Supplemental Material: Network-Initialized Monte Carlo Based on Generative Neural Networks

Hongyu Lu,[1] Chuhao Li,[2,3] Bin-Bin Chen,[1] Wei Li,[4,5,*] Yang Qi,[6,7,†] and Zi Yang Meng[1,‡]

[1]*Department of Physics and HKU-UCAS Joint Institute of Theoretical and Computational Physics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China*
[2]*Beijing National Laboratory for Condensed Matter Physics and Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China*
[3]*School of Physical Sciences, University of Chinese Academy of Sciences, Being 100190, China*
[4]*Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100190, China*
[5]*School of Physics, Beihang University, Beijing 100191, China*
[6]*State Key Laboratory of Surface Physics, Fudan University, Shanghai 200438, China*
[7]*Center for Field Theory and Particle Physics, Department of Physics, Fudan University, Shanghai 200433, China*

PHYSICAL MODELS

1. Model Details

First we consider two-dimensional Ising model on square lattice at the critical point $T_c \approx 2.269J$ and the Hamiltonian is shown in Eq.(1) of the main text, which can be solved with classica Monte Carlo simulations. σ_i and σ_j refers to nearest neighbors of classical spins and take values: ± 1. We set the interaction strength J of all pairs of neighbors to be 1.

Next, we consider Hubbard model on the honeycomb lattice, which can be solved with determinant quantum Monte Carlo (DQMC) at half-filling. In Eq.(3) of main text, t is the hopping parameter for the kinetic energy, U is the repulsive Coulomb interaction between electrons on the same lattice site, $\langle i, j\rangle$ represents a pair of nearest-neighbor sites in the lattice, the operators $c^\dagger_{i,\sigma}$ and $c_{i,\sigma}$ are the fermion creation and annihilation operators for fermions with z component of spin-up ($\sigma = \uparrow$) or spin-down ($\sigma = \downarrow$), and the operators $n_{i,\sigma} = c^\dagger_{i,\sigma} c_{i,\sigma}$ are the number operators which count numbers of fermions of spin σ on the site i. In loss function Eq.(4) in the main text, S refers to the structure factor: $S(\boldsymbol{Q}) = \frac{1}{L^2}\sum_{ij} e^{-i\mathbf{Q}\cdot(\mathbf{r}_i - \mathbf{r}_j)}\langle s^z_i s^z_j\rangle$, and the spin operator $\mathbf{s}_i = \frac{1}{2}\sum_{\alpha\alpha'} c^\dagger_{i,\alpha}\overrightarrow{\sigma}_{\alpha\alpha'} c_{i,\alpha'}$. Note here $\overrightarrow{\sigma}$ denotes the Pauli matrices and so the z-component $s^z_i = \frac{1}{2}(n_{i\uparrow} - n_{i\downarrow})$.

2. Determinant Quantum Monte Carlo (DQMC)

The partition function $Z = \mathrm{Tr}\{e^{-\beta H}\}$ is expressed as a path integral by discretizing the inverse temperature β into L_τ slices of length $\Delta\tau$. Then, after Trotter-Suzuki decomposition, the Hamiltonian are separated in each time slice and Z can be written as

$$Z = \mathrm{Tr}\left[\prod_{l=1}^{L_\tau} e^{-\Delta\tau H_k} e^{-\Delta\tau H_U}\right] + O(\Delta\tau^2), \tag{S1}$$

where $H_k = -t\sum_{\langle i,j\rangle,\sigma}(c^\dagger_{i,\sigma} c_{j,\sigma} + h.c.)$ is the kinetic term and $H_U = U\sum_i (n_{i,\uparrow} - \frac{1}{2})(n_{i,\downarrow} - \frac{1}{2})$ is interaction term of the Hubbard model.

To treat the quartic interaction term, the discrete Hubbard-Stratonovich transformation[1-3] can be applied and then $e^{-\Delta\tau H_U} = e^{-U\Delta\tau(n_{i,\uparrow}-\frac{1}{2})(n_{i,\downarrow}-\frac{1}{2})} = \frac{1}{2}e^{-\frac{U\Delta\tau}{4}}\sum_{s_i=\pm1} e^{\nu s_i(n_{i,\uparrow}-n_{i,\downarrow})}$, where the scalar ν is defined by $\cosh\nu = e^{\frac{U\Delta\tau}{2}}$. Putting kinetic part and interaction part together, the term in partition function becomes

$$e^{-\Delta\tau H_k} e^{-\Delta\tau H_U} = \prod_\sigma e^{-\Delta\tau T_\sigma} e^{\sigma\nu s_i n_{i,\sigma}}, \tag{S2}$$

where H_k is rewritten with operator $T_\sigma = -t\sum_{i,j} c^\dagger_{i,\sigma} c_{j,\sigma} + h.c.$. According to the feature of fermion operator[1], a fermion operator ($\hat{M}_l$ for example) with a quadratic form like $\hat{M}_l = \sum_{i,j} c^\dagger_i (M_l)_{ij} c_j$ satisfies

$$\mathrm{Tr}\left[e^{-\hat{M}_1} e^{-\hat{M}_2}\cdots e^{-\hat{M}_L}\right] = \mathrm{Det}[I + e^{-M_1} e^{-M_2}\cdots e^{-M_L}]. \tag{S3}$$

Then the partition function could be finally written in determinant form as

$$Z = \left(\frac{1}{2}e^{-\frac{U\Delta\tau}{4}}\right)^{NL_\tau}\sum_{s_{i,l}}\prod_\sigma \mathrm{Det}\left[I + B^\sigma(L_\tau, L_\tau - 1)\cdots B^\sigma(1,0)\right], \tag{S4}$$

in which

$$B^{\sigma}(l_2, l_1) = \prod_{l=l_1+1}^{l_2} e^{\sigma\nu \mathrm{Diag}\{s_{i,l}\}} e^{-\Delta\tau T}, \tag{S5}$$

where $\sigma = \{1, -1\}$ in calculation corresponding mark $\{\uparrow, \downarrow\}$, and T is the matrix corresponding to the operator T_σ. Thus, we have introduced a sum over the field of auxiliary variables $s_{i,l}$ in a $(d+1)$-dimension space (d for spatial denoted by i and 1 for imaginary time denoted by l) as shown in Fig. S1 (b), and the fermionic degrees of freedom in the quadratic form have been integrated out analytically. Note B^{σ} is an $N \times N$ matrix that depends on the auxiliary configurations.

3. Autocorrelation Function

The autocorrelation function for an observable O is defined as:

$$A_O(t) = \frac{\langle O(i+t)O(i)\rangle - \langle O\rangle^2}{\langle O^2\rangle - \langle O\rangle^2}. \tag{S6}$$

For MCMC, i and t denotes the simulation time, normally in units of the MC sweeps (one sweep means doing flipping attempts over all the spins of the configuration), and the averages are over the reference time i. For Neural Network results, when computing the formal autocorrelation function, i and t simply refers to the serial numbers of the configurations.

NEURAL NETWORK DETAILS

1. Architecture

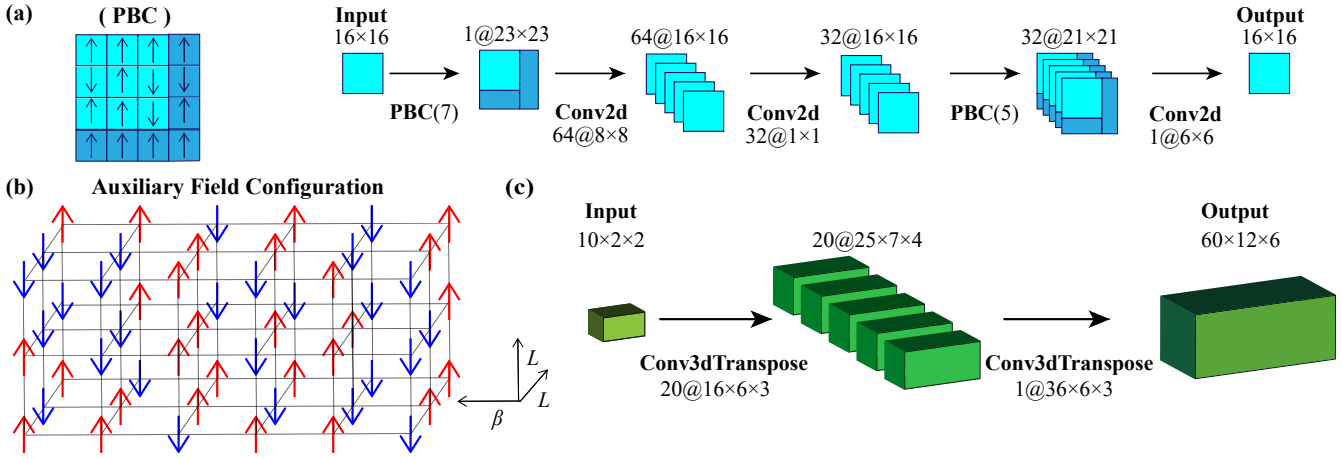

FIG. S1. Schematic figures for the generative neural networks used in this study. (a) shows an example of periodic boundary condition on the left, and we use the neural network structure on the right for the training of the 2d Ising model, where Conv2d stands for 2d convolutional layer. (b) displays the auxiliary field configuration in DQMC for the Hubbard model, with the $(2+1)d$ space-time of $\beta \times L \times L$. The auxiliary fields of ± 1 live on each space-time lattice site. (c) demonstrates the schematic structure of 3d transposed convolutional network we employed for the training of the 2d Hubbard model on $6 \times 6 \times 2$ honeycomb lattice.

For 2d Ising model on square lattice (16×16 for example), the network structure is shown in Fig. S1 (a). For the random input, we fix its shape to be the same as that of the Monte Carlo configurations. Inside the network, inspired by the network structure in Ref.[30], we use three 2d convolutional layers, with 64 filters of size 8×8 for the first layer, 32 filters of size 1 for the second, and 1 filter of size 6×6 for the third layer. We choose rectified linear function ReLU(x) = max(0, x) in the first and second convolutional layers and sigmoid function $\sigma(x) = 1/(1+e^{-x})$ in the third. We also apply periodic boundary condition (PBC) layers each time before the convolutional layers if the kernel dimension is larger than one, which provides the configuration tensors with paddings of boundary elements instead of 0s (the padding size is decided by the size of filters in the corresponding convolutional layer, so as to ensure the shape invariance of the configuration tensors after convolutional operations).

In the case of 2d Hubbard model on the honeycomb lattice ($6\times6\times2$ for example), we switch from classical Ising configurations to the auxiliary field configurations of the DQMC[3,33], with the configuration space of $\beta \times L \times L \times 2$ where $\beta = 1/T$ is the inverse temperature. We set $\beta = 6$ and the Trotter discretization $\Delta\tau = 0.1$, so the DQMC auxiliary field configurations are of shape $60 \times 12 \times 6$ (note the 2 site per unit cell for the honeycomb lattice), as schematically shown in Fig. S1 (b). As shown in Eq.(4), we have to carry out large-scale determinant computation to measure the observables from generated configurations in order to compute the loss, and thus the complexity for the optimization is much larger than the case of classical model. Therefore, instead of the previous PBC-based network structure, we try another architecture of transposed convolutional layers and feed smaller random configurations as input. We find it is still capable of fitting the results we want and optimize faster. As shown in Fig. S1 (c) for the example model, we build two 3d transposed convolutional layers, with all filters using valid padding and sigmoid function as activation. There are 20 filters of size $16 \times 6 \times 3$ in the first layer and 1 filter of size $36 \times 6 \times 3$ in the second and the random input configurations are of shape $10 \times 2 \times 2$ while the predicted output are of shape $60 \times 12 \times 6$.

2. Optimization

In order to optimize such neural networks, for both classical and quantum models, we prepare 1000 sets of observables measured from MC simulations and take 1000 input random configurations which will be processed into generated configurations. The comparison in loss function is randomly distributed without any grouping. For the choice of observables in the defined loss functions, we simply pick some from the ones we usually focus on. In the Ising case, we take the batch size to be 5 and epoch number to be up to 150 as the computation is quite easy. While in the Hubbard case, we run at most 15 epochs with a batch size of 3. We optimize the network parameters using Adam[63] with conventional learning rate 10^{-3}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

* w.li@itp.ac.cn
† qiyang@fudan.edu.cn
‡ zymeng@hku.hk