Reconstructing cold atom wavefunctions with machine learning…

Roger Melko

UNIVERSITY OF WATERLOO

QUANTUM INTELLIGENCE LAB

Canada Research Chairs / Chaires de recherche du Canada

IQC Institute for Quantum Computing

VECTOR INSTITUTE

CREATIVE DESTRUCTION LAB

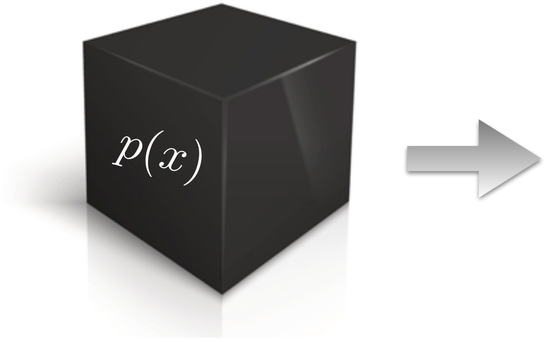
This talk

- ***Generative modelling*** for reconstructing quantum states

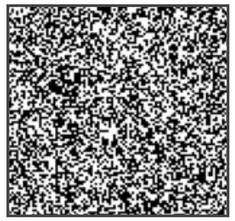
- Target experiments that can produce projective qubit measurements

Start with a classical probability distribution

- Imagine that you have a large number of measurements, samples, or configurations, drawn from some ***unknown*** probability distribution



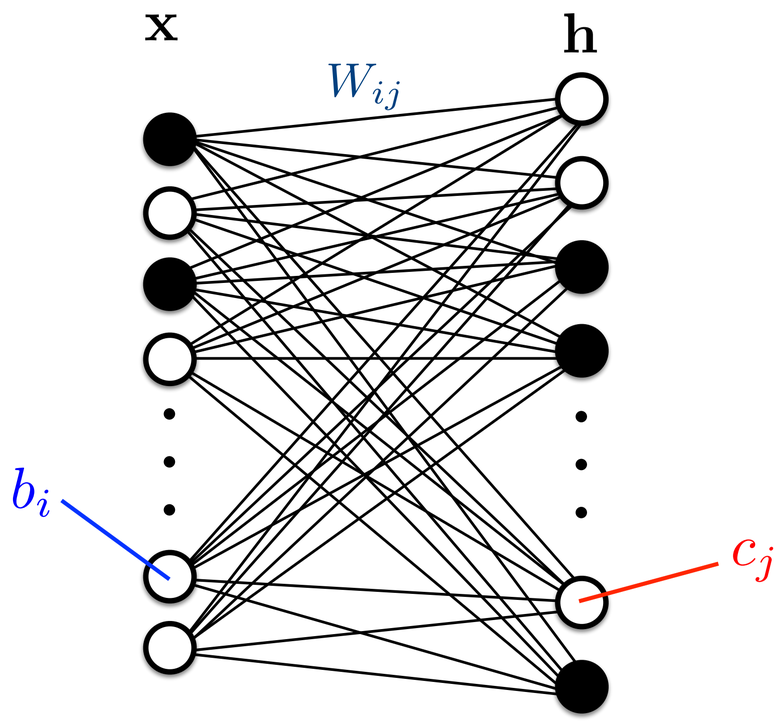
$$\mathbf{x}_1 = (1,0,0,1,1,1,0,0,0,0,\cdots,1)$$
$$\mathbf{x}_2 = (1,1,1,0,1,1,0,1,1,1,\cdots,1)$$
$$\mathbf{x}_3 = (0,1,1,0,0,1,0,1,0,1,\cdots,0)$$
$$\cdots$$
$$\mathbf{x}_{10,000} = (0,0,1,1,1,1,0,1,0,0,\cdots,0)$$



Restricted Boltzmann Machine

Smolensky, Hinton, Salakhutdinov, Bengio

N visible units $\quad$ n_h hidden units



$x_i = 0, 1 \qquad h_j = 0, 1$

RBMs are "energy-based" models:

$$p_\lambda = \frac{1}{Z_\lambda} e^{-E_\lambda(\mathbf{x},\mathbf{h})}$$

joint probability distribution

$$E_\lambda(\mathbf{x}, \mathbf{h}) = -\sum_{ij} W_{ij} x_i h_j - \sum_i b_i x_i - \sum_j c_j h_j$$

"Training" means tuning the machine parameters to get the marginal distribution $p_\lambda(\mathbf{x})$ to approximate the (unknown) target distribution

$$\lambda = \{W, b, c\} \qquad p_\lambda(\mathbf{x}) = \sum_{\mathbf{h}} p_\lambda(\mathbf{x}, \mathbf{h})$$

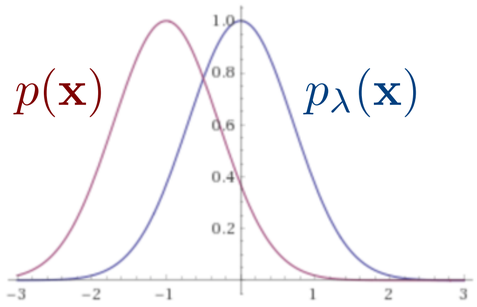
model parameters

Training the RBM

G. Hinton, Neural computation 14, 1771 (2002)

Training means tuning the machine parameters to minimize the difference between the marginal distribution $p_\lambda(\mathbf{x}) = \sum_{\mathbf{h}} p_\lambda(\mathbf{x}, \mathbf{h})$ and the (unknown) physical "target" distribution

Define a cost function to minimize: the ***Kullback-Leibler divergence***



$$\mathrm{KL}(p||p_\lambda) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{p_\lambda(\mathbf{x})} \geq 0$$

- A non-symmetric measure of the distance between two distributions
- Always positive, and zero iff $p = p_\lambda$

$$\mathrm{KL}(p||p_\lambda) = \sum_{\mathbf{x}} p(\mathbf{x}) \log p(\mathbf{x}) - \sum_{\mathbf{x}} p(\mathbf{x}) \log p_\lambda(\mathbf{x})$$

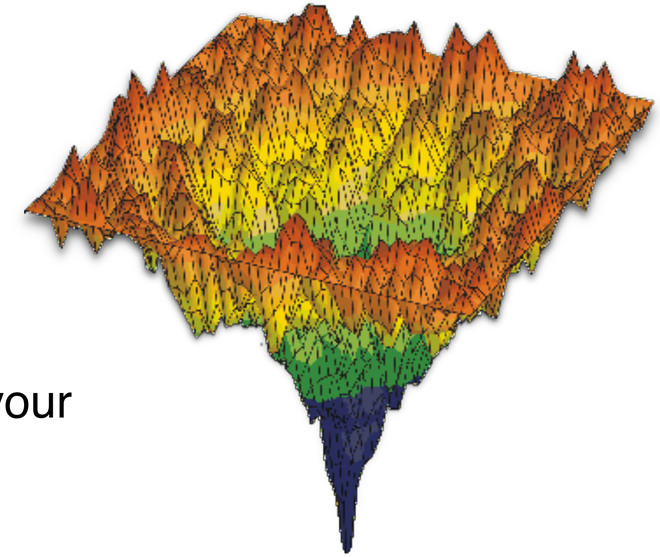
the entropy of p

depends on the parameters over which to optimize

$$= -\langle \log p_\lambda(\mathbf{x}) \rangle_p \approx -\sum_i \log p_\lambda(\mathbf{x}_i)$$

Equivalent to maximizing the "log-likelihood" $\mathcal{L} = \langle \log p_\lambda(\mathbf{x}) \rangle_p$

Stochastic Gradient Descent



The optimization landscape is thus defined - find minima using gradient descent

$$\lambda' = \lambda - \eta \nabla \mathcal{L} \qquad \lambda = \{W, b, c\}$$

The full gradient is too costly to calculate. Instead sample some number m of your dataset, and perform *stochastic* gradient descent

$$\frac{1}{m} \sum_{j=1}^{m} \nabla \mathcal{L}(\mathbf{x}_j) \approx \nabla \mathcal{L}$$

"mini-batch" size = m

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -\sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x}) \frac{\partial E}{\partial \lambda} + \sum_{\mathbf{x},\mathbf{h}} p(\mathbf{x},\mathbf{h}) \frac{\partial E}{\partial \lambda}$$

- The first term is computationally easy to calculate.

- The second term is hard. Requires a MCMC to generate samples from the stationary distribution of the machine. In practice a short chain of k steps is run

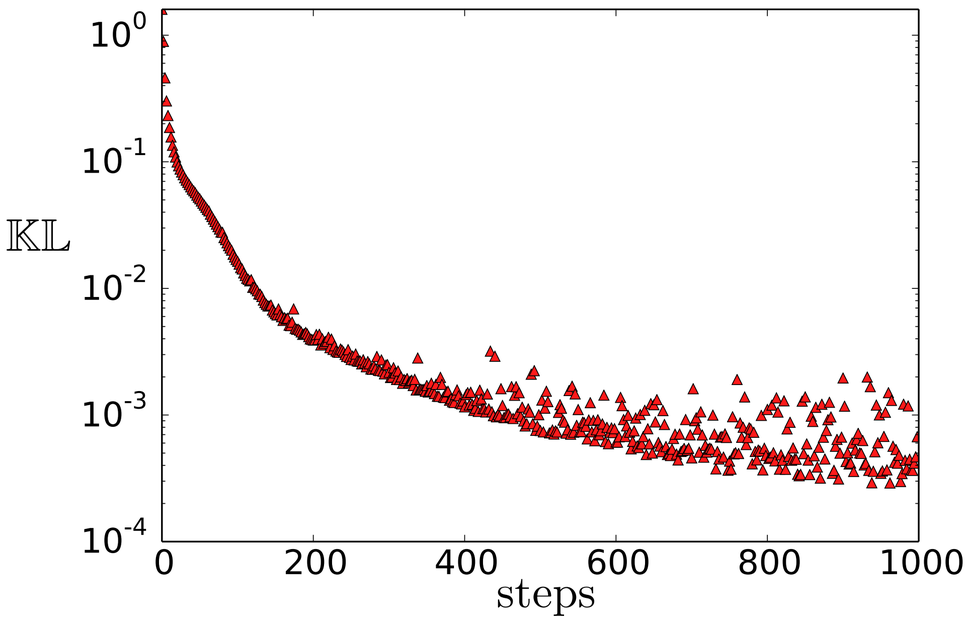
"Contrastive Divergence"

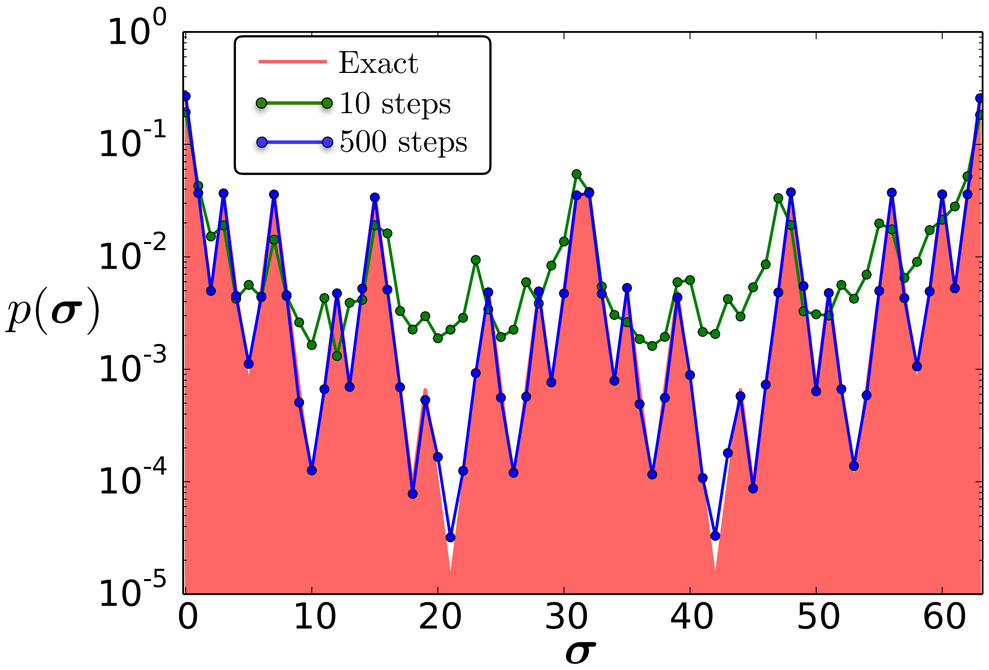
CD_k

Stochastic Gradient Descent

Torlai and RGM, Phys. Rev. B 94, 165134 (2016)

Example: 1D Ising model with 6 spins, trained using CD_5

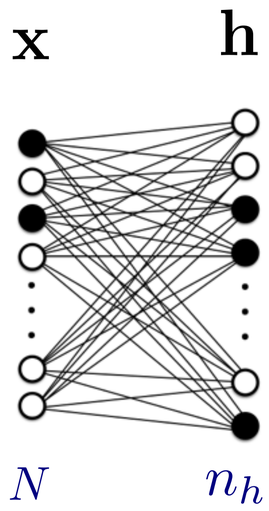




KL not possible to calculate in the general case for larger N.

Other metrics, like physical observables, could be used to validate the training…

Sampling the RBM



After the training is complete, and the weights and biases are stationary, one can use Block Gibbs sampling to produce new configurations

$$x_0 \to h_0 \to x_1 \to h_1 \to \cdots \to x_k \to h_k$$

$$\langle \mathcal{O} \rangle_{\text{joint}} = \frac{1}{Z} \sum_{x,h} \mathcal{O} \cdot p_\lambda(x,h)$$

Restrict the observable to the visible layer only $\mathcal{O} = \mathcal{O}_x$

$$\langle \mathcal{O} \rangle_{\text{joint}} = \sum_x \mathcal{O}_x \underbrace{\boxed{\sum_h p_\lambda(x,h)}}_{p_\lambda(x) \to p(x)} \approx \langle \mathcal{O} \rangle_{\text{physical}}$$

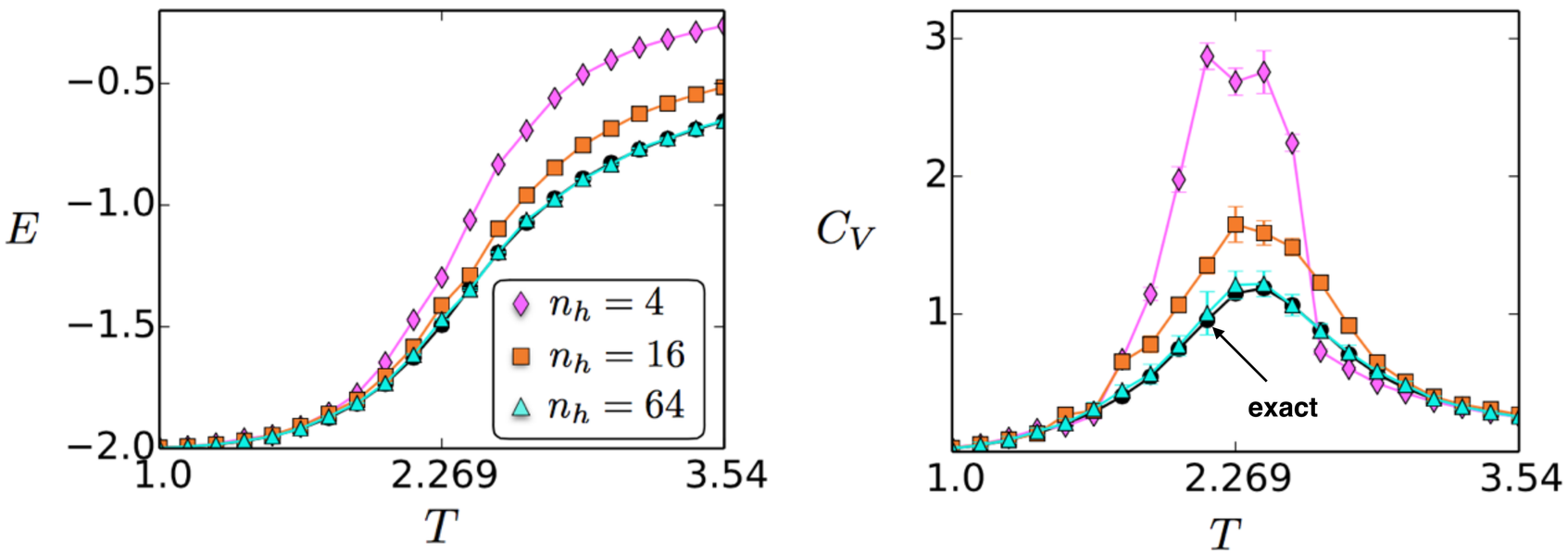
We can ask, how well do physical estimators calculated in this way match the exact values?

Example: 2D (classical) Ising model

Torlai and RGM, Phys. Rev. B 94, 165134 (2016)

Results from the generative model, after training:

$N = 64$



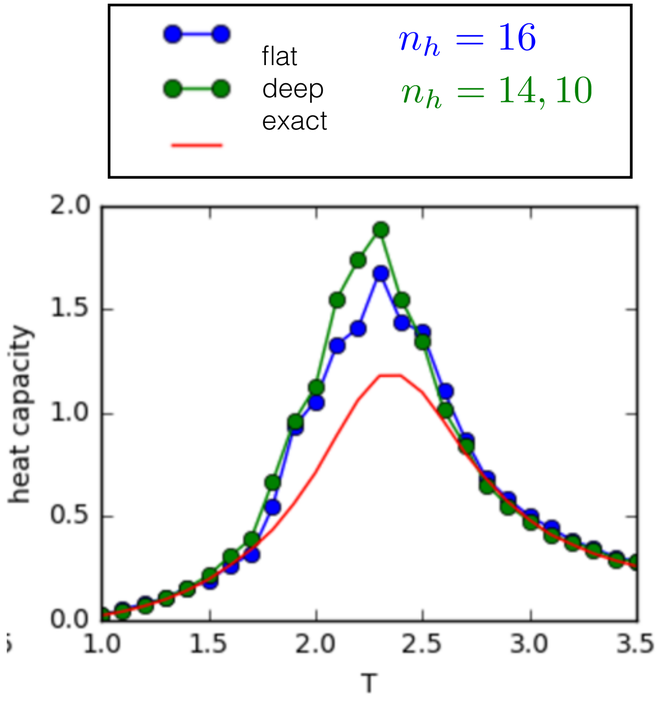
Here, the number of hidden units required for accurate generative modelling is approximately the same as the number of physical spins.

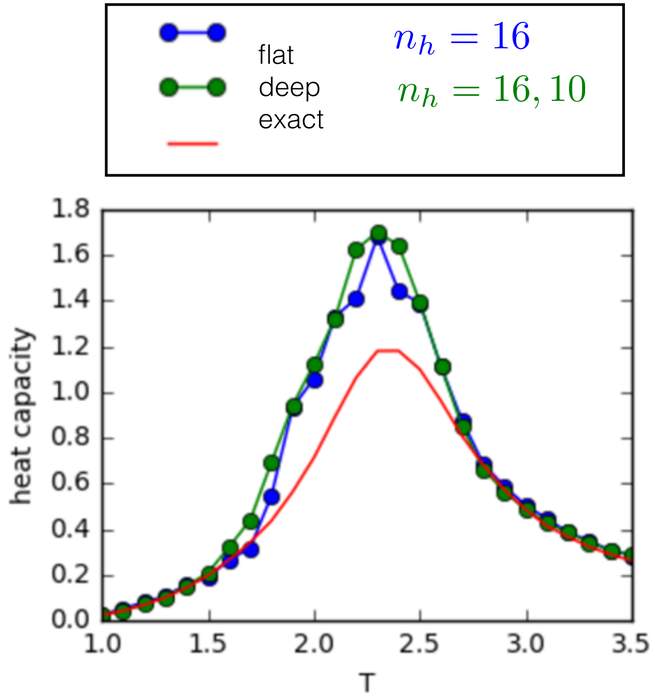
The number of measurements required for training also affects efficiency…

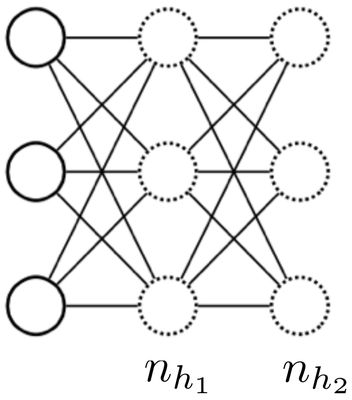
Does representational power depend on depth?

Morningstar and RGM, arXiv:1708.04622

This can be concretely addressed in physical systems like the Ising model







Deep learning and the renormalization group, Cédric Bény, arXiv:1301.3124
An exact mapping between the Variational Renormalization Group and Deep Learning, Mehta, Schwab, arXiv:1410.3831
Why does deep and cheap learning work so well? Lin and Tegmark, arXiv:1608.08225
Efficient Representation of Quantum Many-body States with Deep Neural Networks, Gao, Duan, arXiv:1701.05039

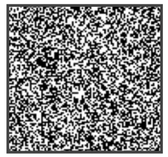
Learning (real +ve) wavefunctions

Torlai, Mazzola, Carrasquilla, Troyer, RGM, Carleo, Nature Physics 14, 447-450 (2018)

In the case where the wavefunction is real and positive in a certain basis

$$\psi_\lambda(\mathbf{x}) \propto \sqrt{p_\lambda(\mathbf{x})}$$

Train with samples in the S^z basis



and after calculate estimators from samples produced on the trained machine

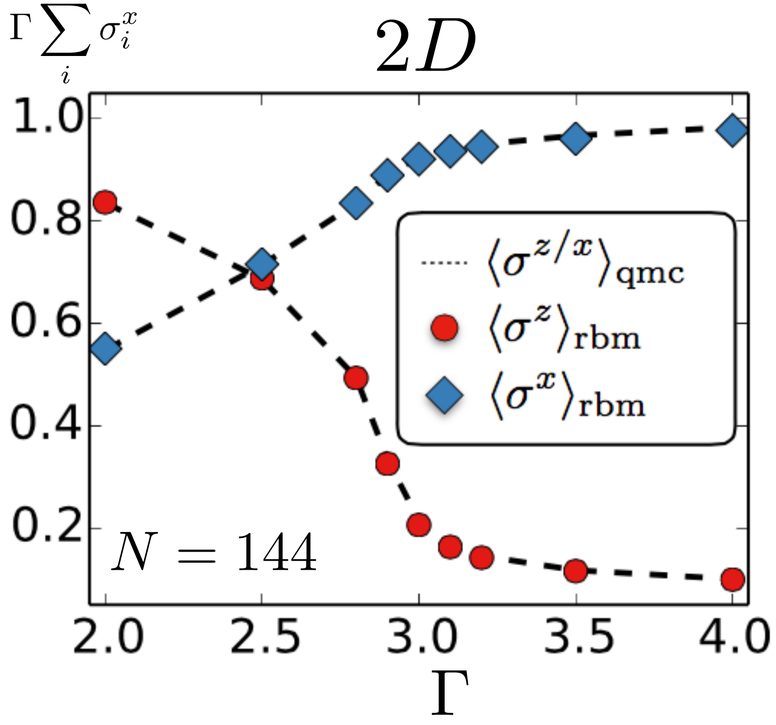
$$\langle \mathcal{O}^{\mathrm{D}} \rangle = \sum_{\mathbf{x}} p_\lambda(\mathbf{x}) \mathcal{O}_{\mathbf{x}}$$

$$\langle \mathcal{O}^{\mathrm{OD}} \rangle = \sum_{\mathbf{x}\mathbf{x}'} \sqrt{p_\lambda(\mathbf{x})}\sqrt{p_\lambda(\mathbf{x}')} \mathcal{O}_{\mathbf{x}\mathbf{x}'}$$

$$= \sum_{\mathbf{x}} p_\lambda(\mathbf{x}) \boxed{\sum_{\mathbf{x}'} \frac{\sqrt{p_\lambda(\mathbf{x}')}}{\sqrt{p_\lambda(\mathbf{x})}} \mathcal{O}_{\mathbf{x}\mathbf{x}'}}$$

"local" estimator

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



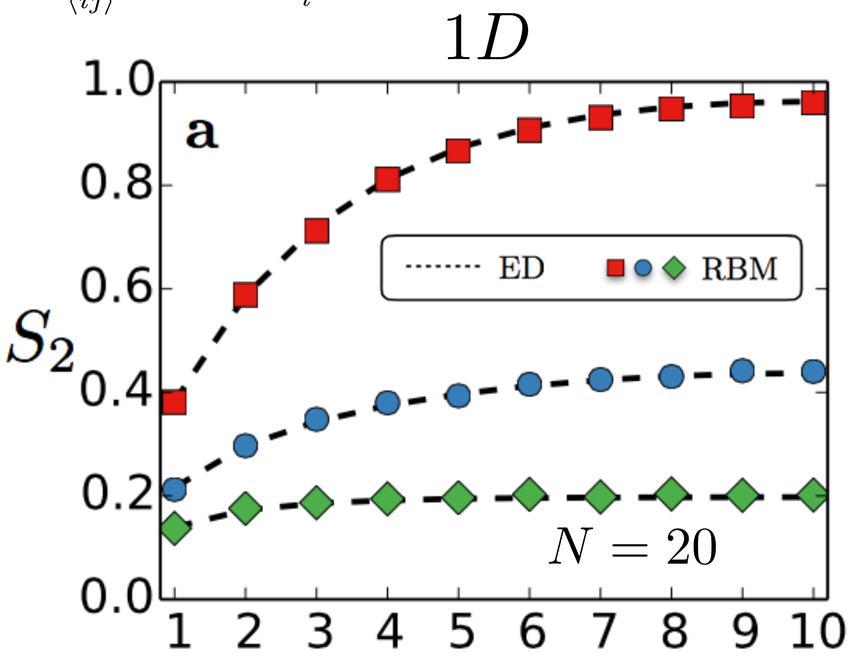
Learning (real +ve) wavefunctions

Torlai, Mazzola, Carrasquilla, Troyer, RGM, Carleo, Nature Physics 14, 447-450 (2018)

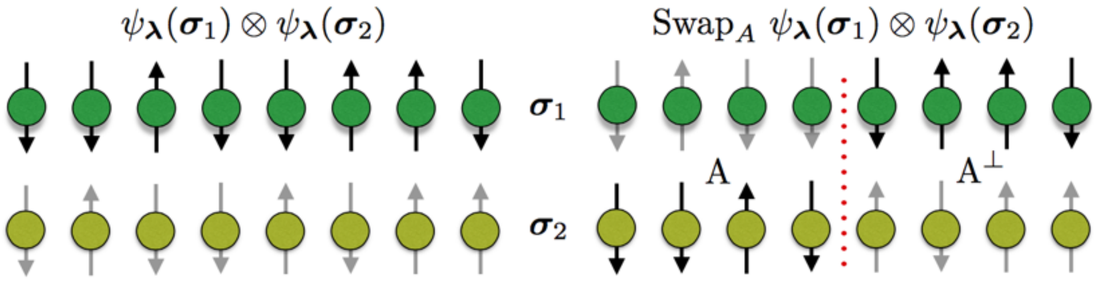
Entanglement entropy:

$$S_2(\rho_A) = -\log\left[\mathrm{Tr}(\rho_A^2)\right]$$

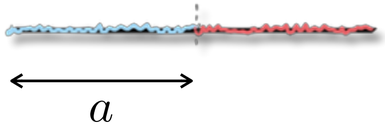
$$H = -\sum_{\langle ij\rangle} \sigma_i^z \sigma_j^z - \Gamma \sum_i \sigma_i^x$$



$$\psi_\lambda(\mathbf{x}) \propto \sqrt{p_\lambda(\mathbf{x})}$$

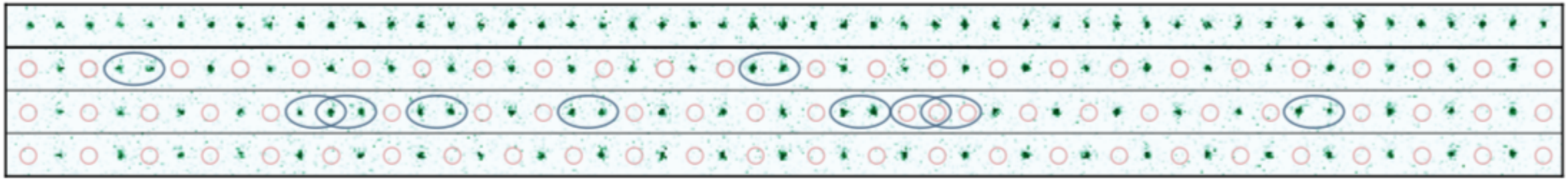


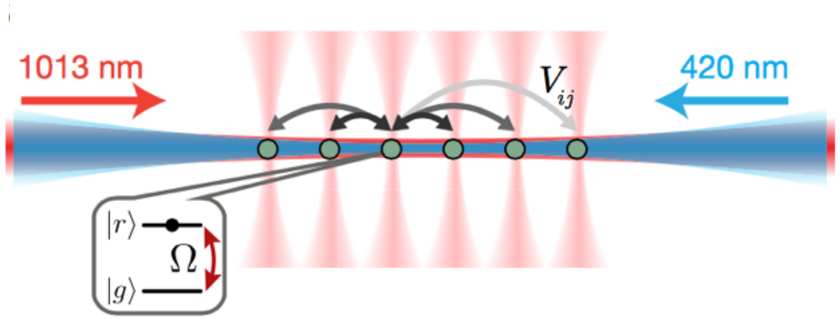
The expectation of the swap operator can be measured in the replicated (trained) RBM with a simple, universal algorithm that scales well.



Experiments: Rydberg atoms

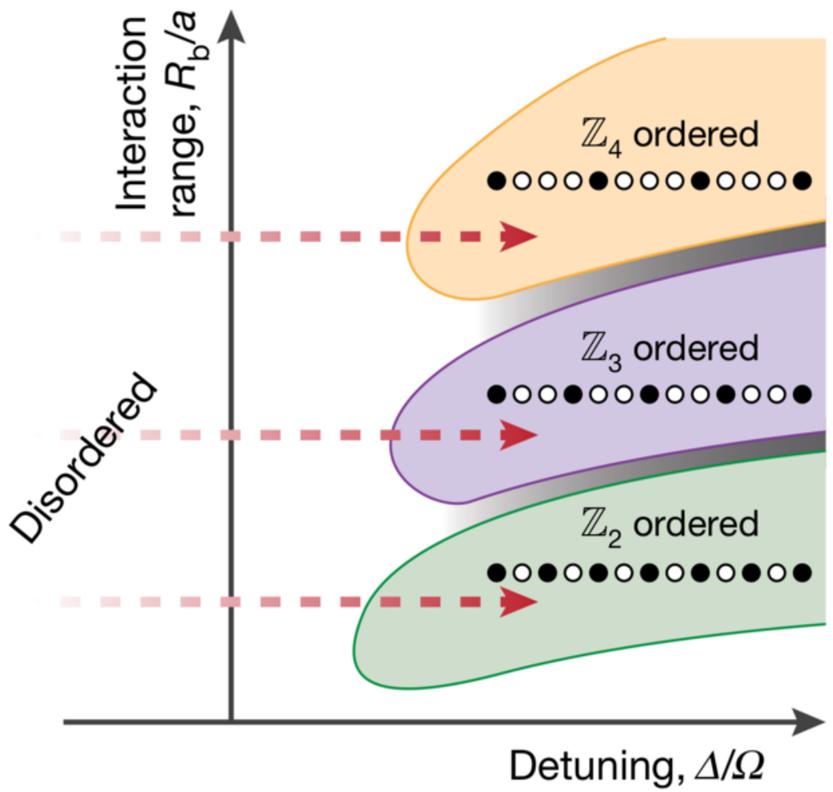
Bernien, Schwartz, Keesling, Levine, Omran, Pichler, Choi, Zibrov, Endres, Greiner, Vuletić, Lukin Nature 551, 579 (2017)





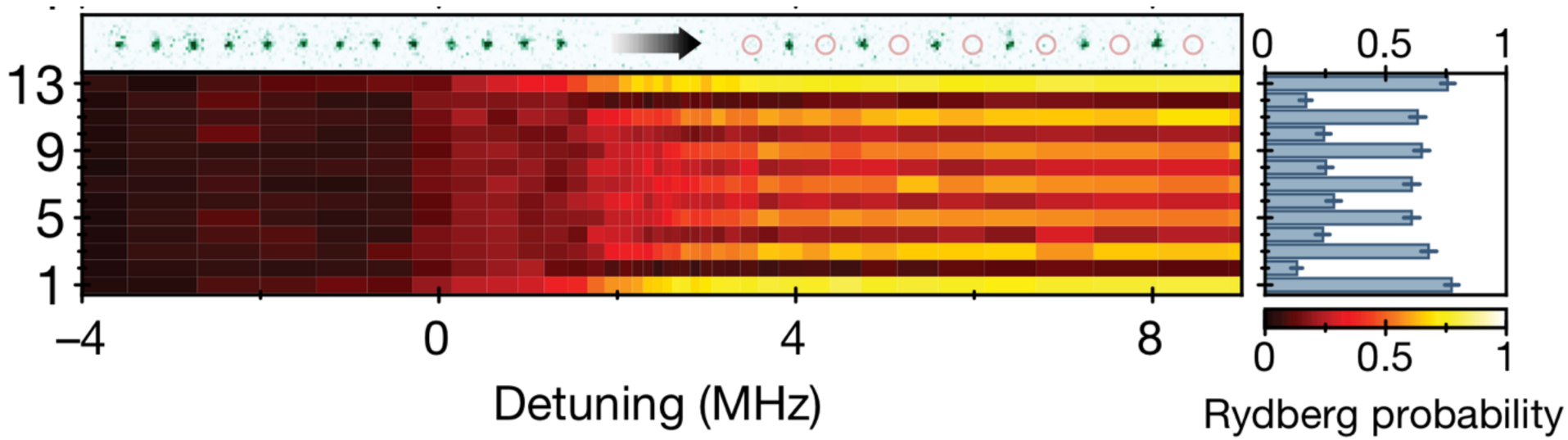
$$H = \Omega \sum_i \sigma_i^x - \Delta \sum_i n_i + \sum_{i<j} V_{ij} n_i n_j$$

$$\sigma^x = |g\rangle\langle r| + |r\rangle\langle g| \qquad n = |r\rangle\langle r|$$



Experiments: Rydberg atoms

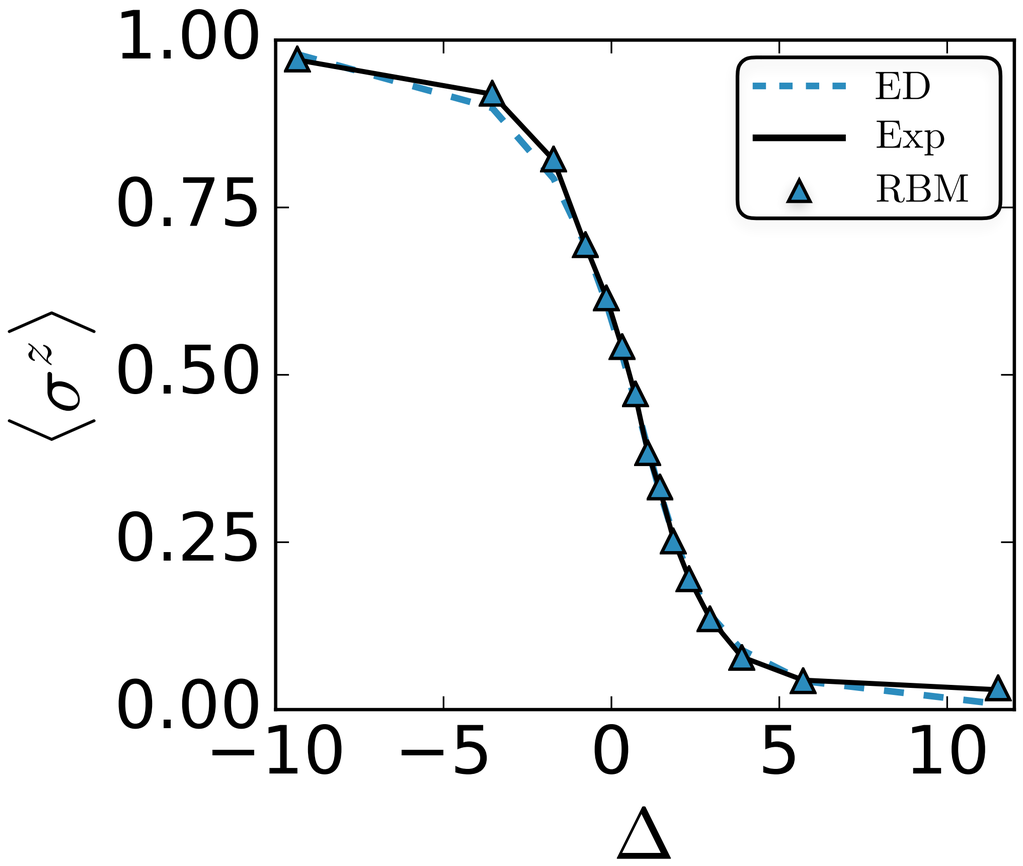
Bernien, Schwartz, Keesling, Levine, Omran, Pichler, Choi, Zibrov, Endres, Greiner, Vuletić, Lukin Nature 551, 579 (2017)

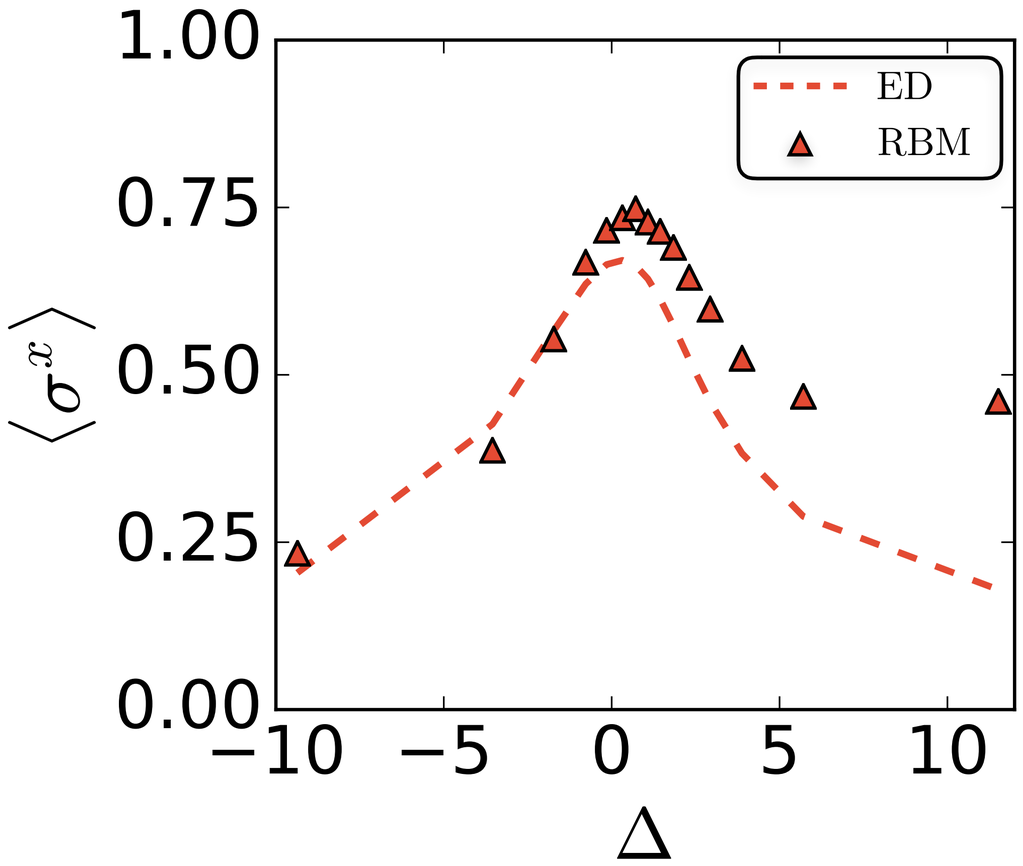


- Approximately 3000 projective measurements per detuning parameter Δ
- Assume purity and positivity of the wavefunction, use this data as a training set for an RBM
- After training, sample the RBM and produce estimators

Reconstructed estimators

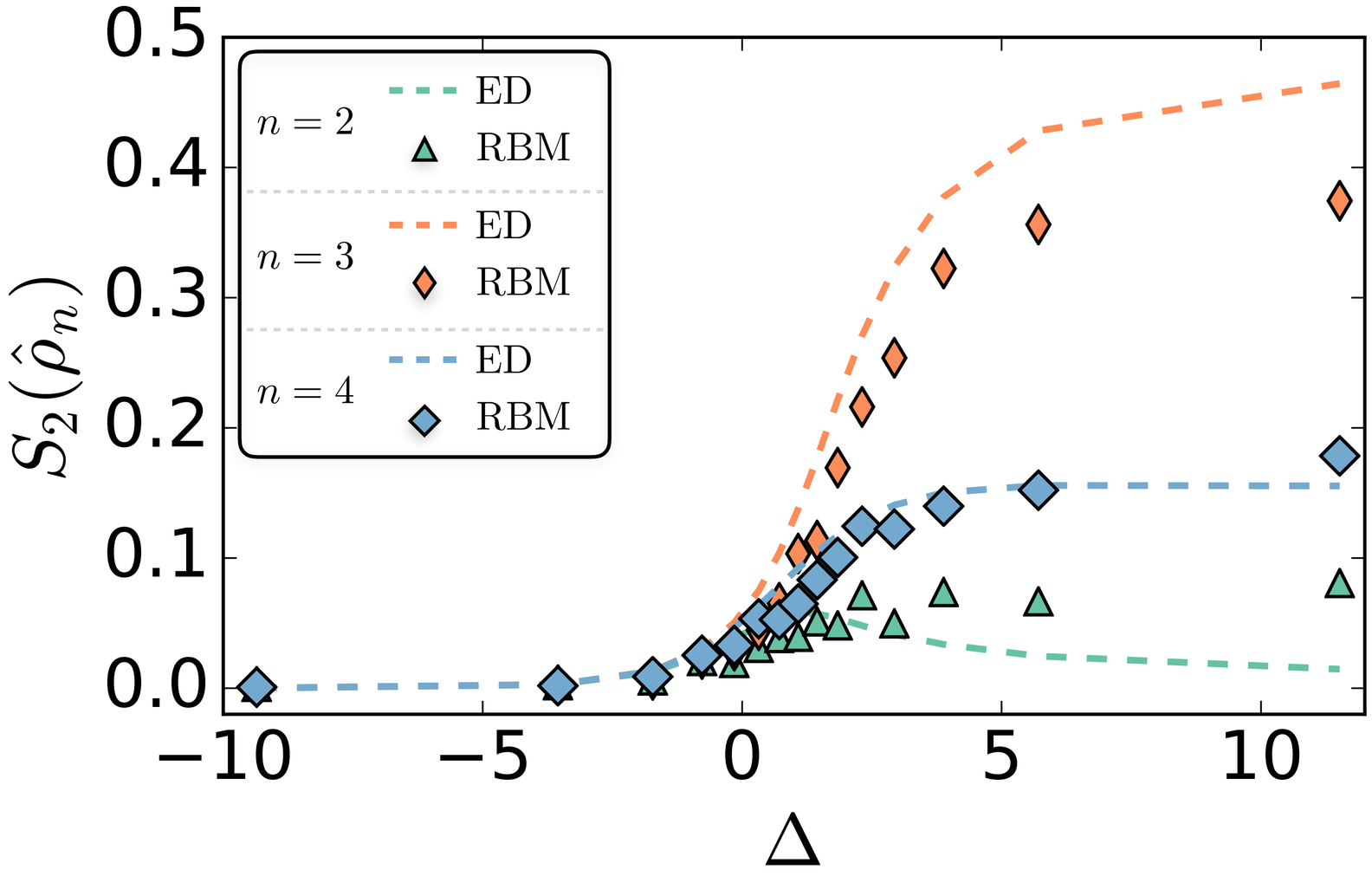
Torlai, Timar, van Nieuwenburg, Levine, Omran, Keesling, Bernien, Greiner, Vuletić, Lukin, RGM, Endres, arXiv:1904.08441





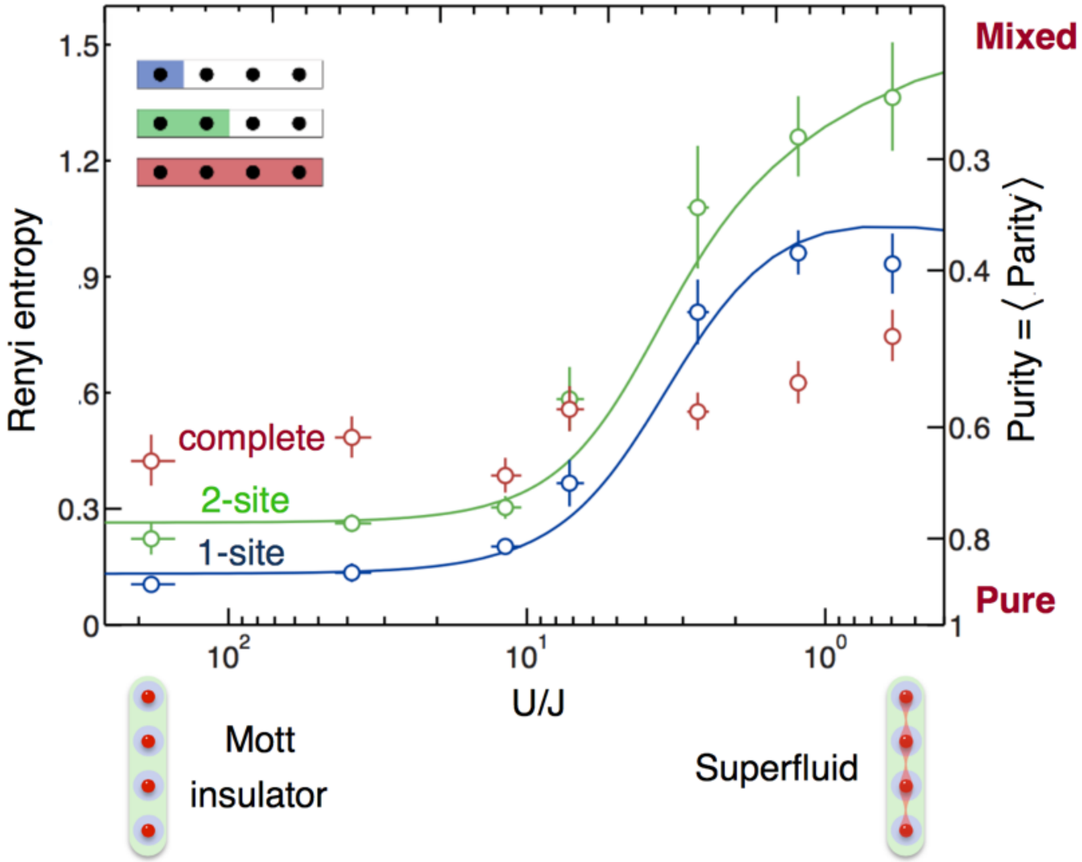
Reconstructed estimators

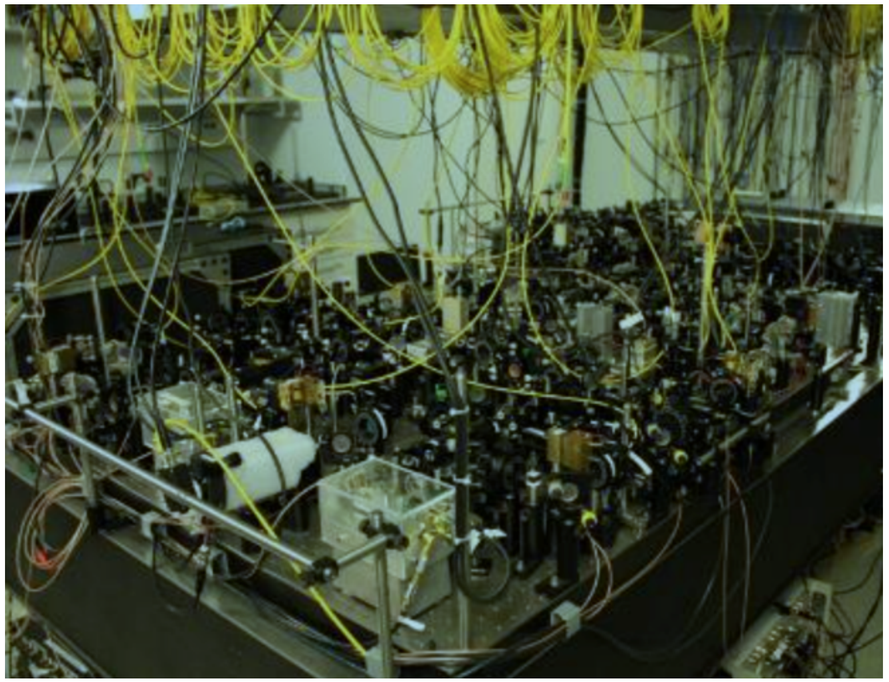
Torlai, Timar, van Nieuwenburg, Levine, Omran, Keesling, Bernien, Greiner, Vuletić, Lukin, RGM, Endres, arXiv:1904.08441

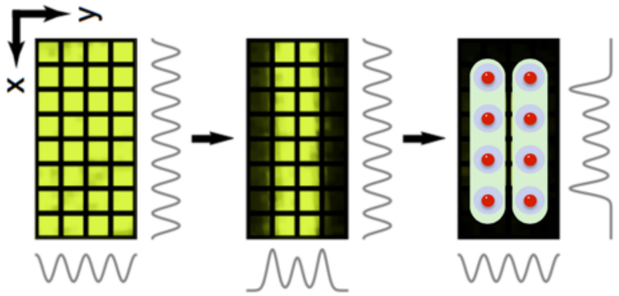


C.f. previous experiments:

Measuring entanglement entropy in a quantum many-body system
Rajibul Islam et. al Nature 528, 77–83 (2015)

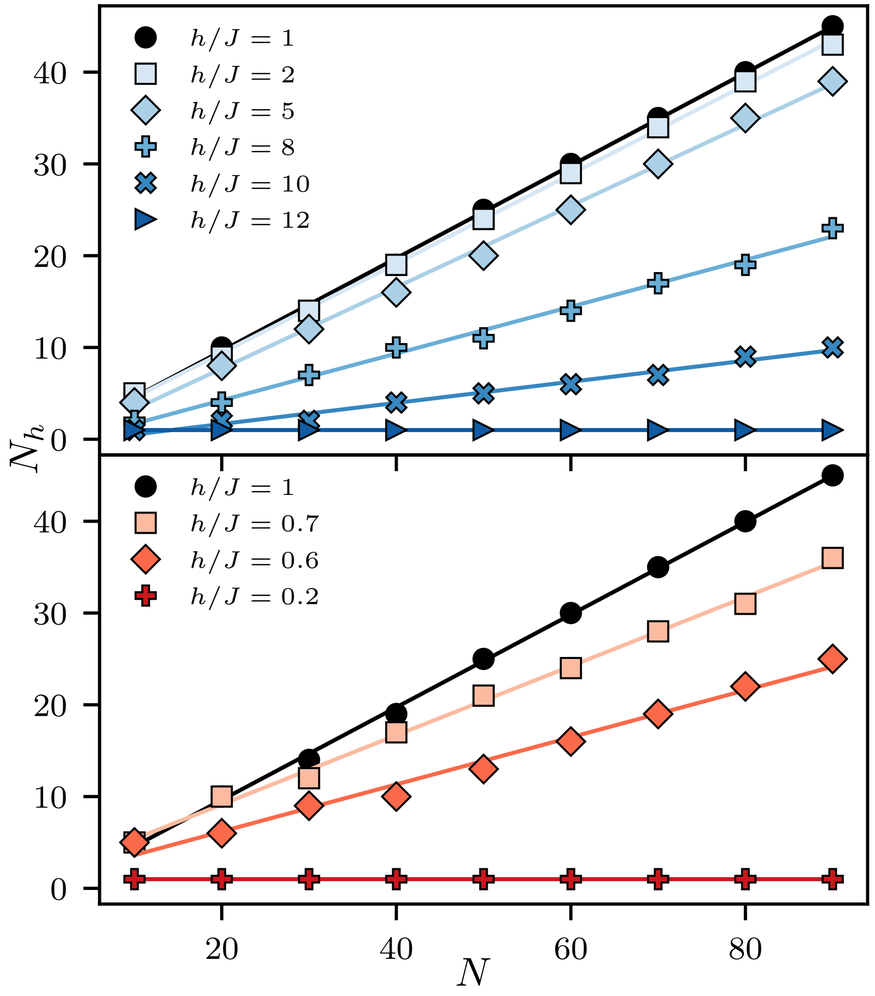




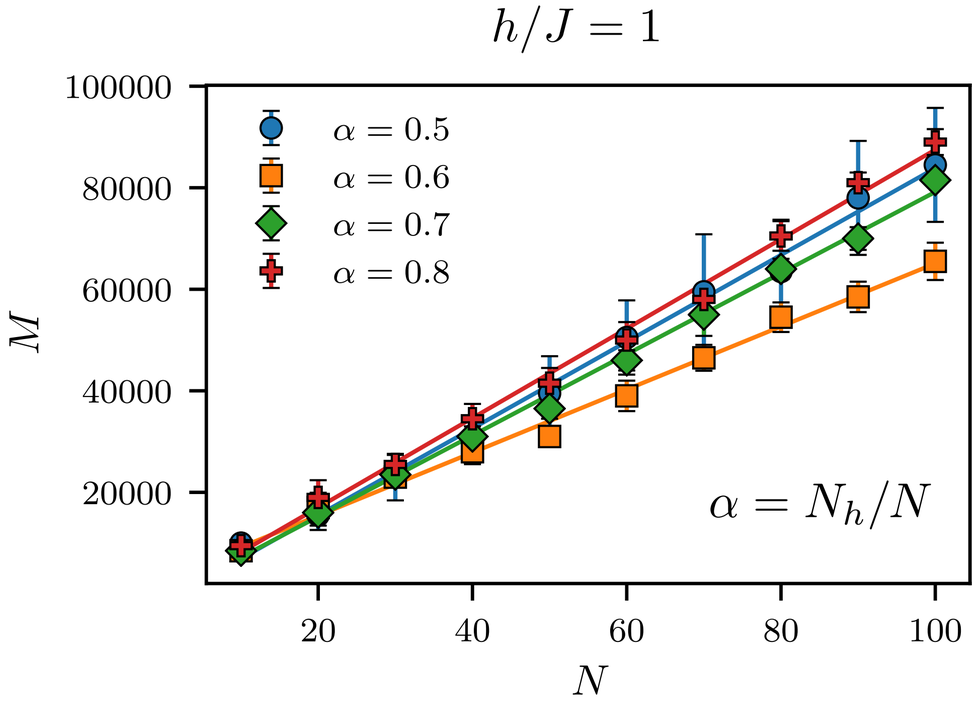


How does this scale?

Sehayek, Golubeva, Albergo, Kulchytskyy, Torlai, RGM
arXiv:1908.07532



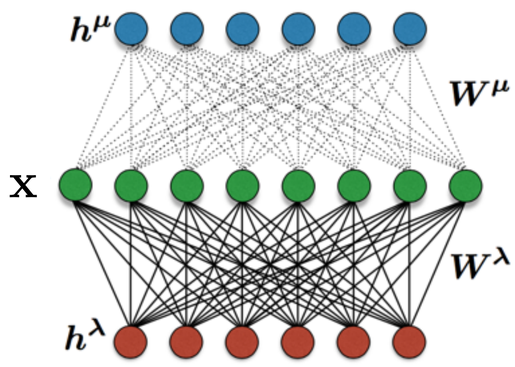
$$H = -\sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - \Gamma \sum_i \sigma_i^x$$



Learning complex wavefunctions

Torlai, Mazzola, Carrasquilla, Troyer, RGM, Carleo, Nature Physics 14, 447-450 (2018)

For a more generic wavefunction with amplitude and phase, represent both with hidden units



$$\psi_{\lambda,\mu}(\mathbf{x}) \propto \sqrt{p_\lambda(\mathbf{x})} e^{i\phi_\mu(\mathbf{x})}$$

Now, different bases are needed to estimate both the amplitude and phases of the target state.

$$\mathcal{L} = \sum_b^{N_b} \sum_{\mathbf{x}_b} \log |\psi_{\lambda,\mu}(\mathbf{x}_b)|^2$$

$N_b =$ number of bases

$\{X, X, Z, Z, \ldots\}, \{Z, X, X, Z, \ldots\}, \{Z, Z, X, X, \ldots\},$

state rotated into basis b with the appropriate unitary

From this, calculate $\nabla_\lambda \mathcal{L}$ and $\nabla_\mu \mathcal{L}$, use stochastic gradient descent, etc.

In practice, training is done in two stages: learning of amplitude first, then optimization of the phase parameters.

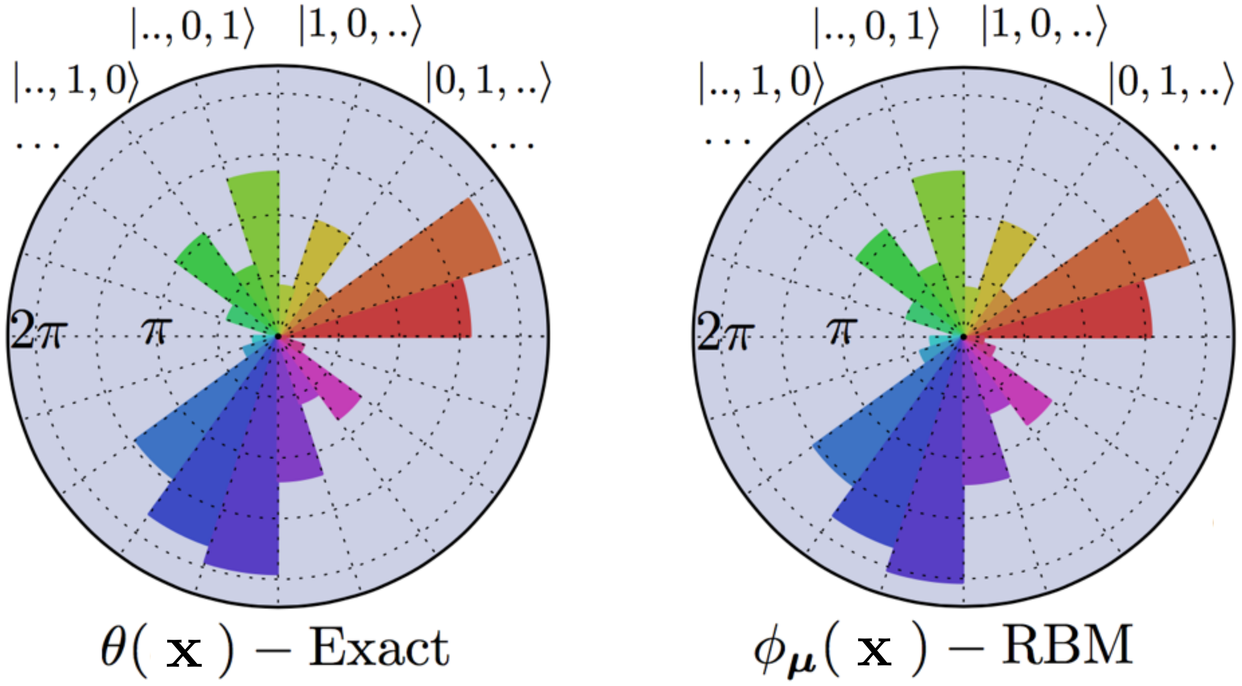
Learning complex wavefunctions

Torlai, *et al.*
Nature Physics 14, 447-450 (2018)

Example: an N-qubit “W” state, with local random phase shifts applied

$$|\tilde{\Psi}_W\rangle = \frac{1}{\sqrt{N}}\left(e^{i\theta_1}|100\ldots\rangle + e^{i\theta_2}|010\ldots\rangle + \cdots + e^{i\theta_N}|0\ldots01\rangle\right)$$

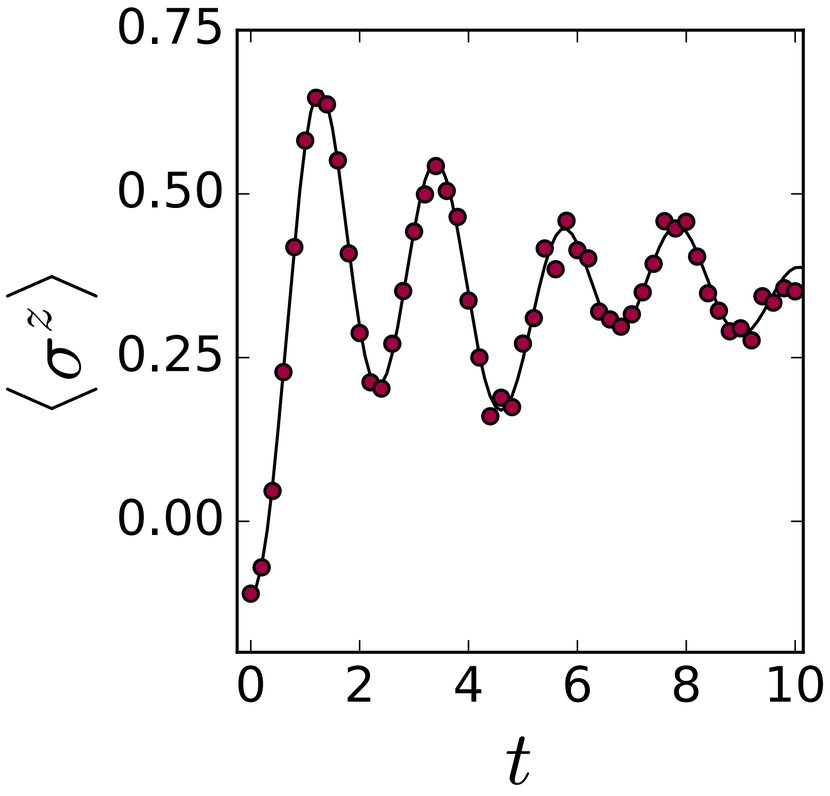
trained on a total of 2N -1 bases:

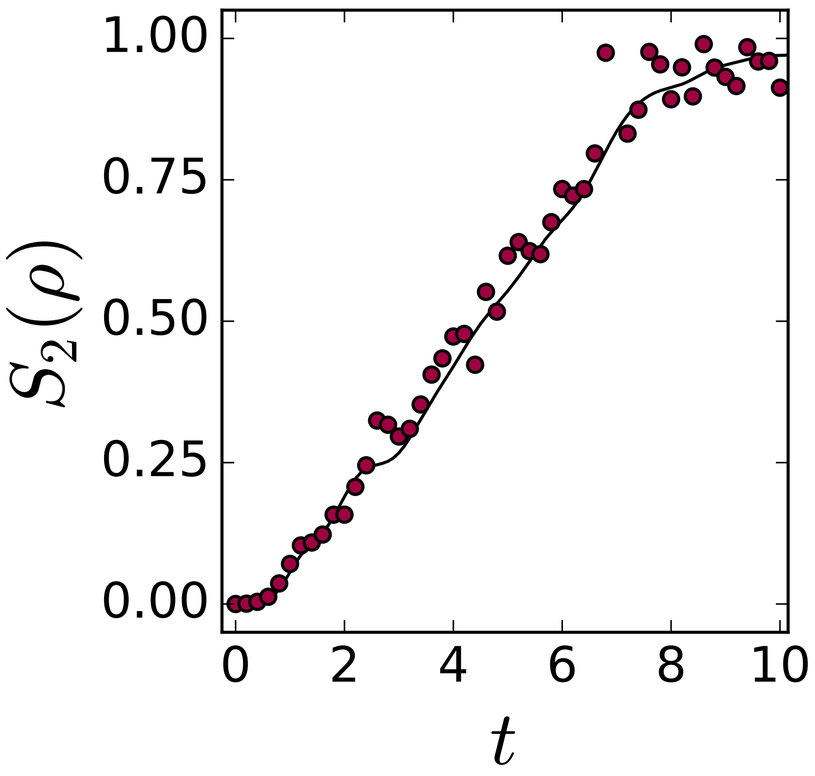


$N = 20$	qubits
99.7%	fidelity
2×10^5	measurements

Time evolution

G. Torlai





- Quenched with Rydberg Hamiltonian, from a Z2 state at t=0
- Measurements obtained in 2N + 1 bases form the training set

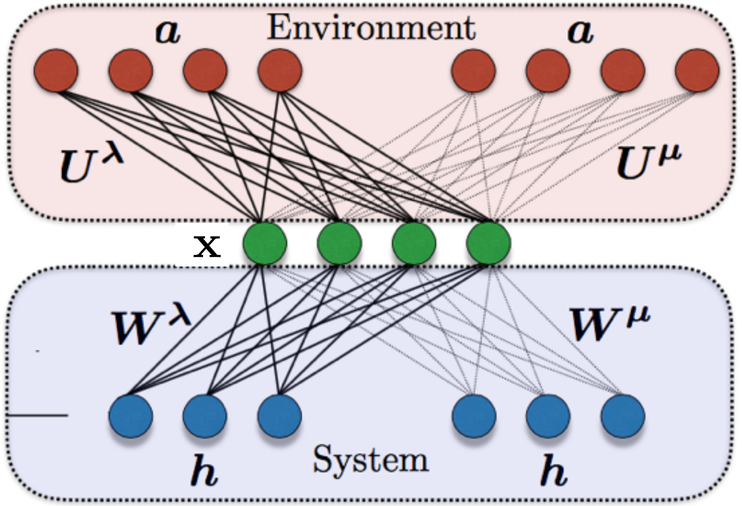
Learning density matrices

Torlai and RGM, Phys. Rev. Lett. 120, 240503 (2018)

Can extend our RBM to represent mixed states described by density matrices

$$\psi_{\lambda,\mu}(\mathbf{x},\mathbf{a}) \propto \sqrt{p_\lambda(\mathbf{x},\mathbf{a})}e^{i\phi_\mu(\mathbf{x},\mathbf{a})}$$

$$\rho_{\lambda,\mu}(\mathbf{x},\mathbf{x}') = \sum_{\mathbf{a}} \psi_{\lambda,\mu}(\mathbf{x},\mathbf{a})\psi^*_{\lambda,\mu}(\mathbf{x}',\mathbf{a})$$



Can introduce a factorization of the state in terms of a probability distribution and a set of tensors representing POVMs - allows for inversion of the Born rule.

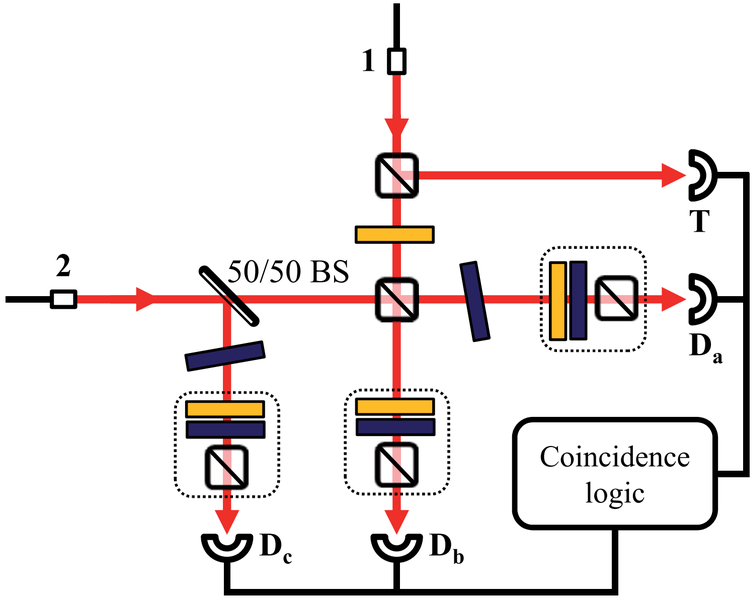
$$p(\mathbf{x}) = \mathrm{Tr}\rho M^{\mathbf{x}}$$

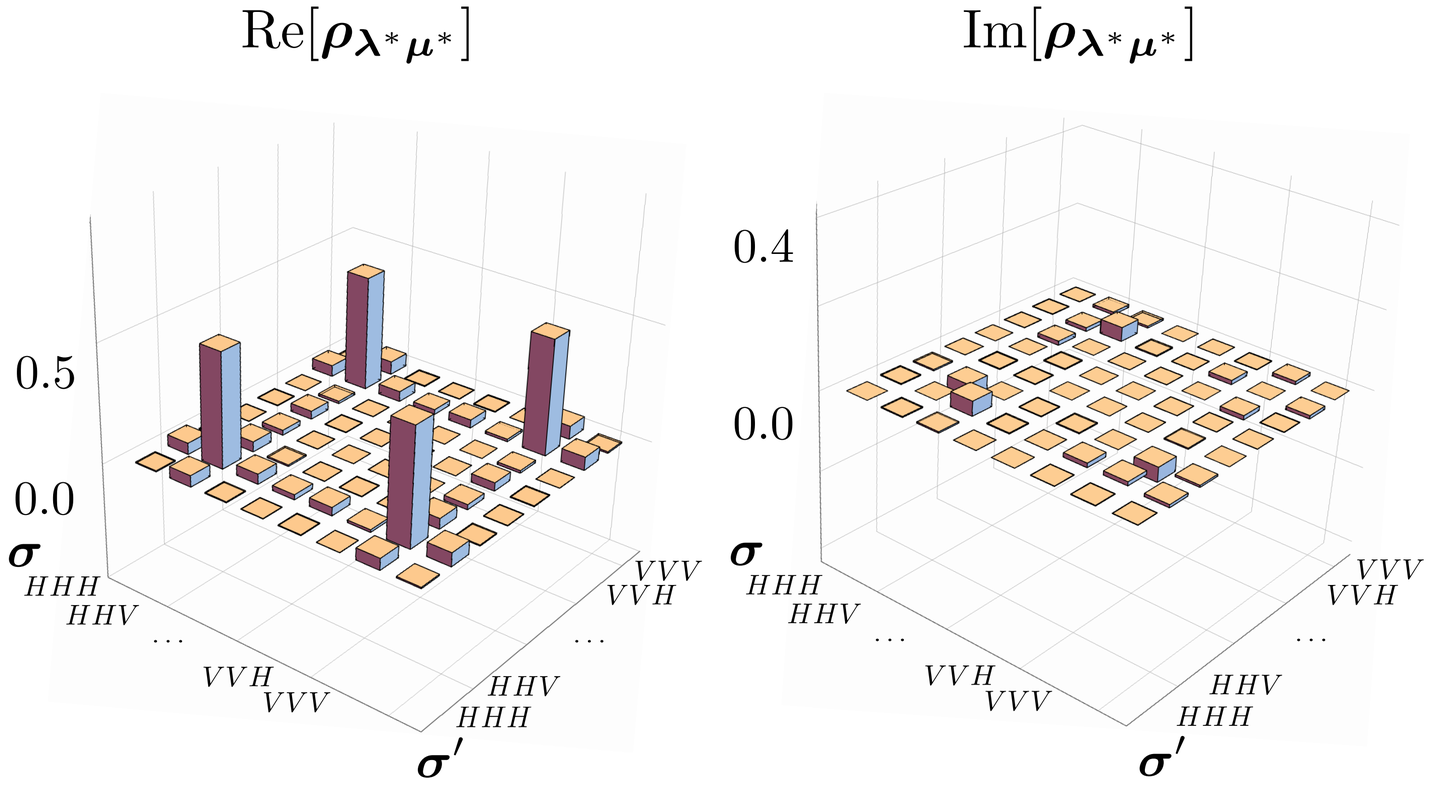
Carrasquilla, Torlai, RGM, Aolita,
Nature Machine Intelligence 1, 155–161 (2019)

RBMs, RNNs, VAEs, GANs, and more…

Experiment: entangled photons

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|HHV\rangle + |VVH\rangle\right)$$





K. Resch lab, IQC

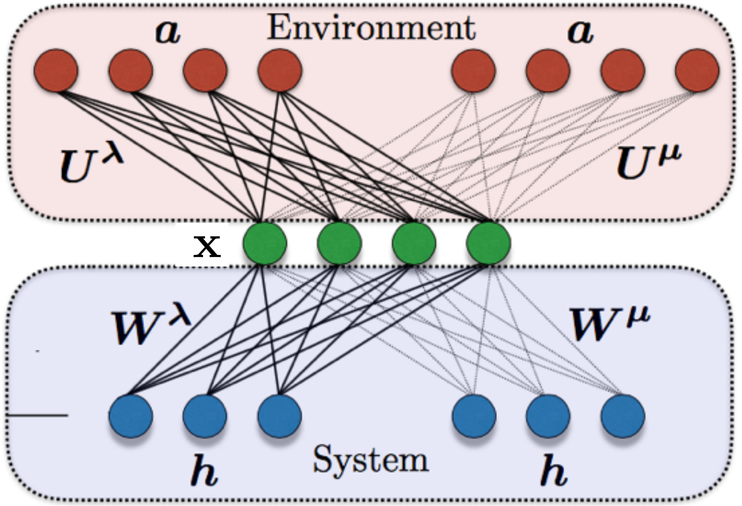
Learning density matrices

Torlai and RGM, Phys. Rev. Lett. 120, 240503 (2018)

Can extend our RBM to represent mixed states described by density matrices

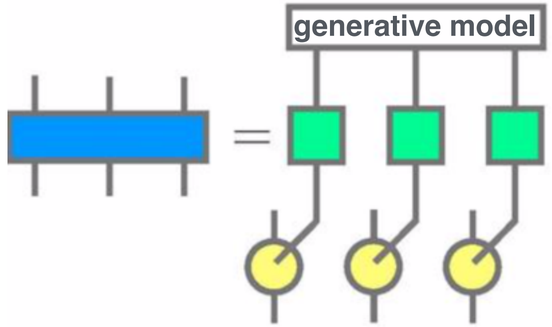
$$\psi_{\lambda,\mu}(\mathbf{x},\mathbf{a}) \propto \sqrt{p_\lambda(\mathbf{x},\mathbf{a})}e^{i\phi_\mu(\mathbf{x},\mathbf{a})}$$

$$\rho_{\lambda,\mu}(\mathbf{x},\mathbf{x}') = \sum_{\mathbf{a}} \psi_{\lambda,\mu}(\mathbf{x},\mathbf{a})\psi^*_{\lambda,\mu}(\mathbf{x}',\mathbf{a})$$



Can introduce a factorization of the state in terms of a probability distribution and a set of tensors representing POVMs - allows for inversion of the Born rule.

$$p(\mathbf{x}) = \mathrm{Tr}\rho M^{\mathbf{x}}$$



Carrasquilla, Torlai, RGM, Aolita,
Nature Machine Intelligence 1, 155–161 (2019)

RBMs, RNNs, VAEs, GANs, and more…

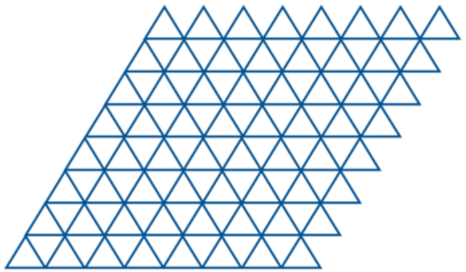
Learning density matrices

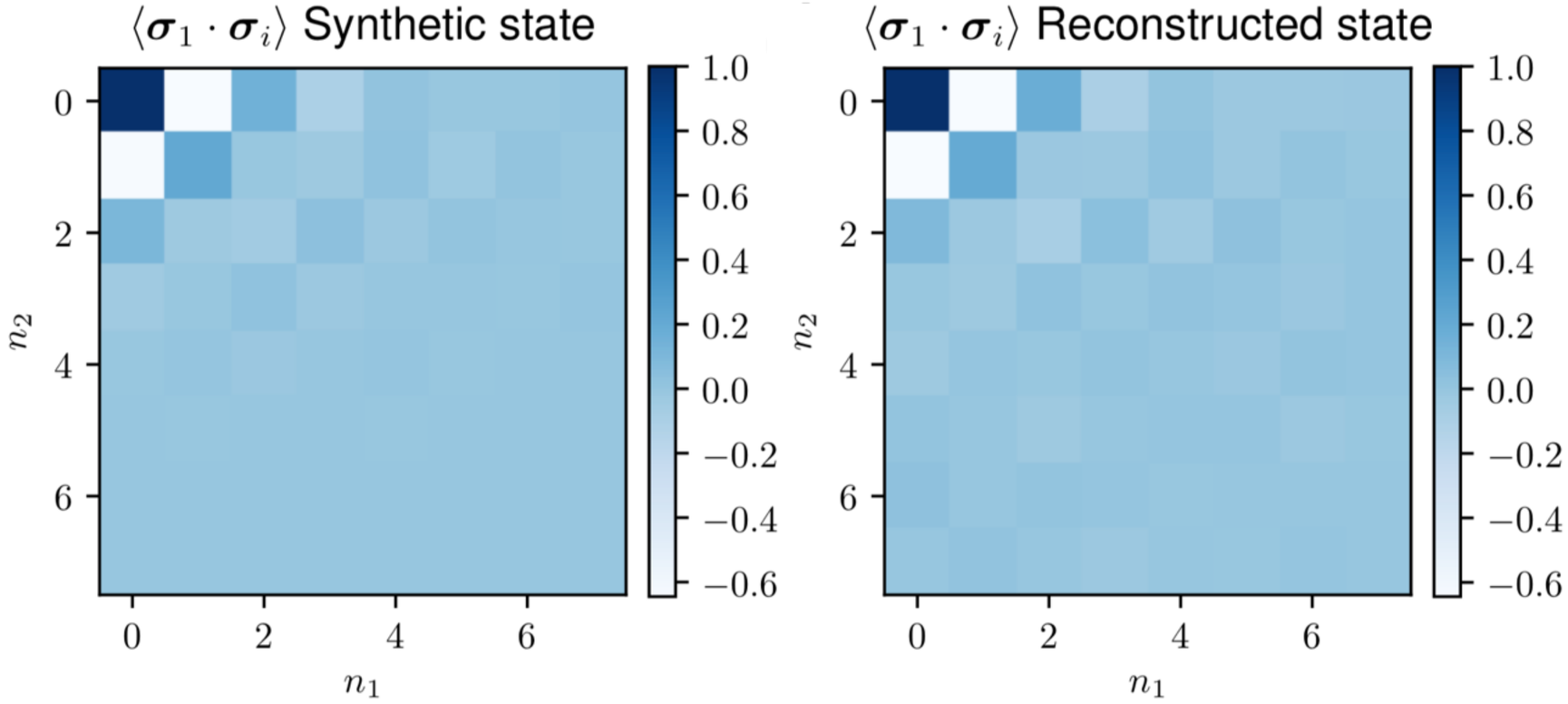
Carrasquilla, Torlai, RGM, Aolita,
Nature Machine Intelligence 1, 155–161 (2019)

- Correlation functions for the ground state of the triangular lattice Heisenberg antiferromagnet

$$H = \sum_{\langle i,j \rangle} \mathbf{S}_i \cdot \mathbf{S}_j$$

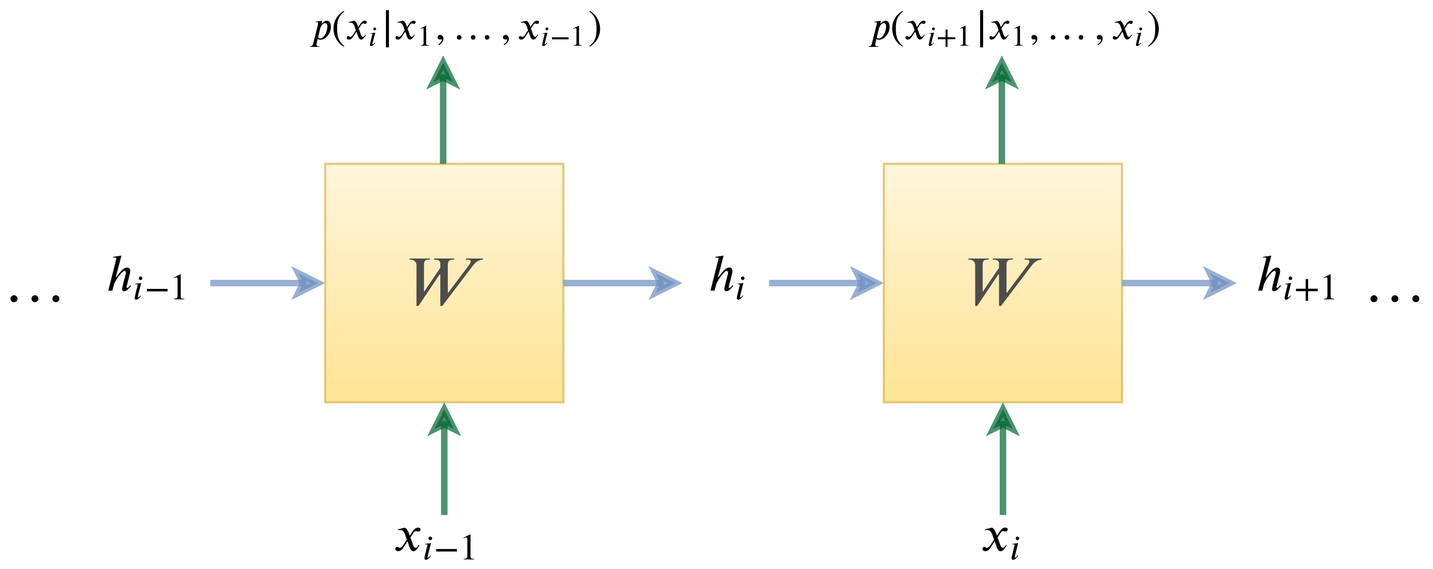
$$N = L \times L = 8 \times 8$$





The future: autoregressive models

Recurrent Neural Network



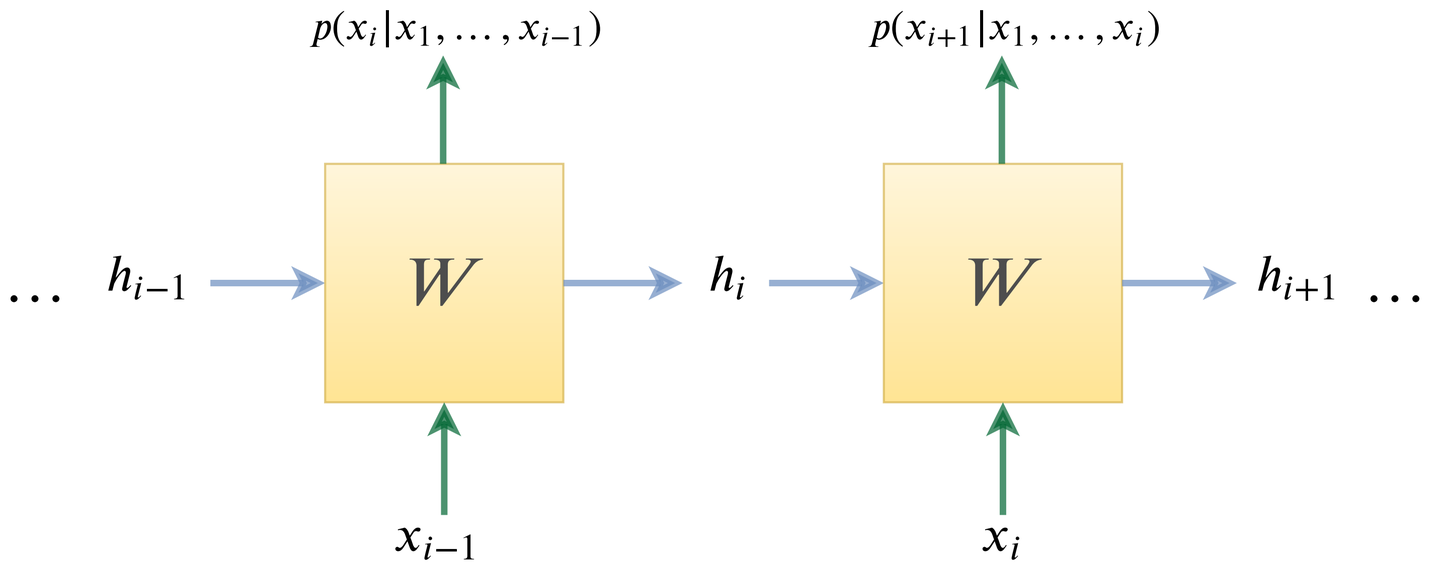
Typically used with word sequences: predict the probability of each word given the previous words.

Applications:

- Text generation (e.g. image descriptions)
- Other sequence generation (e.g. music)
- Machine translation
- Speech recognition

The future: autoregressive models

Recurrent Neural Network

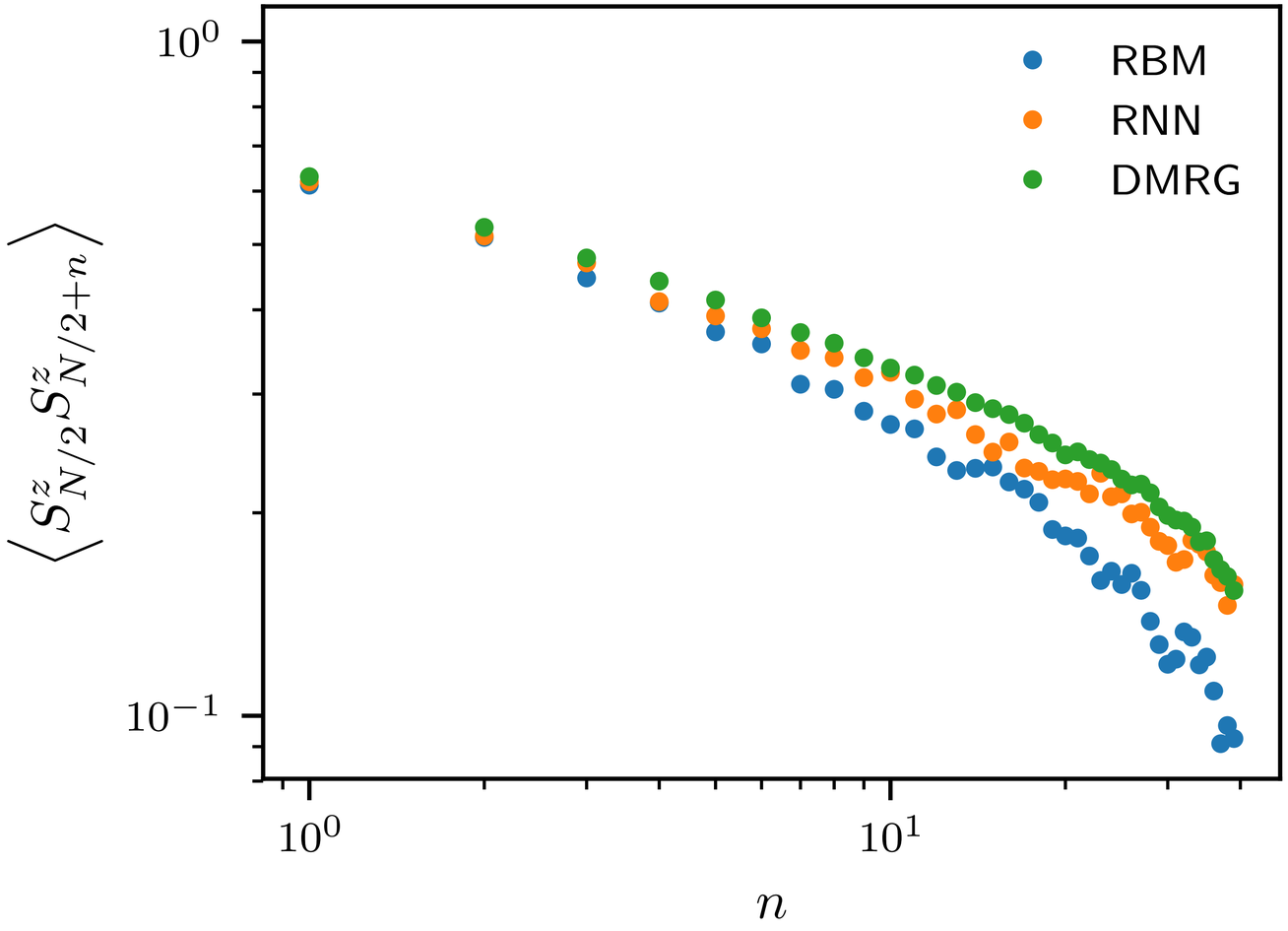


Can be used as generative models:

$$p_\lambda(x_1,\cdots x_N) = \prod_{i=1}^{N} p(x_i|x_1,\ldots,x_{i-1}) \quad = p(x_1)p(x_2|x_1)p(x_3|x_1,x_2)\cdots$$

Tractable: i.e. no need to evaluate a partition function. Can be sampled without autocorrelation

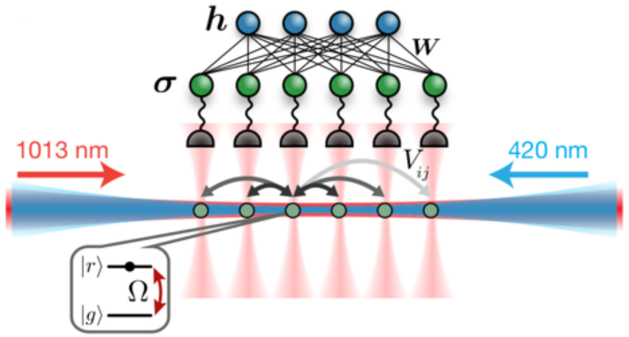
D. Sehayek



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

$$h/J = 1$$

Outlook:



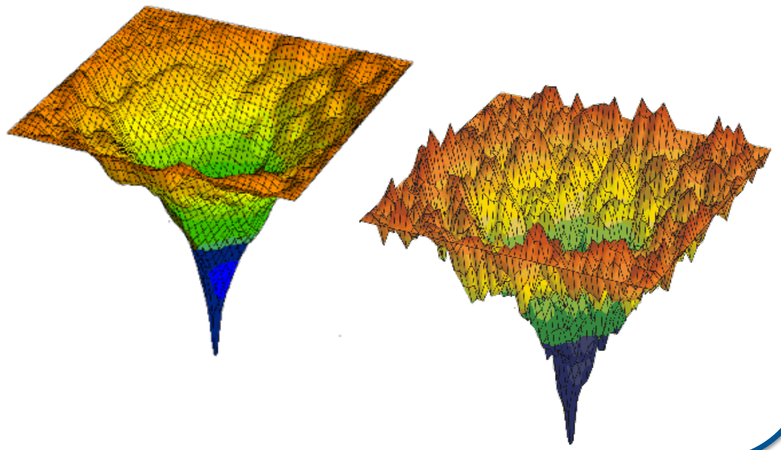
- ***Generative modelling*** is already being used together with a Rydberg atom quantum simulator

- For which quantum states is an ***efficient*** reconstruction possible with machine learning?

$n_h \propto \mathrm{poly}(N)$

$M \propto \mathrm{poly}(N)$

- What useful bases are experimentally accessible, and how does the basis choice affect ***scaling***?

$N_b \propto \mathrm{poly}(N)$

- How is learning from data ***different*** than optimizing neural network parameters variationally with knowledge of the Hamiltonian?