Boosting classical and quantum Monte Carlo simulations using generative neural networks

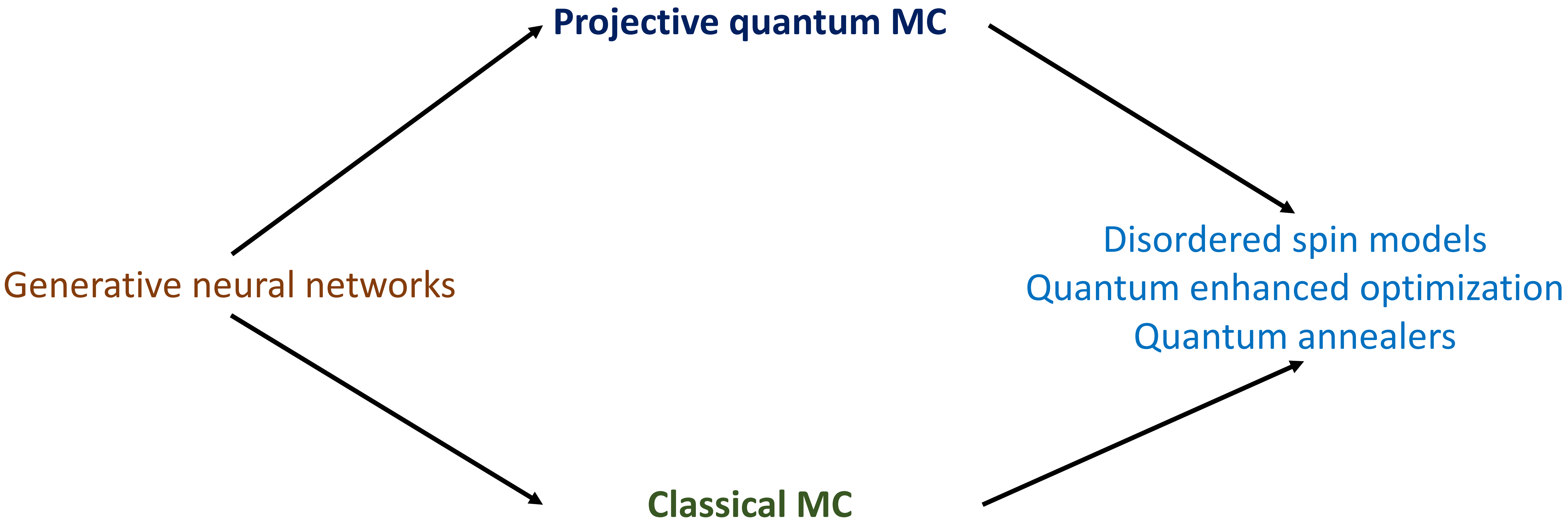
Sebastiano Pilati

University of Camerino

COLLABORATORS:

B. McNaughton (U. Camerino & U. Antwerp)
A. Perali (U. Camerino)
P. Pieri (U. Bologna)
M. Milošević (U. Antwerp)
E. M. Inack (Perimeter Institute, Canada)
G. E. Santoro, G. Giudici, T. Parolini (SISSA, Trieste)
L. Dell'Anna (Uni. Padova)

HSE (Moscow)
April 22° 2021

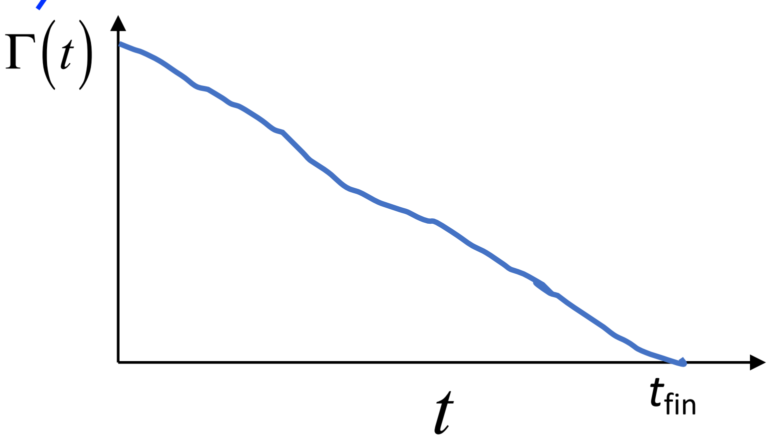

Projective quantum MC
Generative neural networks
Disordered spin models
Quantum enhanced optimization
Quantum annealers
Classical MC

Adiabatic quantum computing / quantum annealing

$$H = H_{\rm cl} + H_{\rm kin}$$

$$H_{\rm cl} = -\sum_{ij} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z$$ Classical Ising glass / QUBO

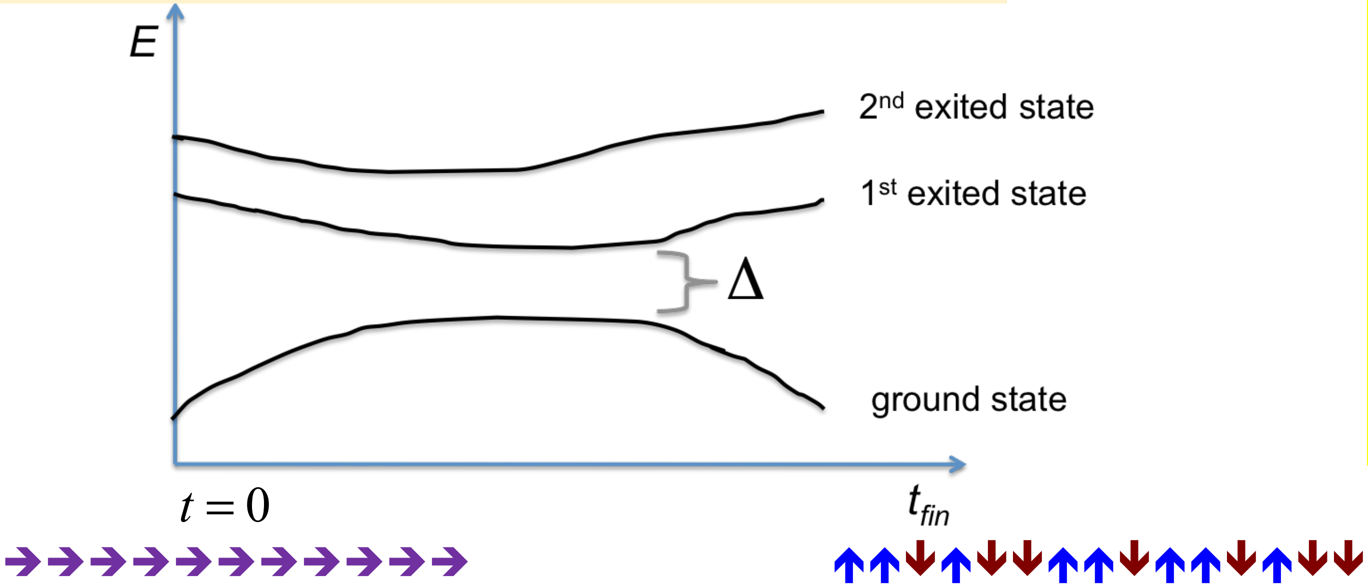
$$H_{\rm kin} = -\Gamma(t)\sum_i \sigma_i^x$$ Driver Hamiltonian



At $t=0$, $\Gamma(t=0) \gg J_{ij}$ ground-state is $|\psi(t=0)\rangle \cong |\rightarrow\rightarrow\ldots\rightarrow\rangle$

At $t=t_{\rm fin}$, $\Gamma(t=t_{\rm fin})=0$ ground-state is $|\psi(t=t_{\rm fin})\rangle \cong |\uparrow\downarrow\downarrow\downarrow\uparrow\ldots\uparrow\downarrow\rangle$

Istantaneous eigenvalues of *H(t)*



How slow?

Adiabatic theorem:

$$t_{\text{fin}} >> \frac{\alpha}{\Delta^2}$$

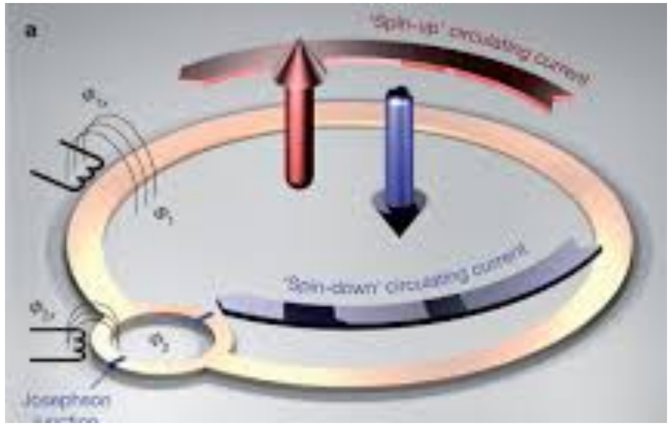
$\Delta \equiv$ smallest gap

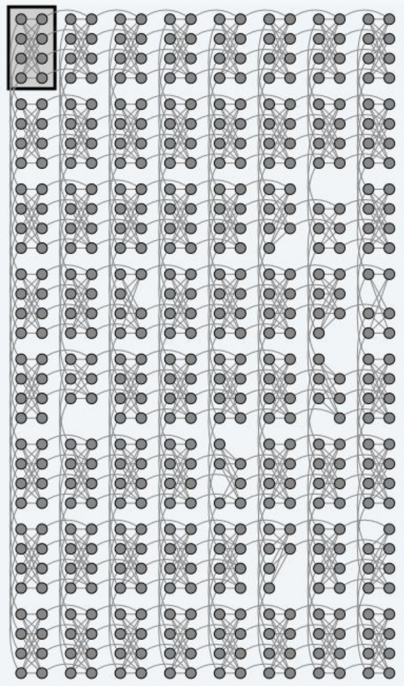
$$s = \frac{t}{t_{\text{fin}}} \in [0,1] \qquad H(s=0) = H_{\text{kin}} \qquad H(s=1) = H_{\text{cl}}$$

Condition for adiabaticity: $t_{\text{fin}} \gg \max\limits_{0 \le s \le 1} \dfrac{\left| \left\langle \psi_1(s) \right| \dfrac{\mathrm{d}H(s)}{\mathrm{d}s} \left| \psi_0(s) \right\rangle \right|}{\Delta_{1,0}(s)^2}$

D-WAVE quantum annelaersadiabatic quantum computer

$$H_{\mathrm{cl}} = -\sum_{ij} J_{ij}\sigma_i^z\sigma_j^z - \sum_i h_i\sigma_i^z \qquad H_{\mathrm{kin}} = -\Gamma(t)\sum_{ij}\sigma_i^x$$

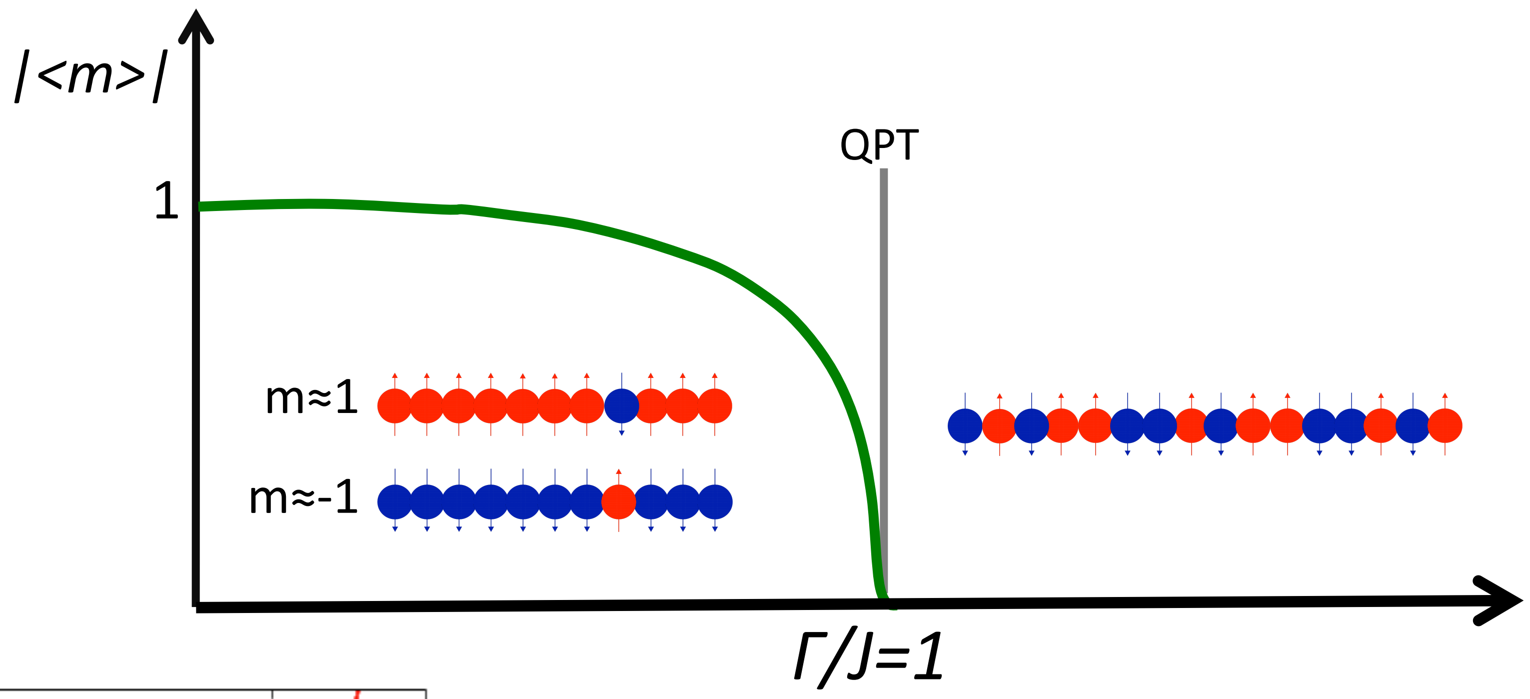


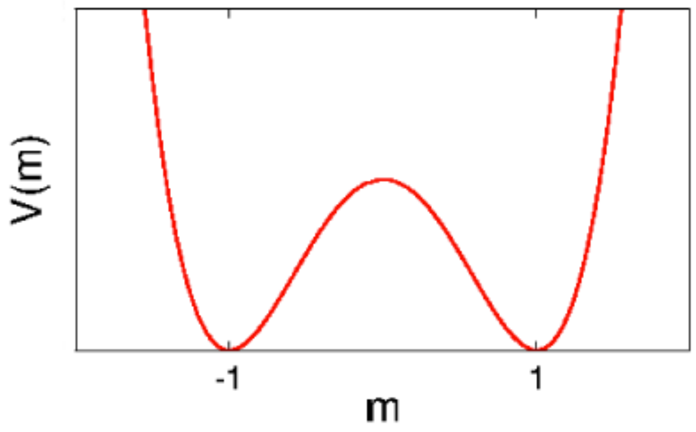


Can a quantum annealer outperform classical optimization methods?

Can quantum MC simulations provide some hints?
Notice: *H* is stoquastic → no negative sign-problem

1D ferromagnetic quantum Ising model: $H = -J\sum_i \ \sigma_i^z \sigma_{i+1}^z - \Gamma \sum_i \ \sigma_i^x$

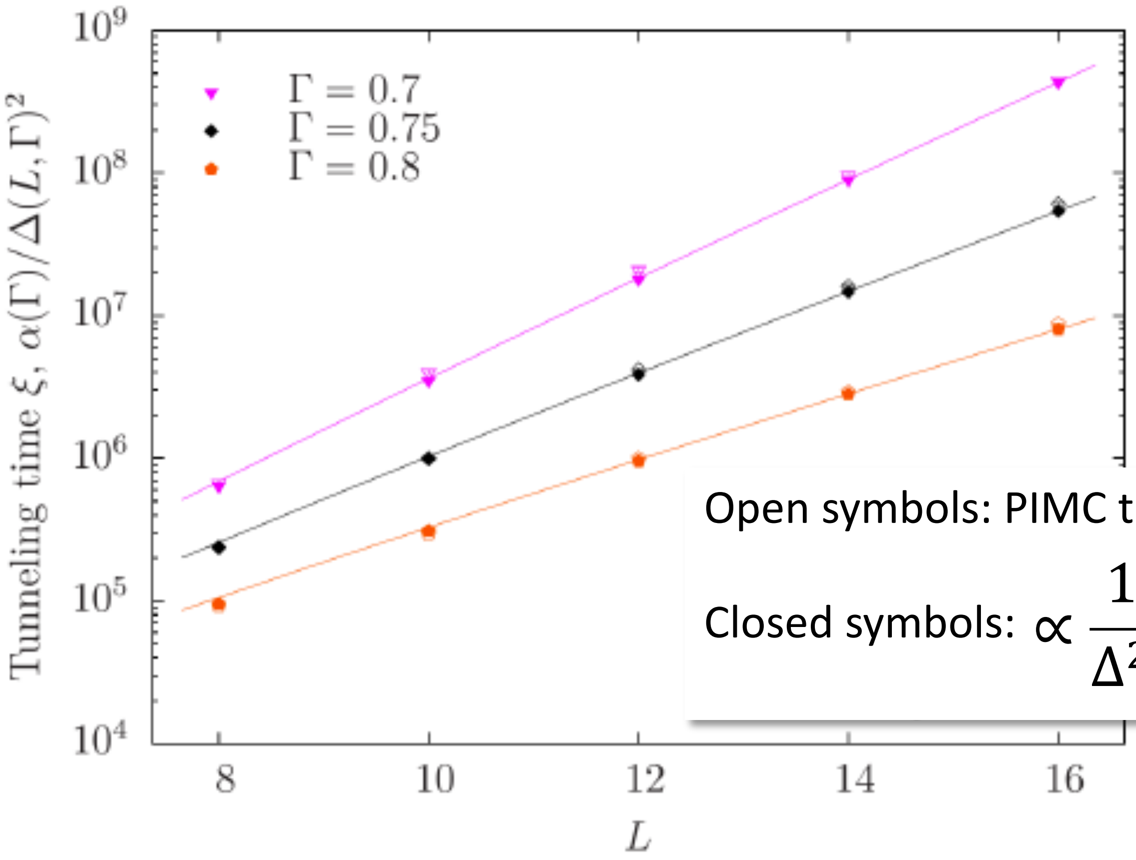




exponentially small energy gap: $\Delta \propto \exp(-\alpha L)$

QMC tunneling time: finite-temperature PIMC

Isakov, Mazzola, Smelyanskiy, Jiang, Boixo, Neven, Troyer, PRL (2016)
Mazzola, Smelyanskiy, Troyer, PRB (2017)



Open symbols: PIMC tunneling time

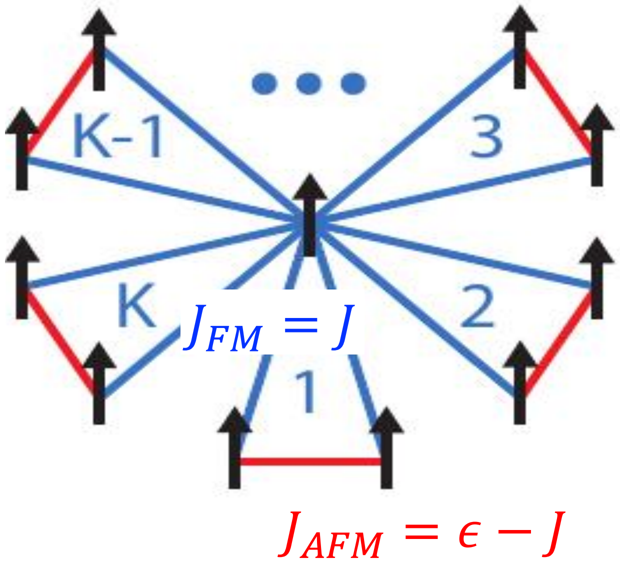
Closed symbols: $\propto \frac{1}{\Delta^2}$

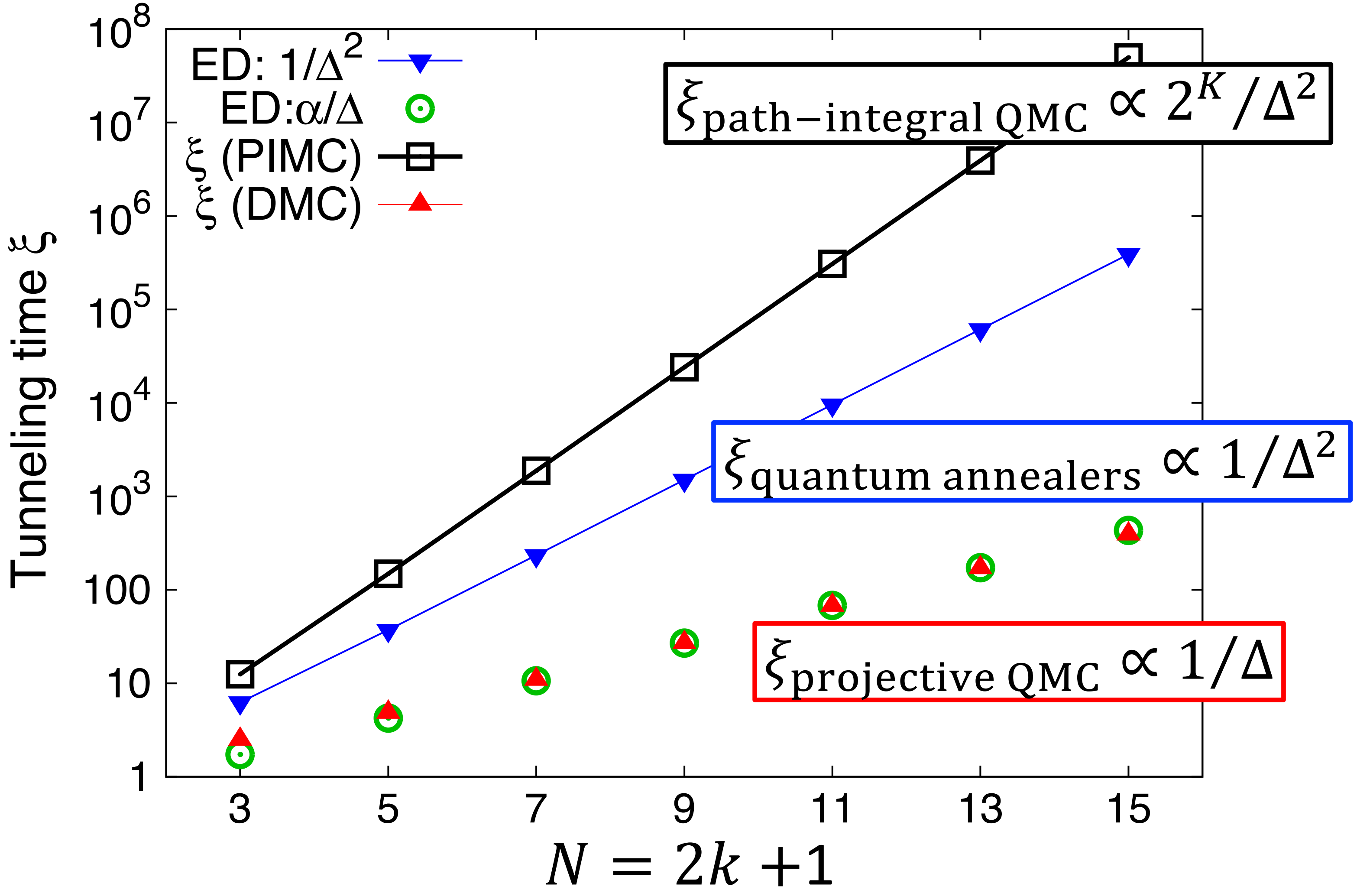
The PIMC algorithm efficiently simulates incoherent quantum tunneling. **Is this general?**

Note: PIMC with open-boundary condition in imaginary time it scales as 1/Δ

Shamrock: a model of frustrated rings

Introduced in:
E. Andriyash and M. H. Amin,
Can quantum Monte Carlo simulate quantum annealing?
arXiv:1703.09277 (2017)





- path-integral QMC dynamics slows down due to “topological” obstruction.
- projective QMC scales like $1/\Delta$

E. M. Inack, G. Giudici, T. Parolini, G.E. Santoro, SP, PRA (2018)

What is the complexity of (simple) projective QMC?

- Any diagonal(classical) Hamiltonian in stoquastic.
- Finding its ground state encompasses hard classical optimization problems such as k-SAT or MAX-CUT.

Bravyi, Quant. Inf. Comp., Vol. 15, No. 13/14, pp. 1122-1140 (2015)

Possible sources of error in projective QMC:

Population of random walkers evolves via random diffusion and killing/cloning process.

Population control bias: systematic error due to correlations among walkers cloned from the same ancestor.

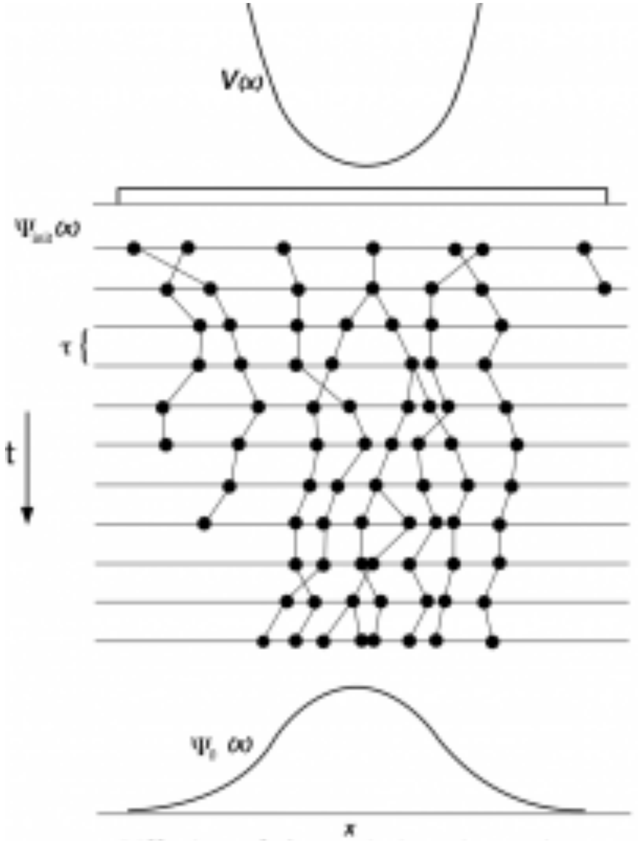


Image from: W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal Rev. Mod. Phys. 73, 33 (2001)

Projective Monte Carlo for Quantum Ising models

$$H=-\sum_{ij} J_{ij}\sigma_i^z\sigma_j^z - \Gamma\sum_i \sigma_i^x$$

$$\psi(\mathbf{S},\tau)=\exp(-\tau H)\psi(\mathbf{S},0) \underset{\tau\to\infty}{\approx} \psi_0(\mathbf{S},0)$$ Schrödinger eq. in imaginary time

$$\psi(\mathbf{S},\tau+\Delta\tau)=\sum_{\mathbf{S}'} G(\mathbf{S}',\mathbf{S},\Delta\tau)\psi(\mathbf{S}',\tau)$$ defines a Markov process $G(\mathbf{S}',\mathbf{S},\Delta\tau)=\langle\mathbf{S}'|\exp(-\Delta\tau H)|\mathbf{S}\rangle$

$G(\mathbf{S}',\mathbf{S},\Delta\tau)\geq 0 \;\Rightarrow$ no sign problem (stoquastic Hamiltonian)

$\sum_{\mathbf{S}'} G(\mathbf{S}',\mathbf{S},\Delta\tau)\neq 1 \Rightarrow$ not a standard Markov process $\Rightarrow$ kill or clone random walkers

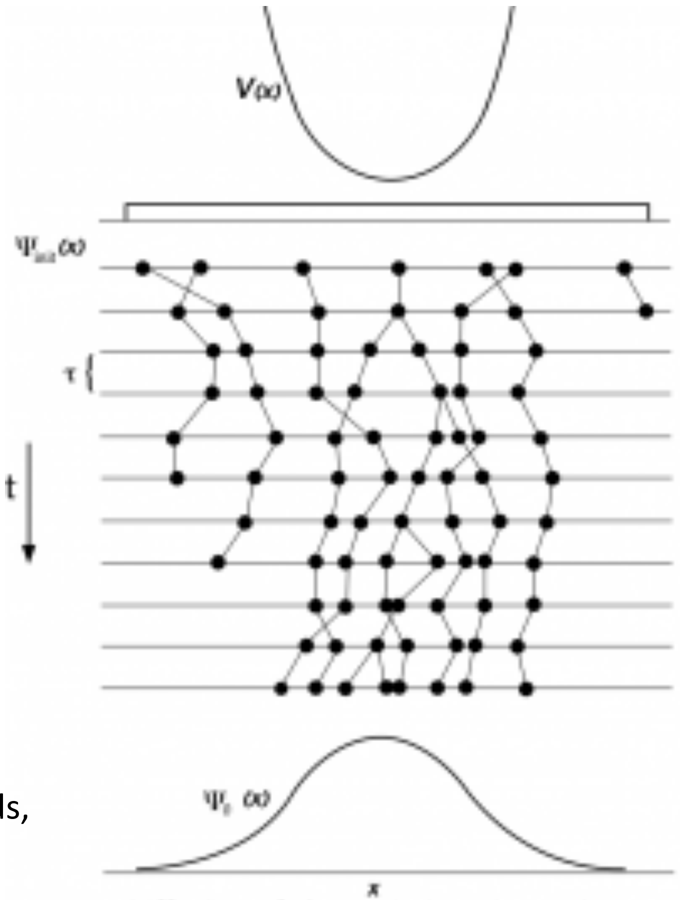
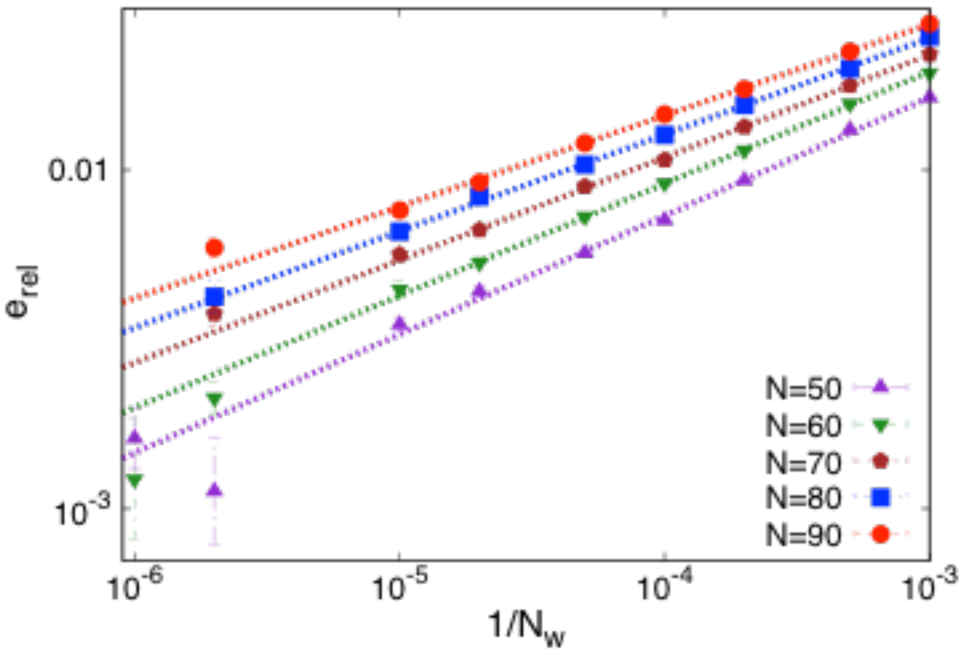


Image from: W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal Rev. Mod. Phys. 73, 33 (2001)

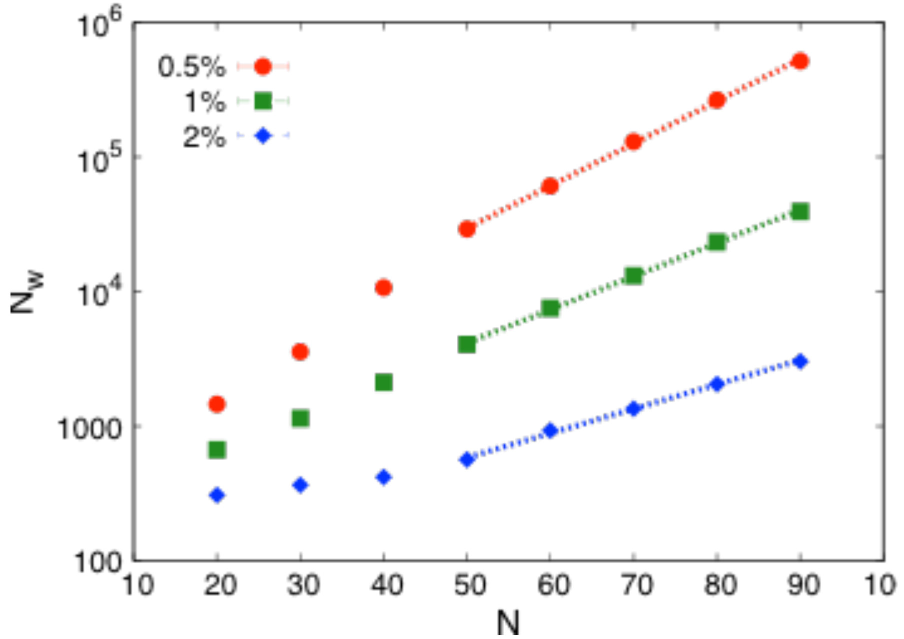
Systematic errors in PQMC algorithms: ferromagnetic Ising chain

Relative error w.r.t. exact Jordan-Wigner theory



Inverse number of random walkers

# of walkers required to keep relative err. fixed



System size

Exponentially growing computational cost

Note: here we use "simple" PQMC algorithm: no guiding wave function.

E. M. Inack, G. Giudici, T. Parolini, G.E. Santoro, SP, PRA (2018)

IMPORTANCE SAMPLING

Introduce guiding wave function $\equiv \psi_G(\mathbf{x})$

Modified master eq.: $\Psi(\mathbf{x},\tau+\Delta\tau)\psi_G(\mathbf{x}) = \sum_{\mathbf{x}'} \tilde{G}(\mathbf{x},\mathbf{x}',\Delta\tau)\Psi(\mathbf{x}',\tau)\psi_G(\mathbf{x}')$

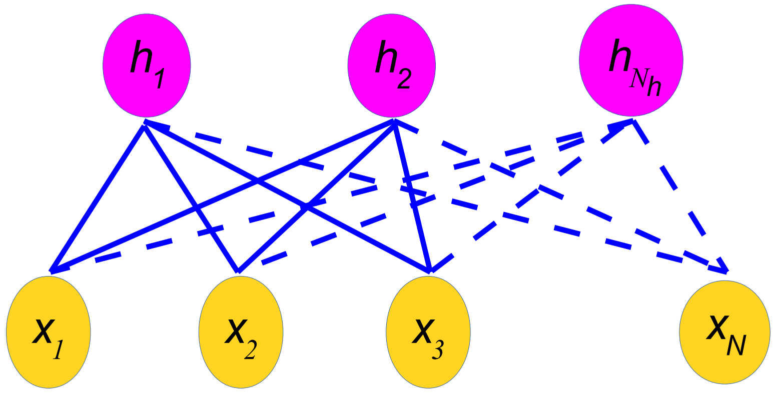
Modified Green's function: $\tilde{G}(\mathbf{x},\mathbf{x}',\Delta\tau) = \langle \mathbf{x} | \exp(-\Delta\tau \hat{H} - E_{\mathrm{REF}}) | \mathbf{x}' \rangle \frac{\psi_G(\mathbf{x})}{\psi_G(\mathbf{x}')}$

The guiding wf reduces computational cost and statistical fluctuations

Here, we adopt neural network states.

Generative neural networks: Restricted Boltzmann machines

- Correlations are introduced via hidden variables.
- Intra-layer correlations are omitted.



$$H_{\mathrm{RBM}}(\mathbf{x},\mathbf{v}) = -\sum_{i=1}^{N_h}\sum_{j=1}^{N_v} w_{ij}h_i x_j - \sum_{j=1}^{N_v} b_j x_j - \sum_{j=1}^{N_h} c_i h_j$$

Model parameters: $\mathbf{W} \equiv w_{ij}, h_i, b_j$

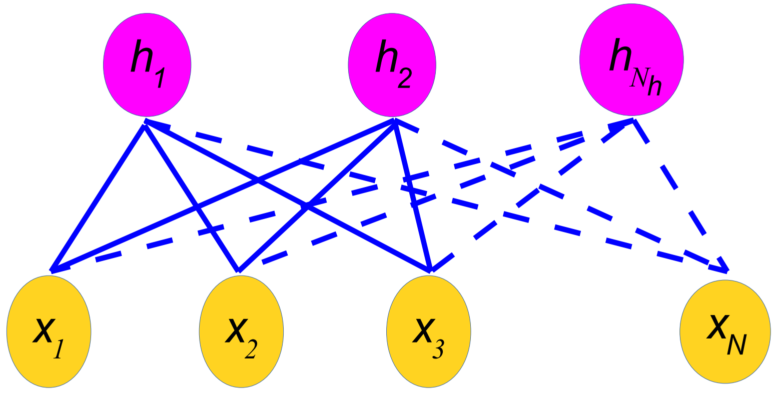
Probability of visible configuration $\mathbf{x}$: $p(\boldsymbol{x}) = \sum_{\mathbf{h}} p(\boldsymbol{x},\mathbf{h}) = \frac{1}{Z}\sum_{\mathbf{h}} \exp[-H_{\mathrm{RBM}}(\mathbf{x},\mathbf{v})]$

Partition function: $\mathrm{Z} = \sum_{\mathbf{x},\mathbf{h}} \exp[-H_{\mathrm{RBM}}(\mathbf{x},\mathbf{v})]$

Variational wave-functions with neural networks

Restricted Boltzmann machines

Carleo, Troyer, Science 2017



$$\psi(\mathbf{x}) = \prod_j \exp(a_j x_j) \prod_i 2\cosh\left(b_i + \sum_j w_{ij} x_j\right)$$

$\propto N \times N_h$ variational parameters

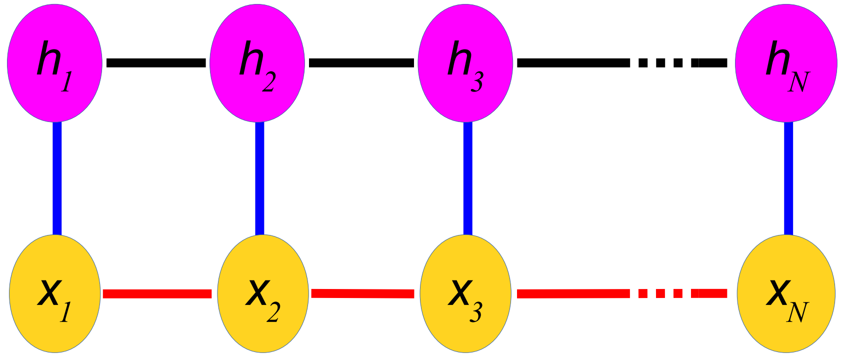
Hidden spins integrated out

unRestricted Boltzmann machines

alias shadow wave-function

Reatto, Masserini, PRB 1988

Vitiello, Runge, Kalos PRL 1988



$$\psi(\mathbf{x}) = \sum_{\mathbf{h}} \phi(\mathbf{x},\mathbf{h})$$

$$\phi(\mathbf{x},\mathbf{h}) = \exp\left(-k_1 \sum_i x_i x_{i+1}\right) \exp\left(-k_2 \sum_i h_i h_{i+1}\right) \exp\left(-k_3 \sum_i x_i h_i\right)$$

$k_1, k_2, k_3 = 3$ variational parameters

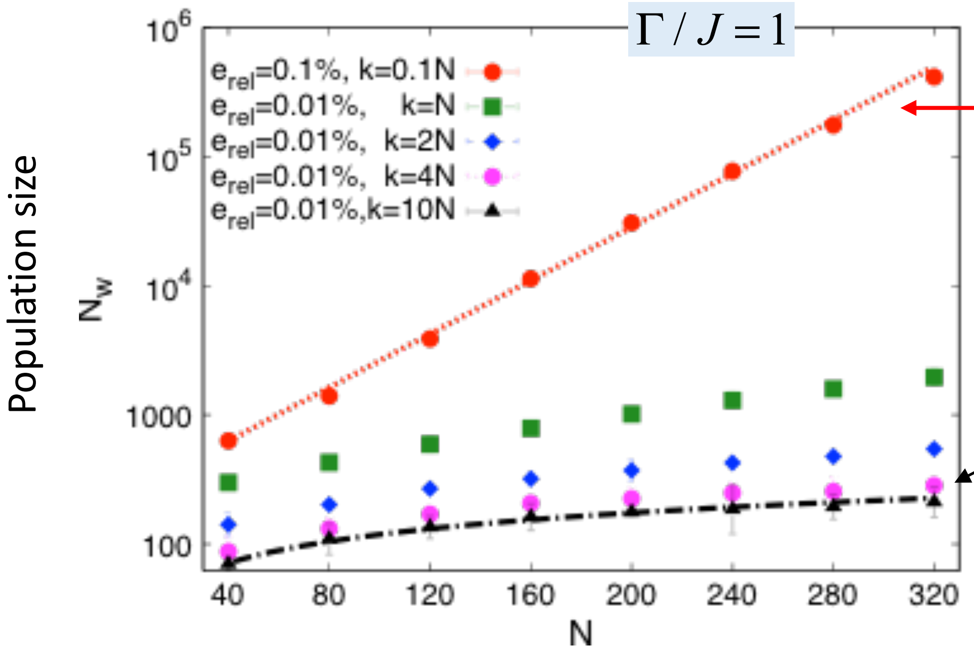
Need to sample hidden spins

Inack, Dell'Anna, Santoro, SP, PRB 2018

Computational complexity of PQMC guided by unRestricted BM: ferromagnetic Ising chain

Inack, Dell'Anna, Santoro, SP, PRB 2018

- Needs combined sampling of both visible and hidden spins



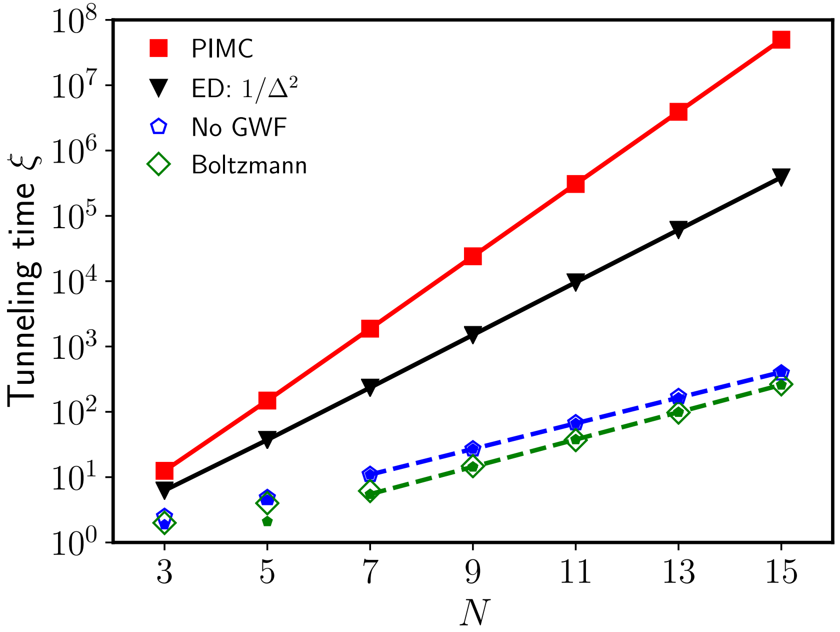
$$N_W \propto \exp(aN) \qquad a = 0.023(3)$$

$$N_W \propto N^b \qquad b = 0.5(1)$$

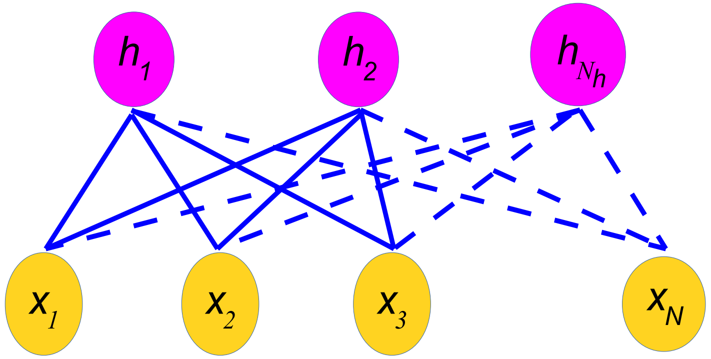
➔ polynomial computational cost

➔ scaling of tunneling time is not affected by guiding wf

Parolini, Inack, Giudici, SP, PRB (2019)



Restricted Boltzmann machine: unsupervised learning



Quantum state tomography:
Torlai, Mazzola, Carrasquilla, Troyer, Melko, Carleo, Nat. Phys. (2018)

Marginal probability: $P_{\mathbf{W}}(\mathbf{x}) = \sum_h P_{\mathbf{W}}(\mathbf{x},\mathbf{h}) = \frac{1}{Z}\sum_h \exp\left[-H_{\text{RBM}}(\mathbf{x},\mathbf{h})\right]$

Partition function: $Z = \sum_{\mathbf{x},\mathbf{h}} \exp\left[-H_{\text{RBM}}(\mathbf{x},\mathbf{h})\right]$

Log-likelihood: $L(\mathbf{W}) = \sum_{k=1}^{N_{\text{train}}} \ln P_{\mathbf{W}}(\mathbf{x}_k)$

Maximize log-likelihood, minimize KL divergence

Gradient ascent update rule: $W_m^{n+1} = W_m^n + \eta \frac{\partial L(\mathbf{W})}{\partial W_m}$

Gradient of log-likelihood: $\frac{\partial L(\mathbf{W})}{\partial J_{ij}} \propto \langle x_j h_i \rangle_{\text{data}} - \langle x_j h_i \rangle_{\text{model}}$

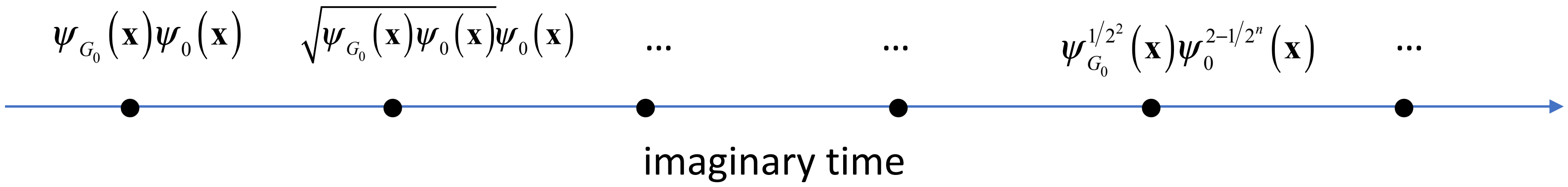
Performed via k-step contrastive divergence

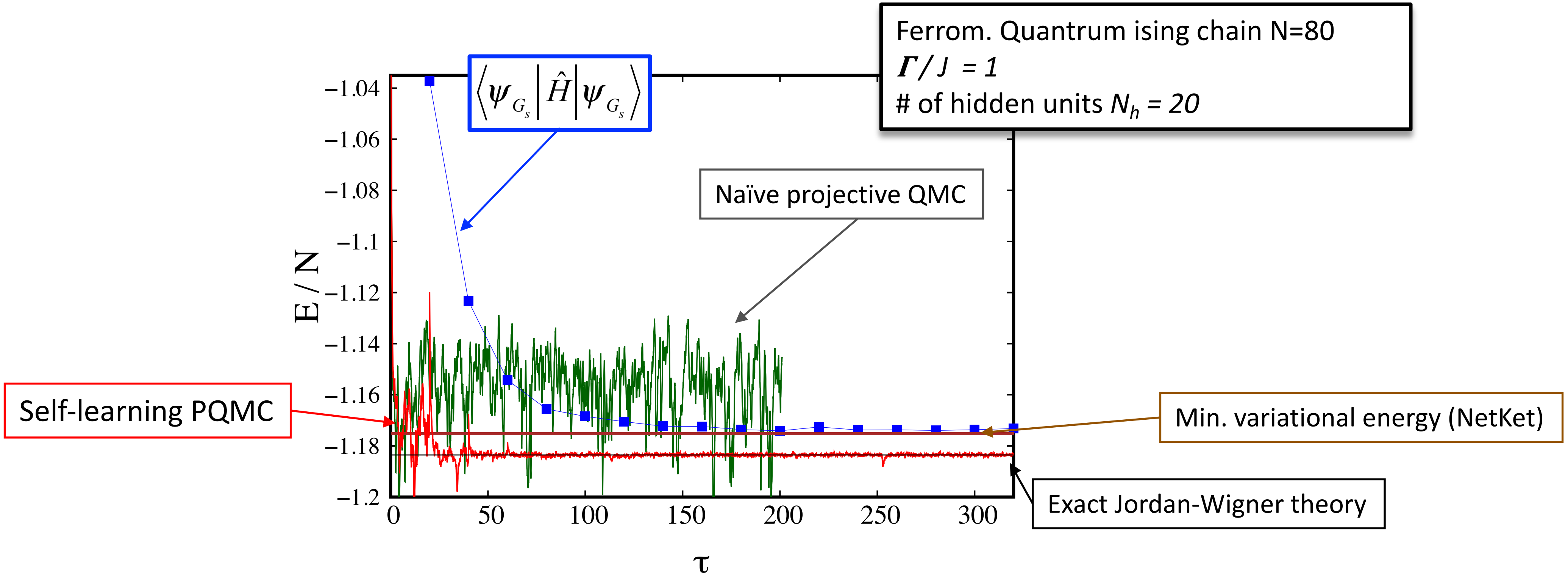
Self-learning projective QMC simulation

SP, Inack, Pieri, PRE (2019)

- The RBM learns the randomw-walker distribution: $P(\mathbf{x}) \propto \psi_G(\mathbf{x})\psi_0(\mathbf{x})$
- Guiding wf for the next run: $\psi_G(\mathbf{x}) = \sqrt{P(\mathbf{x})}$

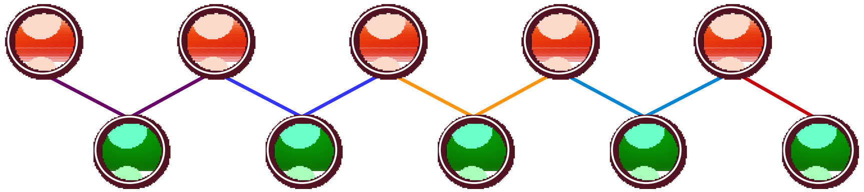
stoquastic model $\Rightarrow \psi_0(\mathbf{x}) \geq 0$



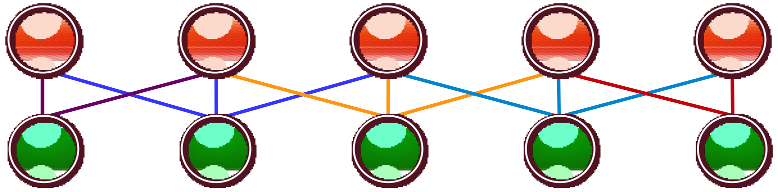


Random Ising chain: sparse neural networks

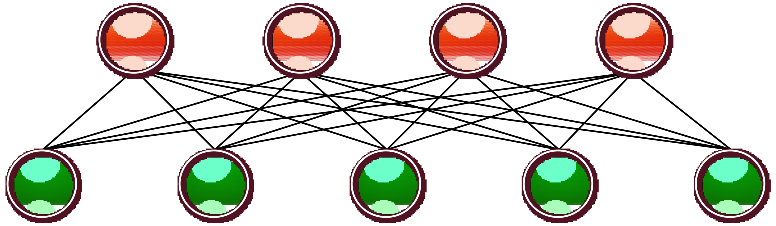
Connectivity $N_c = 2$



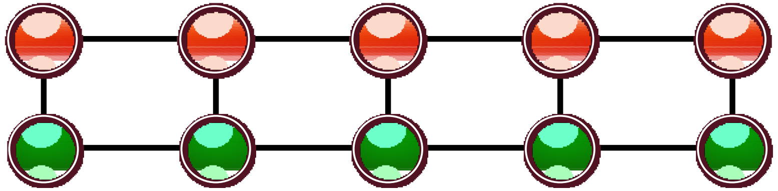
Connectivity $N_c = 3$



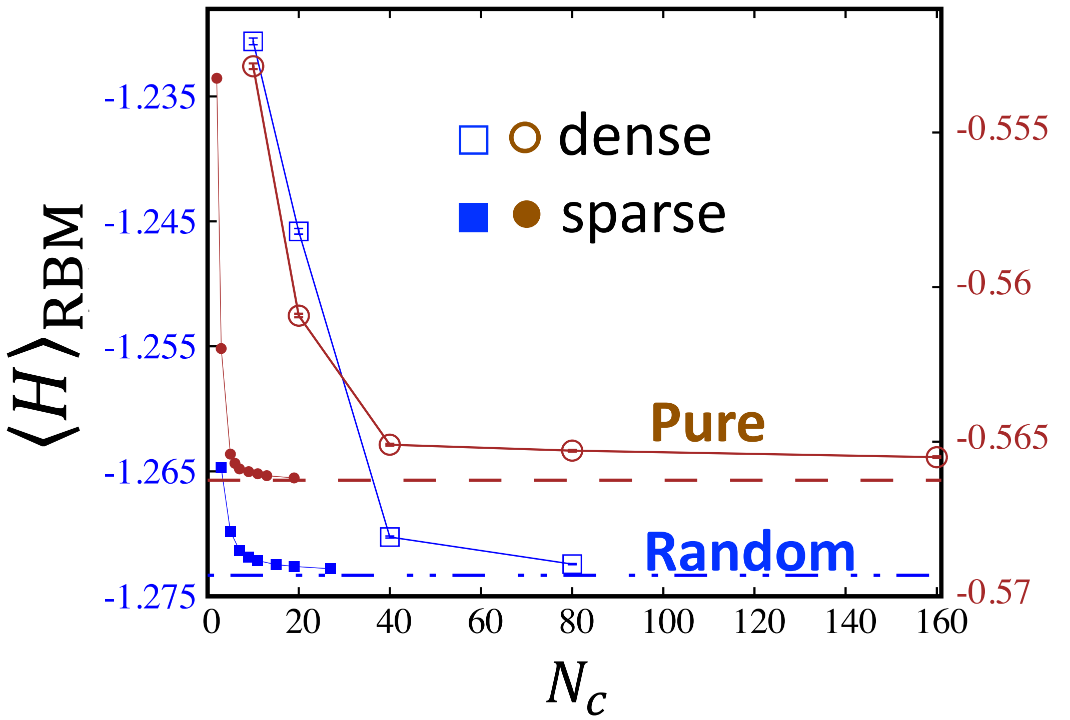
Dense

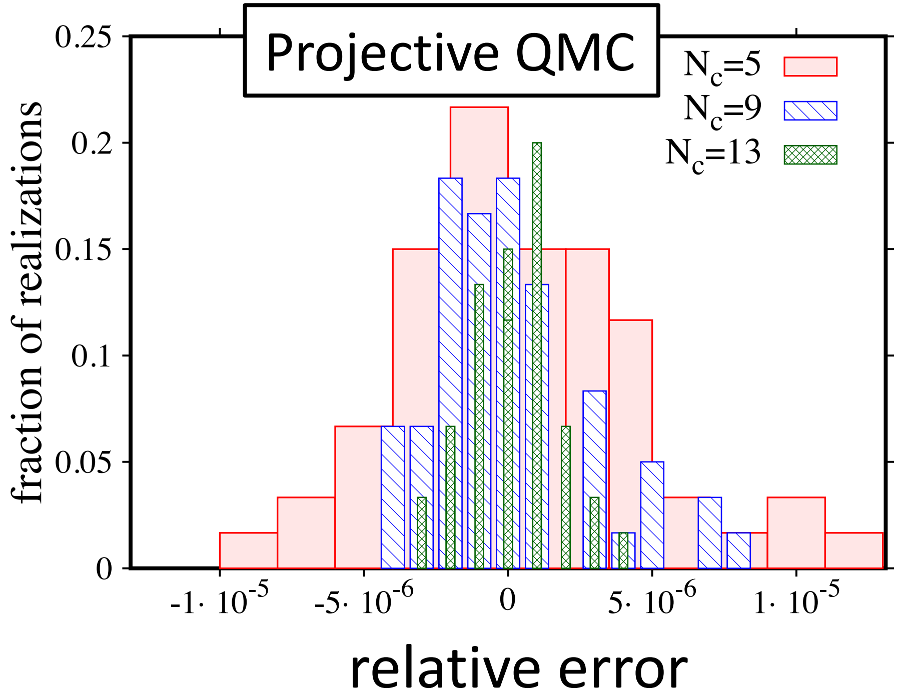


Shadow wf



SP, Pieri, PRE (2020)





Monte Carlo simulation of classical spin glasses

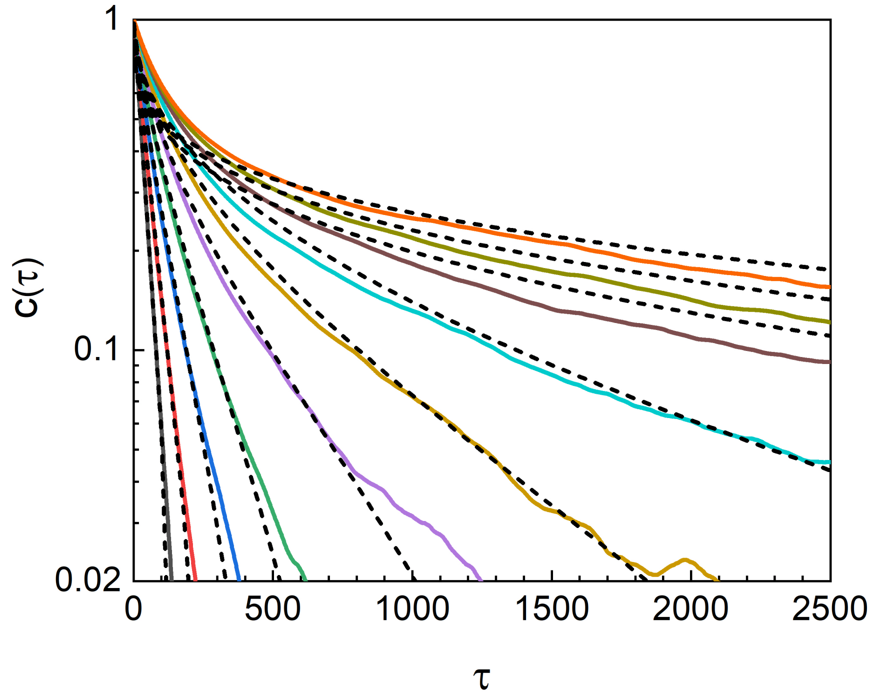
$$H = -\sum_{<ij>} J_{ij} x_i x_j$$

$x_i = \pm 1$ binary variables on a 2D square lattice, only nearest neighbor interaction

$J_{ij} \sim$ standard normal distribution

- Spin-glass phase at $T = 0$
- *"Freezing"* temperature at $k_B T \approx 1$

MC sim. with single-spin flip Metropolis algorithm
Autocorrelations: $c(\tau) = \frac{\langle E_{t+\tau} E_t \rangle - \langle E \rangle^2}{\langle E^2 \rangle - \langle E \rangle^2}$



$\beta = 1$
$\beta = 0.9$
⋮
$\beta = 0.1$

Neural autoregressive distribution estimator (NADE)

Problems with restricted Boltzmann machines:

- The normalization factor is ”intractable”.
- Sampling requires Markov chain Monte Carlo or alternated Gibbs sampling.

Solution: autoregressive property

Chain of conditional distributions:

$p(\mathbf{x}) = \prod_{d=1}^{N_v} p(x_d|\mathbf{x}_{<d})$ → **Ancestral sampling**

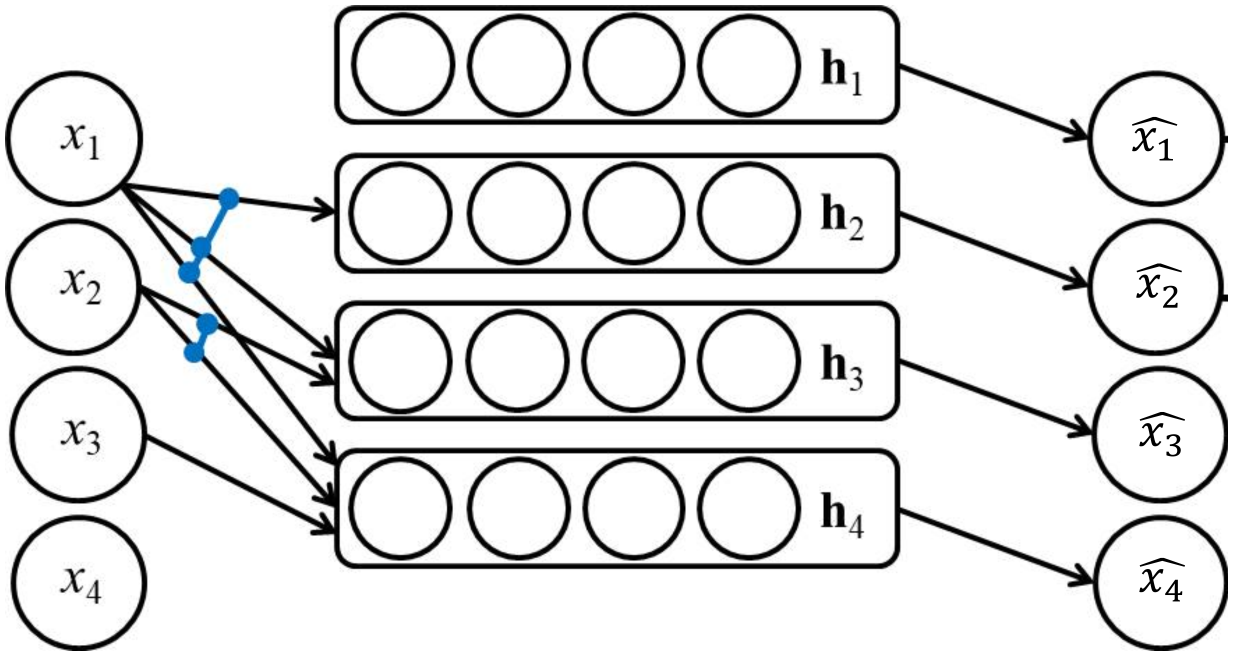
$\mathbf{x} = \left(x_1, \ldots, x_{N_v}\right)$

$\mathbf{x}_{<d} = (x_1, \ldots, x_{d-1})$

NADE:

Hugo Larochelle, Iain Murray, Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15 (2011).

- N_v "parallel" hidden layers
- Each hidden layer is connected only to the previous visible variables.
- Part of the weights from input-visible to hidden layers are shared.



$$\widehat{x_d} \equiv p(x_d|\mathbf{x}_{<d}) = \sigma(b_d + \boldsymbol{V}_d\, \mathbf{h}_d)$$
$$\mathbf{h}_{\boldsymbol{d}} = \sigma\left(\boldsymbol{c} \quad + \boldsymbol{W}_{\cdot,<d}\, \mathbf{x}_{<d}\right)$$

V_d= dth row of matrix $\boldsymbol{V}$

$\boldsymbol{W}_{\cdot,<d}$= matrix with first $d-1$ columns of $\boldsymbol{W}$

Activation function: sigmoid

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

Goal: $p_{\mathrm{NADE}}(\mathbf{x}) \cong p_{\mathrm{Boltzmann}}(\mathbf{x})$

Training configurations: $\boldsymbol{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_{N_{\mathrm{train}}}$

Obtained from:

- previous MCMC simulation
- sequential tempering
- Quantum annealer?

Maximize log-likelihood:

$$\log\left(\prod_{t=1}^{N_{\mathrm{train}}} p(\mathbf{x}_t)\right) = \frac{1}{N_{\mathrm{train}}} \sum_{t=1}^{N_{\mathrm{train}}} \log(p(\mathbf{x}_t)) = \frac{1}{N_{\mathrm{train}}} \sum_{t=1}^{N_{\mathrm{train}}} \sum_{d=1}^{N_v} \log\left(p\left(x_{t,d} | \mathbf{x}_{t,<d}\right)\right)$$

Minimize Kullback-Leibler divergence:

$$\mathrm{KL}(p_{\mathrm{NADE}} | p_{\mathrm{Boltzmann}}) = \langle \log(p_{\mathrm{NADE}}) \rangle_{p_{\mathrm{NADE}}} - \langle \log(p_{\mathrm{Boltzmann}}) \rangle_{p_{\mathrm{Boltzmann}}}$$

Use of trained NADE:

McNaughton, Milošević, Perali, SP, PRE (2020)

Compute expectation values: ancestral sampling instead of MCMC

$$\langle H \rangle_{p_{\mathrm{NADE}}} \cong \langle H \rangle_{p_{\mathrm{Boltzmann}}}$$

Proposal matrix in Metropolis-Hastings algorithm: $w_{\mathbf{x}'\mathbf{x}} = p_{\mathrm{NADE}}(\mathbf{x}')$ (use ancestral sampling)

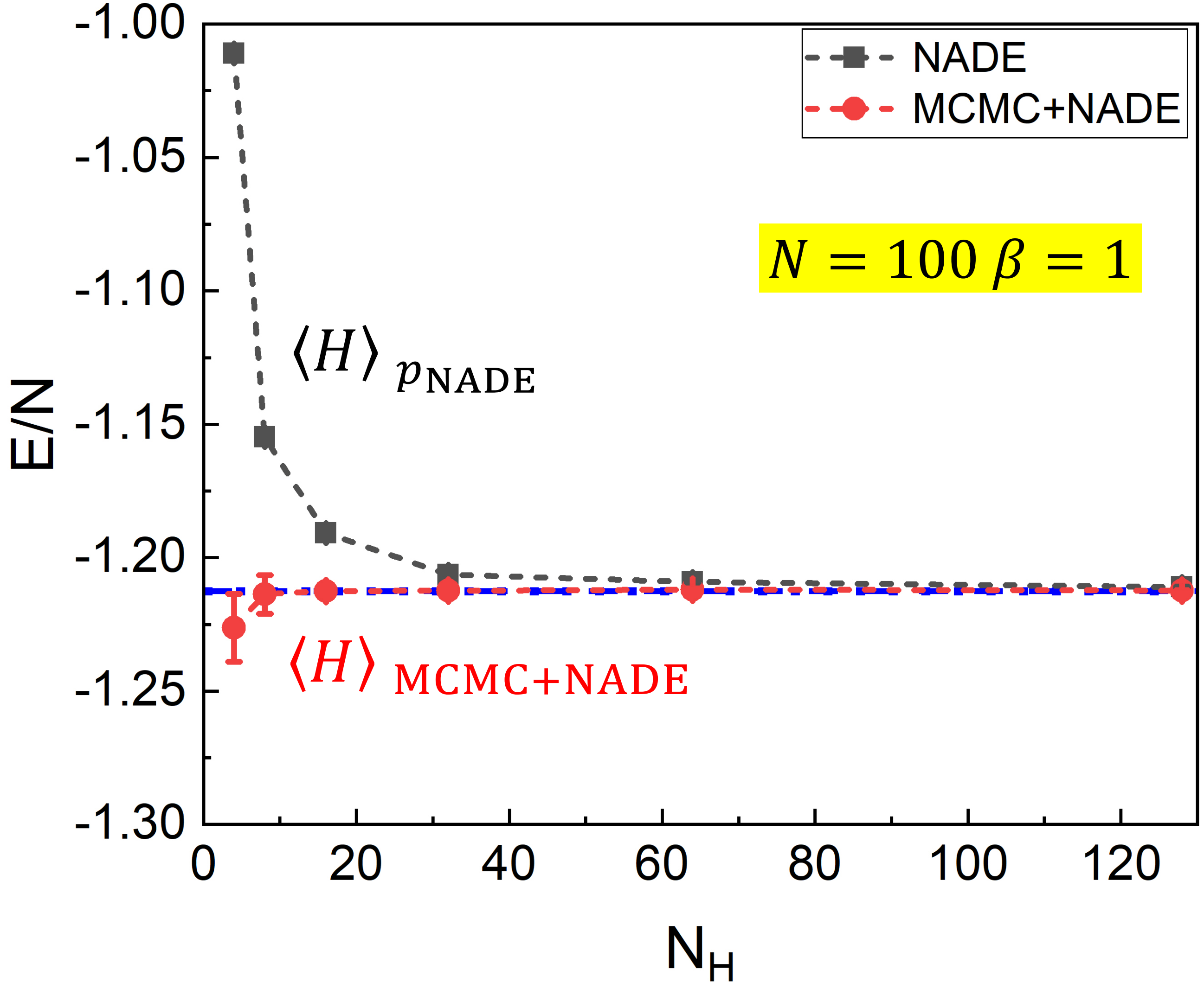
Acceptance probability: $A_{\mathbf{x}'\mathbf{x}} = \min\left(1, \dfrac{p_{\mathrm{Boltzmann}}(\mathbf{x}')w_{\mathbf{x}\mathbf{x}'}}{p_{\mathrm{Boltzmann}}(\mathbf{x})w_{\mathbf{x}'\mathbf{x}}}\right)$

Notice: if $p_{\mathrm{NADE}}(\mathbf{x}) \cong p_{\mathrm{Boltzmann}}(\mathbf{x})$ $\longrightarrow$ $A_{\mathbf{x}'\mathbf{x}} \cong \mathbf{1}$

Related work:

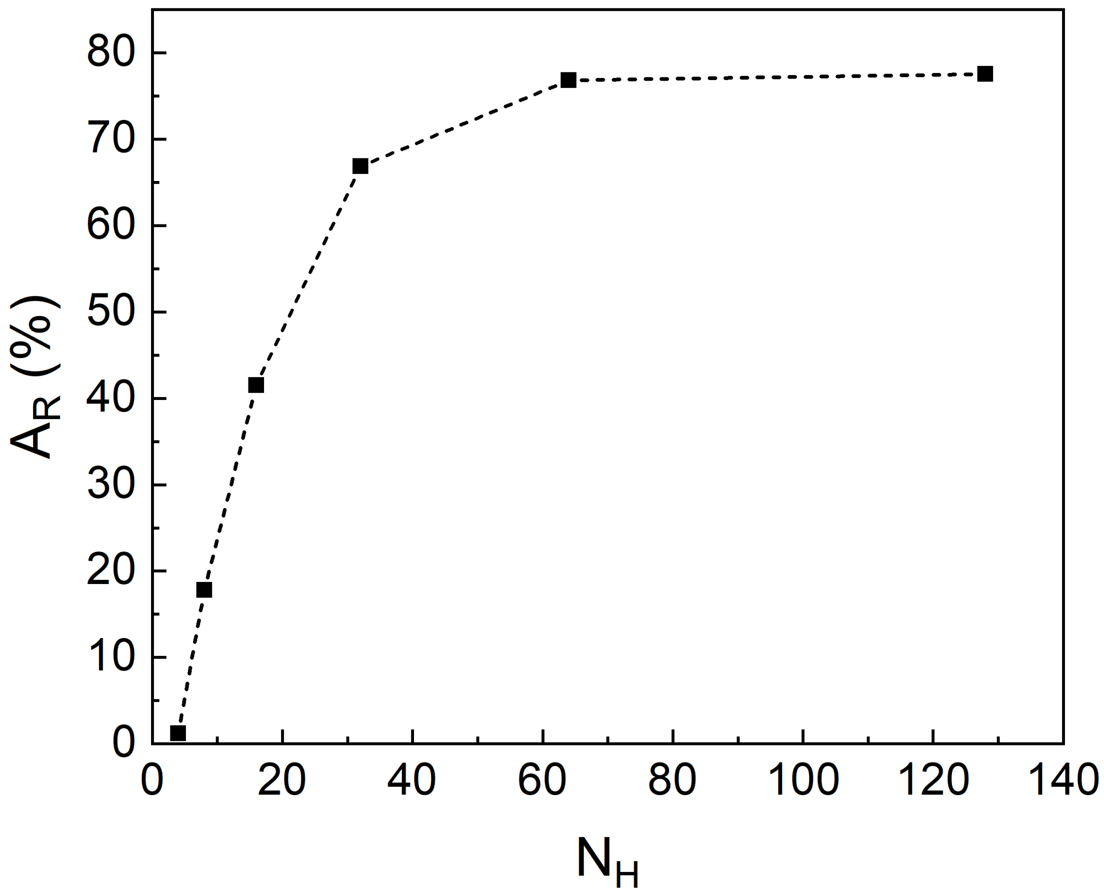
- D. Wu et al., PRL 122, 080602 (2019): variational optimization of autoregressive networks.
- K. A. Nicoli et al., PRE 101, 023304 (2020): MCMC+autoregressive n. for ferromagnetic models, training performed via variational optimization

McNaughton, Milošević, Perali, SP, PRE (2020)



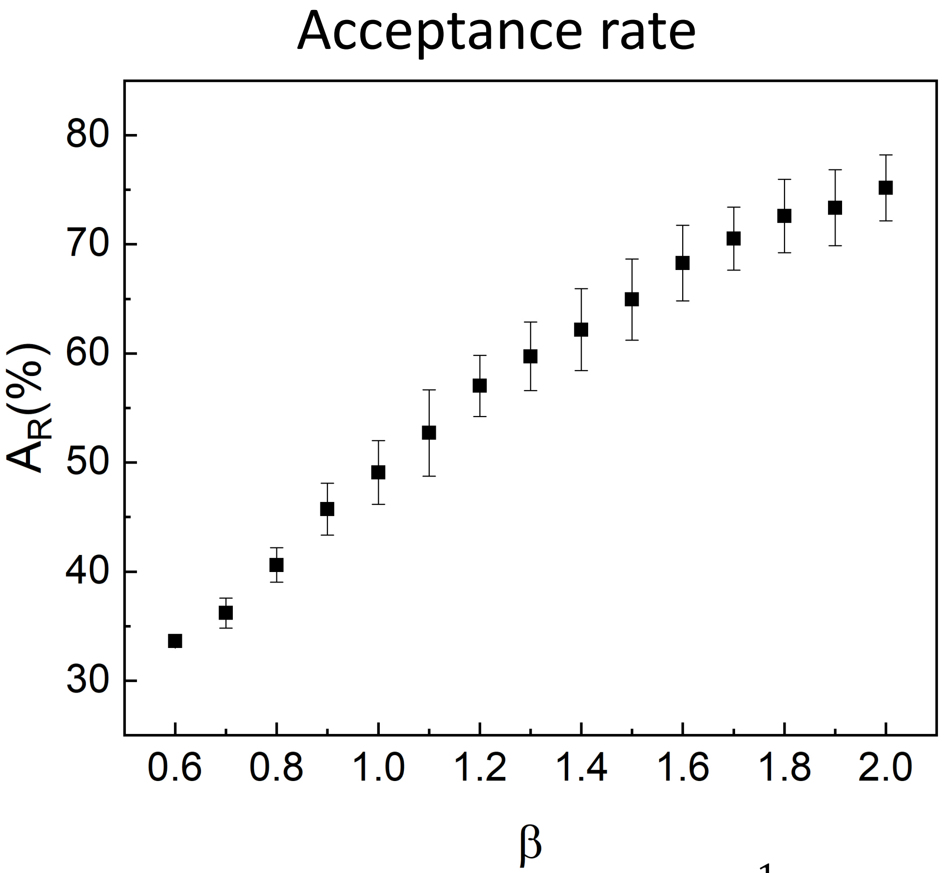
# of hidden neurons

Acceptance rate

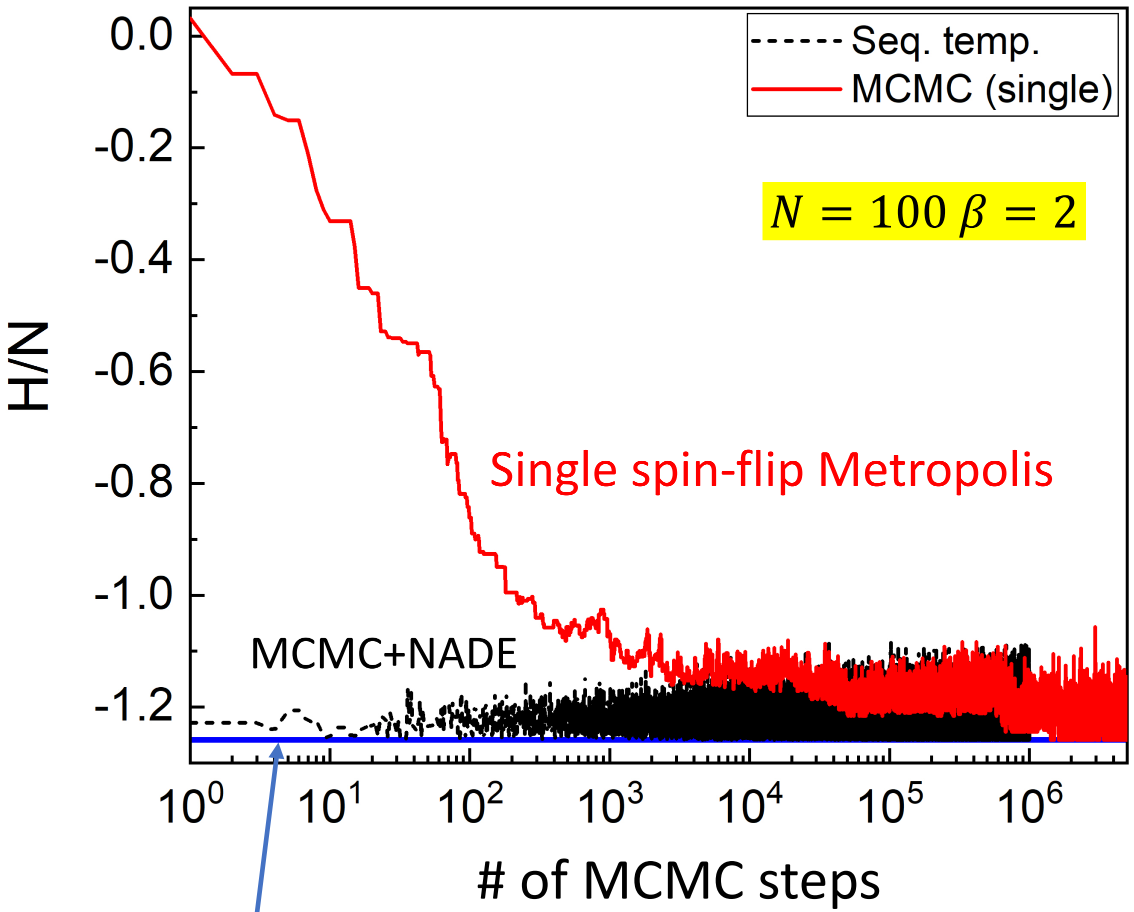


# of hidden neurons

Sequential tempering: train NADE a high(er) *T*, use it to drive MCMC a low(er) *T*



Inverse temperature $\beta = \frac{1}{k_B T}$



Ground state from spin glass server (University of Cologne)

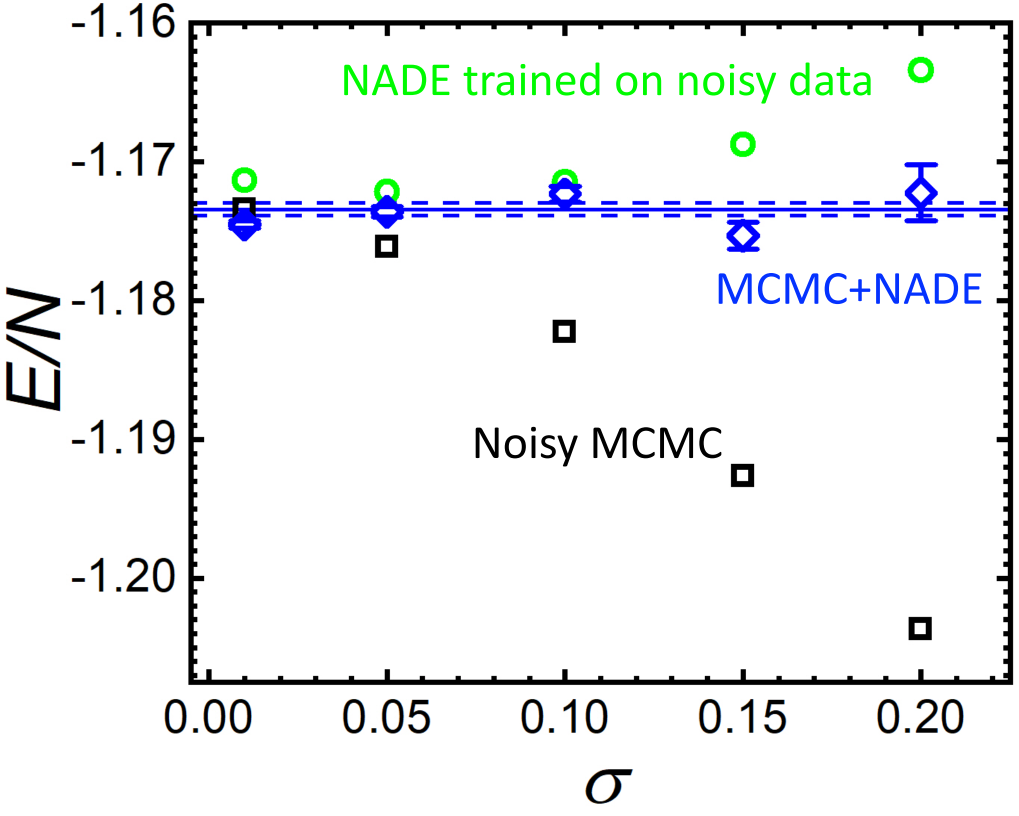
McNaughton, Milošević, Perali, SP, PRE (2020)

WORK IN PROGRESS (preliminary)

Training from noisy data:

-) Prospect of using physical quantum annealers (e.g., D-WAVE).
-) Procedure to “correct” experimental errors.

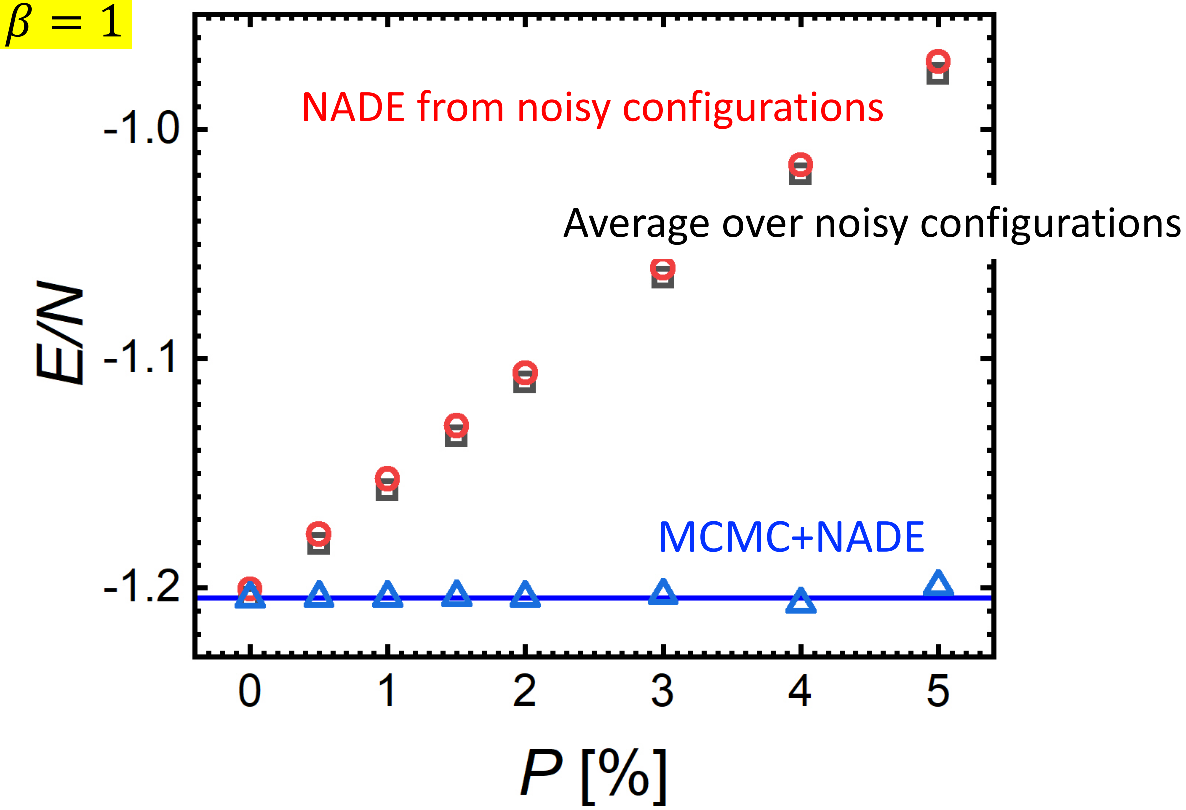
Errors in model implementation



St. dev. of Gaussian noise in coupling

$N = 100\ \beta = 1$

Readout errors



Probability to read flipped spin

Conclusions:

- Guiding wfs for projective QMC can be built using generative neural networks.
- They can be trained within self-learning QMC (unsupervised learning)
- Autoregressive neural networks allow generating smart proposals for the Metropolis-Hasting algorithms.
- Low T spin-glasses can be simulated via sequential tempering.
- Work in progress: autoregressive models can be trained with nosy data -> prospect of using physical quantum annealers.

References:

McNaughton, Milošević, Perali, SP, Physical Review E **101**, 053312 (2020)
SP, Pieri, Physical Review E **101**, 063308 (2020)
SP, Inack, Pieri, Physical Review E **100**, 043301 (2019)
Inack, Santoro, Dell'Anna, SP, Physical Review B **98**, 235145 (2018)
Inack, Giudici, Parolini, Santoro, SP, Physical Review A **97**, 032307 (2018)