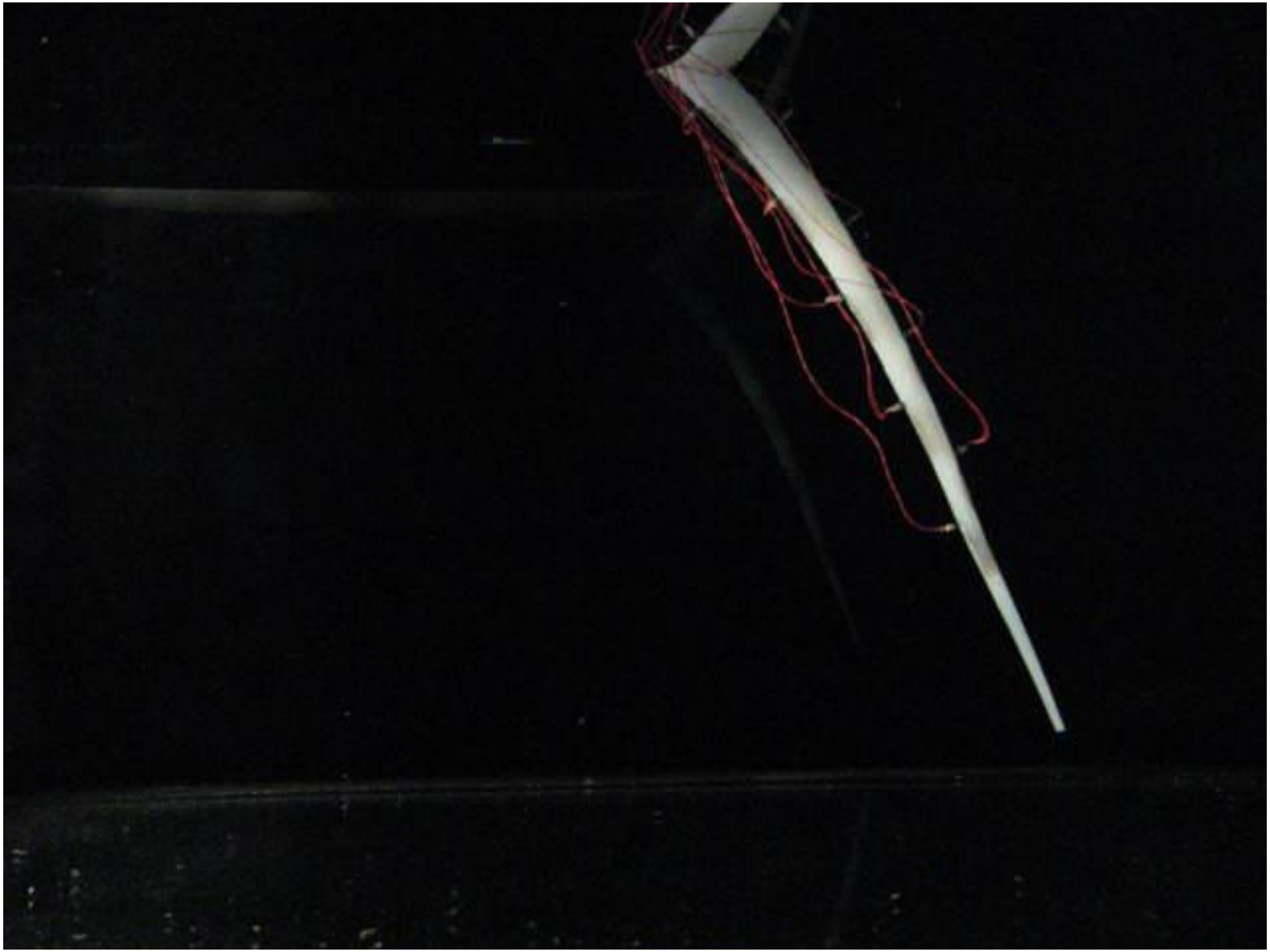
Information processing in soft robots 1: reservoir computing

Kohei Nakajima
University of Tokyo

2025/10/31 Friday, 16:40-18:15
立命館大学：特殊講義（ソフトロボット学）
zoom

Computing with soft robotic arm?



Contents

1. 10/31: Reservoir computing

   Nonlinear dynamics as computational device

2. 11/7: Physical reservoir computing

   Soft body dynamics as computational device

The report topic will be announced at the final lecture (11/7)!

Where I'm from…

(education)

Ph. D. The University of Tokyo, 2009.
M. S. The University of Tokyo, 2006.
B. S. The University of Tokyo, 2004.

(major)
nonlinear dynamics, chaos theory, recurrent neural network, embodiment.

Takashi Ikegami

(research experience)

2009 **Postdoctoral Researcher** (EU project: OCTOPUS)
Department of Informatics,
University of Zürich

2013 **JSPS Postdoctoral Fellow**
ETH Zürich

2014 **Assistant Professor**
The Hakubi Center for Advanced Research,
Kyoto University
(from 2015, 10, JST PRESTO Researcher)

2017 **Project Associate Professor**
The University of Tokyo

2020 **Associate Professor**
The University of Tokyo

Rolf Pfeifer

(topic)
soft robotics, morphological computation, (physical) reservoir computing.

Octopus arm computer

日経サイエンス「人工知能」(2020年6月)

特集　AIの身体性

体で計算するコンピューター

タコは脳を使わず，足だけで複雑な運動を制御する
同様に生物の体などの物理系の動きを利用して計算し
ローコストで機械学習を実行する新たな仕組みが登場した

古田彩（編集部）
協力：中嶋浩平（東京大学）

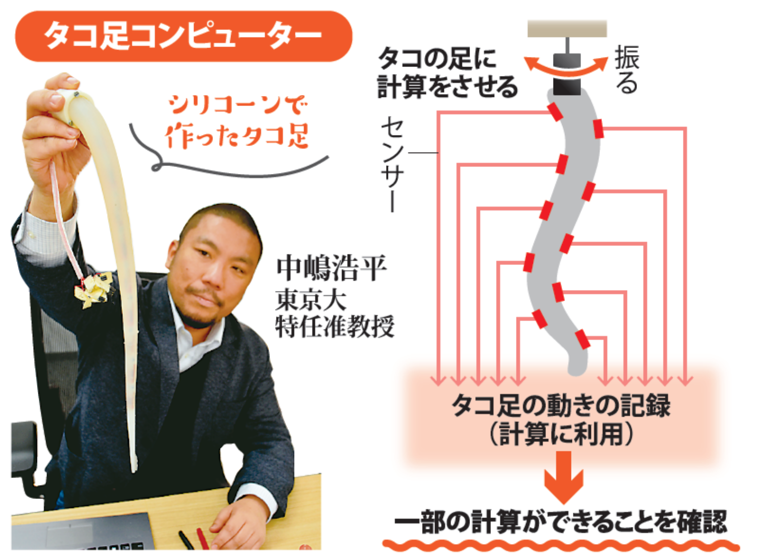
タコ足コンピューター　タコの足は脳を介さず，足だけで複雑な運動をこなす。人工のタコ足も，自らの複雑な動きを使って，様々な計算方法を学習する。

SWI swissinfo.ch　Swiss perspectives in 10 languages

Swissinfo
2013

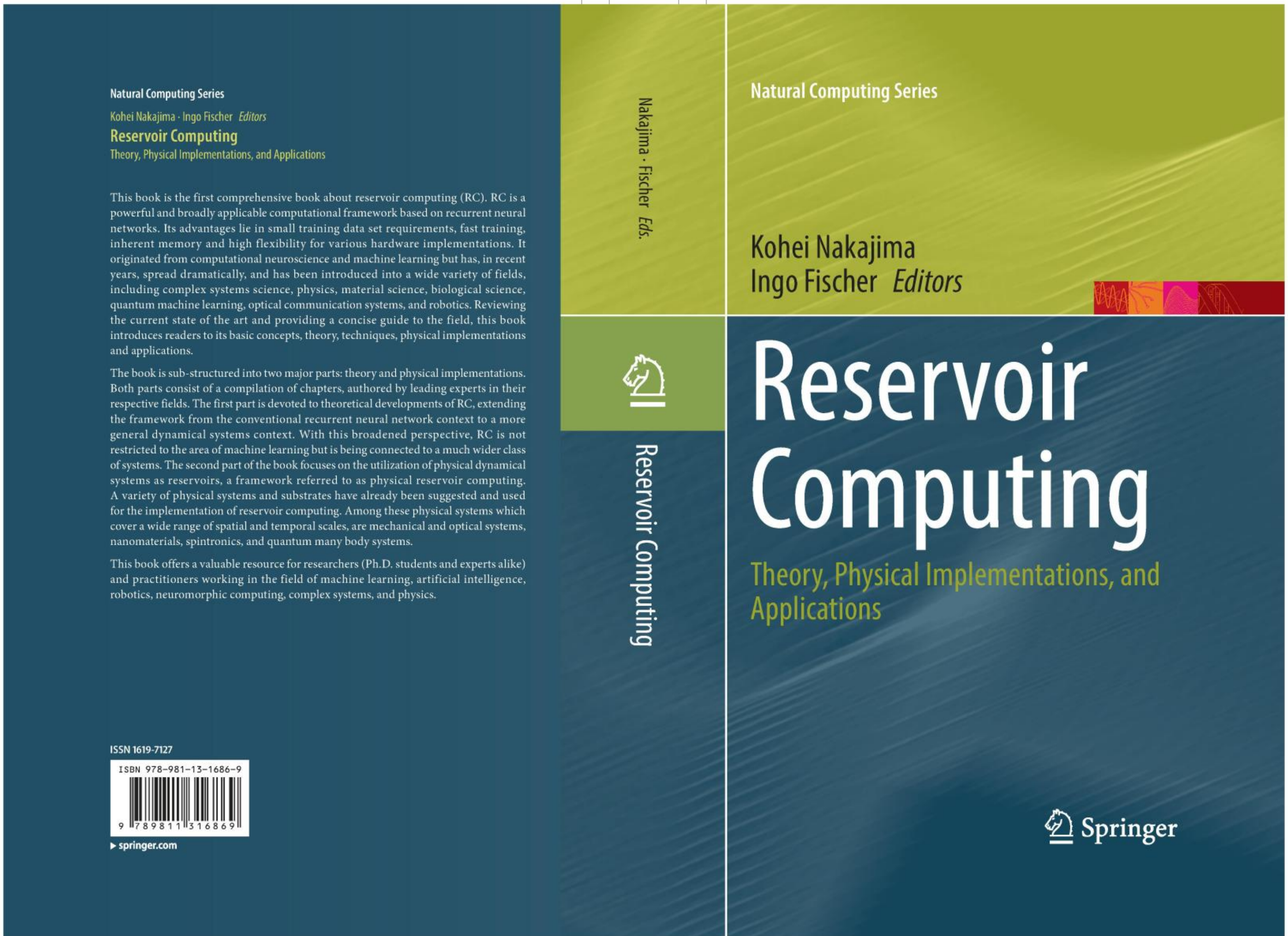
Next generation of robots will have a gentle touch

毎日新聞2020



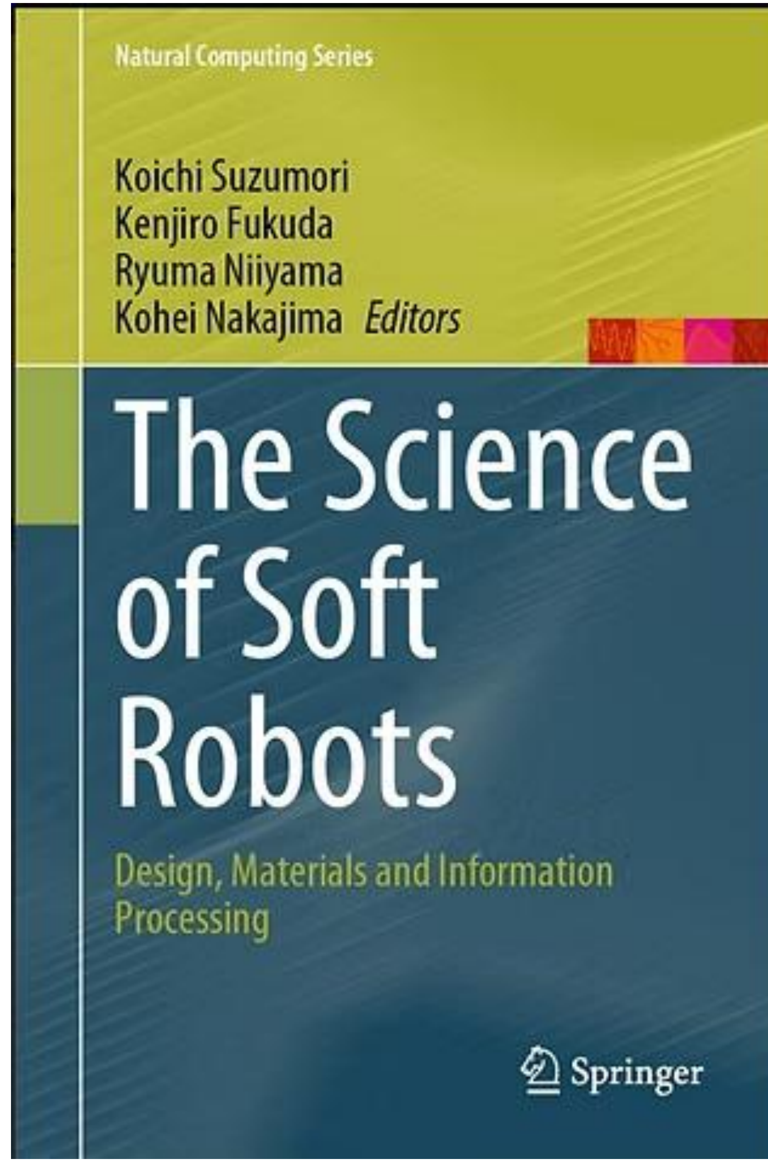
タコとイカ 異質な知性と体に秘められた謎|
ガリレオX第228回 (2020年9月27日)

https://www.youtube.com/watch?v=rn8wKflrm7Y



Nakajima, K., & Fischer, I. (Eds.). (2021). *Reservoir Computing: Theory, Physical Implementations, and Applications.* Springer Nature.

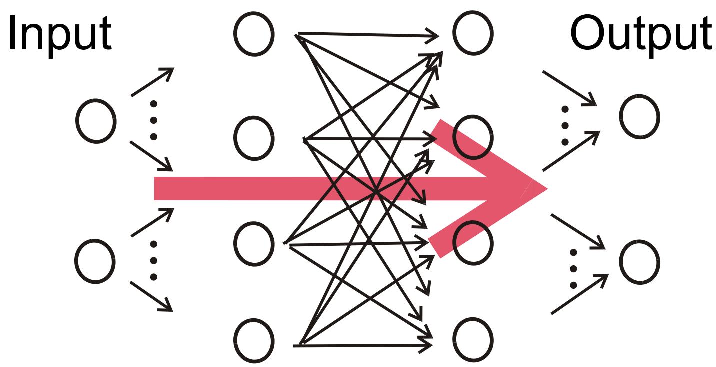
鈴森 康一、中嶋 浩平、新山 龍馬、舛屋 賢 編著、**ソフトロボット学入門 -基本構成と柔軟物体の数理-**、オーム社（2023年1月19日）. ISBN: 978-4-274-22998-5



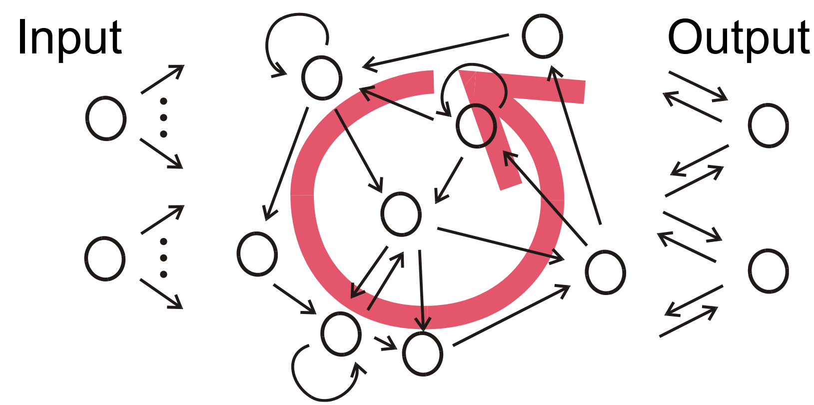
Koichi Suzumori, Kenjiro Fukuda, Ryuma Niiyama, Kohei Nakajima, ***The Science of Soft Robots---Design, Materials and Information Processing***, Springer Singapore, 2023. ISBN: 978-981-19-5173-2

Recurrent Neural Networks

Feedforward NN (FNN) vs Recurrent NN (RNN)

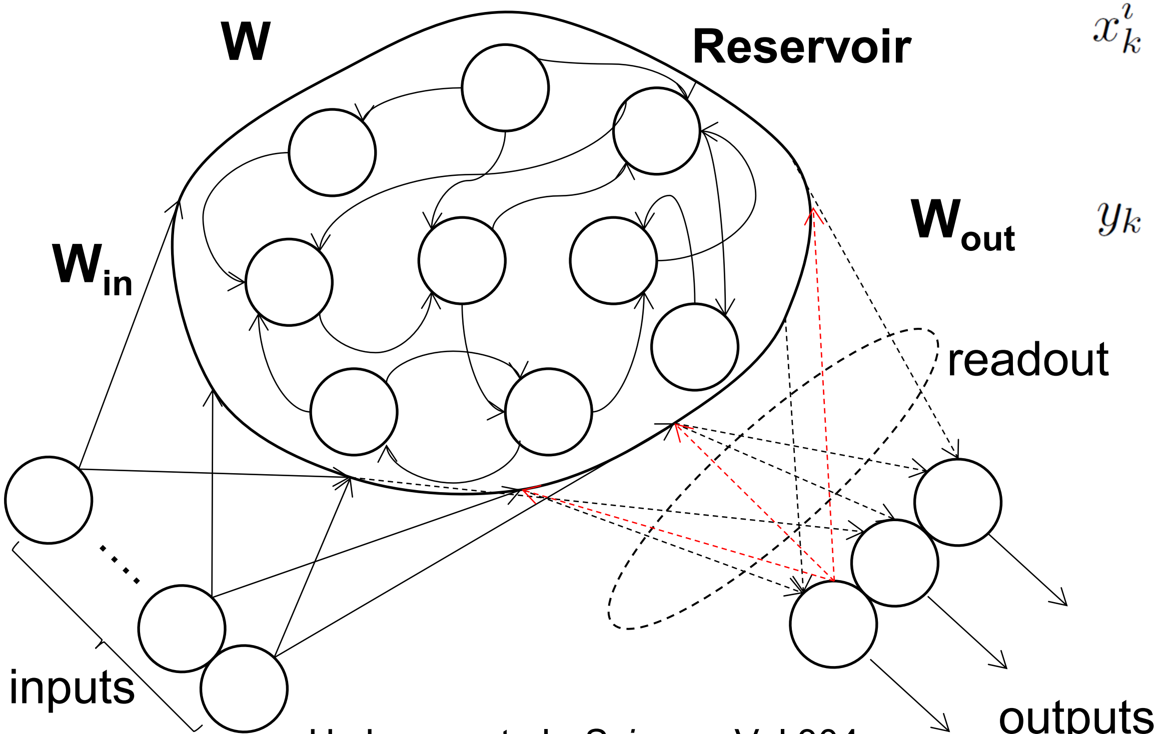


- Activation is fed forward from input to output via hidden layers
- Can approximate arbitrary nonlinear static maps with arbitrary precision
- Static (e.g., image processing)



- Has at least one cyclic path in synaptic connections (memory)
- Can approximate arbitrary nonlinear dynamical systems with arbitrary precision
- Dynamic (e.g., prediction tasks for time series)

Reservoir computing: basic settings



H. Jaeger et al., *Science*, Vol.304. no.5667, pp.78–80 (2004).

$$x_k^i = f(\sum_{j=1}^{M} w^{ij} x_{k-1}^j + w_{in}^i u_k)$$

$$y_k = \sum_{i=0}^{M} w_{out}^i x_k^i, \; f(x) = \tanh(x)$$

Adjust only the readout!

$$W_{out} = (X^T X)^{-1} X^T y$$

Use W_{out} for information processing!

$$\hat{y} = X W_{out}$$

(Good points)

- Learning is fast and stable!
- Feasible for physical platform.

(Computational power)

- **Nonlinearity**
- **Memory**

Two representative models in RC

Echo state network

H. Jaeger, Tech. Rep. No. 148. Bremen: German National Research Center for Information Technology (2001).

H. Jaeger et al., Science, Vol.304. no.5667, pp.78–80 (2004).

Liquid state machine

W. Maass et al., Neural Comput 14 (11): 2531–60, 2002.

W. Maass, & H. Markram, H. Journal of computer and system sciences, 69(4), 593-616, 2004.

Early 2000

Herbert Jaeger

- Randomly coupled network
- Artificial neural network (Sigmoidal function)
- Engineering oriented

Wolfgang Maass

- Often assume space
- Pulse neuron
- Neuroscience oriented

∗ Similar architectures can be found at least in 1990.

Jaeger, H. (2021). In Reservoir Computing. Springer Nature.

Conception in around 2005!

Let us unify the approach in the same umbrella!

Reservoir computing

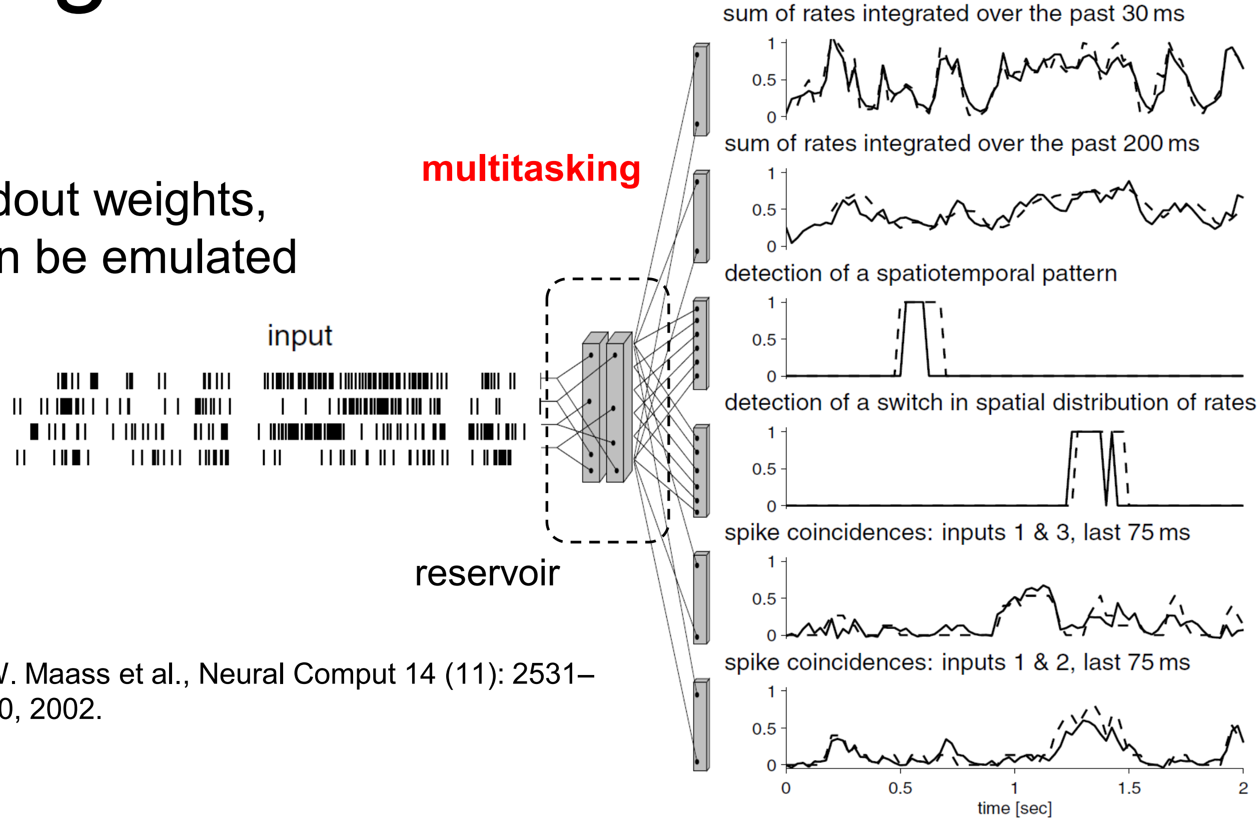
Benjamin Schrauwen, Joni Dambre (University of Gent)

Typical settings

- **Open-loop**

By attaching the readout weights, multiple functions can be emulated simultaneously!

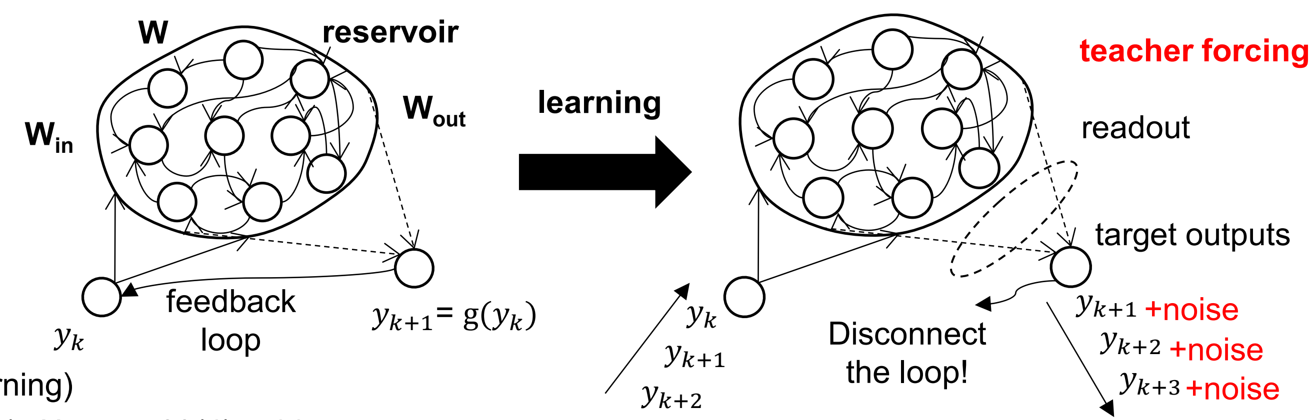
$$x_{k+1} = f(x_k, u_k)$$
$$\hat{y}_k = \hat{\psi}(x_k)$$



W. Maass et al., Neural Comput 14 (11): 2531–60, 2002.

- **Close-loop**

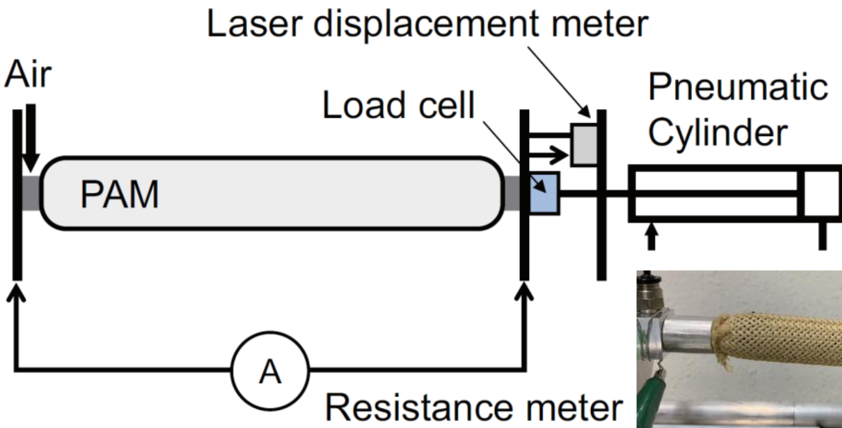
$$\hat{x}_{k+1} = f(\hat{x}_k, \hat{u}_k)$$
$$\hat{u}_k = \hat{\psi}(\hat{x}_k)$$

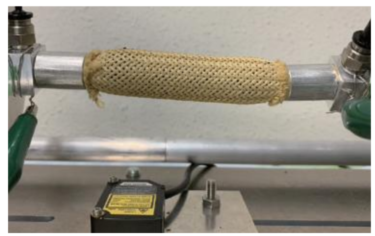


*On-line learning
(e.g. FORCE learning, Innate learning)

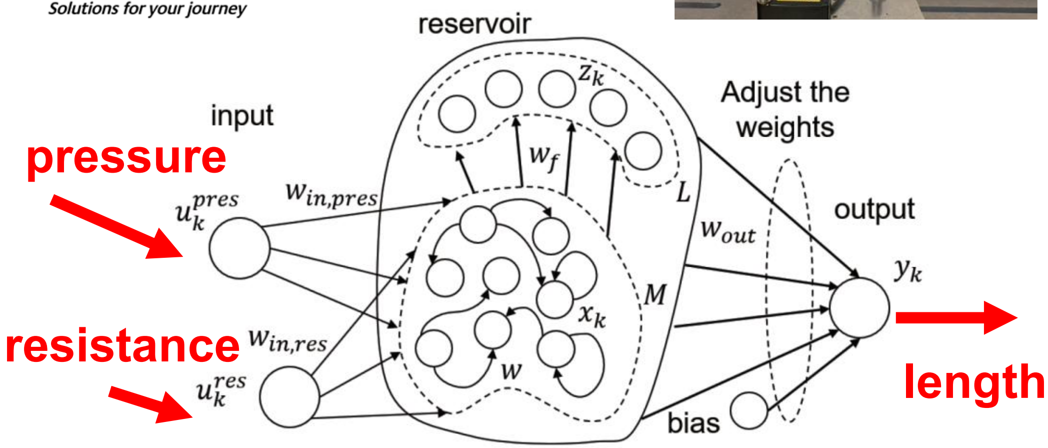
Sussillo, D., & Abbott, L. F. (2009). *Neuron*, 63(4), 544-557.

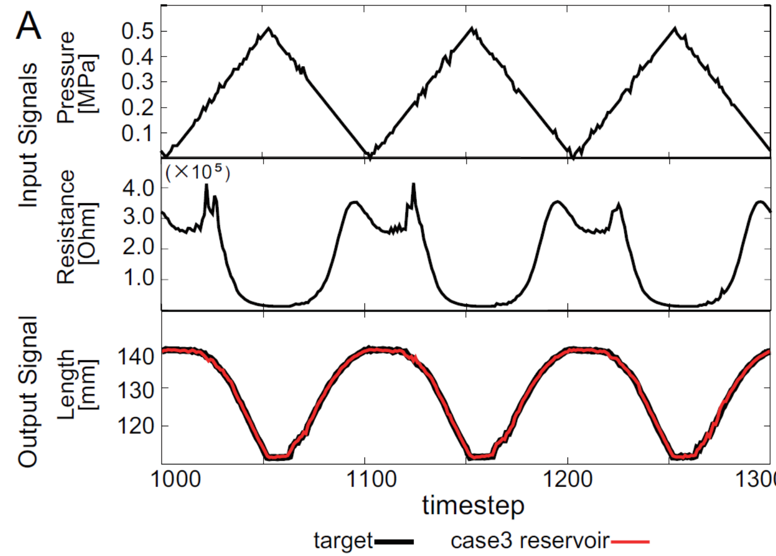
Soft sensing using material dynamics

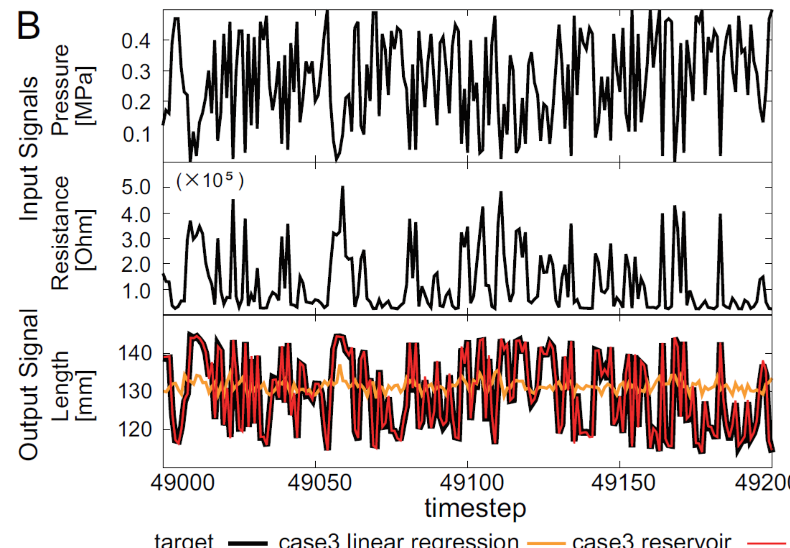




BRIDGESTONE
Solutions for your journey



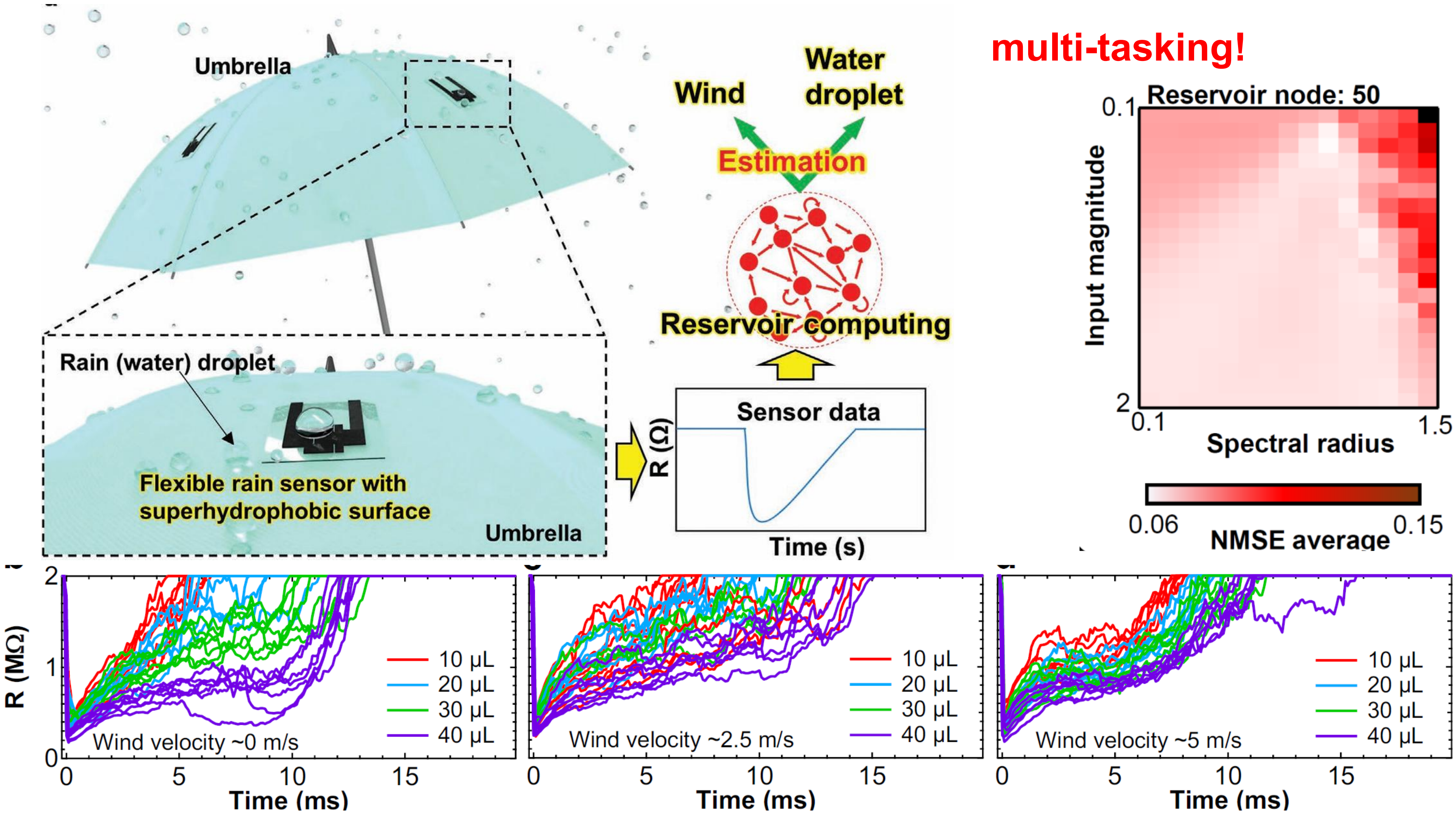




R. Sakurai, et al., Proc. of 2020 3rd IEEE RoboSoft, pp. 710-717, 2020.
W. Sun, et al., Proc. of 2022 IEEE 5th RoboSoft, pp. 409-415, 2022.
N. Akashi, et al., Adv. Intell. Syst. 4: 2200123 (2022).

- Emulating a laser displacement meter in a high precision!
- Using conducting rubbers and do not need to attach the rigid sensors!

Reservoir computing meets flexible sensors

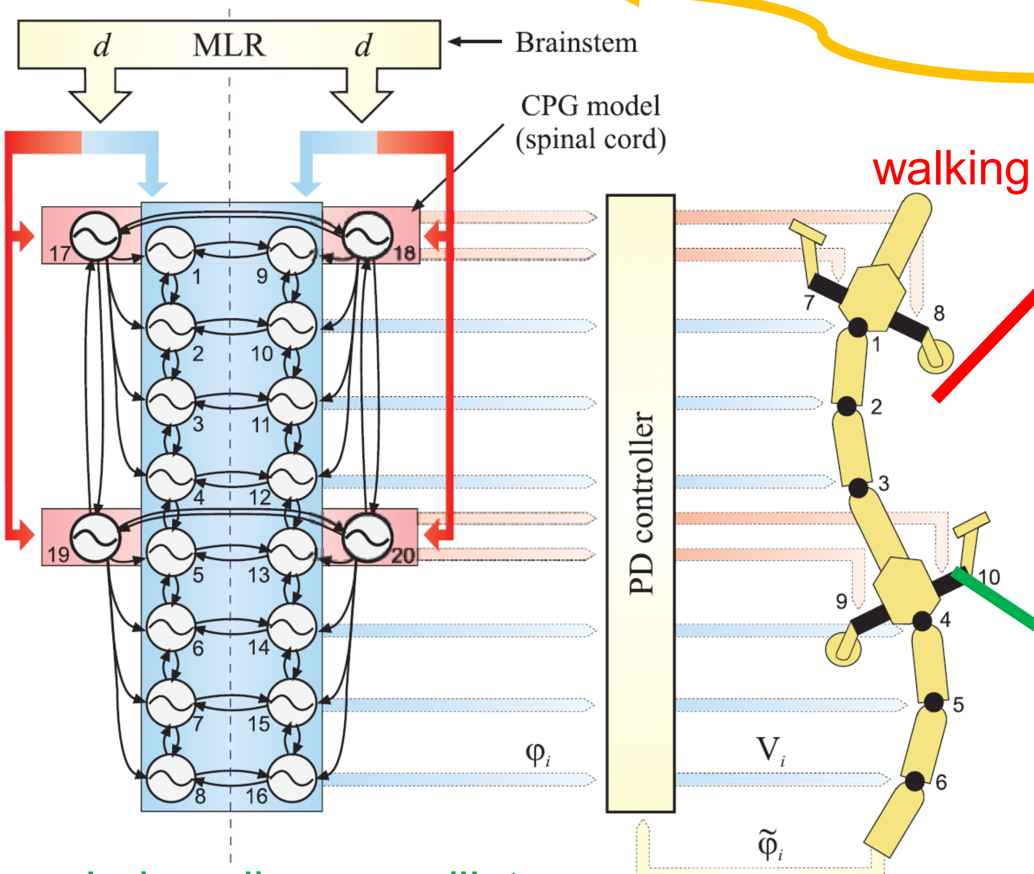


S. Wakabayashi, T. Arie, S. Akita, K. Nakajima, K. Takei, A multi-tasking flexible sensor via reservoir computing, ***Advanced Materials***, 2201663, 2022.
N. Seimiya, K. Uehara, et al., Water-Dynamics Monitoring Using a Flexible Resistive Sensor and Reservoir Computing. ***Small***, 2407698, 2025.

Multi-tasking is easy and learning is quick!

Designing locomotion patterns and switching them

Sensory inputs



walking

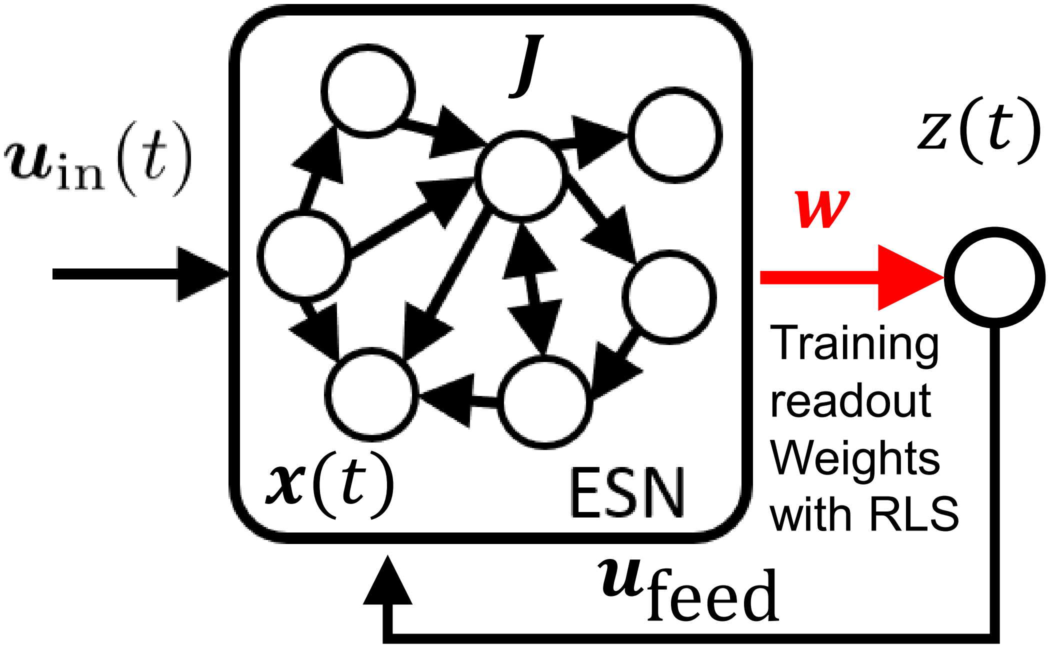
swimming

Coupled nonlinear oscillators

Ijspeert, A. J., Crespi, A., Ryczko, D., & Cabelguen, J. M. (2007). From swimming to walking with a salamander robot driven by a spinal cord model. Science, 315(5817), 1416-1420.

- Locomotion through central pattern generators!
- Switching the locomotion patterns via external stimuli! (e.g., sensors or external controllers)

Locomotion control: emulating oscillators

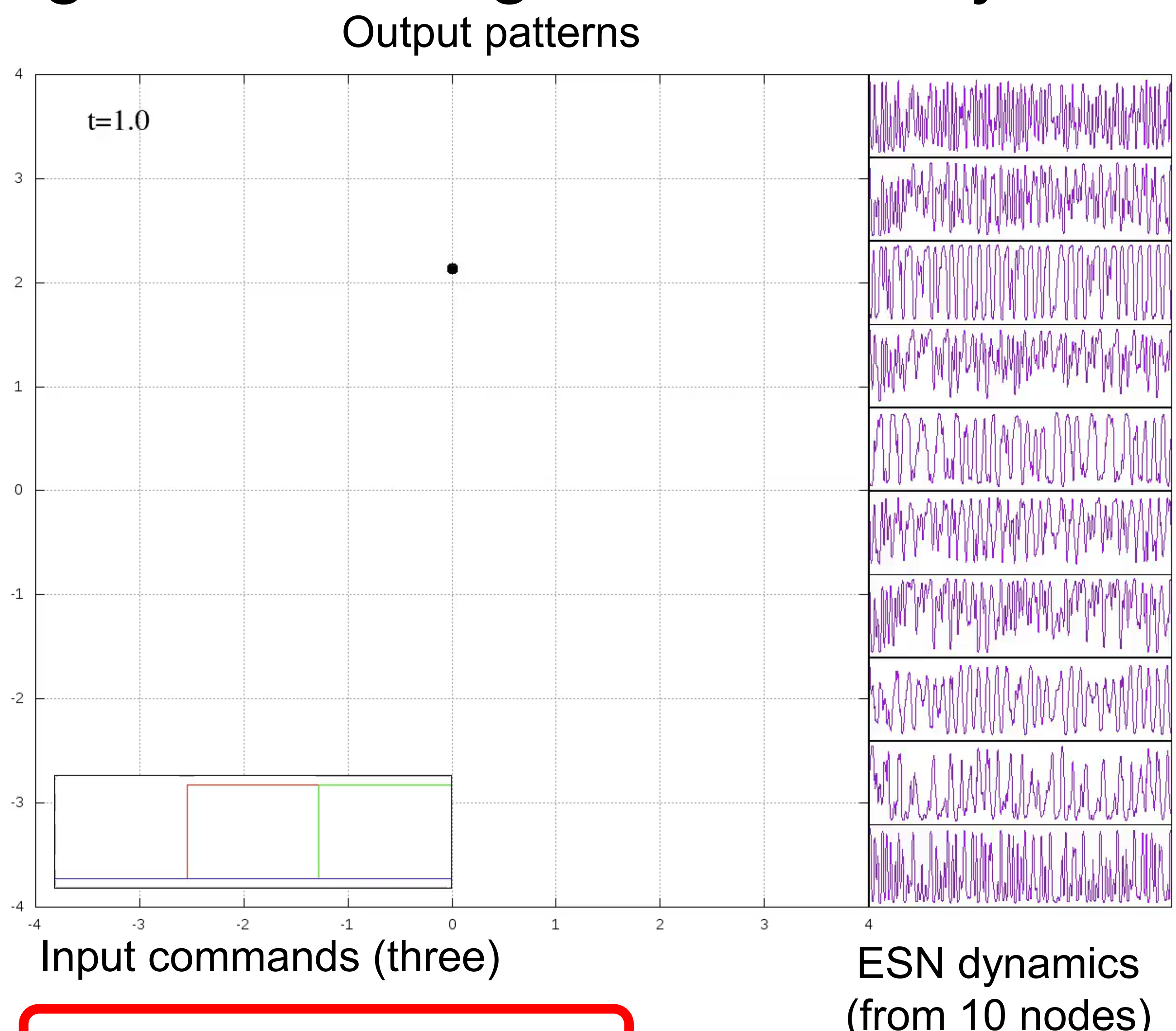


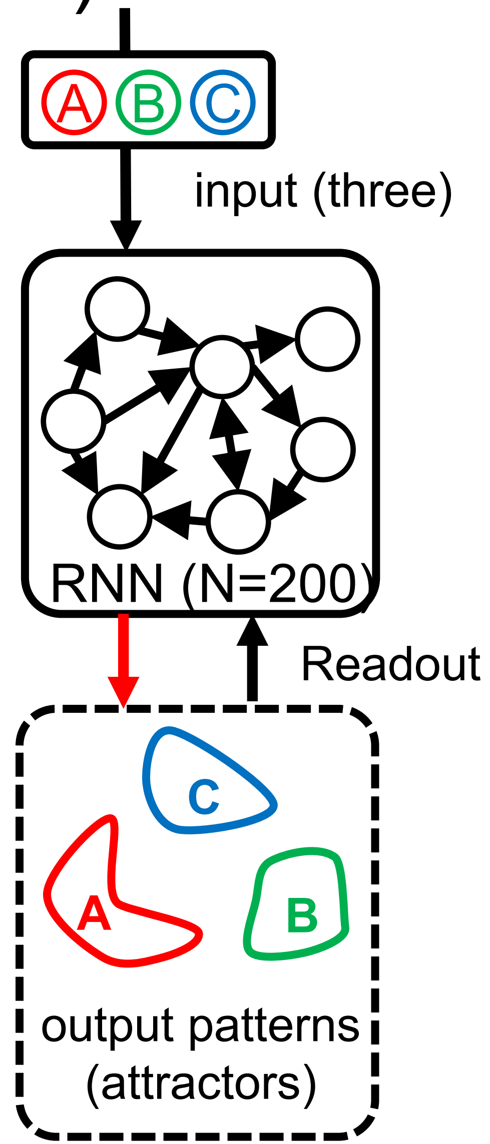
τ: time constant,
$\boldsymbol{x}$: network state, $\boldsymbol{x} \in \mathbb{R}^N$
g: nonlinearity parameter ($g > 1$: chaotic)
$\boldsymbol{J}$: $N \times N$ matrix, $\boldsymbol{J} \sim \mathcal{N}\left(0, \frac{1}{N}\right)$
$\boldsymbol{u}_{\mathrm{in}}$: input function, $\boldsymbol{u}_{\mathrm{in}}: \mathbb{R} \to \mathbb{R}^N$
$\boldsymbol{u}_{\mathrm{feed}}$: $N \times 1$ matrix, $\boldsymbol{u}_{\mathrm{feed}} \sim [-1, 1]$
$\boldsymbol{w}$: readout weights, $N \times 1$ matrix

Inoue, K., Nakajima, K., Kuniyoshi, Y. Proceedings of NOLTA2018, pp. 412-414, 2018.
Inoue, K., Nakajima, K., Kuniyoshi, Y. (2020). Designing spontaneous behavioral switching via chaotic itinerancy. *Science Advances* 6 (46), eabb3989.

$$\tau \frac{d\boldsymbol{x}(t)}{dt} = -\boldsymbol{x}(t) + \tanh\left(g\boldsymbol{J}\boldsymbol{x}(t) + \boldsymbol{u}_{\mathrm{feed}} z(t) + \boldsymbol{u}_{in}(t)\right)$$
$$z(t) = \boldsymbol{w}^T \boldsymbol{x}(t)$$

Designing attractors according to inputs
(e.g., embedding CPGs into dynamics)

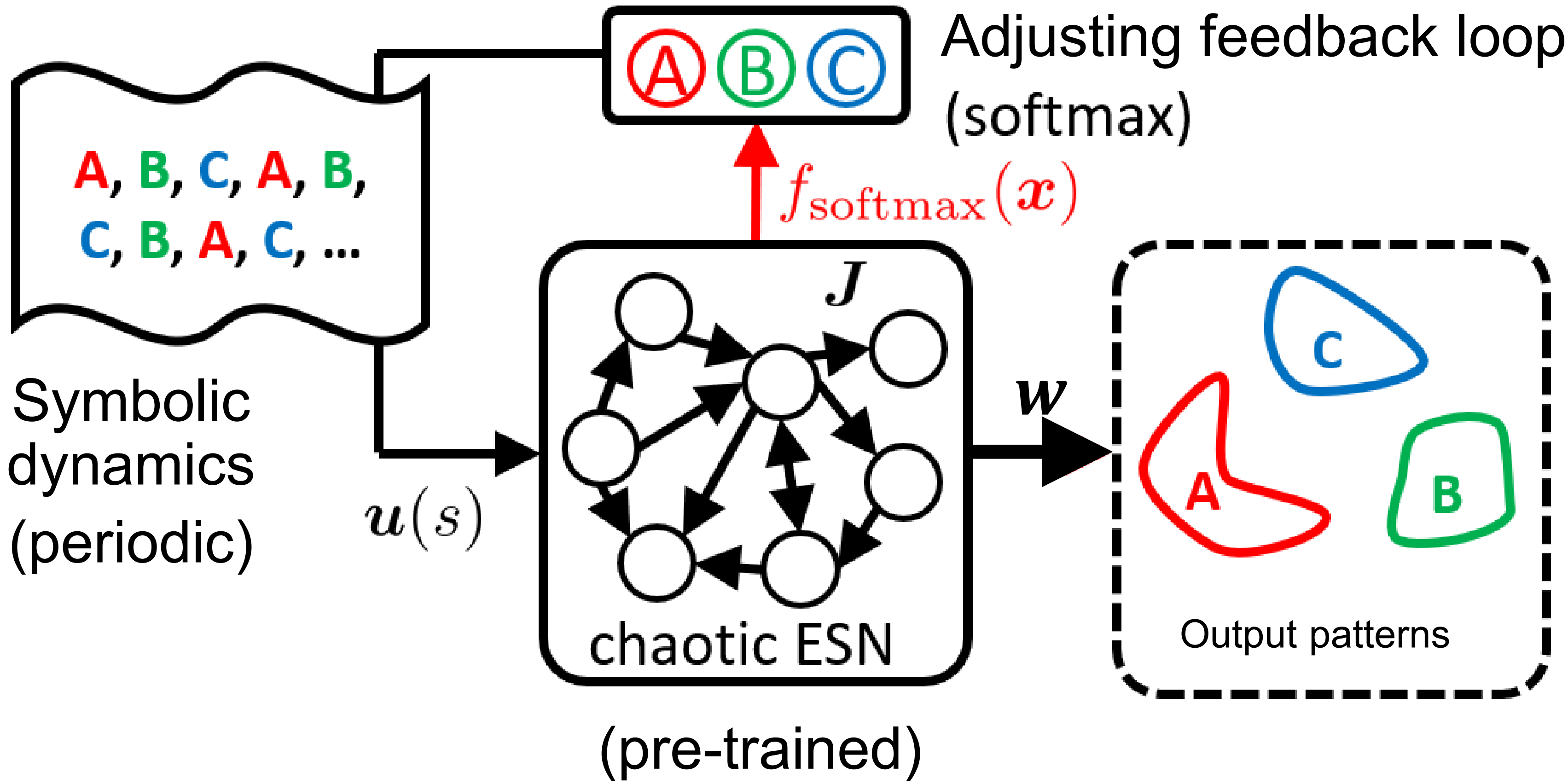




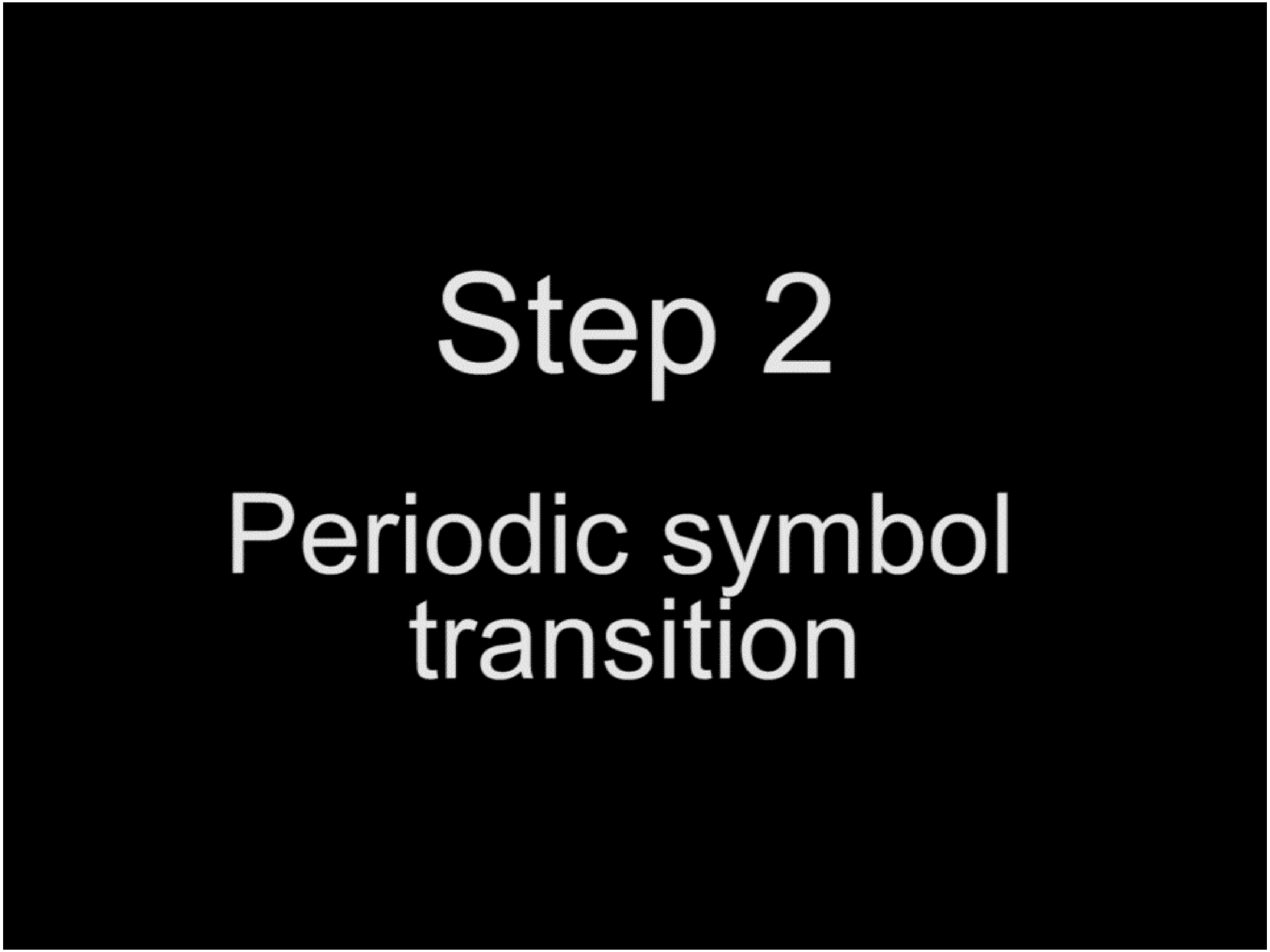
High operability!

K. Inoue, K. Nakajima, Y. Kuniyoshi, in Proceedings of NOLTA2018, pp. 412-414, 2018.

Switching embedded patterns in a periodic manner

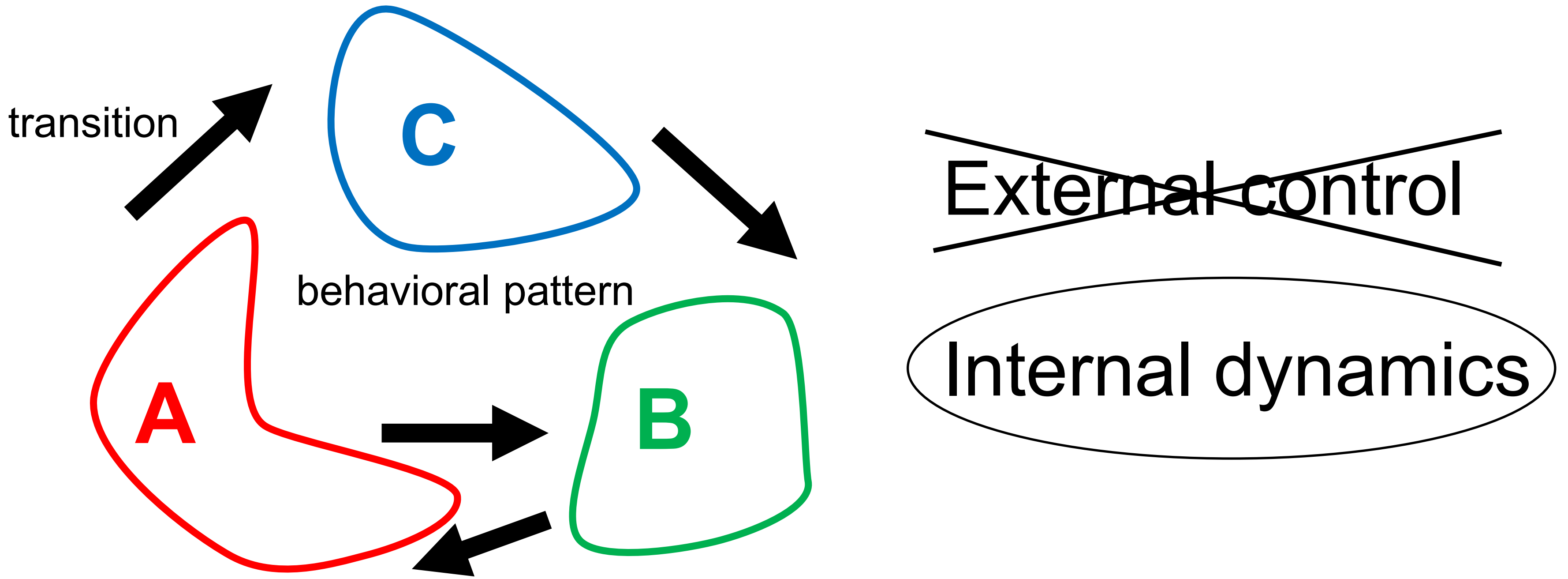


- Internalizing the external control through feedback loop!
- Should estimate the duration of time for each pattern (switch in appropriate timing) using the same reservoir!



Inoue, K., Nakajima, K., & Kuniyoshi, Y. (2020). Designing spontaneous behavioral switching via chaotic itinerancy. *Science Advances* 6 (46), eabb3989.

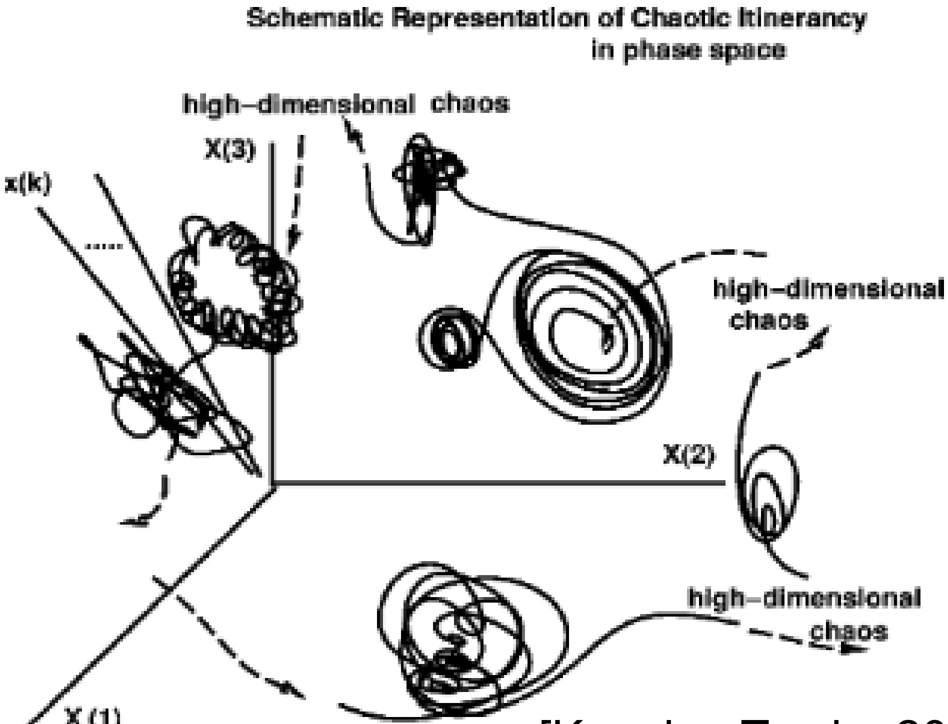
Designing spontaneous behavioral switching



- **Step 1**: Behavioral patterns
- **Step 2**: Periodic transitions among the patterns
  (challenge: **embed timer**)
- **Step 3**: Random transitions among the patterns
  (challenge: **embed timer + random number generators**)

Step 3 is related to chaotic itinerancy!

What is chaotic itinerancy?



[Kaneko, Tsuda, 2003]

FIG. 1. Schematic representation of chaotic itinerancy.

First found in…

- Optical turbulence

  [K. Ikeda et. al., 1989]

- A globally coupled chaotic system

  [K. Kaneko, 1990; 1991]

- Nonequilibrium neural networks

  [I. Tsuda, 1991; 1992]

(Features)

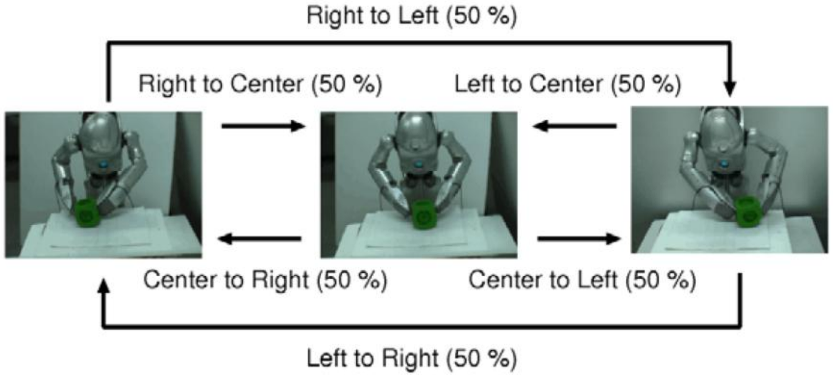
- Frequently observed in high-dimensional nonlinear dynamical systems
- Seemingly random transitions among quasi-attractors

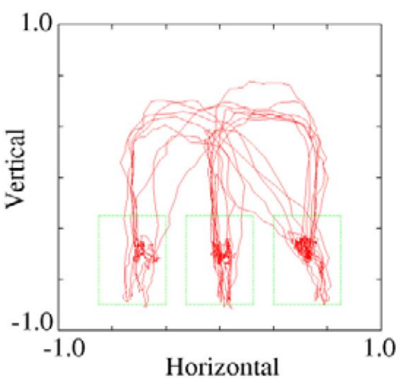
→ Propose a scheme to design CI!

Spontaneous behavioral switching in robots

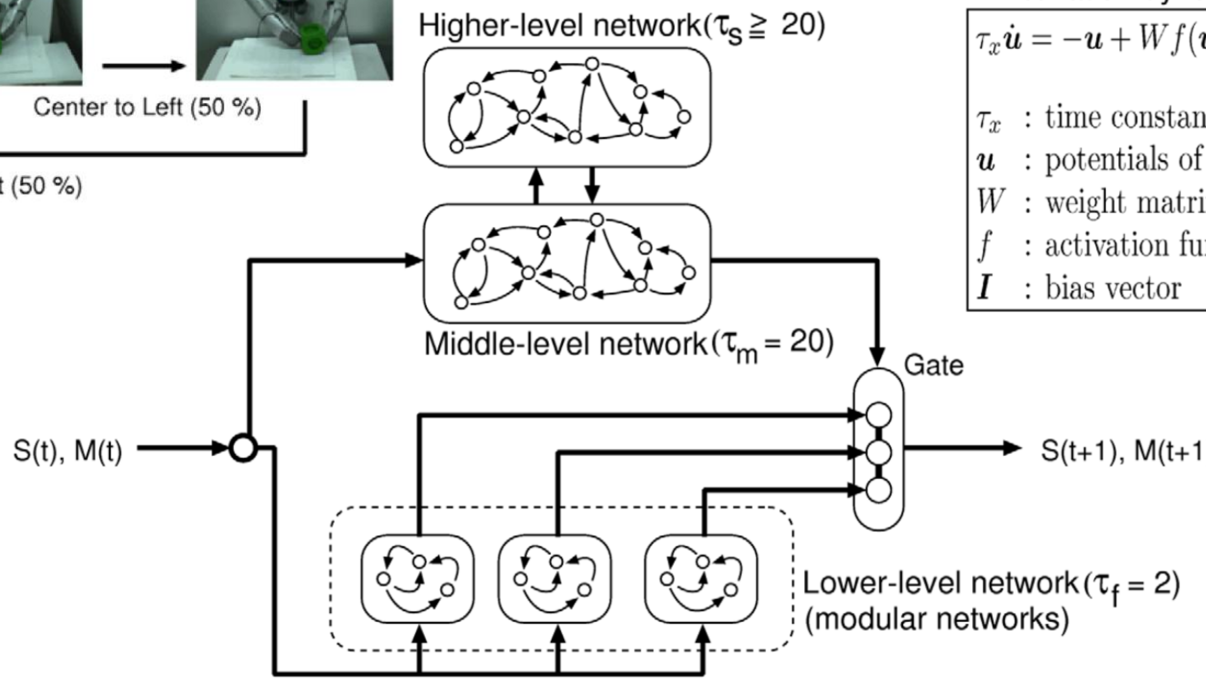
Namikawa, J., Nishimoto, R., & Tani, J. (2011). A neurodynamic account of spontaneous behaviour. *PLoS computational biology*, *7*(10), e1002221.

Behavioral switching





Network architecture



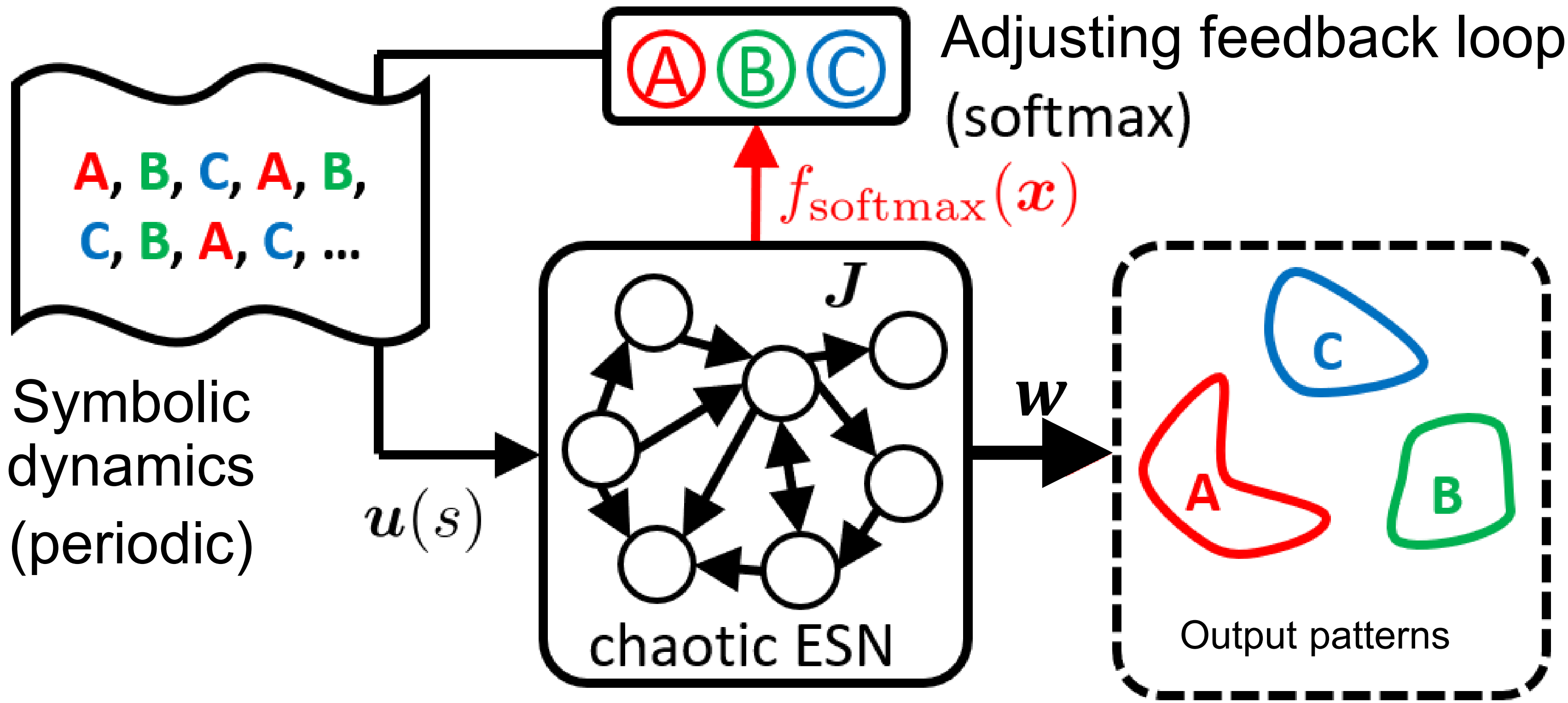
Activation dynamics of the unit

$$\tau_x \dot{\boldsymbol{u}} = -\boldsymbol{u} + W f(\boldsymbol{u}) + \boldsymbol{I}$$

τ_x : time constant (x is s, m, or f)
$\boldsymbol{u}$: potentials of neural units
W : weight matrix
f : activation function of the unit
$\boldsymbol{I}$: bias vector

- Deterministic chaos self-organized to generate stochastic processes.
- Using hierarchical modules beforehand!

Switching patterns autonomously



- Internalizing the external control through feedback loop!
- Should estimate the duration of time for each pattern (switch in appropriate timing) using the same reservoir!

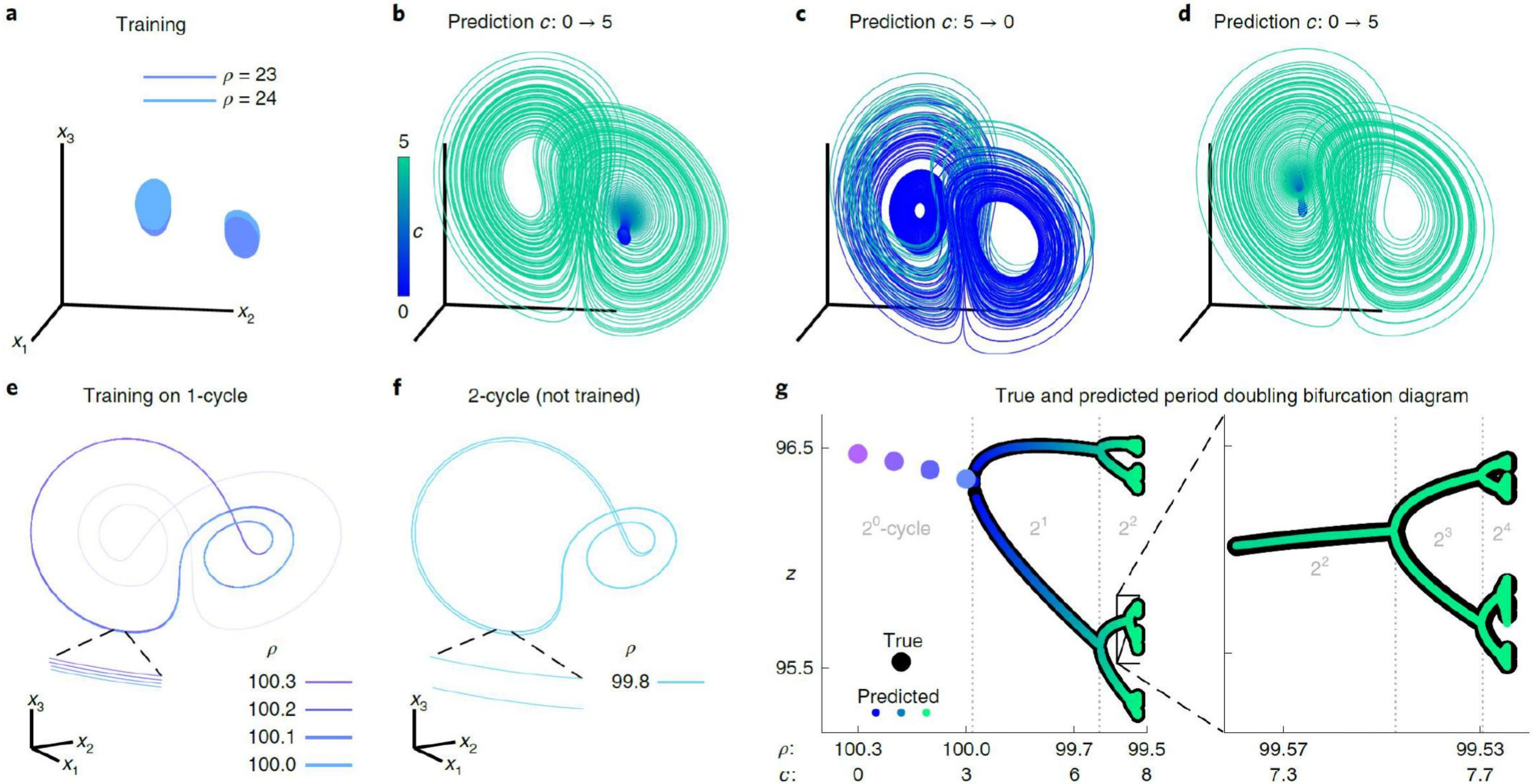
Inoue, K., Nakajima, K., & Kuniyoshi, Y. (2020). Designing spontaneous behavioral switching via chaotic itinerancy. *Science Advances* 6 (46), eabb3989.

Step 3

Designing chaotic itinerancy

Inoue, K., Nakajima, K., & Kuniyoshi, Y. (2020). Designing spontaneous behavioral switching via chaotic itinerancy. *Science Advances* 6 (46), eabb3989.

Learning bifurcations from limited examples through extrapolation



Kim, J. Z., Lu, Z., Nozari, E., Pappas, G. J., & Bassett, D. S. (2021). Teaching recurrent neural networks to infer global temporal structure from local examples. *Nature Machine Intelligence*, *3*(4), 316-323.

- Only by learning several attractors, the reservoir system can learn entire bifurcation structure!

*Prerequisite: "*reproducible response*"

(target function)

$$y_{k+1} = g(u_k, u_{k-1}, \dots)$$

(reservoir computing)

$$\begin{cases} \boldsymbol{x}_{\boldsymbol{k}+\boldsymbol{1}} = f(\boldsymbol{x}_{\boldsymbol{k}}, u_k) \\ \hat{y}_{k+1} = \hat{\psi}(\boldsymbol{x}_{\boldsymbol{k}+\boldsymbol{1}}) \end{cases}$$

$$g(u_k, u_{k-1}, \dots) \approx \hat{\psi}(\boldsymbol{x}_{\boldsymbol{k}+\boldsymbol{1}})$$

We want to emulate (learn) function "g"!

(prerequisite)

- Reproducible response to the same input sequence!
- Reservoir states should not depend on the initial condition!

$$f(\boldsymbol{x}_{\boldsymbol{k}}, u_k) - f(\boldsymbol{x}^*_{\boldsymbol{k}}, u_k) \approx 0 \longleftarrow \boldsymbol{x}_{\boldsymbol{k}} = \boldsymbol{\phi}(u_{k-1}, u_{k-2}, \dots)$$

(common-signal-induced synchronization/ Negative conditional Lyapunov exponents)

Z. Lu, B. R. Hunt, E. Ott, *Chaos* 28, 061104 (2018).

(echo state property (ESP))

H. Jaeger, GMD Technical Report. 148 (2001).
I. B. Yildiz, et al., *Neural netw.* 35 (2012).
G. Manjunath, et al., *Neural comp.* 25 (2013).

One dimensional and linear examples

E.g.) $x_{k+1} = -\frac{1}{2}x_k + u_k$

$$x_{\boldsymbol{k}} = \left(-\frac{1}{2}\right)^{k-1} x_0 + u_{k-1} - \frac{1}{2}u_{k-2} + \cdots + \left(-\frac{1}{2}\right)^{k-1} u_0$$

$\mathrm{k} \to \infty$

$$x_{\boldsymbol{k}} = u_{k-1} - \frac{1}{2}u_{k-2} + \cdots$$

Independent of initial state

$$\boldsymbol{x_k} = \boldsymbol{\phi}(u_{k-1}, u_{k-2}, \ldots)$$

E.g.) $x_{k+1} = -x_k + u_k$

$$x_{\boldsymbol{k}} = (-1)^{k-1} x_0 + u_{k-1} - u_{k-2} + \cdots + (-1)^{k-1} u_0$$

No ESP

$$\boldsymbol{x_k} = \boldsymbol{\phi}(u_{k-1}, u_{k-2}, \ldots, x_0, k)$$

- Chaotic dynamics, limit cycles, trends, noise, … No ESP!

General solution of $x_{k+1} = f(x_k, u_k)$:

$$\boldsymbol{u}_k = \{u_{k-1}, u_{k-2}, \dots, u_0\},\ \boldsymbol{x}_k = \boldsymbol{\phi}(\boldsymbol{u}_k, x_0, k)$$

- $\boldsymbol{x}_k = \boldsymbol{\phi}(\boldsymbol{u}_k)$

  Has ESP. It can be reliably used as reservoir!

- $\boldsymbol{x}_k = \boldsymbol{\phi}(\boldsymbol{u}_k, x_0, k)$

  No ESP. Depends on both initial state and inputs.

- $\boldsymbol{x}_k = \boldsymbol{\phi}(\boldsymbol{u}_k, \boldsymbol{u}'_k, x_0, k)$

  No ESP. Multiple inputs case, such as noise…

Are these dynamics useless for computation?
-> Recent hot topic!

Kubota, T., Imai, Y., Tsunegi, S., & Nakajima, K. (2024). Reservoir Computing Generalized. *arXiv preprint arXiv:2412.12104*.

Kubota, T., Takahashi, H., & Nakajima, K. (2021). Unifying framework for information processing in stochastically driven dynamical systems. *Physical Review Research*, *3*(4), 043135.

Chaos and Lyapunov exponent

1d-DS $\boldsymbol{x(n+1) = f(x(n))}$

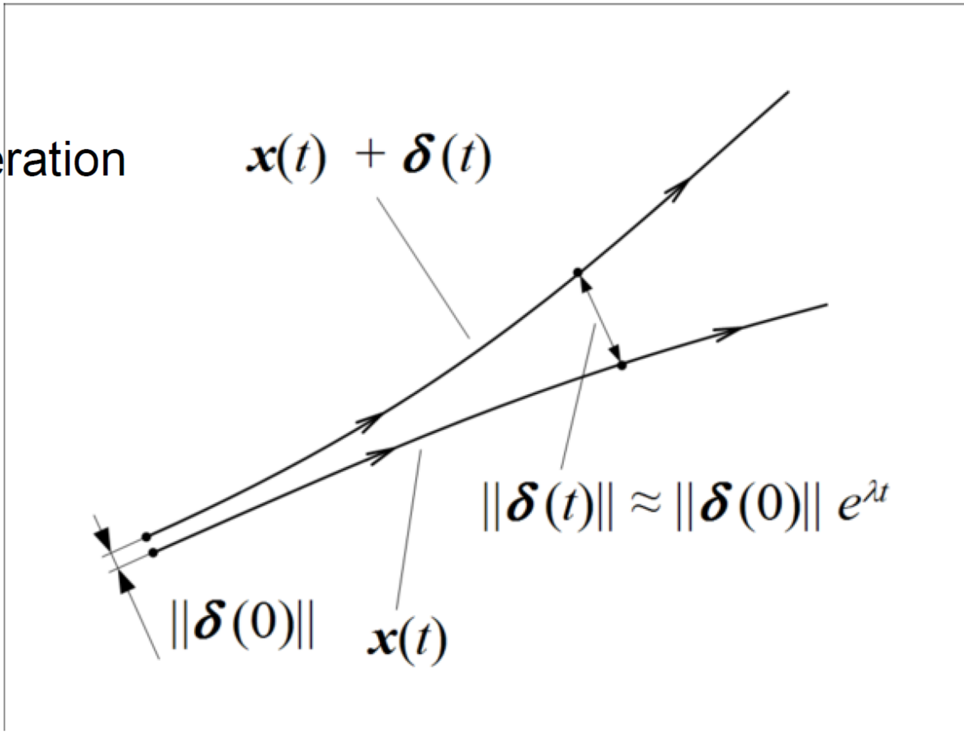
How the tiny distance expands through n iteration

$$|\boldsymbol{\delta(n)}| = |\boldsymbol{\delta(0)}| e^{\lambda n}$$

$$\lambda = \frac{1}{n} \ln \frac{|\boldsymbol{\delta(n)}|}{|\boldsymbol{\delta(0)}|}$$

$|\boldsymbol{\delta(0)}| \longrightarrow 0, n \longrightarrow \infty$

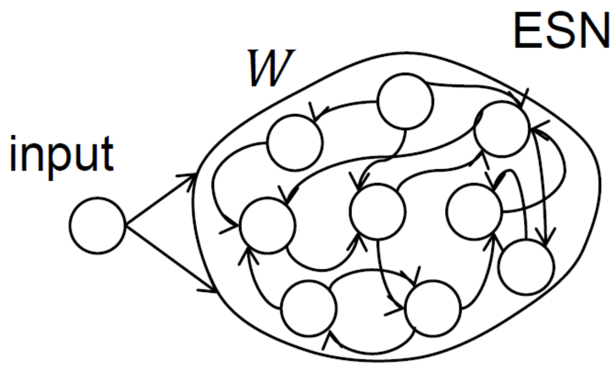
$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln|f'(x(i))|$$



(Wikipedia)

- Sensitive to initial condition!
- Small perturbation expands exponentially, characterized by positive Lyapunov exponents!

Echo-state network (ESN)



- number of nodes : N
- state of neuron i at t : $x_i(t)$
- action potential of neuron i at t : $a_i(t)$
- input : $u(t)$
- internal weights : w_{ij}
- input weights : w_{in}
- activation function : $f(a) = \tanh(a)$
- dynamics :

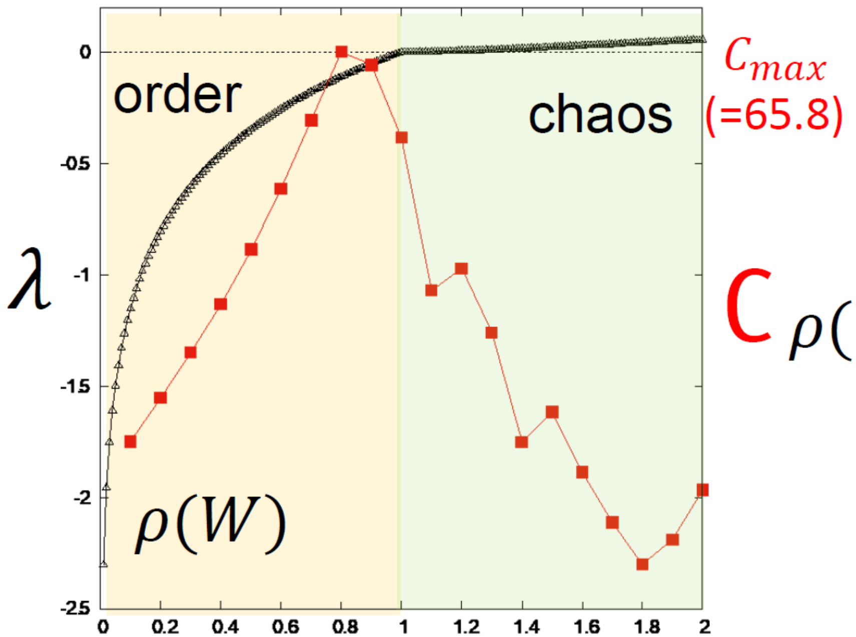
spectral radius of W

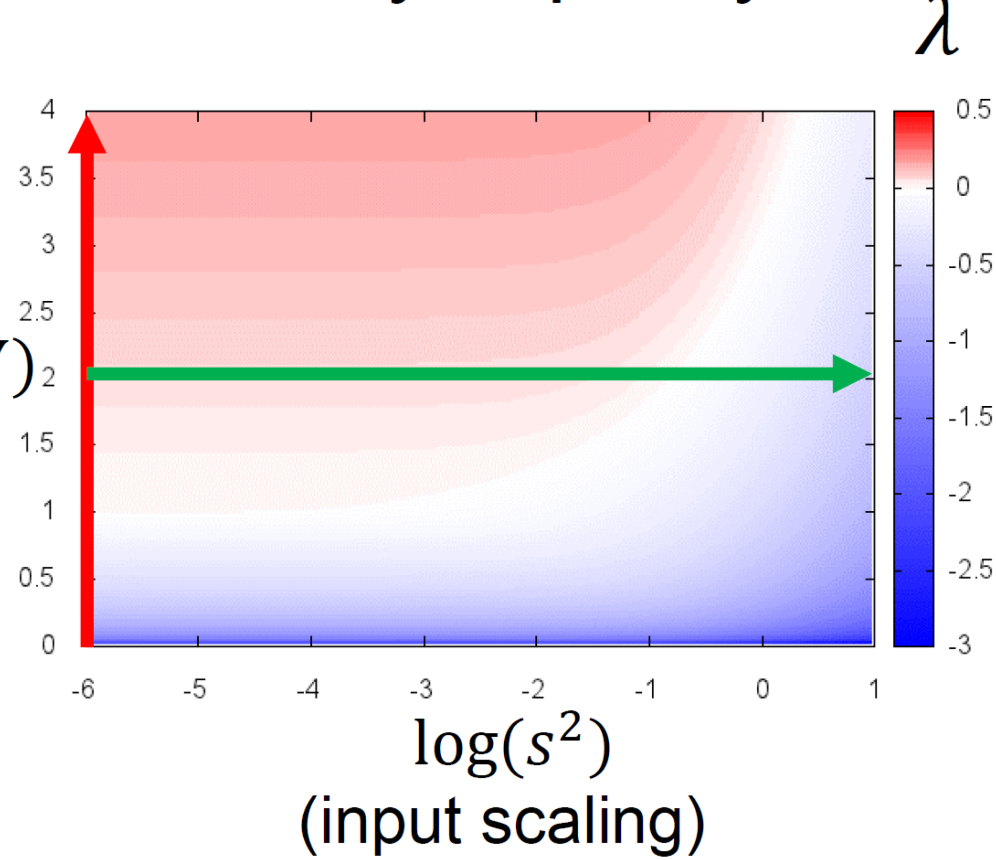
$$\rho(W) = \max\{|\lambda_1|, \dots, |\lambda_N|\}$$

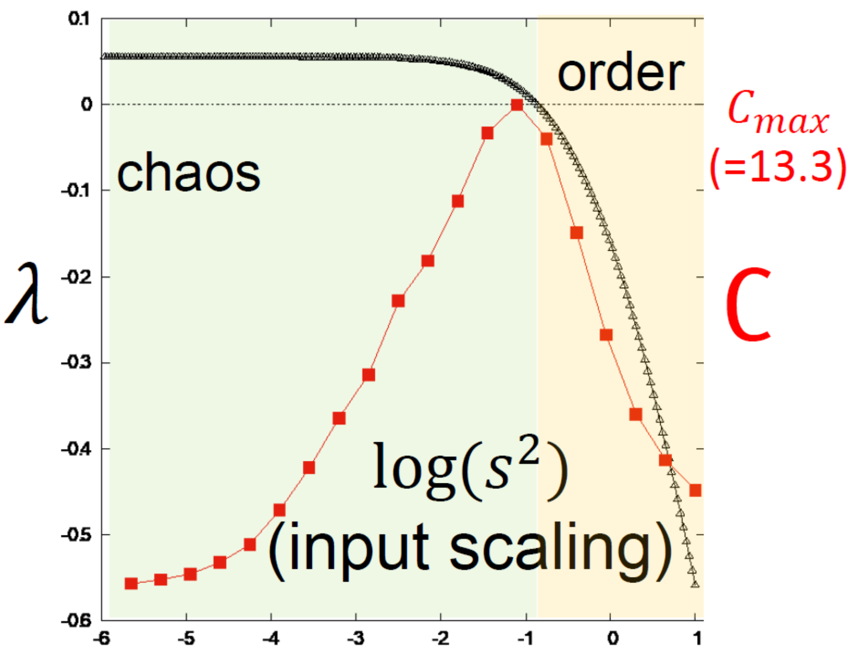
$$x_i(t+1) = f\big(a_i(t)\big), \quad a_i(t) = \sum_{j=1}^{N} w_{ij} x_j(t) + w_{in} u(t)$$

Jaeger, H. (2002). Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach (Vol. 5, p. 01). Bonn: GMD-Forschungszentrum Informationstechnik.

Bifurcation of ESN and its memory capacity







C is maximized around the bifurcation point!

(C is normalized in the left plots)

Attempts to use chaotic dynamics:

- Spontaneous behavioral switching

  Inoue, K., Nakajima, K., & Kuniyoshi, Y. (2020). Designing spontaneous behavioral switching via chaotic itinerancy. Science advances, 6(46), eabb3989.

  Kabayama, T., Komuro, M., Kuniyoshi, Y., Aihara, K., & Nakajima, K. (2025). Crisis-induced intermittency in reservoir computing. *Physical Review Research*, *7*(3), L032058.

- Introduce time invariant transformation

  Kubota, T., Imai, Y., Tsunegi, S., & Nakajima, K. (2024). Reservoir Computing Generalized. arXiv preprint arXiv:2412.12104.

- Designing shape of chaotic attractors

  Kabayama, T., Kuniyoshi, Y., Aihara, K., & Nakajima, K. (2025). Designing chaotic attractors: A semisupervised approach. Physical Review E, 111(3), 034207.

- Chaotic dynamics as deep NN

  Liu, S., Akashi, N., Huang, Q., Kuniyoshi, Y., & Nakajima, K. (2025). Exploiting chaotic dynamics as deep neural networks. *Physical Review Research*, *7*(3), 033031.

Attempts to use chaotic dynamics:

- Spontaneous behavioral switching

  Inoue, K., Nakajima, K., & Kuniyoshi, Y. (2020). Designing spontaneous behavioral switching via chaotic itinerancy. Science advances, 6(46), eabb3989.

  Kabayama, T., Komuro, M., Kuniyoshi, Y., Aihara, K., & Nakajima, K. (2025). Crisis-induced intermittency in reservoir computing. *Physical Review Research*, *7*(3), L032058.

- Introduce time invariant transformation

  Kubota, T., Imai, Y., Tsunegi, S., & Nakajima, K. (2024). Reservoir Computing Generalized. arXiv preprint arXiv:2412.12104.

- Designing shape of chaotic attractors

  Kabayama, T., Kuniyoshi, Y., Aihara, K., & Nakajima, K. (2025). Designing chaotic attractors: A semisupervised approach. Physical Review E, 111(3), 034207.

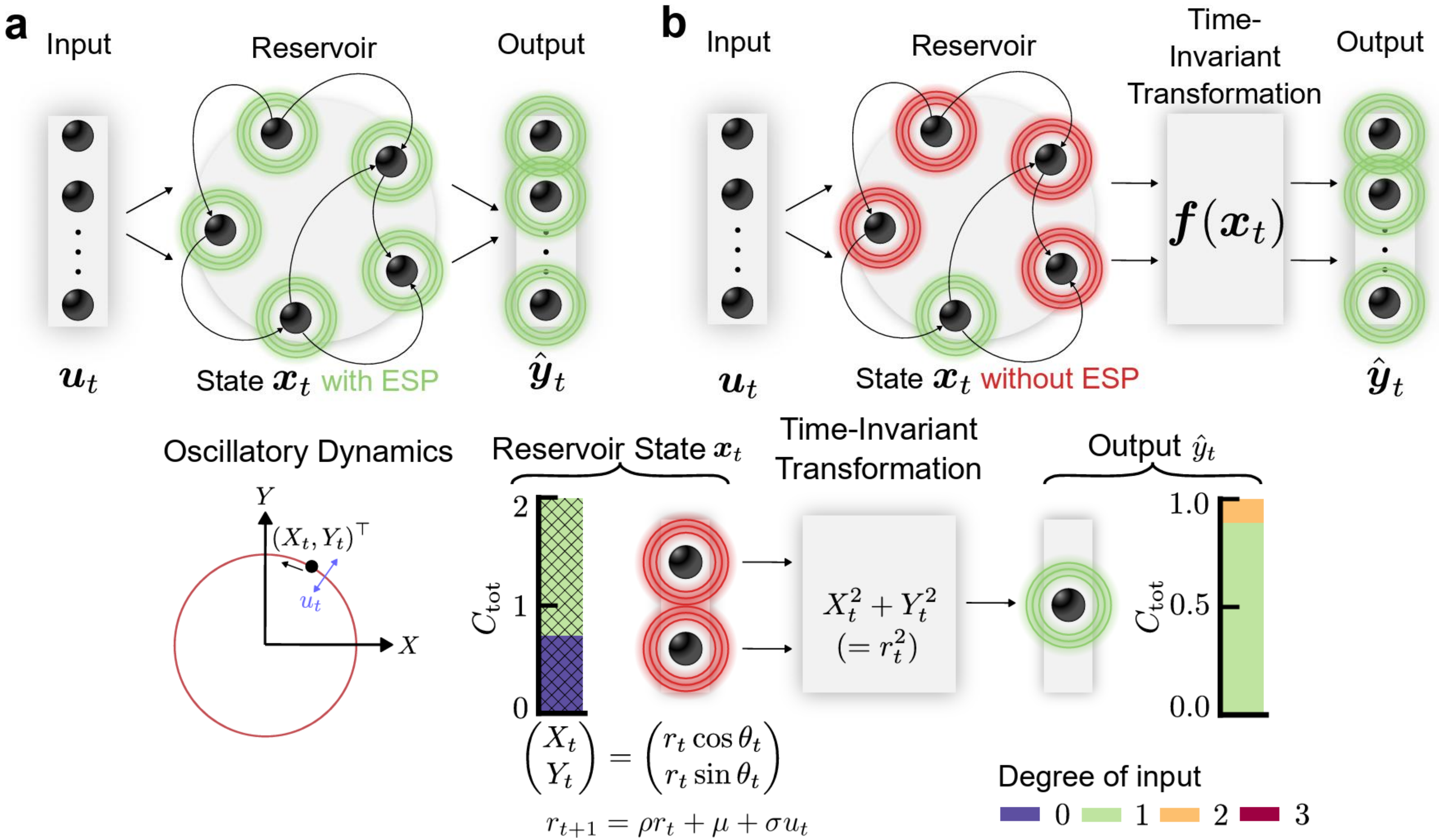
- Chaotic dynamics as deep NN

  Liu, S., Akashi, N., Huang, Q., Kuniyoshi, Y., & Nakajima, K. (2025). Exploiting chaotic dynamics as deep neural networks. *Physical Review Research*, *7*(3), 033031.

Generalized RC

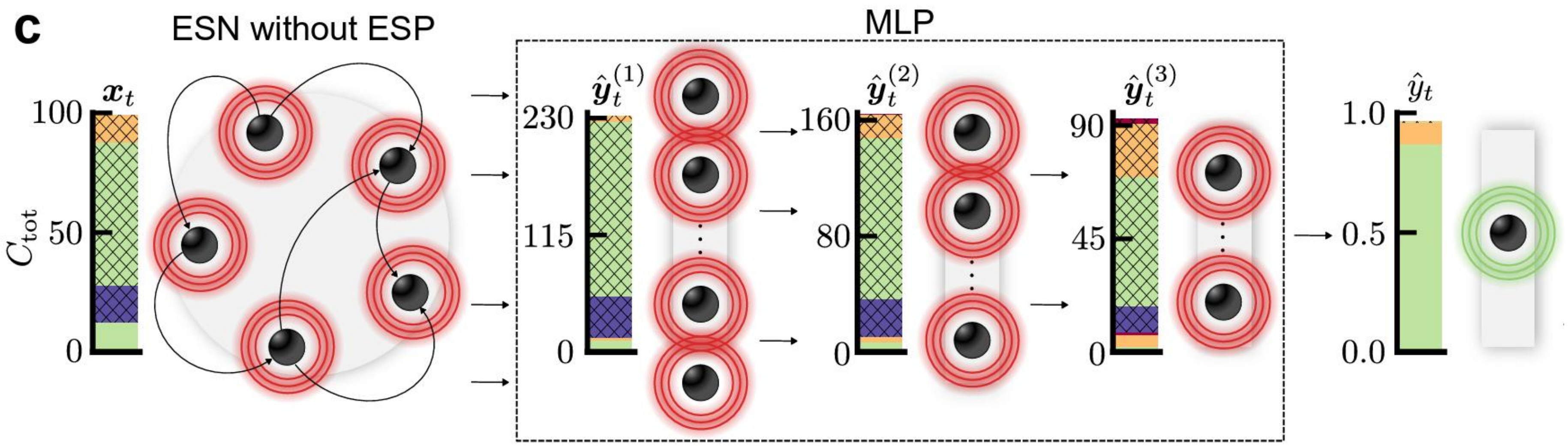
Kubota, T., Imai, Y., Tsunegi, S., & Nakajima, K. (2024). Reservoir Computing Generalized. arXiv preprint arXiv:2412.12104.

RC generalized through time-invariant transformation



- Non-ESP reservoir is still processing inputs!
- Improving ESP at readout layer!

ESN example with DNN readout

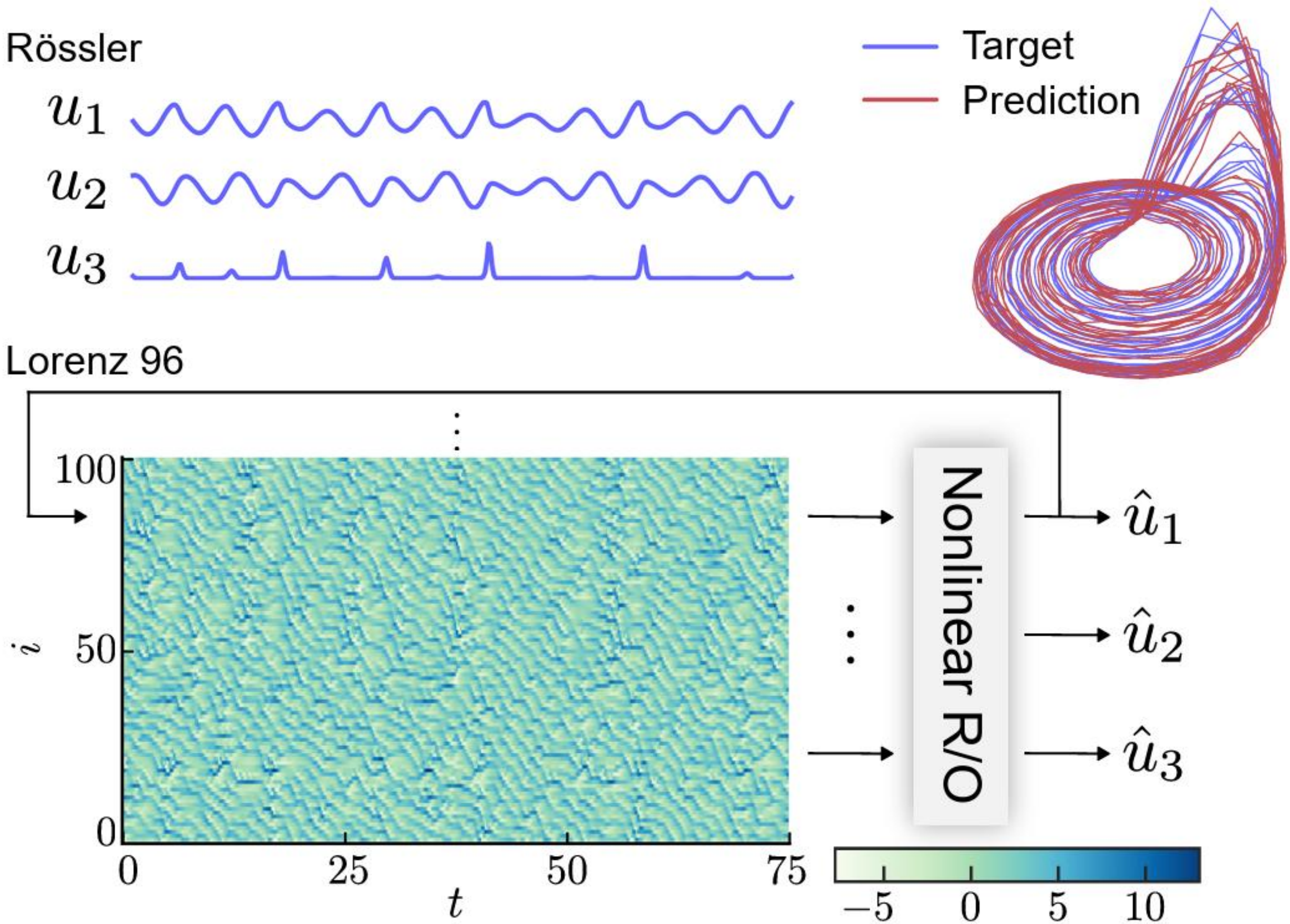


Kubota, T., Takahashi, H., & Nakajima, K. (2021). Unifying framework for information processing in stochastically driven dynamical systems. *Physical Review Research*, *3*(4), 043135.

- Time-invariant transformation using DNN!

Echo State Networks with Time-Variant States Solving NARMA10 Task

Emulating Rossler attractor with Lorenz 96

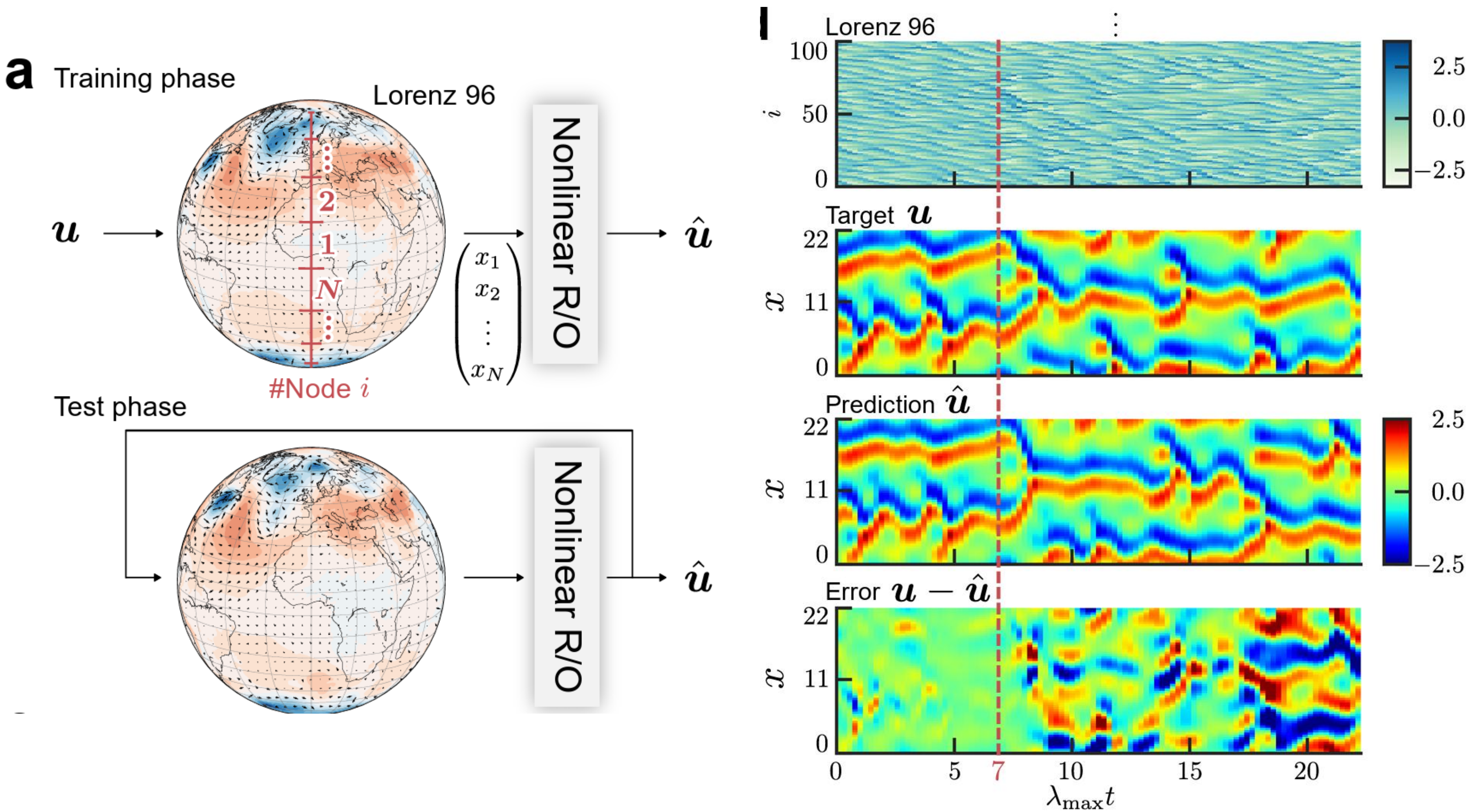


$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + \mu + \iota u_i \ (i = 1, \ldots, N)$$

1^{st} Attractor Embedding Task: Rössler Model into Lorenz 96

2^{nd} Attractor Embedding Task: Lissajous Curves into Lorenz 96

Emulating KS model using Lorenz 96



- Emulating spatiotemporal chaos using spatiotemporal chaos!

3rd Attractor Embedding Task: Kuramoto–Sivashinsky Model into Lorenz 96

note

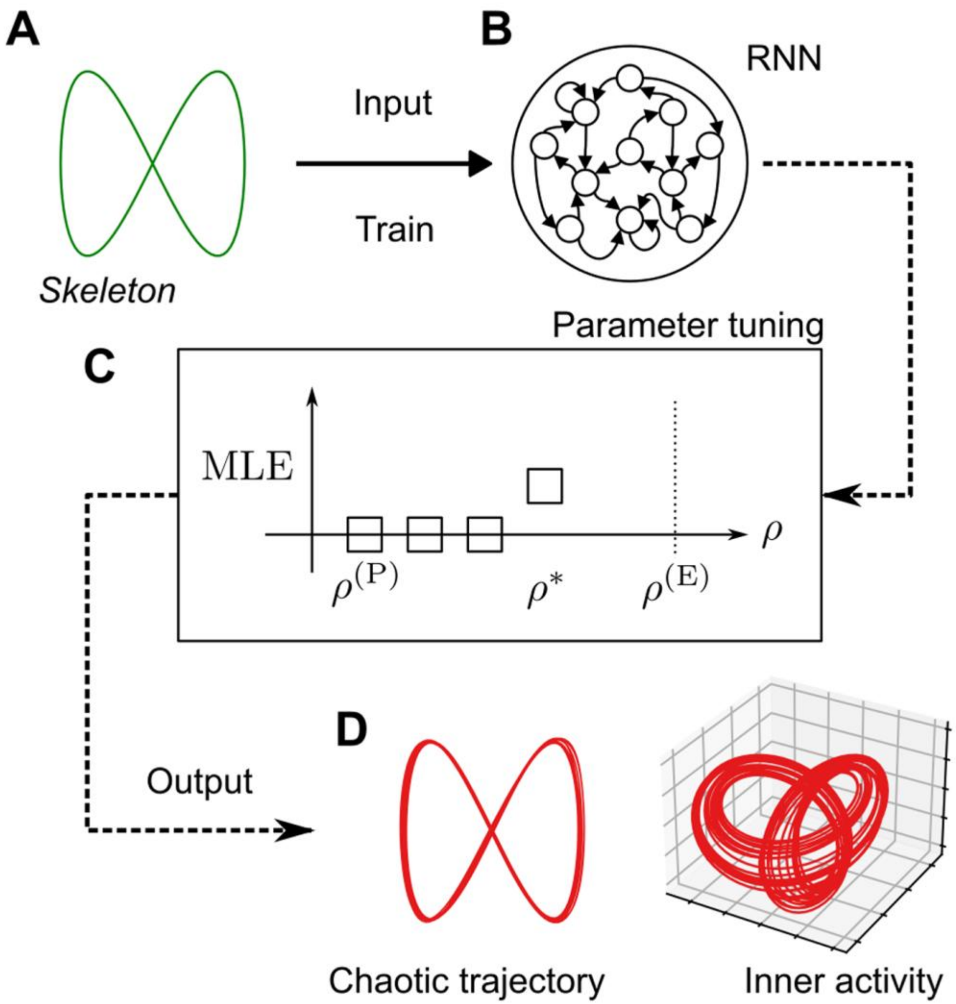
- Experimenters of PRC are already introducing the time-invariant transformation unconsciously in the name of post-processing (**detrending**, **Hilbert transformation**, **noise averaging**, etc.)

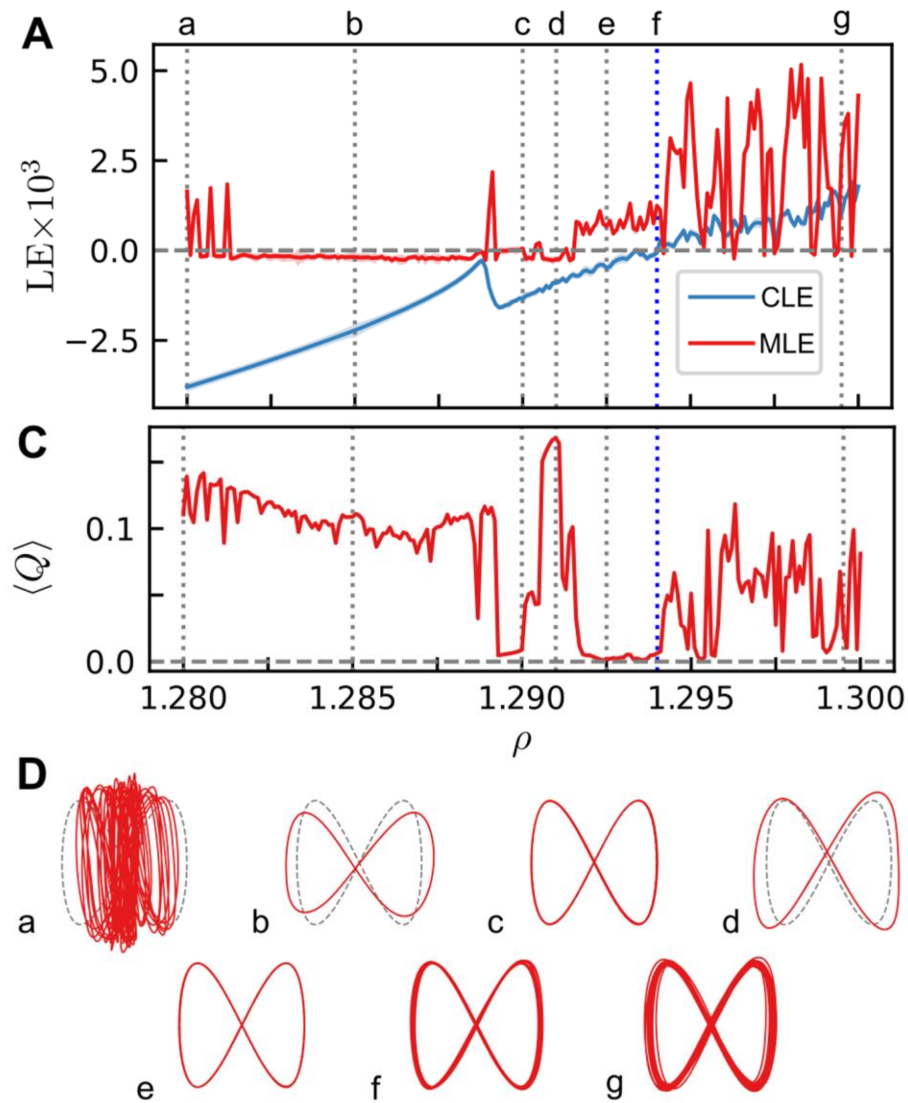
- Time-invariant transformation can be considered as a generalization of these post-processing procedures.

Designing the shape of chaotic attractor

Kabayama, T., Kuniyoshi, Y., Aihara, K., & Nakajima, K. (2025). Designing chaotic attractors: A semisupervised approach. Physical Review E, 111(3), 034207.

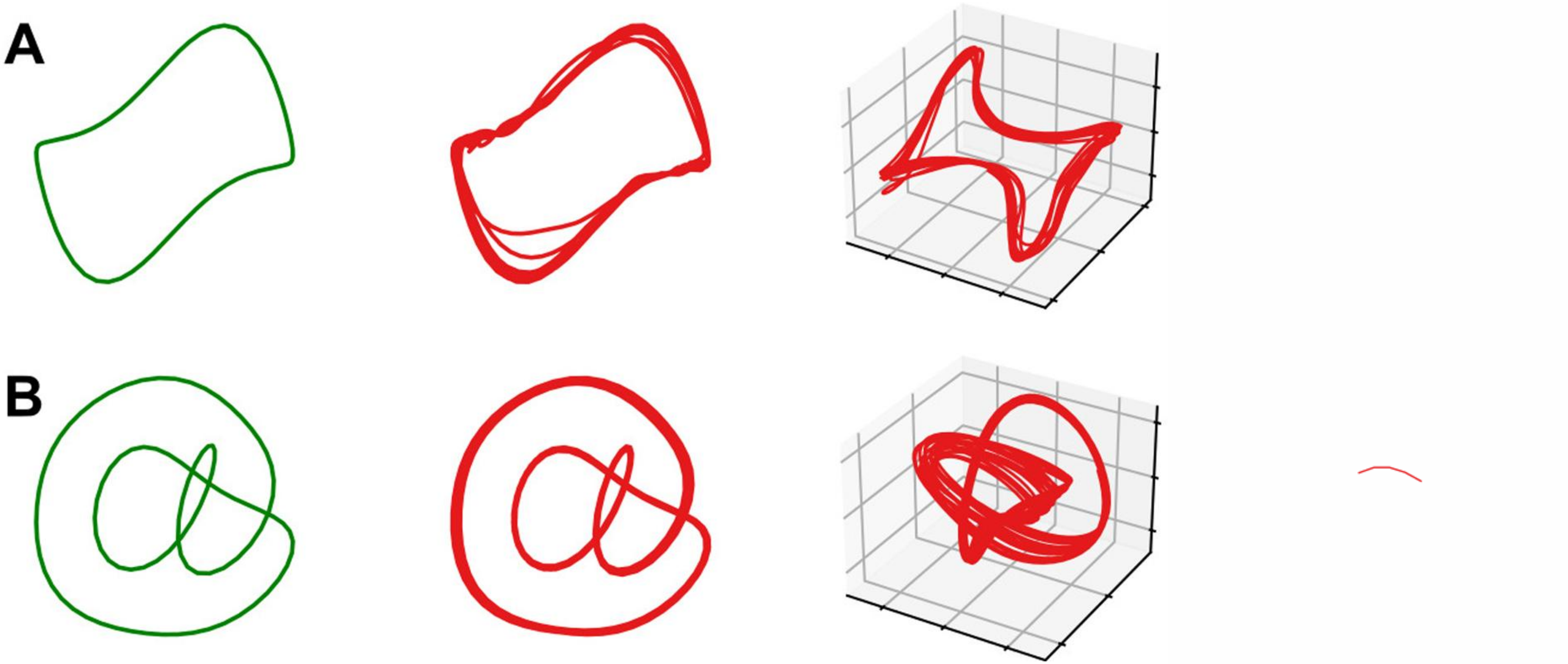
Scheme: using break down of ESP





- “Skeleton” provides template shape and the break down of ESP brings chaoticity!

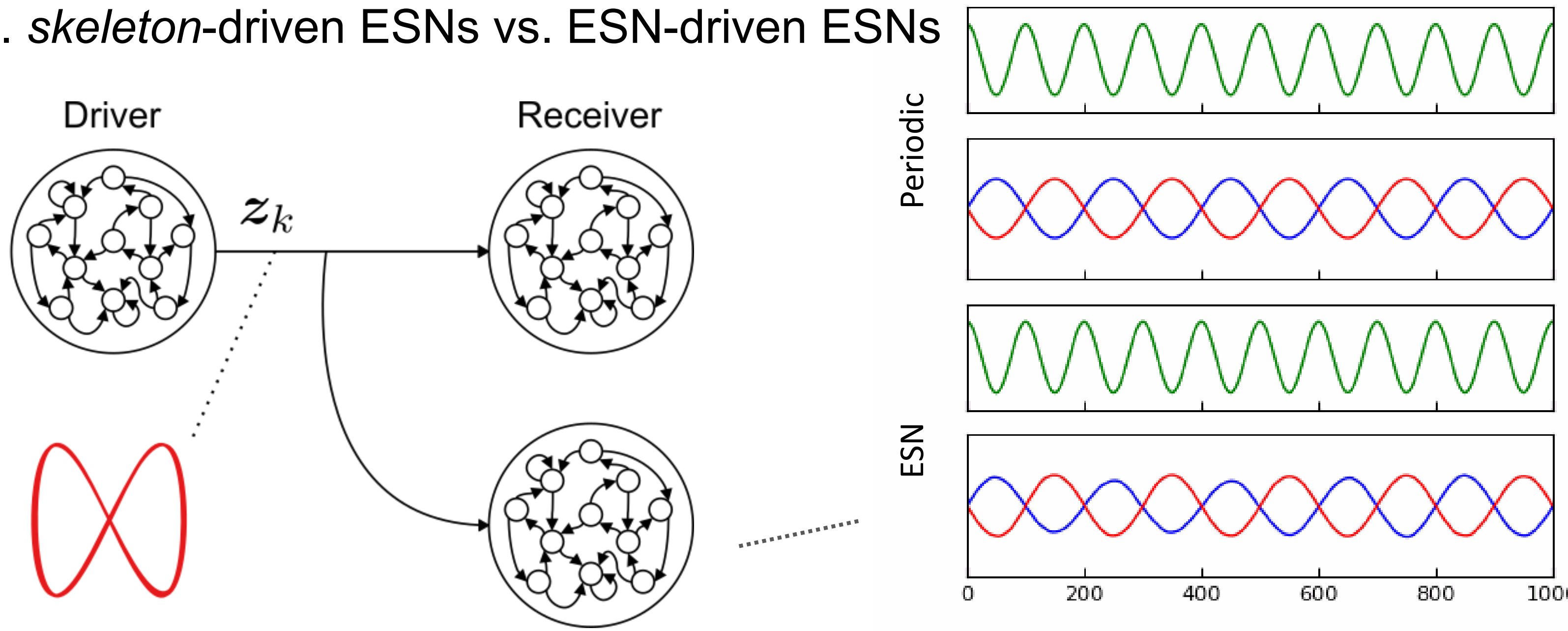
Shape of chaotic attractors can be easily designed!



- Especially useful when the equation is not there!
- Even with random realization of ESN, success ratio ~ 90% (with Lissajous-curve skeleton)

Cf. Application scenario in control

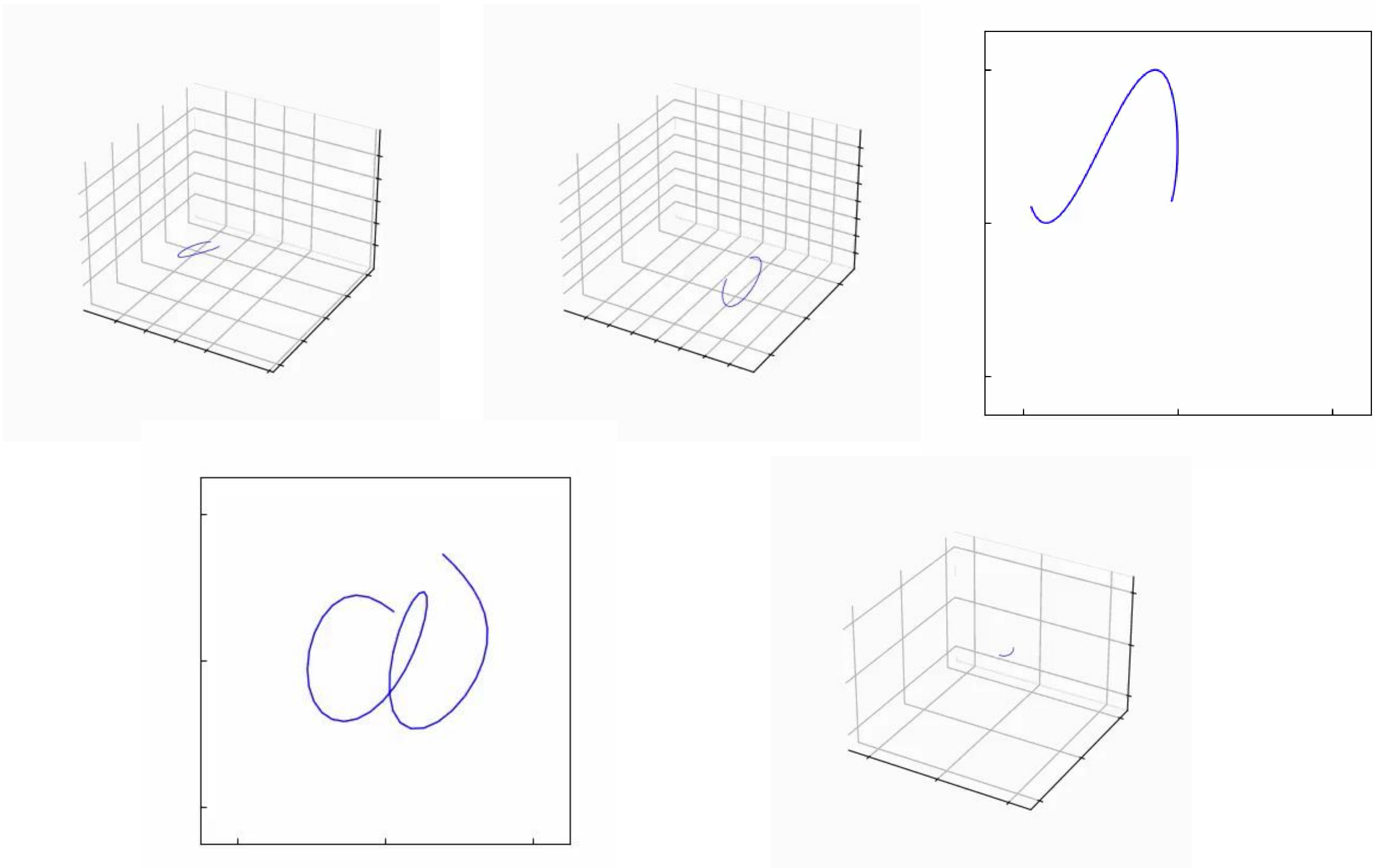
- **Chaotic robot motor command** to avoid stuck[2]
- Pseudo-periodic signal for **synchronization**[3]:
  - ◆Periodic-driven system with multiple-period response
  - ◆**Replacing driver with weak-chaotic one**
  - ◆ex. *skeleton*-driven ESNs vs. ESN-driven ESNs



[2] S. Steingrube, M. Timme, F. Wrögötter, and P. Manoonpong, Self-organized adaptation of a simple neural circuit enables complex robot behaviour, Nature physics 6, 224 (2010).

[3] T. L. Carroll and L. M. Pecora, Using chaos to keep period-multiplied systems in phase, Physical Review E 48, 2426 (1993).

Crisis induced intermittency in RC

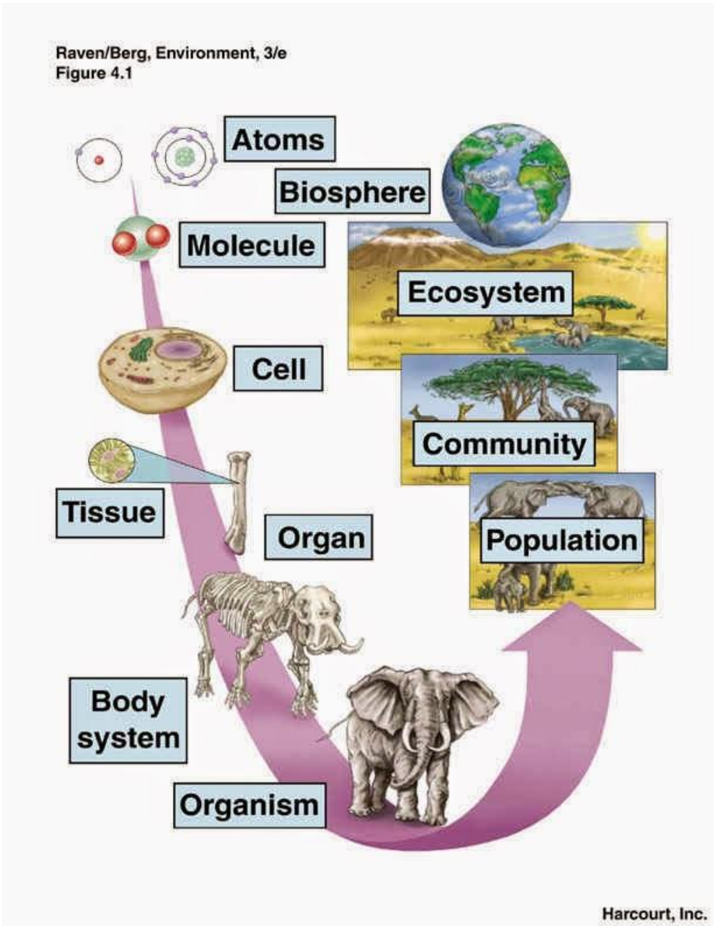


Kabayama, T., Komuro, M., Kuniyoshi, Y., Aihara, K., & Nakajima, K. (2025). Crisis-induced intermittency in reservoir computing. *Physical Review Research*, *7*(3), L032058.

Exploiting Chaotic Dynamics as Deep Neural Networks.

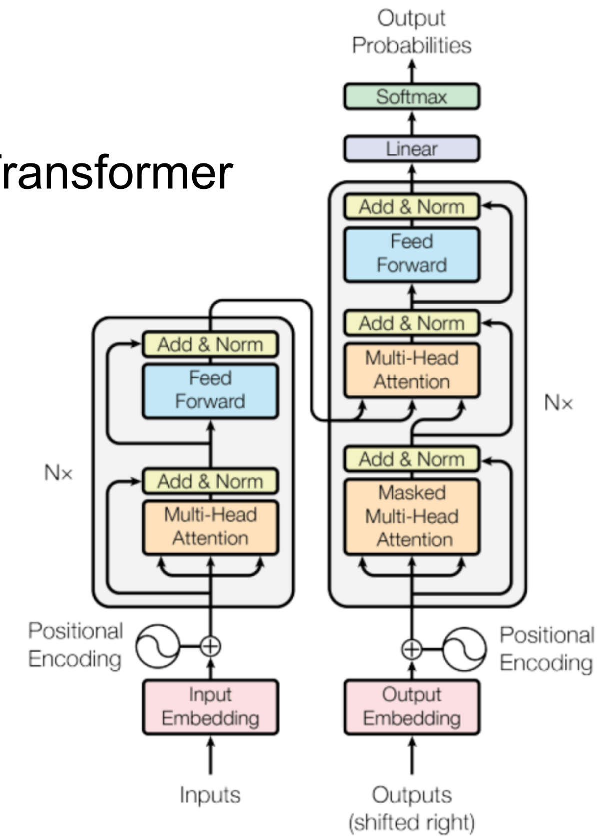
Liu, S., Akashi, N., Huang, Q., Kuniyoshi, Y., & Nakajima, K. (2025). Exploiting chaotic dynamics as deep neural networks. *Physical Review Research*, *7*(3), 033031.

Natural systems vs Large Language Models



- Evolved systems according to their ecological niche

Transformer



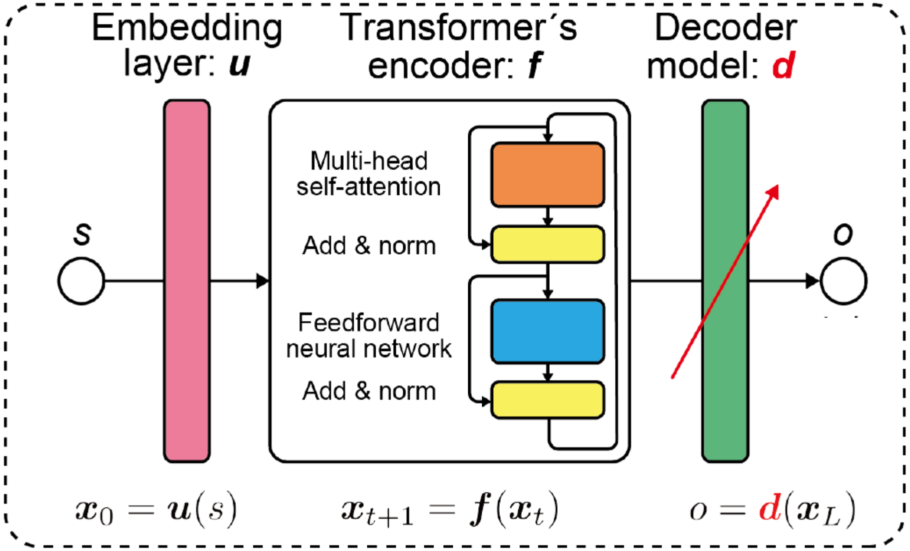
[1] Vaswani, A., et al., (2017). Attention is all you need. Advances in neural information processing systems, 30.
Image from https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

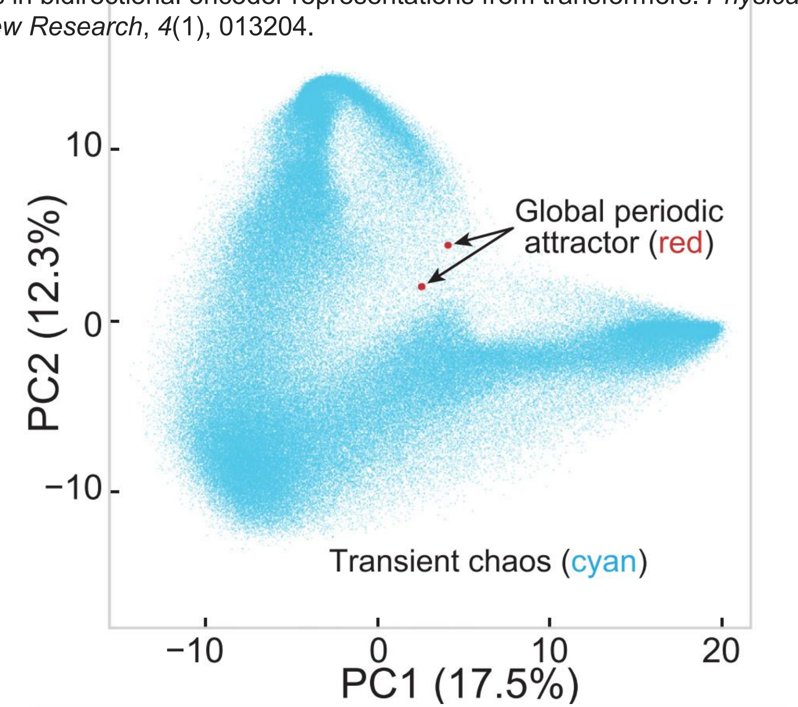
- Trained through huge amount of sentence data (e.g. BERT: masked language modeling task)

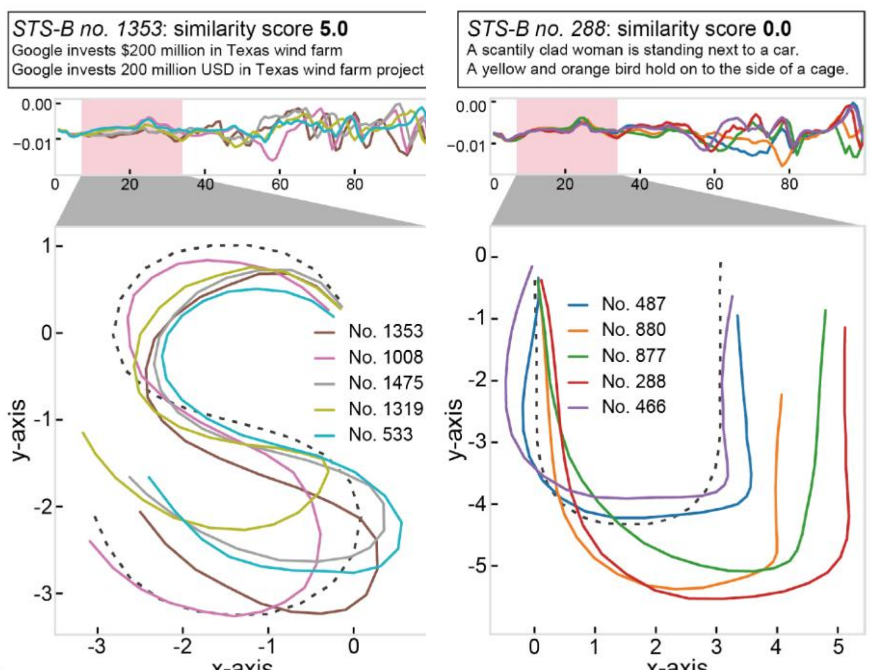
Large language models as a reservoir

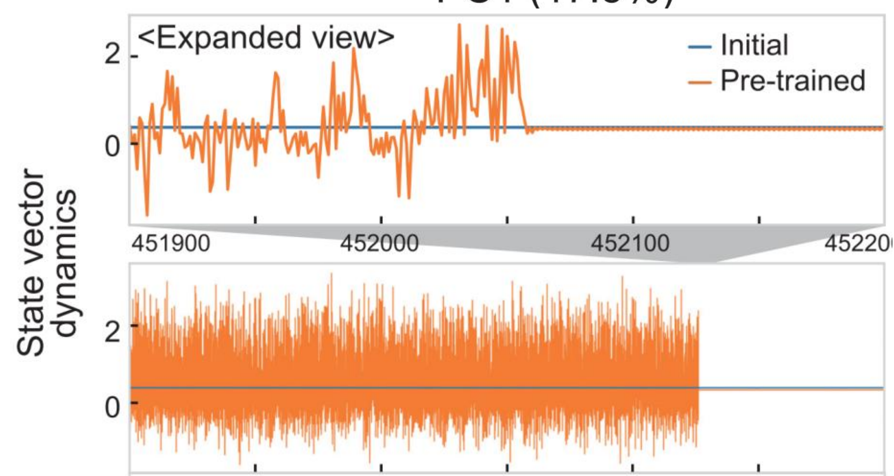
Butterfly Behind LLM

Inoue, K., Ohara, S., Kuniyoshi, Y., & Nakajima, K. (2022). Transient chaos in bidirectional encoder representations from transformers. *Physical Review Research*, *4*(1), 013204.









- Transient chaos exists in pretrained LLM!

5/15 東大よりプレスリリース

東京大学
THE UNIVERSITY OF TOKYO

【記者発表】LLMの情報処理は感覚性失語症の脳活動と似ていた
―LLMと失語症との情報処理ダイナミクスの比較―

2025年5月15日

https://ircn.jp/pressrelease/20250515_takamitsu_watanabe

論文情報

雑誌名：Advanced Science
題　名：Comparison of large language model with aphasia
著者名：Takamitsu Watanabe, Katsuma Inoue, Yasuo Kuniyoshi, Kohei Nakajima, Kazuyuki Aihara
DOI：10.1002/advs.202414016
URL： https://doi.org/10.1002/advs.202414016

UTokyo
THE UNIVERSITY OF TOKYO 150 SINCE 1877

HOME　UTokyo **FOCUS**

Features | Articles | Events | Press releases | Jobs | Find stories

PRESS RELEASES

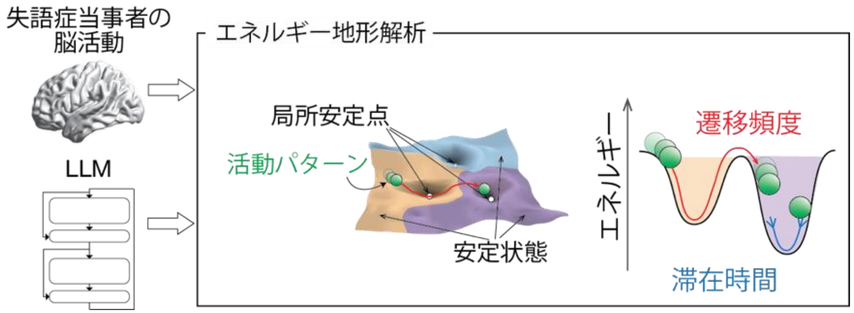
Enter terms　Search

AI overconfidence mirrors human brain condition

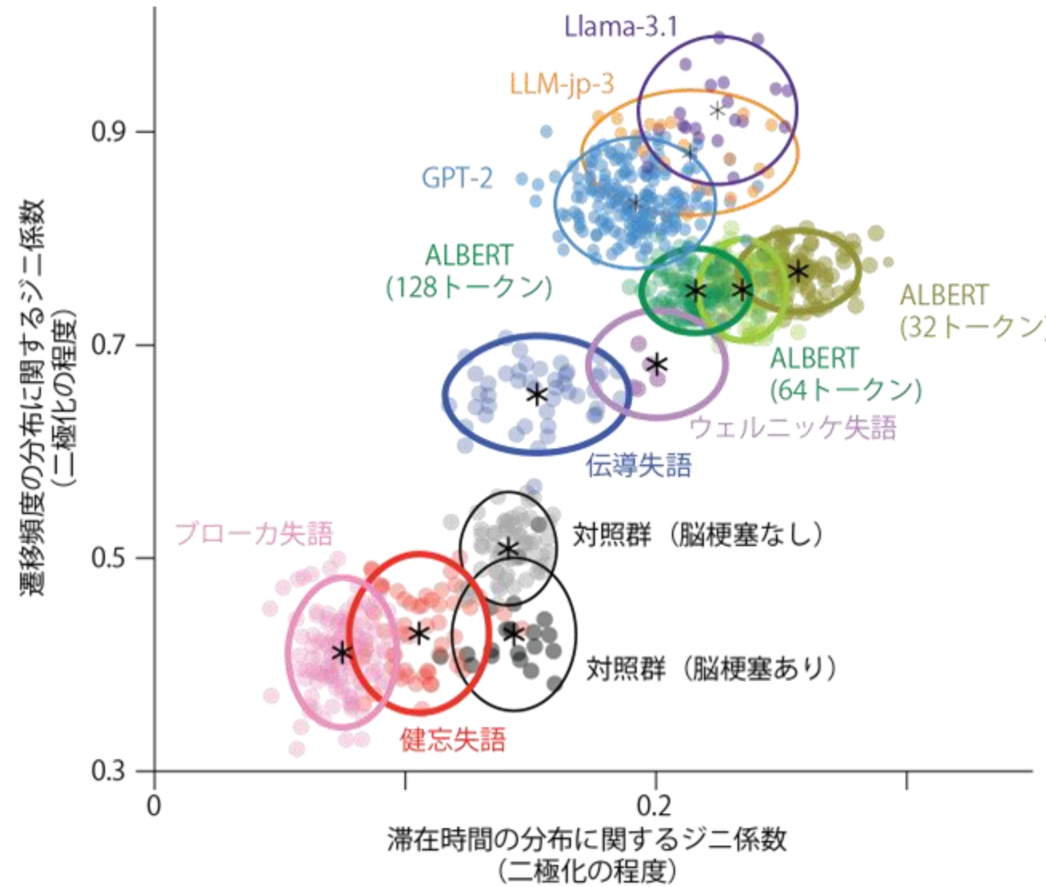
A similarity between language models and aphasia points to diagnoses for both

Research news

LLMの情報処理は感覚性失語症の脳活動と似ている！？

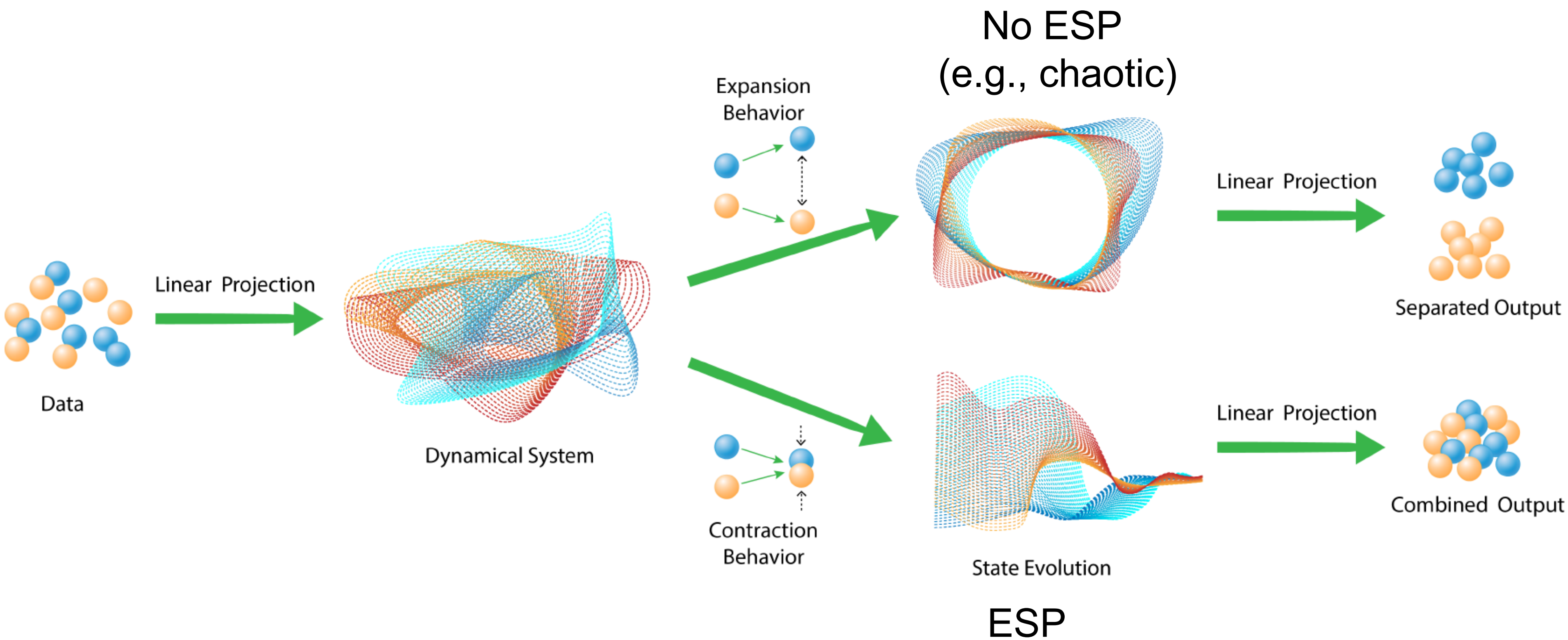


- エネルギー地形解析を失語症患者の脳活動とLLMのダイナミクスにおいて解析！
- LLMの内部状態はウェルニッケ失語症に代表される感覚性失語症に似ている可能性がある



T. Watanabe, K. Inoue, Y. Kuniyoshi, K. Nakajima, K. Aihara, Comparison of large language model with aphasia, ***Advanced Science***, 2414016, 2025.

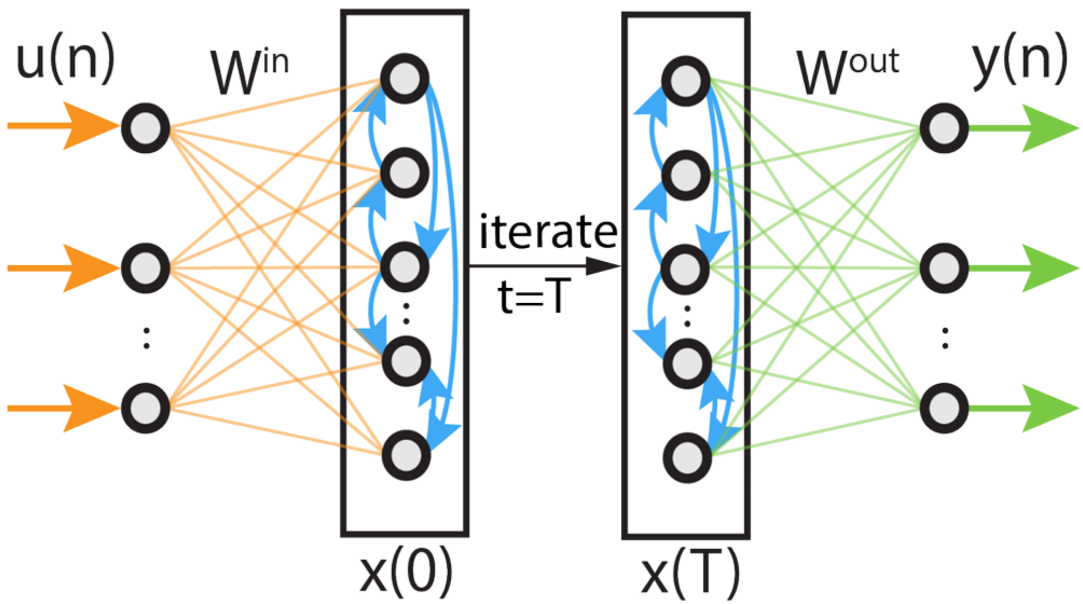
Dynamical system as feed-forward neural networks



Expansion property (no ESP) is rather useful for separation (classification)?

Feed-forward ESN

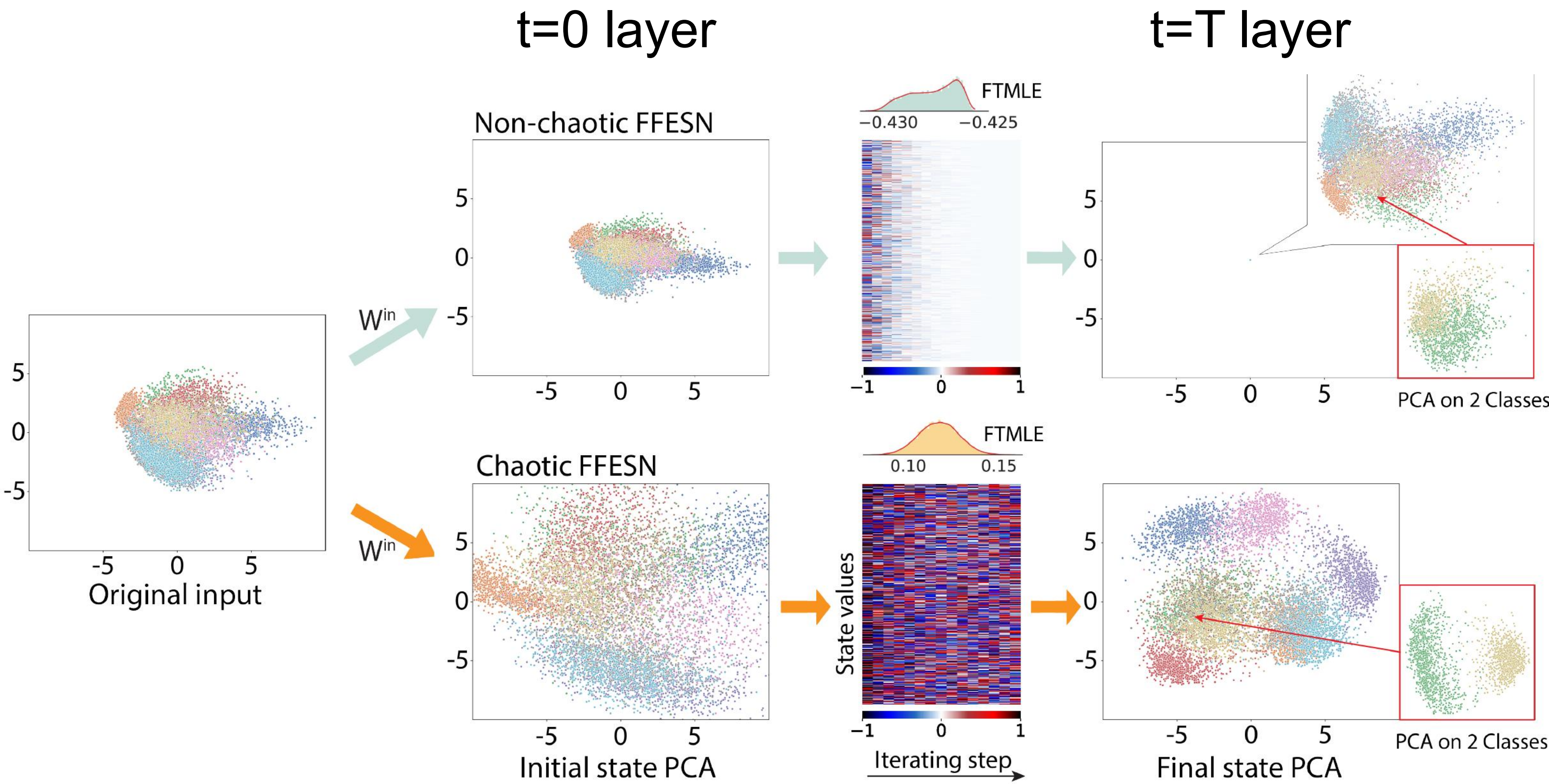
$$x(t+1) = \mathcal{F}(Wx(t))$$



$$x(0) = W_{\text{in}}u \ ,$$
$$x(i+1) = f(W'x(i)) \ ,$$
$$y = W_{\text{out}}x(n) \ .$$

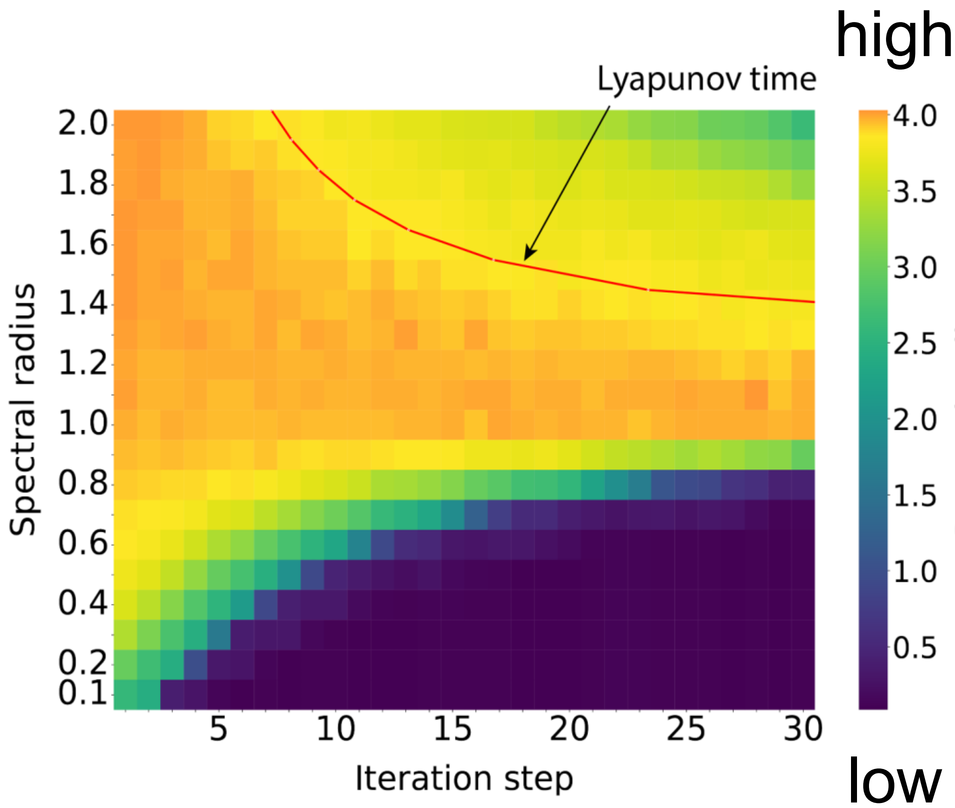
- Iteration timesteps as layers of deep neural networks (DNN)!
- W_{in}, W_{out} are trained through BP
- W_{in} determines how to set the initial conditions

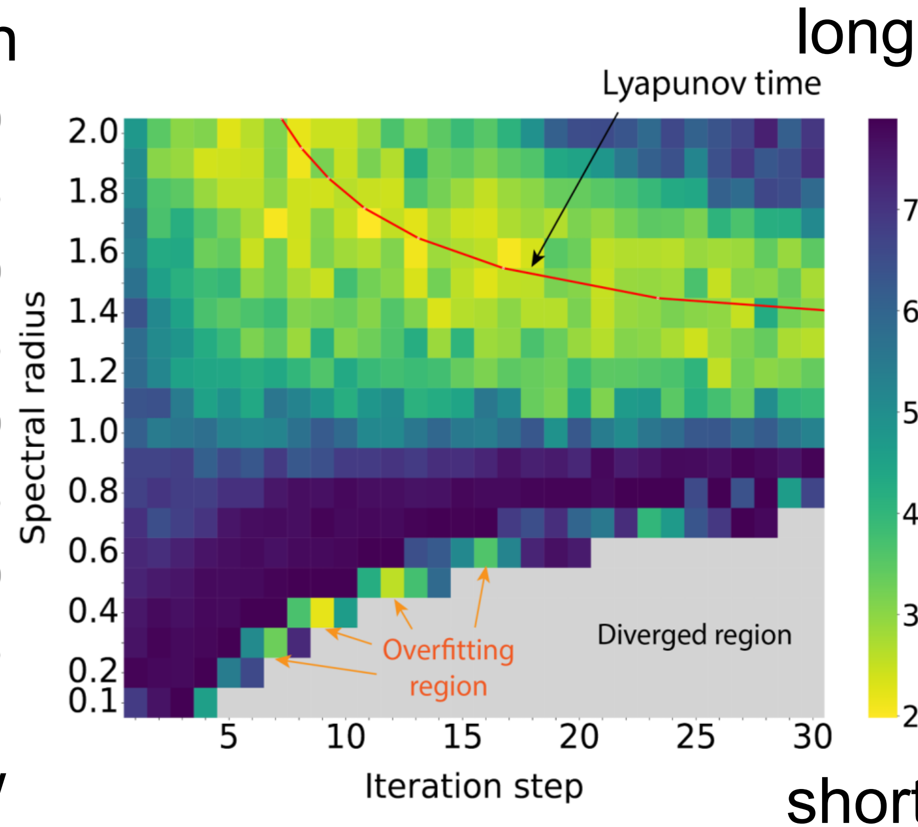
Implementing MNIST classification task



- Memory of initial conditions are rather required!
 (= broken down of ESP)
- Chaos generates high separability!

Performance and convergence time





- Lyapunov time characterizes the optimal timesteps (layers) of the network in chaotic region

High performance similar to DNN

System	High.(%)	Avg.(%)
Linear Regression	92.52	92.48
MLP ($N = 500$)	97.53	97.45
FFESN ($\rho = 0.9$, $N = 500$, $T = 1$)	98.16	98.08
FFESN ($\rho = 1.0$, $N = 500$, $T = 1$)	98.26	98.12
FFESN ($\rho = 1.8$, $N = 500$, $T = 2$)	98.34	98.23
Lorenz 96 ($F = 0.5$, $N = 500$, $T = 0.4$)	98.29	98.17
Lorenz 96 ($F = 4.75$, $N = 500$, $T = 0.2$)	98.20	98.02
CNN ($N = 600$)	98.77	98.69
CSTOs ($N = 600$, $A_{\mathrm{cp}} = 17.8$ Oe, $T = 300$ ps)	98.43	98.33
Deep CSTOs ($A_{\mathrm{cp}} = \{10, 0.1, 10\}$, $T = 100$ ps)	98.56	98.33
CNN + CSTOs ($N = 600$, cp $= 745$ Oe, $T = 200$ ps)	**99.05**	**99.00**

MLP and CNN all have only one hidden layer with N neurons.

- Chaotic dynamics has a potential to be used!

To be continued on 11/7