Stochastic quantum thermodynamics for nanoelectronics

G J Milburn

UNIVERSITY OF SUSSEX

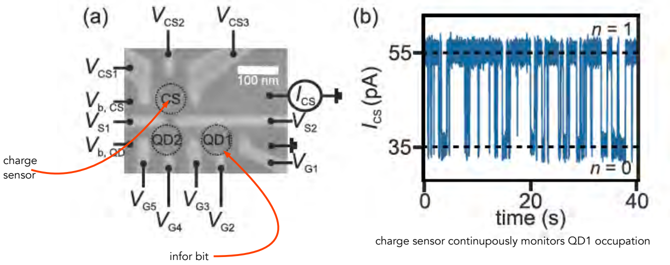
National Quantum Computing Centre

What is stochastic thermodynamics?

PHYSICAL REVIEW RESEARCH 7, L032017 (2025)

Rapid optimal work extraction from a quantum-dot information engine

Kushagra Aggarwal, Alberto Rolandi, Yikai Yang, Joseph Hickie, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Mark T. Mitchison, Martí Perarnau-Llobet, and Natalia Ares

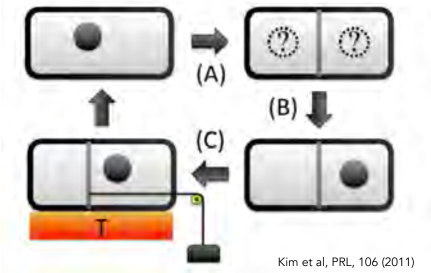


charge sensor continupously monitors QD1 occupation

germanium quantum well.

What is stochastic thermodynamics?

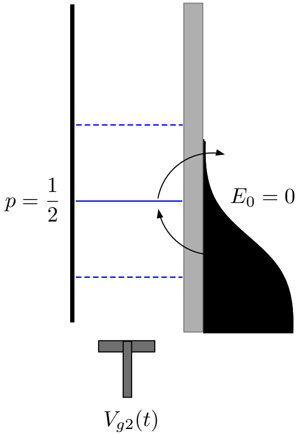
Szilard engine:



Kim et al, PRL, 106 (2011)

What is stochastic thermodynamics?

steady state



What is stochastic thermodynamics?

occupation probability

$$\dot{p} = \Gamma_{in}(\varepsilon)(1-p) - \Gamma_{out}(\varepsilon)p$$

$\Gamma_{in}(\varepsilon) = \gamma f(\varepsilon) \qquad \Gamma_{out}(\varepsilon) = \gamma(1 - f(\varepsilon))$

What is stochastic thermodynamics?

occupation probability

$$\dot{p} = \Gamma_{in}(\varepsilon)(1-p) - \Gamma_{out}(\varepsilon)p$$

$\Gamma_{in}(\varepsilon) = \gamma f(\varepsilon) \qquad \Gamma_{out}(\varepsilon) = \gamma(1 - f(\varepsilon))$

Detailed balance:

$$\frac{\Gamma_{in}}{\Gamma_{out}} = e^{-\beta\varepsilon_0} = 1$$

Choose $\varepsilon_0 = 0$ and $\gamma = 4/3$

What is stochastic thermodynamics?

occupation probability

$$\dot{p} = \Gamma_{in}(\varepsilon)(1-p) - \Gamma_{out}(\varepsilon)p$$

$\Gamma_{in}(\varepsilon) = \gamma f(\varepsilon) \qquad \Gamma_{out}(\varepsilon) = \gamma(1 - f(\varepsilon))$

Detailed balance:

$$\frac{\Gamma_{in}}{\Gamma_{out}} = e^{-\beta\varepsilon_0} = 1$$

Choose $\varepsilon_0 = 0$ and $\gamma = 4/3$

at $\varepsilon = 0$

$$\dot{p} = \frac{2}{3} - \frac{4}{3}p$$

$$p_{ss} = \tfrac{1}{2}$$

What is stochastic thermodynamics?

Vary gate bias $\varepsilon(t) = \varepsilon_0 + (\varepsilon_f - \varepsilon_0)t/\tau$

What is stochastic thermodynamics?

Vary gate bias $\varepsilon(t) = \varepsilon_0 + (\varepsilon_f - \varepsilon_0)t/\tau$

$$\frac{\Gamma_{in}(0)}{\Gamma_{out}(0)} = 1 \to \frac{\Gamma_{in}(\tau)}{\Gamma_{out}(\tau)} = e^{-\beta\varepsilon_f}$$

$$p = \frac{1}{2} \to \frac{e^{-\beta\varepsilon_f}}{1 + e^{-\beta\varepsilon_f}}$$

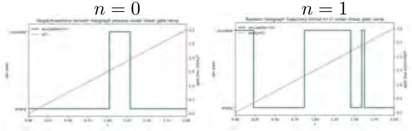
What is stochastic thermodynamics?

Vary gate bias $\varepsilon(t) = \varepsilon_0 + (\varepsilon_f - \varepsilon_0)t/\tau$

$$\frac{\Gamma_{in}(0)}{\Gamma_{out}(0)} = 1 \to \frac{\Gamma_{in}(\tau)}{\Gamma_{out}(\tau)} = e^{-\beta\varepsilon_f}$$

$$p = \frac{1}{2} \to \frac{e^{-\beta\varepsilon_f}}{1 + e^{-\beta\varepsilon_f}}$$

sample random telegraph:



What is stochastic thermodynamics?

Work done on the system:

$$W = \int_0^\tau n(t)\dot{\varepsilon}(t)dt$$

is stochastic in each trial.

What is stochastic thermodynamics?

Jarzynski equality.

$$\underbrace{\langle e^{-\beta w}\rangle}_{\text{av. over trials}} = e^{-\beta \Delta F}$$

$\beta = (k_B T)^{-1}$

ΔF: change in free energy between initial and final thermal equilibrium states.

What is stochastic thermodynamics?

Jarzynski equality.

$$\underbrace{\langle e^{-\beta w} \rangle}_{\text{av. over trials}} = e^{-\beta \Delta F}$$

$\beta = (k_B T)^{-1}$

ΔF: change in free energy between initial and final <u>thermal equilibrium</u> states.

$\varepsilon = 0 \rightarrow \varepsilon_f = 3/\beta,$

$\beta = 1, \tau = 10.0, N = 50,000 : \quad e^{-\beta \Delta F} = 0.525, \quad \langle e^{-\beta W} \rangle = 0.467.$

What is stochastic thermodynamics?

Jarzynski equality.

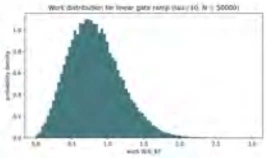
$$\underbrace{\langle e^{-\beta w}\rangle}_{\text{av. over trials}} = e^{-\beta\Delta F}$$

$\beta = (k_B T)^{-1}$

ΔF: change in free energy between initial and final thermal equilibrium states.

$\varepsilon = 0 \to \varepsilon_f = 3/\beta,$

$\beta = 1, \tau = 10.0, N = 50,000: \quad e^{-\beta\Delta F} = 0.525, \quad \langle e^{-\beta W}\rangle = 0.467.$



What is stochastic thermodynamics?

Szilard engine: measure and control.

What is stochastic thermodynamics?

Szilard engine: measure and control.

Modified JE: Sagawa–Ueda, PRL 2012

classical stochastic system X interacting with another degree of freedom Y (memory, controller, demon) via measurement and feedback, and define a trajectory-level entropy production and mutual-information change.

What is stochastic thermodynamics?

Szilard engine: measure and control.

Modified JE: Sagawa–Ueda, PRL 2012

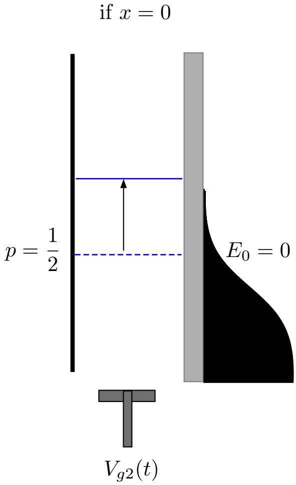
classical stochastic system X interacting with another degree of freedom Y (memory, controller, demon) via measurement and feedback, and define a trajectory-level entropy production and mutual-information change.

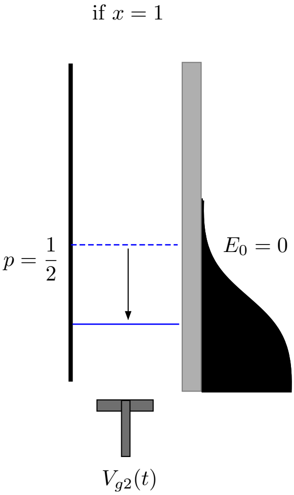
$$\langle e^{-\beta W} \rangle = e^{-\beta \Delta F + I}$$

I is (roughly) the mutual information between system and measurement outcomes accumulated during the protocol.

What is stochastic thermodynamics?

conditional quench:





What is stochastic thermodynamics?

$x = 0$ dot is empty, quench does no work.

What is stochastic thermodynamics?

$x = 0$ dot is empty, quench does no work.

Lower energy via gate bias $V_g(t)$, and $W > 0$ energy gained from system.

What is stochastic thermodynamics?

$x = 0$ dot is empty, quench does no work.

Lower energy via gate bias $V_g(t)$, and $W > 0$ energy gained from system.

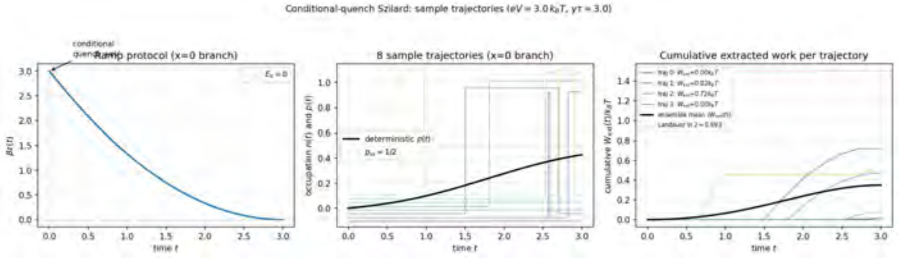
Nonlinear bias $\varepsilon(t)$ ... optimal.

What is stochastic thermodynamics?

$x = 0$ dot is empty, quench does no work.

Lower energy via gate bias $V_g(t)$, and $W > 0$ energy gained from system.

Nonlinear bias $\varepsilon(t)$ ... optimal.



What is stochastic thermodynamics?

$x = 1$ dot is charged, quench to $-eV$.

What is stochastic thermodynamics?

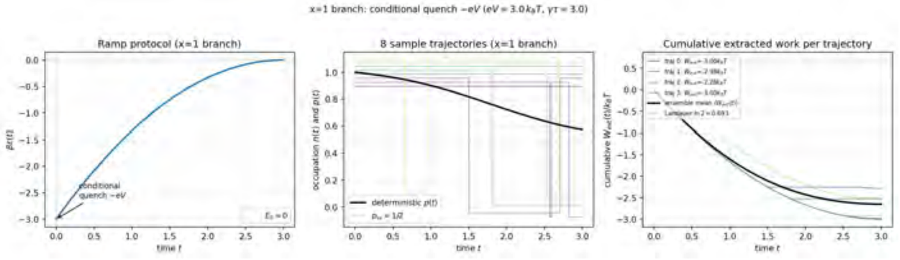
$x = 1$ dot is charged, quench to $-eV$.

Raise energy via gate bias $V_g(t)$

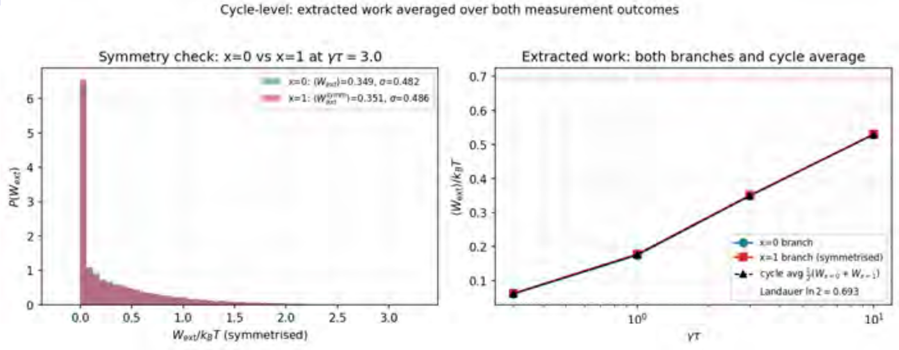
What is stochastic thermodynamics?

$x = 1$ dot is charged, quench to $-eV$.

Raise energy via gate bias $V_g(t)$



What is stochastic thermodynamics?



The per-cycle $\langle W \rangle$ curve saturates below $\ln 2\, k_B T$... the fundamental information-thermodynamic bound.

Quantum measurement theory.

Section 4.9 Quantum Measurement and Control, Wiseman & GJM, CUP (2010).

Chang et al. PRB, 96, 195440 (2017).

$$\dot{\rho} = \Gamma_{in}\mathcal{D}[c^\dagger]\rho + \Gamma_{out}\mathcal{D}[c]\rho + \underbrace{\chi\mathcal{D}[c^\dagger c]\rho}_{\text{measurement dephasing}} \equiv \mathcal{L}\rho$$

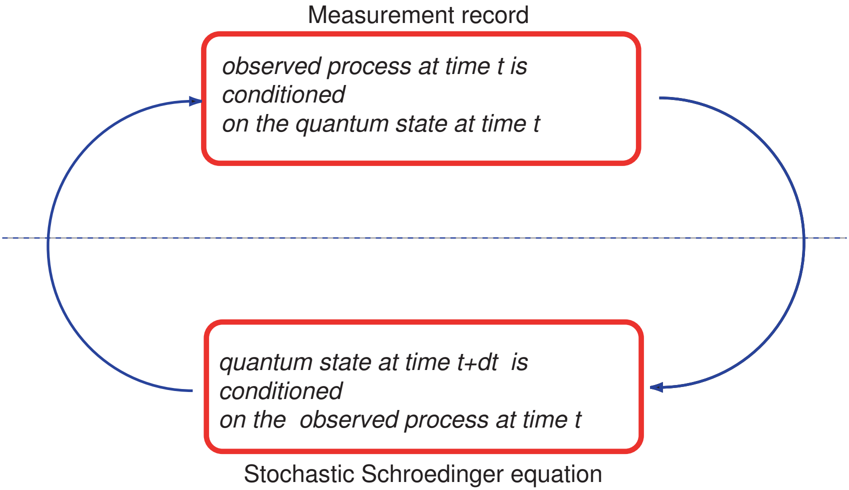
Conditional stochastic m.e.

$$d\rho_c = \mathcal{L}\rho_c dt + \sqrt{\kappa}dW\mathcal{H}[c^\dagger c]\rho_c$$

dW Wiener increment $\{c, c^\dagger\} = 1$

$$\begin{aligned} \mathcal{D}[A]\rho &= A\rho A^\dagger - \frac{1}{2}(A^\dagger A\rho + \rho A^\dagger A) \\ \mathcal{H}[A]\rho &= A\rho + \rho A^\dagger - \mathrm{tr}[(A\rho + \rho A^\dagger)\rho]\rho \end{aligned}$$

Quantum measurement theory.



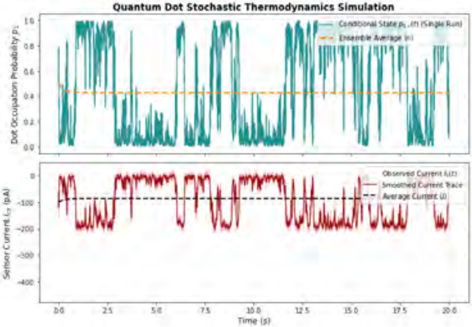
Quantum measurement theory.

Observed current: $I_c(t) = I_0 - \chi\langle\hat{n}\rangle_c - \frac{1}{\sqrt{\kappa}}\xi(t)$, $\mathbb{E}[\xi(t)\xi(t')] = \delta(t-t')$

Conditional probability: Kushner-Stratonovich eq.

$$dp_{1,c} = \Gamma_{in}(1-p_{1,c})dt + \Gamma_{out}p_{1,c}dt + 2\sqrt{\kappa}p_{1,c}(1-p_{1,c})dW$$

Partially observed Markov process.



Probabilities as Computation.

AI and quantum computing share two important features:

Probabilities as Computation.

AI and quantum computing share two important features:

– algorithms for transforming probability distributions over data
– require specialised hardware to run.

Probabilities as Computation.

AI and quantum computing share two important features:

– algorithms for transforming probability distributions over data
– require specialised hardware to run.

This suggests a common framework.

Probabilities as Computation

Machine learning:

learn structure and uncertainty by modifying or mapping simpler, initial distributions to match complex statistical patterns found in real-world datasets.

Probabilities as Computation

Machine learning:

learn structure and uncertainty by modifying or mapping simpler, initial distributions to match complex statistical patterns found in real-world datasets.

$\rightarrow$ Compress information

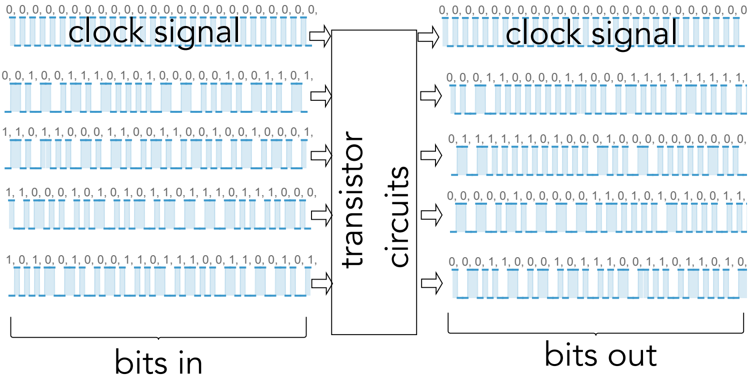
Probabilities as Computation

Machine learning:

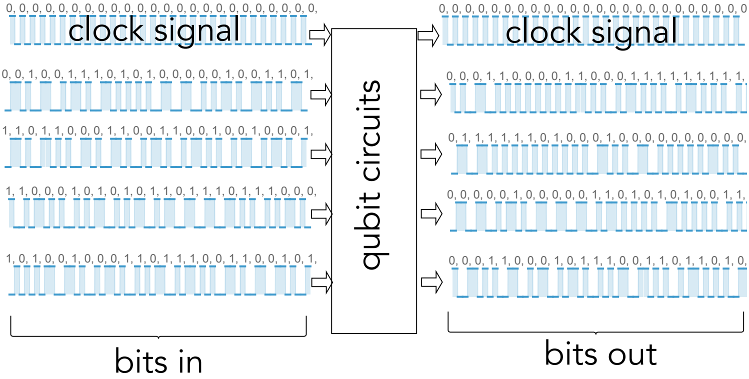
learn structure and uncertainty by modifying or mapping simpler, initial distributions to match complex statistical patterns found in real-world datasets.

$\rightarrow$ Compress information $\rightarrow$Reduce entropy.

Universal classical probability sampling.



Universal quantum probability sampling.



Universal quantum probability sampling.

Quantum computation.

- A quantum computer transforms probabilities of bit strings.
- ... by transforming of probability *amplitudes*.
- Computation = controlled interference of probability amplitudes.
- Algorithms exploit this interference to achieve greater efficiency.

Algorithmic neural networks.

1963, Rosenblatt: perceptron algorithm.

Algorithmic neural networks.

1963, Rosenblatt: perceptron algorithm.

Uses an activation function:

$$f_\lambda(z) = \frac{1}{1 + e^{-\lambda z}}$$

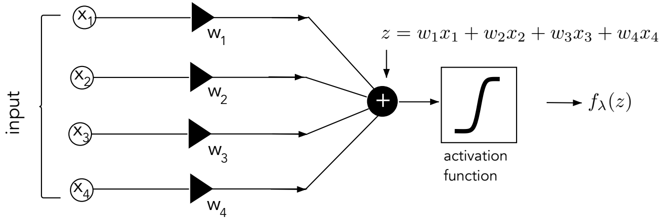
Algorithmic neural networks.

1963, Rosenblatt: perceptron algorithm.

Uses an activation function:

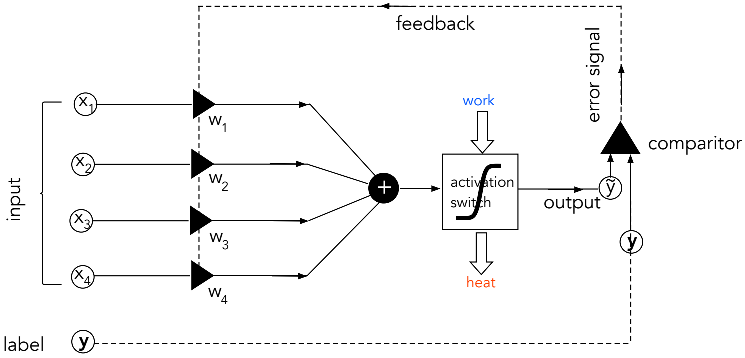
$$f_\lambda(z) = \frac{1}{1 + e^{-\lambda z}}$$

Algorithm flowchart



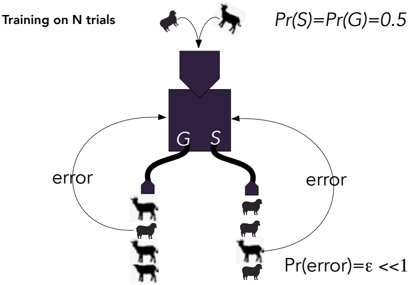
Physical learning machines.

Perceptron.



The thermodynamics of machine learning.

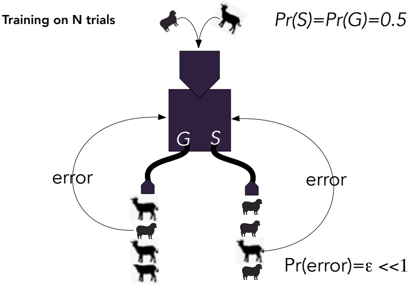
Parable of the sheep and goats.



Entropy before training: $Nk_B T \ln 2$,

The thermodynamics of machine learning.

Parable of the sheep and goats.

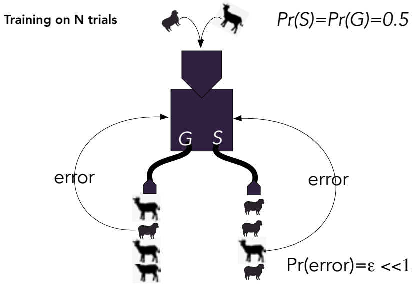


Entropy before training: $Nk_BT\ln 2$,

Entropy after training: $N\epsilon << Nk_BT\ln 2$,

The thermodynamics of machine learning.

Parable of the sheep and goats.



Entropy before training: $Nk_B T \ln 2$,

Entropy after training: $N\epsilon << Nk_B T \ln 2$,

Heat and entropy must be dissipated during training.

Universal physical emulation.

Classical physical perceptron is a dissipative switch subject to **thermal** noise

Universal physical emulation.

Classical physical perceptron is a dissipative switch subject to **thermal** noise

Too much noise, learning rate goes to zero.
Too little noise, learning rate goes to zero.

Universal physical emulation.

Quantum perceptron is a dissipative switch subject to **quantum** noise

Universal physical emulation.

Quantum perceptron is a dissipative switch subject to **quantum** noise

Too much noise, learning rate goes to zero.
Too little noise, learning rate goes to zero.

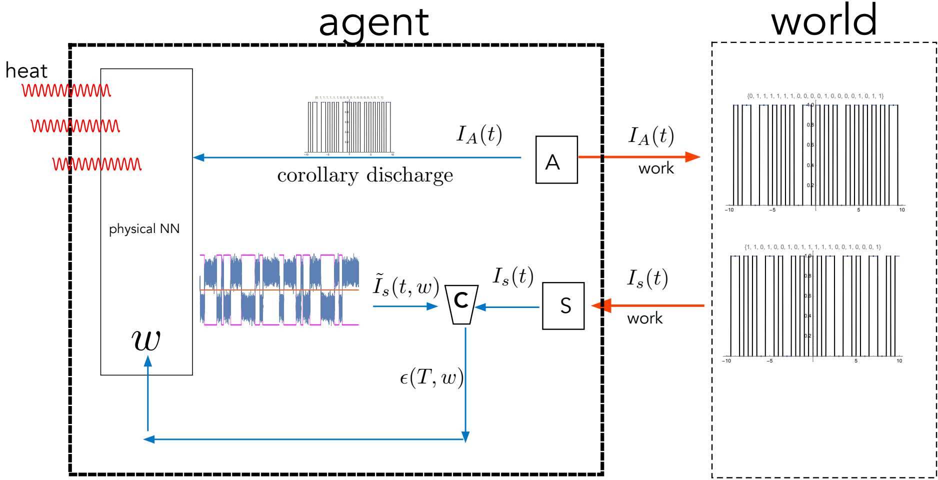
Universal physical emulation.

Quantum perceptron is a dissipative switch subject to **quantum** noise

Too much noise, learning rate goes to zero.
Too little noise, learning rate goes to zero.

Quantum noise ?
...tunnelling, spontaneous emission and
requires entanglement between a local system and its environment.

Learning Boolean functions in continuous time.



Continuous time NOT.

The output is a stochastic signal $\epsilon(T)$ as is the error function

$$\epsilon(T,w) = \underbrace{\frac{1}{T}\int_{-T/2}^{T/2} dt(I_S(t') - \tilde{I}_S(t',w)^2}_{\text{average energy dissipated.}}$$

T is the epoch time.

Continuous time NOT.

The output is a stochastic signal $\epsilon(T)$ as is the error function

$$\epsilon(T,w) = \underbrace{\frac{1}{T}\int_{-T/2}^{T/2} dt(I_S(t') - \tilde{I}_S(t',w)^2}_{\text{average energy dissipated.}}$$

T is the epoch time.
Normalise signal power

$$\epsilon(T,w) = \left[1 - \frac{2}{T}\int_{-T/2}^{T/2} dt' I_S(t')\tilde{I}_S(t',w)\right]$$

Continuous time NOT.

$$\frac{\partial \epsilon(T,w)}{\partial w} = -\frac{2}{T}\int_{-T/2}^{T/2} dt' I_S(t') \frac{\partial \tilde{I}_S(t',w)}{\partial w}$$

Continuous time NOT.

$$\frac{\partial \epsilon(T,w)}{\partial w} = -\frac{2}{T}\int_{-T/2}^{T/2} dt' I_S(t') \frac{\partial \tilde{I}_S(t',w)}{\partial w}$$

Multi layer perceptron

$$\tilde{I}_S(t',w) = \frac{1}{2}\left[1+\tanh(\beta w I_A(t)\right]$$

where β is a positive real parameter.

Continuous time NOT.

If the signal varies on a time scale γ^{-1} such that $\gamma T >> 1$

$$\frac{\partial \epsilon(T,w)}{\partial w} \approx \eta\beta \times \underbrace{I_S(t)}_{\text{labels}} \times \underbrace{I_A(t)}_{\text{data}} \times \mathrm{sech}^2(\beta w \tilde{I}_S(t))$$

where $\eta^{-1} = \gamma T$.

Continuous time NOT.

In the absence of learning/feedback, weights satisfy an Ornstein-Uhlenbeck process with uniform decay plus white noise.

$$d\vec{w}(t) = -\gamma_w \vec{w}(t)dt + \sqrt{D}d\vec{W}(t),$$

$D \propto k_B T$

Continuous time NOT.

In the absence of learning/feedback, weights satisfy an Ornstein-Uhlenbeck process with uniform decay plus white noise.

$$d\vec{w}(t) = -\gamma_w \vec{w}(t)dt + \sqrt{D}d\vec{W}(t),$$

$D \propto k_B T$

update rule

$$dw = 2LI_A(t)I_S(t)\mathrm{sech}^2(\beta w\tilde{I}_S(t))dt - \gamma_w w dt + \sqrt{D}dW,$$

where $L = \mu\eta\beta/2$ is the learning rate.

Continuous time NOT.

In the absence of learning/feedback, weights satisfy an Ornstein-Uhlenbeck process with uniform decay plus white noise.

$$d\vec{w}(t) = -\gamma_w \vec{w}(t)dt + \sqrt{D}d\vec{W}(t),$$

$D \propto k_B T$

update rule

$$dw = 2LI_A(t)I_S(t)\mathrm{sech}^2(\beta w\tilde{I}_S(t))dt - \gamma_w w dt + \sqrt{D}dW,$$

where $L = \mu\eta\beta/2$ is the learning rate. NOT: $I_S(t) = -I_A(t)$,

$$dw = -2LI_A(t)^2\mathrm{sech}^2(\beta w\tilde{I}_S(t,w))dt - \gamma_w w dt + \sqrt{D}dW$$

$$\tilde{I}_S(t,w) = \frac{1}{2}(1 + \tanh(\beta w I_A(t)))$$

Continuous time NOT.

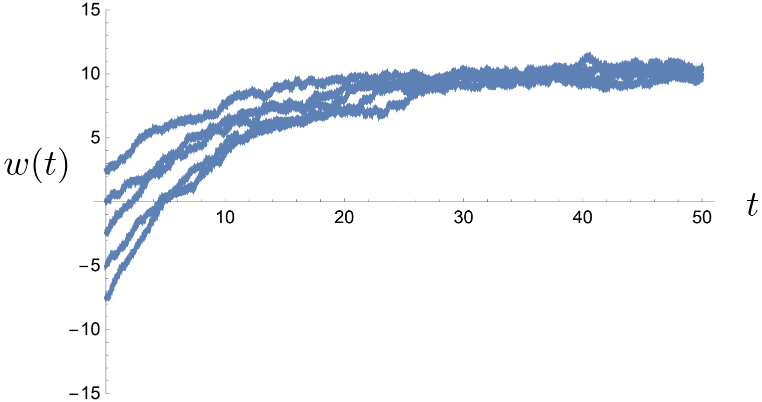
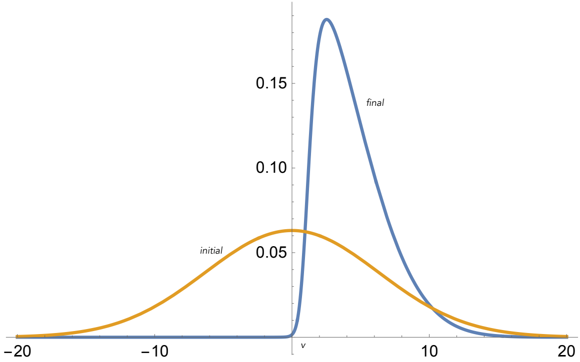


Figure: Stochastic evolution of weights starting from different initial conditions. Parameters are $L = 1.0, \beta = 0.01, \gamma_w = 0.1, D = 0.1$

Example: learning NOT

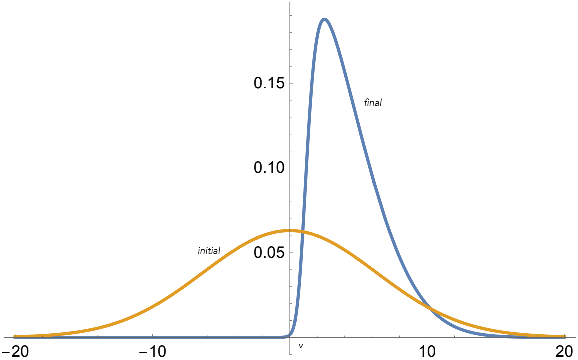
Initial and final distribution of weight:



$v = \beta w/2.$

Example: learning NOT

Initial and final distribution of weight:



$v = \beta w/2.$

learning = cooling in weight space... entropy reduction

Stochastic Thermodynamics of a perceptron.

Thermodynamics: for very low temperature, average energy dissipated per epoch is determined by change in average error per epoch

$$
\begin{aligned}
\Delta\overline{E} &\sim k_B T \Delta\bar{\epsilon} \qquad \text{classical} \\
\Delta\overline{E} &\sim \hbar\mu\Delta\bar{\epsilon} \qquad \text{quantum}
\end{aligned}
$$

μ dissipative quantum tunnelling rate.

Stochastic Thermodynamics of a perceptron.

Thermodynamics: for very low temperature, average energy dissipated per epoch is determined by change in average error per epoch

$$\begin{aligned}
\Delta\overline{E} &\sim k_B T \Delta\bar{\epsilon} \qquad \text{classical} \\
\Delta\overline{E} &\sim \hbar\mu\Delta\bar{\epsilon} \qquad \text{quantum}
\end{aligned}$$

μ dissipative quantum tunnelling rate.

A quantum Landauer's principle.

Stochastic Thermodynamics of a perceptron.

Thermodynamics: for very low temperature, average energy dissipated per epoch is determined by change in average error per epoch

$$\begin{aligned} \Delta\overline{E} &\sim k_B T \Delta\bar{\epsilon} \qquad \text{classical} \\ \Delta\overline{E} &\sim \hbar\mu\Delta\bar{\epsilon} \qquad \text{quantum} \end{aligned}$$

μ dissipative quantum tunnelling rate.

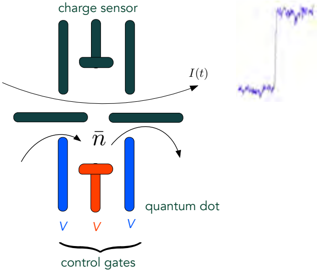
A quantum Landauer's principle.

$$\mu << k_B T$$

Quantum learning machines are thermodynamically more efficient than classical learning machines.

Quantum perceptrons.

Example one: quantum dot tunnelling.

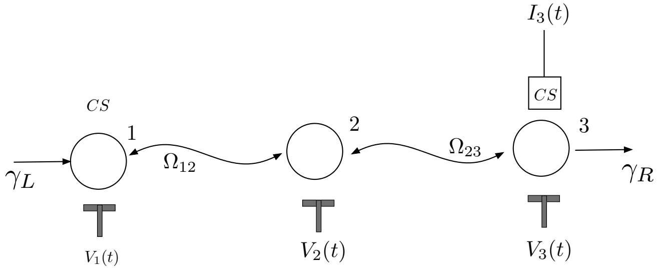


Ares and Fedele, Oxford.

milli-Kelvin, voltage controlled tunnel barriers.

Triple dot

Three dot quantum tunnelling.



$$
\begin{aligned}
H &= e_1 c_1^\dagger c_1 + e_2 c_2^\dagger c_2 + e_3 c_3^\dagger c_3 + g_{12} c_1^\dagger c_1 c_2^\dagger c_2 + g_{13} c_1^\dagger c_1 c_3^\dagger c_3 + g_{23} c_2^\dagger c_2 c_3^\dagger c_3 \\
&\quad + \Omega_{12}(c_1 c_2^\dagger + c_2 c_1^\dagger) + \Omega_{23}(c_2 c_3^\dagger + c_3 c_2^\dagger)
\end{aligned}
$$

where $\{c_i, c_j^\dagger\} = \delta_{ij}$

Triple dot

$$\frac{d\rho}{dt} = -i[H,\rho] + \gamma_L \mathcal{D}[c_1^\dagger]\rho + \gamma_R \mathcal{D}[c_3]\rho$$
$$+\Gamma\mathcal{D}[c_1^\dagger c_1]\rho + \Gamma\mathcal{D}[c_2^\dagger c_2]\rho + \Gamma\mathcal{D}[c_3^\dagger c_3]\rho$$

Measurement induced dephasing decoheres coherent tunnelling.

$$\begin{aligned}
\dot{P}_{000} &= -\gamma_L P_{000} + \gamma_R P_{001} \\
\dot{P}_{100} &= +\gamma_L P_{000} - \gamma_{12}^{(0)} P_{100} + \gamma_{12}^{(0)} P_{010} + \gamma_R P_{101} \\
\dot{P}_{010} &= +\gamma_{12}^{(0)} P_{100} - \left(\gamma_{12}^{(0)} + \gamma_{23}^{(0)} + \gamma_L\right) P_{010} + \gamma_{23}^{(0)} P_{001} + \gamma_R P_{011} \\
\dot{P}_{001} &= +\gamma_{23}^{(0)} P_{010} - \left(\gamma_{23}^{(0)} + \gamma_L + \gamma_R\right) P_{001} \\
\dot{P}_{110} &= +\gamma_L P_{010} - \gamma_{23}^{(1)} P_{110} + \gamma_{23}^{(1)} P_{101} + \gamma_R P_{111} \\
\dot{P}_{101} &= +\gamma_L P_{001} + \gamma_{23}^{(1)} P_{110} - \left(\gamma_{12}^{(1)} + \gamma_{23}^{(1)} + \gamma_R\right) P_{101} + \gamma_{12}^{(1)} P_{011} \\
\dot{P}_{011} &= +\gamma_{12}^{(1)} P_{101} - \left(\gamma_{12}^{(1)} + \gamma_L + \gamma_R\right) P_{011} \\
\dot{P}_{111} &= +\gamma_L P_{011} - \gamma_R P_{111}
\end{aligned}$$

Triple dot

For example:

$$\gamma_{12}^{(n_3)} = \frac{2\Omega_{12}^2\,\Gamma}{\Gamma^2 + \left[(e_2 - e_1) + (g_{23} - g_{13})n_3\right]^2}.$$

Triple dot

The stochastic Master Equation (SME) is ‘quantum ground truth’.

$$d\rho = -i[H,\rho]\,dt + \sum_k \gamma_k \mathcal{D}[L_k]\rho\,dt + \Gamma\mathcal{D}[n_3]\rho\,dt + \sqrt{2\Gamma}\mathcal{H}[n_3]\rho\,dW$$

with $\mathcal{H}[n_3]\rho = n_3\rho + \rho n_3 - 2\langle n_3\rangle\rho$

readout

$$dY_t = \sqrt{\Gamma}\langle n_3\rangle_{\rho_c} dt + dW_t$$

Triple dot

Decohered limit is the Kushner-Stratnovich SDE:

$$dP_s(t) \;=\; \mathcal{L}[P]_s\, dt \;+\; \sqrt{\Gamma}\,\big(n_3^{(s)} - \bar{n}_3(t)\big)\, P_s(t)\, dW$$

This is the Kushner–Stratonovich (nonlinear-filter) equation specialized to a jump-process signal X_t and a diffusive readout of n_3.

Triple dot

Decohered limit is the Kushner-Stratnovich SDE:

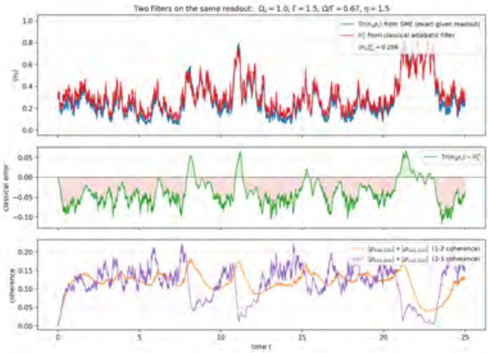
$$dP_s(t) \;=\; \mathcal{L}[P]_s \, dt \;+\; \sqrt{\Gamma}\,\big(n_3^{(s)} - \bar{n}_3(t)\big)\, P_s(t)\, dW$$

This is the Kushner–Stratonovich (nonlinear-filter) equation specialized to a jump-process signal X_t and a diffusive readout of n_3.

The quantum SME 8×8 matrix SDE. The classical KS is only 8-vector SDE

Triple dot

Comparison:



For $\Omega/\Gamma \lesssim 0.5$ the eightfold reduction in representational cost is nearly free — the two filters agree at the percent level.

Triple dot

Two-bit gate:
The input signal is $(s_1, s_2), s_i \in \{-1, 1\}$ applied as bias voltages to change the on-site energy of the dots 1 and 2.

Triple dot

Two-bit gate:
The input signal is $(s_1, s_2), s_i \in \{-1, 1\}$ applied as bias voltages to change the on-site energy of the dots 1 and 2.

The output is charge sensor current on all sites. How does the output respond to the four different input vectors?

Triple dot

Add broadband microwave noise between the dots $\sim$ like adding parallel resistances to a circuit .

Triple dot

Add broadband microwave noise between the dots $\sim$ like adding parallel resistances to a circuit .

Site detuning $\varepsilon_i = (0, 0, 2)$, $\Delta_{12} = 0, \Delta_{23} = 2$

Triple dot

Add broadband microwave noise between the dots $\sim$ like adding parallel resistances to a circuit .

Site detuning $\varepsilon_i = (0, 0, 2)$, $\Delta_{12} = 0, \Delta_{23} = 2$

weak coherent tunnelling $(\Omega = 0.3)$

$$\gamma^{\uparrow}_{i,j}(T) = \gamma_0 \, n_{BE}(\Delta_{ij}/T), \qquad \gamma^{\downarrow}_{j,i}(T) = \gamma_0 \left[n_{BE}(\Delta_{ij}/T) + 1\right],$$

with $n_{BE}(x) = 1/(e^x - 1)$ thermal activation is the dominant transport mechanism and the threshold behaviour is prominent.

Triple dot

Set bias binary signals: $\varepsilon = (0, 0, 2) \rightarrow (0, bs_1, 2 + bs_2)$

Triple dot

Set bias binary signals: $\varepsilon = (0,0,2) \rightarrow (0, bs_1, 2 + bs_2)$

Can the measurement currents, and their correlations, distinguish all four inputs. (two bits) ?

Triple dot

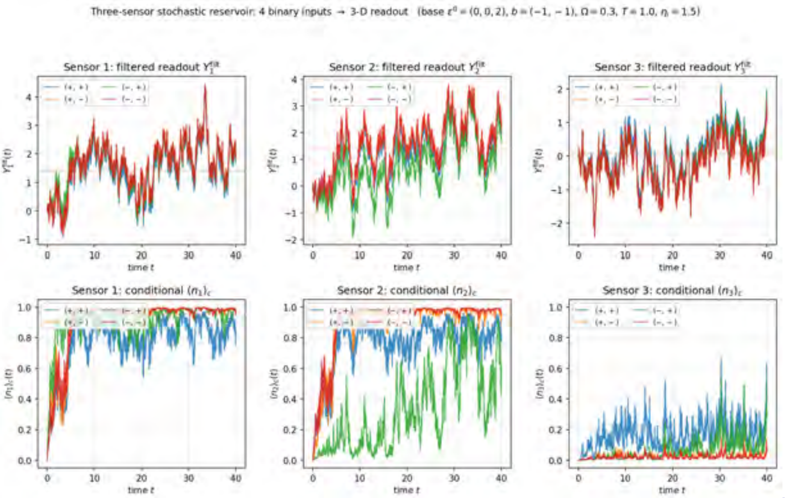
Set bias binary signals: $\varepsilon = (0, 0, 2) \rightarrow (0, bs_1, 2 + bs_2)$

Can the measurement currents, and their correlations, distinguish all four inputs. (two bits) ?

Steady state feature vector

$$f(s_1, s_2) \rightarrow (\langle n_1 \rangle, \langle n_2 \rangle, \langle n_3 \rangle)$$

Triple dot



Three-sensor stochastic reservoir: 4 binary inputs → 3-D readout (base $\epsilon^0 = (0, 0, 2)$, $b = (-1, -1)$, $\Omega = 0.3$, $T = 1.0$, $\eta_i = 1.5$)
Sensor 1: filtered readout $Y_1^{\rm filt}$
Sensor 2: filtered readout $Y_2^{\rm filt}$
Sensor 3: filtered readout $Y_3^{\rm filt}$
Sensor 1: conditional $\langle n_1 \rangle_c$
Sensor 2: conditional $\langle n_2 \rangle_c$
Sensor 3: conditional $\langle n_3 \rangle_c$
(+, +)
(+, −)
(−, +)
(−, −)
time t

Triple dot

Steady-state feature vectors $\mathbf{f}(s_1, s_2) = (\langle n_1\rangle, \langle n_2\rangle, \langle n_3\rangle)_{ss}$ at $T = 1$:

input	$\langle n_1\rangle$	$\langle n_2\rangle$	$\langle n_3\rangle$
$(+,+)$	0.775	0.762	0.225
$(+,-)$	0.951	0.949	0.050
$(-,+)$	0.929	0.288	0.071
$(-,-)$	0.959	0.956	0.041

Pair-wise Euclidean distances in the 3-D feature space:

pair	distance
$(+,+)$ vs $(+,-)$	0.311
$(+,+)$ vs $(-,+)$	0.522
$(+,+)$ vs $(-,-)$	0.325
$(+,-)$ vs $(-,+)$	0.662
$(+,-)$ vs $(-,-)$	**0.014**
$(-,+)$ vs $(-,-)$	0.670

Triple dot

Gaussian channel analysis:

$$dY_i = m_i(S)\,dt + dW_i, \quad m_i(s) = \eta_i \langle n_i \rangle_{\mathrm{ss}}^{(s)}$$

Triple dot

Gaussian channel analysis:

$$dY_i = m_i(S)\,dt + dW_i, \quad m_i(s) = \eta_i \langle n_i \rangle_{\mathrm{ss}}^{(s)}$$

Integrate over a window τ to form estimator $\bar{Y}_i^\tau$.

Triple dot

Gaussian channel analysis:

$$dY_i = m_i(S)\,dt + dW_i, \quad m_i(s) = \eta_i \langle n_i \rangle_{\mathrm{ss}}^{(s)}$$

Integrate over a window τ to form estimator $\bar{Y}_i^{\tau}$.

Gaussian with mean $m_i(s)$ and variance $1/\tau$.

signal-to-noise ratio

$$\mathrm{SNR}_i = \eta_i^2\,\mathrm{Var}_S[\langle n_i \rangle^{(s)}] \cdot \tau$$

Triple dot

Approximate as a Gaussian channel:
Per-sensor Gaussian-channel info:

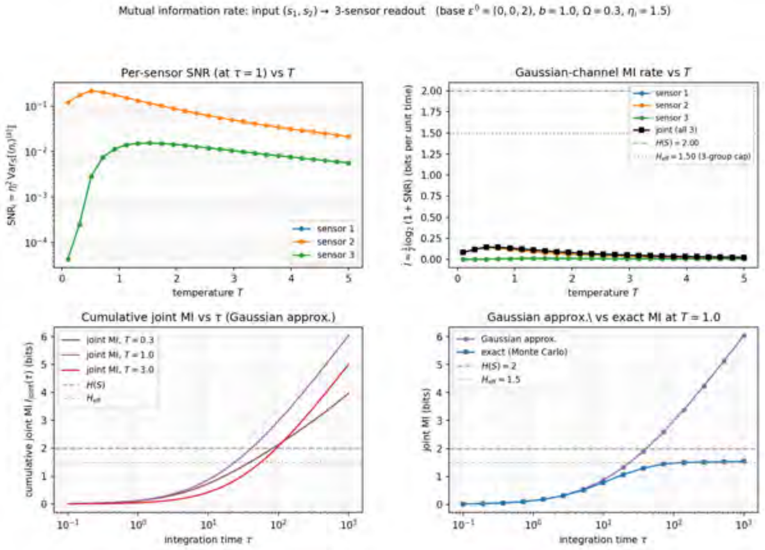
$$I_i(\tau) = \frac{1}{2}\log_2(1 + \mathrm{SNR}_i)$$

Joint 3-sensor (independent noise, correlated signals):

$$I_{\mathrm{joint}}(\tau) = \frac{1}{2}\log_2 \det(I_3 + \tau\,\eta\Sigma_n\eta)$$

Σ_n covariance matrix of the SS occupation feature vector over the 4 equiprobable inputs.

Triple dot



Mutual information rate: input $(s_1, s_2) \to$ 3-sensor readout (base $\varepsilon^0 = (0,0,2)$, $b = 1.0$, $\Omega = 0.3$, $\eta_i = 1.5$)
Per-sensor SNR (at $\tau = 1$) vs T
sensor 1
sensor 2
sensor 3
temperature T
Gaussian-channel MI rate vs T
joint (all 3)
$H(S) = 2.00$
$H_{\rm eff} = 1.50$ (3-group cap)
Cumulative joint MI vs τ (Gaussian approx.)
joint MI, $T = 0.3$
joint MI, $T = 1.0$
joint MI, $T = 3.0$
$H(S)$
$H_{\rm eff}$
integration time τ
cumulative joint MI $I_{\rm joint}(\tau)$ (bits)
Gaussian approx.\ vs exact MI at $T = 1.0$
Gaussian approx.
exact (Monte Carlo)
$H(S) = 2$
$H_{\rm eff} = 1.5$
joint MI (bits)

Triple dot: Entropy production.

Leads:

$$\dot{\sigma}_{\rm leads} = J\,V/T, \quad V = \mu_L - \mu_R, \quad J = \gamma_R \langle n_3 \rangle_{\rm ss}$$

Triple dot: Entropy production.

Leads:

$$\dot{\sigma}_{\rm leads} = J\,V/T, \quad V = \mu_L - \mu_R, \quad J = \gamma_R \langle n_3 \rangle_{\rm ss}$$

Measurement:

$$\dot{\sigma}_{\rm meas} = \Gamma \sum_i \langle n_i \rangle (1 - \langle n_i \rangle)$$

Triple dot: Entropy production.

Leads:

$$\dot{\sigma}_{\rm leads} = J\,V/T, \quad V = \mu_L - \mu_R, \quad J = \gamma_R \langle n_3 \rangle_{\rm ss}$$

Measurement:

$$\dot{\sigma}_{\rm meas} = \Gamma \sum_i \langle n_i \rangle (1 - \langle n_i \rangle)$$

Quantum:

small residual $\dot{\sigma}_{\rm coh} \propto \Omega^2/\Gamma_{\rm deph}$ from the fact that coherent tunneling with dephasing produces classical current in a partly-off-resonance chain.

Triple dot: Landauer.

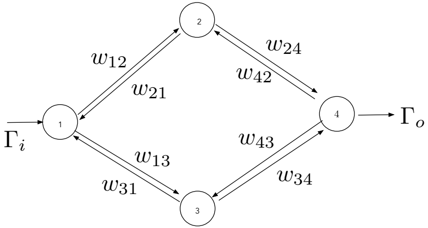
Landauer-like efficiency: $\eta_{\text{info}} = \dot{I}\ \ln 2/\dot{\sigma}$ (bits of input information per bit of dissipated entropy, dimensionless, ≤ 1.

$V_{bias} = 5$

T	$\bar{J}$	$\dot{\sigma}_{\text{leads}}$	$\dot{\sigma}_{\text{meas}}$	$\dot{\sigma}_{\text{total}}$	$\dot{I}$	η_{info}
0.3	0.022	0.358	0.263	0.622 nats/t	0.117 bits/t	**13.0%**
1.0	0.097	0.484	0.636	1.120 nats/t	0.129 bits/t	8.0%
3.0	0.269	0.448	1.337	1.785 nats/t	0.049 bits/t	1.9%

Peak efficiency in the scan: $T \approx 0.5$, $\eta_{\text{info}} \approx 13.8\%$.

Quad dot.



The input signal $s_{in} \in \{-1, +1\}$, can be applied to any pair of dots to change their energy. Initial $\varepsilon = (0, 1, 1, 2)$.

Optimise over all combinations find best is to bias $2, 3$:
breaks the reflection immediately — the upper and lower paths carry different currents.

Quad dot.

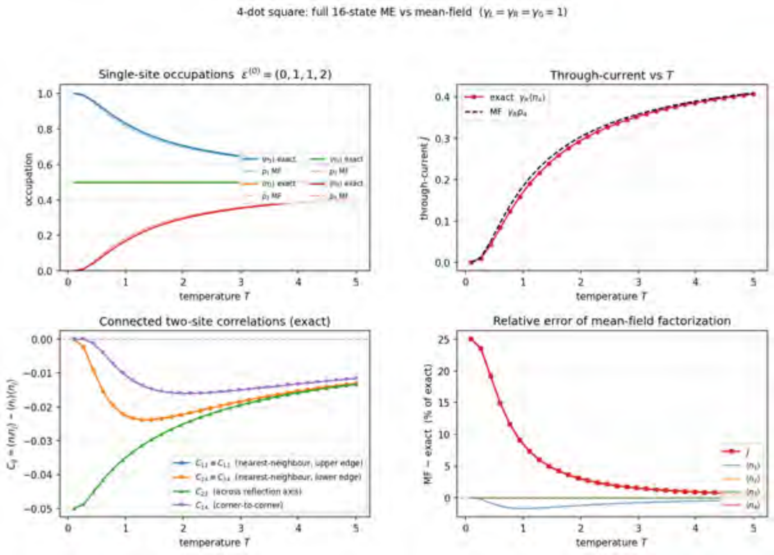
Decohered classical markov process is 16-component vector.

Quad dot.

Decohered classical markov process is 16-component vector.

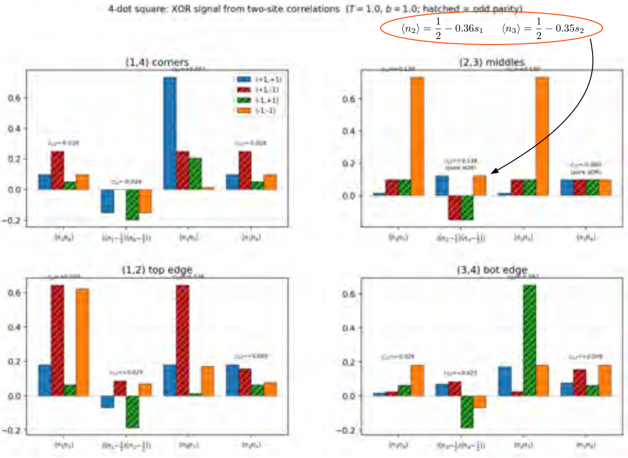
Monitor at all four currents and cross correlations.

Quad dot.



4-dot square: full 16-state ME vs mean-field ($\gamma_L = \gamma_R = \gamma_S = 1$)
Single-site occupations $\varepsilon^{(0)} = (0, 1, 1, 2)$
occupation
temperature T
Through-current vs T
exact $\gamma_R \langle n_4 \rangle$
MF $\gamma_R p_4$
through-current J
temperature T
Connected two-site correlations (exact)
$C_{ij} = \langle n_i n_j \rangle - \langle n_i \rangle \langle n_j \rangle$
$C_{12} = C_{13}$ (nearest-neighbour, upper edge)
$C_{24} = C_{34}$ (nearest-neighbour, lower edge)
C_{23} (across reflection axis)
C_{14} (corner-to-corner)
temperature T
Relative error of mean-field factorization
MF − exact (% of exact)
temperature T

Quad dot XOR.



$$\langle n_2 \rangle = \frac{1}{2} - 0.36 s_1 \qquad \langle n_3 \rangle = \frac{1}{2} - 0.35 s_2$$

Quad dot XOR.

XOR feature: realized by simultaneously measuring dots 2 and 3 and computing the cross-correlation of their occupations.

In experimental terms this is a two-sensor coincidence measurement: one QPC on dot 2, one on dot 3, both feeding into a multiplier or into a joint post-processing stage.

Quad dot: learning XOR

The biases on dots 2 and 3, taken as the learnable parameters.

Quad dot: learning XOR

The biases on dots 2 and 3, taken as the learnable parameters.

The input data pairs (s_1, s_2) are labelled for XOR.

Quad dot: learning XOR

The biases on dots 2 and 3, taken as the learnable parameters.

The input data pairs (s_1, s_2) are labelled for XOR.

Starting with random bias values on dots 2 and 3 train the system using a cost function based on minimising the Landauer cost.

Quad dot: learning XOR

Inputs $(x_1, x_2) \in \{-1, +1\}^2$ enter through corners: $\varepsilon_1 = ax_1$, $\varepsilon_4 = 2 + ax_2$.

Quad dot: learning XOR

Inputs $(x_1, x_2) \in \{-1, +1\}^2$ enter through corners: $\varepsilon_1 = ax_1$, $\varepsilon_4 = 2 + ax_2$.
Learnable weights $\mathbf{w} = (w_2, w_3)$ modify middles: $\varepsilon_2 = 1 + w_2$, $\varepsilon_3 = 1 + w_3$.

Quad dot: learning XOR

Inputs $(x_1, x_2) \in \{-1, +1\}^2$ enter through corners: $\varepsilon_1 = ax_1$, $\varepsilon_4 = 2 + ax_2$.
Learnable weights $\mathbf{w} = (w_2, w_3)$ modify middles: $\varepsilon_2 = 1 + w_2$, $\varepsilon_3 = 1 + w_3$.

Readout: centered cross-correlation $\hat{C}(\mathbf{w}) = \langle n_2 n_3 \rangle - \frac{1}{2}(\langle n_2 \rangle + \langle n_3 \rangle) + \frac{1}{4}$.
- XOR label (in ± 1 encoding): $y = x_1 x_2$ so $y = +1$ is "same" (XOR=0) and $y = -1$ is "different" (XOR=1).

Quad dot: learning XOR

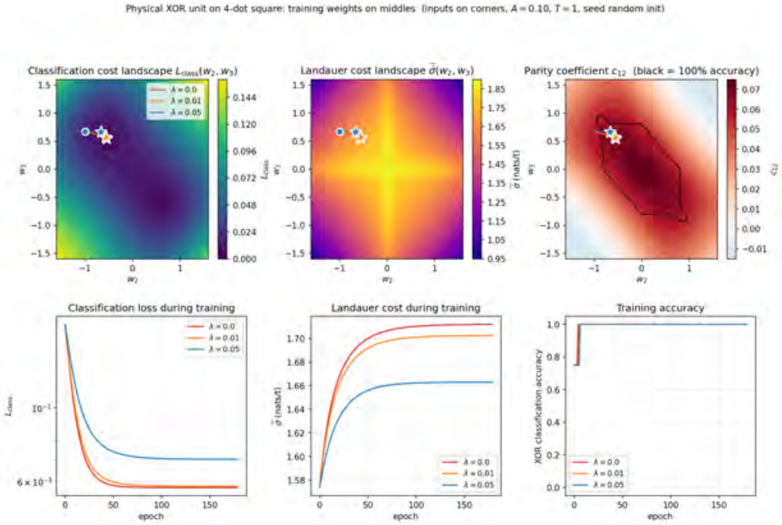
Inputs $(x_1, x_2) \in \{-1, +1\}^2$ enter through corners: $\varepsilon_1 = ax_1$, $\varepsilon_4 = 2 + ax_2$.
Learnable weights $\mathbf{w} = (w_2, w_3)$ modify middles: $\varepsilon_2 = 1 + w_2$, $\varepsilon_3 = 1 + w_3$.

Readout: centered cross-correlation $\hat{C}(\mathbf{w}) = \langle n_2 n_3 \rangle - \frac{1}{2}(\langle n_2 \rangle + \langle n_3 \rangle) + \frac{1}{4}$.
- XOR label (in ± 1 encoding): $y = x_1 x_2$ so $y = +1$ is "same" (XOR=0) and $y = -1$ is "different" (XOR=1).

Landauer-regularized MSE for XOR:

$$L(\mathbf{w}) = \underbrace{\sum_{(x_1,x_2)} \left(\hat{C}^{(x_1,x_2)}(\mathbf{w}) - A \cdot x_1 x_2\right)^2}_{L_{\text{class}}} + \lambda \underbrace{\bar{\dot{\sigma}}(\mathbf{w})}_{L_{\text{Landauer}}}$$

Quad dot: learning XOR



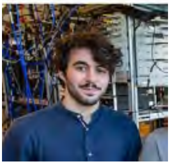
Thanks.

Sahar Basiri-Esfahani
Bristol

Matthew Woolley
UNSW

Josh Burns
U. Sussex

Natalia Ares
Oxford



Federico Fedele
Oxford

Parth Girdhar
Oxford