Solving Quantum Many-Body Problems with DNN

Han Pu

Rice University

Phys. & Astro.

IPAM Workshop “From Passive to Active: Generative and Reinforcement Learning with Physics”， Sept., 2019

Acknowledgment

Li Yang

Wenjun Hu

Ankit Patel

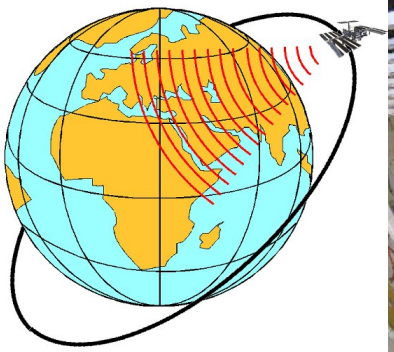
Ultracold atomic physics

Precision measurement:
Measurement of physical constants, Testing fundamental theories,…

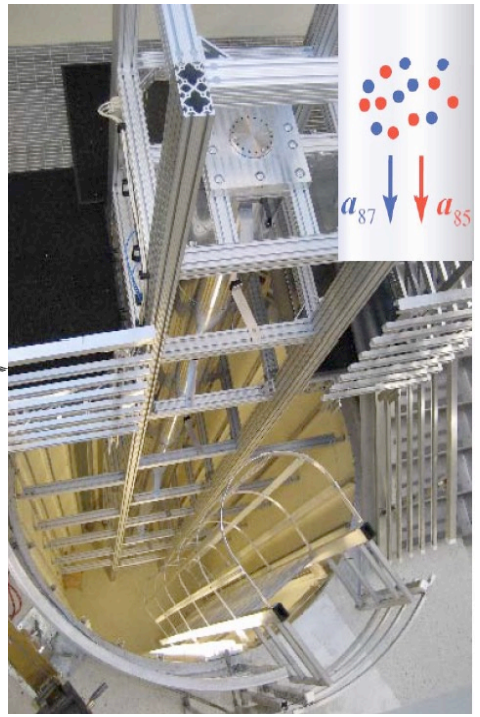
Practical applications in industry, defense, space …

Mine-exploration

Navigation/GPS



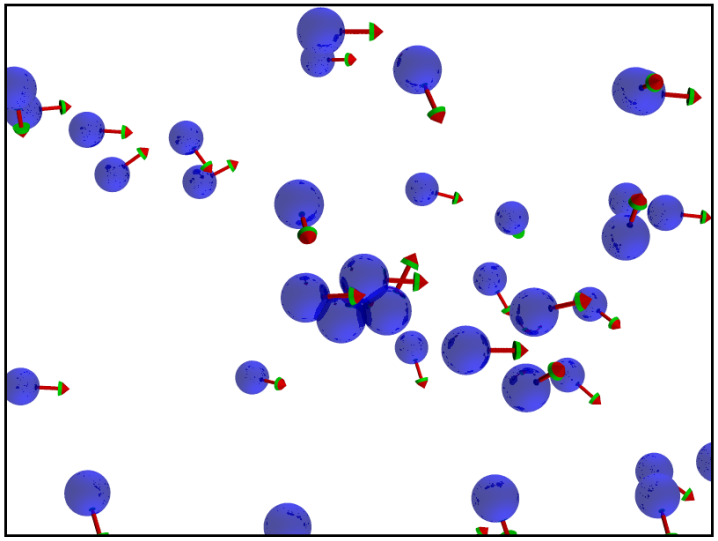
Space-based
Atomic clock



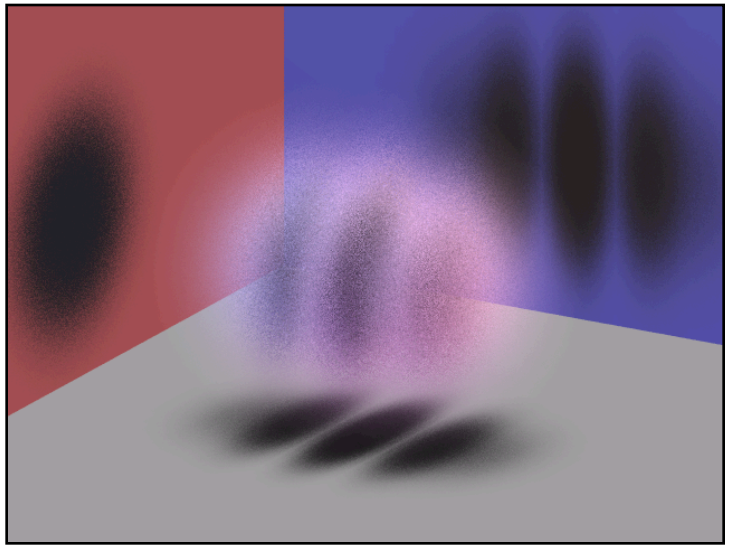
10-meter drop tower
(Kasevich)

Ultracold atomic physics

Coherent Quantum Phenomena



High Temperature:
Random thermal motion dominates



Low Temperature:
Underlying quantum behavior revealed

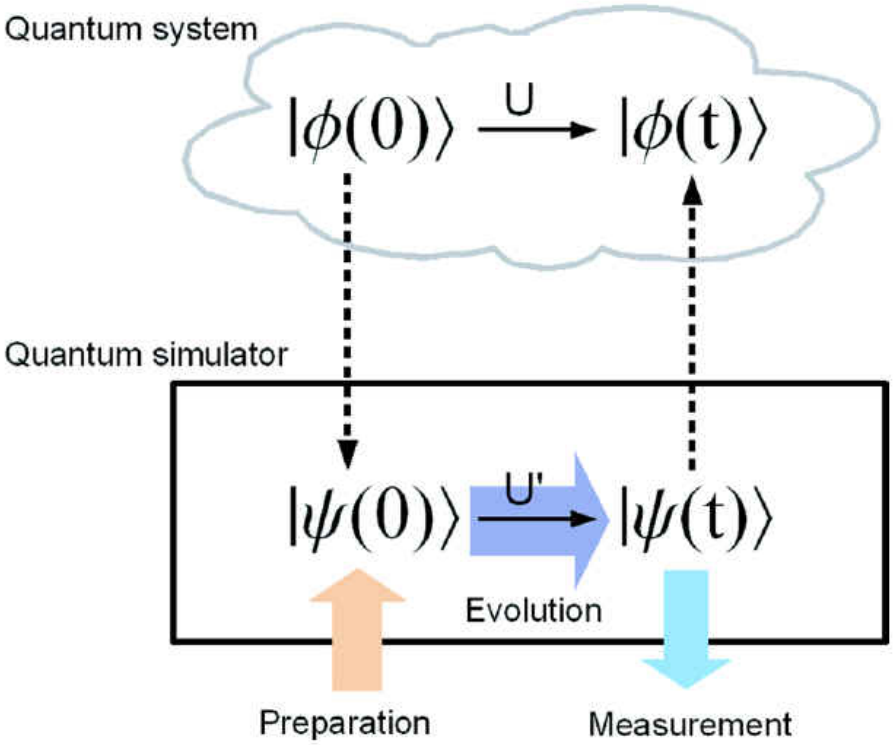
Curse of Hilbert Space

Size of the Hilbert Space for a general quantum many-body system increases exponentially as the number of particles grows.

"Nature isn't classical, and if you want to make a simulation of Nature, you'd better make it quantum mechanical, and it's a wonderful problem, because it doesn't look so easy." (Richard Feynman)

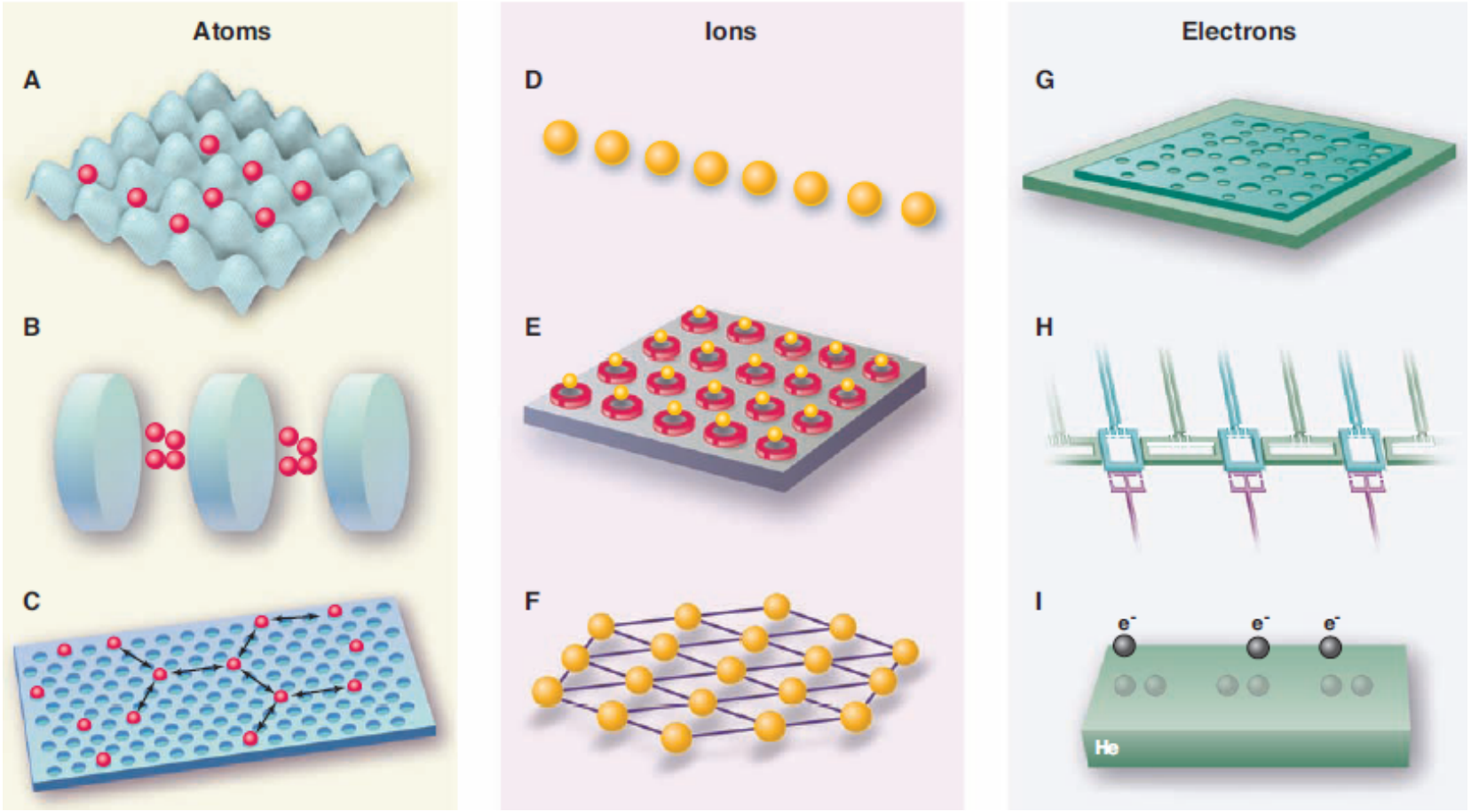
Quantum Simulator

The machine would have the capacity to contain an exponentially large amount of information without using an exponentially large amount of physical resources, thus making it a natural tool to perform quantum simulation.



Quantum Simulator

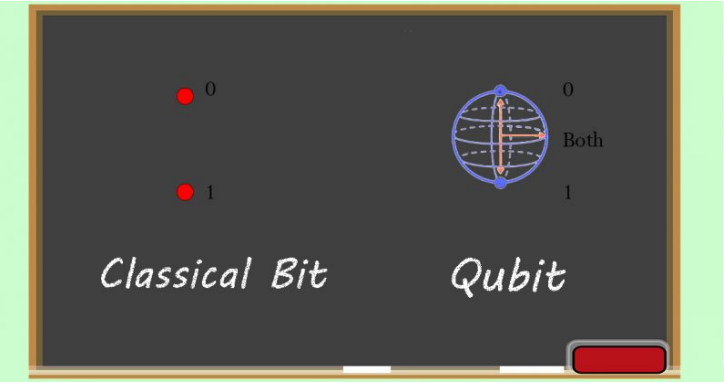
Physical quantum simulators



Iulia Buluta and Franco Nori, Science **326**,108 (2009)

Quantum Simulator

Digital quantum simulators --- quantum computers



Still a long way from reality …

Simulating quantum system on classical computer

In principle, fully characterizing a quantum system requires an exponentially large number of parameters. It would be useful if many-body states could be represented in such a way that some physical quantities could be calculated in a more efficient way.

Matrix product states
Projected entangled-pair states
Tensor product network states
……

Neural networks

This new class of algorithms makes it possible to simulate certain quantum systems which would have not been possible with traditional numerical methods.

Moreover, these methods can be combined with Monte Carlo techniques.

RBM for quantum many-body physics

Variational Monte Carlo

+

Restricted Boltzmann Machine

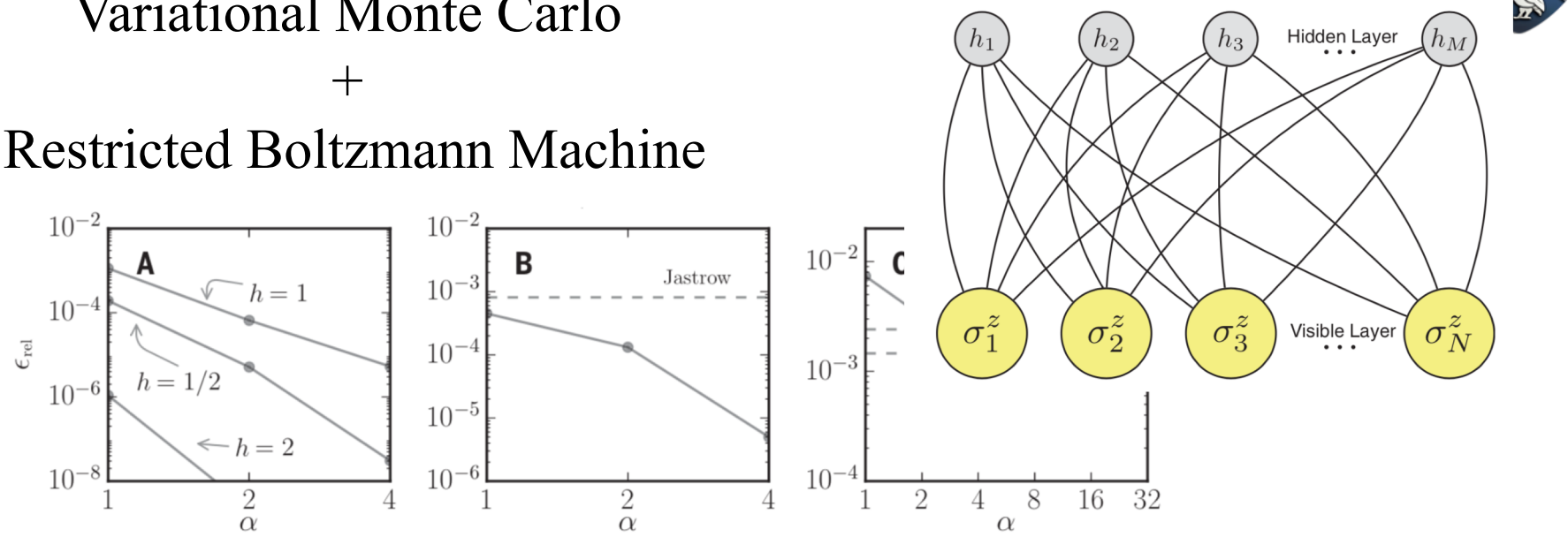
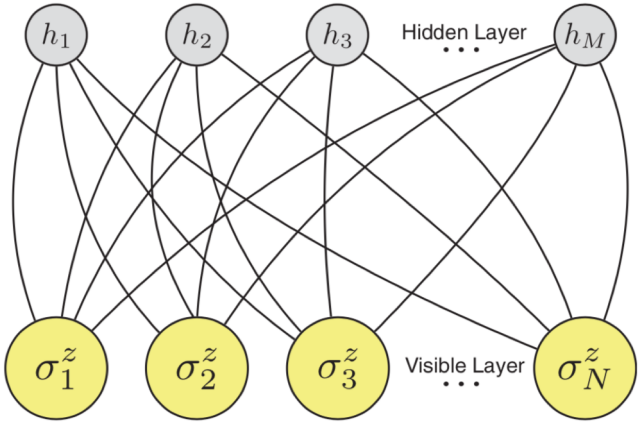


Fig. 3. Finding the many-body ground-state energy with neural-network quantum states (NQS). The error of the NQS ground-state energy relative to the exact value is shown for several test cases. Arbitrary precision on the ground-state energy can be obtained upon increasing the hidden-unit density α. (**A**) Accuracy for the 1D TFI model, at a few values of the field strength *h* and for an 80-spin chain with periodic boundary conditions (PBCs). Points below 10^{-8} are not shown to enhance readability. (**B**) Accuracy for the 1D AFH model, for an 80-spin chain with PBCs, compared with the Jastrow ansatz (horizontal dashed line). (**C**) Accuracy for the AFH model on a 10-by-10 square lattice with PBCs, compared with the precision obtained by EPS [upper dashed line (*35*)] and PEPS [lower dashed line (*36*)]. For all cases considered here, the NQS approach reaches MPS-grade accuracies in one dimension and systematically improves the best known variational states for 2D finite lattice systems.

Ising/Heisenberg

G. Carleo, M. Troyer,
Solving the Quantum Many-Body Problem with Artificial Neural Networks,
Science 355, 602 (2017)

RBM for quantum many-body physics



$$E\left(\vec{v},\vec{h}\right) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_{i,j} W_{ij} v_i h_j$$

$$P\left(\vec{v},\vec{h}\right) = \frac{1}{Z}\exp\left[-E\left(\vec{v},\vec{h}\right)\right]$$

$$\Psi\left(\vec{v}\right) = e^{-\sum_i a_i v_i} \prod_j 2\cosh\left(\sum_i W_{ij} v_i + b_j\right)$$

$$\ln\Psi\left(\vec{v}\right) = -\sum_i a_i v_i + \sum_j \sigma\left(\sum_i W_{ij} v_i + b_j\right)$$

$$\sigma(x) = \ln\left[2\cosh(x)\right]$$

From shallow to deep neural network

PHYSICAL REVIEW LETTERS **122,** 065301 (2019)

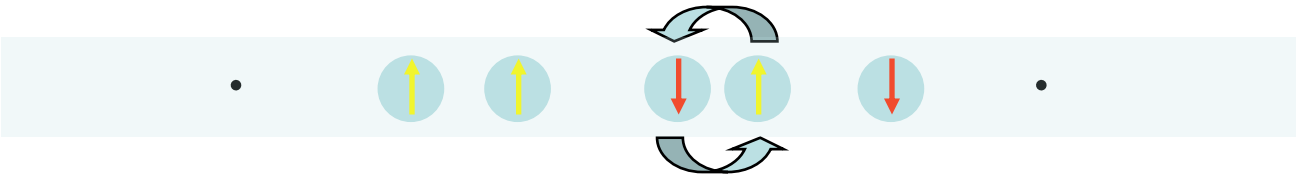
Quantum Entanglement in Deep Learning Architectures

Yoav Levine,[1,*] Or Sharir,[1,†] Nadav Cohen,[2,‡] and Amnon Shashua[1,§]
[1]*The Hebrew University of Jerusalem, 9190401 Israel*
[2]*School of Mathematics, Institute for Advanced Study, Princeton, New Jersey 08540, USA*

Modern deep learning has enabled unprecedented achievements in various domains. Nonetheless, employment of machine learning for wave function representations is focused on more traditional architectures such as restricted Boltzmann machines (RBMs) and fully connected neural networks. In this Letter, we establish that contemporary deep learning architectures, in the form of deep convolutional and recurrent networks, can efficiently represent highly entangled quantum systems. By constructing tensor network equivalents of these architectures, we identify an inherent reuse of information in the network operation as a key trait which distinguishes them from standard tensor network-based representations, and which enhances their entanglement capacity. Our results show that such architectures can support volume-law entanglement scaling, polynomially more efficiently than presently employed RBMs. Thus, beyond a quantification of the entanglement capacity of leading deep learning architectures, our analysis formally motivates a shift of trending neural-network-based wave function representations closer to the state-of-the-art in machine learning.

1D SU(*N*) spin chain

$$H = \sum_{i=1}^{N_{\text{site}}} \mathcal{E}_{i,i+1}$$

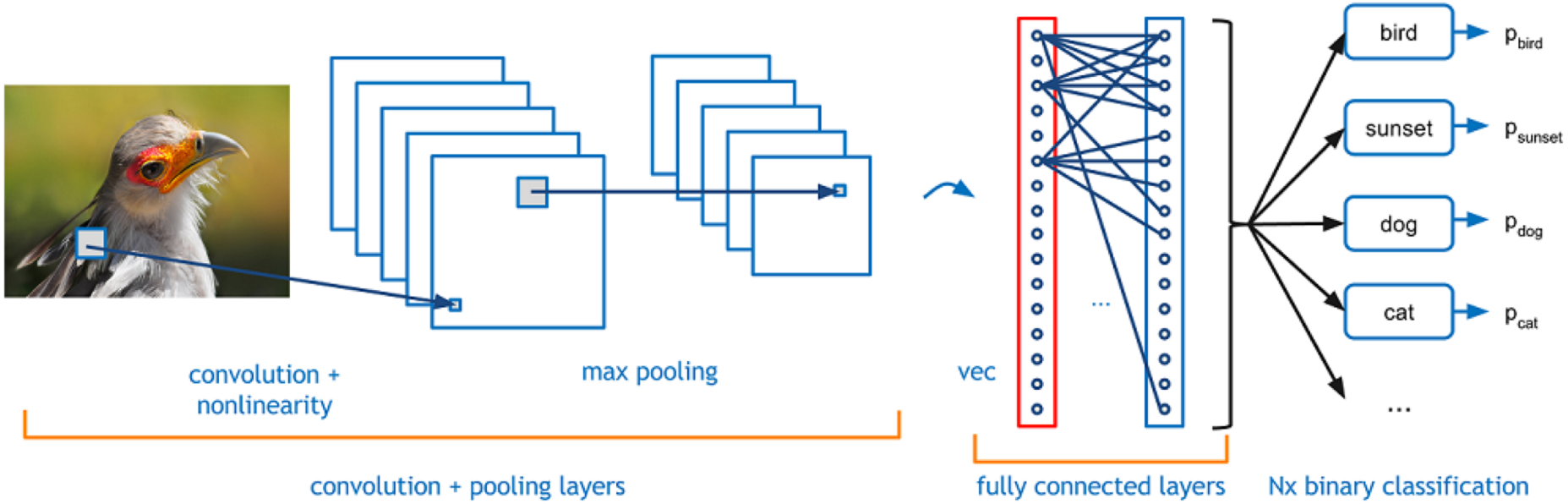


One spin at one site.
Each spin can take N possible orientations (values).
N controls the complexity of the system.

An important model for quantum magnetism.
Effective model for a system of 1D strongly interacting particles.

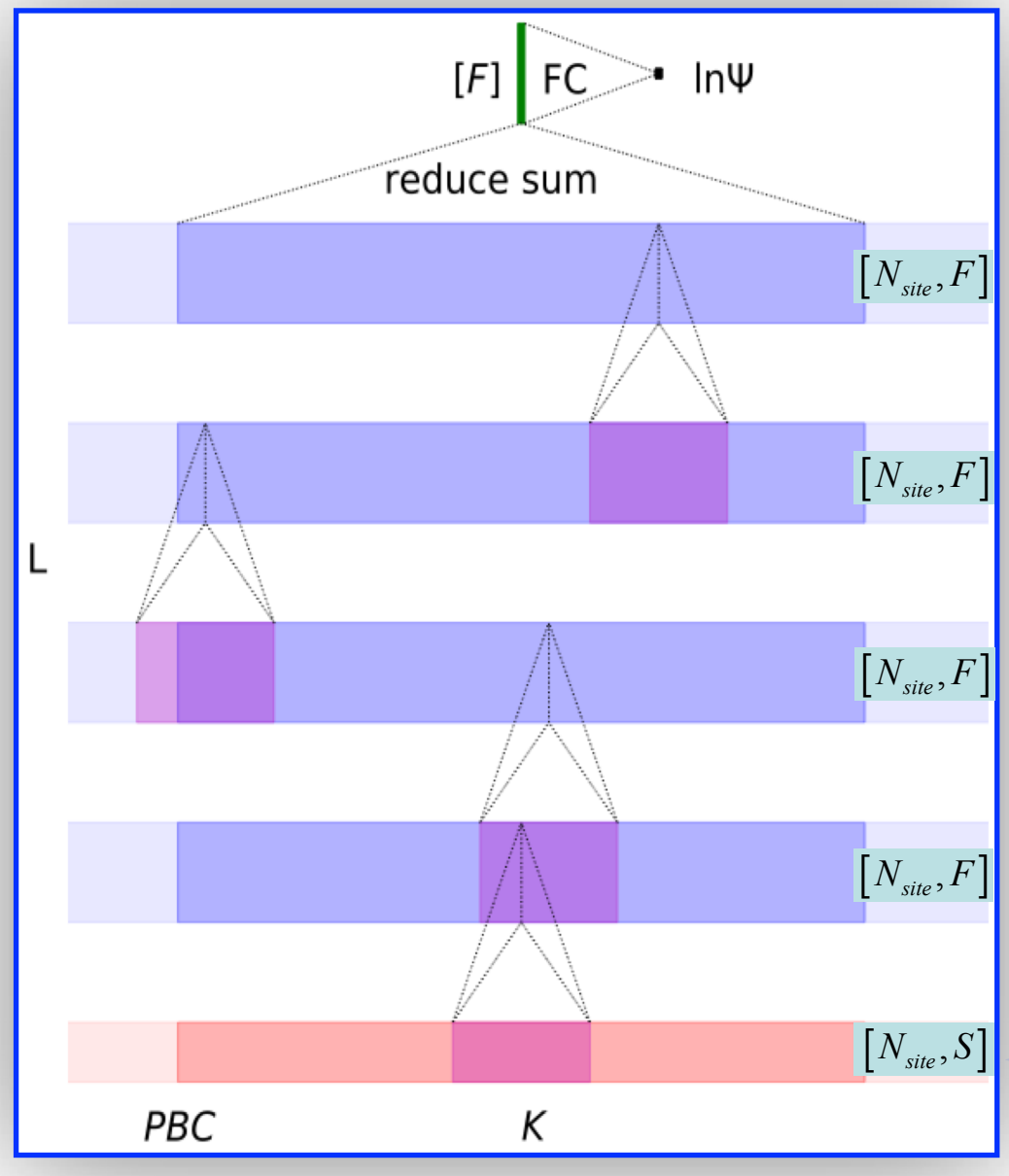
Exact solvable using Bethe Ansatz method [Sutherland, PRB **12**, 3795 (1975)].

Convolutional Neural Network



CNN capable of capturing local features.
Suitable for physical models with short-range interactions.

CNN for quantum many-body physics



- Fully Convolutional
- No Pooling, stride = 1
- Real-valued
- Output $\ln\Psi_s$

K : kernel size

F : number of filters

L : number of layers

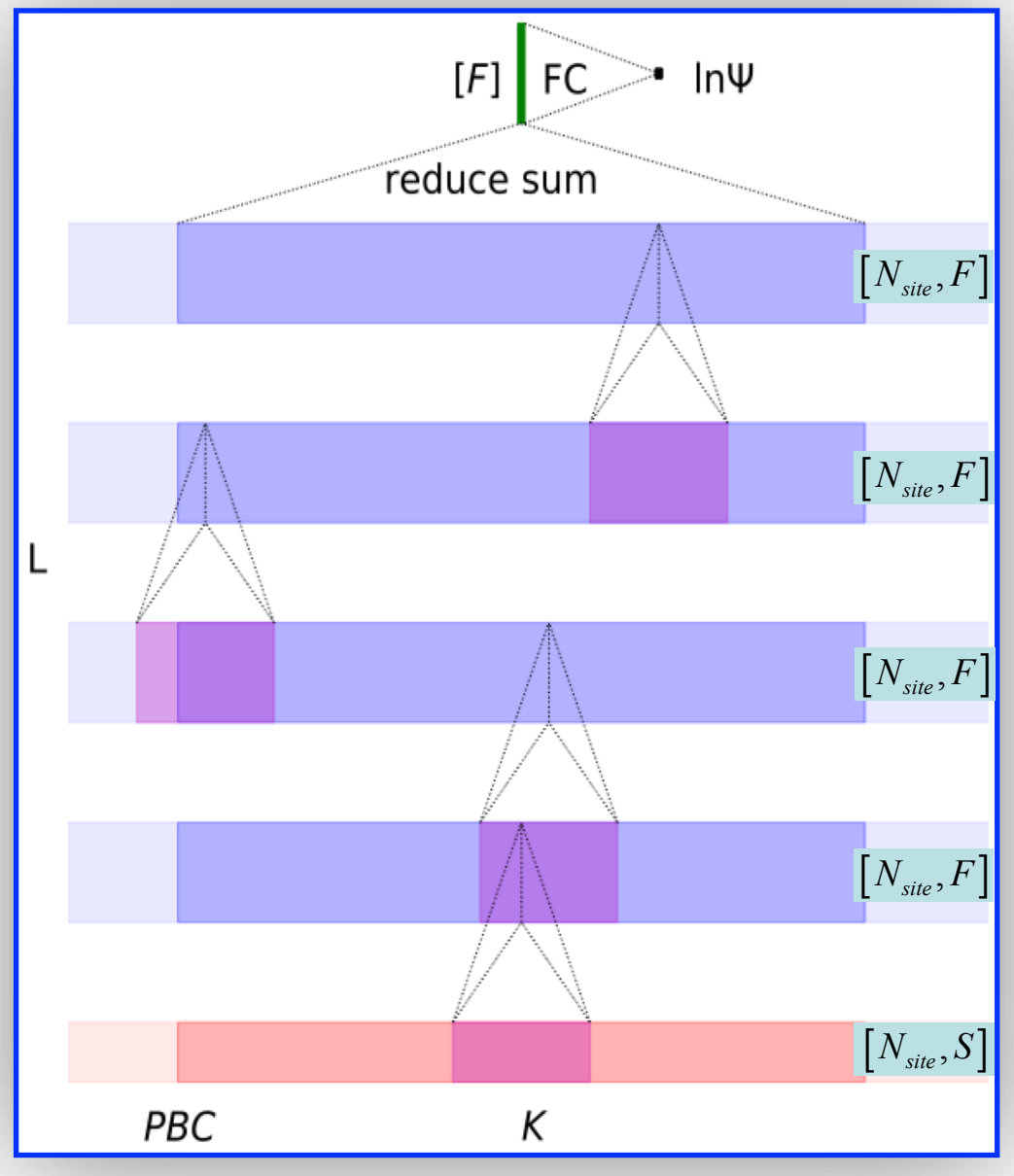
value encoding $S = 1$

$|\uparrow\downarrow\downarrow\uparrow\rangle$: $\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$

one hot encoding $S = N$

$|\uparrow\downarrow\downarrow\uparrow\rangle$: $\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$

CNN for quantum many-body physics



$$H = \sum_{i=1}^{N_{\text{site}}} \mathcal{E}_{i,i+1}$$

$$N_{\text{site}} = 60$$

input: $s_1, s_2, \cdots s_{N_{site}}$

output: $\Psi\left(s_1, s_2, \cdots s_{N_{site}}\right)$

need Ψ to be non-negative

CNN for quantum many-body physics

$$H = \sum_i \mathcal{E}_{i,i+1}$$

$$\mathcal{E}|\alpha,\beta\rangle = |\beta,\alpha\rangle$$

Transform to a new exchange operator :

$$\tilde{\mathcal{E}}|\alpha,\beta\rangle = \begin{cases} |\beta,\alpha\rangle & \alpha = \beta \\ -|\beta,\alpha\rangle & \alpha \neq \beta \end{cases}$$

$$H = \sum_i \tilde{\mathcal{E}}_{i,i+1}$$

All off-diagonal elements becomes negative (Stoquastic Hamiltonian)

Perron–Frobenius theorem guarantees non-negative ground state w.f.

Variational Monte Carlo (VMC)

Variational calculation

$$\Psi_s(w) \equiv \Psi(s_1, s_2, \ldots, s_N; w) \qquad \text{minimize:} \quad E = \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

$$w \leftarrow w - \alpha \frac{\partial E}{\partial w}$$

Problem:

$$\frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\sum_{s,s'} \Psi_s^* H_{s,s'} \Psi_{s'}}{\sum_s \Psi_s^* \Psi_s}$$

exponentially many terms

Solution: stochastic summation (VMC)

$$\frac{\langle \Psi | O | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \sum_s P_s O_s \qquad O_s = \frac{\sum_{s'} O_{s,s'} \Psi_{s'}}{\Psi_s} = \sum_{s'} O_{s,s'} e^{\ln \Psi_{s'} - \ln \Psi_s}$$

$$P_s = \frac{\Psi_s^* \Psi_s}{\sum_s \Psi_s^* \Psi_s}$$

Gradient Optimization

Traditional VMC optimization method

$$\frac{\partial E}{\partial w} = 2\mathrm{Re}\left(\sum_s P_s E_s \partial_w \ln\Psi_s - \sum_s P_s E_s \partial_w \sum_s P_s \partial_w \ln\Psi_s\right)$$

1. Sample a batch of states (MCMC)

$$s^{(0)} \rightarrow s^{(1)} \rightarrow s^{(2)} \ldots \rightarrow s^{(N_{\mathrm{sample}})}$$

2. Calculate the local energy and local derivative for each states

$$\left\{s^{(i)} : E_{s^{(i)}},\ \ln\Psi_{s^{(i)}}\right\}$$

3. Calculate the derivatives of energy function, and update parameters

$$\frac{\partial E}{\partial w} \approx 2\mathrm{Re}\left(\frac{\sum_{i=1}^{N_{\mathrm{sample}}} E_{s^{(i)}} \partial_w \ln\Psi_{s^{(i)}}}{N_{\mathrm{sample}}} - \frac{\sum_{i=1}^{N_{\mathrm{sample}}} E_{s^{(i)}}}{N_{\mathrm{sample}}} \frac{\sum_{i=1}^{N_{\mathrm{sample}}} \partial_w \ln\Psi_{s^{(i)}}}{N_{\mathrm{sample}}}\right)$$

$$w \leftarrow w - \alpha \frac{\partial E}{\partial w}$$

Gradient Optimization

Traditional VMC optimization method

$$\frac{\partial E}{\partial w} = 2\mathrm{Re}\left(\sum_s P_s E_s \partial_w \ln\Psi_s - \sum_s P_s E_s \partial_w \sum_s P_s \partial_w \ln\Psi_s\right)$$

1. Sample a batch of states (MCMC)

2. Calculate the local energy and local derivative for each states

3. Calculate the derivatives of energy function, and update parameters

Each of these steps can be done rather efficiently if the number of variational parameters is small.

In traditional VMC, number of variational parameters: $10^0 \sim 10^2$

This, however, won't be the case for DNN.

Importance Sampling Gradient Optimization

Importance Sampling Gradient Optimizer (ISGO)

$$\frac{\partial E}{\partial w} = 2\mathrm{Re}\left(\sum_s P_s E_s \partial_w \ln\Psi_s - \sum_s P_s E_s \partial_w \sum_s P_s \partial_w \ln\Psi_s\right)$$

1. Sample a batch of states

$$s^{(0)} \rightarrow s^{(1)} \rightarrow s^{(2)} \ldots \rightarrow s^{(N_{\mathrm{sample}})}$$

2. Calculate the local energy and local derivative for each states

$$\left\{s^{(i)} : E_{s^{(i)}},\ \ln\Psi_{s^{(i)}}\right\}$$

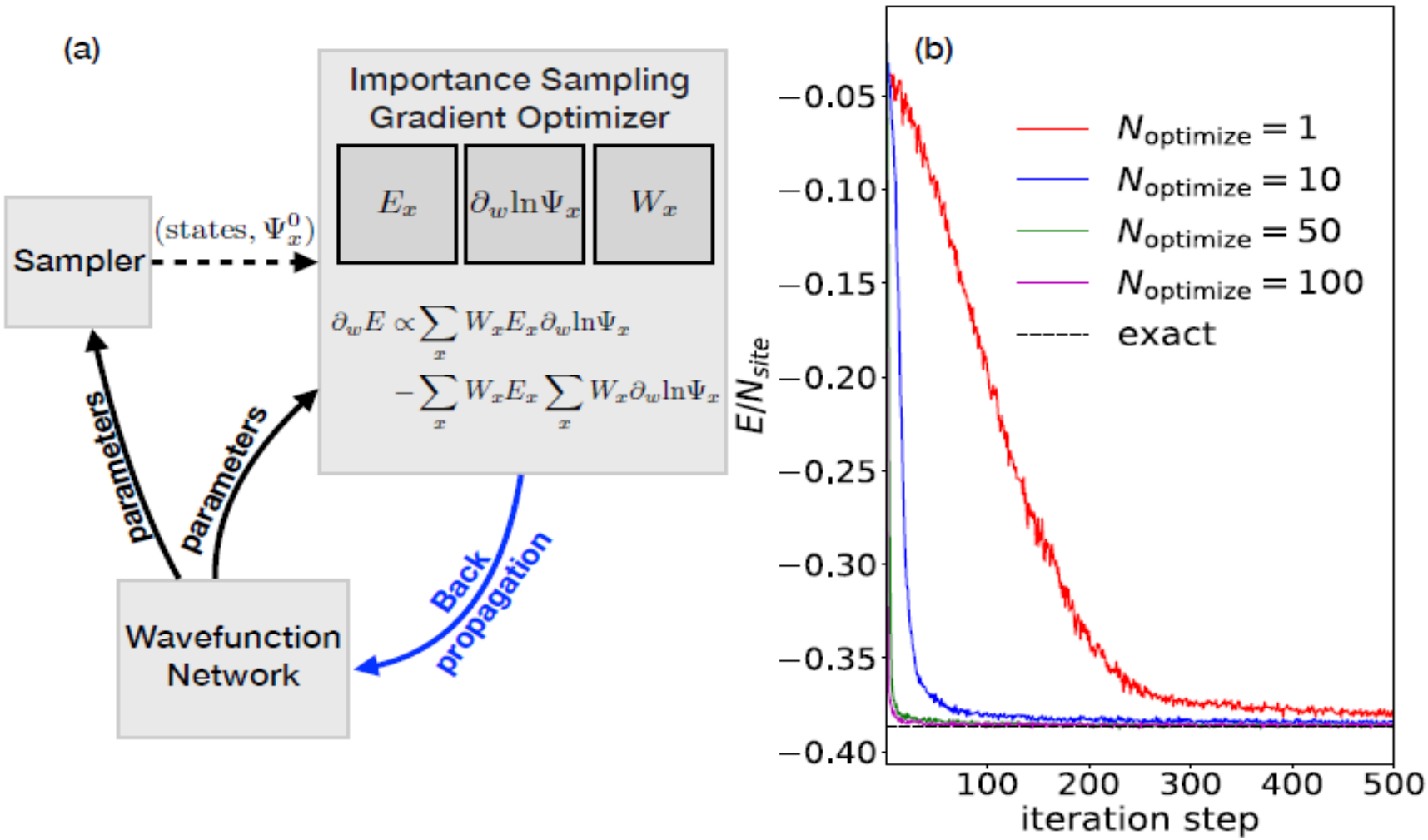
3. Calculate the derivatives of energy function, and update parameters

$$\frac{\partial E}{\partial w} \approx 2\mathrm{Re}\left(\frac{\sum_{i=1}^{N_{\mathrm{sample}}} W_{s^{(i)}} E_{s^{(i)}} \partial_w \ln\Psi_{s^{(i)}}}{N_{\mathrm{sample}}} - \frac{\sum_{i=1}^{N_{\mathrm{sample}}} W_{s^{(i)}} E_{s^{(i)}}}{N_{\mathrm{sample}}} \frac{\sum_{i=1}^{N_{\mathrm{sample}}} W_{s^{(i)}} \partial_w \ln\Psi_{s^{(i)}}}{N_{\mathrm{sample}}}\right)$$

$$W_{s^{(i)}} = C\frac{|\Psi_{s^{(i)}}|^2}{|\Psi^0_{s^{(i)}}|^2} \qquad w \leftarrow w - \alpha\frac{\partial E}{\partial w}$$

LOOP for N_{optimize}

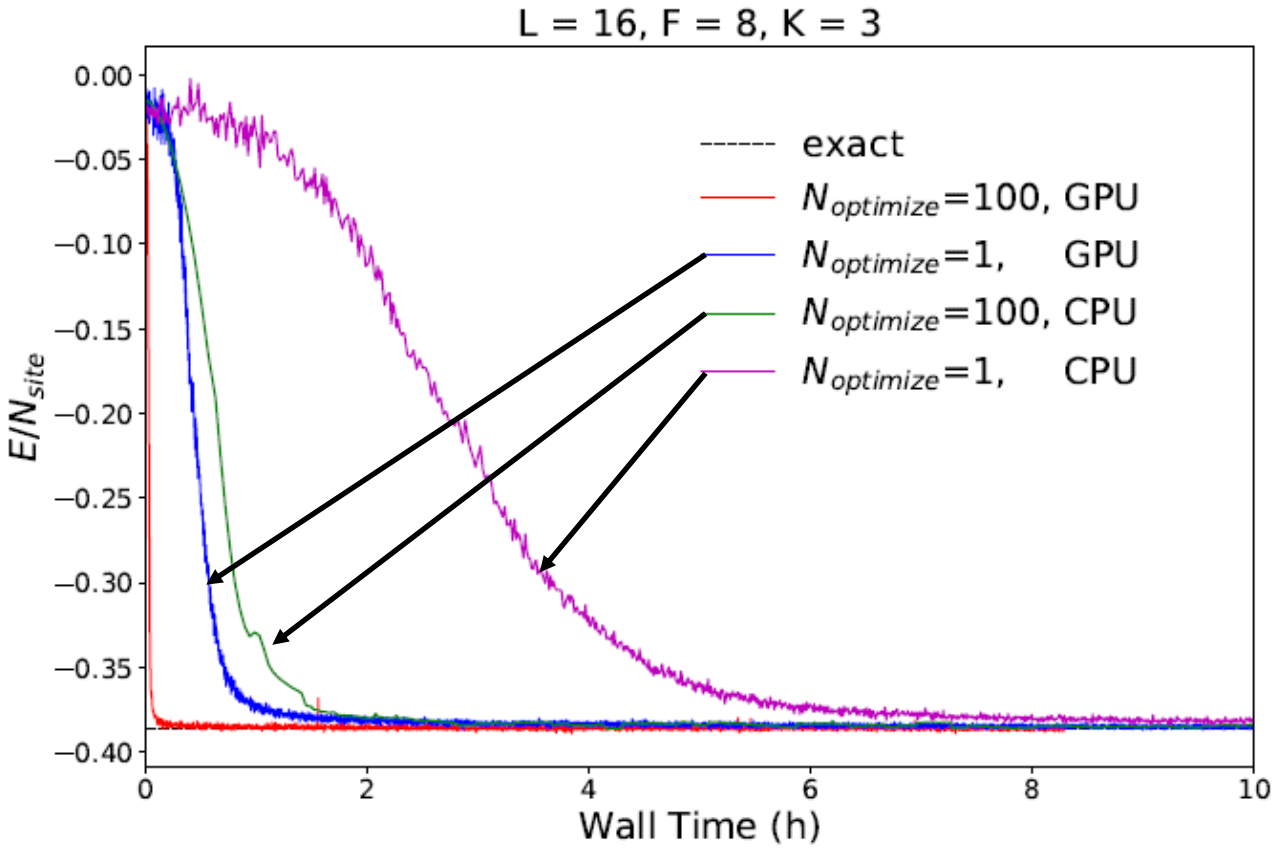
Importance Sampling Gradient Optimization



At each iteration step, network parameters are updated N_{optimize} times.

Importance Sampling Gradient Optimization

CPU vs. GPU, ISGO vs. conventional GO

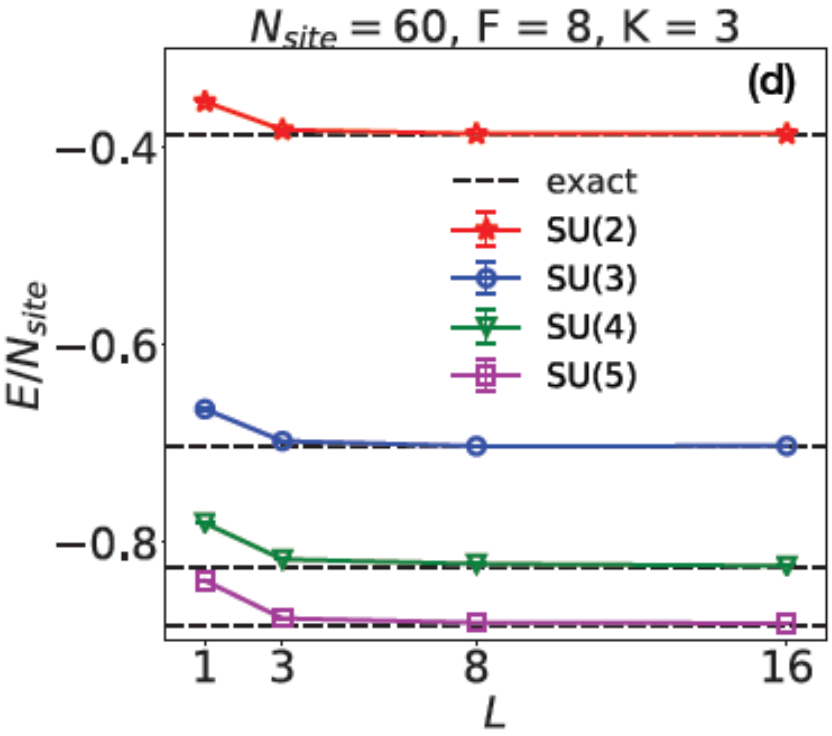


over one order of magnitude speed up with ISGO

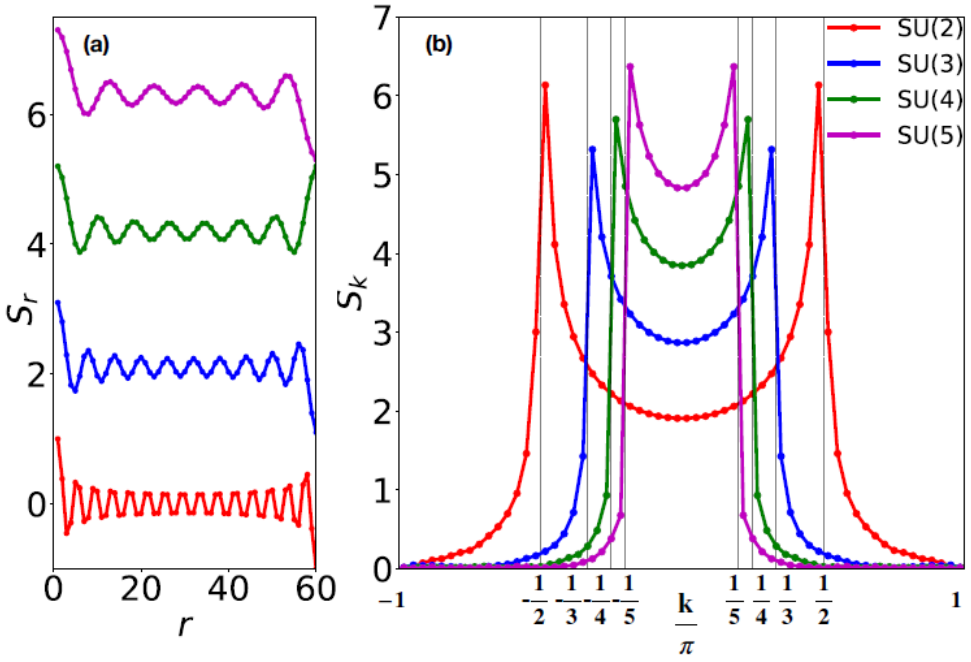
More dramatic speedup on GPU over CPU for more complicated networks

Results

Energy convergence and correlation function



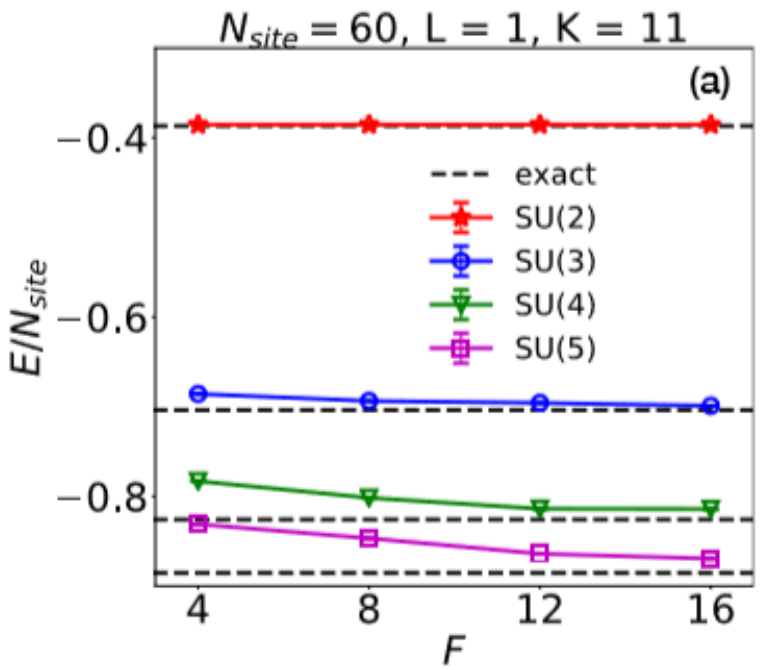
non-local spin correlation function



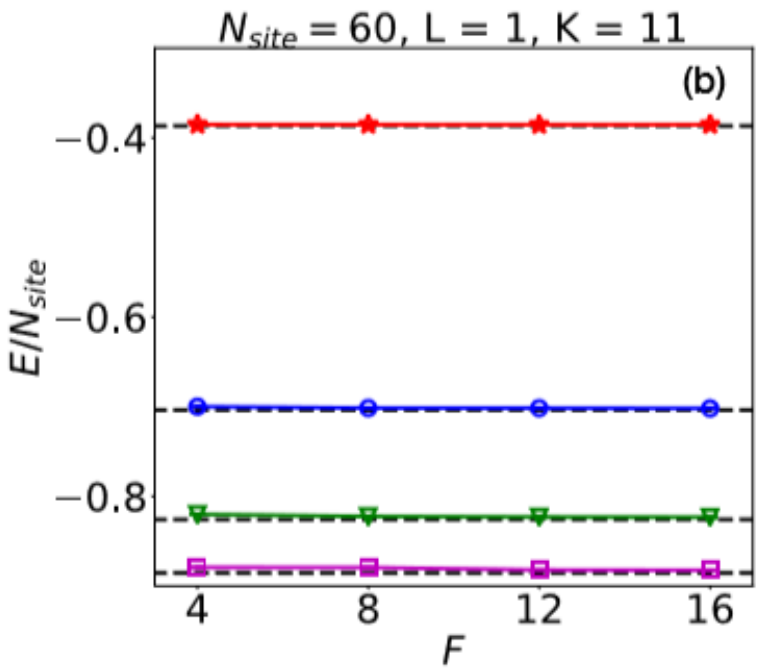
Yang *et al*. arXiv:1905.10730

Results

value encoding vs. one-hot encoding



value encoding



one-hot encoding

Yang *et al*. arXiv:1905.10730

Summary

- We've built a fully convolutional DNN to represent QM w.f.
- Tested its performance on a 1D SU(N) spin-chain model
- Developed the Importance Sampling Gradient Optimization method
- Effects of different encoding on the input state

https://github.com/liyang2019/VMC-ISGO

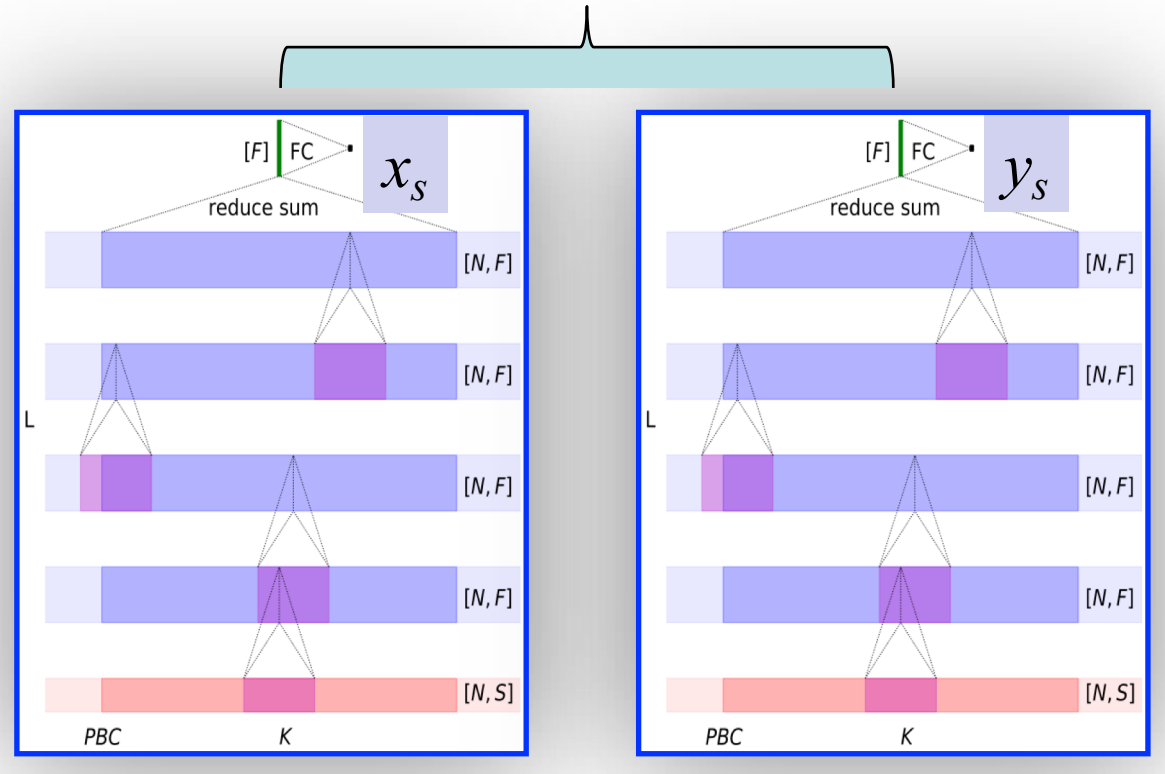
Yang *et al*. arXiv:1905.10730

Outlook

Quantum w.f. are in general complex-valued.

How to handle complex wavefunctions?

$$\Psi_s = e^{x_s + i y_s}$$



Developing efficient complex-valued DNN is essential for quantum physics applications.

Outlook

Many quantum systems possess distinct symmetry properties.

How to impose symmetries on networks?

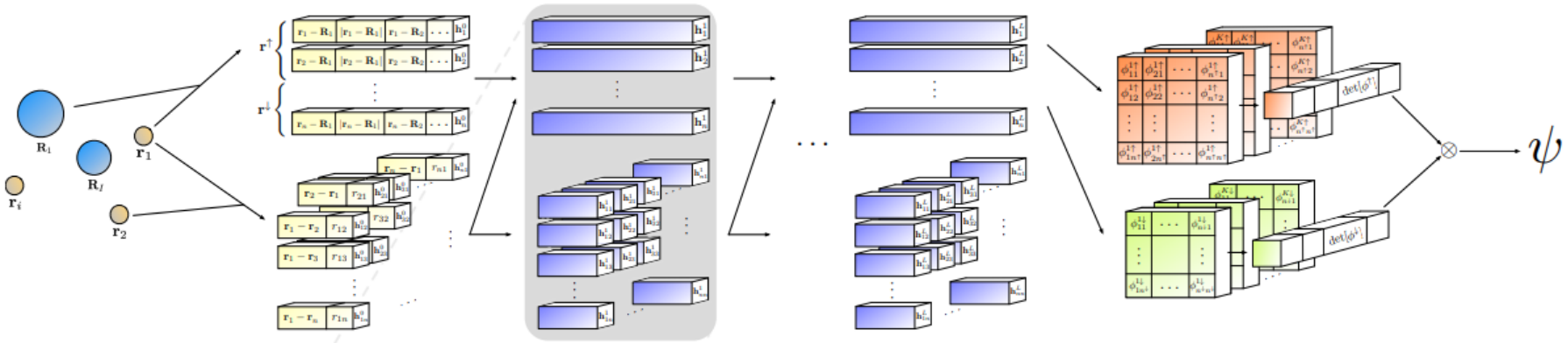
Ab-Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks

David Pfau, James S. Spencer, and Alexander G. de G. Matthews
DeepMind, 6 Pancras Square, London N1C 4AG

W. M. C. Foulkes
Department of Physics, Imperial College London, South Kensington Campus, London SW7 2AZ

(Dated: September 6, 2019)

Fermi Net (built-in Fermi statistics)



arXiv:1909.02487

Outlook

How can such studies help us better understand the mechanism of deep learning?