Some time, some space, and some equations: Machine-learning of model error in dynamical systems

Matthew E. Levine[1] Andrew M. Stuart[1]

[1]Department of Computing and Mathematical Sciences
California Institute of Technology

Third Symposium on Machine Learning and Dynamical Systems
Fields Institute, University of Toronto
September 29, 2022

Caltech

Introduction

- Machine learning works (with enough data)!

Introduction

- Machine learning works (with enough data)!
- Mechanistic models based on physics work (with enough knowledge and compute)!

Caltech

Introduction

- Machine learning works (with enough data)!
- Mechanistic models based on physics work (with enough knowledge and compute)!
- In most open prediction problems, we have SOME data and SOME prior knowledge.

Caltech

Introduction

- Machine learning works (with enough data)!
- Mechanistic models based on physics work (with enough knowledge and compute)!
- In most open prediction problems, we have SOME data and SOME prior knowledge.
- The next generation of high-performing prediction models will **hybridize physics-based and data-driven modeling techniques**
- How can we help lay the groundwork for this future?

Caltech

Our problem

True system (ODE):

$$\begin{aligned}\dot{x} &= f^{\dagger}(x, y)\\ \dot{y} &= \frac{1}{\varepsilon} g^{\dagger}(x, y)\end{aligned} \tag{1}$$

- **Relevance:** across disciplines (climatology, physiology, celestial mechanics, etc.).

Caltech

Our problem

True system (ODE):

$$\begin{aligned} \dot{x} &= f^\dagger(x, y) \\ \dot{y} &= \frac{1}{\varepsilon} g^\dagger(x, y) \end{aligned} \tag{1}$$

- **Relevance:** across disciplines (climatology, physiology, celestial mechanics, etc.).
- **Goal:** Given noisy observations of x, learn predictive model for future x dynamics.

Caltech

Our problem

True system (ODE):

$$\dot{x} = f^{\dagger}(x, y)$$
$$\dot{y} = \frac{1}{\varepsilon} g^{\dagger}(x, y) \tag{1}$$

- **Relevance:** across disciplines (climatology, physiology, celestial mechanics, etc.).
- **Goal:** Given noisy observations of x, learn predictive model for future x dynamics.
- **Methodological constraints:**
    - Partial, noisy observations (e.g. observe x, but not y)

Caltech

Our problem

True system (ODE):

$$\dot{x} = f^\dagger(x, y)$$
$$\dot{y} = \frac{1}{\varepsilon} g^\dagger(x, y) \tag{1}$$

- **Relevance:** across disciplines (climatology, physiology, celestial mechanics, etc.).
- **Goal:** Given noisy observations of x, learn predictive model for future x dynamics.
- **Methodological constraints:**
  - Partial, noisy observations (e.g. observe x, but not y)
  - No knowledge of y, $g^\dagger, \varepsilon$, nor $\dim(y)$

Caltech

Our problem

True system (ODE):

$$
\begin{aligned}
\dot{x} &= f^{\dagger}(x, y) \\
\dot{y} &= \frac{1}{\varepsilon} g^{\dagger}(x, y)
\end{aligned}
\tag{1}
$$

- **Relevance:** across disciplines (climatology, physiology, celestial mechanics, etc.).
- **Goal:** Given noisy observations of x, learn predictive model for future x dynamics.
- **Methodological constraints:**
  - Partial, noisy observations (e.g. observe x, but not y)
  - No knowledge of y, $g^{\dagger}, \varepsilon$, nor $\dim(y)$
  - Observations may be irregularly spaced and noisy

Caltech

Our problem

True system (ODE):

$$\begin{aligned} \dot{x} &= f^{\dagger}(x, y) \\ \dot{y} &= \frac{1}{\varepsilon} g^{\dagger}(x, y) \end{aligned} \tag{1}$$

- **Relevance:** across disciplines (climatology, physiology, celestial mechanics, etc.).
- **Goal:** Given noisy observations of x, learn predictive model for future x dynamics.
- **Methodological constraints:**
  - Partial, noisy observations (e.g. observe x, but not y)
  - No knowledge of y, $g^{\dagger}, \varepsilon$, nor $\dim(y)$
  - Observations may be irregularly spaced and noisy
  - Ability to leverage partial knowledge of $f^{\dagger}$

Caltech

Big picture: Learning dynamics using w/ partial observations $\{x_k\}_{k=0}^{K}$

	Introduce delays	Introduce augmented states
Discrete-time	$x_{k+1} = \Psi(x_k, x_{k-1}, \ldots, x_{k-\kappa}), \quad \kappa \in \mathbb{Z}^+$ Takens • $\Psi \in$ {NNs, GPs, Polynomials, RFs,…}	$x_{k+1} = \Psi_1(x_k, r_k)$ $r_{k+1} = \Psi_2(x_k, r_k), \quad r_k \in \mathbb{R}^d$ Tries to represent the missing states! • RNNs / Reservoir Computers • $\Psi_1, \Psi_2 \in$ {NNs, GPs, Polynomials, RFs,…}
Continuous-time	$\dot{x} = f\left(\{x(t-s)\}_{s=\tau}^{t}\right)$ Takens / Mori-Zwanzig • $f \in$ {NNs, GPs, Polynomials, RFs,…}	$\dot{x} = f_1(x, r)$ $\dot{r} = f_2(x, r), \quad r \in \mathbb{R}^d$ Tries to represent the missing states! • "Continuous-time" RNNs • "Continuous-time" Reservoir Computers • $f_1, f_2 \in$ {NNs, GPs, Polynomials, RFs,…}

Big picture: Learning dynamics using w/ partial observations $\{x_k\}_{k=0}^{K}$

	Introduce delays	Introduce augmented states
Discrete-time	$x_{k+1} = \Psi(x_k, x_{k-1}, \ldots, x_{k-\kappa}), \quad \kappa \in \mathbb{Z}^+$ Takens • Ψ ∈ {NNs, GPs, Polynomials, RFs,…}	$x_{k+1} = \Psi_1(x_k, r_k)$ $r_{k+1} = \Psi_2(x_k, r_k), \quad r_k \in \mathbb{R}^d$ Tries to represent the missing states! • RNNs / Reservoir Computers • Ψ_1, Ψ_2 ∈ {NNs, GPs, Polynomials, RFs,…}
Continuous-time	$\dot{x} = f\left(\{x(t-s)\}_{s=\tau}^{t}\right)$ Takens / Mori-Zwanzig • f ∈ {NNs, GPs, Polynomials, RFs,…}	$\dot{x} = f_1(x, r)$ $\dot{r} = f_2(x, r), \quad r \in \mathbb{R}^d$ Tries to represent the missing states! • "Continuous-time" RNNs • "Continuous-time" Reservoir Computers • f_1, f_2 ∈ {NNs, GPs, Polynomials, RFs,…}

Big picture: Learning dynamics using w/ partial observations $\{x_k\}_{k=0}^K$

	Introduce delays	Introduce augmented states
Discrete-time	$x_{k+1} = \Psi(x_k, x_{k-1}, \ldots, x_{k-\kappa}), \quad \kappa \in \mathbb{Z}^+$ Takens • $\Psi \in$ {NNs, GPs, Polynomials, RFs,…}	$x_{k+1} = \Psi_1(x_k, r_k)$ $r_{k+1} = \Psi_2(x_k, r_k), \quad r_k \in \mathbb{R}^d$ Tries to represent the missing states! • RNNs / Reservoir Computers • $\Psi_1, \Psi_2 \in$ {NNs, GPs, Polynomials, RFs,…}
Continuous-time	$\dot{x} = f\left(\{x(t-s)\}_{s=\tau}^{t}\right)$ Takens / Mori-Zwanzig • $f \in$ {NNs, GPs, Polynomials, RFs,…}	$\dot{x} = f_1(x, r)$ $\dot{r} = f_2(x, r), \quad r \in \mathbb{R}^d$ Tries to represent the missing states! • "Continuous-time" RNNs • "Continuous-time" Reservoir Computers • $f_1, f_2 \in$ {NNs, GPs, Polynomials, RFs,…}

Big picture: Learning dynamics using w/ partial observations $\{x_k\}_{k=0}^{K}$

	Introduce delays	Introduce augmented states
Discrete-time	$x_{k+1} = \Psi(x_k, x_{k-1}, \ldots, x_{k-\kappa}), \quad \kappa \in \mathbb{Z}^+$ Takens • $\Psi \in$ {NNs, GPs, Polynomials, RFs,…}	$x_{k+1} = \Psi_1(x_k, r_k)$ $r_{k+1} = \Psi_2(x_k, r_k), \quad r_k \in \mathbb{R}^d$ Tries to represent the missing states! • RNNs / Reservoir Computers • $\Psi_1, \Psi_2 \in$ {NNs, GPs, Polynomials, RFs,…}
Continuous-time	$\dot{x} = f\Big(\{x(t-s)\}_{s=\tau}^{t}\Big)$ Takens / Mori-Zwanzig • $f \in$ {NNs, GPs, Polynomials, RFs,…}	$\dot{x} = f_1(x, r)$ $\dot{r} = f_2(x, r), \quad r \in \mathbb{R}^d$ Tries to represent the missing states! • “Continuous-time” RNNs • “Continuous-time” Reservoir Computers • $f_1, f_2 \in$ {NNs, GPs, Polynomials, RFs,…}

Big picture: Learning dynamics using w/ partial observations $\{x_k\}_{k=0}^{K}$

	Introduce delays	Introduce augmented states
Discrete-time	$x_{k+1} = \Psi(x_k, x_{k-1}, \ldots, x_{k-\kappa}), \quad \kappa \in \mathbb{Z}^+$ Takens • $\Psi \in$ {NNs, GPs, Polynomials, RFs,…}	$x_{k+1} = \Psi_1(x_k, r_k)$ $r_{k+1} = \Psi_2(x_k, r_k), \quad r_k \in \mathbb{R}^d$ Tries to represent the missing states! • RNNs / Reservoir Computers • $\Psi_1, \Psi_2 \in$ {NNs, GPs, Polynomials, RFs,…}
Continuous-time	$\dot{x} = f\Big(\{x(t-s)\}_{s=\tau}^{t}\Big)$ Takens / Mori-Zwanzig • $f \in$ {NNs, GPs, Polynomials, RFs,…}	$\dot{x} = f_1(x, r)$ $\dot{r} = f_2(x, r), \quad r \in \mathbb{R}^d$ Tries to represent the missing states! • “Continuous-time” RNNs • “Continuous-time” Reservoir Computers • $f_1, f_2 \in$ {NNs, GPs, Polynomials, RFs,…}

Big picture: Learning dynamics using w/ partial observations $\{x_k\}_{k=0}^{K}$

	Introduce delays	Introduce augmented states
Discrete-time	$x_{k+1} = \Psi(x_k, x_{k-1}, \ldots, x_{k-\kappa}), \quad \kappa \in \mathbb{Z}^+$ Takens • $\Psi \in$ {NNs, GPs, Polynomials, RFs,…}	$x_{k+1} = \Psi_1(x_k, r_k)$ $r_{k+1} = \Psi_2(x_k, r_k), \quad r_k \in \mathbb{R}^d$ Tries to represent the missing states! • RNNs / Reservoir Computers • $\Psi_1, \Psi_2 \in$ {NNs, GPs, Polynomials, RFs,…}
Continuous-time	$\dot{x} = f\left(\{x(t-s)\}_{s=\tau}^{t}\right)$ Takens / Mori-Zwanzig • $f \in$ {NNs, GPs, Polynomials, RFs,…}	$\dot{x} = f_1(x, r)$ $\dot{r} = f_2(x, r), \quad r \in \mathbb{R}^d$ Tries to represent the missing states! • "Continuous-time" RNNs • "Continuous-time" Reservoir Computers • $f_1, f_2 \in$ {NNs, GPs, Polynomials, RFs,…}

Leveraging partial knowledge of the dynamics

For any f_0 (regardless of its fidelity), there exists an $m^\dagger(x, y)$ such that (1) can be re-written as

$$\dot{x} = f_0(x) + m^\dagger(x, y) \tag{2a}$$

$$\dot{y} = \frac{1}{\varepsilon} g^\dagger(x, y). \tag{2b}$$

Qianxiao Li, MLDS 2022 slides!!

- Approximation Rates. Let $\mathcal{H} = \cup_m \mathcal{H}_m$, where $\mathcal{H}_m \subset \mathcal{H}_{m+1}$, m measures size of hypothesis space (approximation budget)

$$\boxed{\inf_{\hat{F} \in \mathcal{H}_m} \|F^* - \hat{F}\| \leq \text{Complexity}(F^*)\text{rate}(m), \qquad \text{rate}(m) \to 0.}$$

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \dot{u} = f(u;\ \theta), \quad u &= [x, r]^T \\ Hu &= x \end{aligned}$$

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \qquad \Longleftrightarrow \qquad \begin{aligned} \dot{u} = f(u;\ \theta), \quad u = [x, r]^T \\ Hu = x \end{aligned}$$

Assume noisy observations $z = Hu + \eta$.

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \dot{u} = f(u;\ \theta), \quad u = [x, r]^T \\ Hu = x \end{aligned}$$

Assume noisy observations $z = Hu + \eta$.

Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \dot{u} = f(u;\ \theta), \quad u = [x, r]^T \\ Hu = x \end{aligned}$$

Assume noisy observations $z = Hu + \eta$.
Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Hard Constraint Idea 1: Infer init. cond. and parameters (Rubanova *et al.* 2019)

$$\operatorname*{argmin}_{\theta, u_0} \int_0^T \|z(t) - Hu(t; u_0, \theta)\|^2 dt.$$

- ✗ Poorly-posed with larger T for chaotic systems with sensitivity to u_0.

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \quad \iff \quad \begin{aligned} \dot{u} = f(u;\ \theta), \quad u &= [x, r]^T \\ Hu &= x \end{aligned}$$

Assume noisy observations $z = Hu + \eta$.
Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Hard Constraint Idea 2: Break data into chunks to cope with sensitivities

$$\underset{\theta, \{u_0^{(k)}\}_{k=1}^K}{\operatorname{argmin}} \sum_{k=1}^{K} \int_0^T \|z^{(k)}(t) - Hu(t; u_0^{(k)}, \theta)\|^2 dt.$$

- ✗ Dimensionality of inference grows with T.
- ✗ When chunks are small, can overfit by "cherry-picking" $u_0^{(k)}$.

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \quad \iff \quad \begin{aligned} \dot{u} = f(u;\ \theta), \quad u = [x, r]^T \\ Hu = x \end{aligned}$$

Assume noisy observations $z = Hu + \eta$.
Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Data Assimilation for inferring missing dynamics:

$$\underset{\theta, \{u_0^{(k)}\}_{k=1}^K}{\operatorname{argmin}} \sum_{k=1}^{K} \int_0^T \|z^{(k)}(t) - Hu(t; u_0^{(k)}, \theta)\|^2 dt.$$

- Initial conditions $u_0^{(k)}$ can be inferred using a sequence of warmup data (and assumption on θ) using standard *Data Assimilation* techniques.

Caltech

Learning latent dynamics in continuous-time

$$\dot{x} = f_0(x) + m(x, r;\ \theta) \qquad \Longleftrightarrow \qquad \dot{u} = f(u;\ \theta), \quad u = [x, r]^T$$
$$\dot{r} = g(x, r;\ \theta) \qquad\qquad\qquad Hu = x$$

Assume noisy observations $z = Hu + \eta$.

Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Let $\hat{m}(t, \tau, \theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}})$ be an estimate of $u(t) \ \mid \{z(t-s)\}_{s=0}^{\tau},\ \theta_{\mathrm{DYN}},\ u(t-\tau) = 0$.

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned} \dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta) \end{aligned} \qquad \iff \qquad \begin{aligned} \dot{u} = f(u;\ \theta), &\quad u = [x, r]^T \\ Hu &= x \end{aligned}$$

Assume noisy observations $z = Hu + \eta$.
Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Let $\hat{m}(t, \tau, \theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}})$ be an estimate of $u(t)\ \mid \{z(t-s)\}_{s=0}^{\tau},\ \theta_{\mathrm{DYN}},\ u(t-\tau) = 0.$

DA-based inference: Initial conditions can be estimated jointly with parameters

$$\underset{\theta_{\mathrm{DYN}},\ \theta_{\mathrm{DA}}}{\operatorname{argmin}} \sum_{k=1}^{K} \int_0^T \|z^{(k)}(t) - Hu\big(t; \hat{m}(t_k, \tau, \theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}}), \theta_{\mathrm{DYN}}\big)\|^2 dt.$$

Caltech

Learning latent dynamics in continuous-time

$$\begin{aligned}\dot{x} &= f_0(x) + m(x, r;\ \theta) \\ \dot{r} &= g(x, r;\ \theta)\end{aligned} \qquad \Longleftrightarrow \qquad \begin{aligned}\dot{u} = f(u;\ \theta), &\quad u = [x, r]^T \\ Hu &= x\end{aligned}$$

Assume noisy observations $z = Hu + \eta$.
Let $u(t; v, \theta)$ solve $\dot{u} = f(u; \theta),\ u(0) = v$.

Let $\hat{m}(t, \tau, \theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}})$ be an estimate of $u(t) \ \mid \{z(t-s)\}_{s=0}^{\tau},\ \theta_{\mathrm{DYN}},\ u(t-\tau) = 0$.

DA-based inference: Initial conditions can be estimated jointly with parameters

$$\underset{\theta_{\mathrm{DYN}},\ \theta_{\mathrm{DA}}}{\operatorname{argmin}} \sum_{k=1}^{K} \int_0^T \|z^{(k)}(t) - Hu\big(t; \hat{m}(t_k, \tau, \theta_{\mathrm{DYN}}, \theta_{\mathrm{DA}}), \theta_{\mathrm{DYN}}\big)\|^2 dt.$$

- Here, we perform joint estimation with auto-differentiable 3DVAR
- Chen *et al.* 2021 perform joint estimation with auto-differentiable Ensemble Kalman Filter
- Carassi *et al.* 2021 apply alternating descent (EnKF for $\hat{m}$, supervised SGD for θ)

Caltech

Let $\Psi(v;\theta) := u(\Delta t; v, \theta)$ denote a integrator of our RHS that maps us to the next data element

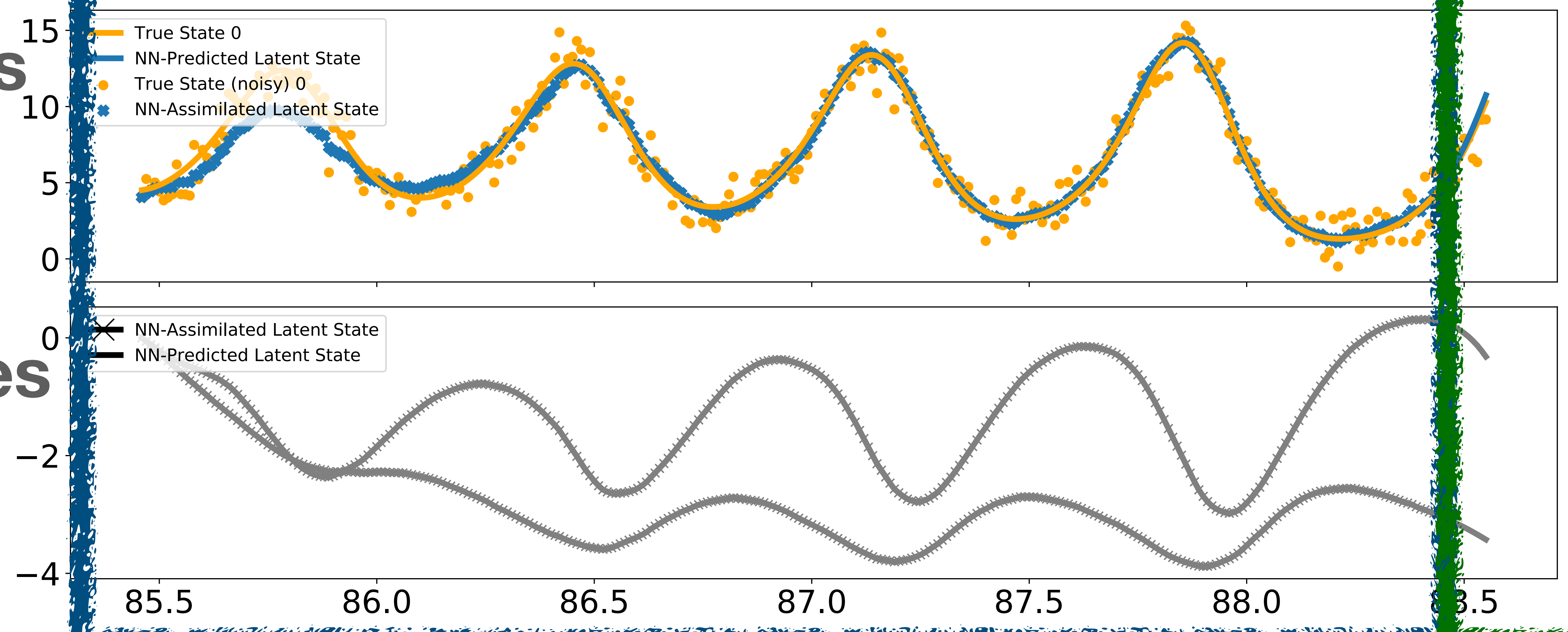
Auto-differentiable WARMUP via Data Assimilation **Auto-differentiable FORECAST**

Observed states

$x_1(t), z(t)$

Unobserved states

$x_2(t), x_3(t)$



$\Psi(\hat{u}_0;\theta)$

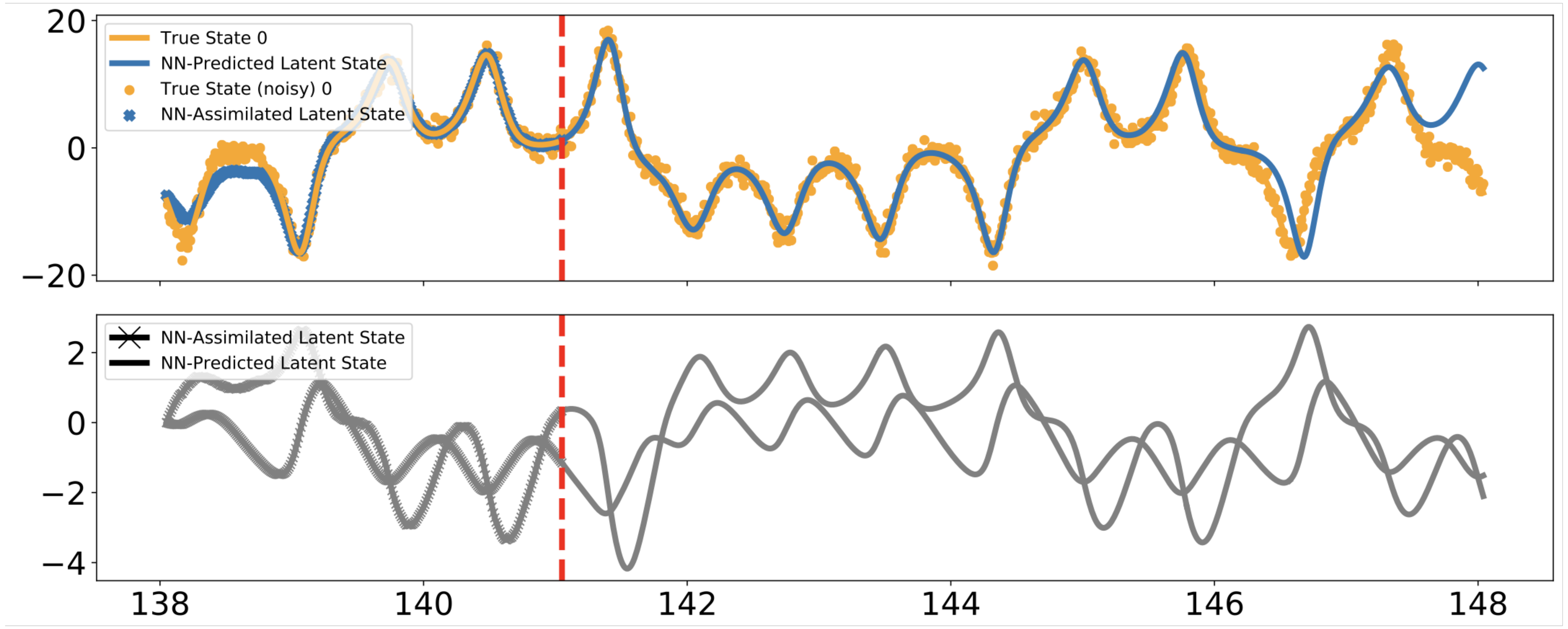
$z_{-\tau}, \quad \cdots \quad , z_0$

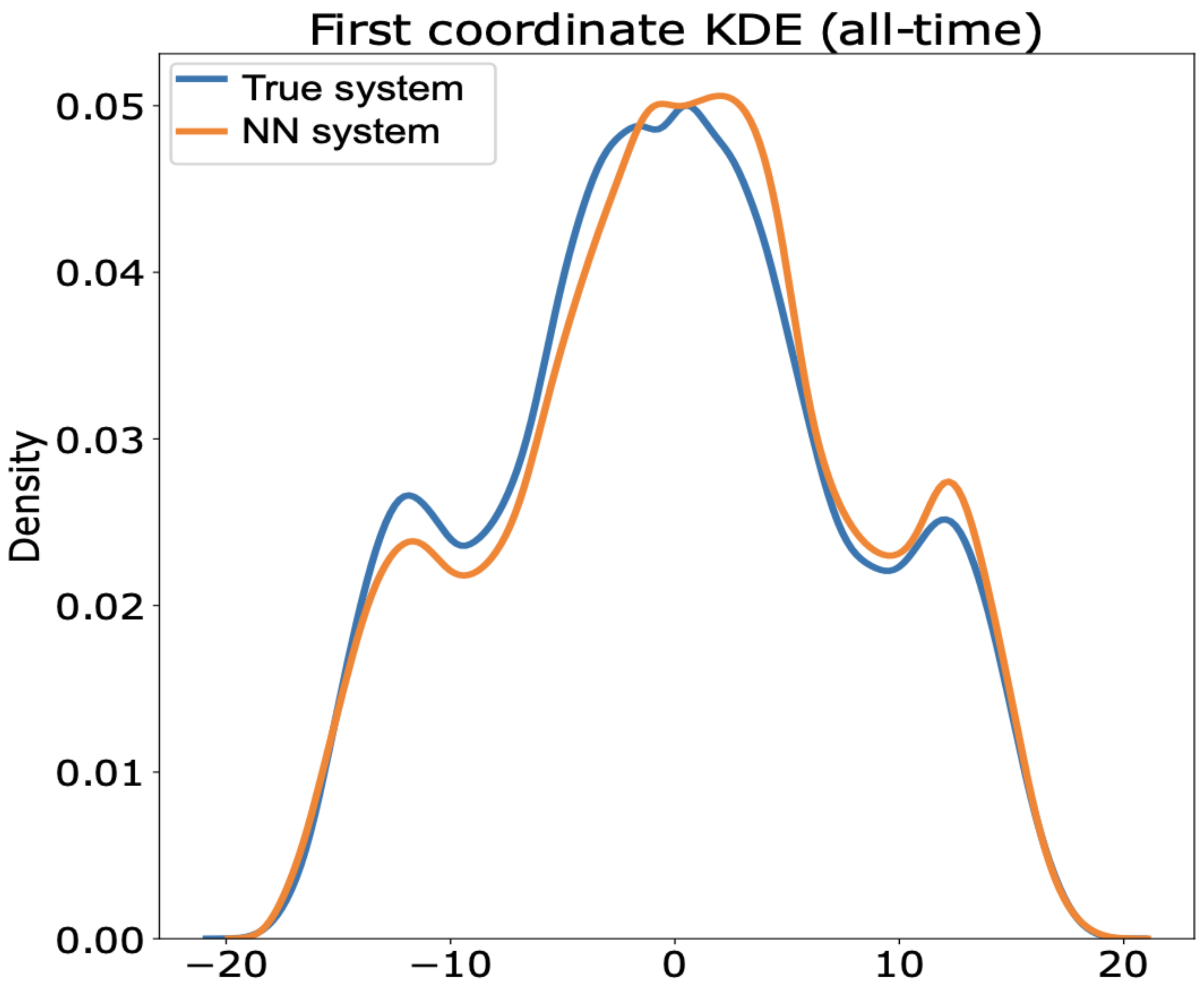
$\hat{u}_{-\tau} \longrightarrow \hat{u}_0$

$$\hat{u}_{k+1} = \Psi(\hat{u}_k;\Delta t) + K\Big(z_{k+1} - H\Psi(\hat{u}_k;\Delta t)\Big)$$

Sequential updates w/ constant gain

Lorenz ’63 with partial, noisy observations—noisily observe only the first component, and model its dynamics using our augmented-state model.

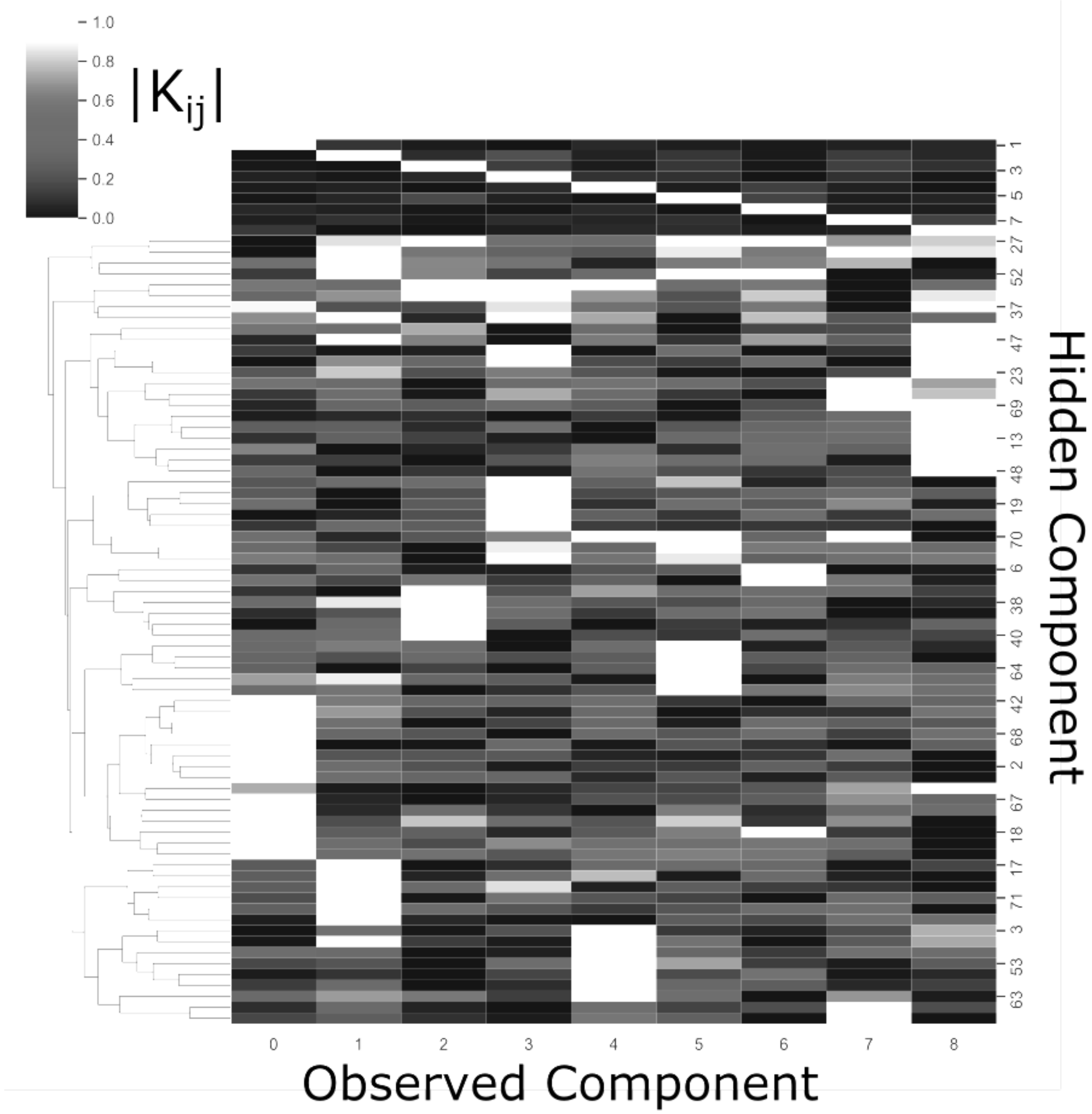




Caltech

Learning L96MS memory-based closure

- The true L96MS system has a clustered subgrouping of fast variables—our model has re-discovered this structure, and the DA gain K has learnt to exploit these correlations for improved filtering.



Caltech

Reservoir computing with connections to random features

$$x_{k+1} = Cr_{k+1} \tag{5a}$$

$$r_{k+1} = \sigma(W_1 r_k + W_2 x_k + b) \tag{5b}$$

- Randomize and fix (W_1, W_2, b, r_0).
- Given $\{x_k\}_{k=0}^K, r_0, W_1, W_2, b$, we can determine $\{r_k\}_{k=1}^K$
- This is great, because now we just need to do a linear regression! $C : r_k \mapsto x_k$.

Debate: For a fully observed system, what choice of W_1 gives the best RC?

Caltech

Reservoir computing with connections to random features

$$x_{k+1} = Cr_{k+1} \tag{5a}$$

$$r_{k+1} = \sigma(W_1 r_k + W_2 x_k + b) \tag{5b}$$

- Randomize and fix (W_1, W_2, b, r_0).
- Given $\{x_k\}_{k=0}^K, r_0, W_1, W_2, b$, we can determine $\{r_k\}_{k=1}^K$
- This is great, because now we just need to do a linear regression! $C: r_k \mapsto x_k$.

Debate: For a fully observed system, what choice of W_1 gives the best RC?

My answer: Easy, just $W_1 = 0$. Then $r_{k+1} = \sigma(W_2 x_k + b)$, and

$$x_{k+1} = \sum_j C^{(j)} \sigma(W_2^{(j)} x_k + b^{(j)}), \qquad \text{a random feature model!}$$

For example, choosing $\sigma := \cos(\cdot)$, $W_2^{(j)} \sim \mathcal{N}(0, \Sigma)$, and $b^{(j)} \sim \mathcal{U}[-2\pi, 2\pi]$, approximates Gaussian Process with RBF kernel in large feature limit.

Caltech

Empirical evaluations of learnt memory

$$x_{k+1} = Cr_{k+1} \tag{6a}$$

$$r_{k+1} = \sigma(W_1 r_k + W_2 x_k + b) \tag{6b}$$

- Often we apply RCs to partially-observed systems, which have substantial Markovian properties on the observables.
- Do you ever worry that your RC performance is not *really* learning *memory*?
- Easy sanity check: Set $W_1 = 0$ and re-tune hyperparameters.
- The difference in performance between this and our RC represents the amount of memory that the RC has accounted for.

Caltech

Expressivity of RCs/RNNs: latent dims vs function complexity

$$x_{k+1} = C r_{k+1}$$
$$r_{k+1} = \sigma(W_1 r_k + W_2 x_k + b)$$

Versus

$$x_{k+1} = C\sigma(A r_{k+1} + a)$$
$$r_{k+1} = B\sigma(W_1 r_k + W_2 x_k + b)$$

- # latent variables $\equiv \dim(r)$
- Expressivity $\propto \dim(r)$
- Approximator limit leads to $\dim(r) = \infty$, even for a finite dimensional system!

- # latent variables $\equiv \dim(r)$
- Expressivity $\propto \dim(A, W_1, W_2)$
- Decouples dimension from expressivity; Can use infinitely many parameters in a finite dimensional space.

Caltech

Expressivity of RCs/RNNs: latent dims vs function complexity

$$x_{k+1} = Cr_{k+1}$$
$$r_{k+1} = \sigma(W_1 r_k + W_2 x_k + b)$$

Versus

$$x_{k+1} = C\sigma(Ar_{k+1} + a)$$
$$r_{k+1} = B\sigma(W_1 r_k + W_2 x_k + b)$$

- $\#$ latent variables $\equiv \dim(r)$
- Expressivity $\propto \dim(r)$
- Approximator limit leads to $\dim(r) = \infty$, even for a finite dimensional system!

- $\#$ latent variables $\equiv \dim(r)$
- Expressivity $\propto \dim(A, W_1, W_2)$
- Decouples dimension from expressivity; Can use infinitely many parameters in a finite dimensional space.

Caltech

Continuous-time RCs and Random Features on Banach Spaces

$$\dot{x} = Cr(t) \tag{7a}$$

$$\dot{r} = \sigma(W_1 r + W_2 x + b) \tag{7b}$$

Question: How to get RCs to work well in continuous time? What distributions should we choose for W_1, W_2?

My answer: Shrug? I haven't gotten it to work well. I would love to hear from you all on this!

Caltech

Conclusions

- We can learn ODEs from partially observed, noisy data by embedding state-estimation techniques within the optimization
  - Can be used for tuning DA parameters
  - Can move this to a derivative-free optimization (if your models are too huge to differentiate through)
  - Currently using for modeling endocrine dynamics in patients with diabetes (joint work with Emily Fox)
- Reservoir Computers and random feature methods are quite connected
  - With $W_r = 0$, RC approximates a markovian GP.
  - With $W_r \neq 0$, something deeper is going on! If you have ideas on this, come see me! We (Ollie Dunbar, Nick Nelsen, and I) are organizing a small ICIAM minisymposium around this topic.
- I'll graduate in 2023 and need a job!
  - I want to deploy this work in biomedical applications and improve the methods until they work with real data and solve real problems!

Caltech

Conclusions

- We can learn ODEs from partially observed, noisy data by embedding state-estimation techniques within the optimization
  - Can be used for tuning DA parameters
  - Can move this to a derivative-free optimization (if your models are too huge to differentiate through)
  - Currently using for modeling endocrine dynamics in patients with diabetes (joint work with Emily Fox)
- Reservoir Computers and random feature methods are quite connected
  - With $W_r = 0$, RC approximates a markovian GP.
  - With $W_r \neq 0$, something deeper is going on! If you have ideas on this, come see me! We (Ollie Dunbar, Nick Nelsen, and I) are organizing a small ICIAM minisymposium around this topic.
- I'll graduate in 2023 and need a job!
  - I want to deploy this work in biomedical applications and improve the methods until they work with real data and solve real problems!

Caltech

Conclusions

- We can learn ODEs from partially observed, noisy data by embedding state-estimation techniques within the optimization
  - Can be used for tuning DA parameters
  - Can move this to a derivative-free optimization (if your models are too huge to differentiate through)
  - Currently using for modeling endocrine dynamics in patients with diabetes (joint work with Emily Fox)
- Reservoir Computers and random feature methods are quite connected
  - With $W_r = 0$, RC approximates a markovian GP.
  - With $W_r \neq 0$, something deeper is going on! If you have ideas on this, come see me! We (Ollie Dunbar, Nick Nelsen, and I) are organizing a small ICIAM minisymposium around this topic.
- I'll graduate in 2023 and need a job!
  - I want to deploy this work in biomedical applications and improve the methods until they work with real data and solve real problems!

Caltech

Related Work

- Kaheman, Kadierdan, Eurika Kaiser, Benjamin Strom, J. Nathan Kutz, and Steven L. Brunton. "Learning Discrepancy Models From Experimental Data." ArXiv:1909.08574 [Cs, Eess, Stat], September 18, 2019. http://arxiv.org/abs/1909.08574.
- Tipireddy, Ramakrishna, Paris Perdikaris, Panos Stinis, and Alexandre Tartakovsky. "A Comparative Study of Physics-Informed Neural Network Models for Learning Unknown Dynamics and Constitutive Relations." ArXiv:1904.04058 [Physics], April 2, 2019. http://arxiv.org/abs/1904.04058.
- Lovelett, Robert J., Jose L. Avalos, and Ioannis G. Kevrekidis. "Partial Observations and Conservation Laws: Grey-Box Modeling in Biotechnology and Optogenetics." ArXiv:1909.04234 [Math], September 9, 2019. http://arxiv.org/abs/1909.04234.
- Pathak, Jaideep, Alexander Wikner, Rebeckah Fussell, Sarthak Chandra, Brian R. Hunt, Michelle Girvan, and Edward Ott. "Hybrid Forecasting of Chaotic Processes: Using Machine Learning in Conjunction with a Knowledge-Based Model." Chaos: An Interdisciplinary Journal of Nonlinear Science 28, no. 4 (April 1, 2018): 041101. https://doi.org/10.1063/1.5028373.

Related Work

- Rico-Martines, R., I. G. Kevrekidis, M. C. Kube, and J. L. Hudson. “Discrete- vs. Continuous-Time Nonlinear Signal Processing: Attractors, Transitions and Parallel Implementation Issues.” In 1993 American Control Conference, 1475–79. San Francisco, CA, USA: IEEE, 1993. https://doi.org/10.23919/ACC.1993.4793116.
- Chang, Bo, Minmin Chen, Eldad Haber, and Ed H. Chi. “AntisymmetricRNN: A Dynamical System View on Recurrent Neural Networks.” ArXiv:1902.09689 [Cs, Stat], February 25, 2019. http://arxiv.org/abs/1902.09689.
- Funahashi, Ken-ichi, and Yuichi Nakamura. “Approximation of Dynamical Systems by Continuous Time Recurrent Neural Networks.” Neural Networks 6, no. 6 (January 1, 1993): 801–6. https://doi.org/10.1016/S0893-6080(05)80125-X.
- Schäfer, Anton Maximilian. “RECURRENT NEURAL NETWORKS ARE UNIVERSAL APPROXIMATORS,” 2007, 11.
- Wan, Zhong Yi, Pantelis Vlachas, Petros Koumoutsakos, and Themistoklis Sapsis. ”Data-Assisted Reduced-Order Modeling of Extreme Events in Complex Dynamical Systems.” PLOS ONE 13, no. 5 (May 24, 2018): e0197704.

Caltech

Related Work

- Lei, Youming, Jian Hu, and Jianpeng Ding. "A Hybrid Model Based on Deep LSTM for Predicting High-Dimensional Chaotic Systems." ArXiv:2002.00799 [Cs, Eess], January 21, 2020. http://arxiv.org/abs/2002.00799.
- Sherstinsky, Alex. "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network." Physica D: Nonlinear Phenomena 404 (March 1, 2020): 132306. https://doi.org/10.1016/j.physd.2019.132306.

Caltech

Thank you!

- Many thanks to my adviser **Andrew Stuart** and the graduate students and postdocs of the Caltech CMS department.
- Many more thanks to **Boumediene** for the kind invitation and to all of the **MLDS organizers** and **members of the Fields Institute** for welcoming us!
- Thanks to the **participants and speakers** who have made this an enriching week!
- For more info, see:
  - **My website:** `mlevine@netlify.com`
  - **Our paper:** Levine and Stuart, A Framework for Machine Learning of Dynamical Systems, To appear in Communications of the AMS: Volume 2. 2022. `https://arxiv.org/abs/2107.06658`

Caltech