

PAPER • OPEN ACCESS

## Deep learning representations for quantum many-body systems on heterogeneous hardware

To cite this article: Xiao Liang *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 015035

View the [article online](#) for updates and enhancements.

You may also like

- [Theoretical characterization of uncertainty in high-dimensional linear classification](#)  
Lucas Clarté, Bruno Loureiro, Florent Krzakala et al.
- [Entangled q-convolutional neural nets](#)  
Vassilis Anagiannis and Miranda C N Cheng
- [Mutual information scaling for tensor network machine learning](#)  
Ian Convy, William Huggins, Haoran Liao et al.



## PAPER

## OPEN ACCESS

RECEIVED  
2 November 2022REVISED  
13 March 2023ACCEPTED FOR PUBLICATION  
17 March 2023PUBLISHED  
30 March 2023

Original Content from  
this work may be used  
under the terms of the  
[Creative Commons  
Attribution 4.0 licence](#).

Any further distribution  
of this work must  
maintain attribution to  
the author(s) and the title  
of the work, journal  
citation and DOI.



# Deep learning representations for quantum many-body systems on heterogeneous hardware

Xiao Liang<sup>1</sup> , Mingfan Li<sup>2</sup>, Qian Xiao<sup>2</sup>, Junshi Chen<sup>2</sup>, Chao Yang<sup>3</sup>, Hong An<sup>2,\*</sup> and Lixin He<sup>1,4,\*</sup> <sup>1</sup> CAS Key Lab of Quantum Information, University of Science and Technology of China, Hefei, People's Republic of China<sup>2</sup> School of Computer Science and Technology, University of Science and Technology of China, Hefei, People's Republic of China<sup>3</sup> School of Mathematical Sciences, Peking University, Beijing, People's Republic of China<sup>4</sup> Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, People's Republic of China

\* Authors to whom any correspondence should be addressed.

E-mail: [han@ustc.edu.cn](mailto:han@ustc.edu.cn) and [helx@ustc.edu.cn](mailto:helx@ustc.edu.cn)**Keywords:** neural network quantum state, deep learning, stochastic reconfiguration, Lanczos, Sunway, heterogeneous architecture

## Abstract

The quantum many-body problems are important for condensed matter physics, however solving the problems are challenging because the Hilbert space grows exponentially with the size of the problem. The recently developed deep learning methods provide a promising new route to solve long-standing quantum many-body problems. We report that a deep learning based simulation can achieve solutions with competitive precision for the spin  $J_1$ – $J_2$  model and fermionic  $t$ - $J$  model, on rectangular lattices within periodic boundary conditions. The optimizations of the deep neural networks are performed on the heterogeneous platforms, such as the new generation Sunway supercomputer and the multi graphical-processing-unit clusters. Both high scalability and high performance are achieved within an AI-HPC hybrid framework. The accomplishment of this work opens the door to simulate spin and fermionic lattice models with state-of-the-art lattice size and precision.

## 1. Introduction

Strongly correlated quantum many-body physics is one of the most fascinating research fields in condensed matter physics. When a large number of microscopic particles interact with each other, quantum entanglement is built among them and exotic physical phenomena emerge, such as (high-temperature) superconductivity [1], quantum Hall effects [2], quantum spin-liquid [3], etc. Solving these problems will greatly deepen our understanding of the basic laws of the natural world and guide us to find novel physical phenomena and new quantum materials.

Although the basic rules of quantum mechanics are known, solving strongly correlated quantum many-particle problems is still extremely challenging. This is because the Hilbert space of the solution grows exponentially with the size of the problem. Furthermore, although the perturbative methods are very successful for simulating weakly correlated material systems, they fail for the strongly correlated systems. Revealing the fascinating physical natures in quantum many-body systems mainly relies on the non-perturbative numerical methods, such as exact diagonalization (ED), the quantum Monte Carlo method (QMC) [4] and the density matrix renormalization group (DMRG) [5].

However, these methods all have serious limitations: e.g. the computational cost of ED grows exponentially with the system size, and therefore the system size of the ED method is limited to less than 50 sites [6]. QMC suffers from the notorious sign problem for fermionic and frustrated systems [7]; and DMRG is limited to 1D or quasi-1D systems and does not work well for higher dimension systems [8]. To maintain the same accuracy, the computational costs of DMRG grow exponentially with the width of the lattice. Thus far, the width of DMRG simulation is limited to  $L \approx 12$  [9]. Very recently, the so-called tensor network methods, e.g. the projected entangled-pair states (PEPSs) method [10] have been developed, which can simulate fermionic and frustrated systems. For example, the PEPS can solve the frustrated  $J_1$ – $J_2$  model to a high accuracy with open boundary conditions (OBCs) [11, 12]. However, the periodic boundary condition

(PBC) usually has smaller boundary effects than OBC, and is therefore more suitable for simulating quantum many-body systems. However due to the high scaling to the bond dimension  $O(D^{18})$ , there is no report of PEPS simulation on PBC with the bond dimension  $D \geq 6$ , which is still too small to catch the essential physics of quantum many-body models.

Recently, the rapid progress of artificial intelligence powered by deep learning [13, 14] has attracted the attention of the scientific community on applying deep learning to solve scientific problems. For example, neural networks are used to solve differential equations [15], accelerate molecular dynamics [16], predict proteins' 3D structures [17], control nuclear fusion [18], and so on. There are also great efforts in applying deep learning methods to study quantum many-body problems.

Compared to traditional deep learning tasks such as classification, there are some major challenges in solving quantum many-body problems via neural networks, because one has to obtain an extremely high accuracy ground state in the exponentially large Hilbert space:

- The neural network's generalization ability should be high enough to represent the quantum state in the exponentially large Hilbert space.
- Double precision of the network parameters is mandatory.
- The ground energy should be the global minimum of the energies. The first-order-gradient based optimizers like Adam, SGD, etc are not efficient, as they can be easily trapped into local minimal. Here, a second-order natural-gradient-like method, such as the stochastic reconfiguration (SR) method is used.
- Finding the solution in the exponentially-large Hilbert space requires an extremely large amount of MCMC samples, which is also required for precise SR optimization.

Great efforts have been made to solve quantum many-body problems via deep learning methods [19, 20]. In 2017, Carleo *et al* solved the Heisenberg model via a restricted Boltzmann machine (RBM), with 323 200 network parameters and obtained rather high precision results on the  $10 \times 10$  square lattice [21]. For the frustrated system such as the spin-1/2  $J_1$ - $J_2$  Heisenberg model, complex network parameters are necessary. In 2018, a single-layered convolutional neural network (CNN) with 11 009 parameters was reported [22] to solved the frustrated  $J_1$ - $J_2$  model with satisfactory precision. To further improve the energy precision on the  $10 \times 10$  lattice, the Gutzwiller projection fermionic wave-function with the RBM [23] and the deep CNN with a sign rule [24] were used. In 2021, on the Fugaku supercomputer, the pair product states with a RBM has achieved high accuracy on the lattice as large as  $18 \times 18$  [25]. In 2022, by increasing the deep CNN parameter number to 106 529, the energy precision is greatly improved and the lattice size is increased to  $24 \times 24$  [26]. Since that various quantum states can be represented by CNN based wavefunctions [27], the quantum state representation is named the convolutional neural quantum state (CNQS).

The Fermion models are even more challenging. In 2019, by employing CNN based Jastrow and sign corrections, the spinless Fermions model was investigated on a  $10 \times 10$  lattice [28]. There are several investigations for the Fermi-Hubbard model. In 2019, by employing a fully-connected neural network (FCNN) in the Slater determinant, the model was studied on a  $4 \times 4$  lattice [29]. In 2021, a FCNN was used as the determinant free wavefunctions to study the model on a  $6 \times 6$  lattice [30]. In 2022, the RBM ansatz solved the Hubbard model on the lattice as large as  $4 \times 16$  [31].

In this work, we benchmark our methods on two important and representative models, namely the  $J_1$ - $J_2$  frustrated spin model and the  $t$ - $J$  fermion model. Due to the quantum correlation, the degeneracy of the ground state is broken. However, there could be many competing states that have energies very close to the ground state energy. Therefore these models are extremely challenging. Both models are investigated on the rectangular lattices with PBC. We firstly optimize the CNN wavefunction through SR, then the wavefunctions are further optimized through a Lanczos step [32]. We develop an hybrid artificial intelligence-high performance computing (AI-HPC) framework to implement the optimizations, and achieve high scalability and high performance on two typical HPC platforms, the new generation Sunway supercomputer and the multi graphical-processing-unit (GPU) clusters. Furthermore, the optimization results show that the CNN based wavefunctions can achieve competitive energy precision on unprecedented lattice sizes.

The  $J_1$ - $J_2$  model is a candidate model for the quantum spin liquids [12, 25], and the Hamiltonian reads:

$$\hat{H} = J_1 \sum_{\langle i,j \rangle} \mathbf{s}_i \cdot \mathbf{s}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \mathbf{s}_i \cdot \mathbf{s}_j \quad (1)$$

where  $\langle i,j \rangle$  and  $\langle\langle i,j \rangle\rangle$  indicate the nearest and next-nearest neighbouring spins pairs.  $\mathbf{s}_i$  is the spin operator on the  $i$ th site. We set  $J_1 = 1$  and  $J_2 = 0.5$  throughout the investigations.

The  $t$ - $J$  is a basic model for the superconductivity [33], and the Hamiltonian is:

$$\hat{H} = -t \sum_{\langle i,j \rangle, \sigma} (c_{i,\sigma}^\dagger c_{j,\sigma} + h.c.) + J \sum_{\langle i,j \rangle} \left( \mathbf{s}_i \cdot \mathbf{s}_j - \frac{n_i n_j}{4} \right) \quad (2)$$

where  $\langle i,j \rangle$  indicate the nearest neighbouring pairs. The operator  $c_{i,s}^\dagger$  creates an electron of spin  $\mathbf{s}$  on site  $i$ ,  $\mathbf{s}_i$  and  $n_i$  are the electron spin moment and charge density operators on site  $i$ . We encode the Fermions into spins via Jordan–Wigner transformation [34]. In the simulations,  $t = 1$ ,  $J = 0.4$  and the hole doping  $n_h = 0.125$  are used.

## 2. Methods

### 2.1. The framework of optimizing the neural network

Taking the quantum spin model as an example, a quantum many-body state is represented by a superposition of the spin configurations,

$$|\Psi\rangle = \sum_S W(S)|S\rangle \quad (3)$$

where  $|S\rangle = |s_1^z, s_2^z, \dots, s_N^z\rangle$  is a set of spins with one on each site of the lattice, and  $N$  is the total number of sites. If the degree of freedom on each site is  $k$ , the Hilbert space (i.e. the number of  $|S\rangle$ ) of the problem is  $k^N$ . The weight of the spin configuration  $W(S)$  is represented by a neural network. The network parameters are optimized to determine the ground state of the system, i.e. the state of the lowest energy  $E_{g.s} = \min \langle \Psi | \hat{H} | \Psi \rangle / \langle \Psi | \Psi \rangle$ , where  $\hat{H}$  is the Hamiltonian of the system.

The self-learning procedure for optimizing the neural network is depicted in figure 1. First, Markov-Chain-Monte-Carlo (MCMC) is performed according to  $|W(S)|^2$ . When collecting a sample, the forward of the neural network gives local energy  $E_{loc}$  and the backward of the neural network gives local derivative  $O_{loc}$ . Secondly, the gradients and the covariance matrix needed for the SR optimization method [35] are calculated. Finally, the network parameters are updated, and the values for updating parameters are  $\delta$ .

After the SR minimization, the wavefunctions can be further optimized by a Lanczos step:

$$|\Psi_p\rangle = \alpha_p |\Psi_{p-1}\rangle + \beta_p |\Psi_{p-1}^\perp\rangle, \quad (4)$$

where  $\alpha_p$  and  $\beta_p$  are the parameters to be determined, and  $|\Psi_0\rangle$  is the variational wavefunction after SR optimization.  $|\Psi_p^\perp\rangle$  is orthogonal to  $|\Psi_p\rangle$ :  $|\Psi_p^\perp\rangle = \frac{1}{\sigma_p} (\hat{H} - E_p) |\Psi_p\rangle$ , where the energy  $E_p = \langle \Psi_p | \hat{H} | \Psi_p \rangle$  and the variance  $\sigma_p^2 = \langle \Psi_p | \hat{H}^2 - E_p^2 | \Psi_p \rangle$ . For the state  $|\Psi_p\rangle$  with energy  $E_p$  and variance  $\sigma_p$ , it is easy to prove that  $E_p \approx E_{\text{exact}} + \text{const.} \times \sigma_p^2$ . Here we consider the average energy per site  $E = E_p/N$  and the variance per site  $\sigma^2/N$ :  $E \approx E_{\text{exact}} + \text{const.} \times \sigma^2/N$ . [32]

### 2.2. Neural network method details

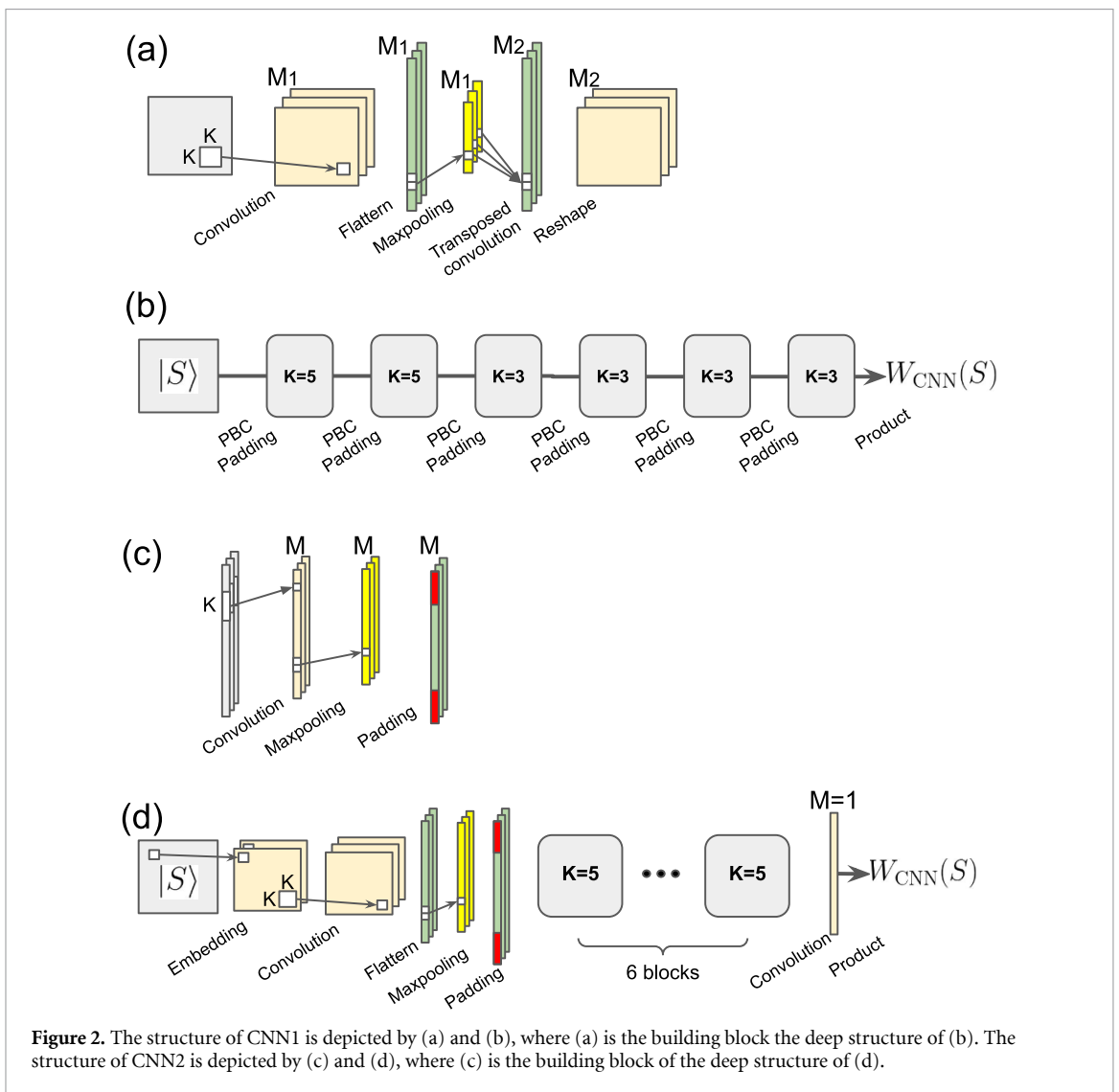
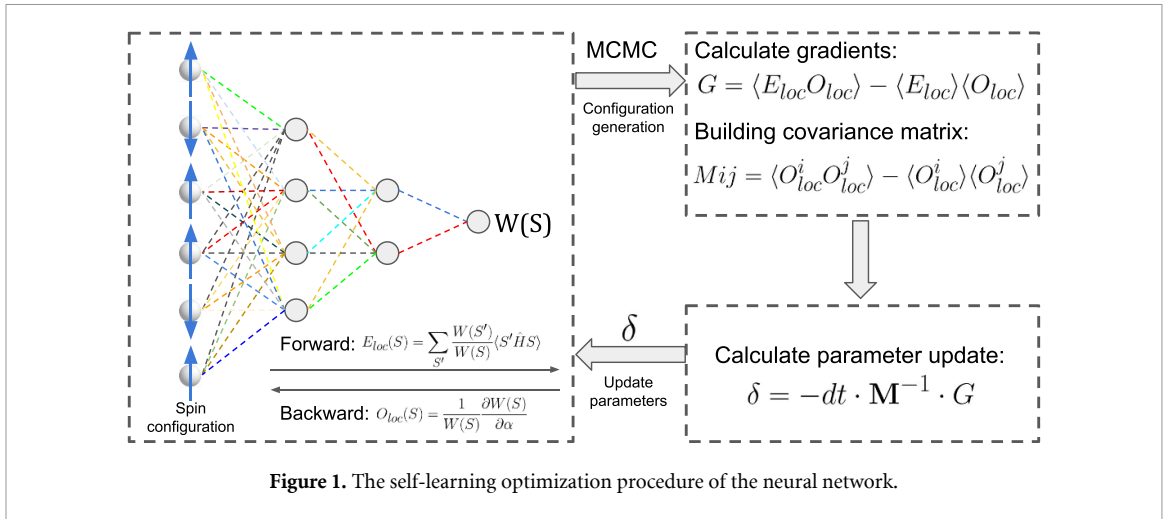
#### 2.2.1. Neural network structure

To meet the strictly demanding requirement, we develop two deep CNN structures for the spin models and fermion models, which are denoted as CNN1 and CNN2 respectively. The computational complexity of CNN is much lower compared to the fully connected structure. By taking advantage of translational invariance, the CNN can scale to very large lattices.

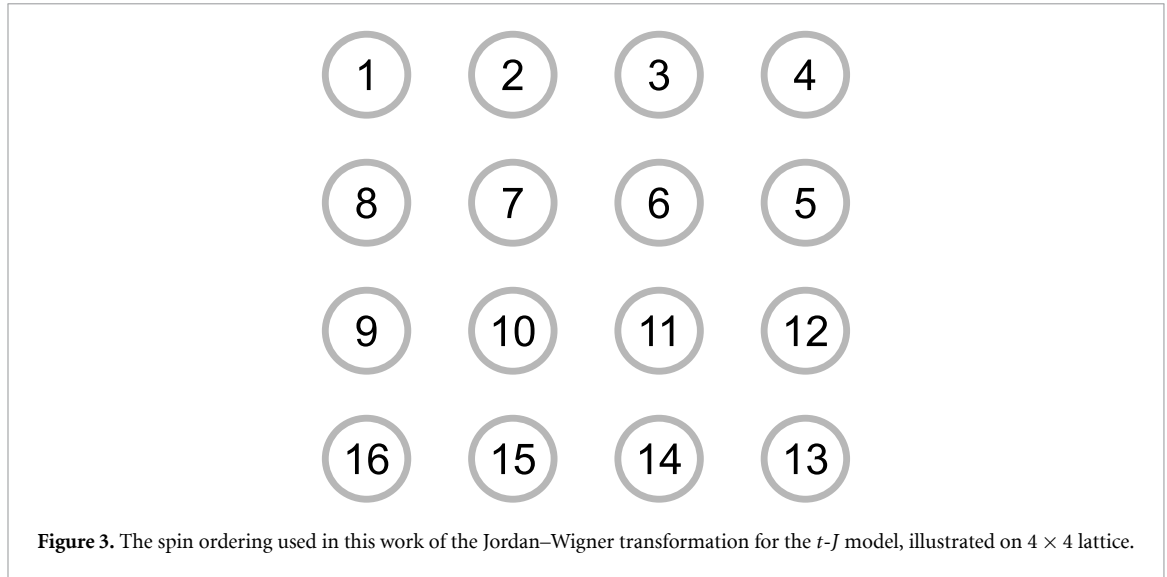
The structure of CNN1 is depicted in figures 2(a) and (b), and the network is built by stacking the building block denoted in figure 2(a) six times [36]. A building block consists of two-dimensional convolution and one-dimensional maxpooling and transposed-convolution. To maintain the dimensions, paddings are employed on the input spin lattice and between two building blocks. The padding scheme is based on the PBCs. The final outputs are the products of the neurons, which are the wavefunction coefficients for the input spin configurations.

The structure of CNN2 is depicted in figures 2(c) and (d). CNN2 originates from CNN1, with two major modifications: (1) the input spin lattice is processed by an embedding layer; (2) the one-dimensional maxpooling is performed with stride one. To maintain the dimensions, padding is performed by copying the first several neurons to the last. The deep structure is built by stacking the building block in figure 2(c) six times. Finally, with a convolution operation, the channel number is decreased to one, the final output is the product of the neurons.

The deep CNN structures used in this work have several important differences compared to other CNN structures for quantum many-body problems [21, 24, 28]. First, the nonlinearity in our deep CNN is



induced by the maxpooling, instead of traditional activation functions. The maxpooling picks up the most important degree of freedom in a convolution filter, which is similar to the coarse-grained process in a renormalization group theory [37]. Second, the wavefunction coefficients are generated by the products of neurons, thus the CNN can give the  $\pm 1$  signs within real network parameters, which differs from the exponential function in RBM based structures.



**Figure 3.** The spin ordering used in this work of the Jordan–Wigner transformation for the  $t$ - $J$  model, illustrated on  $4 \times 4$  lattice.

### 2.2.2. Neural network wavefunction representation

The CNN1 structure in this work is based on the deep CNN previously reported in [26, 36], and the wavefunction is preconditioned by the Marshall-sign-rule (MSR)[38, 39]:

$$\tilde{W}(S) = (-1)^{MA} W_{\text{CNN1}}(S), \quad (5)$$

where MA is the magnetization on the equivalent sublattice A. Although the MSR is only exact for the case of  $J_2 = 0$ , however, because the CNN1 is able to give the  $\pm 1$  signs, it is able to correct the MSR for large  $J_2$ .

Enforcing symmetries can significantly reduce the optimization difficulty [40]. In the case of  $J_2 = 0.5$ , the final wavefunction coefficient has an enforced rotational symmetry on the square lattice [24]:

$$W(S) = \sum_i \tilde{W}(\hat{T}^i S), \quad (6)$$

where  $\hat{T}$  is the spin rotation operator and  $i = 0, 1, 2, 3$ .

Based on the structure of CNN1, we design CNN2 to solve the  $t$ - $J$  model. To encode the Fermions into spins, Jordan–Wigner transformation is used. Figure 3 denotes the spin ordering for the JW transformation on  $4 \times 4$  square lattice. We have compared several spin orderings on the  $8 \times 8$  square lattice, and the spin ordering denoted in figure 3 gives the lowest variational energy.

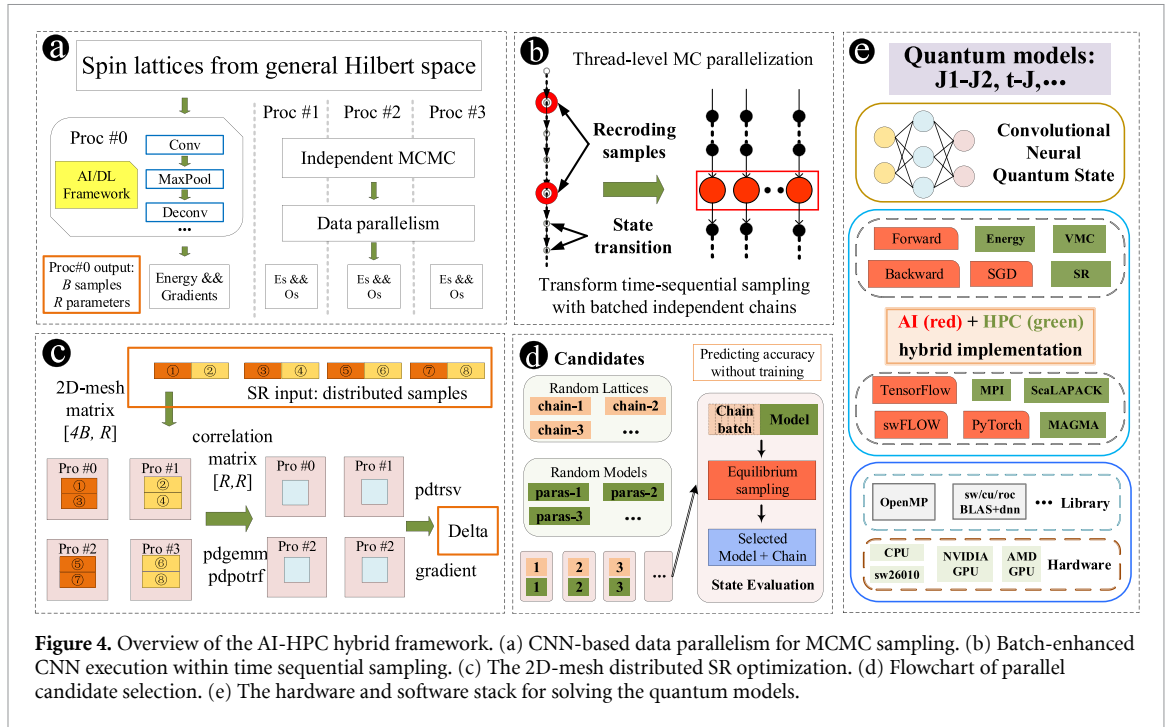
There are three degrees of freedom in each site for the  $t$ - $J$  model, which correspond to no-occupation, one-occupation with spin up and spin down. The Hilbert space of the  $t$ - $J$  model is much larger compared to spin models, thus a higher representation ability of the CNN is required. To enhance the representation ability, an embedding layer is employed to transfer the input of each degree of freedom to a trainable two-element vector. Even though the original Hamiltonian of the  $t$ - $J$  model has rotational symmetry, the JW transformation breaks this symmetry, and we do not enforce any additional symmetry of the CNN wave-function, and the final wave-function coefficient is directly generated from the CNN2:  $W(S) = W_{\text{CNN2}}(S)$ .

### 2.2.3. Transfer learning

For the  $J_1$ - $J_2$  model, direct training of the model in large lattice starting from random parameters is very difficult. The convolution operation is intrinsically scalable for different lattice sizes. Furthermore, the energy convergence on a small lattice is fast, as the lattice size increases, the ground energy will converge with respect to the lattice size. Initially the network is trained on  $6 \times 6$  lattice from randomly initialized parameters. The network parameters are then checkpointed and used for optimizing the larger  $10 \times 10$  system as the pretrained model. With multiple stages of transfer learning, we finally increased the lattice size to  $36 \times 36$  and obtained a good initial energy. Thanks to transfer learning, the local optimizations are avoided and the training steps for large lattices are significantly reduced.

### 2.2.4. Parallel initial state selection

For the  $t$ - $J$  model, a proper initial model parameter and the initial spin configuration are beneficial for avoiding local optimizations, and therefore crucial for fast energy convergence. Here we calculate the energy



**Figure 4.** Overview of the AI-HPC hybrid framework. (a) CNN-based data parallelism for MCMC sampling. (b) Batch-enhanced CNN execution within time sequential sampling. (c) The 2D-mesh distributed SR optimization. (d) Flowchart of parallel candidate selection. (e) The hardware and software stack for solving the quantum models.

of a randomly initialized CNN by performing MCMC sampling, assuming that the CNN with lower initial energy will have a better convergence after SR optimizations. With a preselection stage, the CNN parameters together with the initial spin configuration that has the lowest energy result are selected. After the initial state selections, the optimization steps are significantly reduced especially on the lattice as large as  $12 \times 12$ .

### 2.3. AI-HPC hybrid parallel optimization framework

To optimize the CNN wavefunctions in an efficient and scalable way, we propose an AI-HPC hybrid framework to implement the algorithm introduced in figure 1, and the framework is depicted in figure 4. Overall, the AI-HPC hybrid framework can be divided into two parts: MCMC sampling and SR. The first part can be seen as data parallelism, where different processes hold the same model parameters but compute different input data. The second part is responsible for handling MCMC samples and computing the parameter update  $\delta$  of network parameters in SR by constructing a large covariance matrix.

#### 2.3.1. Process- and thread-level parallel MCMC

Different from the distributed data parallel training in conventional deep learning tasks, the proposed AI-HPC framework only holds the same model state, while it replaces training data with on-the-fly MC sampling. In principle, importance sampling in the Hilbert space is achieved by multiple independent Markov chains. Considering that the MC sampling tasks are intrinsically parallel, they can be distributed across all participating processes, where the parallel execution among the 4 processes is pictured in figure 4(a). Inside each process, there is an independent chain of MC sampling, which generates  $E_{loc}$  and  $O_{loc}$  with model forward and backward computations. In the figure, each process produces  $B$  samples, where  $R$  equals the number of network parameters. The input data is naturally distributed along with Markov chains. Implementing the process-level parallel sampling only requires broadcasting the initial parameter number and the parameter update.

The MCMC sampling is time sequential, and numerous state transitions exist between two collected samples. The state transition along the Markov chain requires little computation, which leads to significant resource under-utilization, especially for today's powerful heterogeneous devices. In this case, an alternative by managing multiple independent chains in one process is proposed, as depicted in figure 4(b). In each step, one sampling step from many chains increases the batch size during the model computation. The CNN computation is further divided into multiple computing layers and then offloaded onto heterogeneous many-core processor with modest thread-level parallel optimization. The repeated CNN execution with varying lengths can be organized with the fixed-length batch to improve performance. Additionally, such method theoretically increases the sample independence, which may help faster model convergence comparing to sampling from a single chain.



The flexible parallel MCMC scheme (i.e. MPI-parallel among different processes and batch-parallel among independent chains) provides an alternative to navigate the aforementioned initial state selection in section 2.2.4. To adapt to the two-level MCMC sampling, the selection paradigm is described in figure 4(d). The parallel processes is designed to find suitable initial model parameters, while the chain batch inside each process aims to find the best initial spin lattice. Taking the advantage of HPC infrastructure to simultaneously examine millions of sampling results, it dramatically offsets the difficulty in finding a reasonable initial configuration in the exponentially large Hilbert space.

### 2.3.2. Distributed SR computation

With sufficient samples collected from MCMC, the values for updating network parameters  $\delta$  are obtained through SR. The procedure for computing  $\delta$  includes three steps: (1) adjusting the data format; (2) constructing the covariance matrix; and (3) solving a system of linear equations.

After MCMC sampling, the input for the SR procedure is the evenly distributed among all participating processes. These samples are then organized in a 2D-mesh style. For  $P$  processes, the collected samples are divided into  $\sqrt{P}$  blocks. Naturally, the processes are split into  $\sqrt{P}$  groups, where each group contains  $\sqrt{P}$  processes. The data block is scattered into corresponding processes within the group.

Figure 4(c) denotes an example of 4 processes (Proc-0 and Proc-1 in group-1, Proc-2 and Proc-3 in group-2). The block exchange between 2 processes can be illustrated by swapping blocks of ② and ③ in group 1 (⑥ and ⑦ in group 2). As depicted in the figure, for a CNN with the parameter number  $R$ , it generates the 2D-mesh distributed matrix  $[4B, R]$ . The following covariance matrix  $[R, R]$  is constructed in parallel. The final  $\delta$  is obtained by combining the matrix with the first-order gradients of the energy. Similar to data parallelism training, all processes shares the same  $\delta$  by MPI broadcast.

The scalability of SR is limited on constructing and solving the covariance matrix. In previous investigations, small network parameter number achieves moderate energy precision [36]. In this work, the network parameter number is further increased to the magnitude of  $10^5$ . With the help of modern parallel programming paradigm and high performance heterogeneous devices, accurately constructing and storing the full covariance matrix is still feasible. However, there still exists scalability issue as the parameter number continues to increase. One solution is to adopt approximate solutions, such as the sparse distributed solver as introduced in NetKet [41] and hybrid low-rank natural gradient method in [42]. Investigating the performance of these highly scalable sparse algorithms is one of our future directions.

### 2.3.3. Hybrid implementation in AI and HPC

Figure 4(e) depicts the proposed AI-HPC software stack in solving CNQS. In this work, CNQS is built by a series of common deep learning operators, such as *Conv*, *Deconv*, *Maxpool* and *Embedding*.

The gradients of CNQS are obtainable through the MCMC. As introduced in section 2.1, the forward and backward calculations of the CNN give the local energy and the local derivative during the MCMC. With the gradients, it is straightforward to use optimizers, such as SGD, Adam, etc. The forward and backward computation of CNN, as well as the optimizers are supported by mainstream deep learning libraries, such as TensorFlow and PyTorch, etc. The SR method is handled by the MPI and ScaLAPACK, which further supports heterogeneous devices through the MAGMA library.

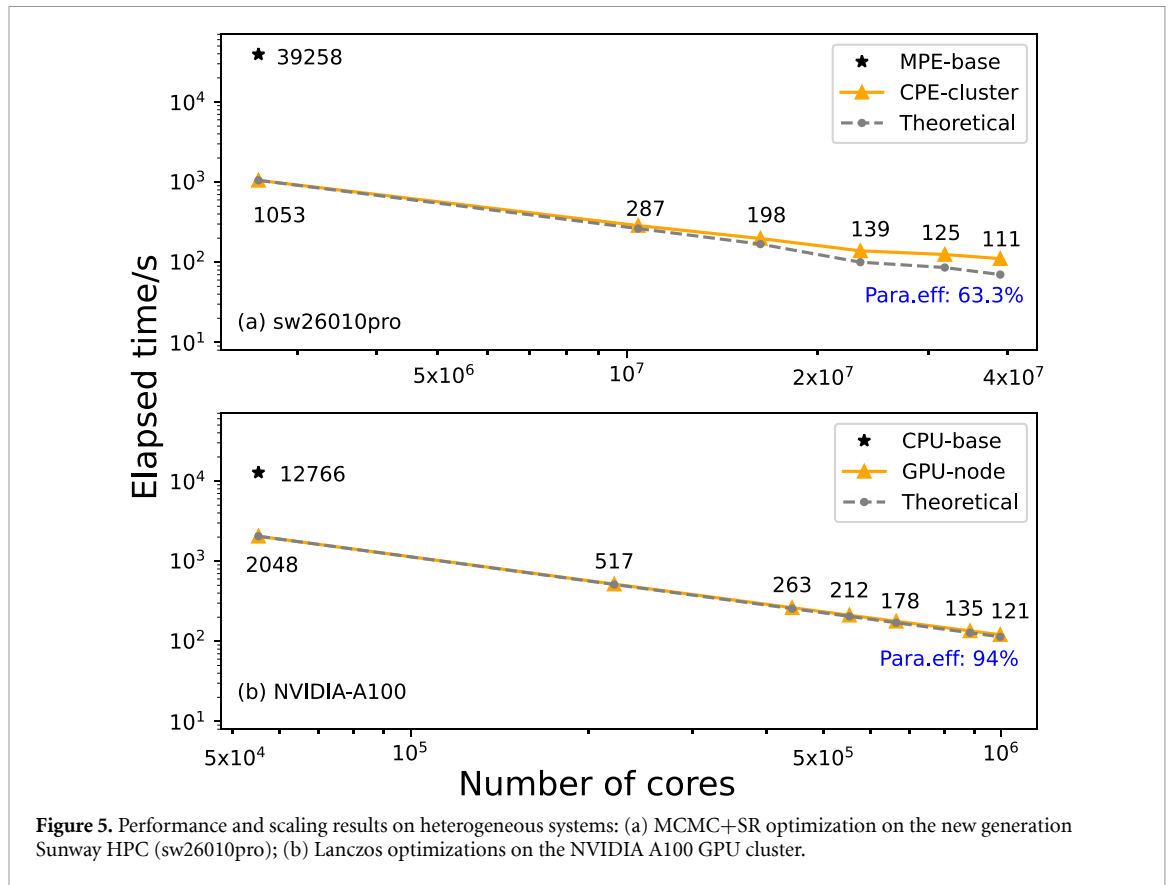
The computation of CNN can be significantly accelerated by heterogeneous hardware, such as Intel Xeon Phi, NVIDIA (AMD) GPUs, sw26010pro CPUs, etc. The sw26010pro many-core processor is the upgraded version of sw26010 released with Sunway TaihuLight [43–45]. Taking sw26010pro as an example, there are six core groups on one chip. In each core group, there is one MPE (management process element) that acts as control elements and 64 CPEs (computing process elements) that act as computing elements. By offloading the computation into the CPEs, a high acceleration ratio is achieved for deep learning operators, e.g. *Conv* and *Deconv* [26].

Thanks to the modern programming paradigm and flexible libraries (i.e. sw/cu/roc-dnn or BLAS), the optimization of CNQS can be easily adapted onto both the traditional CPU and diverse heterogeneous accelerators.

## 3. Results

In this section, we demonstrate the computational efficiency and parallel scalability of the proposed hybrid AI-HPC framework on two typical heterogeneous systems. Specifically, two optimization algorithms (i.e. MCMC+SR and Lanczos elaborated in this paper) are thoroughly evaluated on the new generation of Sunway supercomputer and NVIDIA A100 cluster, respectively. Meanwhile, by solving two long-standing quantum models: the  $J_1$ – $J_2$  model and  $t$ - $J$  model, the competitive results of variational energies demonstrate the effectiveness of the scalable deep learning method.





**Figure 5.** Performance and scaling results on heterogeneous systems: (a) MCMC+SR optimization on the new generation Sunway HPC (sw26010pro); (b) Lanczos optimizations on the NVIDIA A100 GPU cluster.

### 3.1. Performance and Scalability on heterogeneous architecture

The optimization of the CNN based wavefunction is achieved in two stages: first MCMC+SR optimization, and then Lanczos optimization. Here, we show that the computation of CNNs can be significantly accelerated by the heterogeneous cores, and the optimization of CNN based wavefunctions is perfectly scalable on the HPC platform. For example, MCMC+SR has high scalability among up to 40 million heterogeneous cores on the new generation Sunway supercomputer, and the Lanczos optimization has perfect scalability demonstrated in multi-GPU systems.

#### 3.1.1. Performance enhancements by CPEs

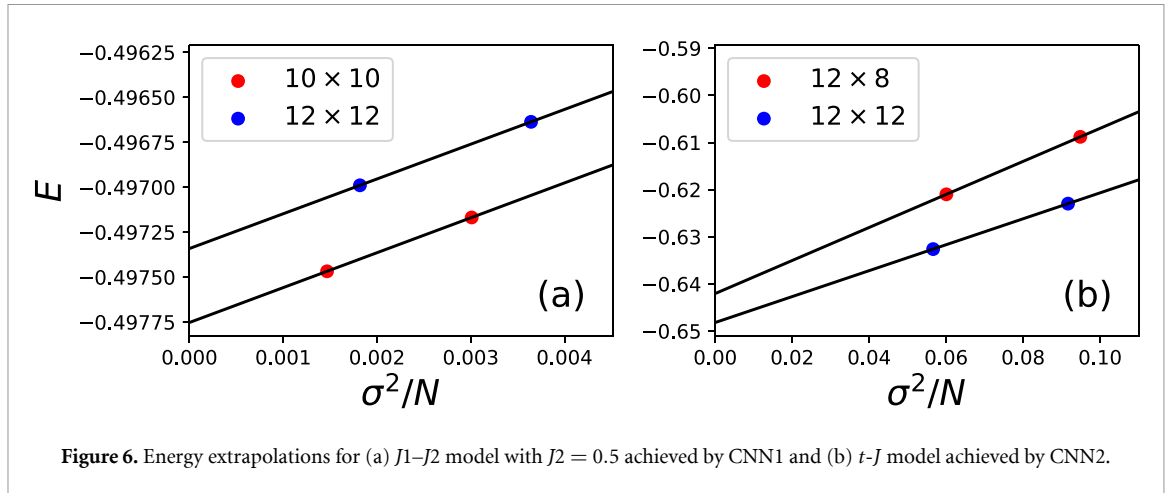
For the sw26010pro CPU delivered with the new generation Sunway supercomputer, most computing power for each core group is provided by the 64 CPEs. Therefore it is important to utilize the CPEs to achieve high performance. Taking the compute-intensive operators used in CNN1 and CNN2 as examples, the many-core acceleration ratios for *Conv2D*, *Conv2DBackpropInput* and *Conv2DBackpropFilter* are  $538\times$ ,  $211\times$  and  $505\times$ , respectively. Comparing to the compute-intensive operators, the many-core acceleration ratios of memory-intensive operators are limited due to the inability to achieve the same high CPE utilizations. The many-core acceleration ratio reaches  $33.21\text{--}35.93$  for *Tile* and  $34.2\text{--}44.8$  for *Slice*.

With the performance optimized operators, the overall acceleration ratio achieves considerable performance improvement over original MPE version. The many-core acceleration ratio for the CNN1 with 106 529 and 421 953 parameters are  $90\times$  and  $130\times$ , respectively.

#### 3.1.2. Performance and scalability results

For the MCMC+SR implemented on Sunway HPC, the elapsed time for one optimization step with respect to the core number is depicted in figure 5(a). In this case,  $J_2 = 0.5$  and  $L = 10$ , the total sample number is about  $5 \times 10^6$  and the network parameter number is 106 529. Under 40 000 processes, the elapsed time of one MCMC+SR step requires 1053 s with 64 CPEs acceleration, while the elapsed time for the MPE version is over 39 258 s. Therefore an acceleration ratio of approximately 37 times is achieved by utilizing the CPE cluster, compared to the MPE only. The MCMC+SR can be easily scaled to the whole system, with a maximum of 608 400 processes. It reports nearly  $10\times$  throughput improvement and achieves 63% parallel efficiency when scaling to nearly 40 million cores.

For the Lanczos optimization, it is necessary to calculate the high order expectation of Hamiltonian like  $\langle \hat{H}^2 \rangle$  and  $\langle \hat{H}^3 \rangle$ . Figure 5(b) depicts the elapsed time for calculating  $\langle \hat{H}^2 \rangle$  with respect to the NVIDIA A100



CUDA core number. In this case  $J_2 = 0.5$  and  $L = 8$ , the total sample number is 30 240 and the total number of spin configurations is on the order of magnitude of  $30\,240 \times L^4$ . Different from the SR stage for handling the distributed covariance matrix, the Lanczos optimization only requires substantial CNN computation, which incurs little communication. Consequently, the forward calculations of all spin configurations are perfectly scalable to all the GPUs. As depicted by the figure, the parallel efficiency on 995 328 CUDA cores (144 GPUs) is 94%. In addition to the high scalability, for the double precision calculation of CNN wavefunctions, the speedup ratio of 55 296 CUDA cores (8 GPUs) to 640 CPU cores is 6.23.

Therefore, the optimization of CNN based wavefunctions has a high scalability and can achieve high acceleration ratio on heterogeneous systems.

### 3.2. Simulation results

#### 3.2.1. Ground state energies

Based on the linear relationship between energy and variance introduced in section 2.1:

$E \approx E_{\text{exact}} + \text{const.} \times \sigma^2/N$ , figure 6(a) reveals the energy extrapolations achieved by CNN1 within 106 529 parameters for the  $J_1$ - $J_2$  model, and figure 6(b) reveals the energy extrapolations achieved by CNN2 within 113 815 parameters for  $t$ - $J$  model. Note that the extrapolated values are used only for estimation purposes, as only two data points are used.

As depicted by figure 6(a), for the  $J_1$ - $J_2$  model in the maximal frustration region  $J_2 = 0.5$ , the variational energies on the  $10 \times 10$  lattice for  $p = 0$  and  $p = 1$  are  $-0.497\,168$  and  $-0.497\,468$  respectively, and the extrapolated energy is  $-0.497\,752$ . Furthermore, on the  $12 \times 12$  lattice, the variational energies for  $p = 0$  and  $p = 1$  are  $-0.496\,636$  and  $-0.496\,989$  respectively, and the extrapolated energy is  $-0.497\,341$ .

As depicted by figure 6(b), for the  $t$ - $J$  model with hole doping  $n_h = 0.125$ , the variational energies on the  $12 \times 8$  lattice for  $p = 0$  and  $p = 1$  are  $-0.608\,793$  and  $-0.620\,9887$  respectively, and the extrapolated energy is  $-0.642\,031$ . On the  $12 \times 12$  lattice, the variational energies for  $p = 0$  and  $p = 1$  are  $-0.622\,963$  and  $-0.632\,605$  respectively, and the extrapolated energy is  $-0.648\,179$ . From above results, we see that the Lanczos step can further improve the ground state energy.

The CNN1 used in this work has high quantum state representation ability, and therefore is more efficient comparing to other kinds of neural networks [46]. For example, for the  $10 \times 10$  lattice with  $J_2 = 0.5$ , the ground state energy obtained by the CNN1 with 16 443 parameters without Lanczos step (i.e.  $p = 0$ ) is  $-0.496\,687$ , which is lower than  $-0.4960$  obtained by the GNN-2 and the  $-0.4957$  obtained by the lattice convolutional network (LCN) reported in [46]. Remarkably, the number of parameters of both GNN-2 and LCN is on the order of a million, which is significantly higher than that of CNN1.

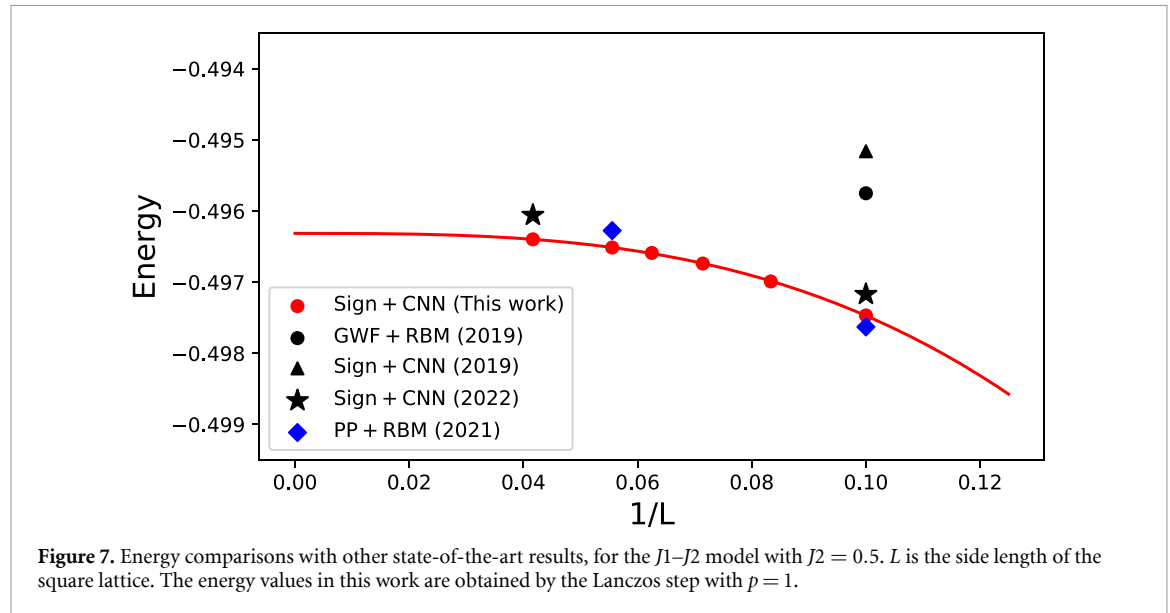
#### 3.2.2. Comparing to previous results

Here we present the competitive energy results of the  $J_1$ - $J_2$  model and  $t$ - $J$  model achieved in this work, and compare them to other state-of-the-art results. The energies are obtained by the CNN wavefunctions with a Lanczos step.

For the  $J_1$ - $J_2$  model, the energy values achieved by CNN1 with 106 529 parameters for various  $J_2$  and lattice sizes are listed in table 1. The energies are obtained by CNN1 optimized by MCMC+SR, and a Lanczos step. Specifically, in the maximal frustration region  $J_2 = 0.5$ , the convergence of energy with respect to the square lattice size  $L$  is depicted in figure 7 by red dots. In the figure, the solid line is fitted with the function  $E = a/L^3 + b$ . Other state-of-the-art results are also shown in the figure for comparison. When  $L = 10$ , the

**Table 1.** Energy values for various  $J_2$  values and lattice sizes, achieved by CNN1 with 106 529 parameters. The values are obtained by the Lanczos step with  $p = 1$ .

Description	0.46	0.48	0.50	0.55	0.60
$10 \times 10$	-0.507 44	-0.502 31	-0.497 47	-0.486 70	-0.478 39
$12 \times 12$	-0.506 97	-0.501 83	-0.496 99	-0.486 03	-0.477 15
$14 \times 14$	-0.506 75	-0.501 59	-0.496 74	-0.485 81	-0.476 78
$16 \times 16$	-0.506 62	-0.501 47	-0.496 59	-0.485 65	-0.476 56
$18 \times 18$	-0.506 51	-0.501 39	-0.496 51	-0.485 57	-0.476 52



energy in this work is  $-0.497\,468$ , which has a small distance comparing to the  $-0.497\,629$  achieved by the PP+RBM method [25]. Increasing CNN parameter number and more than once independent optimization can further improve the energy precision. However, because of transfer learning, the energy obtained in this work for  $L = 18$  is  $-0.496\,513$ , which is  $4.5 \times 10^{-4}$  lower than the  $-0.496\,275$  reported in [25].

To date, there are no reference ground state energies for the  $t$ - $J$  with PBC, therefore we compare the results to those of OBC calculated via PEPS in [47]. The parameter number of CNN2 is 113 815. On  $8 \times 8$  lattice, the energy achieved by CNN2 is  $-0.60\,958$ , which is  $3 \times 10^{-3}$  higher than that reported in [47]. On  $12 \times 12$  lattice, the energy achieved by CNN2 is  $-0.63\,261$ , which is  $3.8 \times 10^{-3}$  lower than that reported in [47]. We would like to note that the energies of different boundary conditions cannot be compared directly, which may have small differences. The results show here are only a demonstration of the effectiveness of CNN2 for fermionic lattice systems.

#### 4. Discussion and outlook

In this work, we report simulations of challenging quantum many-body problems that combine the state-of-the-art deep-learning method and efficient implementation on the heterogeneous computational platform. We have obtained the ground state energies of the  $J_1$ - $J_2$  model and  $t$ - $J$  model with accuracy competitive to current state-of-the-art results. With this ability, we are able to simulate a large class of important physical models with high accuracy to make confident conclusions about the physics. The method therefore opens up a new promising path to solve the extremely important and challenging problems that have plagued people for more than half a century, and would help us to understand exotic physical phenomena, such as quantum spin liquids, high temperature superconductivity [48], supersolid [49], heavy fermions [50], fractional quantum Hall effects, [51–53] and many more, which is not only essential for the understanding of fundamental physics but also have many important applications.

On the other hand, this work demonstrated the effectiveness of solving the quantum many-body problem via deep learning methods on the supercomputer platforms, which is completely different from traditional machine learning tasks. These applications represent a large class of scientific problems that are more challenging in the representation ability of the network and the accuracy requirements of the results. It also puts forwards higher requirements for efficient optimization algorithms and implementations. This

framework may apply to other problems, e.g. the simulation of quantum circuits on classic supercomputers. Currently tensor network methods have shown high efficiency for quantum circuit simulations on supercomputers [45] on the lattice as large as  $10 \times 10$ . The scalable neural network based quantum state representation may be capable of performing quantum circuit simulations for an even larger number of qubits.

Our MCMC+SR optimization framework is not limited to Sunway, and it can be easily ported to other supercomputing platforms. However, as the computational power increases, the overall application performance will be limited by the gap between the computing capability and the memory access bandwidth. Here we take the computational power v.s. memory bandwidth ratio (FLOP/Byte) to measure the potential performance. Because the single precision does not meet the requirements of achieving faithful ground state energies, we focus here on the double precision FLOP/Byte. With the double precision support of tensor cores, the FLOP/Byte ratio on an A100 GPU is 9.56 (19.5 TFLOPS/2039 GB/s = 9.56 FLOP/Byte). The double precision FLOP/Byte ratio on a Fujitsu A64FX CPU is 3.3 (3.3792 TFLOPS/1024 GB/s = 3.3 FLOP/Byte). Considering that the performance of the LDM used in sw26010pro is not competitive with the performance of HBM2 used in A100 and A64FX, therefore the performance of our framework can be further boosted on other HPC systems.

### Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://doi.org/10.5281/zenodo.6976345> and <https://doi.org/10.5281/zenodo.6975696>.

### Acknowledgment

This work is supported financially by the National Key Research and Development Program of China (Grant Numbers: 2016YFB1000403, 2017YFB0202002) and the Innovation Program for Quantum Science and Technology (Grant Number: 2021ZD0301200, Hefei National Laboratory). Numerical calculations were performed partly on the new generation Sunway supercomputer, and partly on the GPU instances in the supercomputing center of USTC.

### ORCID iDs

Xiao Liang  <https://orcid.org/0000-0002-7882-3571>

Lixin He  <https://orcid.org/0000-0003-2050-134X>

### References

- [1] Zhou X, Lee W-S, Imada M, Trivedi N, Phillips P, Kee H-Y, Törmä Paivi and Eremets M 2021 High-temperature superconductivity *Nat. Rev. Phys.* **3** 462–5
- [2] Wen X-G 1992 Theory of the edge states in fractional quantum Hall effects *Int. J. Mod. Phys. B* **6** 1711–62
- [3] Broholm C, Cava R J, Kivelson S A, Nocera D G, Norman M R and Senthil T 2020 Quantum spin liquids *Science* **367** eaay0668
- [4] Anderson J B 2007 *Quantum Monte Carlo: Origins, Development, Applications* (Oxford: Oxford University Press)
- [5] White S R 1992 Density matrix formulation for quantum renormalization groups *Phys. Rev. Lett.* **69** 2863–6
- [6] Savary L and Balents L 2017 Quantum spin liquids: a review *Rep. Prog. Phys.* **80** 016502
- [7] Troyer M and Wiese U-J 2005 Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations *Phys. Rev. Lett.* **94** 170201
- [8] Schollwöck U 2011 The density-matrix renormalization group in the age of matrix product states *Ann. Phys., NY* **326** 96–192
- [9] Wang L and Anders W S 2018 Critical level crossings and gapless spin liquid in the square-lattice spin-1/2  $J_1 - J_2$  Heisenberg antiferromagnet *Phys. Rev. Lett.* **121** 107202
- [10] Verstraete F, Murg V and Cirac J I 2008 Matrix product states, projected entangled pair states and variational renormalization group methods for quantum spin systems *Adv. Phys.* **57** 143–224
- [11] Lixin H et al 2018 Peps++: towards extreme-scale simulations of strongly correlated quantum many-particle models on Sunway TaihuLight *IEEE Trans. Parallel Distrib. Syst.* **29** 2838–48
- [12] Liu W-Y, Gong S-S, Yu-Bin Li, Poilblanc D, Chen W-Q and Zheng-Cheng G 2022 Gapless quantum spin liquid and global phase diagram of the spin-1/2  $J_1 - J_2$  square antiferromagnetic Heisenberg model *Sci. Bull.* **67** 1034–41
- [13] Silver D et al 2016 Mastering the game of go with deep neural networks and tree search *Nature* **529** 484–9
- [14] Silver D et al 2017 Mastering the game of go without human knowledge *Nature* **550** 354–9
- [15] Han J, Jentzen A and Weinan E 2018 Solving high-dimensional partial differential equations using deep learning *Proc. Natl Acad. Sci.* **115** 8505–10
- [16] Jia W, Wang H, Chen M, Denghui L, Lin L, Roberto Car E W and Zhang L 2020 Pushing the limit of molecular dynamics with *ab initio* accuracy to 100 million atoms with machine learning *SC20: Int. Conf. for High Performance Computing, Networking, Storage and Analysis* (IEEE) pp 1–14
- [17] Jumper J et al 2021 Highly accurate protein structure prediction with AlphaFold *Nature* **596** 583–9
- [18] Degraeve J et al 2022 Magnetic control of tokamak plasmas through deep reinforcement learning *Nature* **602** 414–9

- [19] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002
- [20] Carrasquilla J 2020 Machine learning for quantum matter *Adv. Phys. X* **5** 1797528
- [21] Carleo G and Troyer M 2017 Solving the quantum many-body problem with artificial neural networks *Science* **355** 602–6
- [22] Liang X, Liu W-Y, Lin P-Z, Guo G-C, Zhang Y-S and Lixin H 2018 Solving frustrated quantum many-particle models with convolutional neural networks *Phys. Rev. B* **98** 104426
- [23] Ferrari F, Becca F and Carrasquilla J 2019 Neural Gutzwiller-projected variational wave functions *Phys. Rev. B* **100** 125131
- [24] Choo K, Neupert T and Carleo G 2019 Two-dimensional frustrated  $J_1 - J_2$  model studied with neural network quantum states *Phys. Rev. B* **100** 125124
- [25] Nomura Y and Imada M 2021 Dirac-type nodal spin liquid revealed by refined quantum many-body solver using neural-network wave function, correlation ratio and level spectroscopy *Phys. Rev. X* **11** 031034
- [26] Mingfan Li, Chen J, Xiao Q, Wang F, Jiang Q, Zhao X, Lin R, Hong A, Liang X and Lixin H 2022 Bridging the gap between deep learning and frustrated quantum spin system for extreme-scale simulations on new generation of Sunway supercomputer *IEEE Trans. Parallel Distrib. Syst.* **33** 2846–59
- [27] Levine Y, Sharir O, Cohen N and Shashua A 2019 Quantum entanglement in deep learning architectures *Phys. Rev. Lett.* **122** 065301
- [28] Stokes J, Robledo Moreno J, Pnevmatikakis E A and Carleo G 2020 Phases of two-dimensional spinless lattice fermions with first-quantized deep neural-network quantum states *Phys. Rev. B* **102** 205122
- [29] Luo Di and Clark B K 2019 Backflow transformations via neural networks for quantum many-body wave functions *Phys. Rev. Lett.* **122** 226401
- [30] Inui K, Kato Y and Motome Y 2021 Determinant-free fermionic wave function using feed-forward neural networks *Phys. Rev. Res.* **3** 043126
- [31] Robledo Moreno J, Carleo G, Georges A and Stokes J 2022 Fermionic wave functions from neural-network constrained hidden states *Proc. Natl Acad. Sci.* **119** e2122059119
- [32] Wen-Jun H, Becca F, Parola A and Sorella S 2013 Direct evidence for a gapless  $Z_2$  spin liquid by frustrating Néel antiferromagnetism *Phys. Rev. B* **88** 060402
- [33] Capello M, Raczkowski M and Poilblanc D 2008 Stability of RVB hole stripes in high-temperature superconductors *Phys. Rev. B* **77** 224502
- [34] Batista C D and Ortiz G 2001 Generalized Jordan–Wigner transformations *Phys. Rev. Lett.* **86** 1082–5
- [35] Sorella S 1998 Green function Monte Carlo with stochastic reconfiguration *Phys. Rev. Lett.* **80** 4558–61
- [36] Liang X, Dong S-J and Lixin H 2021 Hybrid convolutional neural network and projected entangled pair states wave functions for quantum many-particle states *Phys. Rev. B* **103** 035138
- [37] Schollwöck U 2005 The density-matrix renormalization group *Rev. Mod. Phys.* **77** 259–315
- [38] Richter J, Ivanov N B and Retzlaff K 1994 On the violation of Marshall-Peierls sign rule in the frustrated  $J_1$ - $J_2$  Heisenberg antiferromagnet *Europhys. Lett.* **25** 545
- [39] Szabó A and Castelnovo C 2020 Neural network wave functions and the sign problem *Phys. Rev. Res.* **2** 033075
- [40] Nomura Y 2021 Helping restricted Boltzmann machines with quantum-state representation by restoring symmetry *J. Phys.: Condens. Matter* **33** 174003
- [41] Vicentini F et al 2022 NetKet 3: machine learning toolbox for many-body quantum systems *SciPost Phys. Codebases* **7** 7
- [42] Mu B, Soorim S, Can B, Gübtüzbalaban M and Dehnavi M M 2022 HyLo: a hybrid low-rank natural gradient descent method *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis* vol 47 pp 1–16
- [43] Dongarra J 2016 Sunway TaihuLight supercomputer makes its appearance *Natl Sci. Rev.* **3** 265–6
- [44] Shang H et al 2021 Extreme-scale *ab initio* quantum Raman spectra simulations on the leadership HPC system in China *Proc. Int. Conf. for High Performance Computing, Networking, Storage and Analysis* pp 1–13
- [45] Liu Y (A) et al 2021 Closing the “Quantum Supremacy Gap”: Achieving Real-Time Simulation of a Random Quantum Circuit Using a new Sunway Supercomputer. SC ’21 (New York: Association for Computing Machinery)
- [46] Cong F, Zhang X, Huixin Z, Ling H, Shenglong X and Shuiwang J 2022 Lattice convolutional networks for learning ground states of quantum many-body systems (arXiv:2206.07370)
- [47] Dong S-J, Wang C, Han Y, Guo G-C and Lixin H 2019 Gradient optimization of fermionic projected entangled pair states on directed lattices *Phys. Rev. B* **99** 195153
- [48] Dagotto E 1994 Correlated electrons in high-temperature superconductors *Rev. Mod. Phys.* **66** 763–840
- [49] Boninsegni M and Prokof’ev N V 2012 Supersolids: what and where are they? *Rev. Mod. Phys.* **84** 759–76
- [50] Steppke A et al 2013 Ferromagnetic quantum critical point in the heavy-fermion metal  $\text{YbNi}_4(\text{P}_{1-x}\text{As}_x)_2$  *Science* **339** 933–6
- [51] Tsui D C, Stormer H L and Gossard A C 1982 Two-dimensional magnetotransport in the extreme quantum limit *Phys. Rev. Lett.* **48** 1559–62
- [52] Laughlin R B 1983 Anomalous quantum Hall effect: an incompressible quantum fluid with fractionally charged excitations *Phys. Rev. Lett.* **50** 1395–8
- [53] Stormer H L 1999 Nobel lecture: the fractional quantum Hall effect *Rev. Mod. Phys.* **71** 875–89