mpipks

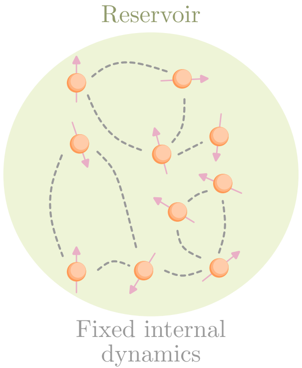
A (very) short introduction to Quantum Reservoir Computing

Gabriel O. Alves

June 24, 2026

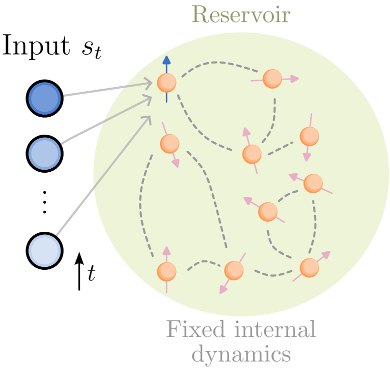
Starting point: Can we harness the *inherent* complexity and internal dynamics of quantum systems for learning?

Quantum Reservoir Computing



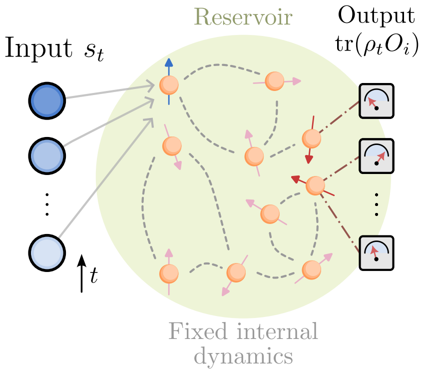
Quantum Reservoir Computing

Input $s_t \rightarrow$ Reservoir



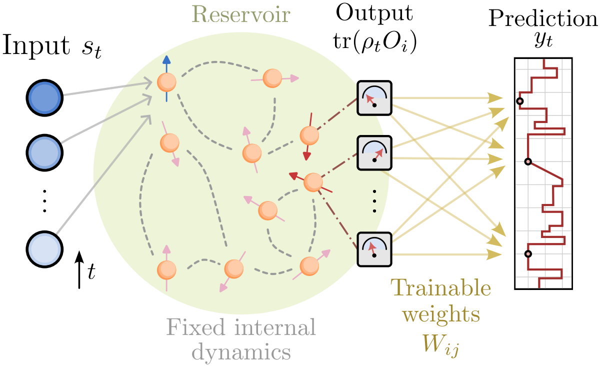
Quantum Reservoir Computing

Input $s_t \rightarrow$ Reservoir $\rightarrow$ Measure observables O_i



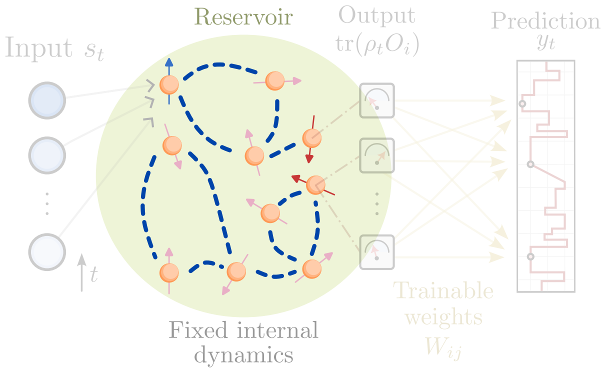
Quantum Reservoir Computing

Input $s_t \to$ Reservoir $\to$ Measure observables $O_i \to$ Prediction y_t



Quantum Reservoir Computing

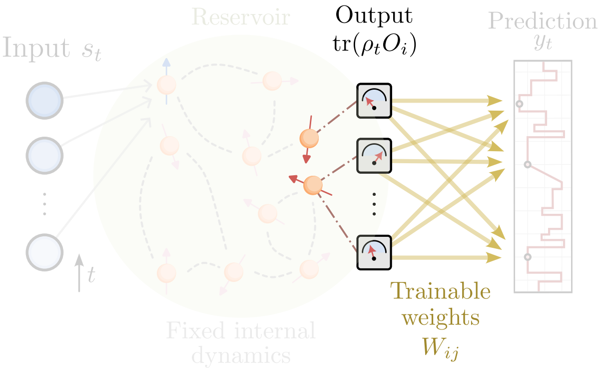
Input $s_t \to$ Reservoir $\to$ Measure observables O_i



Acts as a fixed/untrainable hidden layers

Quantum Reservoir Computing

Input $s_t \to$ Reservoir $\to$ Measure observables O_i

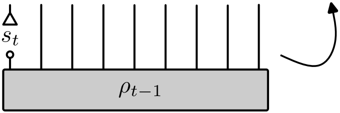


Single trainable/visible layer: simple regression

Quantum Reservoir Computing - 1. Encoding

Input encoding via reset-and-inject protocol by Fujii and Nakajima [1]

$$\mathcal{E}(\rho_{t-1}) := |s_t\rangle \langle s_t| \otimes \mathrm{PTr}(\rho_{t-1})$$

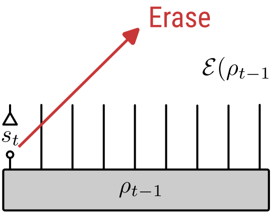


[1] [Physical Review Applied 8, 024030]

Quantum Reservoir Computing - 1. Encoding

Input encoding via reset-and-inject protocol by Fujii and Nakajima [1]

1. Reset the first qubit



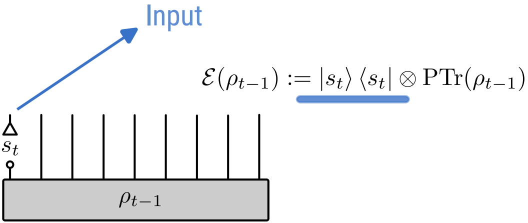
$$\mathcal{E}(\rho_{t-1}) := |s_t\rangle \langle s_t| \otimes \mathrm{PTr}(\rho_{t-1})$$

[1] [Physical Review Applied 8, 024030]

Quantum Reservoir Computing - 1. Encoding

Input encoding via reset-and-inject protocol by Fujii and Nakajima [1]

1. Reset the first qubit
2. Encode the external input $|s_t\rangle = \sqrt{1 - s_t}\,|0\rangle + \sqrt{s_t}\,|1\rangle$

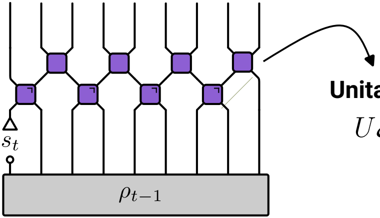


[1] [Physical Review Applied 8, 024030]

Quantum Reservoir Computing - 1. Encoding

Input encoding via reset-and-inject protocol by Fujii and Nakajima [1]

Concrete reservoir example: **brickwork quantum circuit**



Unitary evolution:

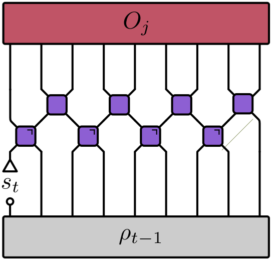
$U\mathcal{E}(\rho_{t-1})U^\dagger$

[1][Physical Review Applied 8, 024030]

Quantum Reservoir Computing - 1. Encoding

Input encoding via reset-and-inject protocol by Fujii and Nakajima [1]

Concrete reservoir example: **brickwork quantum circuit**

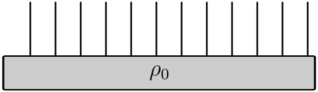


Features: expected values

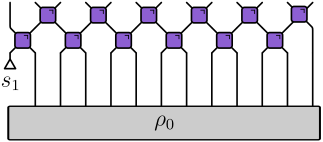
$$\mathrm{Tr}(\rho O_j) =: x_{t,j}$$

[1][Physical Review Applied 8, 024030]

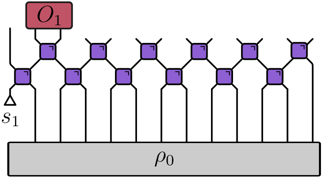
Constructing the feature matrix



Constructing the feature matrix



Constructing the feature matrix

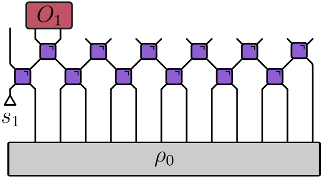


Constructing the feature matrix

$$\mathbf{x} = \left(\quad \right)$$

Compute expected value at time t

$$x_{t,j} = \langle O_j \rangle_t = \mathrm{Tr}(\rho_t O_j)$$

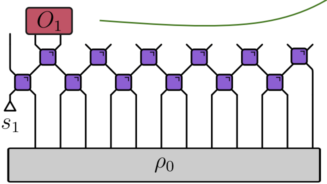


Constructing the feature matrix

$$\mathbf{x} = \begin{pmatrix} x_{11} & & \\ & & \\ & & \end{pmatrix}$$

Compute expected value at time t

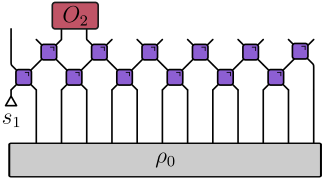
$$x_{t,j} = \langle O_j \rangle_t = \mathrm{Tr}(\rho_t O_j)$$



Constructing the feature matrix

$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & & \\ & & & \\ & & & \end{pmatrix}$$

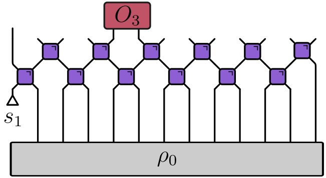
We compute **another** observable



Constructing the feature matrix

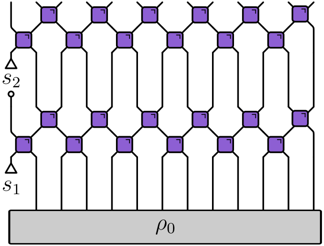
$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots \\ & & & \\ & & & \end{pmatrix}$$

and so on...



Constructing the feature matrix

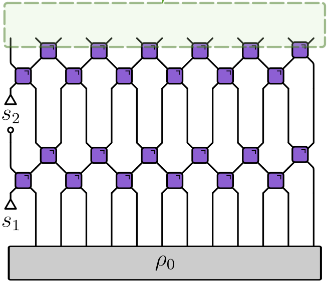
Insert new input and circuit layer



$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots \\ & & & \\ & & & \end{pmatrix}$$

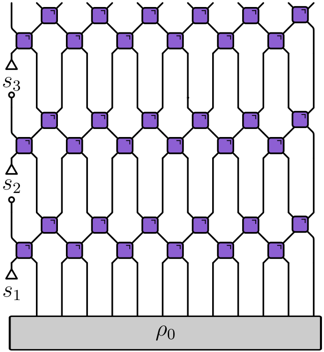
Constructing the feature matrix

Compute expectations at t = 2



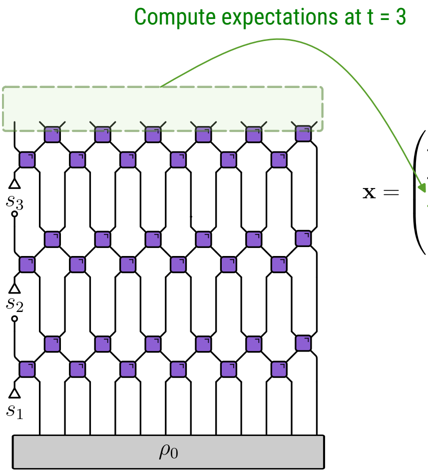
$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots \\ x_{21} & x_{22} & x_{23} & \cdots \\ & & & \end{pmatrix}$$

Constructing the feature matrix



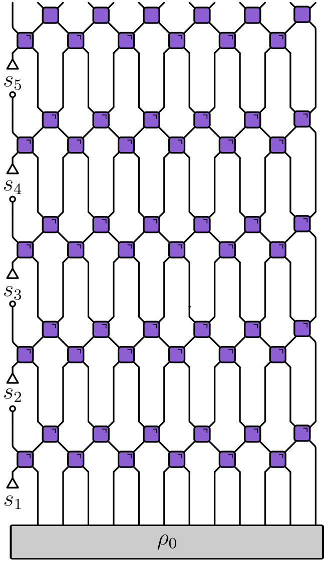
$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots \\ x_{21} & x_{22} & x_{23} & \cdots \\ x_{31} & x_{32} & x_{33} & \cdots \\ & & & \end{pmatrix}$$

Constructing the feature matrix



$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots \\ x_{21} & x_{22} & x_{23} & \cdots \\ x_{31} & x_{32} & x_{33} & \cdots \\ & & & \end{pmatrix}$$

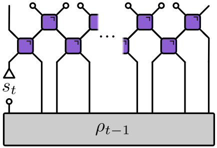
Constructing the feature matrix

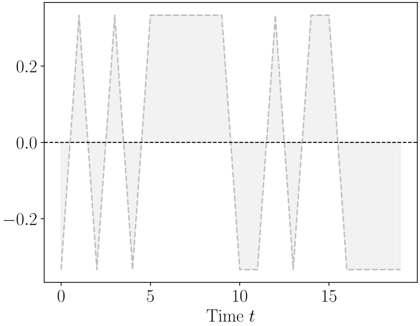


$$\mathbf{x} = \begin{pmatrix} x_{11} & x_{12} & x_{13} & \cdots \\ x_{21} & x_{22} & x_{23} & \cdots \\ x_{31} & x_{32} & x_{33} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

1. Erase-input encoding - concrete example

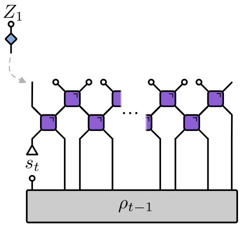
Consider a numerical example for the dynamics and a few observables:

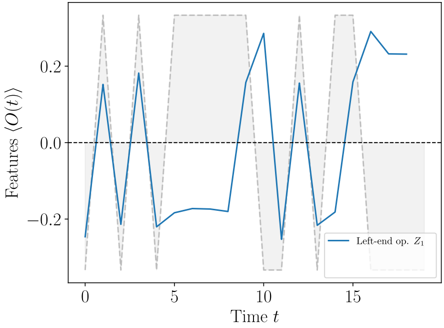




1. Erase-input encoding - concrete example

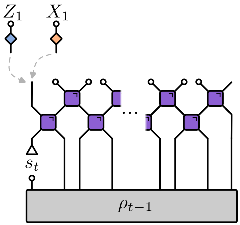
Consider a numerical example for the dynamics and a few observables:

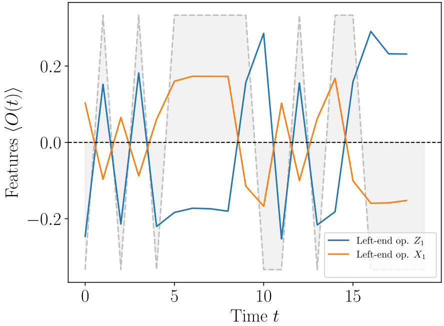




1. Erase-input encoding - concrete example

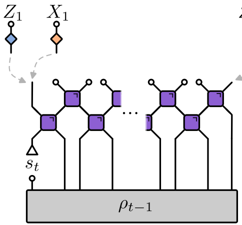
Consider a numerical example for the dynamics and a few observables:

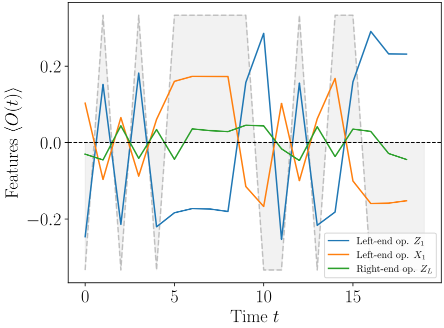




1. Erase-input encoding - concrete example

Consider a numerical example for the dynamics and a few observables:





2. Benchmark tasks

Reservoirs need to capture **memory and nonlinearity**

☞ Short-term memory with time-lag τ:

$$y_t = s_{t-\tau}, \quad s \in [0, 1]$$

2. Benchmark tasks

Reservoirs need to capture **memory and nonlinearity**

- ☞ Short-term memory with time-lag τ:

$$y_t = s_{t-\tau}, \quad s \in [0, 1]$$

- ☞ Parity check (highly nonlinear)

$$y_t = \sum_{k=1}^{\tau} s_{t-k} \mod 2, \quad s \in \{0, 1\}$$

2. Benchmark tasks

Reservoirs need to capture **memory and nonlinearity**

- ☞ Short-term memory with time-lag τ:

$$y_t = s_{t-\tau}, \quad s \in [0,1]$$

- ☞ Parity check (highly nonlinear)

$$y_t = \sum_{k=1}^{\tau} s_{t-k} \mod 2, \quad s \in \{0,1\}$$

- ☞ NARMA(τ) task: combines memory and nonlinearity

2. Benchmark tasks

Reservoirs need to capture **memory and nonlinearity**

- ☞ Short-term memory with time-lag τ:

$$y_t = s_{t-\tau}, \quad s \in [0, 1]$$

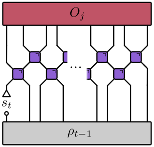
- ☞ Parity check (highly nonlinear)

$$y_t = \sum_{k=1}^{\tau} s_{t-k} \mod 2, \quad s \in \{0, 1\}$$

- ☞ NARMA(τ) task: combines memory and nonlinearity

By tuning τ we will probe the memory of the reservoir.

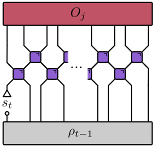
3. Training

Given the features $x_{t,j} = \mathrm{tr}\{\rho_t O_j\} =$ 

:

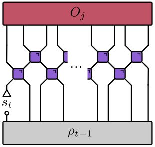
☞ Ridge regression: $\mathbf{W} = (\mathbf{x}^T\mathbf{x} + \alpha\mathbf{I})^{-1}\mathbf{x}^T\mathbf{y}$

3. Training

Given the features $x_{t,j} = \mathrm{tr}\{\rho_t O_j\} =$  :

- ☞ Ridge regression: $\mathbf{W} = (\mathbf{x}^T\mathbf{x} + \alpha\mathbf{I})^{-1}\mathbf{x}^T\mathbf{y}$
- ☞ Prediction: $\mathbf{y} = \mathbf{W}\mathbf{x}$

3. Training

Given the features $x_{t,j} = \mathrm{tr}\{\rho_t O_j\} =$  :

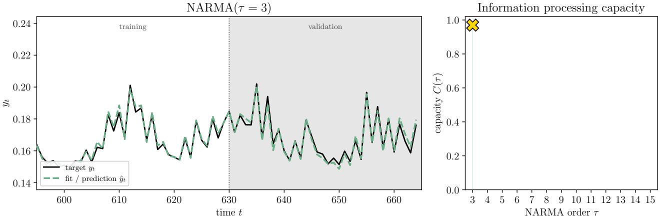
- Ridge regression: $\mathbf{W} = (\mathbf{x}^T\mathbf{x} + \alpha\mathbf{I})^{-1}\mathbf{x}^T\mathbf{y}$
- Prediction: $\mathbf{y} = \mathbf{W}\mathbf{x}$
- We evaluate performance through the information processing capacity (IPC): $C = \frac{\mathrm{cov}^2(\mathbf{y}_{\mathrm{val}}, \mathbf{y}_{\mathrm{tar}})}{\mathrm{var}(\mathbf{y}_{\mathrm{val}})\mathrm{var}(\mathbf{y}_{\mathrm{tar}})}$

Learning in practice: trying to learn a NARMA time-series with a small brickwork circuit as a reservoir.

Learning the NARMA(τ) task

Concrete numerics for $L = 7$ and 39 nodes

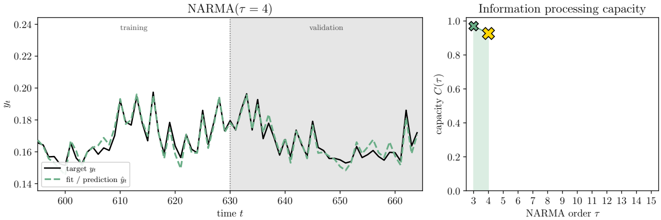
$$\text{NARMA}(\tau) \text{ task: } y_k = 0.3\, y_{k-1} + 0.05\, y_{k-1} \sum_{i=1}^{\tau} y_{k-i} + 1.5\, s_{k-\tau}\, s_{k-1} + 0.1$$



Learning the NARMA(τ) task

Concrete numerics for $L = 7$ and 39 nodes

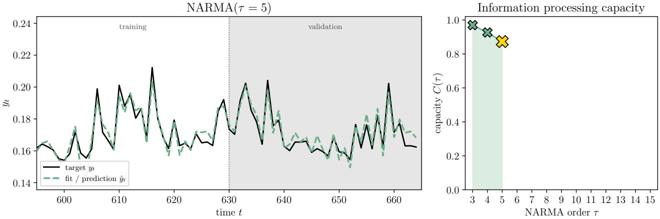
$$\text{NARMA}(\tau)\text{ task: } y_k = 0.3\, y_{k-1} + 0.05\, y_{k-1} \sum_{i=1}^{\tau} y_{k-i} + 1.5\, s_{k-\tau}\, s_{k-1} + 0.1$$



Learning the NARMA(τ) task

Concrete numerics for $L = 7$ and 39 nodes

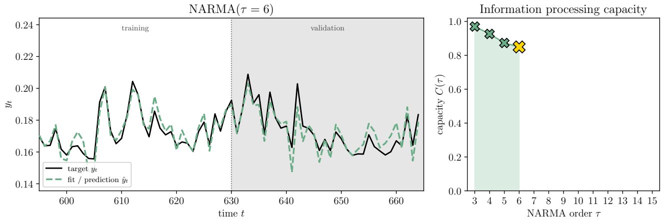
$$\text{NARMA}(\tau) \text{ task: } y_k = 0.3\, y_{k-1} + 0.05\, y_{k-1} \sum_{i=1}^{\tau} y_{k-i} + 1.5\, s_{k-\tau}\, s_{k-1} + 0.1$$



Learning the NARMA(τ) task

Concrete numerics for $L = 7$ and 39 nodes

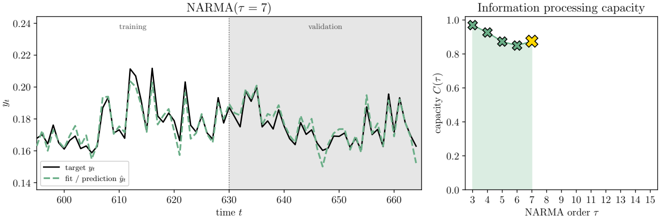
$$\text{NARMA}(\tau) \text{ task: } y_k = 0.3\, y_{k-1} + 0.05\, y_{k-1} \sum_{i=1}^{\tau} y_{k-i} + 1.5\, s_{k-\tau}\, s_{k-1} + 0.1$$



Learning the NARMA(τ) task

Concrete numerics for $L = 7$ and 39 nodes

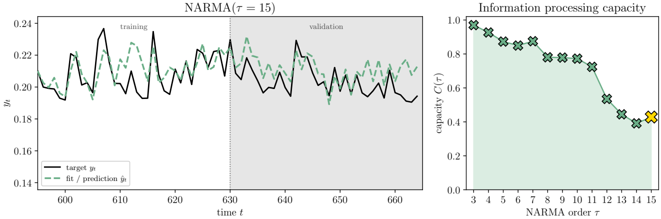
$$\text{NARMA}(\tau)\text{ task: } y_k = 0.3\, y_{k-1} + 0.05\, y_{k-1} \sum_{i=1}^{\tau} y_{k-i} + 1.5\, s_{k-\tau}\, s_{k-1} + 0.1$$



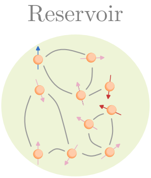
Learning the NARMA(τ) task

Concrete numerics for $L = 7$ and 39 nodes

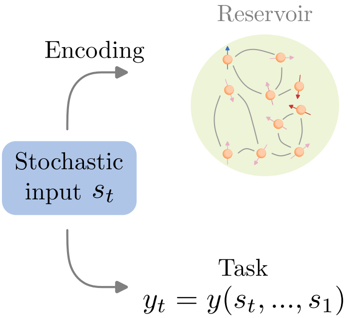
$$\text{NARMA}(\tau) \text{ task: } y_k = 0.3\, y_{k-1} + 0.05\, y_{k-1} \sum_{i=1}^{\tau} y_{k-i} + 1.5\, s_{k-\tau}\, s_{k-1} + 0.1$$



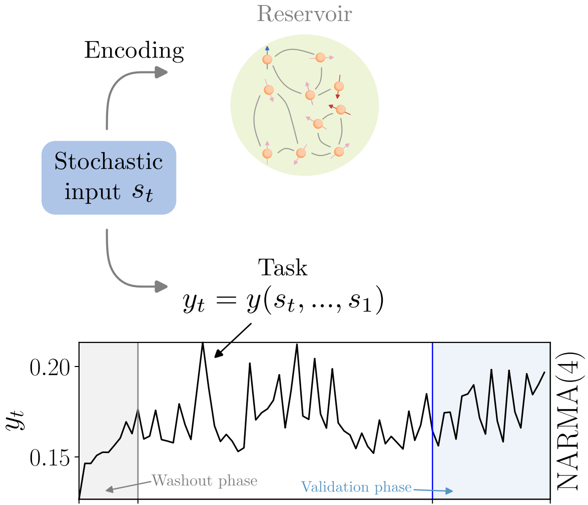
Quantum Reservoir Computing - Summary



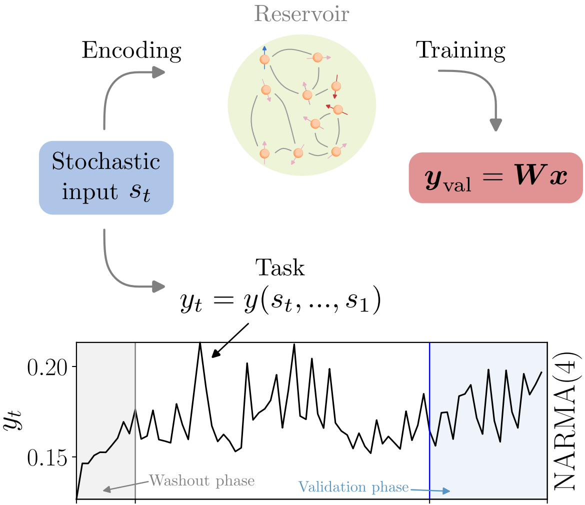
Quantum Reservoir Computing - Summary



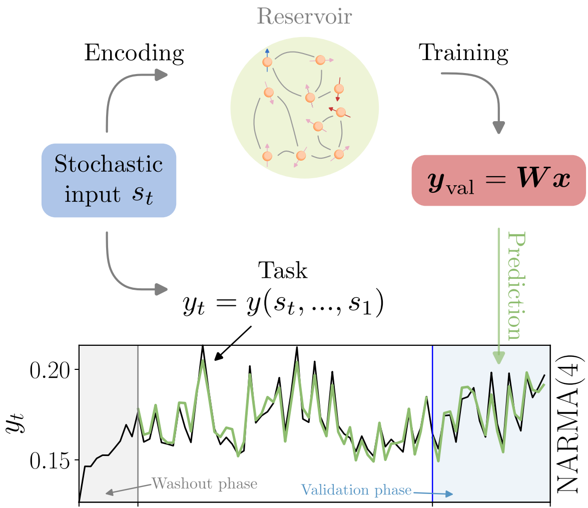
Quantum Reservoir Computing - Summary



Quantum Reservoir Computing - Summary

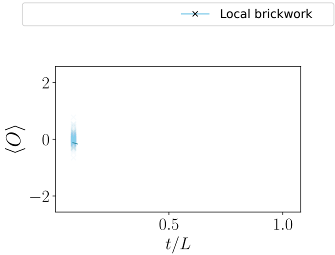


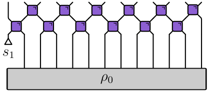
Quantum Reservoir Computing - Summary



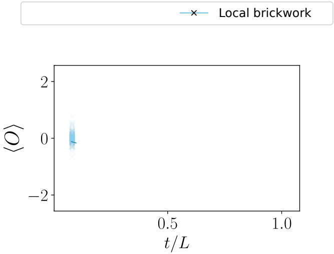
What are the limitations of QRC?

The problem of exponential concentration

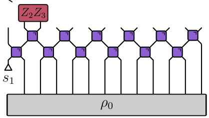




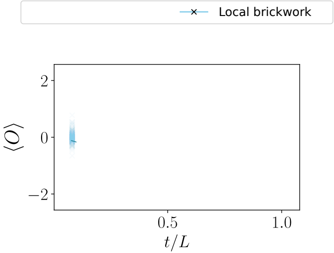
The problem of exponential concentration



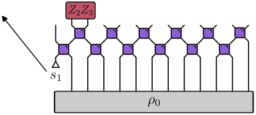
We pick some arbitrary local operator and plot its expectation values



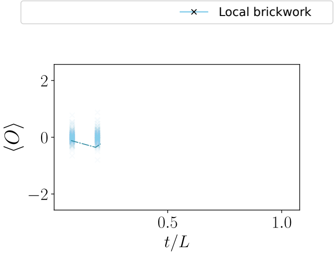
The problem of exponential concentration

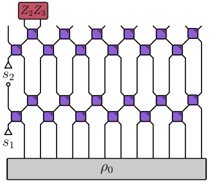


Repeat the process
for many stochastic
inputs $s_1 \in [0, 1]$

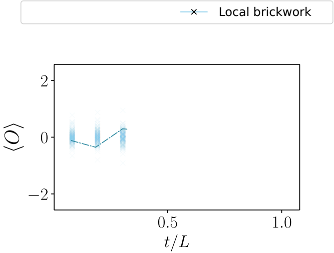


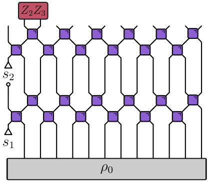
The problem of exponential concentration



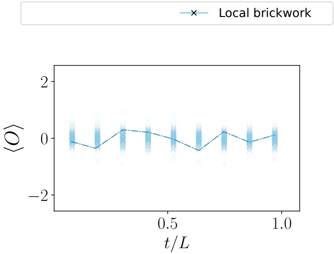


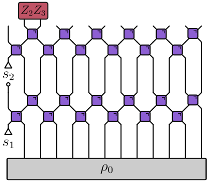
The problem of exponential concentration



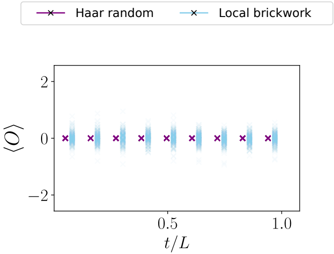


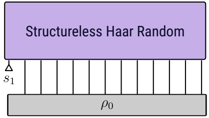
The problem of exponential concentration



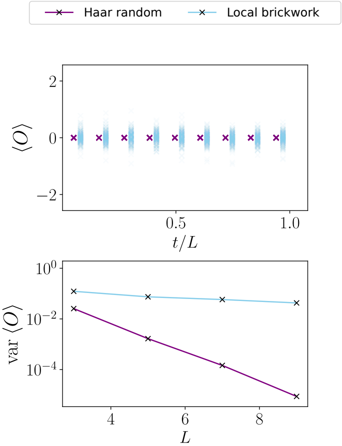


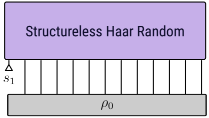
The problem of exponential concentration





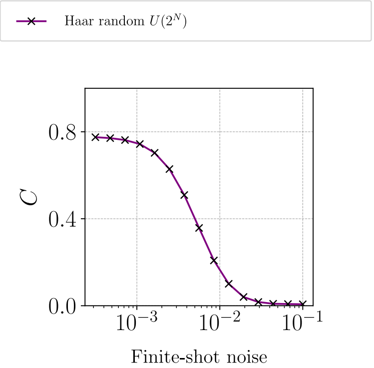
The problem of exponential concentration





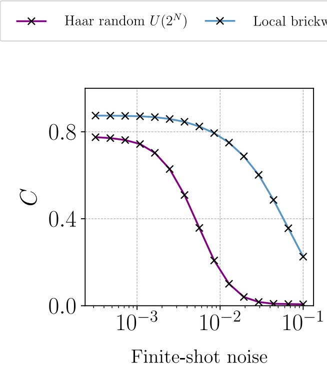
The problem of exponential concentration

Finite-shot noise degrades learning!



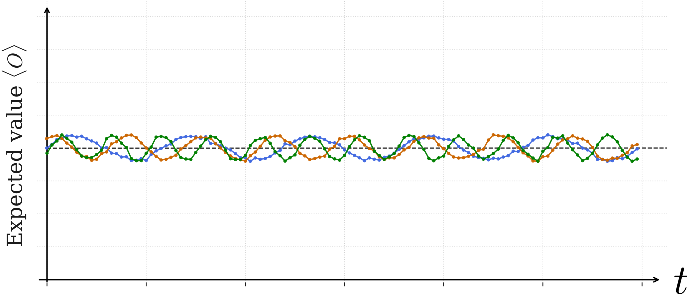
The problem of exponential concentration

Finite-shot noise degrades learning! Previous slide: Local structure can help [2]



[2] W. Xiong, Z. Holmes, A. Angrisani, Y. Suzuki, T. Chotibut, and S. Thanasilp, “Role of scrambling and noise in temporal information processing with quantum systems,” arXiv:2505.10080 (2025).

Interpretation



Interpretation

Conclusion

- ☞ We only discussed one reservoir architecture; many other physical platforms are available (e.g. photonics, Bose-Hubbard lattices, ...)

  G. Llodrà, P. Mujal, R. Zambrini, and G. L. Giorgi, “Quantum reservoir computing in atomic lattices,” arXiv:2411.13401 (2024).

- ☞ Rigorous approaches to learning theory: what conditions are necessary for QRC to work?

  A. Sannia, R. Martínez-Peña, M. C. Soriano, G. L. Giorgi, and R. Zambrini, “Dissipation as a resource for Quantum Reservoir Computing,” Quantum 8, 1291 (2024).

- ☞ Interplay between physical properties and learning performance (e.g. chaos, thermalization, and many-body localization)

  R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, “Dynamical phase transitions in quantum reservoir computing,” Physical Review Letters 127, 100502 (2021).