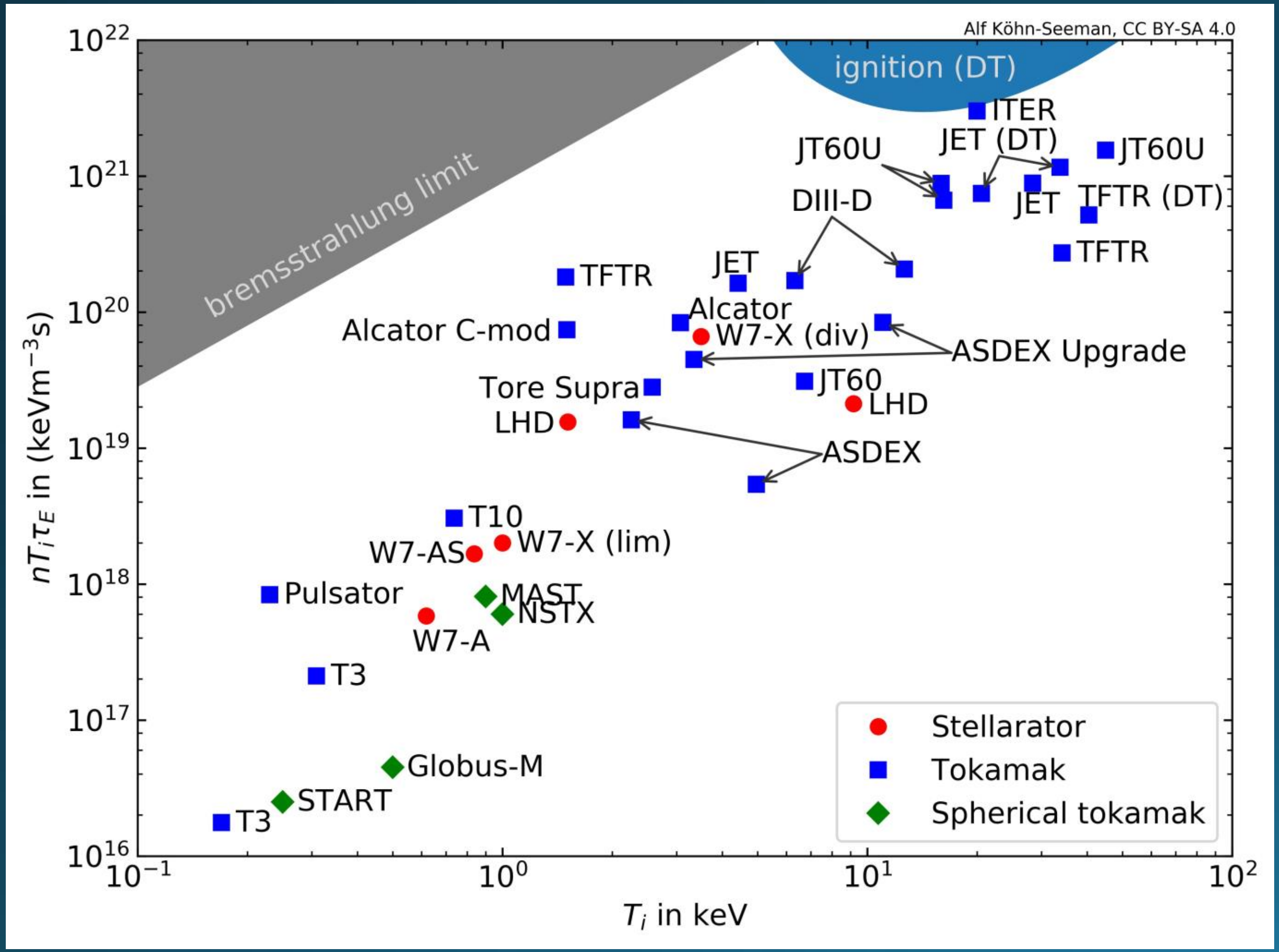
Modeling turbulence in tokamak plasmas with reservoir computing

Nathaniel Barbour

Advisor: Professor Bill Dorland

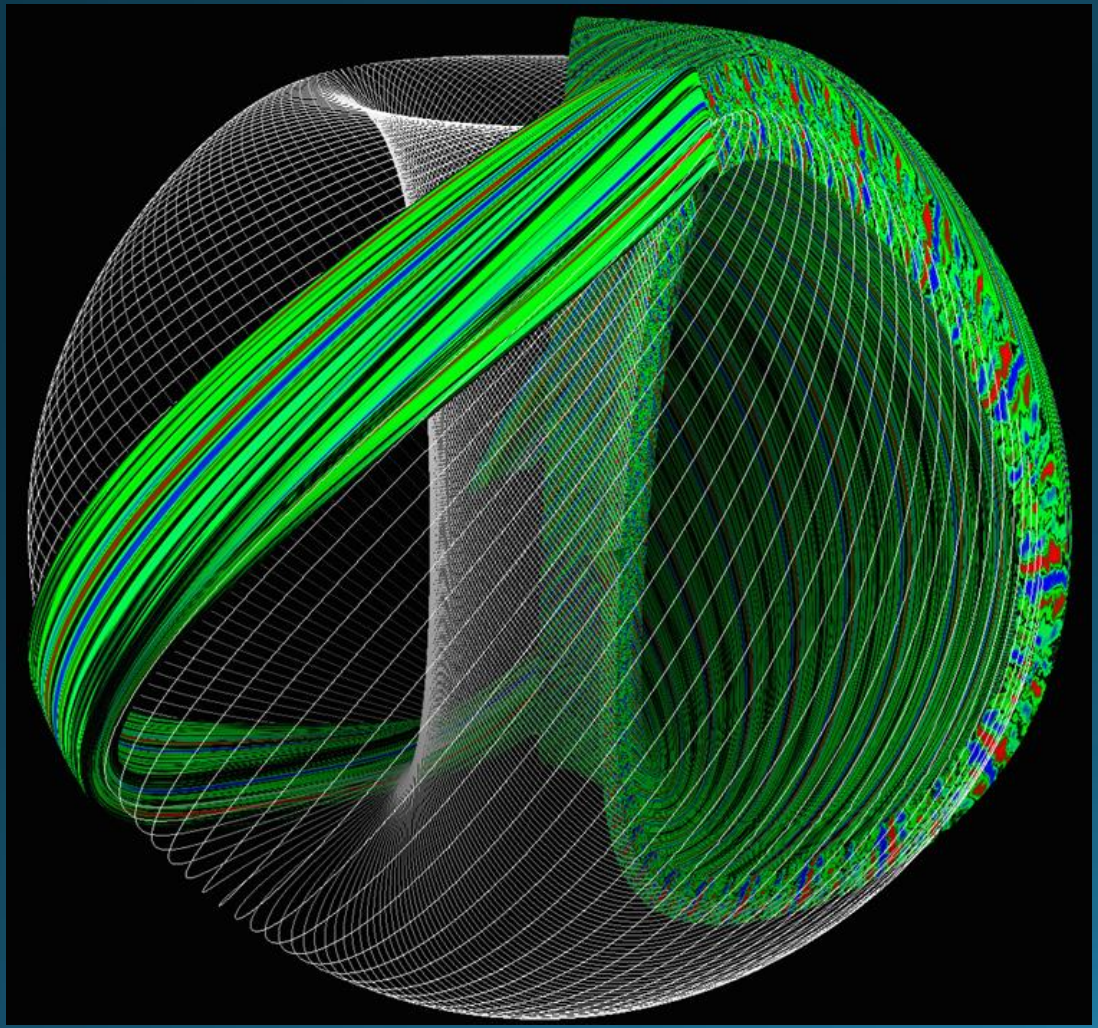
Department of Physics

Lawson criterion / triple product



- nTτ = density * temperature * energy confinement time
- Ignition is a critical goal for fusion reactors.
- Must increase τ to achieve ignition
- Mitigate turbulence -> increase τ

GS2 domain



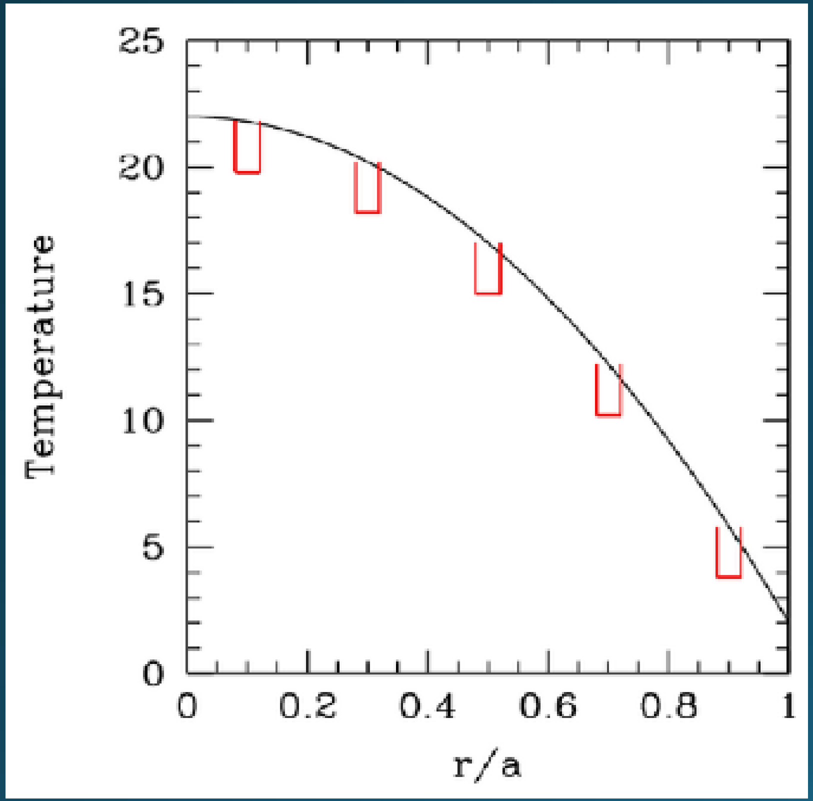
- Assume slow evolution of equilibrium profile compared to fluctuations
- Assume small correlation lengths
- 3 spatial dimensions, 2 velocity dimensions
- Domain restricted to flux tubes

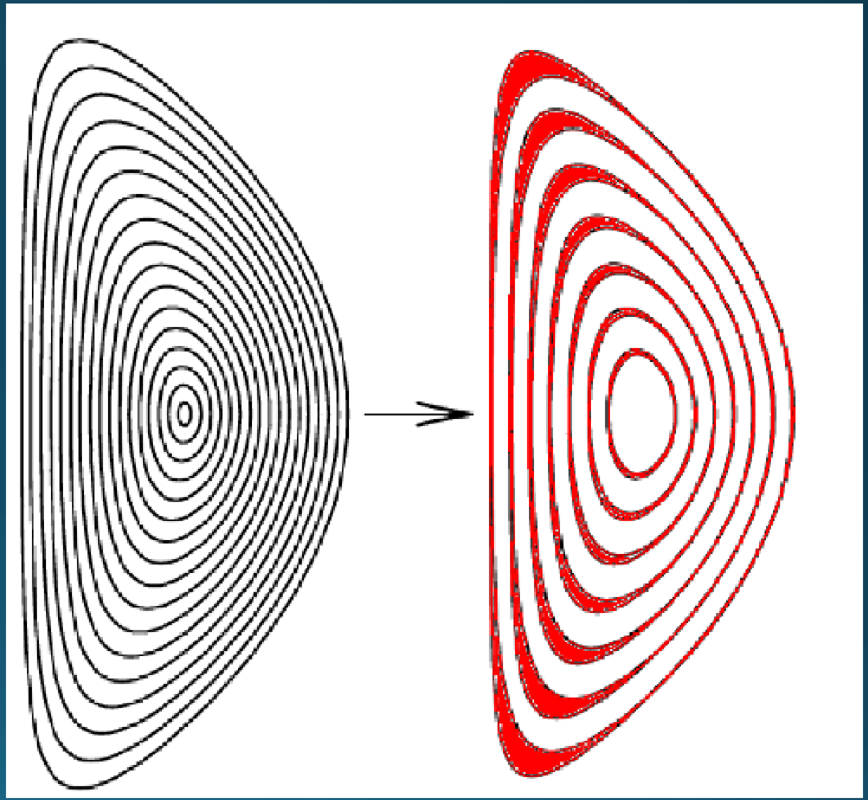
Multiscale modeling

- Turbulence simulation -> fast scale, computationally expensive
- Radial heat diffusion -> slow scale

- Confinement time ~0.25 s
- Turbulence modes ~0.00001 s

Heat flux





Guiding Physics Equation

$$\frac{\partial g_s}{\partial t} + \left[v_{ts} v_{\|} \hat{\mathbf{b}} + < \mathbf{v}_E > + \frac{\tau_s}{Z_s} \mathbf{v}_d \right] \cdot \nabla h_s + < \mathbf{v}_E > \cdot \nabla F_{Ms} - v_{ts} \mu \left(\hat{\mathbf{b}} \cdot \nabla B \right) \frac{\partial h_s}{\partial v_{\|}} = C(h_s).$$

- Electrostatic gyrokinetic equation
- Describes time evolution of gyrokinetic distribution
- Heat flux derived from radial velocity perturbation
- Can we find the heat flux without fully solving the GK equation?

Current standard for turbulence modeling

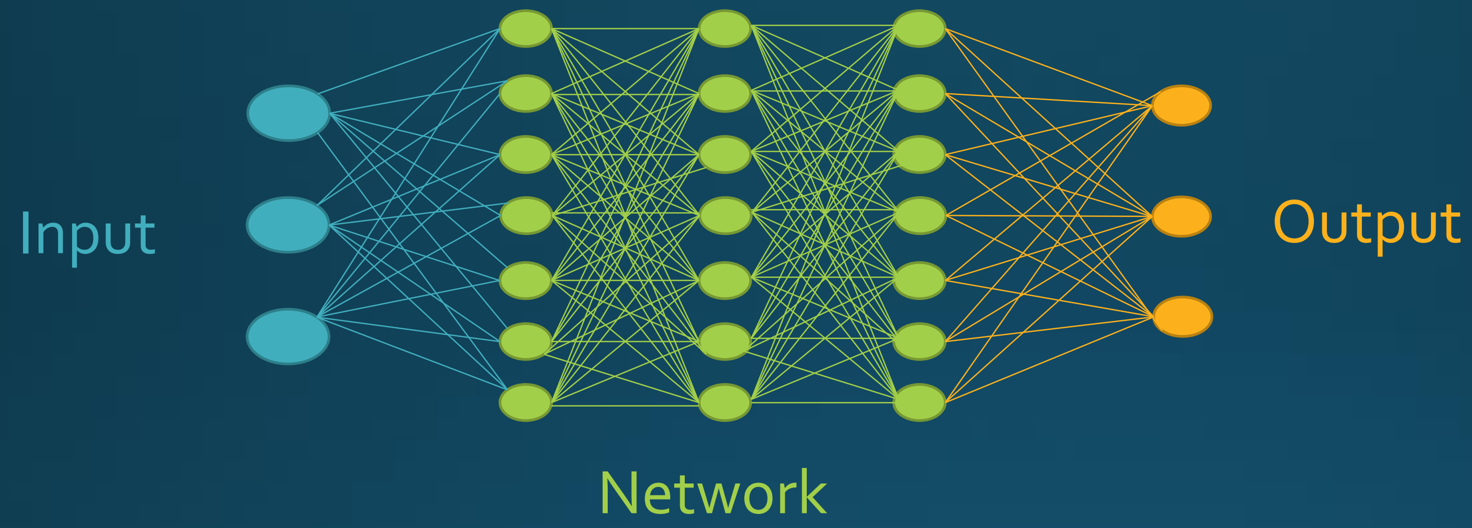
- GS2: W. Dorland et al. Phys. Rev. Lett. **85,** 5579 (2000).
  - Industry standard
  - Parallelized 5-D gyrokinetic code
- GX:
  - Unpublished but optimized and benchmarked
  - Orders of magnitude faster than GS2
- Trinity 1-D transport solver: M. Barnes Ph.D. Thesis (2008)
  - Extracts diffusion parameters from turbulence codes
  - Calculates heat flux

1-D Kuramoto-Sivashinsky (KS) Equation

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

- $u(x, t)$ is periodic on $[0, L)$
- Arises in plasma physics: trapped ion mode instabilities
- Quadratic nonlinear term
- Higher-order dissipation
- Nontrivial chaotic dynamics
- A long-wavelength limit of the equations in turbulence code

Traditional Artificial Neural Networks



- Basic structure: feed-forward series of layers of neuron-like units.
- Artificial neurons receive weighted inputs and process into outputs.
- Weights between layers are optimized on a training set.
- Primarily used for pattern recognition or classification tasks

Recurrent neural networks

- Network weights updated with backpropagation through time
- Advantage: feedbacks permit system memory
- Disadvantage: higher training cost
- Useful for studying time series data

Reservoir Computer

- Recurrent ANN with distinctions such as:
  - Connections defined by **random and sparse** adjacency matrix.
  - Input and internal **weights remain fixed.**
- Advantages:
  - Simpler training process
  - Output parameters can quickly be reused at a later time.
- Disadvantages:
  - Like other ANN methods, black box
  - Reservoir size scales with problem size unfavorably

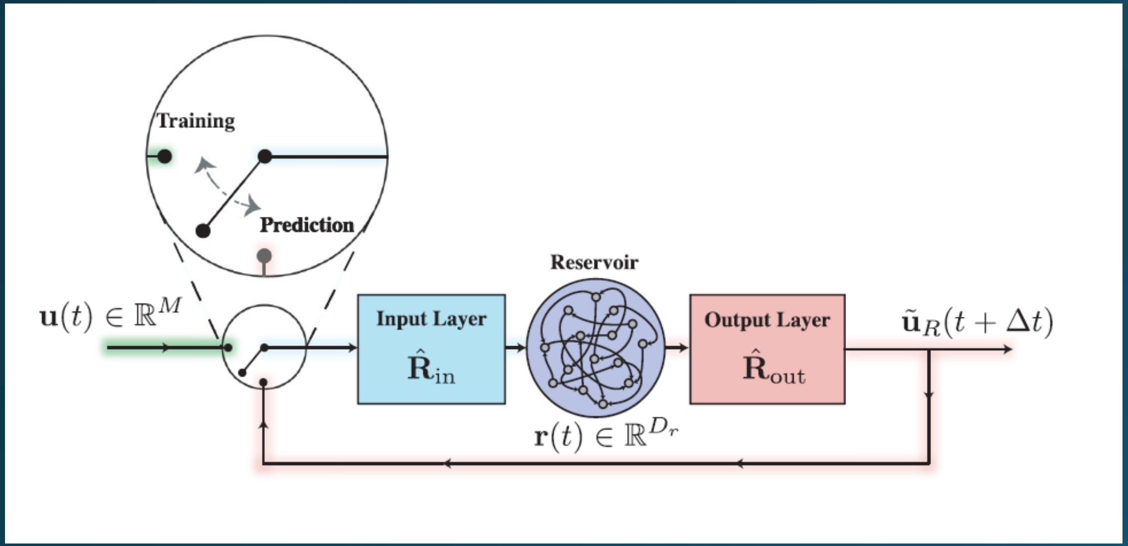
Project goals

- Implement reservoir using tensor library
- Train reservoir to predict future states of turbulent plasma
- Use reservoir to predict turbulent heat flux within a specified tolerance
- Determine if reservoir's time to solution is faster than GX alone

Reservoir Details

- Input of dimension M
- Network of D_R neuron units with sparse adjacency matrix $\mathbf{A}$
- $\mathbf{W}_{in}$ Input coupling matrix of dimension D_R x M
- $\mathbf{W}_{out}$ Output coupling matrix of dimension M x D_R
- State vector $\mathbf{r}(t + \Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t)]$

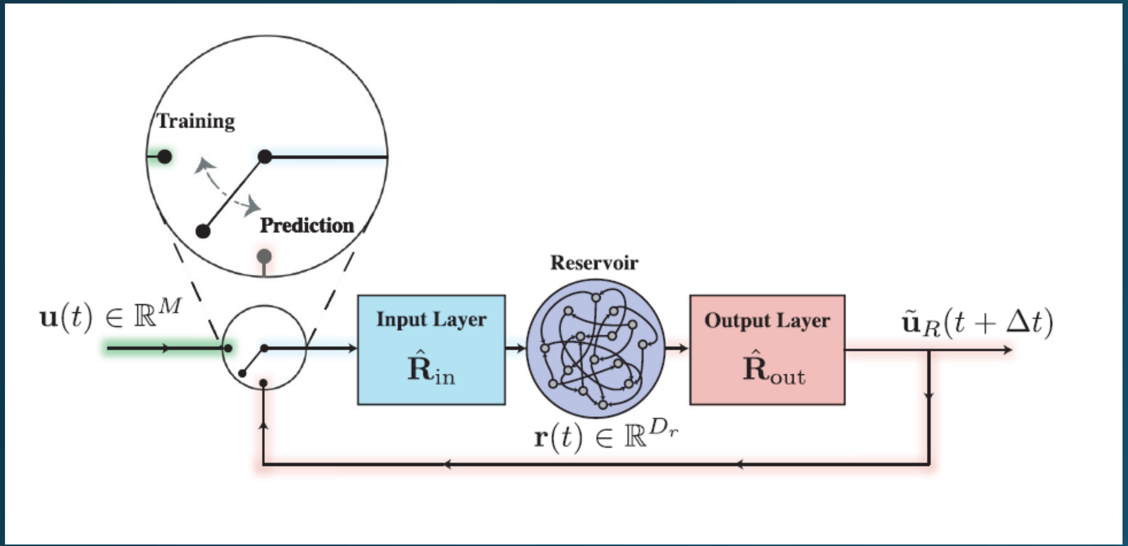
Reservoir-only approach



- **u**(t) : full GX solution
- $\hat{R}_{in}$: linear map to reservoir ANN
- Reservoir: recurrent neural network
- $\hat{R}_{out}$: linear map from reservoir to output

J. Pathak *et al.* Chaos **28**, 041101 (2018); https://doi.org/10.1063/1.5028373

Reservoir-only approach



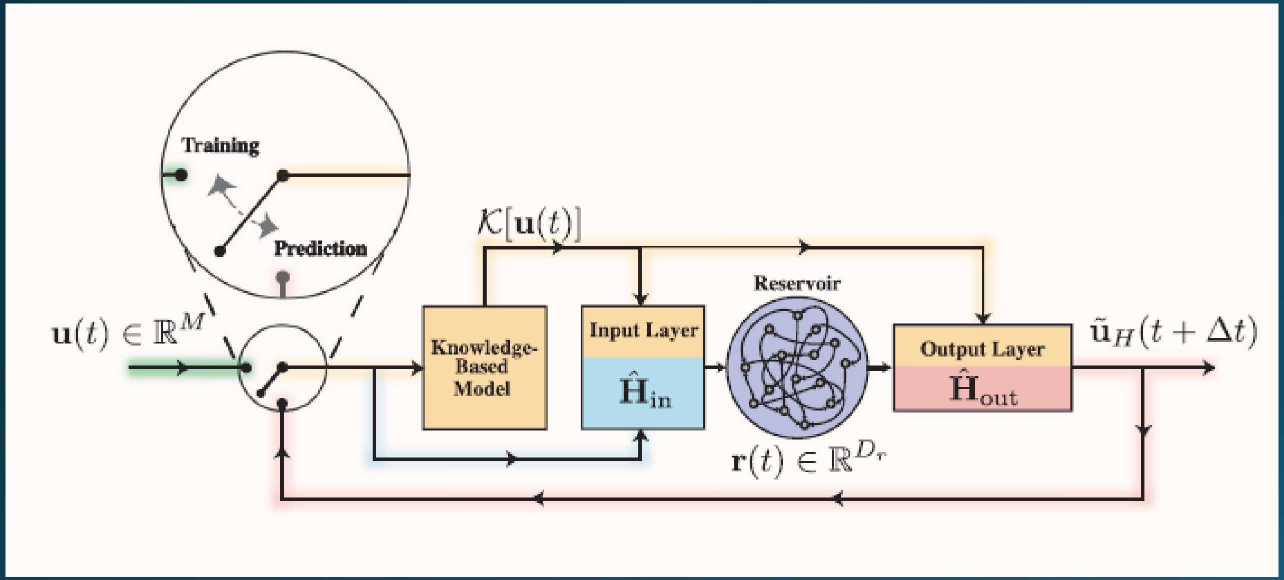
J. Pathak *et al.* Chaos **28**, 041101 (2018); https://doi.org/10.1063/1.5028373

- u(t) = full GX solution
- $\hat{R}_{in}$= linear map to reservoir ANN
- $\hat{R}_{out}$ = linear map from reservoir to output
- Evolve reservoir state:

$$\mathbf{r}(t + \Delta t) = \tanh\left[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}}\mathbf{u}(t)\right]$$

- Optimize output layer using Tikhonov-regularized linear regression

Hybrid reservoir approach



J. Pathak *et al.* Chaos **28**, 041101 (2018);
https://doi.org/10.1063/1.5028373

- Allow u(t) to be the standard timestep for integration
- Knowledge-Based Model = GX running with less resolution
- In some cases, enables smaller reservoir -> faster training and solution

Verification

- 1-D Kuramoto-Sivashinsky Equation
  - Compare to published solutions from existing Matlab code
- 5-D gyrokinetic turbulence solution
  - Compare plasma states and turbulent heat fluxes to GX
- Will separately test input and output interfaces

Implementation

- C++
- CUDA
- cuTENSOR
- NVIDIA A100 Tensor Core GPU

AMSC 663 timeline

- October-early November:
  - Develop C++ solver for 1-D Kuramoto-Sivashinsky (KS) equation
  - Build solver using GX and verify
- Mid November – December:
  - Implement reservoir using cuTENSOR
  - Train reservoir to predict 1-D KS states
  - Reproduce results of J. Pathak et al. *Phys. Rev. Lett.* **120**, 024102 (2018).
  - Reproduce 1-D KS result from J. Jiang and Y. Lai. *Phys. Rev. Research* **1,** 033056 (2019).

Alternative AMSC 663 timeline

- October-early November:
  - Develop C++ solver for 1-D Kuramoto-Sivashinsky (KS) equation
  - ~~Build solver using GX and verify~~
- Mid November – December:
  - Implement reservoir using cuTENSOR
  - Train reservoir to predict 1-D KS states
  - Reproduce results of J. Pathak et al. *Phys. Rev. Lett.* **120**, 024102 (2018).
  - ~~Reproduce 1-D KS result from J. Jiang and Y. Lai. *Phys. Rev. Research* **1**, 033056 (2019).~~
- Still provides foundation for AMSC 664

AMSC 664 timeline

- February-early March:
  - Build reservoir for the 5-D gyrokinetic turbulence code GX
  - Calculate macroscopic average turbulent heat fluxes
  - Compare time to solution with direct numerical solution
- March-May:
  - If reservoir is faster, call reservoir from 1-D transport code Trinity
  - Benchmark solutions against existing codes
  - If reservoir is outside of tolerance, implement hybrid reservoir and test

Deliverables

- Proposal, progress reports, presentations
- Trained reservoirs
- C++/CUDA codes: documented and in Github
- Figures comparing time to solution for both methods
- Uncertainty estimates for reservoir solutions
- Sample input files

References

- M. Barnes. Ph.D. Thesis (2008) arxiv:0901.2868
- W. Dorland et al. Phys. Rev. Lett. **85**, 5579 (2000).
- J. Jiang and Y. Lai. *Phys. Rev. Research* **1**, 033056 (2019).
- J. Pathak et al. Chaos **28**, 041101 (2018).
- J. Pathak et al. *Phys. Rev. Lett.* **120**, 024102 (2018).

Additional slides

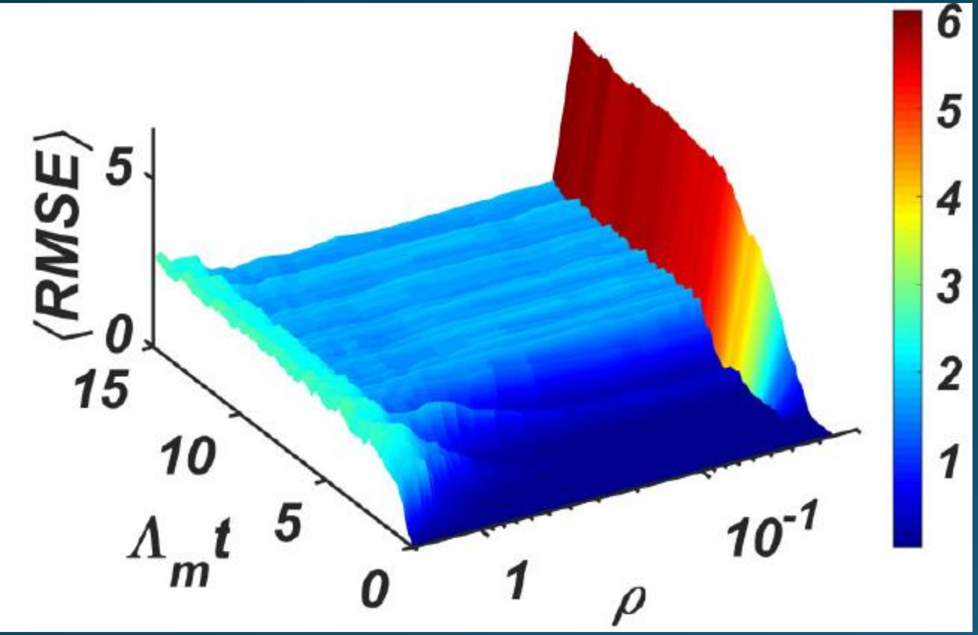
Training

- Tikhonov-Regularized Linear Regression
- Minimize:

$$\sum_{m=1}^{T/\Delta t} \|\mathbf{u}(-m\Delta t) - \tilde{\mathbf{u}}_R(-m\Delta t)\|^2 + \beta\|\mathbf{W}_{\text{out}}\|^2$$

- Regularization parameter to mitigate potential overfitting

Impact of spectral radius of reservoir network for 1-D KS equation



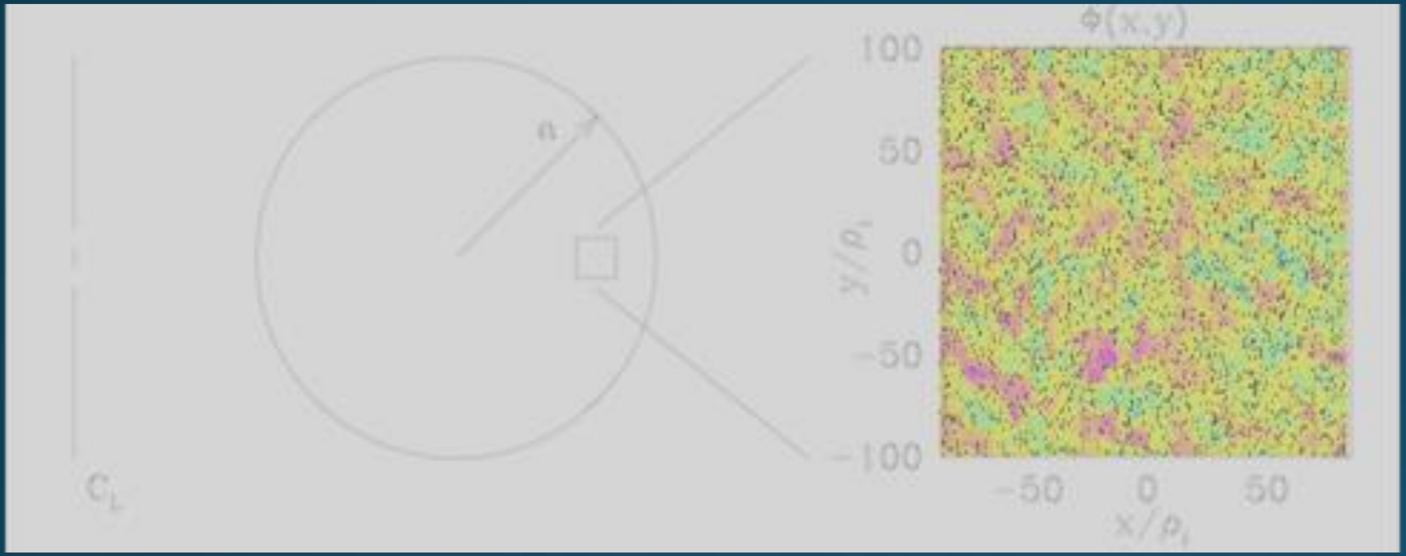
Jiang and Lai. Phys. Rev. Research **1**, 033056 (2019).
DOI:10.1103/PhysRevResearch.1.033056

- Reservoir is scaled by a parameter to set the spectral radius.
- Spectral radius impacts ensemble-averaged RMSE
- **Potential challenge for the 5-D gyrokinetic case**

Advantages of magnetic fusion energy

- Baseload power supply replacement
- No CO_2 emission in power plant operation
- Safe waste product: helium
- Abundant fuel: water and lithium
- No risk of "meltdown"

Turbulence



- Simulate small-scale turbulence in the flux tube
- Mitigate turbulence -> increase confinement time τ