf{x}_n - \hat{\mathbf{x}}_{\epsilon,n}\|^2\right]} \leq C\left(\epsilon + \Delta t \left(\frac{\log(N)}{N}\right)^{\frac{\ell}{d}}\right)$

$\frac{\ell}{d}$: Manifold smoothness / Intrinsic dimension

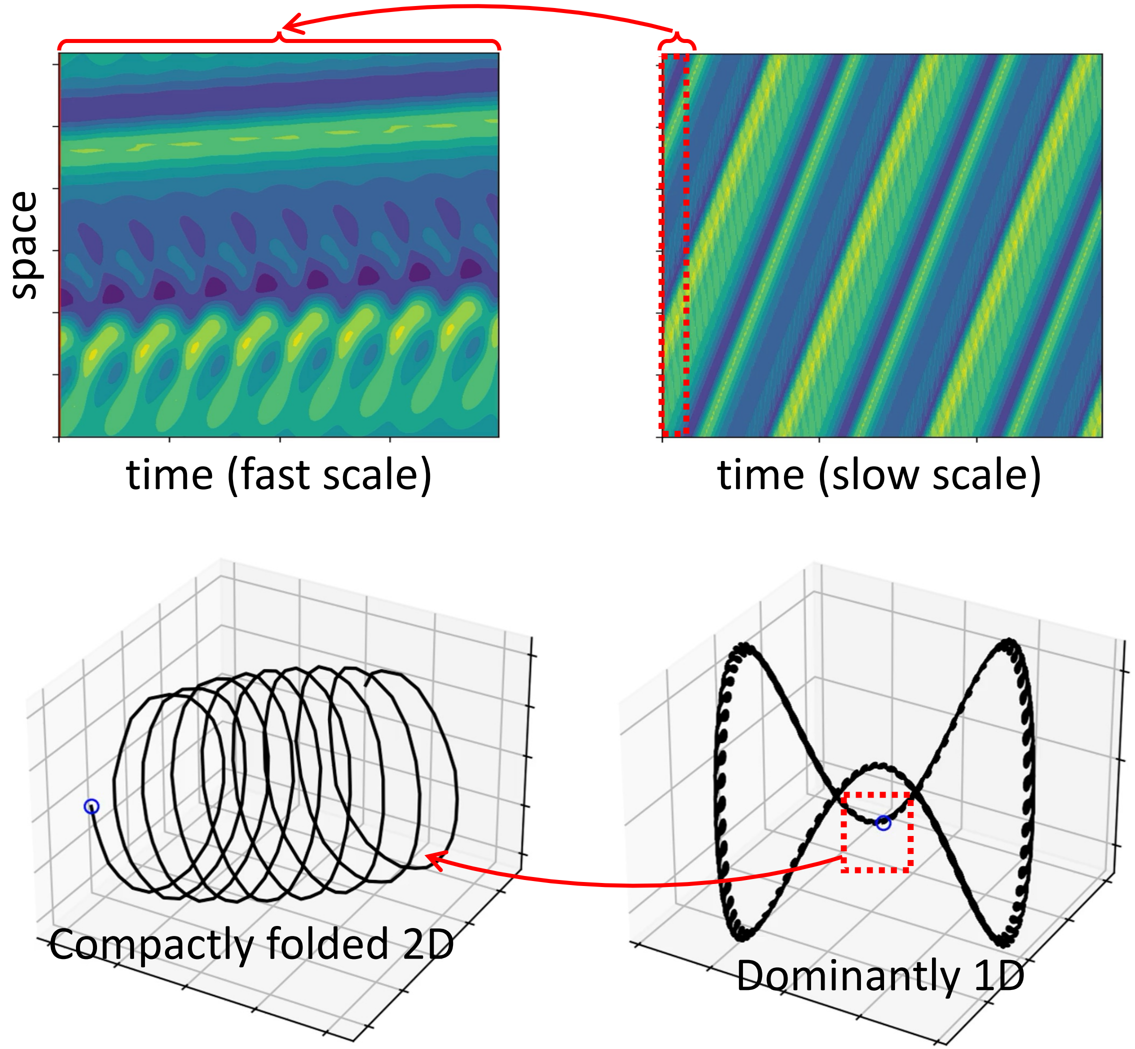
Expected error at $t = n\Delta t$

Error of learned vector field

Step size (first-order accurate)

# of data points

Numerical demonstrations



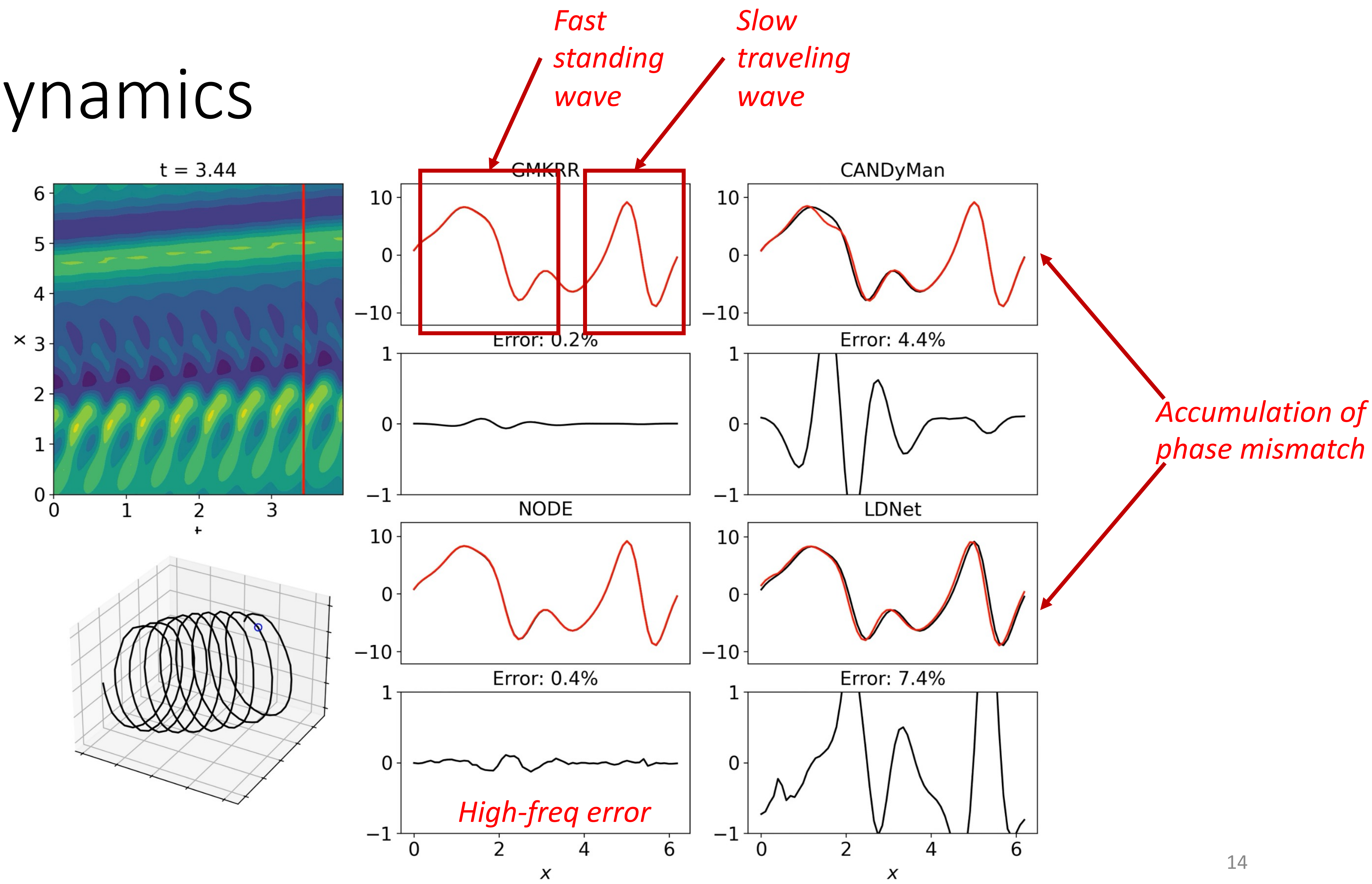
Example: 1D Kuramoto-Sivashinsky

$$u_t + uu_x + u_{xx} + \nu u_{xxxx} = 0$$

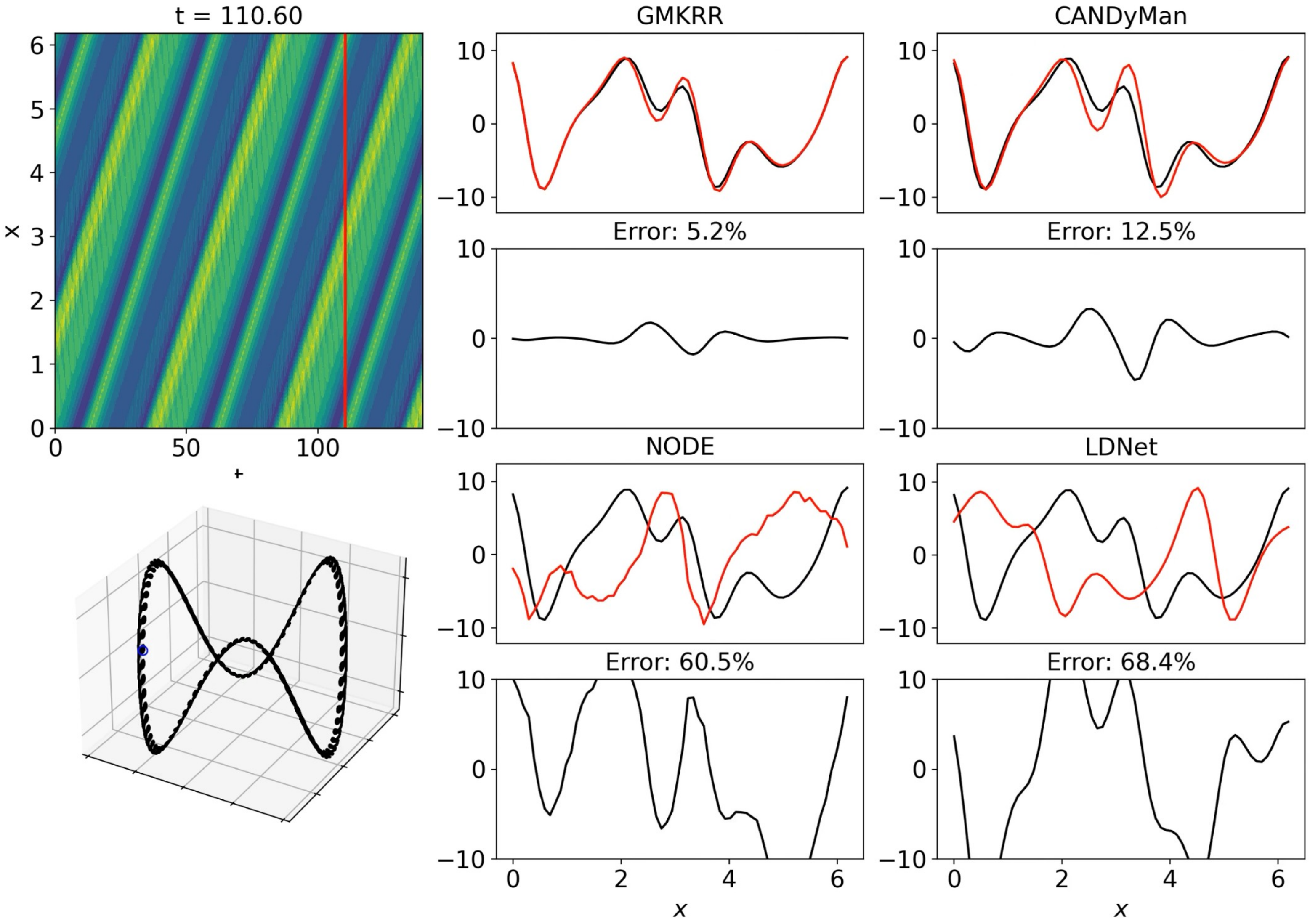
- Inertia manifold with $\nu = \frac{4}{87}$ and periodic BC
- Time scale disparity: fast and slow

- Intrinsic dimension: 2
- Ambient dimension: 64

Fast Dynamics



Slow Dynamics



Loss of fine structure at fast time scale

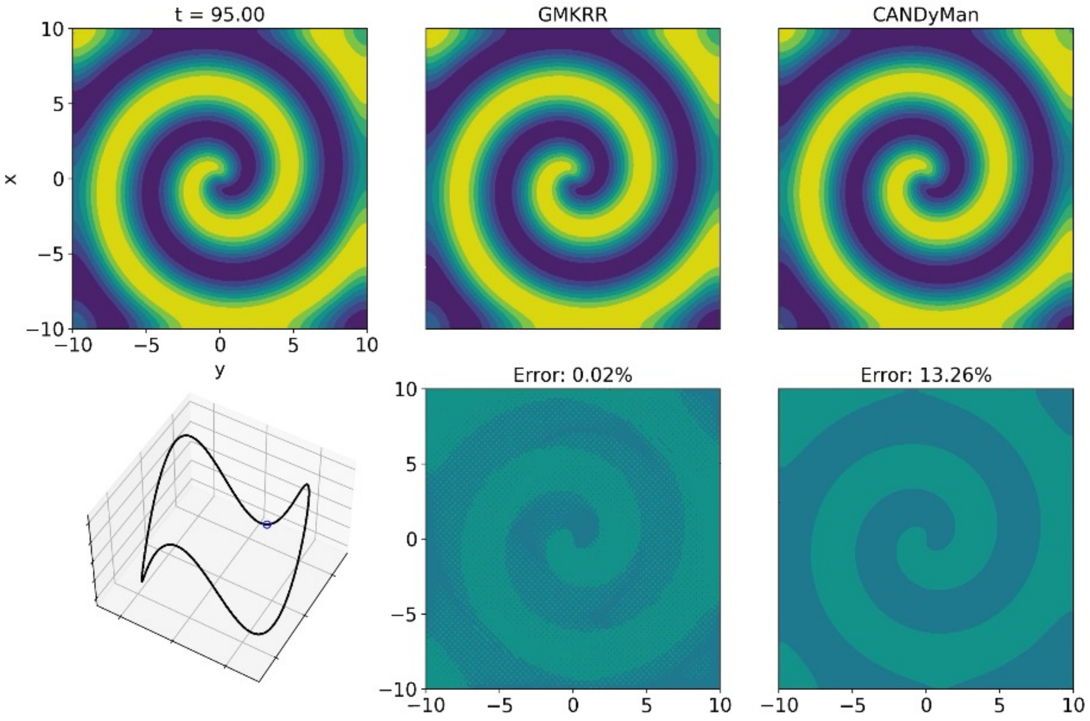
Complete loss of phase and shape

Summary

#1 Optimal compression – up to intrinsic dimension
#2 Constrained dynamics – by operator-valued kernel
#3 Time integrator – preserving geometry
#4 Efficient training – no NN's at all

CASE	MODEL	Training Time (s)	Pred. Time (s)	RMSE	Maximum Error
KS Beating	CANDyMan	48.04	0.6703	0.2744	2.130
	GMKRR	**0.01367**	**0.5133**	**0.02549**	**0.1972**
KS Travelling	CANDyMan	70.10	19.44	2.391	15.22
	NODE	395.6	**4.166**	6.690	22.14
	LDNet	77.87	13.53	7.600	22.46
	GMKRR-Full	11.68	20.82	1.604	12.45
	GMKRR-FFT	**0.04550**	24.41	**0.3671**	**3.088**
Reaction-Diffusion	CANDyMan	265.8	14.59	0.1797	0.5517
	GMKRR-Full	3.723	175.4	7.394E-4	2.959E-3
	GMKRR-PCA-T0	**0.01953**	**4.786**	7.399E-4	2.961E-3
	GMKRR-PCA-T4	0.1187	8.5855	**1.216E-4**	**1.339E-3**

Quantitative comparison



Reaction-diffusion
Ambient dim.: 20402

Learning vector fields of Differential equations on manifolds with Geometrically constrained operator-valued kernels

Daning Huang[1], Hanyang He[2], John Harlim[3], Yan Li[2]

Paper and Code