10th Festival de Thoérie, July 2019, Aix-en-Provence, France

MACHINE LEARNING FORECASTING OF CHAOS (INCLUDING THAT OF LARGE SPATIALLY DISTRIBUTED SYSTEMS)

Edward Ott
University of Maryland

Collaborators:
Jaideep Pathak, Zhixin Lu, Sarthak Chandra,
Alex Wikner, Rebbeckah Fussell,
Prof. Michelle Girvan, Prof. Brian Hunt

THE GENERAL MACHINE LEARNING (ML) SET UP FOR THIS TALK *

- **The box:** represents a complex nonlinear I/O relationship with a very large number of adjustable parameters.
- **Training:** Based on many examples of inputs and their corresponding desired outputs, adjust the parameters to best fit the desired output. (And it is hoped that these best fit outputs are indeed ***exceedingly*** close to the ideally desired outputs.)
- **Two Big Questions:** (1) Is training practically feasible, given the large number of parameters to be determined? (2) If similar inputs were continued after the training, would the desired outputs continue to be produced (i.e., does the trained ML device "***generalize***" from the specific training examples to the broad class from which the training examples were drawn)?
- **ML type:** What is in the box? It is important in choosing what is in the box that it should result in 'yes' answers to the above two questions.

*Other applicable terms: "statistical learning" and "supervised learning."

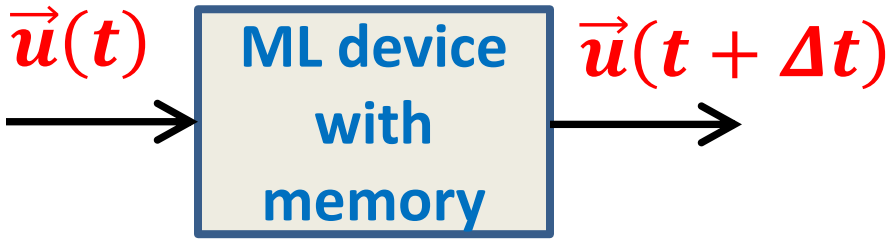
ML APPLIED TO DYNAMICS

DYNAMICAL SYSTEM: A system whose 'state' evolves in time and for which the future evolution of the system state depends only on the current state.

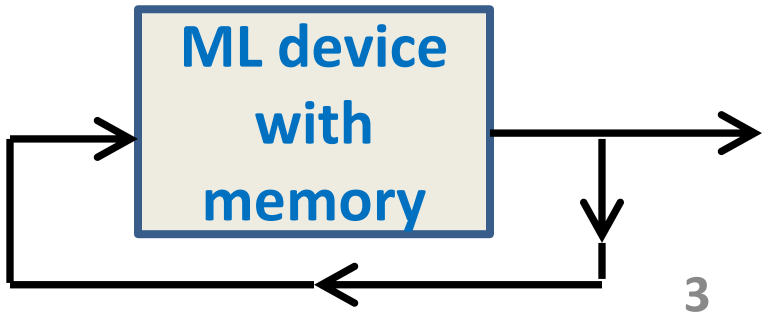
AN ILLUSTRATIVE MACHINE LEARNING TASK IN DYNAMICS:
Given a time series of past state dependent measurements from an evolving *unknown* dynamical system, predict the future evolution of those measurements.

Assuming the system is stationary in time [Jaeger & Haas, Science (2004)]:

Training



Prediction



Reservoir Computing*

In this talk we focus on a particular type of of machine learning called "reservoir computing." We note, however, that other types of machine learning (notably types of 'deep learning') could potentially be used for the problems we consider.

*** Original papers on reservoir computing:**
"Liquid State Machines" [W. Maas et al. (2002)]
"Echo State Networks" [H. Jaeger (2001)]

Outline

Using reservoir computing for model-free prediction from data.

Examples extending the above:

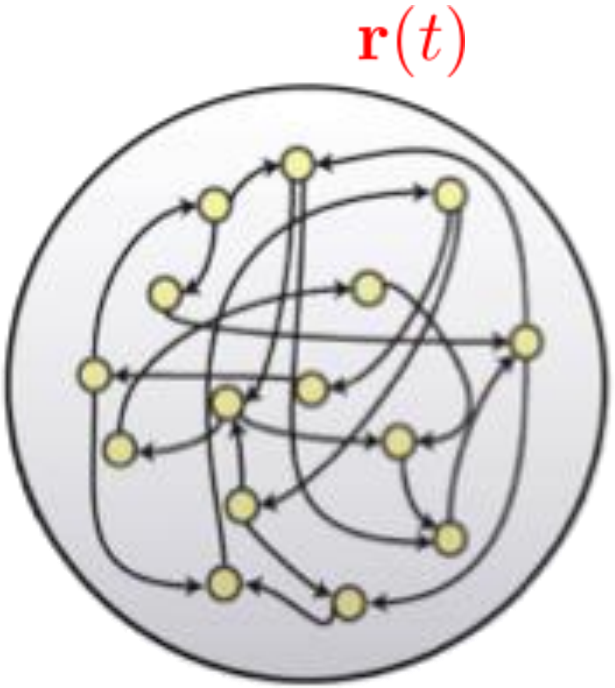
(a)Prediction of a spatiotemporally chaotic system
(b)Parallel approach to prediction of large systems*
(c) Hybrid knowledge-based/machine-learning approach**
(d)Combining the parallel and hybrid approaches***

Conclusion

*** J. Pathak, B. Hunt, M. Girvan Z. Lu and E. Ott, Phys. Rev. Lett. 120, 024102 (2018).**
**** J. Pathak, A. Wikner, R.Fussell, S.Chandra, B.R.Hunt, M. Girvan and E. Ott, Chaos 28, 041101 (2018).**
*****In preparation.**

Artificial Neural Network Reservoir Implementation*

- In our examples the reservoir is a directed weighted network of *D* neuron-like units which are the nodes of the network.
- Each node *i* has multiple inputs and outputs and its scalar state is the *i th* entry of the *D*-dimensional reservoir state vector $\mathbf{r}(t)$.
- The weighted connections between the nodes are represented by a *D* by *D* adjacency matrix **A**.



* There are other ways to physically implement a reservoir (optical, FPGAs, ….).

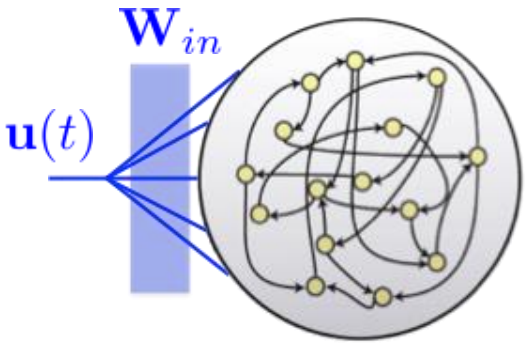
Reservoir Response Data

An input vector* is coupled to the reservoir network through a fixed, randomly generated input matrix.

Response Data $(-T \leq t \leq 0)$

$$\mathbf{r}(t+\Delta t) = \tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t))$$

$\mathbf{u}(t)$ and $\mathbf{r}(t)$ are recorded and stored.



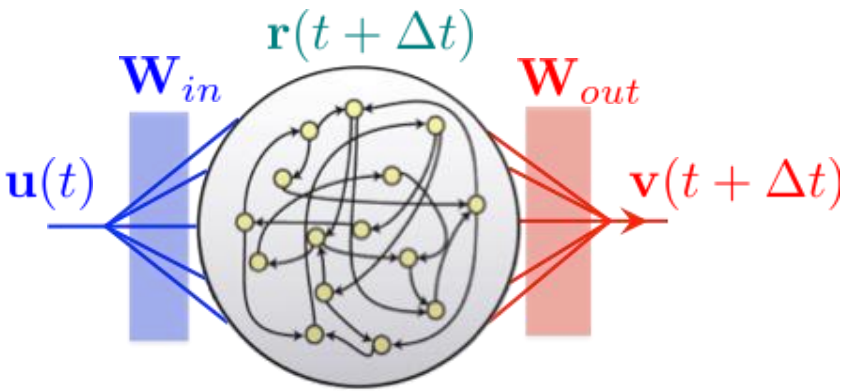
Neuromorphic motivation.

*During this reservoir response data acquisition phase, the elements of the input vector ***u(t)*** are measurements of state variables from the dynamical system of interest. Following this phase, our goal will be to predict the future evolution of ***u(t).***

Training for Prediction

An input is coupled to the reservoir network through a fixed, randomly generated input matrix.

$$\mathbf{v}(t) = \mathbf{W}_{out}\mathbf{r}(t)$$
$$\mathbf{v}(t) \simeq \mathbf{u}(t)$$



Response Data $(-T \le t \le 0)$

$$\mathbf{r}(t+\Delta t) = \tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t))$$

$\mathbf{u}(t)$ and $\mathbf{r}(t)$ are recorded and stored.

Training. Determine the output weight matrix by minimizing the following function:

$$\sum_{t=-T}^{0} \|\mathbf{W}_{out}\mathbf{r}(t) - \mathbf{u}(t)\|^2$$

+ (a Tikinov regularization term to prevent overfitting and promote generalization)

Prediction

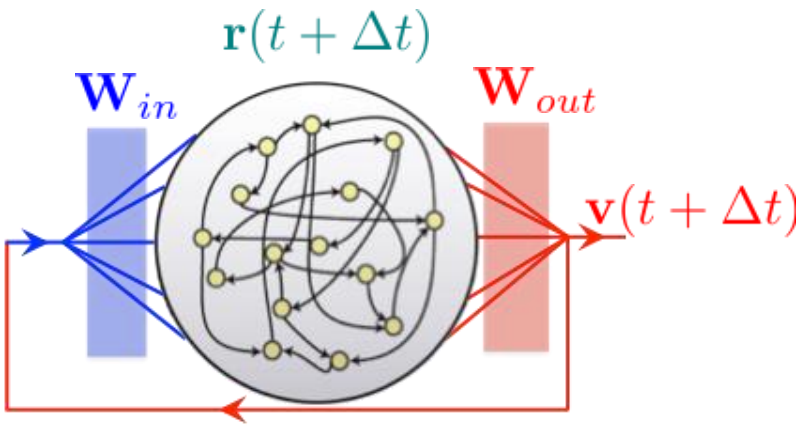
An input is coupled to the reservoir network through a fixed, randomly generated input matrix.

$$\mathbf{v}(t) = \mathbf{W}_{out}\mathbf{r}(t)$$
$$\mathbf{v}(t) \simeq \mathbf{u}(t)$$

Response Data $(-T \le t \le 0)$

$$\mathbf{r}(t+\Delta t) = \tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t))$$

$\mathbf{u}(t)$ and $\mathbf{r}(t)$ are recorded and stored



Training. Determine the output weight matrix by minimizing the following function:

$$\sum_{t=-T}^{0} \|\mathbf{W}_{out}\mathbf{r}(t) - \mathbf{u}(t)\|^2$$

+ (a Tikhonov regularization term to prevent overfitting)

Prediction $(t > 0)$

$$\mathbf{r}(t+\Delta t) = \tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{v}(t))$$

This implementation of reservoir computing prediction has previously been successful for low dimensional chaotic systems [Jaeger & Haas, Science (2004)]. Our first question is whether it can be used for high dimensional systems.

Outline

- **Using reservoir computing for model-free prediction from data.**
- **Examples extending the first bullet:**

(a)Prediction of a spatiotemporally chaotic system
(b)Parallel approach for large spatiotemporal systems*
(c) Hybrid knowledge-based/machine-learning approach**
(d)Combining the parallel and hybrid approaches***

- **Conclusion**

*** J. Pathak, B. Hunt, M. Girvan, Z. Lu and E. Ott, Phys. Rev. Lett. 120, 024102 (2018).**
**** J. Pathak, A. Wikner, R.Fussell, S.Chandra, B.R.Hunt, M. Girvan and E. Ott, Chaos 28, 041101 (2018).**
***** In preparation.**

Our Illustrative Example: The Kuramoto-Sivashinsky Equation

The Kuramoto-Sivashinsky (KS) equation is a spatiotemporally chaotic system:

$$y_t = -yy_x - y_{xx} - y_{xxxx}$$

$$y(x,t) = y(x+L,t)$$

$$\Lambda_{max} : \text{Largest Lyapunov Exponent}$$

We use numerical solutions of the KS equation to produce 'simulated measurements' from which we attempt to predict the future evolution without knowledge of the KS equation itself. The input to the reservoir system (i.e., the simulated data) is the vector whose elements are the *Q* values of *y(x,t)* at the grid points, *x = L/Q, 2L/Q, …. , L*.

For this example, we take the spatial interval between measurements to be small compared to the spatial extent of features of the KS equation, so that the measurements accurately approximate *y(x,t)* as a continuous function of *x*.

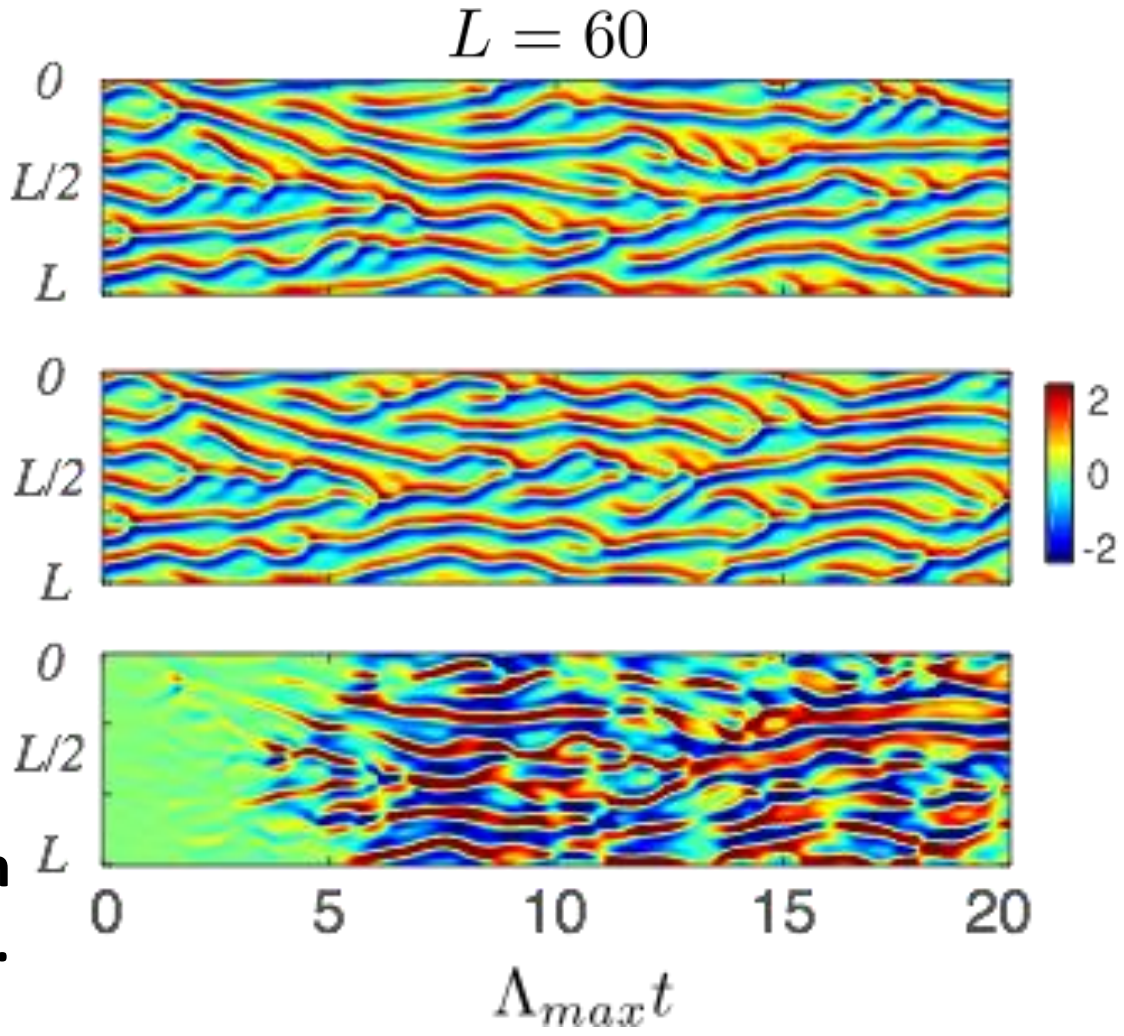
Results: Short Term Forecasting of Chaos

Actual Evolution

Reservoir Prediction x

(9000 reservoir nodes)

Error [(top panel) – (middle panel)]



*We achieve good prediction quality for 5 Lyapunov times.

*Training reusability.

The attractor dimension is about 13.

But

- **We desire the ability to treat much larger grid size (e.g., 2 or 3 dimensional grids).**
- **This potentially requires a vastly larger reservoir size to resolve the evolving spatial pattern.**
- **However, the computational time for training the reservoir scales poorly with the size of the reservoir.**

Outline

Using reservoir computing for model-free prediction from data.

Examples extending the first bullet:

(a)Prediction of a spatiotemporally chaotic system

(b)Parallel approach to prediction of large systems*

(c) Hybrid/knowledge-based machine-learning approach**

(d)Combining the parallel and hybrid approaches***

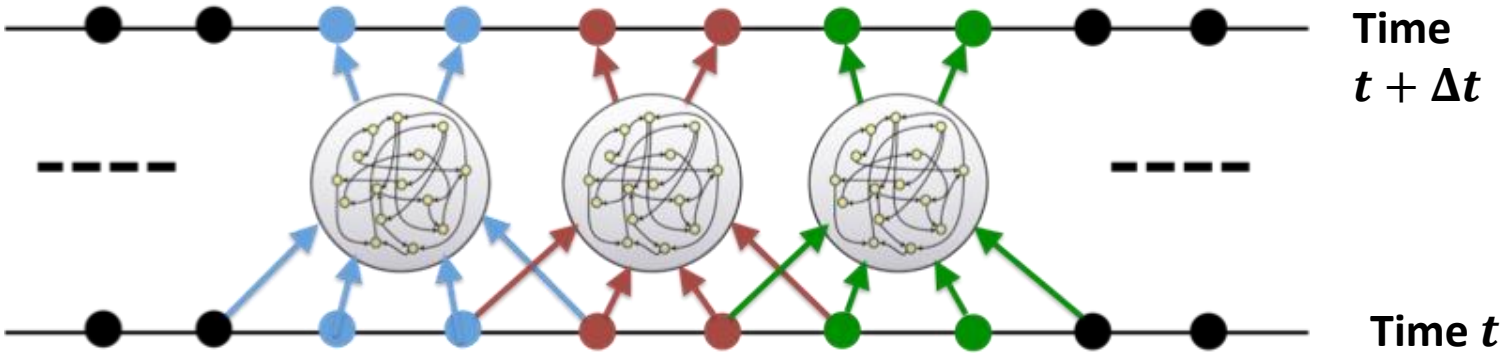
Conclusion

*** J. Pathak, B. Hunt, M. Girvan Z. Lu and E. Ott, Phys. Rev. Lett. 120, 024102 (2018).**

**** J. Pathak, A. Wikner, R.Fussell, S.Chandra, B.R.Hunt, M. Girvan and E. Ott, Chaos 28, 041101 (2018).**

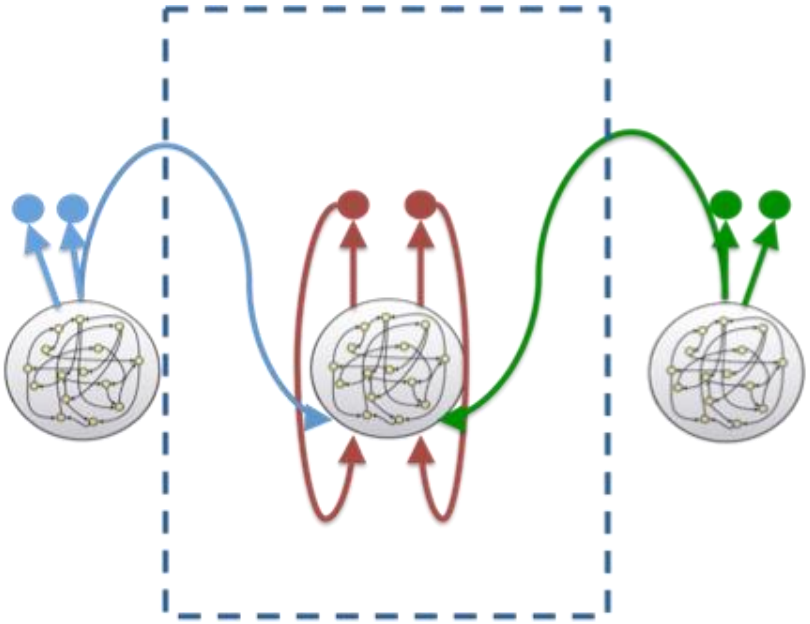
*****In preparation.**

Localized Prediction with Reservoirs in Parallel



- Based on the locality of short term causal interactions for the systems we consider.
- We use a stack of reservoirs which combine together and run in parallel.
- In the training phase, each reservoir takes inputs from a local neighborhood on the model grid and predicts a subset of its inputs. The local neighborhood consists of the nodes to be predicted plus buffer zones on each side.
- Each of the relatively small parallel reservoirs has its own relatively small output matrix (which can be independently trained).

Parallel Scheme in the Prediction Phase



During the prediction phase, for a given reservoir, the inputs that it predicts are replaced by its own output, while the inputs from its buffer zones are taken from the outputs of its two neighboring reservoirs on each side.

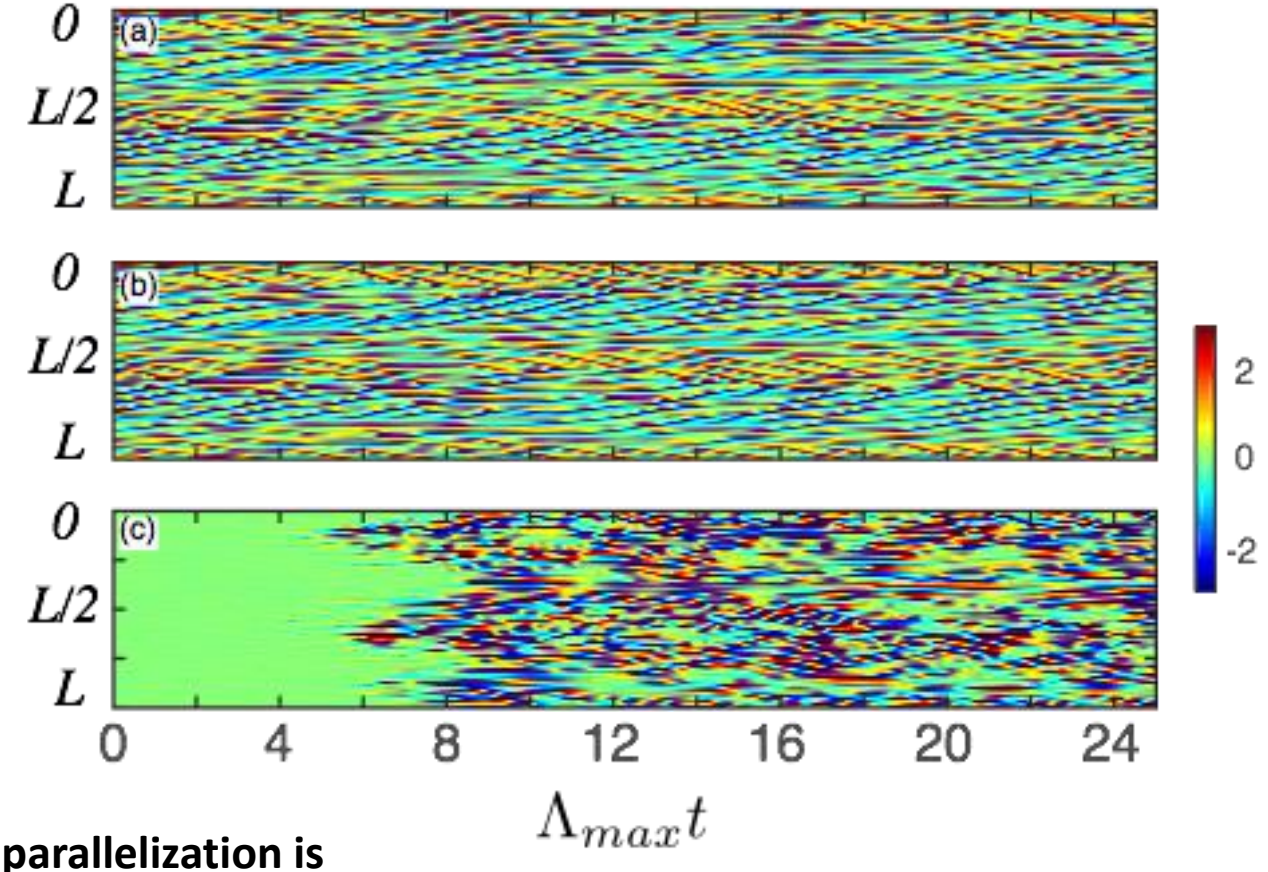
A Result for Parallel Prediction of a Large System

$L = 200, D_{KY} = 43$

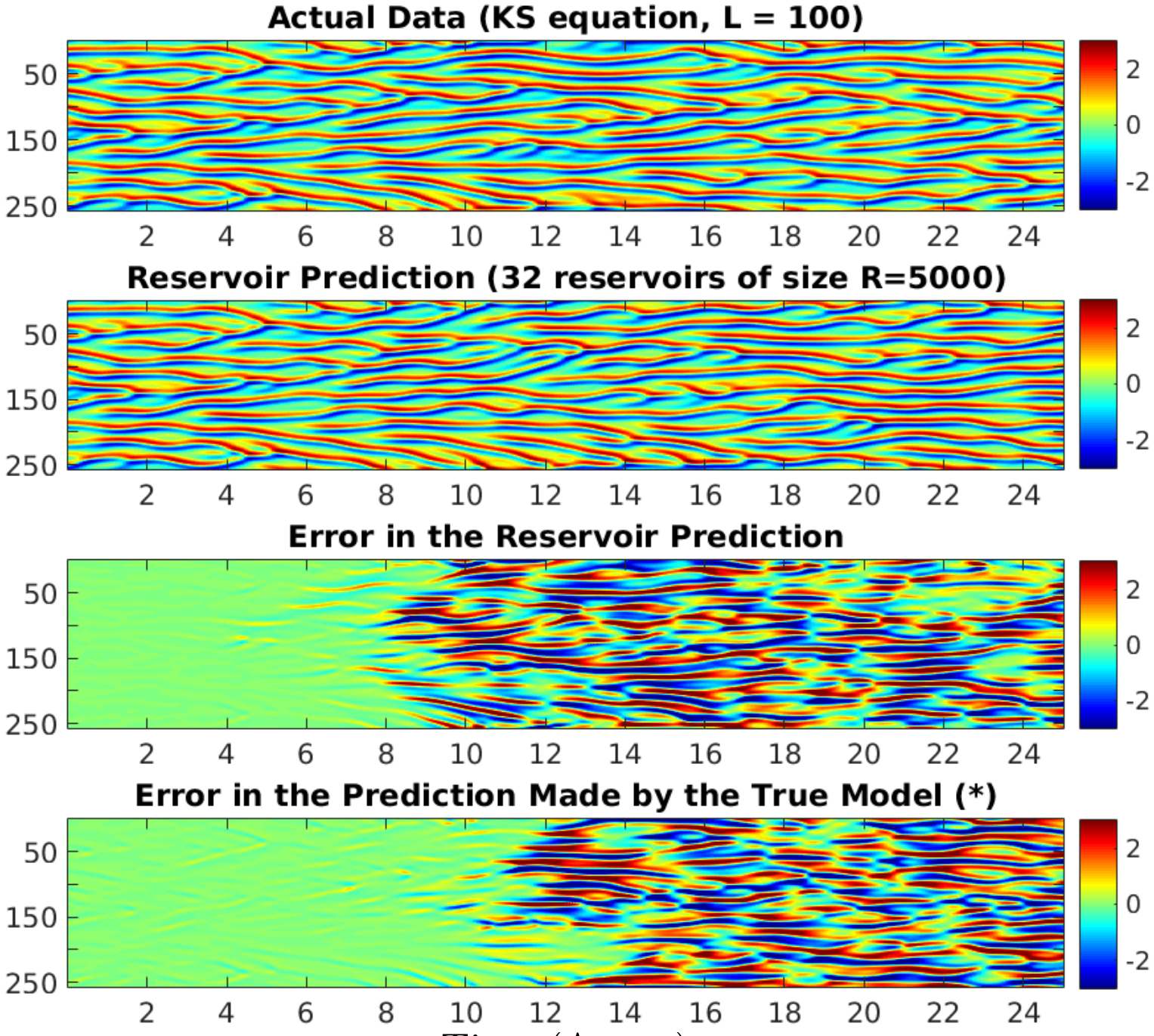
Actual Evolution

Parallel Reservoir Prediction (64 parallel reservoirs each with 5000 nodes)

Parallel Reservoir Prediction Error

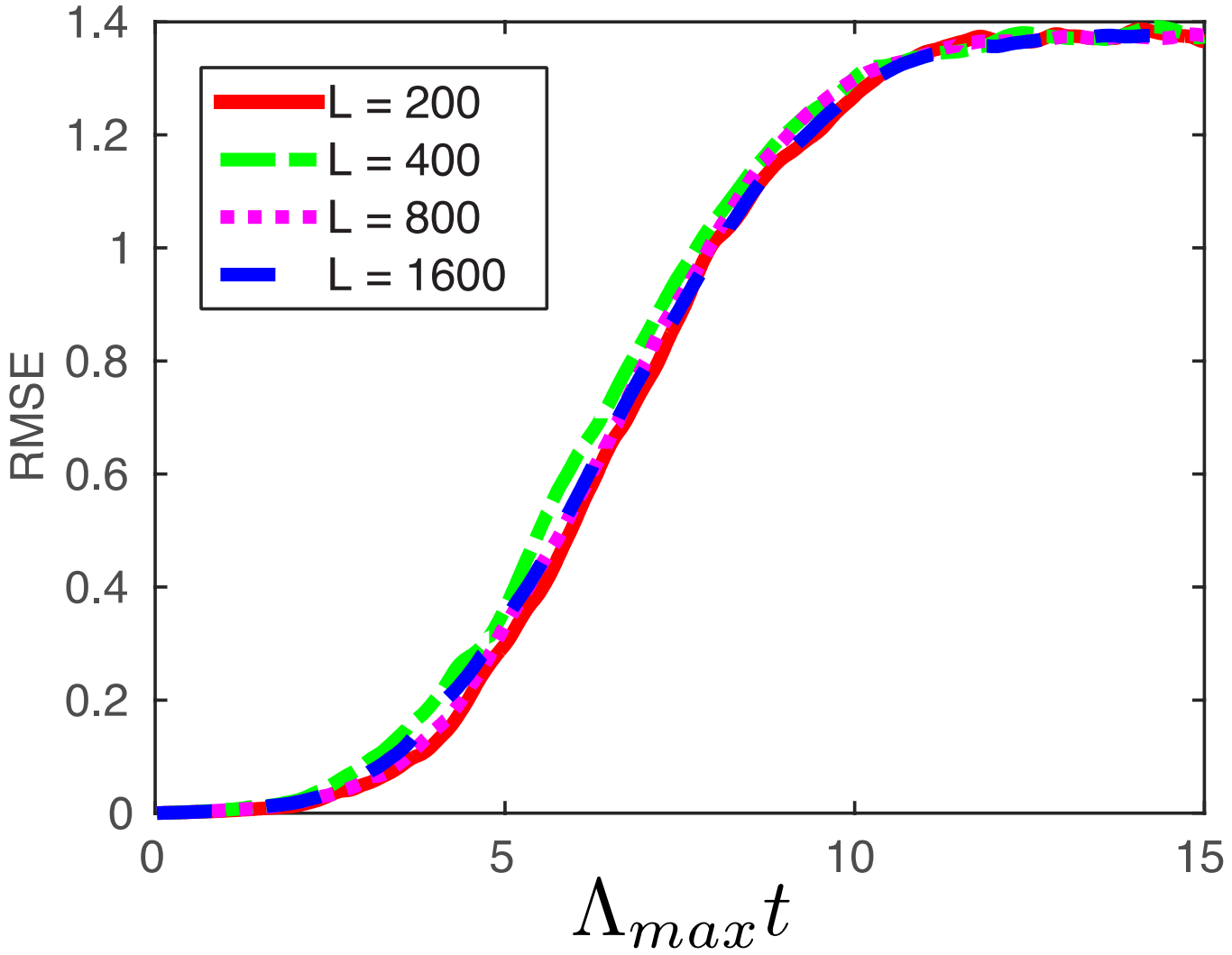


With our computer resources, parallelization is necessary in order for us to treat this *L=200* case.



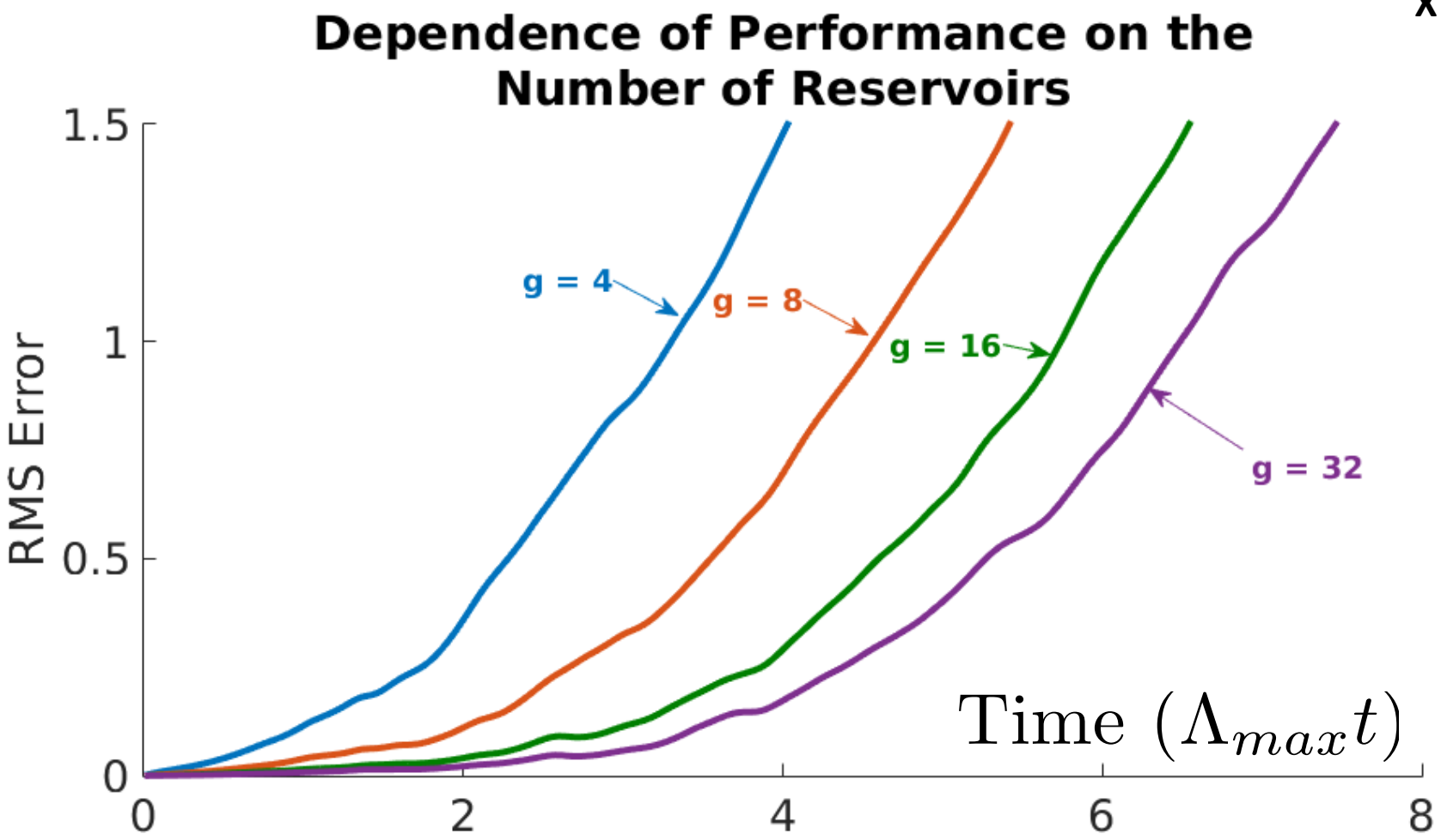
X

(*) The true model is given the imperfect initial condition provided by the first prediction step of of the reservoir



For the different curves in this plot, the size of the individual reservoirs (5000), and [(# of reservoirs) / L] = 0.32 are both kept fixed as L is increased.

x



g = (number of reservoirs)

Each reservoir has 5000 nodes

A Comment

The ML grid density need not be the same as the grid density of the knowledge-based component, and the ML grid density can also be inhomogeneous.

This freedom can be utilized for providing enhanced resolution (through greater density of the ML grid) or as a strategy for enhanced regional forecasting (by restricting the denser ML component to selected area).

Outline

- **Using reservoir computing for model-free prediction from data.**
- **Examples extending the first bullet:**

  (a) Prediction of a spatiotemporally chaotic system
  (b) Parallel approach for large spatiotemporal systems*
  (c) Hybrid knowledge-based/machine-learning approach**
  (d) Combining the parallel and hybrid approaches***

- **Conclusion**

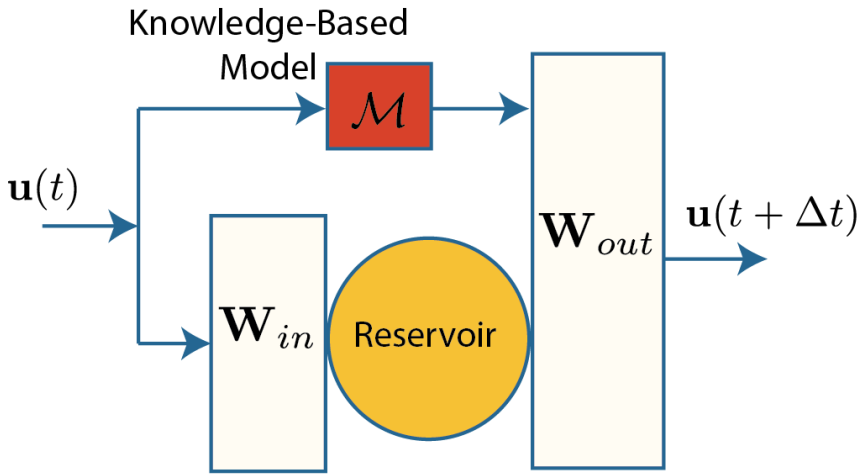
*** J. Pathak, B. Hunt, M. Girvan, Z. Lu and E. Ott, Phys. Rev. Lett. 120, 024102 (2018).**

**** J. Pathak, A. Wikner, R.Fussell, S.Chandra, B.R.Hunt, M. Girvan and E. Ott, Chaos 28, 041101 (2018).**

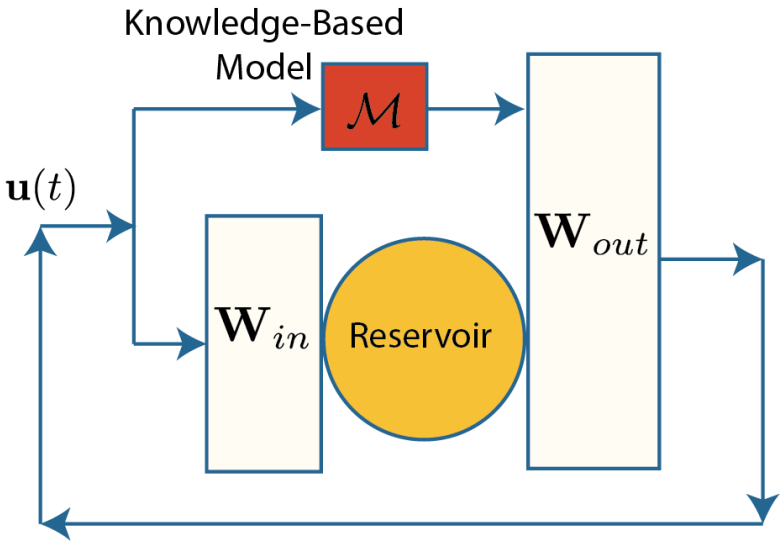
*****In preparation.**

HYBRID APPROACH

(a) **Open-loop Configuration (Training)**



(b) **Closed-loop Configuration (Prediction)**



Comments:

(1) It should be advantageous to utilize all potentially valid information whether it is in the form of data or physical laws.

(2) It is expected that the minimization of error in the training phase will lead to combining of disparate prediction components in such a way that, if one component is superior for some aspect, then it will be more determinative for that aspect of the combined prediction.

Reference:

J. Pathak, A. Wikner, R. Fussell, S. Chandra, B. R. Hunt, M. Girvan, and E. Ott,
Chaos 28, 041101 (2018).

The "true" unknown system:

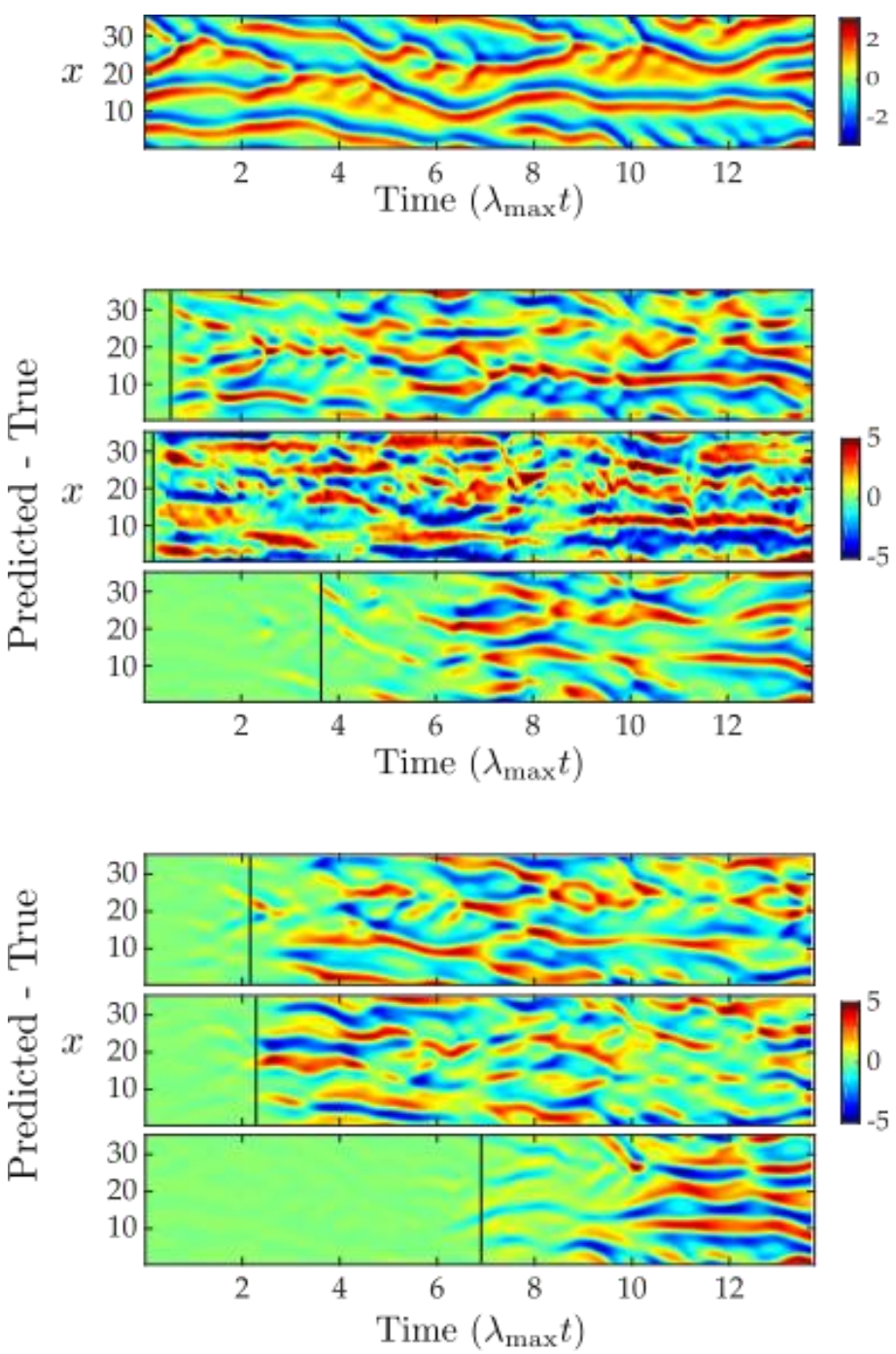
$$y_t = -yy_x - y_{xx} - y_{xxxx}$$

The imperfect model:

$$y_t = -yy_x - (1+\epsilon)y_{xx} - y_{xxxx}$$

$\mathcal{E}$= 0.1
500 reservoir nodes
hybrid

$\mathcal{E}$ = 0.01 ε
8000 reservoir nodes
hybrid



Outline

- **Using reservoir computing for model-free prediction from data.**
- **Examples extending the first bullet:**

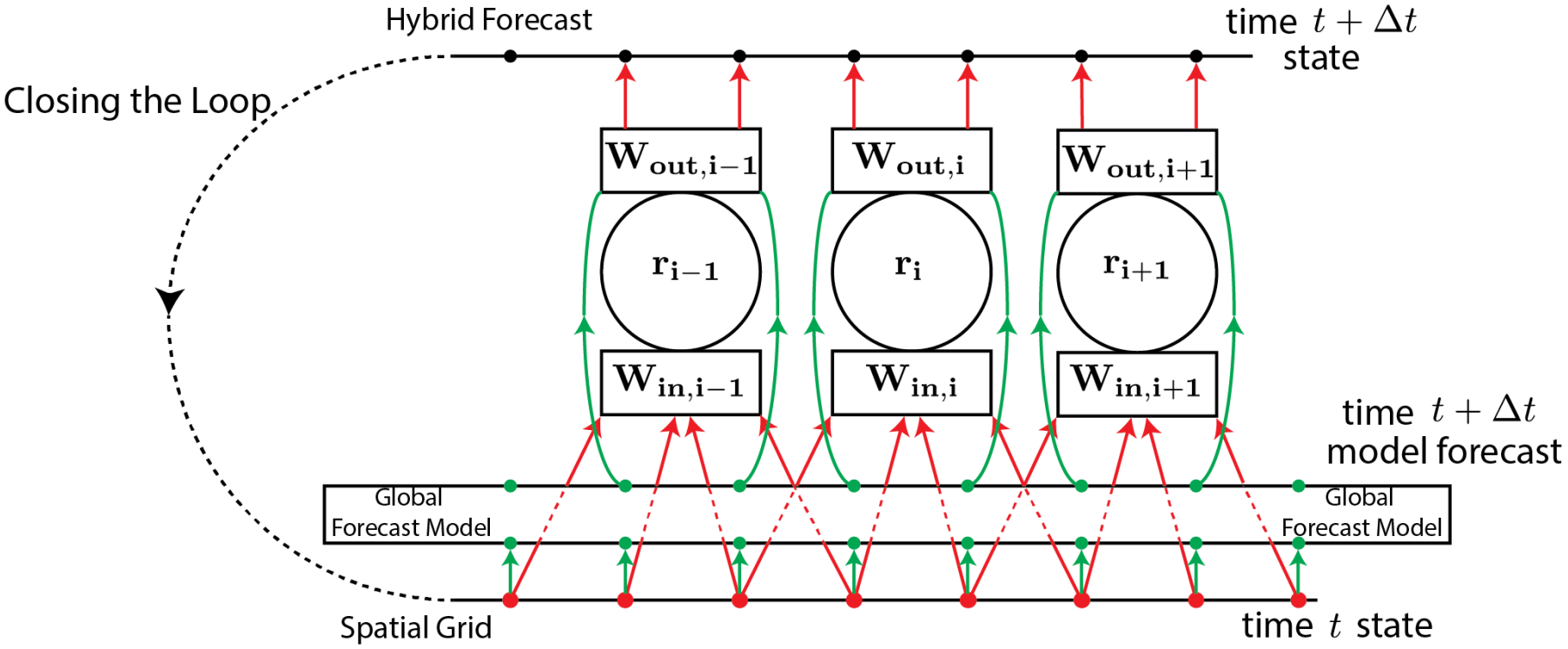
  (a) Prediction of a spatiotemporally chaotic system
  (b) Parallel approach for large spatiotemporal systems*
  (c) Hybrid knowledge-based/machine-learning approach**
  (d) Combining the parallel and hybrid approaches***

- **Conclusion**

*** J. Pathak, B. Hunt, M. Girvan, Z. Lu and E. Ott, Phys. Rev. Lett. 120, 024102 (2018).**
**** J. Pathak, A. Wikner, R.Fussell, S.Chandra, B.R.Hunt, M. Girvan and E. Ott, Chaos 28, 041101 (2018).**
*****In preparation.**

COMBINED PARALLEL/HYBRID SCHEME

For use in very large complex spatiotemporal systems (as in weather forecasting), we think that it will be crucial to use both our hybrid approach (to combine learning from data with physical knowledge) and our parallel approach (to make the learning component of such a huge complex system as the Earth's atmosphere feasible).



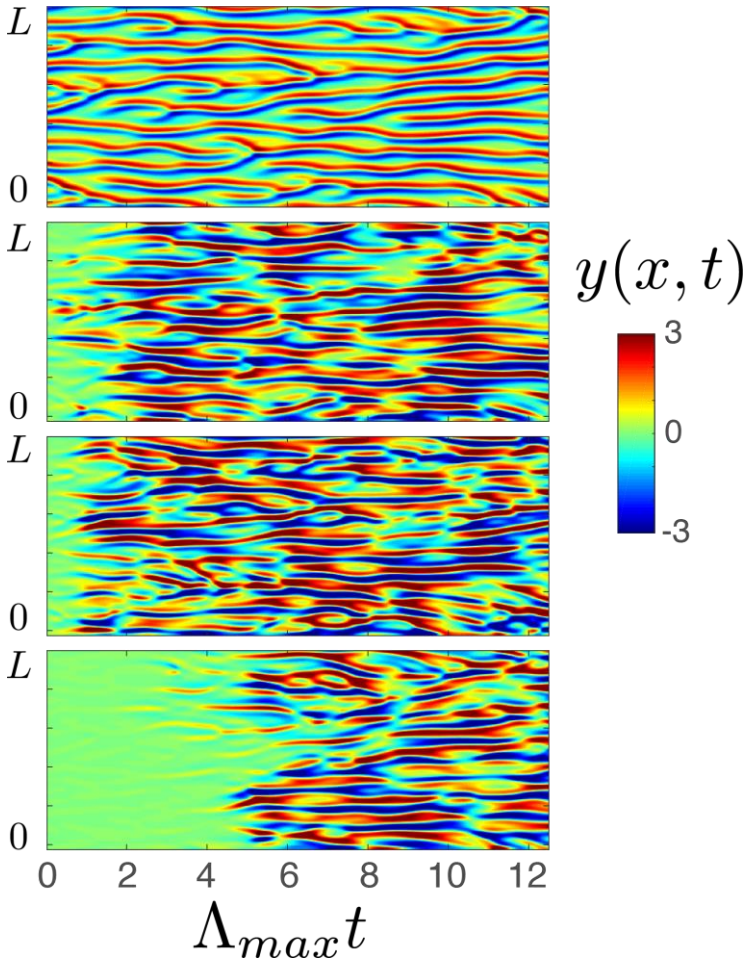
APPLICATION TO KS EXAMPLE (L=100)

Actual evolution

ML Parallel prediction error (4 reservoirs, 1000 nodes each)

Imperfect model error with 10% error factor

Parallel/Hybrid error



Another Example:

Models from Lorenz, J.Atmos.Sci. 62, 1374 (2005)

In this paper, Lorenz discusses three toy models of successively increasing 'reality' and complexity.

These toy models are for a single scalar variable with one-dimensional spatial variation and periodic boundary conditions (like our previous KS example).

They were designed by Lorenz so as to mimic features of weather (e.g., atmospheric Rossby waves) as one traverses a latitude circle around the Earth (hence the periodic boundary condition).

We refer to Lorenz's most 'realistic'/complex model as "Lorenz Model 3."

Lorenz Models 2 and 3

Lorenz Model 3: **Designed to roughly mimic the fact that atmospheric processes have space/time variations on multiple scales. The model has two scales, one is relatively long/slow; the other (which has substantially smaller amplitude) is short/fast. The two scales interact and influence each other's evolution.**

Lorenz Model 2: **Lorenz Model 3, but without the short/fast component.**

Our numerical experiment:

The 'true' evolution is from Model 3.

The imperfect knowledge-based model is a corresponding version of Model 2.

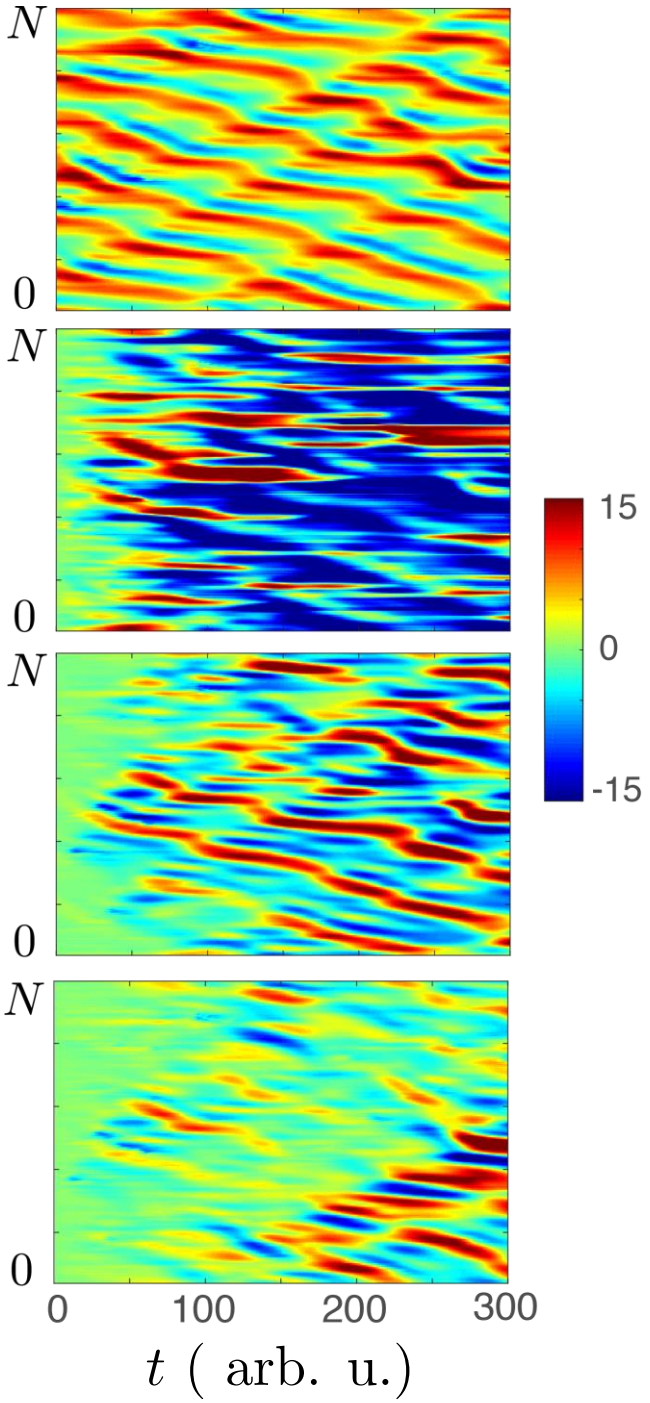
A PRELIMINARY RESULT

Actual Evolution
(Model 3)

Prediction Error
Parallel ML

Prediction Error
Imperfect Model
(Model 2)

Prediction Error
Hybrid Approach



Current Work of Our Group: Application to Weather Forecasting

- **Implementation of a parallel/hybrid system using a simplified, but semi-realistic, global weather code (SPEEDY) as the knowledge-based component.**
- **Incorporation of 6 hour cyclic forecasting (typical at operational weather forecasting centers) with an associated *data assimilation* procedure based on adaptation of the Local Ensemble Transform Kalman Filter (LETKF). We have preliminary results on this.**
- **The issue of a source of training data for the ML component.**

***E.g., atmospheric 'reanalysis data,' which is obtained by probabilistically estimating retrospective states via optimizing their consistency with both past and future evolution measurements.**

Conclusion

- **With more research, we believe machine learning will become a standard tool for studying the dynamics of a broad range of high dimensional complex chaotic systems.**
- **There are many potential applications (e.g., weather prediction, oceans, etc.)**

Our Papers Applying Machine Learning to NLD

- "Reservoir observers: Model-free inference of unmeasured variables," Z. Lu et al., Chaos 27, 041102 (2017).
- "Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data," J. Pathak et al. Chaos 27 , 121102 (2017).
- "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," J. Pathak et al., Phys. Rev. Lett. 120 , 024102 (2018).
- "Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model," J. Pathak et al. , Chaos 28, 041101 (2018).
- "Attractor reconstruction by machine learning," Z. Lu, et al., Chaos 28, 061104 (2018).

Blue color: Topics not discussed in this talk.