COLLÈGE
DE FRANCE
1530

Chaire de Physique
de la Matière Condensée
Antoine Georges

Réseaux de Neurones, Apprentissage et Physique Quantique

Cours 2:
Représentation d'Etats Quantiques par Réseaux de Neurones (Neural Quantum States)

Cycle 2022-2023
9 mai 2023

Image Credit: Javier Robledo-Moreno

COLLÈGE DE FRANCE
1530

Chaire de Physique
de la Matière Condensée
Antoine Georges

Machine Learning and Neural Networks for Quantum Physics

Lecture 2

Introduction to Neural Quantum States

Cycle 2022-2023
9 mai 2023

Image Credit: Javier Robledo-Moreno

Mailing List

(Weekly announcement of lecture and seminar, etc.)

Send email to: listes-diffusion.cdf@college-de-france.fr
Subject line: subscribe chaire-pmc.ipcdf
…or: unsubscribe chaire-pmc.ipcdf

You can also just send me an email to be placed on the list

Website:

https://www.college-de-france.fr/site/antoine-georges/index.htm

Lectures are video recorded
and available on the website

NQS in a Nutshell:

- Quantum Many-Body Problem
- Variational *ansatz* for the ground-state wavefunction:

$$\Psi(x_1, \cdots, x_N; \boldsymbol{\theta})$$

- Represent this wavefunction by a neural network (many possible architectures)
- Variational principle: optimize the parameters using the energy as loss function

Science 355, 602 (2017)

RESEARCH

RESEARCH ARTICLE

MANY-BODY PHYSICS

Solving the quantum many-body problem with artificial neural networks

Giuseppe Carleo[1]* and Matthias Troyer[1,2]

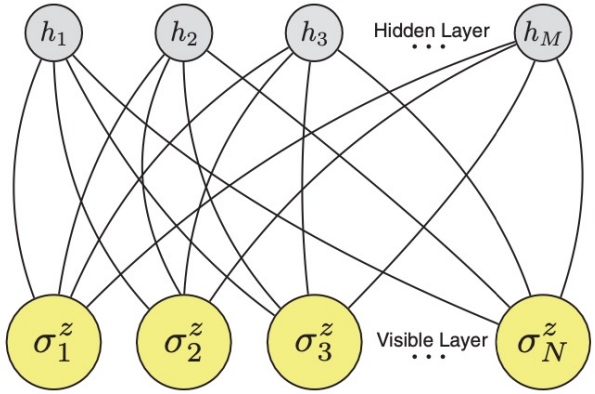


Fig. 1. Artificial neural network encoding a many-body quantum state of *N* spins. A restricted Boltzmann machine architecture that features a set of *N* visible artificial neurons (yellow dots) and a set of *M* hidden neurons (gray dots) is shown. For each value of the many-body spin configuration $\mathcal{S} = (\sigma_1^z, \sigma_2^z, \ldots, \sigma_N^z)$, the artificial neural network computes the value of the wave function $\Psi(\mathcal{S})$.

Some early papers (for the 1-particle Schrödinger equation):
Lagaris et al. Comp Phys Comm 104, 1 (1997); IEEE Trans Neural Net 9, 987 (1998)
Sugawara et al. Comp Phys Comm 140, 366 (2001)

Quantum Many Body Problem

Hilbert Space: Exponentially Large!

For example - N spin-1/2 or Spinless Fermions on N sites: Dimension = 2^N

Frequently Encountered `Simple' Hamiltonians:

- Transverse Field Ising Model

$$\sum_{\langle ij \rangle} J_{ij}\, \hat{S}_i^z \hat{S}_j^z + h_\perp \sum_i \hat{S}_i^x \equiv \sum_{\langle ij \rangle} J_{ij}\, Z_i Z_j + h_\perp \sum_i X_i$$

- Heisenberg Model

$$\sum_{\langle ij \rangle} J_{ij}\, \vec{S}_i \cdot \vec{S}_j = \sum_{\langle ij \rangle} J_{ij} \left[\hat{S}_i^z \hat{S}_j^z + \frac{1}{2} (\hat{S}_i^+ \hat{S}_j^- + \hat{S}_i^- \hat{S}_j^+) \right]$$

- Spinless Fermions

$$-t \sum_{\langle ij \rangle} (c_i^+ c_j + c_j^+ c_i) + V \sum_{\langle ij \rangle} \hat{n}_i \hat{n}_j$$

- Hubbard Model

$$- \sum_{\langle ij \rangle ; \sigma = \uparrow, \downarrow} t_{ij} (c_{i\sigma}^+ c_{j\sigma} + c_{j\sigma}^+ c_{i\sigma}) + U \sum_{\langle ij \rangle} \hat{n}_{i\uparrow} \hat{n}_{j\downarrow}$$

Constructing an N-particle basis from 1-particle states

The simple case of N spin-1/2:

$$H_N = H_1 \otimes \cdots \otimes H_1 = H_1^N$$

Basis of H_1: $|\pm\rangle = |S^z = \pm\hbar/2\rangle$

Basis of H_N: $|n\rangle = |n_1, \cdots, n_N\rangle\,,\; n_i = -, + \,(\text{or } 0, 1)$

N-spin quantum state: 2^N complex coefficients

$$|\Psi\rangle = \sum_{n_1 \cdots n_N} \psi(n_1, \cdots, n_N)|n_1 \cdots n_N\rangle = \sum_n \psi(n)|n\rangle$$

$$\sum_n |\psi(n)|^2 = 1$$

$$\boxed{\psi : \{0,1\}^N \to \mathbb{C}}$$

Variational Principle

Ground State Energy:

$$E_0 = \mathrm{Min}_\Psi \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}$$

$$\frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\sum_\alpha |c_\alpha|^2 E_\alpha}{\sum_\alpha |c_\alpha|^2}$$

$$= E_0 + \frac{\sum_\alpha |c_\alpha|^2 (E_\alpha - E_0)}{\sum_\alpha |c_\alpha|^2} \geq E_0$$

We have used a complete set of N-body eigenstates

Neural Quantum State

Introduce a neural network parametrization
of the amplitudes: $\psi : \{0,1\}^N \to \mathbb{C}$

$$|\Psi_{\text{var}}\rangle = \sum_n \psi_\theta(n) \, |n\rangle$$

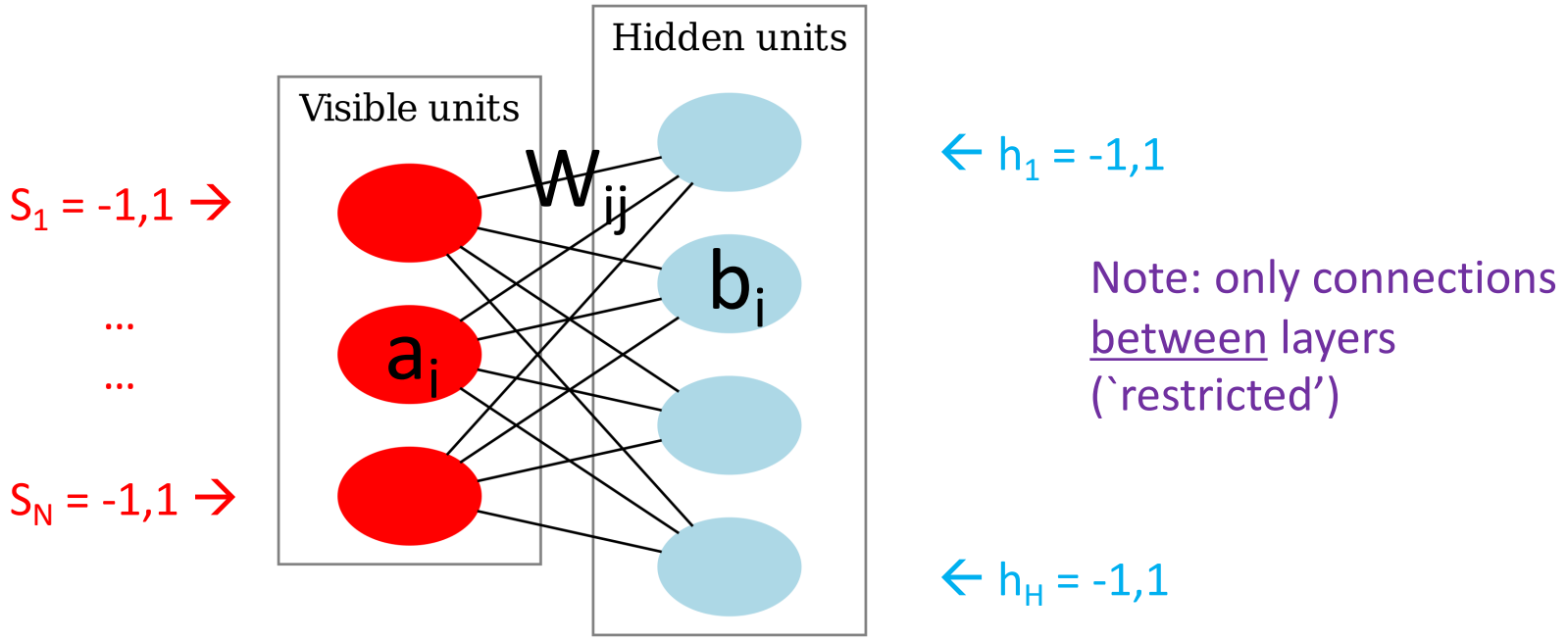
Optimize over the parameters θ using the energy as a loss function
and gradient descent - or another optimizer

Many possible NN architectures can be used:

- Restricted Botzmann Machine (RBM)
- Convolutional Neural Networks (CNN)
- Recurrent (autoregressive) Neural Network (see seminar on May 23)

A recent discussion comparing the performance of different architectures
and especially of different implementations of symmetries:
Reh, Schmittt and Gärtnner: https://arxiv.org/abs/2301.06788

Restricted Boltzmann Machines



Note: only connections between layers (`restricted')

$$\psi_\theta(s_1, \cdots, s_N) = \sum_{h_j = \pm 1} e^{\sum_i a_i s_i + \sum_j b_j h_j + \sum_{ij} W_{ij} s_i h_j}$$

$$\theta = \{a_i, b_j, W_{ij}\}$$

a,b: biases; W: weights
(Complex numbers in general)

Performing the sum over hidden variables:

$$\psi_\theta(\{s\}) = e^{\sum_{i=1}^{N} a_i s_i} \prod_{j=1}^{H} 2\cosh\left[b_j + \sum_i W_{ij} s_i\right]$$

Affine composition

Non-Linear Transformation
cf. Perceptron
(Note: the cosh could be replaced by another function)

Density of network:

$$\alpha = \frac{H}{N}$$

Controls the quality of the representation
(compression level,
cf. bond dimension for MPS)

Note: For a wave-function (in contrast to learning a probability density), we will have to allow the biases and weights to be *complex.* Other options: separate networks for amplitude and phase, etc.

RBMs as Universal Approximators

› Neural Comput. 2008 Jun;20(6):1631-49. doi: 10.1162/neco.2008.04-07-510.

Representational power of restricted boltzmann machines and deep belief networks

Nicolas Le Roux [1], Yoshua Bengio

(Probability) Density Estimate with KL divergence as loss function

1. `Better models with increasing number of hidden units'

Theorem 2.3. *Let p_0 be an arbitrary distribution over $\{0,1\}^n$ and let R_p be an RBM with marginal distribution p over the visible units such that $KL(p_0||p) > 0$. Then there exists an RBM $R_{p_{w,c}}$ composed of R_p and an additional hidden unit with parameters (w, c) whose marginal distribution $p_{w,c}$ over the visible units achieves $KL(p_0||p_{w,c}) < KL(p_0||p)$.*

2. `Huge models can represent any distribution'

Theorem 2.4. *Any distribution over $\{0,1\}^n$ can be approximated arbitrarily well (in the sense of the KL divergence) with an RBM with $k+1$ hidden units where k is the number of input vectors whose probability is not 0.*

See also: G.Hinton `A practical guide to training restricted Boltzmann machines'

Overview of different uses of RBMs in quantum physics:

nature physics

PERSPECTIVE

https://doi.org/10.1038/s41567-019-0545-1

Restricted Boltzmann machines in quantum physics

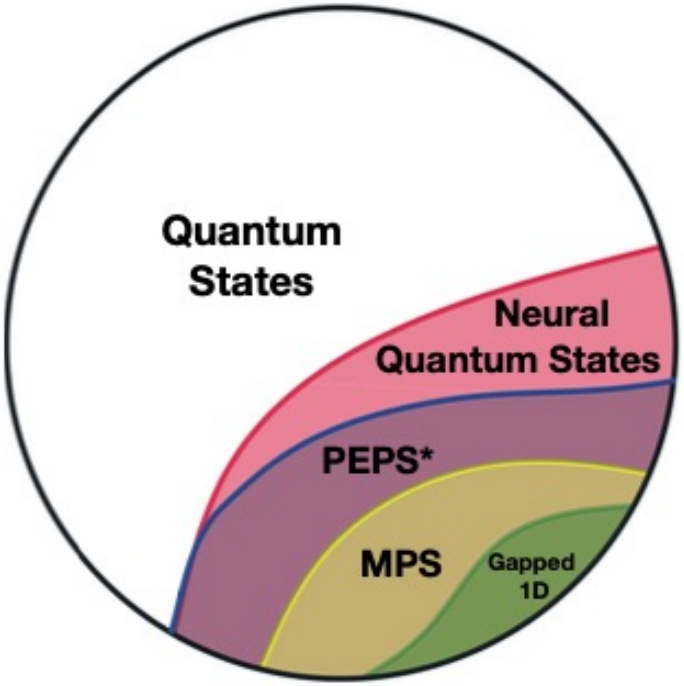
Roger G. Melko[1,2*], Giuseppe Carleo[3], Juan Carrasquilla[4] and J. Ignacio Cirac[5]

A type of stochastic neural network called a restricted Boltzmann machine has been widely used in artificial intelligence applications for decades. They are now finding new life in the simulation of complex wavefunctions in quantum many-body physics.

Expressive power of NQS

Connections to other wave function compression methods

See e.g. Sharir, Shashu and Carleo PRB 106, 205136 (2022)



Contains statements about NQS ability to represent MPS states at polynomial cost

Training an RBM in the context of NQS (for training in other contexts, see Hinton)

- The variational energy can be written as a sum over samples n (see below)
- Generate Samples {n} - Here a sample is $\{s_1,\ldots,s_N\}$
- [A Metropolis procedure is often used for generating samples: Variational Monte Carlo – see below]
- Compute the gradients of the variational energy and the metric g_{lm} as a sum over samples
- Update parameters according to:

$$\sum_l g_{ml}(\theta)\delta\theta_l = -\eta \frac{\partial E_\theta}{\partial \theta_m}$$

- Repeat process until convergence

Expression of the energy, gradient and metric tensor (not specific to NQS, general for any variational wave-function method)

1. The Energy:

$$E_\theta = \frac{\langle \Psi_\theta | H | \Psi_\theta \rangle}{\langle \Psi_\theta | \Psi_\theta \rangle} = \frac{\sum_{nn'} \psi_\theta(n)^* \psi_\theta(n') \langle n | H | n' \rangle}{\sum_n |\psi_\theta(n)|^2}$$

$$\boxed{E_\theta = \sum_n P_\theta(n)\, e_\theta(n)}$$

$$P_\theta(n) = \frac{|\psi_\theta(n)|^2}{\sum_n |\psi_\theta(n)|^2}$$

Probability distribution: relative weight
Of each basis state/configuration

$$e_\theta(n) = \sum_{n'} \frac{\psi_\theta(n')}{\psi_\theta(n)} \langle n | H | n' \rangle = \frac{\langle n | H | \Psi_\theta \rangle}{\psi_\theta(n)}$$

`Local Energy'

2. The Gradients

$$|\delta\Psi_\theta\rangle = \sum_l \delta\theta_l \sum_n \frac{\partial\psi_\theta(n)}{\partial\theta_l}|n\rangle + O(\delta\theta^2)$$

$$D_l \equiv \sum_n \frac{\partial \ln\psi_\theta(n)}{\partial\theta_l}\,|n\rangle\langle n|$$

$$|\delta\Psi_\theta\rangle = \sum_l \delta\theta_l\, D_l|\Psi_\theta\rangle + O(\delta\theta^2)\ ,\quad \frac{\partial}{\partial\theta_l}|\Psi_\theta\rangle = D_l|\Psi_\theta\rangle$$

$$\partial_l\langle\Psi_\theta|\Psi_\theta\rangle = \langle\partial_l\Psi_\theta|\Psi_\theta\rangle + \langle\Psi_\theta|\partial_l\Psi_\theta\rangle = \langle\Psi_\theta|D_l^+ + D_l|\Psi_\theta\rangle$$

$$\partial_l\langle\Psi_\theta|H|\Psi_\theta\rangle = \langle\Psi_\theta|D_l^+H + HD_l|\Psi_\theta\rangle$$

With the notation: $\langle\hat{O}\rangle_\theta \equiv \dfrac{\langle\Psi_\theta|\hat{O}|\Psi_\theta\rangle}{\langle\Psi_\theta|\Psi_\theta\rangle}$

$$\begin{aligned}\partial_l E_\theta &= \langle D_l^+H + HD_l\rangle_\theta - \langle D_l^+ + D_l\rangle_\theta\langle H\rangle_\theta \\ &= 2\mathrm{Re}\left[\langle HD_l\rangle_\theta - \langle H\rangle_\theta\langle D_l\rangle_\theta\right]\end{aligned}$$

This can be rewritten as an average over configurations (samples):

$$\langle \Psi_\theta | H D_l | \Psi_\theta \rangle = \sum_n \langle \Psi_\theta | H | n \rangle \langle n | D_l | \Psi_\theta \rangle =$$

$$= \sum_n \langle \Psi_\theta | H | n \rangle D_{l,\theta}(n) \psi_\theta(n) = \sum_n |\psi_\theta(n)|^2 e^*_\theta(n) D_{l,\theta}(n)$$

Defining:

$$\boxed{\langle F \rangle_P \equiv \sum_n P_\theta(n)\, F(n) \;\;,\;\; P_\theta(n) = \frac{|\psi_\theta(n)|^2}{\sum_n |\psi_\theta(n)|^2}}$$

$$\boxed{\begin{aligned} \partial_l E_\theta &= \langle e^*_\theta D_{l\theta} \rangle_P - \langle e^*_\theta \rangle_P \langle D_{l\theta} \rangle_P \\ &= \langle (e^*_\theta - \langle e^* \rangle_P)(D_{l\theta} - \langle D_{l\theta} \rangle_P \rangle_P \end{aligned}}$$

With:

$$D_{l\theta}(n) = \frac{\partial \ln \psi_\theta(n)}{\partial \theta_l}$$

3. The quantum geometric metric tensor

Fubini-Study metric over wave-functions:

$$(\Psi, \Phi)_{FS} = \arccos \frac{|\langle\Psi|\Phi\rangle|}{\sqrt{\langle\Psi|\Psi\rangle\langle\Phi|\Phi\rangle}}$$

= 0 for identical states, = π/2 for orthogonal states

Define distance in parameter space as:

$$(\Psi_\theta, \Psi_{\theta+\delta\theta})^2_{FS} = \sum_{lm} g_{lm}(\theta)\, \delta\theta_l \delta\theta_m + \cdots$$

After some calculation…:

$$g_{lm} = \mathrm{Re}\langle \left(D_l^+ - \langle D_l^+\rangle_\theta\right)\left(D_m - \langle D_m\rangle_\theta\right)\rangle_\theta$$

$$g_{lm} = \mathrm{Re}\Big\{ \sum_n P_\theta(n)\, \left(\partial_l \ln \psi_\theta(n)\right)^* \partial_m \ln \psi_\theta(n) -$$

$$- \sum_n P_\theta(n)\, \left(\partial_l \ln \psi_\theta(n)\right)^* \sum_n P_\theta(n)\, \partial_m \ln \psi_\theta(n) \Big\}$$

Note similarity to Fisher information matrix from Lecture 1!

If instead we define averages over configurations with the uniform metric:

$$\langle F \rangle_u \equiv \frac{1}{\mathcal{N}} \sum_n F(n) \ , \ \mathcal{N} = \sum_n |\psi_\theta(n)|^2$$

We can get rid of the log's and obtain:

$$g_{lm} = \mathrm{Re}\left\{ \langle (\partial_l \psi_\theta(n))^* \, \partial_m \psi_\theta(n) \rangle_u - \langle (\partial_l \psi_\theta(n))^* \rangle_u \, \langle \partial_m \psi_\theta(n) \rangle_u \right\}$$

Quantum geometric metric

Imaginary part related to Berry curvature

See: J.Stokes arXiv:1909.02108

RBM: Explicit expression of the gradients

$$\ln \psi_\theta = \sum_i a_i s_i + \sum_j \ln \left[2 \cosh \left(b_j + \sum_i s_i W_{ij} \right) \right]$$

$$\frac{\partial}{\partial a_i} \ln \psi_\theta(\{s\}) = s_i$$

$$\frac{\partial}{\partial b_j} \ln \psi_\theta = \tanh \Theta_j(\{s\})$$

$$\frac{\partial}{\partial W_{ij}} \ln \psi_\theta = s_i \tanh \Theta_j(\{s\})$$

$$\Theta_j(\{s\}) \equiv b_j + \sum_i s_i W_{ij}$$

Generating Samples

We want to generate samples according to the probability distribution:

$$P_\theta(n) = \frac{|\psi_\theta(n)|^2}{\sum_n |\psi_\theta(n)|^2}$$

For small systems we can do that by `exact sampling':

netket.sampler.ExactSampler

class **netket.sampler.ExactSampler**

Bases: `Sampler`

This sampler generates i.i.d. samples from $|\Psi(\sigma)|^2$.

In order to perform exact sampling, $|\Psi(\sigma)|^2$ is precomputed an all the possible values of the quantum numbers σ. This sampler has thus an exponential cost with the number of degrees of freedom, and cannot be used for large systems, where Metropolis-based sampling are instead a viable option.

Need to calculate the amplitudes for all samples to ensure normalisation

Metropolis Sampling (Monte Carlo)

Stochastic process: random walk in the space of configuration, discrete `time' steps. Transition probability from n to n' (note order) $w_{n'n}$

Master equation:

$$P_{t+1}(n') = \sum_n w_{n'n}\, P_t(n)$$

$$\forall n, \sum_{n'} w_{n'n} = 1 \Rightarrow \forall t, \sum_n P_t(n) = 1$$

Can be rewritten:

$$P_{t+1}(n') - P_t(n') = \sum_n \left[w_{n'n}\, P_t(n) - w_{nn'} P_t(n') \right]$$

Detailed balance insures stationarity of equilibrium distribution:

$$\boxed{w_{n'n}\, P_{eq}(n) \;=\; w_{nn'}\, P_{eq}(n')}$$

Note: Convergence to equilibrium and uniqueness require additional care/proof

Metropolis:

(Metropolis-Hastings if trial rate not symmetric)

Propose a move n→ n' according to some trial probability $t_{nn'}$ and accept or reject it according to acceptance rate:

$$a(n', n) = \text{Min}\left\{1, \frac{P_{eq}(n')}{P_{eq}(n)}\right\}$$

In practice this means:

- Propose a move n → n'
- If P(n') > P(n), accept the move
- If P(n') < P(n): accept with probability P(n')/P(n), reject with 1-P(n')/P(n)
- i.e. pick a random number from uniform distribution on [0,1] - accept if r < P(x')/P(x)

It is easily checked that this rule obeys detailed balance.

Important remark : this procedures does not require a calculation of the normalisation of $P_{eq}(n)$ – which is computationally demanding.

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*
(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

I. INTRODUCTION

THE purpose of this paper is to describe a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. Classical statistics is assumed,

netket.sampler.MetropolisSampler

class **netket.sampler.MetropolisSampler**

Bases: `Sampler`

Metropolis-Hastings sampler for a Hilbert space according to a specific transition rule.

The transition rule is used to generate a proposed state s', starting from the current state s. The move is accepted with probability

$$A(s \to s') = \min\left(1, \frac{P(s')}{P(s)} e^{L(s,s')}\right),$$

where the probability being sampled from is $P(s) = |M(s)|^p$. Here $M(s)$ is a user-provided function (the machine), p is also user-provided with default value $p = 2$, and $L(s, s')$ is a suitable correcting factor computed by the transition kernel.

The dtype of the sampled states can be chosen.

NQS for Quantum Spin Models: Illustrative Results

Transverse Field Ising Model in One Dimension

Carleo, Troyer Science 355, 602 (2017)

$$H = J \sum_{i=1}^{L} S_i^z S_{i+1}^z + h \sum_i S_i^x$$

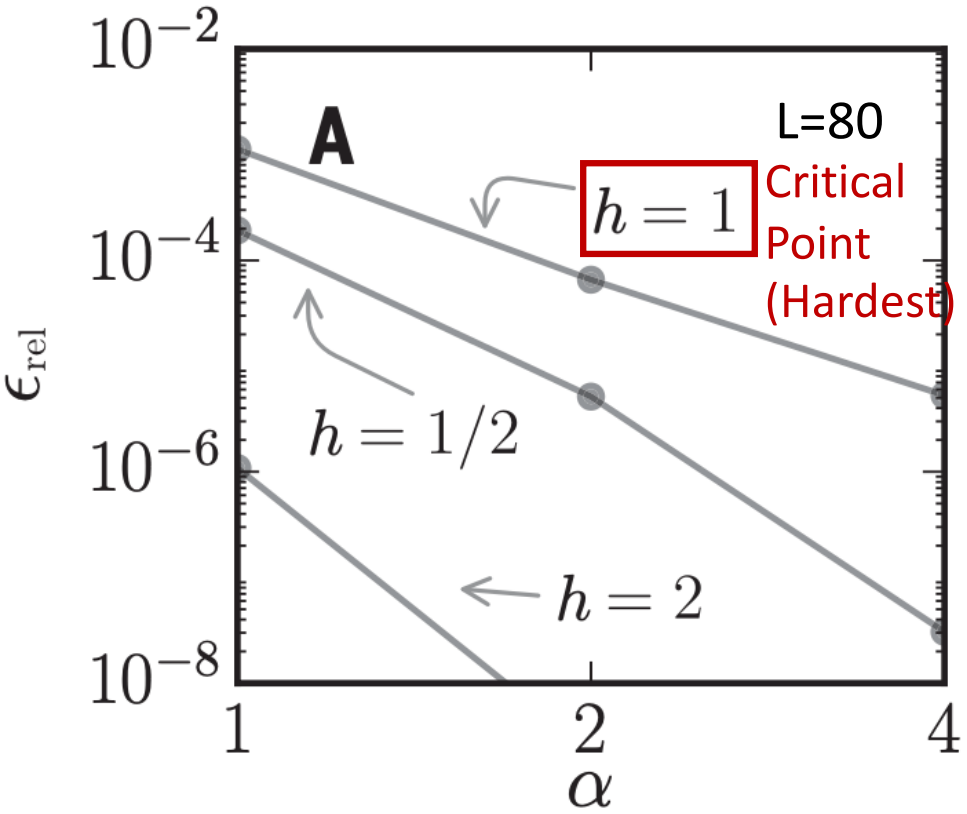
This is an exactly solvable model:
Mapping on free fermions
by Jordan-Wigner transformation

Ground-state energy:
Relative error

$$\epsilon_{rel} = \frac{E_{RBM} - E_{exact}}{E_{exact}}$$

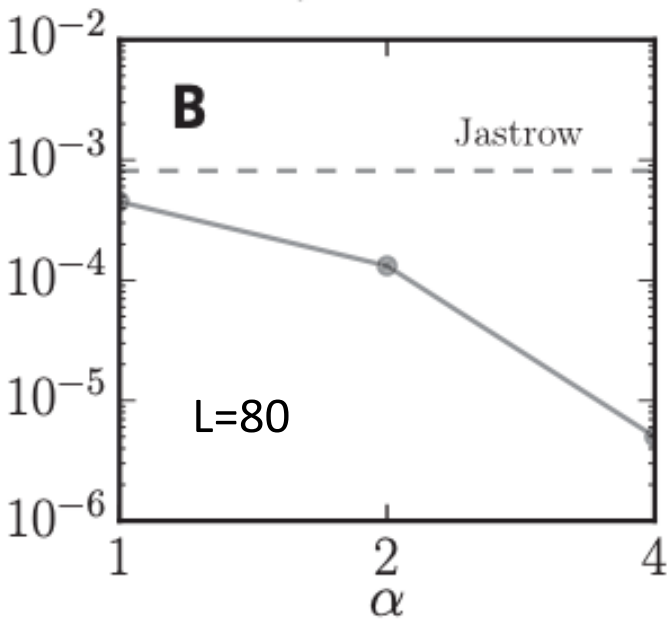
vs `density' of network $\alpha = \frac{H}{L}$

h<1: Ising ordered (twofold);
h>1 Polarized along x, no Ising order;
h=1 Gapless



AF Heisenberg Model, d=1 and d=2

Carleo, Troyer Science 355, 602 (2017)

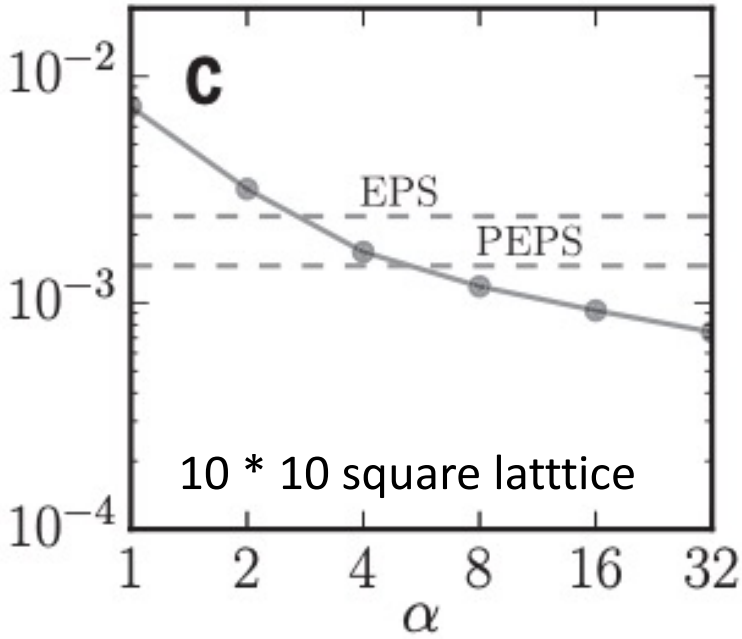


d=1: Exact ground-state energy known from Bethe Ansatz.

Comparison to Jastrow variational wave-function

$$\hat{H} = J \sum_{\langle ij \rangle} \left[\hat{S}_i^x \hat{S}_j^x + \hat{S}_i^y \hat{S}_j^y + \hat{S}_i^z \hat{S}_j^z \right]$$

d=2: PEPS benchmark
(Projector Entangled Pair State)
= State of the art tensor network method



Convergence of the optimization procedure

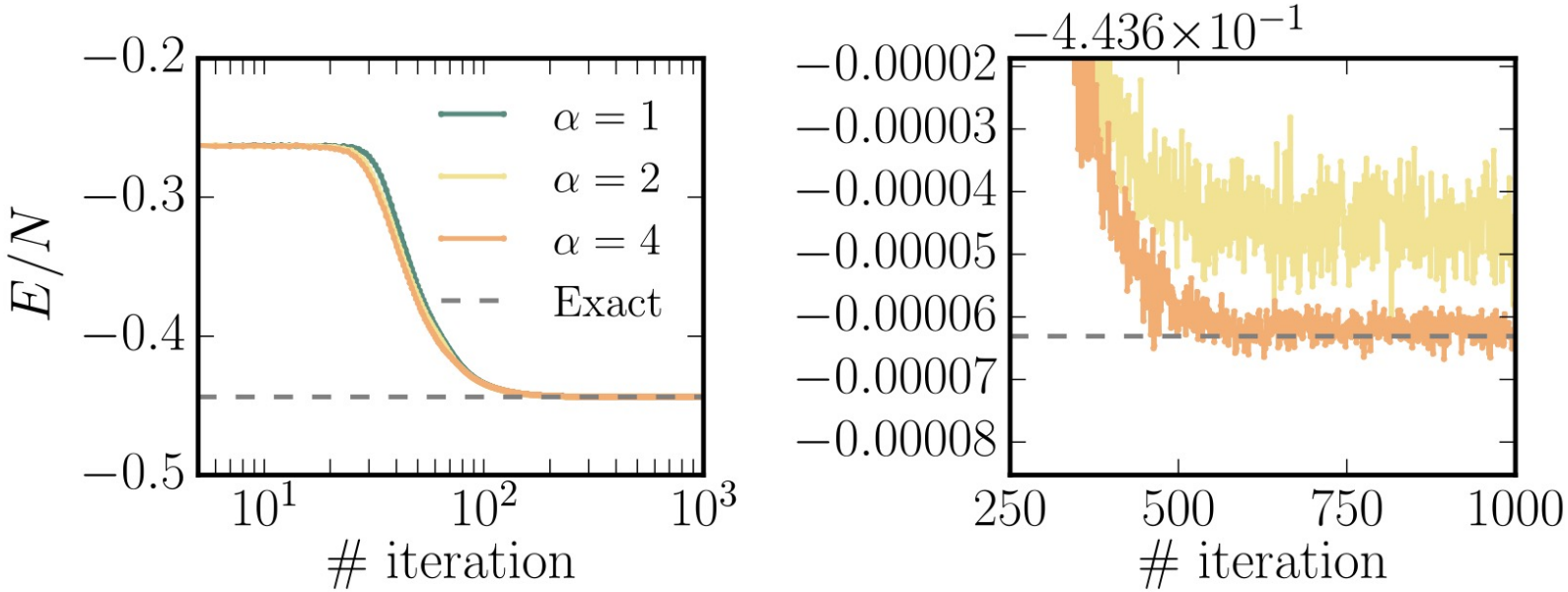


Figure S1.

Convergence properties of the stochastic optimization. Variational energy for the 1D Heisenberg model as a function of the Stochastic Reconfiguration iterates, and for different values of the hidden units density α. The system has PBC over a chain of $N = 40$ spins. The energy converges smoothly to the exact energy (dashed horizontal line) upon increasing α. In the Left panel, we show a complete view of the optimization procedure and on the Right panel a zoom in the neighborhood of the exact energy.

From Carlo and Troyer, Supp. Mat.

This is a case where optimization works very nicely. There are more tricky cases…

Examples of recent work

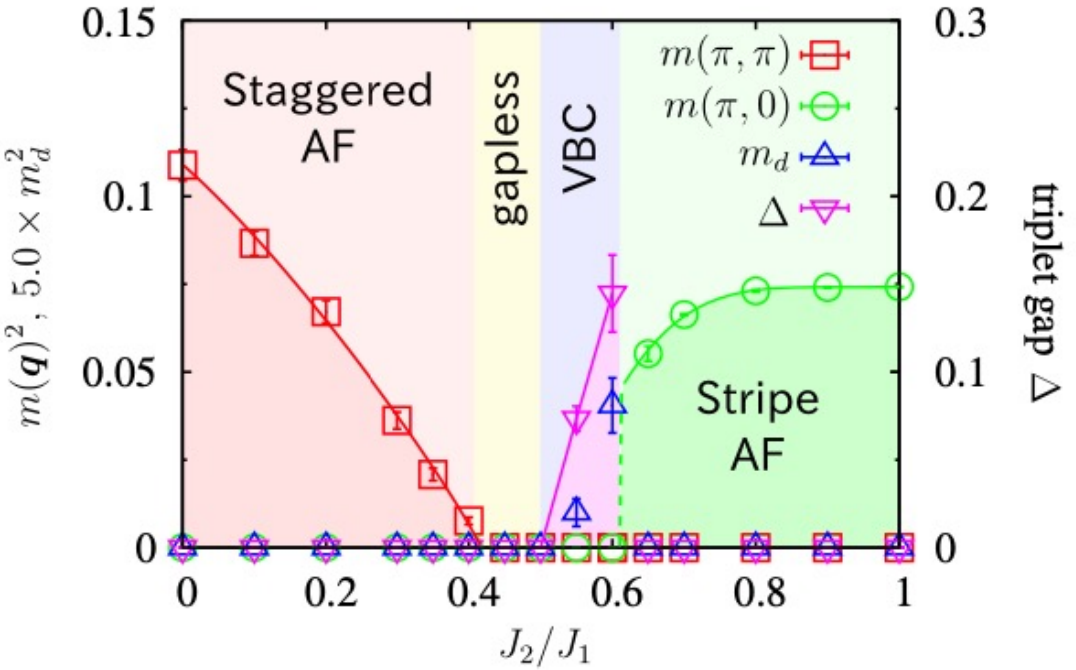
(Not discussed in details…)

J_1-J_2 Heisenberg 2D Model

Choo, Neupert and Carleo PRB 100, 125124 (2019)

$$\hat{H} = J_1 \sum_{\langle ij \rangle} \hat{\vec{S}}_i \cdot \hat{\vec{S}}_j + J_2 \sum_{\langle\langle ij \rangle\rangle} \hat{\vec{S}}_i \cdot \hat{\vec{S}}_j$$

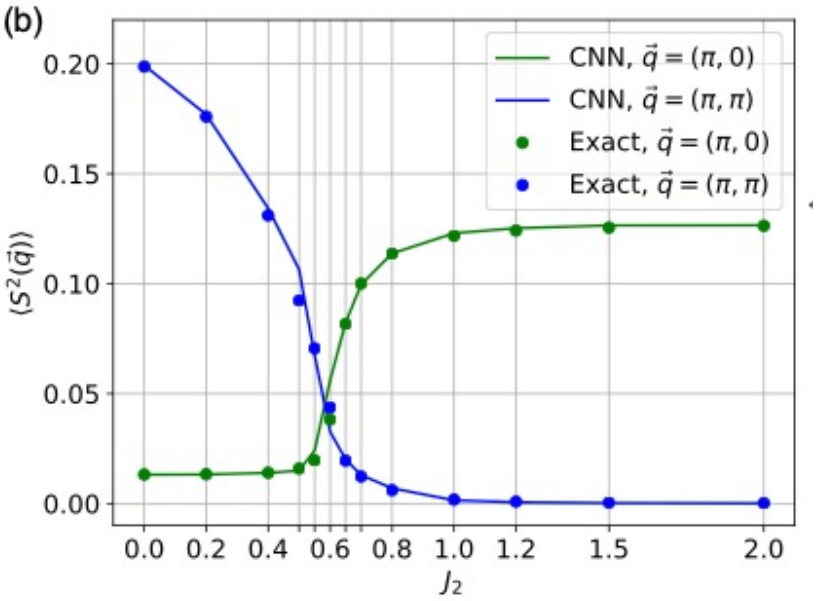
The J1-J2 models is known to have an intermediate phase(s) – nature yet to be clarified



VMC study: Morita, Kaneko and Imada JPSJ 84, 024720 (2015)

Two-dimensional frustrated J_1-J_2 model studied with neural network quantum states

Kenny Choo,[1] Titus Neupert,[1] and Giuseppe Carleo[2]



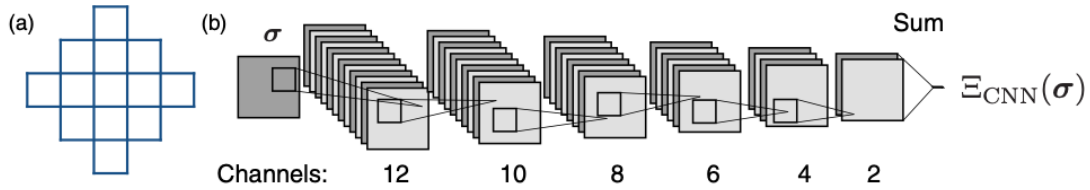


FIG. 1. Network architecture. (a) Shape of the filter which we apply across the square lattice. (b) Full architecture of the convolutional neural network used in this work. There are six convolutional layers followed by an output layer which simply sums the values of the prenultimate layer. In the first layer, we use the logarithm of the hyperbolic cosine as the activation function $g_{\mathrm{lncosh}}(z) = \log[\cosh(z)]$, while in all other layers the activation function is given by the complex generalization of the ReLU [51,52]. The total number of complex-valued parameters in the network is 3838.

TABLE I. Comparison between Gutzwiller-projected mean-field fermionic wave functions VMC [40] (on the 10×10 case the energies were provided by F. Ferrari and F. Becca), DMRG with 8192 SU(2) states, or equivalently 32 000 U(1) states [41] and the CNN used in this paper. The exact energies on the 6×6 case were take from Ref. [47].

6×6	$J_2 = 0.0$	$J_2 = 0.2$	$J_2 = 0.4$	$J_2 = 0.45$	$J_2 = 0.5$	$J_2 = 0.55$	$J_2 = 0.6$	$J_2 = 0.8$	$J_2 = 1.0$
Exact	−0.678 872	−0.599 046	−0.529 745	–	−0.503 810	−0.495 178	−0.493 239	−0.586 487	−0.714 360
VMC	–	–	−0.52715(1)	−0.51364(1)	−0.50117(1)	−0.48992(1)	–	–	–
DMRG	–	–	−0.529 744	−0.515 655	−0.503 805	−0.495 167	–	–	–
CNN	−0.67882(1)	−0.59895(1)	−0.52936(1)	−0.51452(1)	−0.50185(1)	−0.49067(2)	−0.49023(1)	−0.58590(1)	−0.71351(1)
10×10	$J_2 = 0.0$	$J_2 = 0.2$	$J_2 = 0.4$	$J_2 = 0.45$	$J_2 = 0.5$	$J_2 = 0.55$	$J_2 = 0.6$	$J_2 = 0.8$	$J_2 = 1.0$
VMC	−0.66935(1)	−0.59082(1)	−0.52229(1)	−0.50764(1)	−0.49439(1)	−0.48227(1)	−0.47259(1)	−0.56899(1)	−0.69123(1)
DMRG	–	–	−0.522 391	−0.507 976	−0.495 530	−0.485 434	–	–	–
CNN	−0.67135(1)	−0.59275(1)	−0.52371(1)	−0.50905(1)	−0.49516(1)	−0.48277(1)	−0.47604(1)	−0.57383(1)	−0.69636(1)

M.Reh, M.Schmitt and M.Gärttner
arXiv:2301.067788

Optimizing Design Choices for Neural Quantum States

Comparison of different architectures: RBM, Convolutional (CNN),
Recurrent networks (RNN) of different types
(Long short-term memory LSTM, Gated Recurrent Unit GRU)
and especially different ways of implementing symmetries

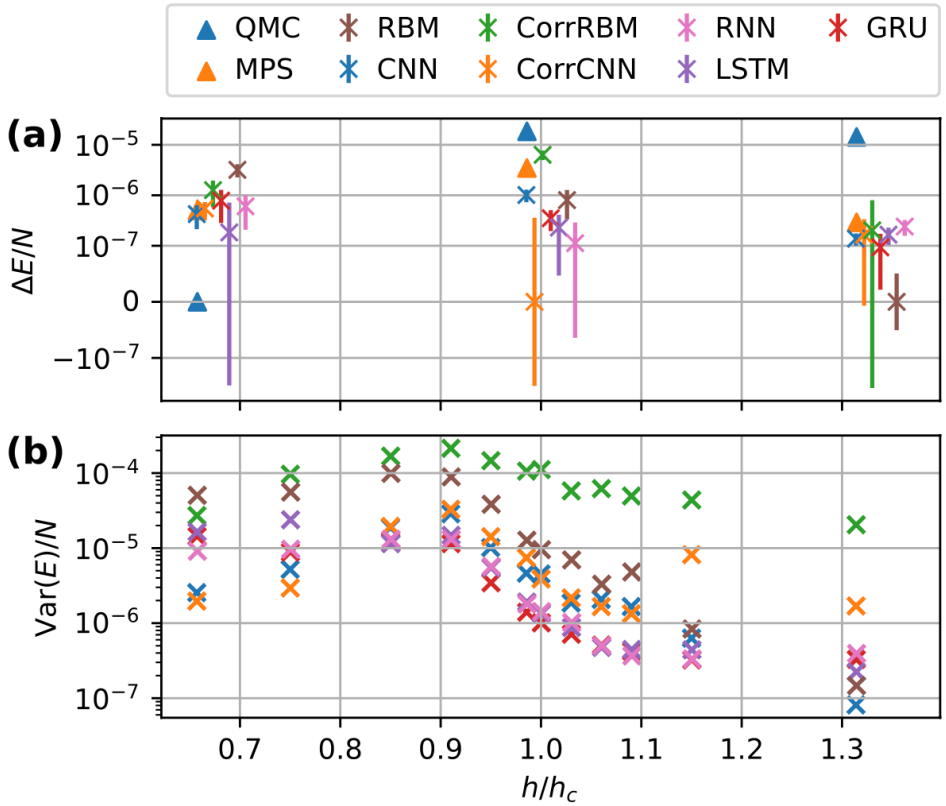


FIG. 1. Performance of the tested network architectures for finding the ground state of the 2D 12×12 TFIM at different magnetic field strengths. Here all networks use the bare-symmetrization as no phase is modelled. In (a) the deviation from the lowest observed energy density is shown for the magnetic fields $h \in [2, 3, 4]$. The data points are artificially shifted horizontally such that differences are more easily visible. In (b) the energy variance per spin is plotted, which serves as an additional performance indicator. The QMC and MPS data was taken from [20] and [21].

Different implementations of symmetries Reh et al. arXiv:2301.067788

- **Bare-Symmetry:**

Parametrized by NN

$$\psi_{\boldsymbol{\theta}}^{S}(\mathbf{s}) = \exp\left(\frac{1}{|\mathcal{S}|}\sum_{\mathbf{s}'\in\mathcal{S}(\mathbf{s})}\chi_{\boldsymbol{\theta}}(\mathbf{s}')\right)$$

Invariant

- **Exp-Symmetry:**

$$\psi_{\boldsymbol{\theta}}^{S}(\mathbf{s}) = \frac{1}{|\mathcal{S}|}\sum_{\mathbf{s}'\in\mathcal{S}(\mathbf{s})}\exp\left(\chi_{\boldsymbol{\theta}}(\mathbf{s}')\right)$$

Equivariant

- **Sep-Symmetry:**

$$\psi_{\boldsymbol{\theta}}^{S}(\mathbf{s}) = \sqrt{\frac{1}{|\mathcal{S}|}\sum_{\mathbf{s}'\in\mathcal{S}(\mathbf{s})}\exp\left(2\mathrm{Re}\left[\chi_{\boldsymbol{\theta}}(\mathbf{s}')\right]\right)} \times \exp\left(i\arg\left(\sum_{\mathbf{s}'\in\mathcal{S}(\mathbf{s})}\exp\left(i\ \mathrm{Im}\left[\chi_{\boldsymbol{\theta}}(\mathbf{s}')\right]\right)\right)\right)$$

Implementing symmetries is an active research topic in the field

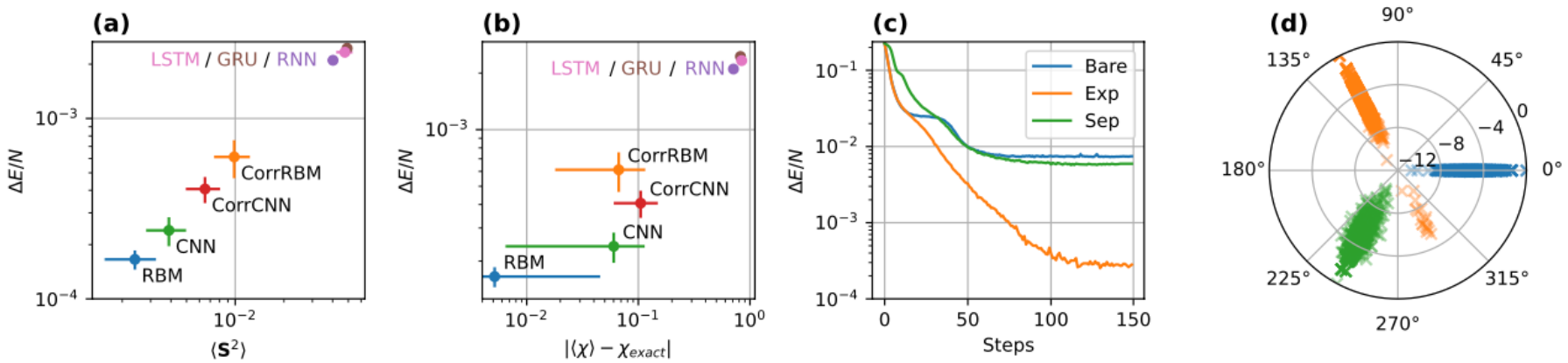


FIG. 2. Results on the 2D 6×6 J_1-J_2 model. In (a) and (b) the performance of the different architectures is depicted as a function of observables of interest. The errorbars which are smaller than the filled in circles are omitted. In (c), the effect of the different symmetrization procedures is shown for the case of the RBM architecture, leaving all other specifications of the network unchanged. In (d), sampled wave function coefficients of the network are depicted in the complex plane for the symmetrizations shown in (c), with a logarithmic scale for the absolute value. Only in the exponential case were we able to observe a non-trivial sign structure with deviations from the Marshall sign rule, in which there were samples on the opposite site of the bulk.

Reh et al. arXiv:2301.067788

Fu et al. arXiv:2206.07370 Lattice Convolutional Networks for Learning Ground States of Quantum Many-Body Systems

Deep learning methods have been shown to be effective in representing ground-state wave functions of quantum many-body systems. Existing methods use convolutional neural networks (CNNs) for square lattices due to their image-like structures. For non-square lattices, existing method uses graph neural network (GNN) in which structure information is not precisely captured, thereby requiring additional hand-crafted sublattice encoding. In this work, we propose lattice convolutions in which a set of proposed operations are used to convert non-square lattices into grid-like augmented lattices on which regular convolution can be applied. Based on the proposed lattice convolutions, we design lattice convolutional networks (LCN) that use self-gating and attention mechanisms. Experimental results show that our method achieves performance on par or better than existing methods on spin 1/2 J_1-J_2 Heisenberg model over the square, honeycomb, triangular, and kagome lattices while without using hand-crafted encoding.

Lattice	Size	J_2	GNN	GNN-2	LCN (ours)	Reference Energy
Square	36	0	-0.6788*	-	-0.6788(1)	-0.678872†
		0.5	-0.5022(4)	-0.5023(5)	-0.5022(2)	-0.503810†
	100	0	-0.6708(0)	-	-0.6708(1)	0.671549(4)§
		0.5	-0.4955(4)	-0.4960(5)	**-0.4957(4)**	-0.497629(1)‡
Honeycomb	32	0	-0.551*	-	-0.5516(1)	-0.5517†*
		0.2	-0.4563(6)	-0.4564(7)	**-0.4563(1)**	-0.4567†*
	98	0	-	-	-0.5421(4)	-0.5448∞*
		0.2	-0.4528(2)	-0.4536(5)	**-0.4538(4)**	-0.4527∞*
Triangular	36	0	-	-0.55889	**-0.5601(4)**	-0.5603734†
		0.08	-0.5221*	-	**-0.5273(4)**	-0.5286†*
		0.125	-0.512(2)	-0.5131(8)	**-0.5126(6)**	-0.515564†
	108	0	-0.5508(8)	-0.5519(4)	-0.5475(4)	-0.551∞
		0.125	-0.500(9)	-0.5069(8)	**-0.5110(4)**	-0.5126¶
Kagome	36	0	-0.434(1)	-0.4338(6)	**-0.4367(4)**	-0.43837653†
		-0.02	-0.4339*	-	**-0.4384(5)**	-0.4399†*
	108	0	-0.4302(1)	-0.4314(7)	-0.4276(3)	-0.4386¶

Table 1: Estimated energy per site of the learned wave function (with error bars in parenthesis). Lower is better. Four kinds of lattice with various sizes are used for comparison. The J_2 value controls next nearest neighboring interaction on the lattice, resulting in differences in the ground states. The GNN-2 model doubles the parameters and computation by using separate branches for predicting the amplitude and argument parts of the wave function. Therefore, we directly compare our model against the single branch GNN model, and LCN results better than GNN are denoted in bold. We use $*$ to denote the results measured from plots in their reference papers. For the reference ground state energies we also annotate the employed methods: † for exact diagonalization, ∞ for infinite-size estimates, ‡ for RMB+PP, § for QMC methods, and ¶ for DMRG methods.

Topics for further discussion (among many)

- Comparison between architectures
- Implementation of symmetries
- Difficulties in representing ground-states with topological order (e.g. spin-1 Heisenberg/Haldane) and how to overcome them
- Connection between NN representation and Matrix Product States [Compression of wave functions in a broader perspective]– Several recent papers such as:

From Tensor Network Quantum States to Tensorial Recurrent Neural Networks

Dian Wu,* Riccardo Rossi,† Filippo Vicentini,‡ and Giuseppe Carleo§
Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland