Using RNNs to learn a quantum many-body wavefunction

Mary Letey

Supervised by Professor Roger Melko and Schuyler Moss

Perimeter Institute Quantum Intelligence Lab

Quantum Many-Body Problems

- Emergent macroscopic behavior from microscopic interactions
- Typical example: Ising model

Phase transition between **disorder** (no magnetisation) and **order**

- Realistic hamiltonians – computations are hard

Applying ML

- Quantum State tomography
- Curse of dimensionality
- Efficiently extract physical quantities

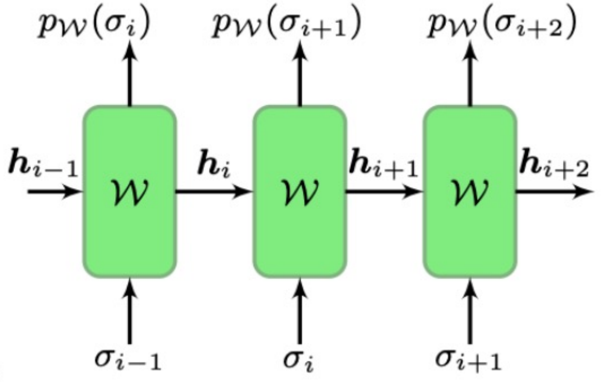
  + noisy experimental datasets
- Neural Networks: learn the underlying probability distribution?

Background Setup

- Array of Rydberg atoms near-criticality
- Ground state |0> or excited state |1>

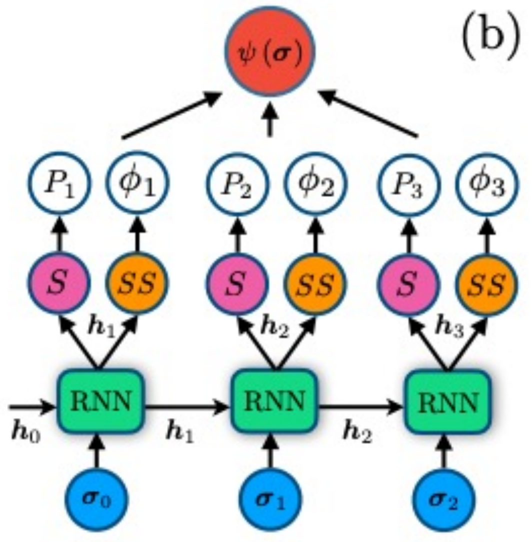
$$\hat{H} = -\frac{\Omega}{2}\sum_{i=1}^{N}\hat{\sigma}_i^x - \delta\sum_{i=1}^{N}\hat{n}_i + \sum_{i,j}V_{ij}\hat{n}_i\hat{n}_j$$

- Use RNN to approximate wavefunction



$$\Psi_{\mathcal{W}}(\boldsymbol{\sigma}) = \sqrt{\prod_i p_{\mathcal{W}}(\sigma_i)}$$

Why RNNs?



- Can also represent complex wavefunctions

$$|\Psi\rangle = \sum_{\boldsymbol{\sigma}} \exp(\mathrm{i}\phi(\boldsymbol{\sigma}))\sqrt{P(\boldsymbol{\sigma})}\,|\boldsymbol{\sigma}\rangle$$

$$\phi(\boldsymbol{\sigma}) \equiv \sum_{n=1}^{N} \phi_n$$

$$P(\boldsymbol{\sigma}) \equiv \Pi_{n=1}^{N} P_n$$

- Long-range correlations
- Autoregressive property

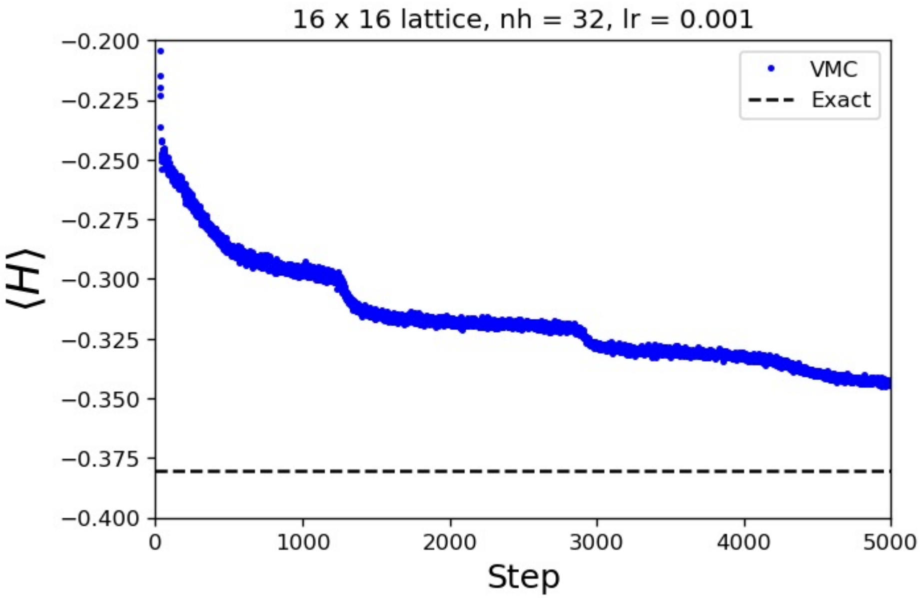
Loss Function: Hamiltonian Driven

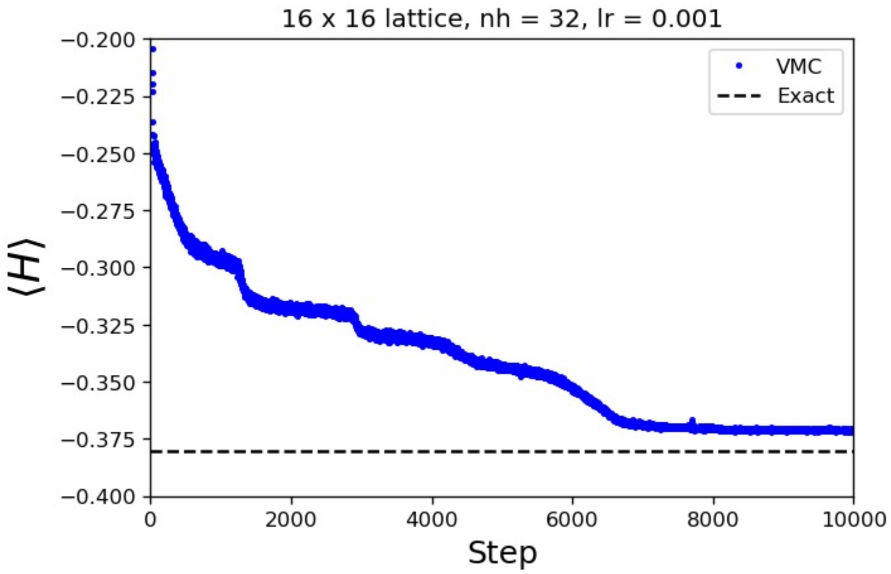
Adjust weights according to “energy” of spin configuration

$$H_{RNN} \approx \frac{1}{N_s} \sum_{\boldsymbol{\sigma} \sim p_{RNN}(\boldsymbol{\sigma};\mathcal{W})} H_{loc}(\boldsymbol{\sigma})$$

where $$H_{\mathrm{loc}}(\boldsymbol{\sigma}) = \frac{\langle \boldsymbol{\sigma} | \hat{H} | \Psi_{\mathcal{W}} \rangle}{\langle \boldsymbol{\sigma} | \Psi_{\mathcal{W}} \rangle}$$

Loss Function: Hamiltonian Driven

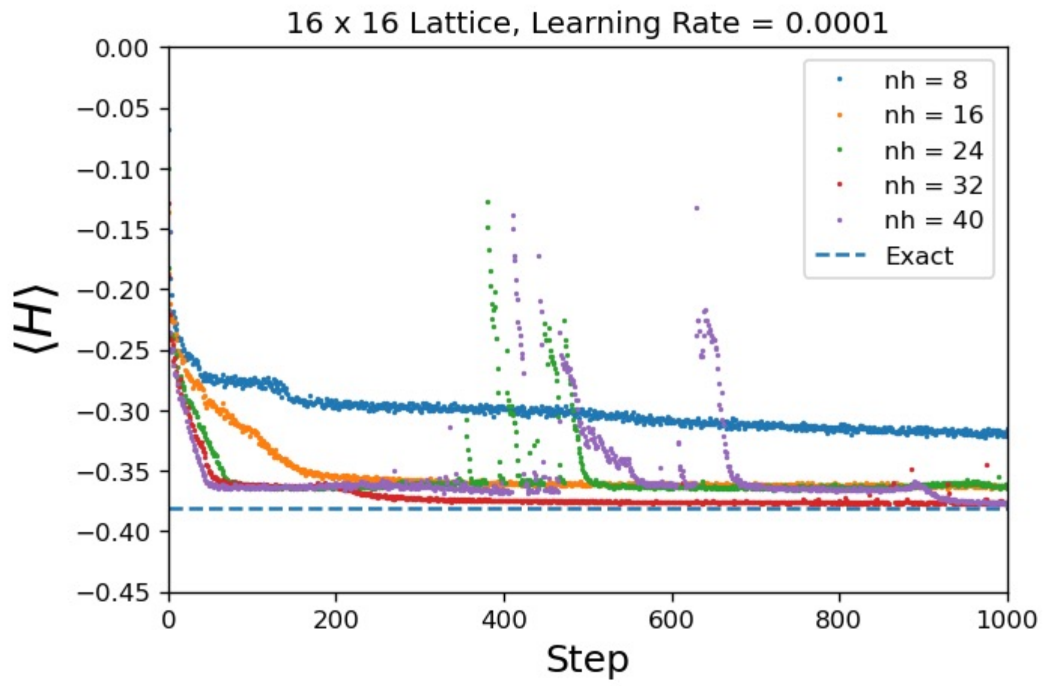




Loss Function: Data Driven

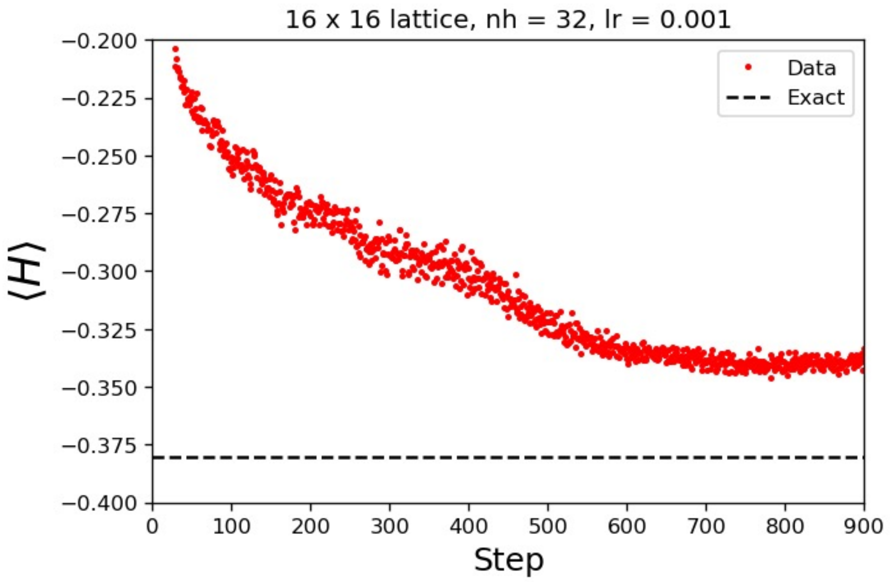
$$\mathcal{L}_{KL}(\mathcal{W}) = \sum_{\{\boldsymbol{\sigma}\}} p_{\mathcal{D}}(\boldsymbol{\sigma}) \log \frac{p_{\mathcal{D}}(\boldsymbol{\sigma})}{p_{RNN}(\boldsymbol{\sigma}; \mathcal{W})}$$

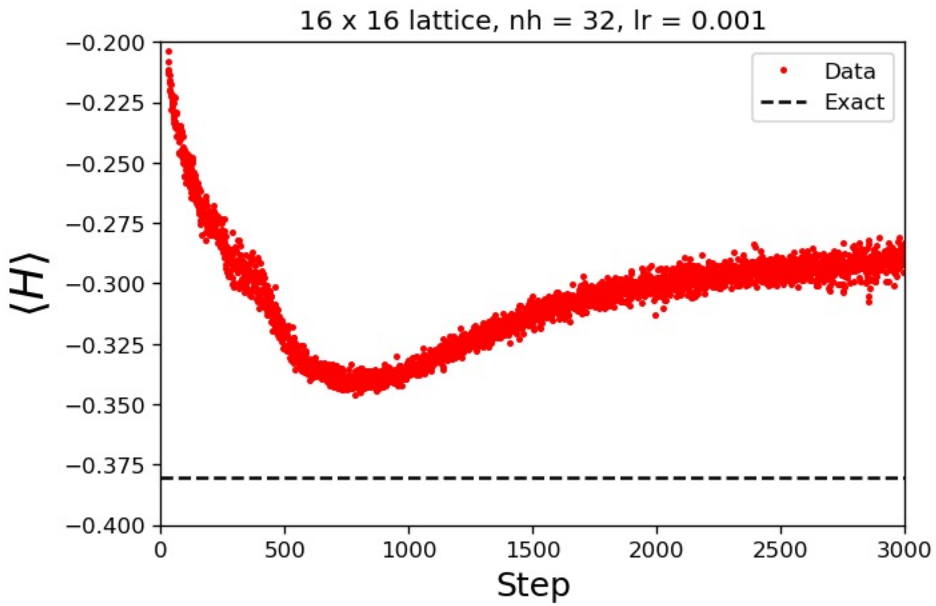
Performs well given enough data



Loss Function: Data Driven

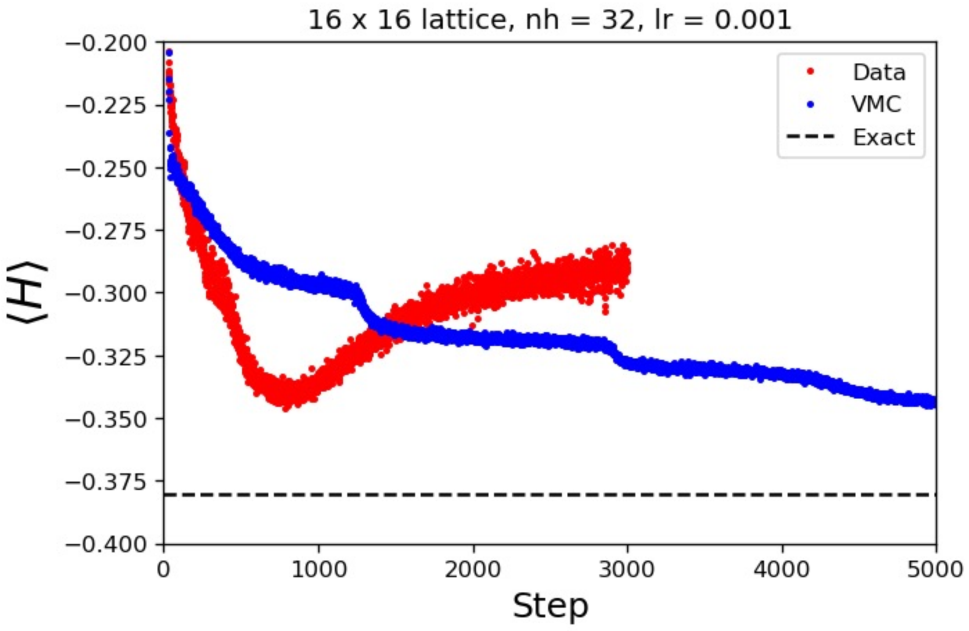
- However, real measurement data is expensive to generate.
- Likely sample sizes are only 1000

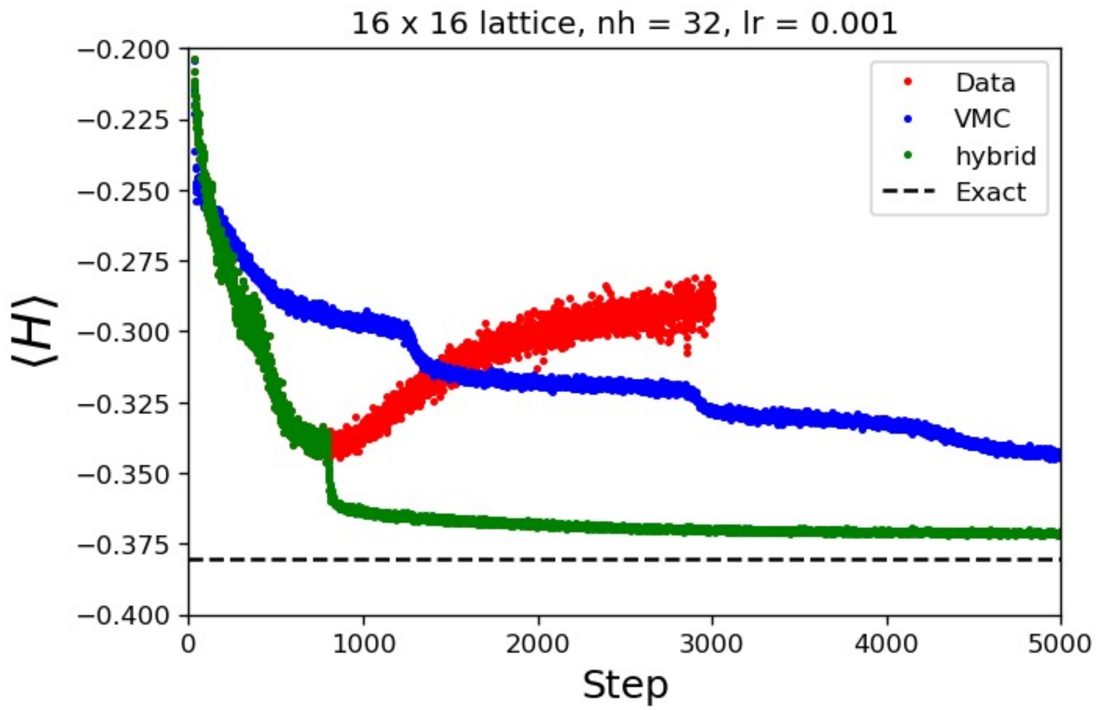




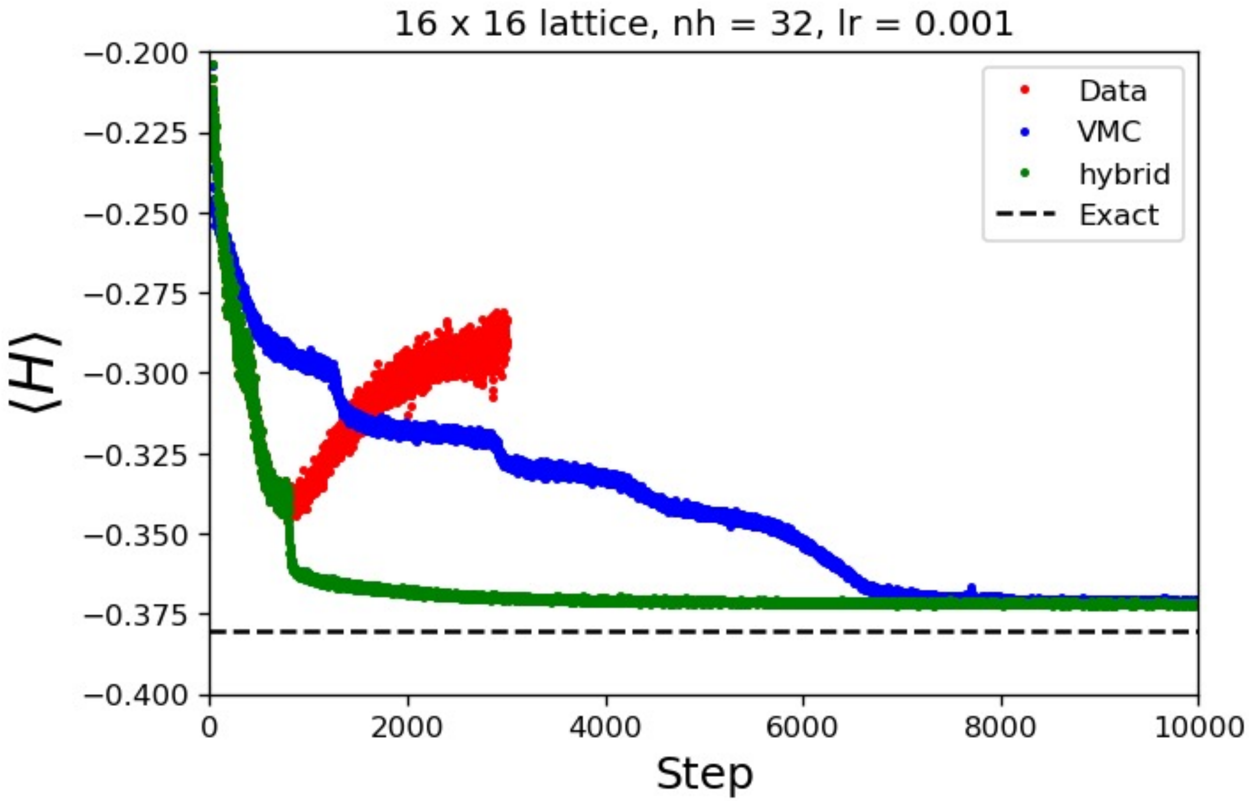
Hybrid training

Initialise on KL loss, then train variationally using Hamiltonian loss





Hybrid Training



Much better convergence time

Final results agree

Noisy Data

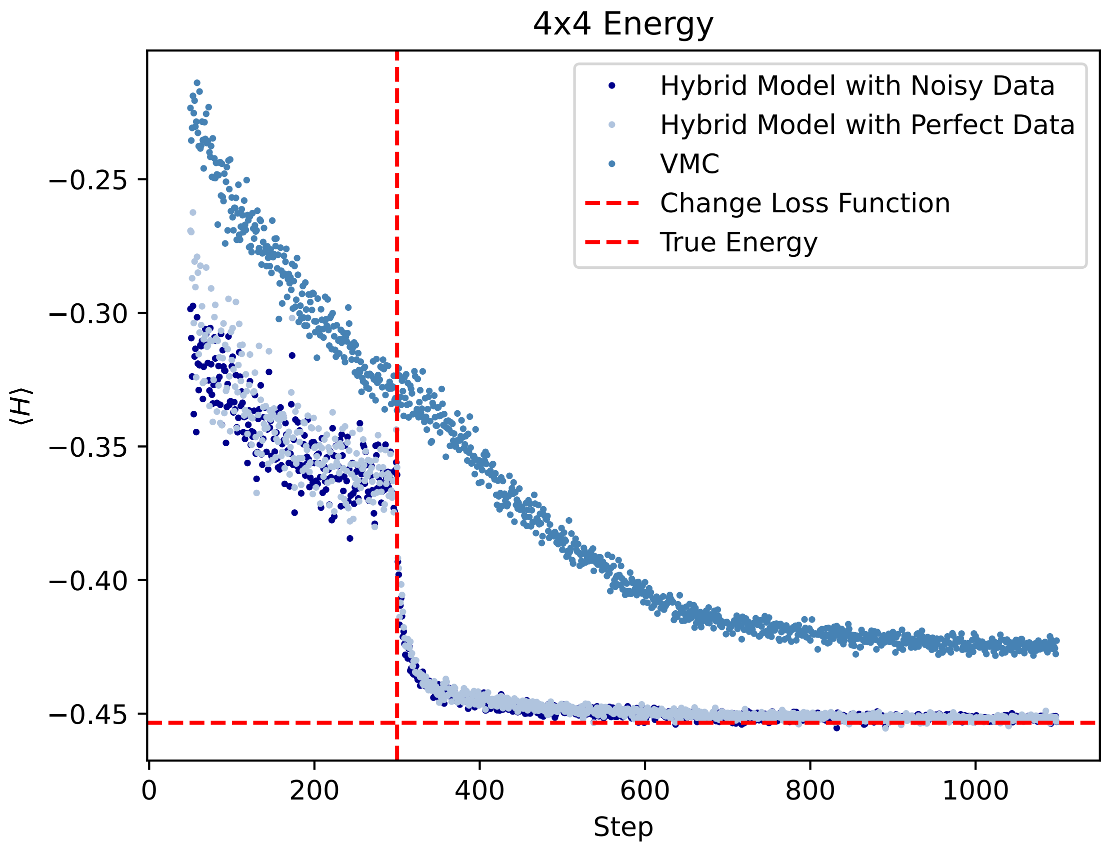
Real-world measurement data will have some noise.

$$p(1|0) \sim 1\%,\ p(0|1) \sim 4\%$$

How does this affect convergence?

How does this affect accuracy?

Preliminary Results – 4x4 lattice



Extensions

- Torlai et. al. Integrating Neural Networks with a Quantum Simulator for State Reconstruction
  - Uses RBM with a noise layer
  - Our implementation: encoder-decoder mechanism for learning error distribution and de-noising
- Better loss schedule?
- What value does data provide? What is happening during Hamiltonian training?