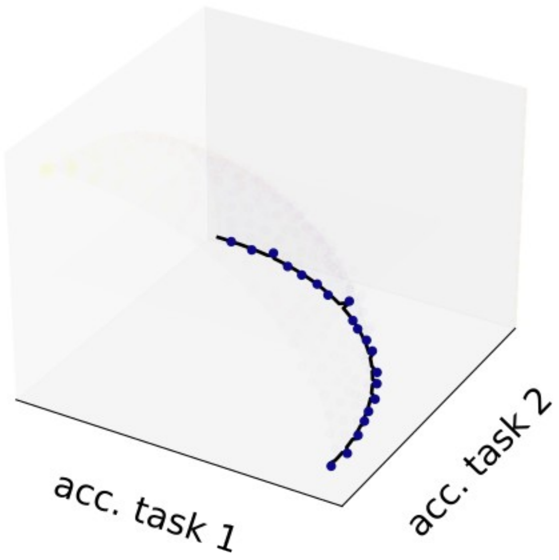
A computation-dissipation tradeoff for machine learning at the mesoscale

Alessandro Ingrosso

joint work with E. Panizon

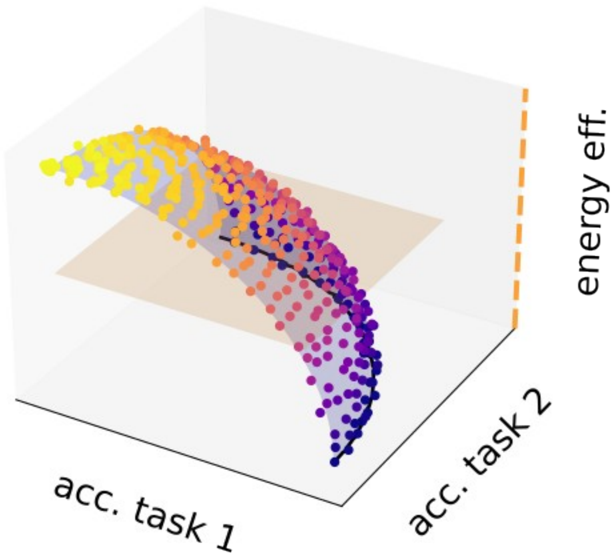
Physical Review E 109 (1), 014132

The elephant in the room of Machine Learning

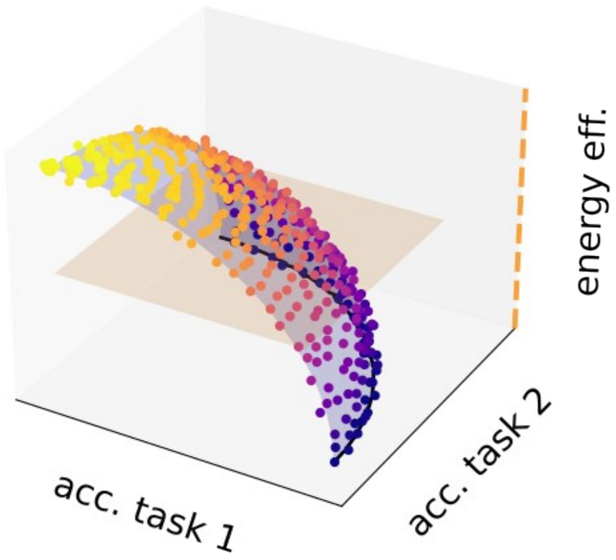


The elephant in the room of Machine Learning

The electricity bill



An energy-aware framework for Machine Learning



What is the impact of **energetic costs** on **performance** and **internal representations**?

How can we construct **cost-effective** neural networks?

Outline

- Two or Three Things I Know About Stochastic Thermodynamics

Outline

- Two or Three Things I Know About Stochastic Thermodynamics
- Stochastic Thermodynamics and computation

Outline

- Two or Three Things I Know About Stochastic Thermodynamics
- Stochastic Thermodynamics and computation
- Computation-dissipation bottleneck in a toy model of computation

Outline

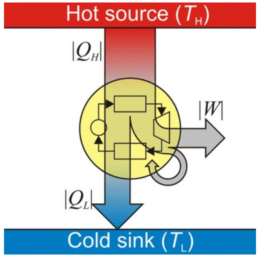
- Two or Three Things I Know About Stochastic Thermodynamics
- Stochastic Thermodynamics and computation
- Computation-dissipation bottleneck in a toy model of computation
- Supervised learning in mesoscopic networks

Outline

- Two or Three Things I Know About Stochastic Thermodynamics
- Stochastic Thermodynamics and computation
- Computation-dissipation bottleneck in a toy model of computation
- Supervised learning in mesoscopic networks
- Conclusions and Perspectives

Stochastic Themodynamics: fundamentals

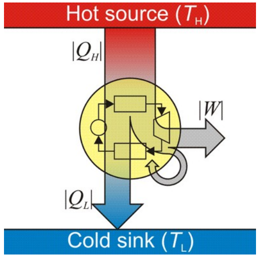
Thermodynamics



Macroscopic observable on
equilibrium ensembles

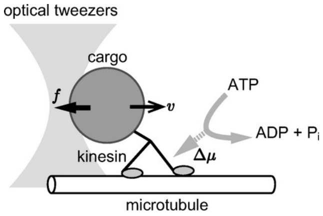
Stochastic Themodynamics: fundamentals

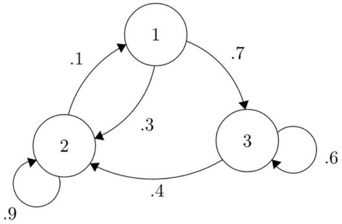
Thermodynamics



Macroscopic observable on **equilibrium ensembles**

Stochastic Thermodynamics





Fluctuating thermodynamic quantities (e.g. heat, work, entropy) on single trajectories of **mesoscopic systems**

Stochastic Themodynamics: fundamentals

Continuous-time Markov Chain:

states of a *coarse-grained* system

$$\frac{d}{dt}p\left(s,t\right)=\sum_{s'}\left[k_{ss'}\left(t\right)p\left(s',t\right)-k_{s's}\left(t\right)p\left(s,t\right)\right]$$

Stochastic Themodynamics: fundamentals

Continuous-time Markov Chain:

states of a *coarse-grained* system

$$\frac{d}{dt}p\left(s,t\right)=\sum_{s'}\left[k_{ss'}\left(t\right)p\left(s',t\right)-k_{s's}\left(t\right)p\left(s,t\right)\right]$$

Thermodynamic consistence: jump rates from interaction with a (many) reservoir(s);

$$\frac{k_{ss'}}{k_{s's}}=e^{q_{ss'}/\kappa_B T}$$

Stochastic Themodynamics: fundamentals

Continuous-time Markov Chain:

states of a *coarse-grained* system

$$\frac{d}{dt}p\left(s,t\right)=\sum_{s'}\left[k_{ss'}\left(t\right)p\left(s',t\right)-k_{s's}\left(t\right)p\left(s,t\right)\right]$$

Thermodynamic consistence: jump rates from interaction with a (many) reservoir(s);

$$\frac{k_{ss'}}{k_{s's}}=e^{q_{ss'}/\kappa_B T}$$

Entropy production:

$$\sigma=\underbrace{\frac{1}{2}\sum_{s,s'}\mathcal{J}_{ss'}\log\frac{k_{ss'}}{k_{s's}}}_{\frac{Q^{res}}{T}}+\underbrace{\frac{1}{2}\sum_{s,s'}\mathcal{J}_{ss'}\log\frac{p\left(s'\right)}{p\left(s\right)}}_{\dot{S}^{sys}}$$

Stochastic Themodynamics: fundamentals

Continuous-time Markov Chain:

states of a *coarse-grained* system

$$\frac{d}{dt}p\left(s,t\right)=\sum_{s'}\left[k_{ss'}\left(t\right)p\left(s',t\right)-k_{s's}\left(t\right)p\left(s,t\right)\right]$$

Thermodynamic consistence: jump rates from interaction with a (many) reservoir(s);

$$\frac{k_{ss'}}{k_{s's}}=e^{q_{ss'}/\kappa_B T}$$

Entropy production:

$$\sigma=\underbrace{\frac{1}{2}\sum_{s,s'}\mathcal{J}_{ss'}\log\frac{k_{ss'}}{k_{s's}}}_{\frac{Q^{res}}{T}}+\underbrace{\frac{1}{2}\sum_{s,s'}\mathcal{J}_{ss'}\log\frac{p\left(s'\right)}{p\left(s\right)}}_{\dot{S}^{sys}}$$

Langevin eq:

$$\frac{dx}{dt}=\mu_P\mathcal{F}\left(x,t\right)+\sqrt{2D}\xi\left(t\right)$$

Thermodynamic consistence: Einstein Relation

Entropy production:

$$\dot{s}^{tot}=\underbrace{\dot{s}^{res}}_{\text{reservoir}}+\underbrace{\frac{d}{dt}\left[-\kappa_B\log p\left(x,t\right)\right]}_{\text{system}}$$

⬇

$$\left\langle\frac{ds^{tot}}{dt}\right\rangle=\kappa_B\int dx\,\frac{J^2\left(x,t\right)}{Dp\left(x,t\right)}$$

Stochastic Themodynamics: fundamentals

entropy production ⟷ irreversibility

$$s^{tot}\left(\boldsymbol{x}\right) = \kappa_B \log\left(\frac{P^F_{\boldsymbol{x}}\left(\lambda\right)}{P^B_{\hat{\boldsymbol{x}}}\left(\hat{\lambda}\right)}\right)$$

Stochastic Themodynamics: fundamentals

entropy production ⟷ irreversibility

$$s^{tot}(\boldsymbol{x}) = \kappa_B \log\left(\frac{P^F_{\boldsymbol{x}}(\lambda)}{P^B_{\hat{\boldsymbol{x}}}\left(\hat{\lambda}\right)}\right)$$

Fluctuation Relations

detailed

$$\frac{P_F\left(s^{tot};\lambda\right)}{P_B\left(-s^{tot};\hat{\lambda}\right)} = e^{s^{tot}/\kappa_B T}$$

integral

$$\left\langle e^{-s^{tot}(\boldsymbol{x})/\kappa_B T}\right\rangle_F = 1$$

Stochastic Themodynamics: fundamentals

entropy production ⟷ irreversibility

$$s^{tot}(\boldsymbol{x}) = \kappa_B \log\left(\frac{P^F_{\boldsymbol{x}}(\lambda)}{P^B_{\hat{\boldsymbol{x}}}\left(\hat{\lambda}\right)}\right)$$

Fluctuation Relations

detailed

integral

$$\frac{P_F\left(s^{tot};\lambda\right)}{P_B\left(-s^{tot};\hat{\lambda}\right)} = e^{s^{tot}/\kappa_B T}$$

$$\left\langle e^{-s^{tot}(\boldsymbol{x})/\kappa_B T}\right\rangle_F = 1$$

$$\frac{P_F\left(w;\lambda\right)}{P_B\left(-w;\hat{\lambda}\right)} = e^{(w-\Delta F)/\kappa_B T}$$

$$\left\langle e^{-w(\boldsymbol{x})/\kappa_B T}\right\rangle_F = e^{-\Delta F/\kappa_B T}$$

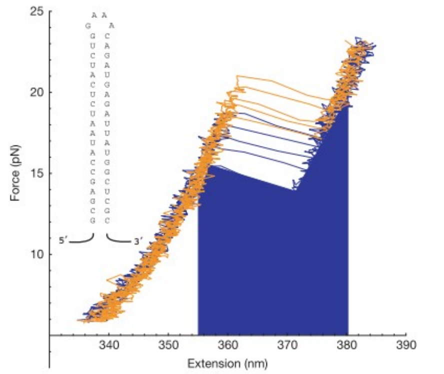
Crooks relation

Jarzinski equality

Stochastic Themodynamics: fundamentals

entropy production ⟷ irreversibility

$$s^{tot}(\boldsymbol{x}) = \kappa_B \log\left(\frac{P^F_{\boldsymbol{x}}(\lambda)}{P^B_{\hat{\boldsymbol{x}}}\left(\hat{\lambda}\right)}\right)$$



Fluctuation Relations

detailed

$$\frac{P_F\left(s^{tot};\lambda\right)}{P_B\left(-s^{tot};\hat{\lambda}\right)} = e^{s^{tot}/\kappa_B T}$$

$$\frac{P_F\left(w;\lambda\right)}{P_B\left(-w;\hat{\lambda}\right)} = e^{(w-\Delta F)/\kappa_B T}$$

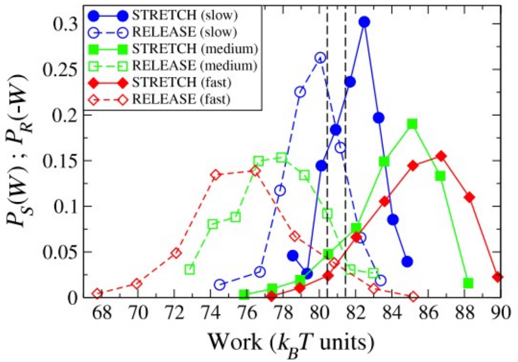
Crooks relation

integral

$$\left\langle e^{-s^{tot}(\boldsymbol{x})/\kappa_B T}\right\rangle_F = 1$$

$$\left\langle e^{-w(\boldsymbol{x})/\kappa_B T}\right\rangle_F = e^{-\Delta F/\kappa_B T}$$

Jarzinski equality



[*Collin et al, Nature 2005*]

Towards a ST of computation

Landauer Bound

Logically irreversible manipulation → **entropy increase** in non-information bearing degrees of freedom

Towards a ST of computation

Landauer Bound

Logically irreversible manipulation → **entropy increase** in non-information bearing degrees of freedom

Informally: Erasing a bit costs

$$Q \geq \kappa_B T \log 2$$

Towards a ST of computation

Landauer Bound

Logically irreversible manipulation → **entropy increase** in non-information bearing degrees of freedom

Informally: Erasing a bit costs $Q \geq \kappa_B T \log 2$

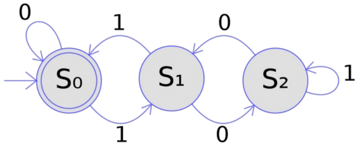
An ***abstract*** formulation of dissipation in computing machines:
- generalization of Landauer Bound;
- “mismatch” cost;
- …

mostly agnostic to implementation.

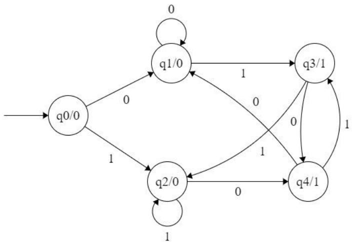
David Wolpert

Finite automata

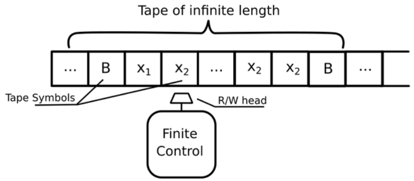
…



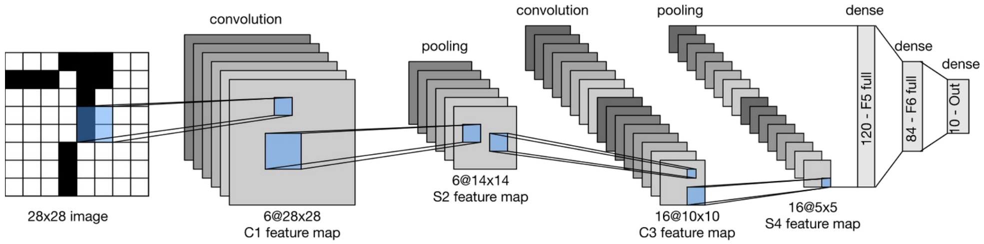
Transducers



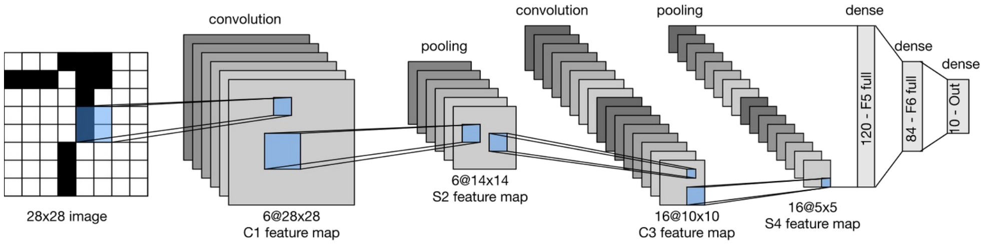
Turing Machines



Can we build a ST of neural networks?

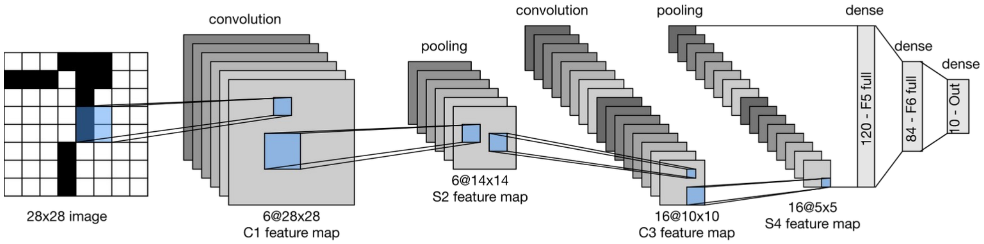


Can we build a ST of neural networks?



Not really!

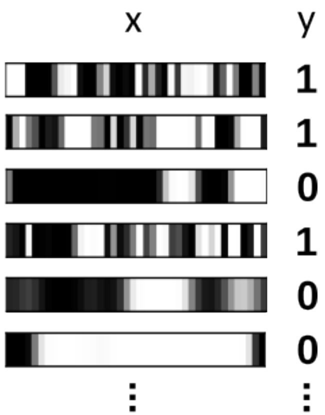
Can we build a ST of neural networks?



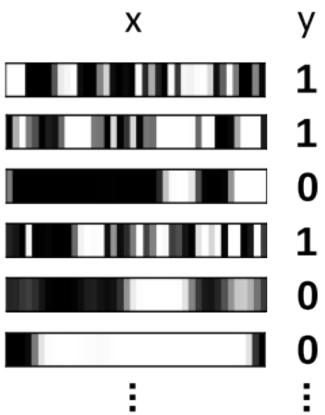
Not really! But...

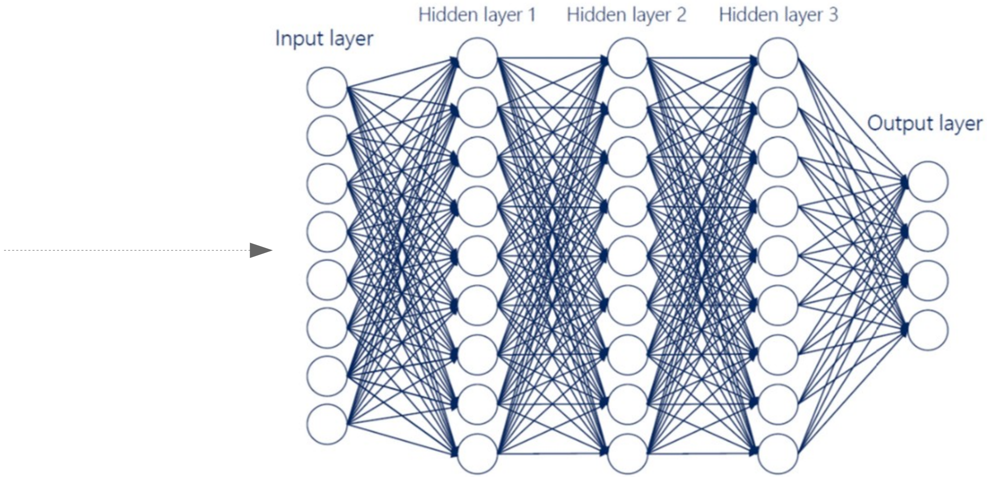
an implementation-aware theory in **mesoscopic**, **stochastic** versions of neural networks

Supervised learning in neural networks

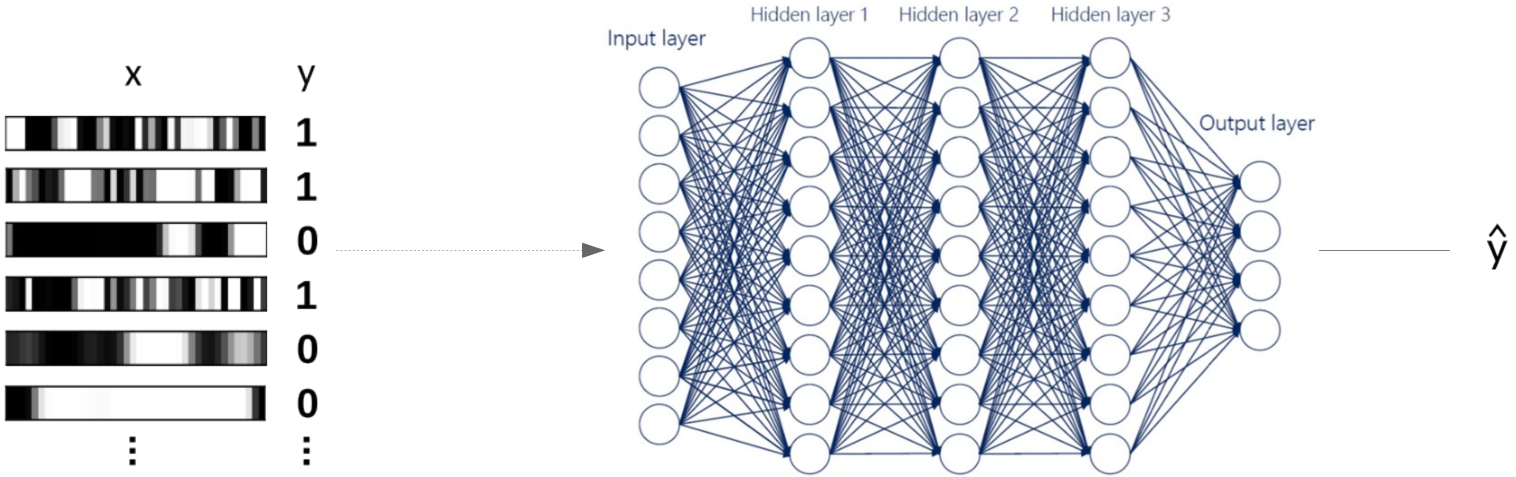


Supervised learning in neural networks

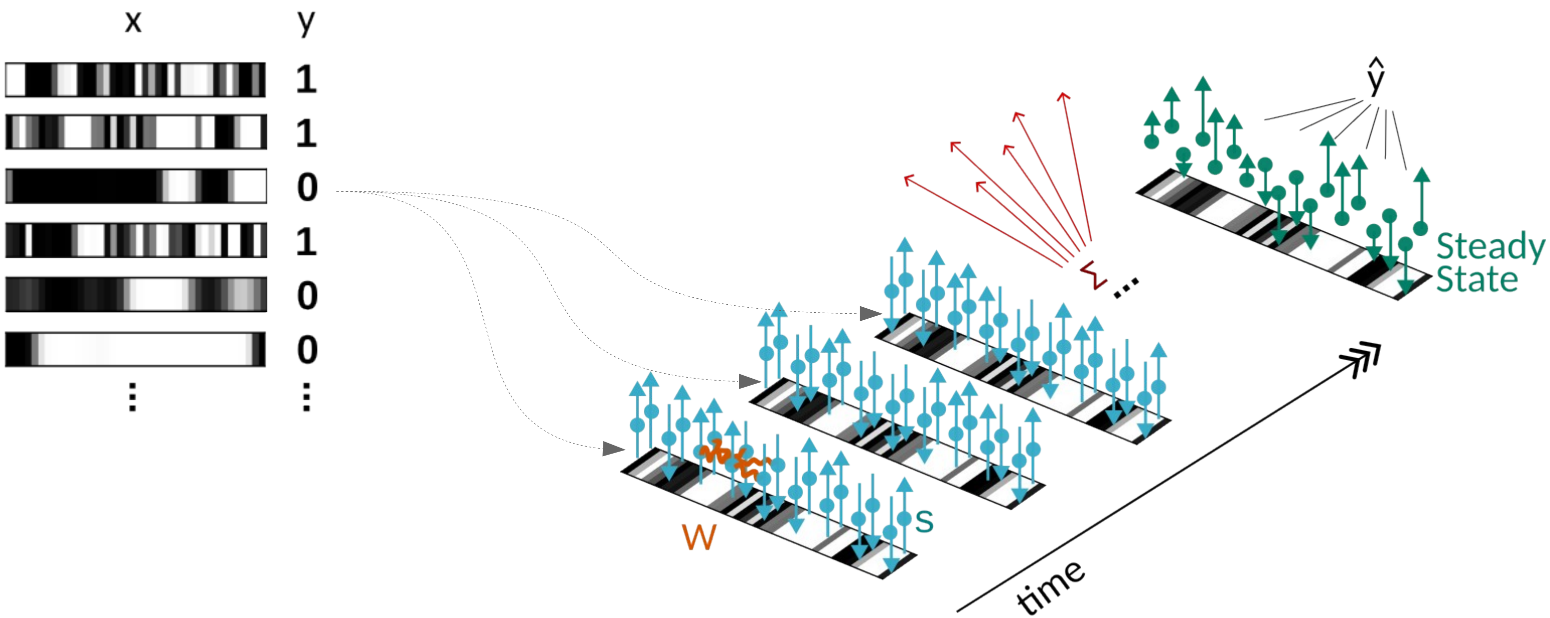




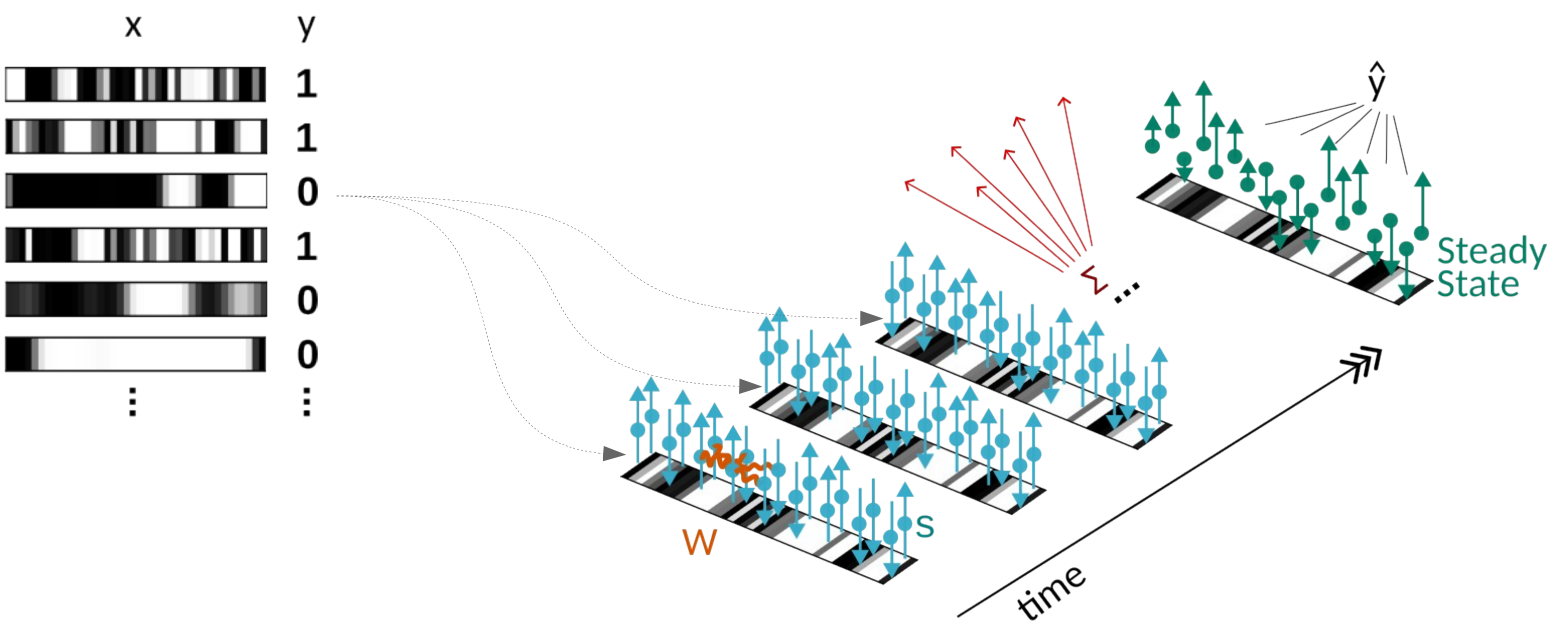
Supervised learning in neural networks



Supervised learning in mesoscopic networks



Supervised learning in mesoscopic networks



Caveat

Constraints on the connectivity should be taken into account in general for a consistent physical interpretation of our general parametrization.

Computation-dissipation bottleneck

Supervised learning task: $p(x, y)$ or $\mathcal{D} = (x, y)$

Computation-dissipation bottleneck

Supervised learning task: $p\left(x, y\right)$ or $\mathcal{D} = \left(x, y\right)$

Construct optimal representations through dynamics:

$$x \xrightarrow{\theta} s \rightarrow y$$

Computation-dissipation bottleneck

Supervised learning task: $p\,(x, y)$ or $\mathcal{D} = (x, y)$

Construct optimal representations through dynamics:

$$x \xrightarrow{\theta} s \rightarrow y$$

Trade off performance and entropy production at steady state:

$$\mathcal{L}\,(\theta) = \mathcal{G}(\theta) - \alpha\,\Sigma\,(\theta)$$

Max computational performance

Min entropy production at SS

Computation-dissipation bottleneck

Supervised learning task: $p\left(x,y\right)$ or $\mathcal{D}=\left(x,y\right)$

Construct optimal representations through dynamics:

$$x \xrightarrow{\theta} s \rightarrow y$$

Trade off performance and entropy production at steady state:

$$\mathcal{L}\left(\theta\right)=\mathcal{G}(\theta)-\alpha\,\Sigma\left(\theta\right)$$

Max computational performance

Min entropy production at SS

e.g. $\mathcal{G}=I\left(s,y\right)$

$$\left[\, I\left(s,y\right)=H\left(s\right)-H\left(s|y\right) \,\right]$$

Computation-dissipation bottleneck

Supervised learning task: $p\,(x, y)$ or $\mathcal{D} = (x, y)$

Construct optimal representations through dynamics:

$$x \xrightarrow{\theta} s \rightarrow y$$

Trade off performance and entropy production at steady state:

$$\mathcal{L}\,(\theta) = \mathcal{G}(\theta) - \alpha\,\Sigma\,(\theta)$$

Max computational performance

Min entropy production at SS

e.g. $\mathcal{G} = I\,(s, y)$

$$I\,(s, y) = H\,(s) - H\,(s|y)$$

$$\Sigma(\theta) = \sum_x p\,(x)\sigma\,(x, \theta)$$

or

$$\Sigma(\theta) = \frac{1}{|\mathcal{D}|} \sum_x \sigma\,(x, \theta)$$

Computation-dissipation bottleneck

Supervised learning task: $p\left(x,y\right)$ or $\mathcal{D}=\left(x,y\right)$

Construct optimal representations through dynamics:

$$x\xrightarrow{\theta}s\rightarrow y$$

Trade off performance and entropy production at steady state:

$$\mathcal{L}\left(\theta\right)=\mathcal{G}(\theta)-\alpha\,\Sigma\left(\theta\right)$$

Max computational performance

Min entropy production at SS

e.g. $\mathcal{G}=I\left(s,y\right)$

$$I\left(s,y\right)=H\left(s\right)-H\left(s|y\right)$$

$$\Sigma(\theta)=\sum_{x}p\left(x\right)\sigma\left(x,\theta\right)$$

or

$$\Sigma(\theta)=\frac{1}{|\mathcal{D}|}\sum_{x}\sigma\left(x,\theta\right)$$

Information Bottleneck (Tishby et al)

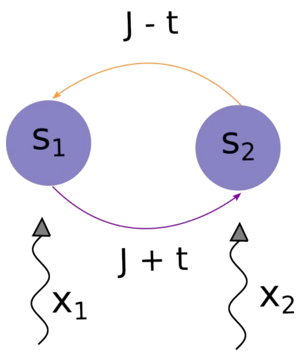
$$x\rightarrow s\rightarrow y$$

$$\mathcal{L}\left[p\left(s|x\right)\right]=I\left(s,x\right)-\beta I\left(s,y\right)$$

Max information compression

Min loss of *task-relevant* information

A tractable 2d **spin** system



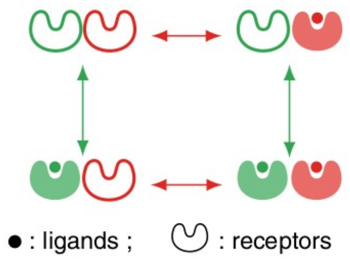
Rates:

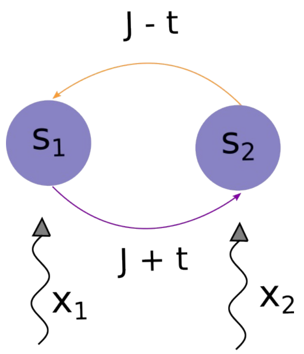
$$k_s^{(i)} \propto e^{-\beta s_i (Ws+x)_i}$$

A tractable 2d **spin** system

A plausible model for a **sensory complex:**

i) coupled chemoreceptors
ii) transcription regulation with cross-feedback
iii) ...



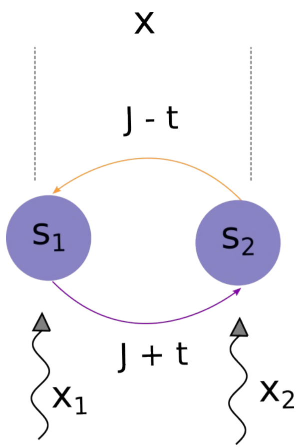


Rates:

$$k_s^{(i)} \propto e^{-\beta s_i (Ws+x)_i}$$

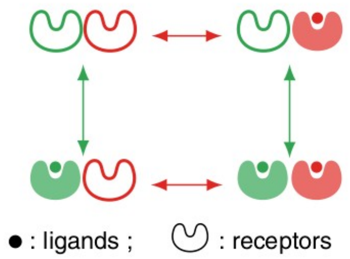
[*Ngampruetikorn et al, Nat Comm (2020)*]

A tractable 2d **spin** system



A plausible model for a **sensory complex:**

i) coupled chemoreceptors
ii) transcription regulation with cross-feedback
iii) ...

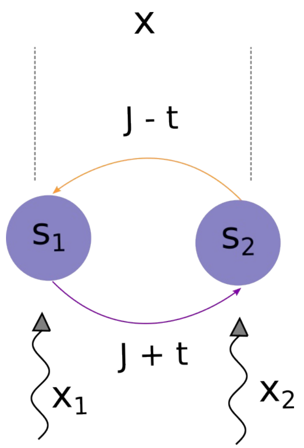


Rates:

$$k_s^{(i)} \propto e^{-\beta s_i (Ws + x)_i}$$

[*Ngampruetikorn et al, Nat Comm (2020)*]

A tractable 2d **spin** system

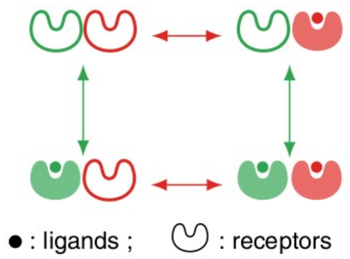


Rates:

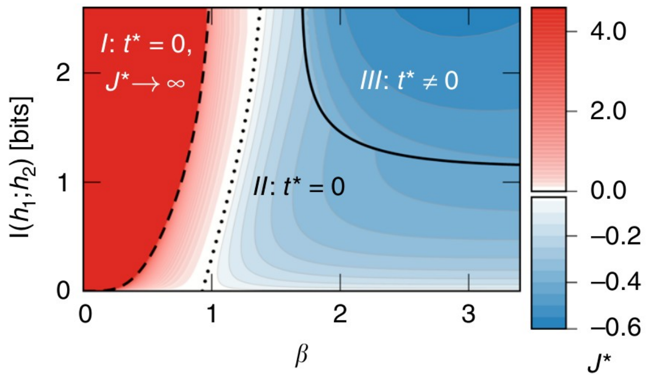
$$k_s^{(i)} \propto e^{-\beta s_i (Ws+x)_i}$$

A plausible model for a **sensory complex:**

i) coupled chemoreceptors
ii) transcription regulation with cross-feedback
iii) ...

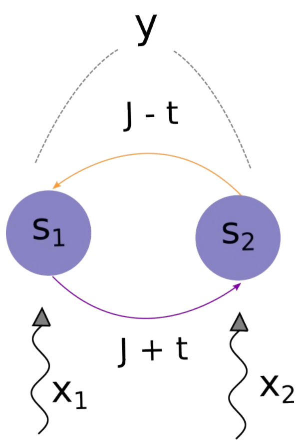


Nonequilibrium steady states are optimal for $I\,(s,x)$ at high correlation and small noise



[*Ngampruetikorn et al, Nat Comm (2020)*]

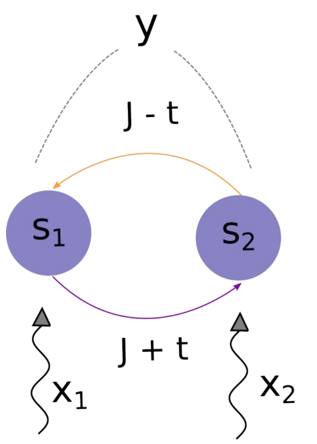
A tractable 2d **spin** system



Rates:

$$k_s^{(i)} \propto e^{-\beta s_i (Ws + x)_i}$$

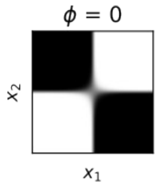
A tractable 2d **spin** system



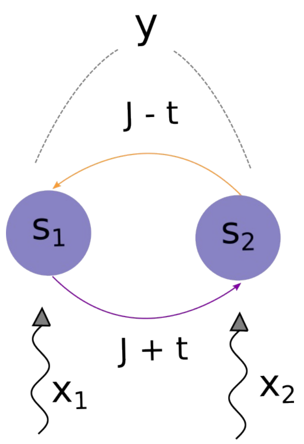
Rates:

$$k_s^{(i)} \propto e^{-\beta s_i (Ws + x)_i}$$

Input-output rule:
Stochastic parity gate



A tractable 2d **spin** system

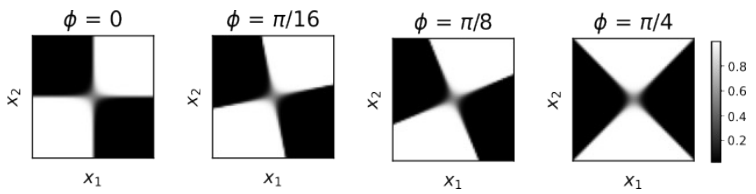


Rates:

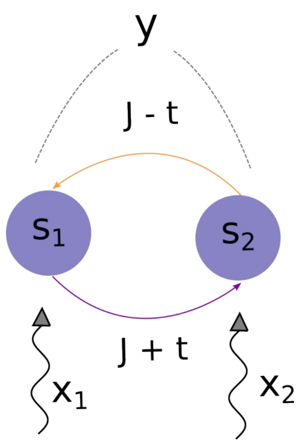
$$k_s^{(i)} \propto e^{-\beta s_i (Ws+x)_i}$$

Input-output rule:
Stochastic parity gate

$$p\left(y=1|x\right) = \mathrm{sigmoid}\left(\eta R^{\phi} x_1 R^{\phi} x_2\right)$$



A tractable 2d **spin** system

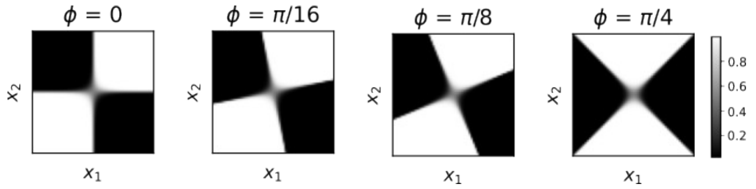


Rates:

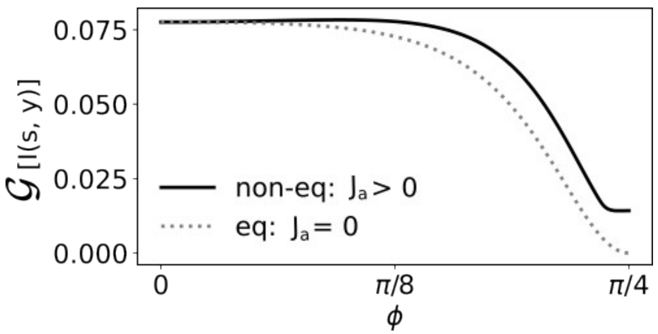
$$k_s^{(i)} \propto e^{-\beta s_i (Ws+x)_i}$$

Input-output rule:
Stochastic parity gate

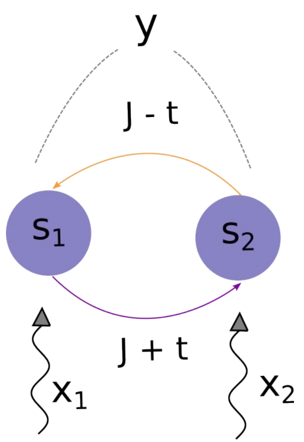
$$p\left(y=1|x\right) = \mathrm{sigmoid}\left(\eta R^{\phi} x_1 R^{\phi} x_2\right)$$



Optimal performance:
equilibrium vs **nonequilibrium**

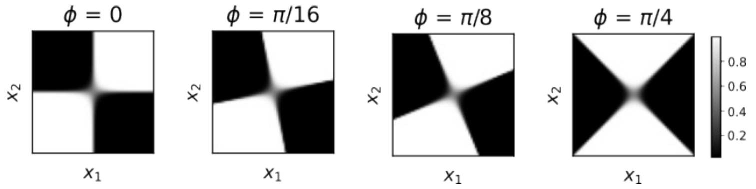


A tractable 2d **spin** system

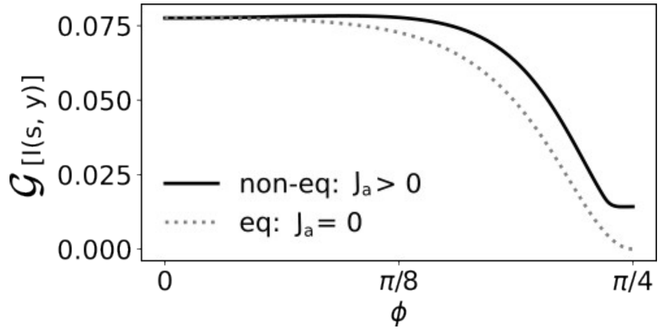


Input-output rule:
Stochastic parity gate

$$p\left(y=1|x\right)=\mathrm{sigmoid}\left(\eta R^{\phi}x_1 R^{\phi}x_2\right)$$



Optimal performance:
equilibrium vs **nonequilibrium**

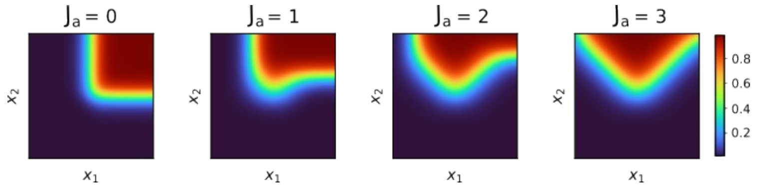


Rates:

$$k_s^{(i)} \propto e^{-\beta s_i (Ws+x)_i}$$

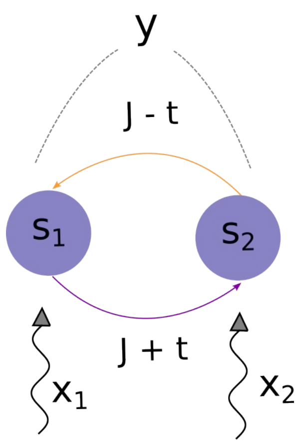
Enhanced **expressivity** of
nonequilibrium steady state

$$p\left(s=(+1,+1)|x\right)$$



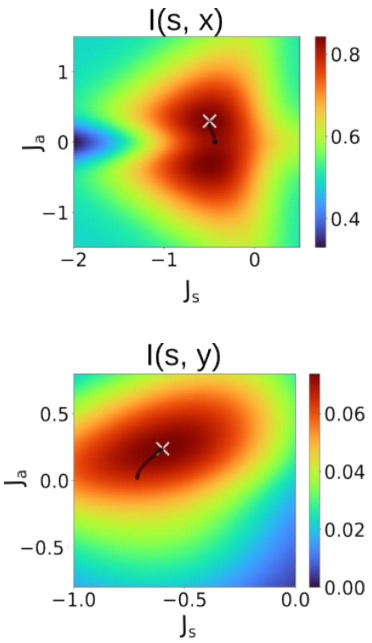
A tractable 2d **spin** system

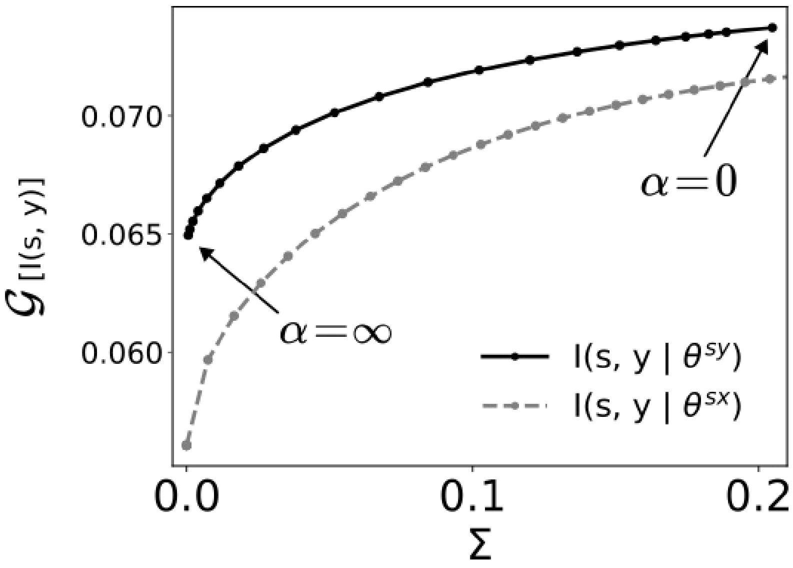
Optimal Input Information	$\neq$	Optimal **Task-relevant** Information



Rates:

$$k_s^{(i)} \propto e^{-\beta s_i (Ws + x)_i}$$





*Quick aside: A tractable **linear** system*

A multi-dimensional Ornstein-Uhlenbeck process:

$$\dot{s} = Ws + x + \sigma_s \xi$$

A stochastic linear input-output rule:

$$y = w_0^T x + \xi_y$$

*Quick aside: A tractable **linear** system*

A multi-dimensional Ornstein-Uhlenbeck process:

$$\dot{s} = Ws + x + \sigma_s \xi$$

A stochastic linear input-output rule:

$$y = w_0^T x + \xi_y$$

$$I(s, y) = \frac{1}{2} \log \det \left(W^{-1} C_x W^{-T} + C \right) + - \frac{1}{2} \log \det \left(C_s - C_{sy} C_y^{-1} C_{ys} \right)$$

with:

$$WC + CW^T + \sigma_s^2 \mathcal{I} = 0$$

$$\sigma = \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \mathcal{E}(\omega)$$

with:

$$\mathcal{E}(\omega) = \frac{1}{2} \mathrm{Tr} \left[C(\omega) C^{-1}(-\omega) - \mathcal{I} \right]$$

*Quick aside: A tractable **linear** system*

A multi-dimensional Ornstein-Uhlenbeck process:

$$\dot{s} = Ws + x + \sigma_s \xi$$

A stochastic linear input-output rule:

$$y = w_0^T x + \xi_y$$

$$I\left(s,y\right) = \frac{1}{2}\log\det\left(W^{-1}C_x W^{-T} + C\right) + - \frac{1}{2}\log\det\left(C_s - C_{sy}C_y^{-1}C_{ys}\right)$$

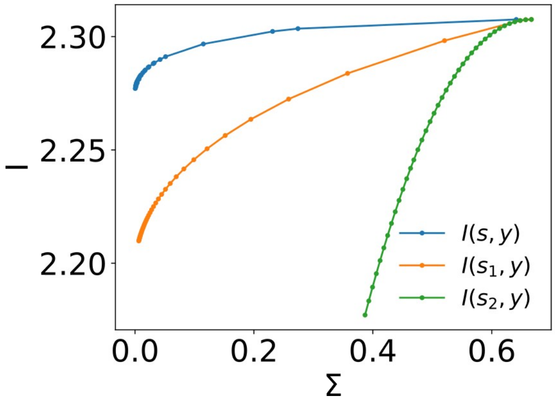
with:

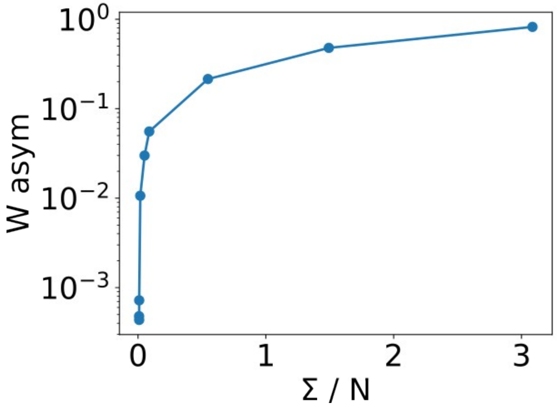
$$WC + CW^T + \sigma_s^2 \mathcal{I} = 0$$

$$\sigma = \int_{-\infty}^{+\infty} \frac{d\omega}{2\pi} \mathcal{E}\left(\omega\right)$$

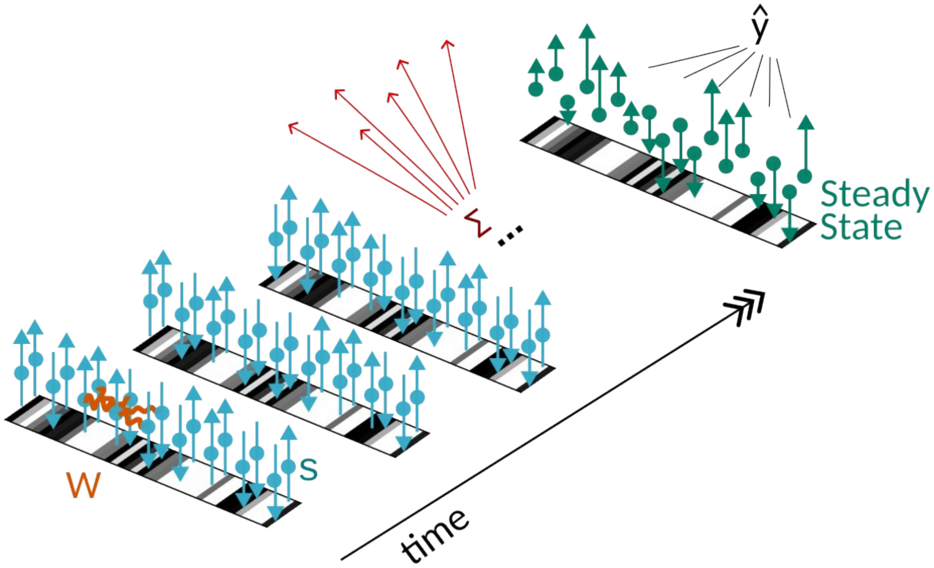
with:

$$\mathcal{E}\left(\omega\right) = \frac{1}{2}\mathrm{Tr}\left[C\left(\omega\right)C^{-1}\left(-\omega\right) - \mathcal{I}\right]$$

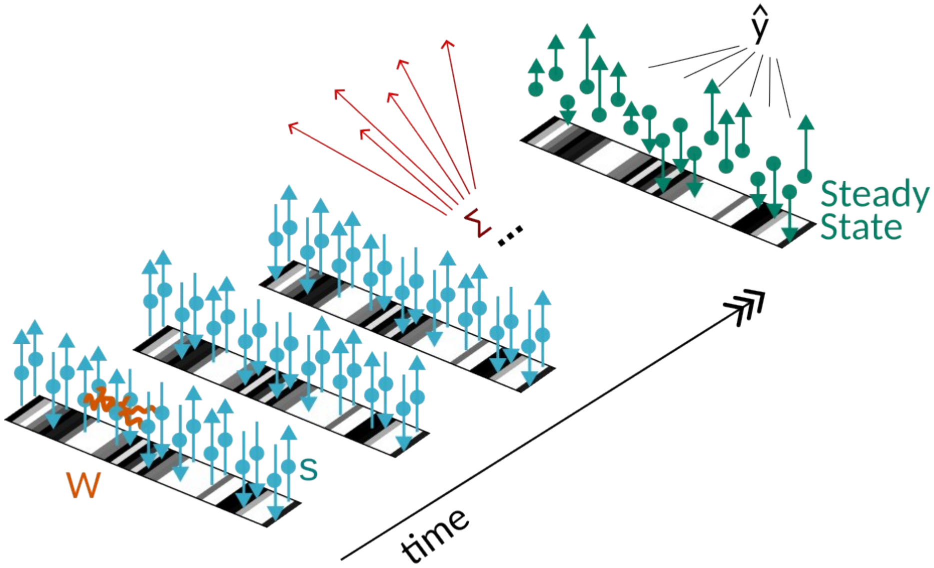




Training mesoscopic networks



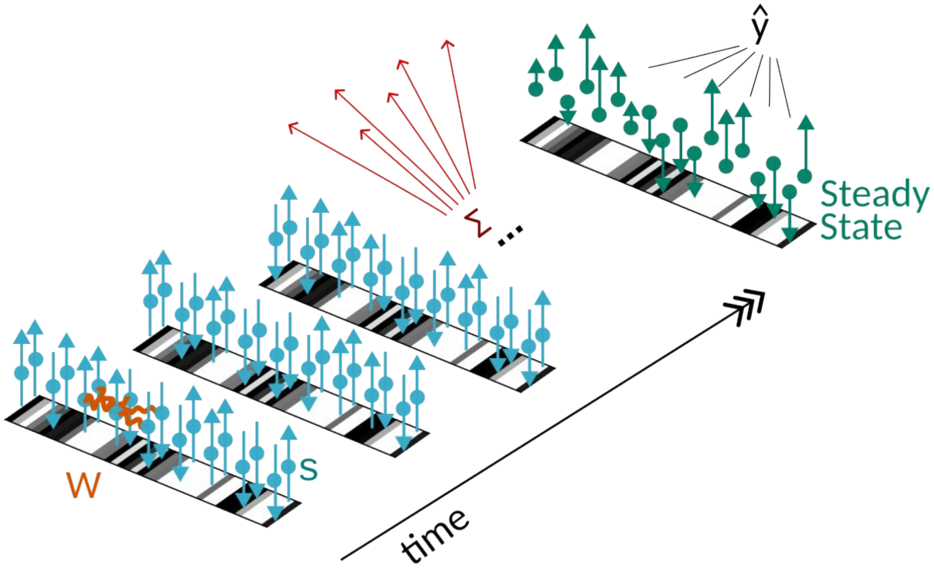
Training mesoscopic networks



$m_x = \langle s|x \rangle$

Average spin activity

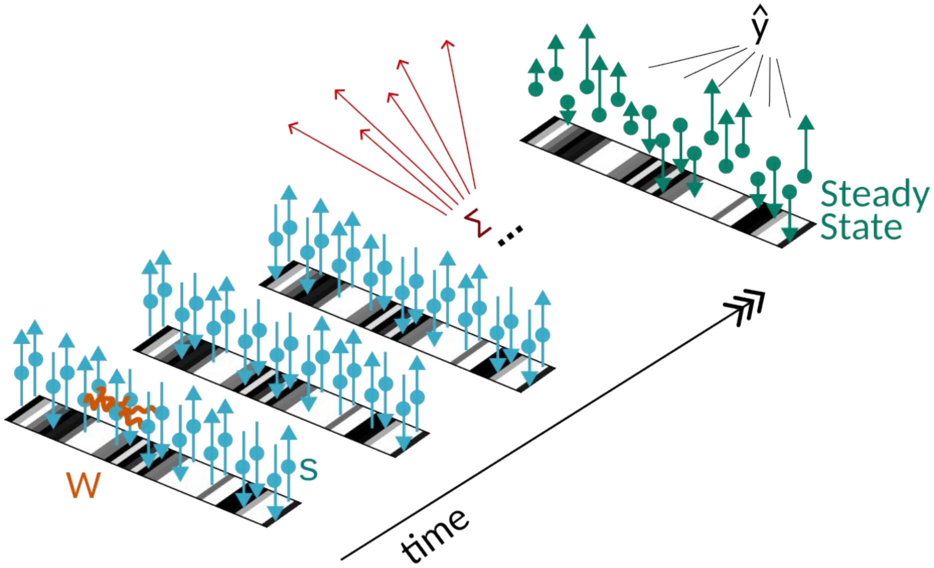
Training mesoscopic networks



$\hat{y} = W_{out} m_x$ — Linear readout

$m_x = \langle s | x \rangle$ — Average spin activity

Training mesoscopic networks



$\hat{y} = W_{out} m_x$ Linear readout

$m_x = \langle s|x \rangle$ Average spin activity

Performance: - loss function

Mean Square Error $\mathcal{G} = -\frac{1}{2M}\sum_{\mu=1}^{M}(y^\mu - W_{out} m_{x^\mu})^2$

Cross-entropy $\mathcal{G} = \frac{1}{M}\sum_{\mu=1}^{M}\log p^{\mu}_{y^\mu}$

Training mesoscopic networks

NEURO

MESO

Training mesoscopic networks

NEURO

MESO

Forward pass

Training mesoscopic networks

NEURO

MESO

Forward pass

Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**

Training mesoscopic networks

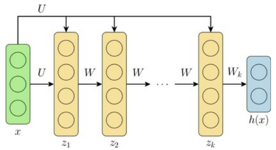
NEURO

Forward pass

MESO

Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**

ML: implicit layers
[Bai et al, NeurIPS 2019]



Mean-field approx in kinetic Ising models
[Aguilera et al, Nat Comm (2021)]

$$m_{t+1} = \tanh\left(Jm_t + h_t\right)$$

Training mesoscopic networks

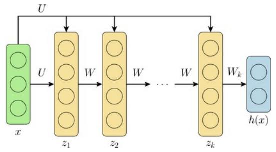
NEURO

MESO

Forward pass

Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**

ML: implicit layers
[Bai et al, NeurIPS 2019]



Mean-field approx in kinetic Ising models
[Aguilera et al, Nat Comm (2021)]

$$m_{t+1} = \tanh\left(J m_t + h_t\right)$$

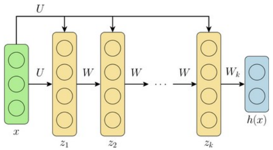
(Stochastic) Gradient Descent through automatic differentiation

Training mesoscopic networks

NEURO

MESO

ML: implicit layers [*Bai et al, NeurIPS 2019*]



Mean-field approx in kinetic Ising models [*Aguilera et al, Nat Comm (2021)*]

$$m_{t+1} = \tanh\left(J m_t + h_t\right)$$

Forward pass

Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**

(Stochastic) Gradient Descent through automatic differentiation

(Stochastic) Gradient Descent via **Simultaneous Perturbation Stochastic Approximation** [*Spaal IEEE Trans Aut Contr (1992)*]

Training mesoscopic networks

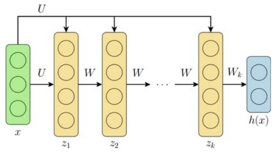
NEURO

MESO

Forward pass

Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**

ML: implicit layers
[Bai et al, NeurIPS 2019]



Mean-field approx in kinetic Ising models
[Aguilera et al, Nat Comm (2021)]

$$m_{t+1} = \tanh\left(J m_t + h_t\right)$$

(Stochastic) Gradient Descent through automatic differentiation

(Stochastic) Gradient Descent via **Simultaneous Perturbation Stochastic Approximation**
[Spaal IEEE Trans Aut Contr (1992)]

<u>Step 1</u>: run two Gillespie sims with perturbed couplings

$$W^{\pm} = W \pm \delta W$$

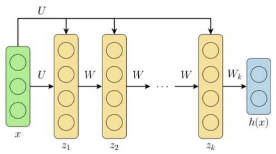
<u>Step 2</u>: compute gradient approximation

$$\nabla \mathcal{L}|_W \approx \left[\mathcal{L}^{+} - \mathcal{L}^{-}\right] \frac{\delta W}{2|\delta W|}$$

Training mesoscopic networks

NEURO	MESO
Forward pass	Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**
(Stochastic) Gradient Descent through automatic differentiation	(Stochastic) Gradient Descent via **Simultaneous Perturbation Stochastic Approximation** *[Spaal IEEE Trans Aut Contr (1992)]*
L2/L1 regularization, batch/layer norm, activity regularization...	

ML: implicit layers *[Bai et al, NeurIPS 2019]*



Mean-field approx in kinetic Ising models *[Aguilera et al, Nat Comm (2021)]*

$$m_{t+1} = \tanh\left(Jm_t + h_t\right)$$

Step 1: run two Gillespie sims with perturbed couplings

$$W^{\pm} = W \pm \delta W$$

Step 2: compute gradient approximation

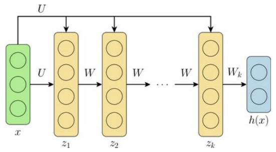
$$\nabla \mathcal{L}|_W \approx \left[\mathcal{L}^+ - \mathcal{L}^-\right] \frac{\delta W}{2|\delta W|}$$

Training mesoscopic networks

NEURO

MESO

ML: implicit layers
[Bai et al, NeurIPS 2019]



Forward pass

Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**

Mean-field approx in kinetic Ising models
[Aguilera et al, Nat Comm (2021)]

$$m_{t+1} = \tanh\left(Jm_t + h_t\right)$$

(Stochastic) Gradient Descent through automatic differentiation

(Stochastic) Gradient Descent via **Simultaneous Perturbation Stochastic Approximation** *[Spaal IEEE Trans Aut Contr (1992)]*

<u>Step 1</u>: run two Gillespie sims with perturbed couplings

$$W^{\pm} = W \pm \delta W$$

<u>Step 2</u>: compute gradient approximation

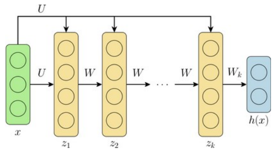
$$\nabla \mathcal{L}|_W \approx \left[\mathcal{L}^+ - \mathcal{L}^-\right] \frac{\delta W}{2|\delta W|}$$

L2/L1 regularization, batch/layer norm, activity regularization...

Entropy production regularization

Training mesoscopic networks

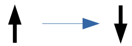
NEURO	**MESO**	
Forward pass	Convergence of **Gillespie** dynamics to a **Non Equilibrium Steady State**	ML: implicit layers *[Bai et al, NeurIPS 2019]* Mean-field approx in kinetic Ising models *[Aguilera et al, Nat Comm (2021)]*
(Stochastic) Gradient Descent through automatic differentiation	(Stochastic) Gradient Descent via **Simultaneous Perturbation Stochastic Approximation** *[Spaal IEEE Trans Aut Contr (1992)]*	Step 1: run two Gillespie sims with perturbed couplings Step 2: compute gradient approximation
L2/L1 regularization, batch/layer norm, activity regularization...	**Entropy production** regularization	Online calculation at each spin flip *[Martynec et al, NJP (2020)]*



$$m_{t+1} = \tanh\left(Jm_t + h_t\right)$$

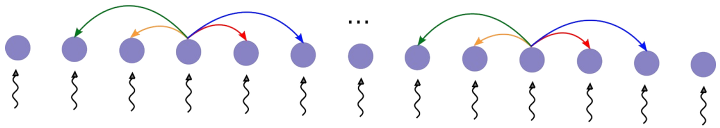
$$W^{\pm} = W \pm \delta W$$

$$\nabla \mathcal{L}|_W \approx \left[\mathcal{L}^+ - \mathcal{L}^-\right] \frac{\delta W}{2|\delta W|}$$



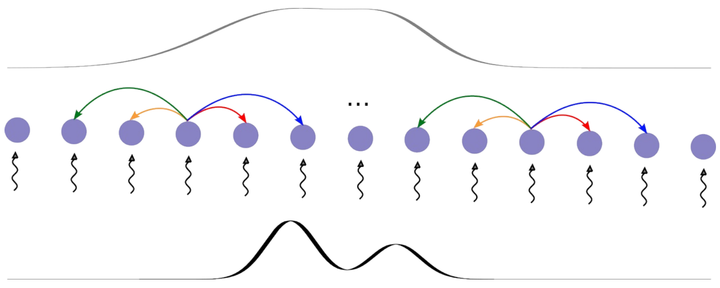
Supervised learning in mesoscopic networks

Mesoscopic equivalent to a **Convolutional Neural Network**



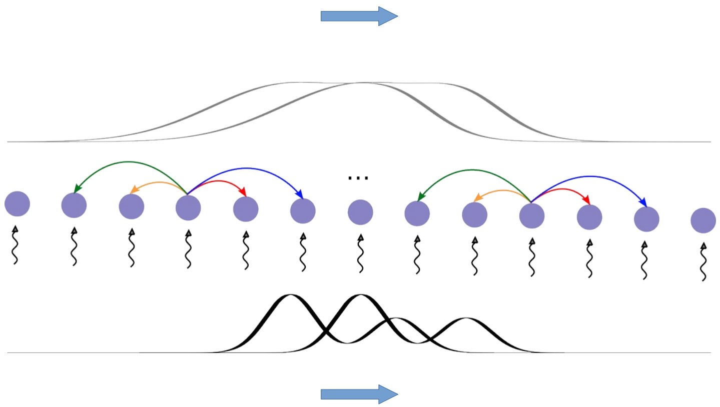
Supervised learning in mesoscopic networks

Mesoscopic equivalent to a **Convolutional Neural Network**



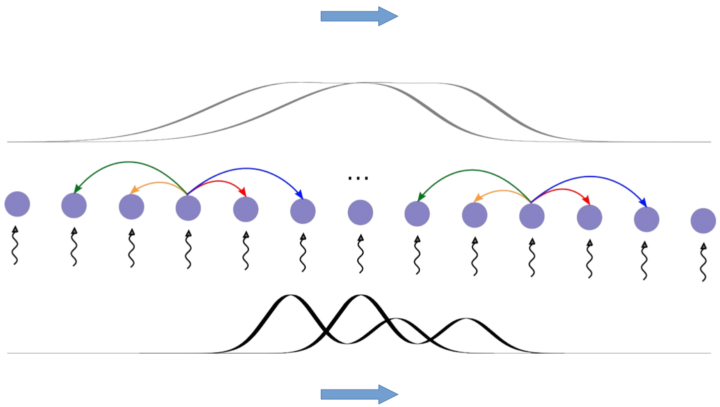
Supervised learning in mesoscopic networks

Mesoscopic equivalent to a **Convolutional Neural Network**

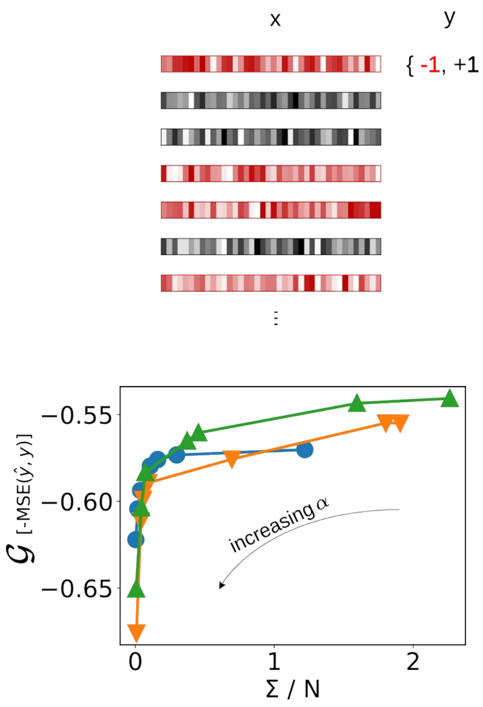


Supervised learning in mesoscopic networks

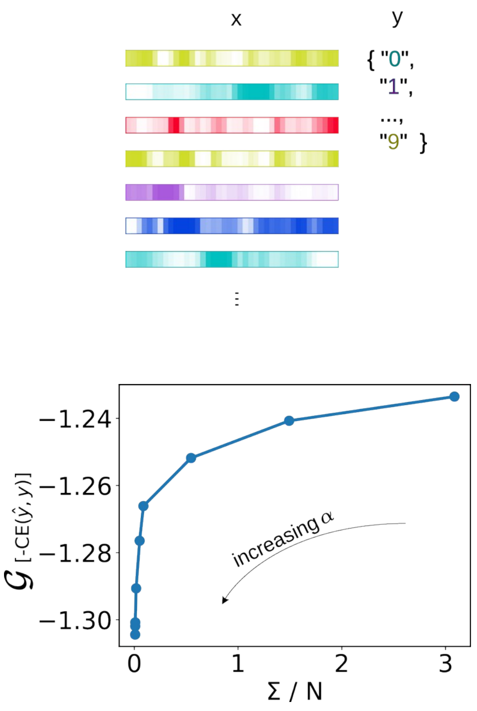
Mesoscopic equivalent to a **Convolutional Neural Network**



Random Input-Output

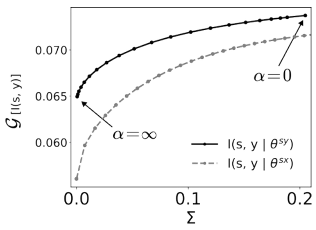


MNIST1D

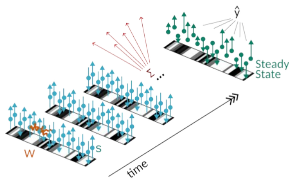


Conclusion

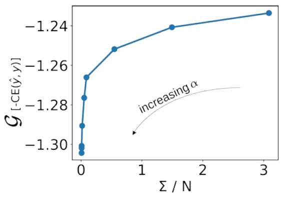
- Optimization of **task-relevant information** can lead to non-equilibrium systems independently of input information optimization



- Mesoscopic sytems can be ***trained*** to perform supervised learning problems at the stationary state

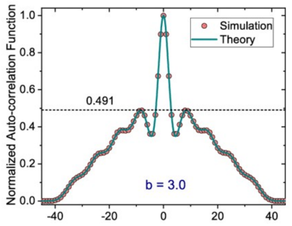


- **Non-reciprocity** of interaction enhances **expressivity** in computation at the NESS at the cost of positive entropy production

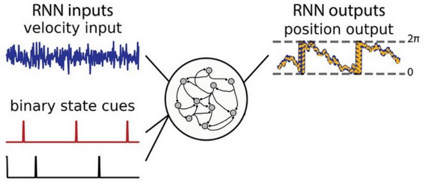


Perspectives

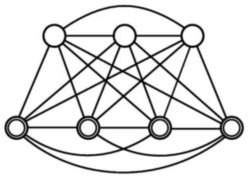
- From one-**time statistics** to **time correlations** at steady state.



- **Computation through transients**: non-stationary protocols → speed-accuracy-dissipation tradeoffs



- Stochastic networks with **hidden units**:
  - higher-order statistical interactions
  - stochastic attention mechanism?



THE END