Transfer learning for scalability of neural-network quantum states

Zen, Remmy; My, Long; Tan, Ryan; Hébert, Frédéric; Gattobigio, Mario; Miniatura, Christian; Poletti, Dario; Bressan, Stéphane

2020

Zen, R., My, L., Tan, R., Hébert, F., Gattobigio, M., Miniatura, C., . . . Bressan, S. (2020). Transfer learning for scalability of neural-network quantum states. *Physical Review E*, 101(5), 053301-. doi:10.1103/physreve.101.053301

<https://hdl.handle.net/10356/146572>

<https://doi.org/10.1103/PhysRevE.101.053301>

© 2020 American Physical Society. All rights reserved. This paper was published in *Physical Review E* and is made available with permission of American Physical Society.

Downloaded on 26 Apr 2023 20:10:29 SGT

Transfer learning for scalability of neural-network quantum states

Remmy Zen¹, Long My¹, Ryan Tan,² Frédéric Hébert,³ Mario Gattobigio,³ Christian Miniatura^{4,5,6,7,8,3},
Dario Poletti,^{9,2,4,*} and Stéphane Bressan¹

¹*School of Computing, National University of Singapore, 117417 Singapore, Singapore*

²*Engineering Product Development Pillar, Singapore University of Technology and Design, 8 Somapah Road, 487372 Singapore, Singapore*

³*Université Côte d'Azur, CNRS, Institut de Physique de Nice, 06560 Valbonne, France*

⁴*MajuLab, CNRS-Université Côte d'Azur-Sorbonne Université-National University of Singapore-Nanyang Technological University, Singapore*

⁵*Centre for Quantum Technologies, National University of Singapore, 117543 Singapore, Singapore*

⁶*Department of Physics, National University of Singapore, 2 Science Drive 3, 117542 Singapore, Singapore*

⁷*School of Physical and Mathematical Sciences, Nanyang Technological University, 637371 Singapore, Singapore*

⁸*Yale-NUS College, 16 College Avenue West, 138527 Singapore, Singapore*

⁹*Science and Mathematics Cluster, Singapore University of Technology and Design, 8 Somapah Road, 487372 Singapore, Singapore*

(Received 2 September 2019; revised manuscript received 4 February 2020; accepted 30 March 2020; published 5 May 2020)

Neural-network quantum states have shown great potential for the study of many-body quantum systems. In statistical machine learning, transfer learning designates protocols reusing features of a machine learning model trained for a problem to solve a possibly related but different problem. We propose to evaluate the potential of transfer learning to improve the scalability of neural-network quantum states. We devise and present physics-inspired transfer learning protocols, reusing the features of neural-network quantum states learned for the computation of the ground state of a small system for systems of larger sizes. We implement different protocols for restricted Boltzmann machines on general-purpose graphics processing units. This implementation alone yields a speedup over existing implementations on multicore and distributed central processing units in comparable settings. We empirically and comparatively evaluate the efficiency (time) and effectiveness (accuracy) of different transfer learning protocols as we scale the system size in different models and different quantum phases. Namely, we consider both the transverse field Ising and Heisenberg XXZ models in one dimension, as well as in two dimensions for the latter, with system sizes up to 128 and 8×8 spins. We empirically demonstrate that some of the transfer learning protocols that we have devised can be far more effective and efficient than starting from neural-network quantum states with randomly initialized parameters.

DOI: [10.1103/PhysRevE.101.053301](https://doi.org/10.1103/PhysRevE.101.053301)

I. INTRODUCTION

Strongly interacting quantum systems are notoriously hard to simulate because the size of their many-body Hilbert vector space grows exponentially fast with the number of particles, restricting exact diagonalization methods to few particles in practice. However, over the years, various advanced numerical methods have been developed to study these systems with an increasing degree of success. Some of the most commonly used and successful ones are quantum Monte Carlo methods [1,2], tensor network algorithms [3–6] (which stemmed from the density matrix renormalization group [7]), and dynamical mean-field theory [8–10]. Recently, a new class of techniques in the family of variational quantum Monte Carlo methods was introduced in the field and quickly came to prominence: neural-network quantum states [11–14]. Starting with [11], it was shown that even simple neural networks, such as restricted Boltzmann machines, can accurately describe the ground state of a many-body quantum system and reconstruct its state and its dynamics [15–17]. This is why neural-network

quantum states have opened a new research direction that is widely explored today even though, for some of the most challenging systems (like frustrated two-dimensional quantum systems), more established methods can provide lower ground-state energies [18,19].

It is now possible to use open-source code libraries, such as NETKET [20], to implement neural-network quantum states. NETKET is a PYTHON framework implemented in C++ with support for the Message Passing Interface for distributed and parallel computing. At the same time, the recent advances in machine learning are due not only to better algorithms and packages, but also to the use of graphical processing units, which can significantly speed up the computations at parity of hardware cost.

A noticeable tool in the toolbox of machine learning techniques is transfer learning [21]. Transfer learning proposes to use a machine learning model trained for a particular task to perform another, possibly related but different task. Here we use transfer learning to scale up the quantum many-body system under study by transferring the optimal neural-network quantum state parameters obtained for an initial system to a similar system of larger size. We study the efficiency and effectiveness of various physics-inspired transfer learning

*dario_poletti@sutd.edu.sg

protocols in different quantum phases. By efficiency we mean the time needed to optimize the neural-network quantum states, while the effectiveness measures the accuracy of the state obtained. A transfer learning protocol combining both good efficiency and effectiveness as the system size grows provides good scalability. We empirically demonstrate that some of the transfer learning protocols that we have devised are far more effective and efficient than starting from neural-network quantum states with randomly initialized parameters. Concurrently, we have ported the NETKET implementation of the neural-network quantum states to the TENSORFLOW machine learning platform [22]. Consequently, our code can be readily used on general-purpose graphics processing units. This port immediately achieves a significant performance speedup over the NETKET code running on a cluster of commodity servers at a comparable cost of hardware.

The remainder of this paper is structured as follows. In Sec. II we summarize the neural-network quantum state technique. We describe the models used in Sec. III. In Sec. IV we describe the transfer learning protocols proposed for these models. In Sec. V we explain how we analyze the performances of our transfer learning protocol. We first introduce the observables analyzed in Sec. V A and the neural-network quantum state implementation in Sec. V B. Section VI is devoted to our results: Sec. VI A details the performance of our general-purpose graphics processing unit over distributed central processing units and Sec. VI B compares the performances of different transfer learning protocols. We briefly summarize our key results and conclude in Sec. VII.

II. NEURAL-NETWORK QUANTUM STATES

We summarize here how the neural-network quantum states are used to estimate the ground state of a quantum many-body system. This is a standard optimization problem based on the Ritz variational method [1]. Given a Hamiltonian H , the expectation value of the energy in any given state $|\psi\rangle$ is always greater than or equal to the ground-state energy E_0 , that is,

$$E[\psi] = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle} \geq E_0. \quad (1)$$

Therefore, in order to estimate the ground-state energy of H , we start with a trial wave function $\psi(\theta)$ that depends on some parameters collectively labeled by θ and minimize $E[\psi(\theta)]$ with respect to θ .

In this work we use a particular type of neural network, called a restricted Boltzmann machine, to represent the wave function. It is a generative energy-based probabilistic graphical model made of two layers. The first one, called the visible layer $\mathbf{x}$, consists of L_v nodes and is in one-to-one correspondence with the configuration space of the system. The second one, called the hidden layer $\mathbf{h}$, consists of an arbitrary number $L_h = \alpha L_v$ of nodes (the rational parameter α sets the ratio between the numbers of nodes in the two layers). Restricted Boltzmann machines can provide an effective representation of quantum states, including some with volume-law entanglement [14,15,23].

In this work, following [11], we concentrate on spin-1/2 models, such as the transverse field Ising or the anisotropic

FIG. 1. Structure of a restricted Boltzmann machine with L_v visible nodes and L_h hidden nodes. The visible layer consists of visible nodes $x_1, \dots, x_{L_v}$. The hidden layer consists of hidden nodes $h_1, \dots, h_{L_h}$. The connections between the visible and hidden layer are given by the weight matrix $\mathbf{W}$. The biases for the visible and hidden layers are represented by $a_1, \dots, a_{L_v}$ and $b_1, \dots, b_{L_h}$, respectively.

Heisenberg models. To describe such systems, binary values $\{+1, -1\}$, corresponding to the projection of the spins along the z axis, are then assigned to the visible node variables x_j , which span the entire Hilbert space of the system. The same binary values are used for the hidden nodes h_i .

As shown in Fig. 1, the structure of a restricted Boltzmann machine forms a bipartite graph in which the hidden and visible nodes are associated with a set of weights $\mathbf{W} = \{W_{ji}\}$ where the first and second index in the matrix label, respectively, the nodes of the visible and hidden layers (e.g., $i \in \{1, \dots, L_h\}$ and $j \in \{1, \dots, L_v\}$). The neural network has also a visible bias vector $\mathbf{a} = \{a_j\}$ and a hidden bias vector $\mathbf{b} = \{b_i\}$, which couple to each node in their respective layer. In our work, all the elements of $\mathbf{W}$, $\mathbf{a}$, and $\mathbf{b}$ are taken as real numbers, which is sufficient to represent the ground states of the systems we are studying. Using the restricted Boltzmann machine, the joint probability distribution of a configuration of the visible layer $\mathbf{x}$ and a configuration of the hidden layer $\mathbf{h}$, for given $\mathbf{W}$, $\mathbf{a}$, and $\mathbf{b}$, is given by

$$p_{\text{RBM}}(\mathbf{x}, \mathbf{h}; \mathbf{W}, \mathbf{a}, \mathbf{b}) = \frac{\exp(\mathbf{a} \cdot \mathbf{x} + \mathbf{b} \cdot \mathbf{h} + \mathbf{x} \cdot \mathbf{W} \mathbf{h})}{Z}, \quad (2)$$

where $\mathbf{W} \mathbf{h}$ is the vector obtained by applying the matrix $\mathbf{W}$ onto $\mathbf{h}$ and Z normalizes the probability so that the sum over all possible combinations of $\mathbf{x}$ and $\mathbf{h}$ is 1. From Eq. (2), by integrating out the hidden layer, we can compute the marginal distribution of the visible layer $p_{\text{RBM}}^m(\mathbf{x}; \mathbf{W}, \mathbf{a}, \mathbf{b})$:

$$p_{\text{RBM}}^m(\mathbf{x}; \mathbf{W}, \mathbf{a}, \mathbf{b}) = \frac{1}{Z} \exp(\mathbf{a} \cdot \mathbf{x}) \times \prod_i 2 \cosh \left(\sum_j x_j W_{ji} + b_i \right). \quad (3)$$

As in [11], we use the marginal distribution provided by the restricted Boltzmann machine to represent the probability of a given configuration $\mathbf{x}$ of the ground state with parameters $\theta = \{\mathbf{W}, \mathbf{a}, \mathbf{b}\}$. Using the bra-ket quantum notation, the (normalized to 1) trial wave function $|\psi(\theta)\rangle = \sum_{\mathbf{x}} \psi(\mathbf{x}; \theta) |\mathbf{x}\rangle$

returned by the restricted Boltzmann machine is thus defined by $\psi(\mathbf{x}; \boldsymbol{\theta}) = \sqrt{p_\psi(\mathbf{x}; \boldsymbol{\theta})} = \langle \mathbf{x} | \psi(\boldsymbol{\theta}) \rangle$, where

$$p_\psi(\mathbf{x}; \boldsymbol{\theta}) = p_{\text{RBM}}^m(\mathbf{x}; \mathbf{W}, \mathbf{a}, \mathbf{b}). \quad (4)$$

Having real and positive coefficients $\psi(\mathbf{x}; \boldsymbol{\theta})$, the trial wave function $|\psi(\boldsymbol{\theta})\rangle$ is well adapted to the models we intend to study because their respective ground states can be taken positive.

From Eqs. (1), (2), and (4), the problem of estimating the ground-state energy becomes the problem of minimizing the function $E[\psi(\boldsymbol{\theta})] = E(\boldsymbol{\theta})$ given by

$$E(\boldsymbol{\theta}) = \frac{\langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle}{\langle \psi(\boldsymbol{\theta}) | \psi(\boldsymbol{\theta}) \rangle} = \sum_{\mathbf{x}} p_\psi(\mathbf{x}; \boldsymbol{\theta}) E_{\text{loc}}(\mathbf{x}; \boldsymbol{\theta}). \quad (5)$$

Using the completeness relation $\sum_{\mathbf{x}} |\mathbf{x}\rangle \langle \mathbf{x}| = \mathbb{1}$, the local energy E_{loc} is given by

$$E_{\text{loc}}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{\mathbf{x}'} \langle \mathbf{x} | H | \mathbf{x}' \rangle \frac{\psi(\mathbf{x}'; \boldsymbol{\theta})}{\psi(\mathbf{x}; \boldsymbol{\theta})}. \quad (6)$$

The minimization of $E(\boldsymbol{\theta})$ uses a stochastic gradient descent algorithm to iteratively update the parameters $\boldsymbol{\theta}$. The value of $E(\boldsymbol{\theta})$ and its gradients are calculated by taking samples from $p_\psi(\mathbf{x}; \boldsymbol{\theta})$.

The sampling procedures rely on Gibbs sampling and the Metropolis-Hastings algorithm. For Gibbs sampling, starting from an initial visible configuration $\mathbf{x}$, a hidden configuration is generated by sampling from the conditional probability $p(\mathbf{h}|\mathbf{x})$ given by

$$p(\mathbf{h}|\mathbf{x}) = \prod_j \text{sigmoid} \left[2 \left(\sum_i W_{ji} h_i + a_j \right) x_i \right]. \quad (7)$$

Then, from this hidden configuration a new visible configuration is generated by sampling from the conditional probability $p(\mathbf{x}|\mathbf{h})$ given by

$$p(\mathbf{x}|\mathbf{h}) = \prod_i \text{sigmoid} \left[2 \left(\sum_j W_{ji} x_j + b_i \right) h_i \right]. \quad (8)$$

For the Metropolis-Hastings algorithm, starting from an initial visible configuration $\mathbf{x}$, we get a new visible configuration $\mathbf{x}'$ through an arbitrary strategy and decide to accept or reject it depending on the relative probability $p_{\text{RBM}}^m(\mathbf{x}'; \boldsymbol{\theta}) / p_{\text{RBM}}^m(\mathbf{x}; \boldsymbol{\theta})$. Gibbs sampling is typically much faster than the Metropolis-Hastings sampling algorithm, as autocorrelation times in the sampling are longer in the latter case. However, the Metropolis-Hastings algorithm is much more suitable to impose constraints on the possible samples. For example, for cases where the states are restricted to a fixed total magnetization along the z axis (which means that the total number of up and down spins is fixed), a sampling algorithm obeying such a criterion can be readily implemented in the Metropolis-Hastings algorithm by using spin exchanges to propose new configurations.

III. MODELS ANALYZED

We conduct numerical experiments on two different quantum models, namely, the transverse field Ising and Heisenberg

XXZ models. Hereafter, we refer to the former as the Ising model and to the latter as the Heisenberg model. They are described, respectively, by the Hamiltonians

$$H_I = -J_I \sum_{\langle l,m \rangle} \sigma_l^z \sigma_m^z - h \sum_l \sigma_l^x \quad (9)$$

and

$$H_{\text{XXZ}} = -J_{\text{XXZ}} \sum_{\langle l,m \rangle} (\sigma_l^x \sigma_m^x + \sigma_l^y \sigma_m^y + \Delta \sigma_l^z \sigma_m^z), \quad (10)$$

where the $\sigma_l^{x,y,z}$ are operators acting on site l and corresponding to the respective Pauli matrices. For the Ising model, we use the parameters J_I for the spin coupling strength and h for the transverse field, while for the Heisenberg model we use J_{XXZ} for the in-plane spin coupling strength and Δ for the dimensionless anisotropy factor between the xy plane and z axis coupling strengths. For both models, we consider open boundary conditions, for which simplifications of the restricted Boltzmann machine leveraging on translation invariance [11,12] cannot be used.

For the Ising model, the spin coupling J_I favors a ferromagnetic (F) ground state when positive and an antiferromagnetic (A) one when negative. At the same time, a strong magnetic field h favors spins that are aligned paramagnetically, pointing along the x axis in the positive ($h > 0$) or negative ($h < 0$) direction. In the one-dimensional model, a quantum phase transition occurs at $|J_I/h| = 1$, with the ground state being ferromagnetic along the z axis for $J_I > |h|$, antiferromagnetic along z for $J_I < -|h|$, and paramagnetic (PM) otherwise.

For the Heisenberg model, there is an exact mapping between Hamiltonians with parameters (J_{XXZ}, Δ) and $(-J_{\text{XXZ}}, -\Delta)$ and we will then keep $J_{\text{XXZ}} = 1$. In one dimension, for $J_{\text{XXZ}} > 0$, the ground state can be in three different phases: ferromagnetic along z for $\Delta > 1$, showing no magnetic order along z for $|\Delta| < 1$ (where the model orders ferromagnetically in the xy plane), and antiferromagnetic along z for $\Delta < -1$. When $\Delta = 1$, the model is isotropic.

An important difference between the Ising and Heisenberg models is that the latter preserves the total spin along z , which commutes with H_{XXZ} and is then a conserved quantity, while for the former there is no such conservation. In the following, we restrict our study of the Heisenberg model to the sector where the total magnetization along z is zero. Then, if the system becomes ferromagnetic along the z axis, its lowest-energy configuration will correspond to a symmetric superposition of two states, each composed of one spin-up domain and one spin-down domain of equal sizes, separated by a single domain wall exactly located in the middle of the system.

IV. TRANSFER LEARNING PROTOCOLS

In machine learning, transfer learning refers to the reuse of the features of a machine learning model learned for the resolution of a problem to initialize a machine learning model used to solve another problem [21]. The authors of [24] discuss the transferability of features between deep neural networks with the same architecture, i.e., the same nodes and connections.

In our case, a many-body system is simulated by a restricted Boltzmann machine. We consider the transfer of parameters to networks having, however, different architectures. With transfer learning, we aim to solve a larger system problem using the solution of a smaller one, although there are applications for the reverse. The transfer learning approach is as follows. First, we train a network using randomly initialized parameters to minimize the energy of a system. Following the terminology in [24], we refer to this network as the base network. Second, after the base network has been trained, we copy the parameters of the network to another larger system, which is called the target network. Third, we train the target network using the transferred parameters as the initialized parameters of the network. We call this process the fine-tuning of the network. Finally, we evaluate the transfer learning protocol by comparing the performance of the network trained with the transfer learning protocol and the network trained from scratch with randomly initialized parameters. The former is commonly called the hot-start network while the latter is called the cold-start network.

In this paper we devise different transfer learning protocols and comparatively evaluate their efficiency and effectiveness. To evaluate the efficiency, we compare the times needed to reach a minimum energy. To evaluate the effectiveness, we probe the accuracy of the wave function. Starting from a one-dimensional system of L_v spins, we increase the number of spins to $L'_v = 2L_v, 3L_v, 4L_v, \dots$ at each iteration. We denote the weights of the base network by $\tilde{W}_{ji}$ and the weights of the target network by W_{ji} . The transfer learning rule, or protocol, is then specified by the mapping $\tilde{W}_{ji} \rightarrow W_{ji}$. Mathematically, there is a large number of possible ways to transfer the weights from a smaller system to a larger one and it would not be possible, or meaningful, to consider all of them. Instead, we consider a transfer learning protocol which we refer to as (k, p) tiling, where groups of k weights calculated for a system at a certain size are repeated p times to initialize the weights for a larger system. Even if our transfer learning protocols only focus on weight transfer, our simulations show that, after the fine-tuning process, the bias values are generally negligible. We have further checked that simulations with biases set to zero and with variable biases yield very close results.

The different tilings that we used are detailed below and depicted in Fig. 2. Do note that each tiling is not just performed over the same hidden nodes but is distributed in equal parts over portions of the hidden nodes. By doing so, all the hidden nodes of the target network retain some knowledge acquired by the base network. This was found to help the optimization process of the target network.

(i) *(1,2) tiling* [see Fig. 2(a)]. We focus on doubling the system ($p = 2$). The weights of the target network are initialized with $W_{2j-1,i} = W_{2j,i} = \tilde{W}_{j,i}$ for $j \in [1, L_v/2]$, $W_{2j-1,i+L_h} = W_{2j,i+L_h} = \tilde{W}_{j,i}$ for $j \in [L_v/2 + 1, L_v]$, and a random value for all the other terms. This protocol is expected to work well for paramagnetic or ferromagnetic phases, but it may not be the ideal transfer learning for a system in the antiferromagnetic phase. This tiling indeed favors situations in which each site is equivalent, e.g., considering the state $|\uparrow\uparrow\uparrow\uparrow\rangle$, the doubling will give a bias towards the state $|\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\rangle$ (spins in red are “copied” from the base network). This could be seen as

FIG. 2. Schematic representation of the different transfer learning protocols used to scale up one-dimensional systems. Here L_v and L_h , and L'_v and L'_h correspond to the number of visible and hidden nodes for the base and target networks, respectively. The different panels show how to construct the target weight matrix $\mathbf{W}$ from the base weight matrix $\tilde{\mathbf{W}}$ by replicating the colored rows and filling the gray cells with random entries: (a) (1,2) tiling, (b) (2,2) tiling, (c) $(L, 2)$ tiling (the scaling factor is 2), and (d) $(L, 4)$ tiling (scaling up the network by a factor 4; see the text for a detailed explanation).

favoring ferromagnetic correlations. We see a similar effect in a ferromagnetic state where the total magnetization along z is fixed to zero ($|\uparrow\uparrow\downarrow\downarrow\rangle$ to $|\uparrow\uparrow\uparrow\uparrow\downarrow\downarrow\downarrow\rangle$). If, however, the base network represents an antiferromagnetic state $|\uparrow\downarrow\uparrow\downarrow\rangle$, then the initialization of the target network would bias towards the state $|\uparrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\rangle$, which has much weaker antiferromagnetic correlations.

(ii) *(2,2) tiling* [see Fig. 2(b)]. This protocol is similar to the (1,2) tiling just described, but instead of doubling a single spin, we double a pair of them since $k = 2$. We use $W_{4j-3,i} = W_{4j-1,i} = \tilde{W}_{2j-1,i}$ and $W_{4j-2,i} = W_{4j,i} = \tilde{W}_{2j,i}$ for $j \in [1, L_v/4]$, $W_{4j-3,i+L_h} = W_{4j-1,i+L_h} = \tilde{W}_{2j-1,i}$ and $W_{4j-2,i+L_h} = W_{4j,i+L_h} = \tilde{W}_{2j,i}$ for $j \in [L_v/4 + 1, L_v/2]$, and a random value for all the other terms. This protocol is expected to work better for antiferromagnetic phases compared to the (1,2)-tiling protocol, as unit cells of two sites are copied, while still being effective for ferromagnetic phases. A state $|\uparrow\downarrow\uparrow\downarrow\rangle$ would give a bias towards the state $|\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\rangle$, which preserves antiferromagnetic correlations. Similarly in the ferromagnetic phase, a state $|\uparrow\uparrow\uparrow\uparrow\rangle$ would bias towards the state $|\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\rangle$ and, in the case of a zero magnetization ferromagnetic state, $|\uparrow\uparrow\downarrow\downarrow\rangle$ would bias towards the state $|\uparrow\uparrow\uparrow\uparrow\downarrow\downarrow\downarrow\downarrow\rangle$, which preserves ferromagnetic correlation.

(iii) (L, p) tiling [see Fig. 2(c) for the $(L, 2)$ tiling and Fig. 2(d) for the $(L, 4)$ tiling]. By setting $k = L_v$, we transfer all the base network weights $\tilde{W}_{j,i}$ on the first L_v visible nodes and then repeat them for the other $(p-1)L_v$ half on the visible layer nodes but coupled to the other hidden nodes. We stress here that the symbol L denotes all the visible nodes $L = L_v$. More precisely, we match $W_{j+\eta L_v, i+\eta L_h} = \tilde{W}_{j,i}$ for $j \in [1, L_v]$, $i \in [1, L_h]$, and $\eta \in [0, p-1]$. This protocol could be favorable in the antiferromagnetic phase because a state $|\uparrow\downarrow\downarrow\rangle$ would give a bias towards the state $|\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\rangle$ and in the ferromagnetic phase because a state $|\uparrow\uparrow\uparrow\rangle$ would give a bias towards the state $|\uparrow\uparrow\uparrow\uparrow\uparrow\uparrow\rangle$. However, for a ferromagnetic state where the sum of spins is fixed to zero, the state $|\uparrow\uparrow\downarrow\downarrow\rangle$ would give some bias towards $|\uparrow\uparrow\downarrow\downarrow\uparrow\uparrow\downarrow\downarrow\rangle$, which has weaker ferromagnetic correlations.

Examples of the $(1,2)$ -, $(2,2)$ -, and $(L, 2)$ -tiling protocols are depicted in Figs. 2(a)–2(c), in the case where the transfer is from $L_v = 4$ to $L'_v = 8$ with $\alpha = 2$. Figure 2(d) shows the $(L, 4)$ tiling from $L_v = 4$ to $L'_v = 16$.

A natural generalization of (k, p) tilings to lattices in higher dimensions $d > 1$ are $(\mathbf{k}, \mathbf{p})$ tilings where $\mathbf{k}$ and $\mathbf{p}$ are d -dimensional vectors. In this case, each couple (k_a, p_a) ($1 \leq a \leq d$) tells how many groups of weights computed for a certain system size to consider and how many times to repeat them to initialize the weights for a larger system size along the a th axis. Hereafter, for simplicity, we consider the case of the isotropic transfer learning protocol where $k_a = k$ and $p_a = p$ for all a . Hence we will once again just refer to this transfer protocol as the (k, p) tiling. The above-mentioned protocols may not be exhaustive, but, as we will see later, they already give interesting insight into the efficiency and effectiveness of the transfer learning method.

V. PERFORMANCE EVALUATION

In this section we describe the quantities we use to evaluate the performance in Sec. V A and the implementation details of the neural-network quantum states and its minimization procedure in Sec. V B. To explore the different phases of the one-dimensional Ising and Heisenberg models, we fix $h = 1$ and $J_{XXZ} = 1$ (the latter ensures that the ground state can be taken as positive) and we vary J_I and Δ , concentrating on three values, each corresponding to a different phase.

For the one-dimensional Ising model, we use $J_I = 2$ in the ferromagnetic phase (Ising F), $J_I = -2$ in the antiferromagnetic one (Ising A), and $J_I = 0.5$ in the paramagnetic phase (Ising PM). For the one-dimensional Heisenberg model, we study the cases $\Delta = -2$ (ferromagnetic along the z axis, Heis F), $\Delta = 2$ (antiferromagnetic along the z axis, Heis A), and $\Delta = -0.5$ (ferromagnetic in the xy plane but showing no order along the z axis, Heis XY). For the two-dimensional Heisenberg model, we focus on the case $\Delta = -1$ and $J_{XXZ} = 1$, which is, through the previously mentioned mapping, equivalent to the $\Delta = 1$ and $J_{XXZ} = -1$ isotropic antiferromagnetic Heisenberg model.

In the following, we consider chains containing $L_v = \{4, 8, 16, 32, 64, 128\}$ spins for the one-dimensional Ising and Heisenberg models and $L_v = \{2 \times 2, 4 \times 4, 8 \times 8\}$ square lattices for the two-dimensional Heisenberg model. As the minimization procedures that we use to compute the ground

state have random components, each estimate of an observable is an average over 20 realizations of the same calculation.

A. Evaluation methods

We evaluate both the efficiency and the effectiveness of the proposed transfer learning protocols and compare the results to other readily available methods. For efficiency, we refer to the time needed to reach the stopping criterion of the minimization (see below for a description of this criterion). We compare the time needed for the cold start with the time required by the hot start plus the time of the previous iterations. For instance, for a one-dimensional system with $L_v = 32$, we compare the time required by a cold start for this size to the time needed by a cold start for $L_v = 4$ plus the time for hot starts at $L_v = 8$, $L_v = 16$, and finally $L_v = 32$.

For effectiveness, we refer to the quality of the representation of the ground state by our ansatz. As energy alone might not be a sufficient indicator of the quality of the ground state, we also consider the spin-spin correlator between two spins l and m , $C_{l,m}^z = \langle \sigma_l^z \sigma_m^z \rangle$. This is computed by

$$C_{l,m}^z = \sum_{\mathbf{x}} p(\mathbf{x}) \langle \mathbf{x} | \sigma_l^z \sigma_m^z | \mathbf{x} \rangle, \quad (11)$$

which is evaluated with Monte Carlo sampling over the possible spin configurations $\mathbf{x}$. From Eq. (11) we define the correlators C_d^F and C_d^A , designed to better identify the ferromagnetic (F) and antiferromagnetic (A) orders

$$C_d^F = \frac{1}{d-1} \sum_{l=2}^d C_{1,l}^z, \quad (12)$$

and

$$C_d^A = \frac{1}{d-1} \sum_{l=2}^d (-1)^{l-1} C_{1,l}^z. \quad (13)$$

Here d is a range of distances between the two spins that we consider, for instance, half the length of the spin chain. As these correlators are specific for antiferromagnetic and ferromagnetic orders along the z axis, we do not evaluate these correlators in phases that do not show a magnetic order along z .

The behavior of C_d^F is different in the ferromagnetic phases of the Ising and Heisenberg models. For the Ising model, we can expect C_d^F to be almost constant for all d . On the contrary, for the Heisenberg model, as we work at fixed zero magnetization, C_d^F should be almost constant for $d < L_v/2$ and should then decrease towards zero for $d > L_v/2$ (or vice versa) as the system is separated into two ferromagnetic domains with opposite spins.

For one-dimensional Ising and Heisenberg models, the results from neural-network quantum state calculations for ground-state energy and correlations are compared to accurate matrix product state simulations [6]. For the matrix product state simulations, we use both a non-number conserving code for the Ising model and a number conserving code for the Heisenberg model, with a bond dimension D up to 1000 for most of the computations. For the two-dimensional Heisenberg model, with $J_{XXZ} = 1$ and $\Delta = -1$, the ground-state energy is compared to quantum Monte Carlo results [2] with

$\beta = 50.0$, 20 bins, and 100 000 steps per bin. We use the mean relative error as the effectiveness metric of the ground-state energy and the correlations. Furthermore, we plot the ferromagnetic correlator C_d^F and the antiferromagnetic correlator C_d^A for a different range of distances d to evaluate the correlations qualitatively.

To quantify by how much the parameters of the restricted Boltzmann machine change during an optimization from a hot start, we measure the transfer distance $\mathcal{D}$ between the initial weights $W^{(\text{init})}$ of the target network, obtained after transfer from the base network, and the final weights $W^{(\text{final})}$ of the target network obtained after fine-tuning:

$$\mathcal{D}(W^{(\text{init})}, W^{(\text{final})}) = \sum_j^{L_v} \sum_i^{L_h} \frac{|W_{i,j}^{(\text{init})} - W_{i,j}^{(\text{final})}|}{L_v L_h}. \quad (14)$$

A small $\mathcal{D}$ value means that the transfer gives already very good parameters for the network and little change is needed, while a large $\mathcal{D}$ value implies a less adequate transfer.

B. Details of the neural-network quantum state implementation

The actual implementation of the neural-network quantum states and its minimization relies on several parameters and steps, which we present here in greater detail. The values of the parameters presented here are determined from the literature or by using a grid search.

The optimization process is done by iteration. At each iteration step, we take 10^4 samples to evaluate the energy $E(\theta)$ and its gradients using the current parameters of the restricted Boltzmann machine; then we update the parameters of the restricted Boltzmann machine using a gradient descent algorithm. In our case, we use adaptive learning rate strategies RMSProp [25] and Adam [26]. We set the initial learning rate to 0.001. For the Ising model, the samples are obtained by a single-iteration Gibbs sampling. For the Heisenberg model, we instead use a Metropolis-Hastings algorithm with exchange strategy, for which two randomly chosen spins are exchanged to obtain a new configuration while keeping the total number of spins up and of spins down invariant. This strategy is used in the Heisenberg model to conserve the total spin magnetization. We use 10% of the number of Monte Carlo samples to implement these thermalization steps.

Based on our exploratory simulations, we find that a restricted Boltzmann machine with twice as many hidden nodes as visible ones ($\alpha = 2$) is a good compromise considering the efficiency and effectiveness trade-off. During transfer learning, or for cold starts, we randomly set some parameters of the restricted Boltzmann machine. We sample the weight matrix W entries from the normal distribution $\mathcal{N}(0, 0.01)$ with zero mean and standard deviation 0.01. Following [27] and our exploratory experiments, both bias vectors $\mathbf{a}$ and $\mathbf{b}$ are initialized to zero.

To halt the simulations, we use a dynamic stopping criterion based on the zero-variance principle in quantum Monte Carlo [1]. We compute the standard deviation of $E_{\text{loc}}(\mathbf{x})$,

$$\sigma_{E_{\text{loc}}} = \sqrt{\sum_{\mathbf{x}} [E_{\text{loc}}(\mathbf{x}, \theta) - E(\theta)]^2 / N}, \quad (15)$$

where N is the number of configurations $\mathbf{x}$ considered, while $E(\theta)$ and $E_{\text{loc}}(\mathbf{x}, \theta)$ are given, respectively, in Eqs. (5) and (6). We stop the simulation when the ratio $\sigma_{E_{\text{loc}}}/E(\theta)$ is less than some chosen threshold value ε_σ . If $\sigma_{E_{\text{loc}}}/E(\theta)$ does not fall below ε_σ , we stop the simulation after $\varepsilon_{\text{iter}}$ iterations. In our evaluations, we take $\varepsilon_\sigma = 0.005$ and $\varepsilon_{\text{iter}} = 30\,000$.

This stopping criterion is motivated by the fact that if the neural-network quantum state represents any eigenstate of the Hamiltonian, then $E_{\text{loc}}(\mathbf{x}, \theta)$ returns the energy of the eigenstate for any configuration $\mathbf{x}$. Combining this with the fact that the algorithm searches for the lowest energy, if $\sigma_{E_{\text{loc}}}$ is very small then the neural-network quantum states should be a good representation of the ground state.

To compute the observables, we sample over a certain number of configurations $\mathbf{x}$ taken from $p_\psi(\mathbf{x}, \theta)$ with 500 steps and 5000 steps for Gibbs sampling and the Metropolis-Hastings algorithm, respectively. These values were determined by looking at the autocorrelation times in our exploratory experiments.

VI. RESULTS

A. Efficiency of graphics processing unit

We implement the neural-network quantum state code with the machine learning platform TENSORFLOW [22]. TENSORFLOW allows us to deploy and run the code on a general-purpose graphics processing unit server, which we then also compare to a parallel central processing unit implementation, for which we use the NETKET library.¹ For the NETKET library, all experiments in this section have been run on machines with Ubuntu 16.04 operating systems with two 1.8-GHz Intel® Xeon® Silver 4108 processors and equipped with 128 GB DDR4 memory. For the TENSORFLOW implementation, at a comparable hardware purchase price,² all experiments in this section have been run on NVIDIA GeForce GTX 1080 graphics processing unit with 2560 CUDA cores and 8 GB memory. We implemented our numerical codes [28] using PYTHON 2.7.15.

The efficiency of our TENSORFLOW implementation on a general-purpose graphics processing unit server is higher than that of the original NETKET implementation on a multicore server as soon as the system size is sufficiently large. This is illustrated, for reference only, in Fig. 3, where the computation time t (in seconds) needed by different numerical strategies to reach the stopping criterion is plotted as a function of the number of spins L_v for the one-dimensional Ising model at the quantum phase transition point ($J_I = 1.0$). For a system

¹We have compared both codes with the same set of parameters, using restricted Boltzmann machines and sampling with the Metropolis-Hastings algorithm that flips a random spin. One should note that the NETKET implementation uses a different definition of the wave function than ours, namely, $\psi(\mathbf{x}, \theta) = p_{\text{RBM}}^m(\mathbf{x}; \theta)$. For a fair comparison testing of the performances of both codes, we have used the NETKET definition instead of ours in Sec. II. The rest of the results were obtained with the method presented in Sec. II, which allows the use of the Gibbs sampling.

²As a reference, at the time of writing, the cost of each processor and the cost of the graphics processing units are around 500 USD.

FIG. 3. Computation times t (in seconds) obtained for different numerical strategies for the one-dimensional ferromagnetic Ising model ($J_I = 1$ and $h = 1$) as a function of the number of spins $L_v = 8, 16, 32, 64$, and 128 . The purple dash-dotted line with diamonds shows t using four central processing units with NETKET. The red dashed line with triangles shows t using eight central processing units with NETKET. The blue solid line with squares shows t using 16 central processing units with NETKET. The black dotted line with circles shows t using a graphics processing unit with TENSORFLOW.

of 128 spins, the TENSORFLOW implementation runs about 5 times faster than the NETKET implementation.

B. Evaluation of transfer learning protocols

All experiments in this section have been run on an NVIDIA DGX-1 server equipped with NVIDIA Tesla V100 graphics processing units with 640 tensor cores, 5120 CUDA cores, and 16 GB memory.

1. Efficiency

We first compare the efficiency of different transfer learning protocols and cold starts for the Ising and Heisenberg models using three different Hamiltonian parameters. To do so, we plot the computation time required to reach the stopping criterion for each of these systems as a function of its size (see Fig. 4). Here Figs. 4(a), 4(c), and 4(e) show results for the Ising model with $J_I = -2, 0.5$, and 2 , respectively, while Figs. 4(b), 4(d), and 4(f) show results for the Heisenberg model with $\Delta = -2, -0.5$, and 2 , respectively. The error bars represent the interval of ± 1 standard deviation obtained from 20 realizations. The error bars are used to evaluate the robustness of the transfer learning protocol, e.g., smaller error bars correspond to more consistent results between different realizations.

For the cold start, we start from a randomly initialized network at each system size. For the transfer learning protocols, the initial parameters at size $2L_v$ are obtained from the parameters obtained at size L_v once the stopping criterion is reached. Thus, the full computation time at size $2L_v$ is given by the sum of the computation time at size L_v and the time to reach the stopping criterion at size $2L_v$ from the transferred parameters. The sequence starts with a system at size $L_v = 4$ being randomly initialized with a cold start. Note that, due

FIG. 4. Efficiency of different transfer learning protocols: (a) Ising antiferromagnetic, (b) Heis antiferromagnetic, (c) Ising paramagnetic, (d) Heis XY, (e) Ising ferromagnetic, and (f) Heis ferromagnetic. Each panel shows the computation time t (in seconds) obtained for different transfer learning protocols until the stopping criterion is reached as a function of the number of spins L_v . The blue solid line with squares shows the cold start, the red dashed line with circles ($L, 2$) tiling, the black dotted line with diamonds ($2,2$) tiling, and the green dash-dotted line with triangles ($1,2$) tiling. The error bars represent the ± 1 standard deviation of the interval. The left column shows the Ising model with $h = 1$ and (a) $J_I = -2$, (c) $J_I = 0.5$, and (e) $J_I = 2$, while the right column shows the Heisenberg model with $J_{XXZ} = 1$ and (b) $\Delta = -2$, (d) $\Delta = -0.5$, and (f) $\Delta = 2$.

to the large horizontal scale, the increase of t with L_v for the $(1,2)$ - and $(2,2)$ -tiling protocols is not visible in Fig. 4(f).

For the Ising model in the antiferromagnetic phase, the $(1,2)$ tiling has the largest computation time, even larger than the cold start [see Fig. 4(a)]. We note that this protocol is also not robust since we get large error bars and thus a large variance. Such a low performance is expected since the network is initialized in a state that does not favor an antiferromagnetic order as discussed in Sec. IV. For the Ising model, we generally observe a much better performance from the $(L, 2)$ tiling, which also seems to be less sensitive to the phase of the system [see Figs. 4(a), 4(c), and 4(e)]. We also observe that transfer learning protocols are more robust in the paramagnetic phase since the variance is very small. In the Appendix we show and analyze the convergence plots of different transfer learning protocols for one-dimensional ferromagnetic and antiferromagnetic Ising models.

For the Heisenberg model, we observe that the cold start is always more efficient than the other transfer learning protocols, i.e., it is the fastest to reach the stopping criterion. However, efficiency does not necessarily imply effectiveness. In fact, in this case, the system is trapped in a local minimum prematurely. However, as we will see later, this is not the case for the transfer learning protocols. For example, the $(L, 2)$ -tiling protocol works efficiently except in the ferromagnetic phase ($\Delta = 2$). As discussed in Sec. IV, this is expected since the network is initialized in a state that does not favor a ferromagnetic order when magnetization is conserved. Instead,

FIG. 5. Weight matrices for a single configuration of the one-dimensional ferromagnetic Ising model ($J_I = 2$ and $h = 1$) with 16 spins and $\alpha = 2$. The rows and columns of the matrix correspond to the visible ($L_v = 16$) and the hidden ($L_h = \alpha L_v = 32$) nodes of the restricted Boltzmann machine, respectively. For each row, the left panel shows the initial weight matrix $W^{(\text{init})}$ obtained for a given transfer learning protocol from an eight-spin chain and the right panel shows the final fine-tuned weight matrix $W^{(\text{final})}$ obtained after the stopping criterion is met: (a) and (b) (1,2)-tiling protocol for transfer distance $\mathcal{D} = 0.01145$, (c) and (d) (2,2)-tiling protocol for $\mathcal{D} = 0.01704$, and (e) and (f) ($L, 2$)-tiling protocol for $\mathcal{D} = 0.00162$.

the (1,2)- and (2,2)-tiling protocols are more efficient in the ferromagnetic phase. Contrary to the Ising model, we observe that the transfer learning protocols are more robust in the ferromagnetic phase except for the ($L, 2$)-tiling protocol.

Other optimization techniques have been proposed (for instance, in [29,30]) that seek to avoid the system being trapped in local minima. We expect that these optimization techniques can leverage transfer learning to further improve efficiency.

2. Transfer distance

We now analyze how the weight matrix W of the restricted Boltzmann machine changes when different transfer learning protocols, starting from a single configuration of the Ising model, reach the stopping criterion. We consider weight transfers from $L_v = 8$ to $L_v = 16$ spins with $\alpha = 2$. Figures 5 and 6 show the matrix entries $W_{j,i}$ ($j \in [1, 16]$ and $i \in [1, 32]$) obtained in the ferromagnetic ($J_I = 2$ and $h = 1$) and antiferromagnetic ($J_I = -2$ and $h = 1$) phases, respectively. From Figs. 5 and 6 we observe that the ($L, 2$)-tiling protocol is generically better for the Ising model as it requires smaller weight changes to reach the stopping criterion. This is particularly striking when compared to the (1,2)-tiling protocol in the antiferromagnetic phase [Figs. 6(a) and 6(b)], which requires significant changes of all the weights, resulting in a transfer

FIG. 6. Same as Fig. 5 for the one-dimensional antiferromagnetic Ising model ($J_I = -2$ and $h = 1$). The transfer distances $\mathcal{D}$ for the (1,2)-tiling, (2,2)-tiling, and ($L, 2$)-tiling protocols are 0.05423, 0.01858, and 0.00224, respectively.

distance $\mathcal{D} \approx 0.054$. Comparatively, only some hidden-visible node connections are enhanced for the ($L, 2$)-tiling protocol [Figs. 6(e) and 6(f)], resulting in a transfer distance $\mathcal{D} \approx 0.002$ smaller by a factor 25. Qualitatively, we further observe that the weight matrices in Fig. 6 form a checkerboard pattern which resembles the alternating spins in the antiferromagnetic phase. With the (1,2)-tiling protocol, the checkerboard pattern is destroyed, as shown in Fig. 6(a), and the network tries to restore this pattern, as shown in Fig. 6(b), which makes the transfer distance $\mathcal{D}$ bigger than with other protocols.

Quantitatively, the average transfer distances $\mathcal{D}$ computed over 20 realizations for the (1,2)-tiling, (2,2)-tiling, and ($L, 2$)-tiling transfer protocols from 64 to 128 spins in the antiferromagnetic phase ($J_I = -2.0$) are 6.5×10^{-3} , 1.0×10^{-3} , and 5.0×10^{-5} , respectively. In the ferromagnetic phase ($J_I = 2.0$), they are 6.5×10^{-4} , 9.6×10^{-4} , and 5.0×10^{-5} , respectively.³ These results show that the ($L, 2$)-tiling protocol is indeed the most adequate transfer learning protocol for the Ising model.

For the Heisenberg model ($J_{XXZ} = 1$), the obtained weight matrix patterns are similar to those of the Ising model and we do not plot them here. In the antiferromagnetic phase ($\Delta = -2$), the average transfer distances $\mathcal{D}$ (computed over 20 realizations) for the (1,2)-tiling, (2,2)-tiling, and ($L, 2$)-tiling protocols for 128 spins are 4.9×10^{-3} , 4.5×10^{-3} , and 2.7×10^{-4} , respectively. In the ferromagnetic phase ($\Delta = 2$), they are 1.6×10^{-4} , 2.0×10^{-4} , and 4.3×10^{-4} , respectively. These values confirm our previous results on the efficiency evaluation of the Heisenberg model and show that the (1,2)-

³Here, and for the Heisenberg model discussed in the next paragraph, we have used the weights of a system with 64 spins from the respective transfer learning protocol, as in Fig. 4.

FIG. 7. Relative ground-state energy errors $\Delta E/E_0$ obtained for different transfer learning protocols (including the cold start) and relative to accurate matrix product state simulations. Computations are run on a one-dimensional system with 128 spins initialized with a weight matrix generated from the fine-tuned weight matrix obtained with the most effective transfer learning protocol for the same system with 64 spins until the stopping criterion is reached; $\Delta E/E_0$ is computed when the stopping criterion is met. (a) Ising model with $J_I = -2, 0.5$, and 2 ($h = 1$) and (b) Heisenberg model with $\Delta = -2, -0.5$, and 2 ($J_{XXZ} = 1$). The blue open bar shows the cold start, the red closed bar ($L, 2$) tiling, the green closed bar $(1, 2)$ tiling, and the black open bar $(2, 2)$ tiling. The error bars represent the interval of the 9th percentile and the 91st percentile.

tiling protocol is the most adequate transfer learning protocol for the ferromagnetic phase, while the $(L, 2)$ -tiling protocol works best for the antiferromagnetic phase.

3. Effectiveness

While efficiency is an important attribute of a numerical method, effectiveness cannot be compromised, i.e., the representation of the ground state needs to be accurate. To assess the effectiveness of each transfer learning protocol, we have compared the computed ground-state energy E_0 and the correlations generated by the associated computed ground-state wave function to the results of a very accurate matrix product state algorithm for one-dimensional models.

In Fig. 7 we plot the relative error $\Delta E/E_0$ obtained for a system of 128 spins with the cold-start, $(1, 2)$ -tiling, $(2, 2)$ -tiling, and $(L, 2)$ -tiling protocols until the stopping criterion is reached. Figures 7(a) and 7(b) give our results for the Ising and Heisenberg models, respectively. For a fair comparison between the different protocols, we have considered (i) the same base network and (ii) an initial weight matrix generated from the most effective protocol for 64 spins (i.e., the protocol matching best with the predictions of the matrix product state algorithm for both the ground-state energy and spin-spin correlations when the stopping criterion is met).

For the Ising model, as one can see, the $(L, 2)$ tiling is the most effective protocol in all instances. One may also note that the cold start fails to provide an accurate and robust energy prediction except in the paramagnetic phase ($J_I = 0.5$). In the antiferromagnetic phase, as expected, the $(1, 2)$ -tiling protocol proves particularly ineffective and even worse than the cold-start result. It is also less robust since it shows significant sample-to-sample variations compared to the other tiling protocols.

For the Heisenberg model, the $(L, 2)$ tiling is again the most effective protocol except in the ferromagnetic phase

FIG. 8. Ferromagnetic (C_d^F) and antiferromagnetic (C_d^A) ground-state correlators for one realization of a one-dimensional system of 128 spins as a function of the spin-spin separation d ($d = 1, \dots, 60$). The different panels give the results obtained with the matrix product state algorithm, the cold-start, and the different transfer learning protocols when the stopping criterion is met. The initial weight matrix of the different tiling protocols is generated from the fine-tuned weights produced by the most effective transfer learning protocol operated on the same system with 64 spins. The ferromagnetic correlator C_d^F is shown for the (a) Ising ($J_I = 2$ and $h = 1$) and (b) Heisenberg ($\Delta = 2$ and $J_{XXZ} = 1$) models and the antiferromagnetic correlator C_d^A is shown for the (c) Ising ($J_I = -2$ and $h = 1$) and (d) Heisenberg ($\Delta = -2$ and $J_{XXZ} = 1$) models. The cyan solid line shows the matrix product state result, the blue solid line the cold-start protocol, the red dashed line the $(L, 2)$ -tiling protocol, the green dash-dotted line the $(1, 2)$ -tiling protocol, and the black dotted line the $(2, 2)$ -tiling protocol.

($\Delta = 2$). Noticeably, the $(1, 2)$ -tiling and $(2, 2)$ -tiling protocols are less effective than a cold start except in the ferromagnetic phase ($\Delta = 2$). In our simulations, we have also noticed that some realizations fail to converge when using the $(1, 2)$ -tiling and $(2, 2)$ -tiling protocols with $\Delta = -0.5$ and $\Delta = -2$. This happens when the state initialized by the transfer learning protocol generates large gradients. Then, instead of minimizing the energy, the optimization algorithm is driven towards high-energy states instead of low-energy ones. These extreme cases are not taken into account for the calculation of the error bars presented in Fig. 7(b).

Another way of assessing the effectiveness of the different transfer learning protocols is to compute the ground-state correlations between two distant spins and compare the results to accurate calculations using a matrix product state algorithm. In Fig. 8 we consider a single realization of the Ising and Heisenberg models in their ferromagnetic and antiferromagnetic phases and plot the correlators C_d^A and C_d^F [Eqs. (12) and (13)] as a function of the spin-spin distance d . For the ferromagnetic Heisenberg model, the spin-spin correlation starts to decay around $d \approx 15$ for the cold-start protocol and around $d \approx 30$ for the $(L, 2)$ -tiling protocol [see Fig. 8(b)]. This ineffectiveness of the $(L, 2)$ -tiling protocol is expected in this magnetization-conserving case (see Sec. IV) since it builds on the configuration of the $L_v = 64$ system,

FIG. 9. Zoomed-in version of Fig. 8 to better see the performance achieved by each transfer learning protocol. We refer to the caption of Fig. 8 for a complete description of this figure.

which has itself a domain wall located around $d \simeq 32$. As already shown earlier, we observe in Figs. 8(c) and 8(d) that the (1,2)-tiling protocol performs worst for both models in their antiferromagnetic phases.

Figure 9 shows the zoomed-in version of Fig. 8 to better compare which transfer learning protocol performs the best. As one can see, the $(L, 2)$ tiling is the best protocol (it reproduces very well the matrix product state simulations) except for the ferromagnetic Heisenberg model, where it completely fails, in both effectiveness (correlations and energy) and efficiency, as previously seen in Fig. 4(f). In this case, the (2,2)-tiling protocol performs significantly better. Furthermore, cold starts are less effective than some of the transfer learning protocols for both models and quantum phases.

4. Alternate scenarios for size increase

Up to now, we have only considered size-doubling scenarios. However, it is also possible to use transfer learning protocols when system sizes are increased by other increments. We consider here concatenation of different (L, p) -tiling protocols (see the end of Sec. IV) to reach 128 spins starting from four spins for the one-dimensional Ising model. For the sake of clarity, we will pictorially represent an (L, p) -tiling step by $L \xrightarrow{p} pL$. In Fig. 10 we evaluate the efficiency and effectiveness of the following seven scenarios that we run until each reaches its stopping criterion.

- (i) SCE:JUMP implements $4 \xrightarrow{32} 128$.
- (ii) SCE:DOUBLE implements $4 \xrightarrow{2} 8 \xrightarrow{2} \dots \xrightarrow{2} 64 \xrightarrow{2} 128$.
- (iii) SCE:4-8 implements $4 \xrightarrow{4} 16 \xrightarrow{8} 128$.
- (iv) SCE:8-4 implements $4 \xrightarrow{8} 32 \xrightarrow{4} 128$.
- (v) SCE:2-4-4 implements $4 \xrightarrow{2} 8 \xrightarrow{4} 32 \xrightarrow{4} 128$.
- (vi) SCE:4-2-4 implements $4 \xrightarrow{4} 16 \xrightarrow{2} 32 \xrightarrow{4} 128$.
- (vii) SCE:4-4-2 implements $4 \xrightarrow{4} 16 \xrightarrow{4} 64 \xrightarrow{2} 128$.

FIG. 10. Efficiency and effectiveness of different size-increase scenarios to reach 128 spins starting from four spins for the one-dimensional Ising model at $h = 1$ for different J_I . All quantities are computed until the stopping criterion is reached. (a) Efficiency as reflected by the computation time t . (b) Effectiveness as reflected by the ground-state energy error $\Delta E/E_0$ relative to the matrix product state simulation. (c) Effectiveness as reflected by the spin-spin correlation errors $\Delta C_{60,0}^F/C_0^F$ ($J_I = 2$) and $\Delta C_{60,0}^A/C_0^A$ ($J_I = -2$) computed at spin separation distance $d = 60$ and relative to the matrix product state simulation. The different scenarios tested are (see the text for full details) SCE:JUMP (black open bar), SCE:DOUBLE (blue closed bar), SCE:4-8 (green open bar), SCE:8-4 (red closed bar), SCE:2-4-4 (cyan closed bar), SCE:4-2-4 (blue open bar), and SCE:4-4-2 (green closed bar). The error bars represent the interval of the 9th percentile and of the 91st percentile.

We observe that SCE:JUMP performs the worst in both the ferromagnetic and antiferromagnetic phases. This is because the initial system is too small to develop reliable long-range correlations and cannot readily approximate the large system. Nevertheless, we note that SCE:JUMP still performs better than a cold start. As a reference, the average cold-start computation time is about 292 s in the Ising antiferromagnetic phase. This is about $\frac{1}{8}$ the time of SCE:JUMP. Furthermore, the relative energy error of SCE:JUMP is 30% better than a cold start.

We observe that there is a trade-off between efficiency and effectiveness in the choice of the transfer learning protocols. For example, SCE:DOUBLE is effective but not efficient. As an alternative, SCE:4-4-2 is similarly effective but more efficient. The most efficient scenario is SCE:8-4. It is also effective in the ferromagnetic phase but loses some effectiveness in the antiferromagnetic phase. In the paramagnetic phase, all scenarios work similarly well. We also observe that it is better to have large increments in early transfers rather

FIG. 11. Efficiency and effectiveness of cold-start and different transfer learning protocols run for the two-dimensional isotropic antiferromagnetic Heisenberg model ($J_{XXZ} = 1$ and $\Delta = -1$) on a square lattice of linear size D with open boundary conditions until the stopping criterion is reached. (a) and (b) Efficiency as reflected by the computation times t (in seconds) achieved by the different transfer learning protocols for a lattice of linear size (a) $D = 4$ and (b) $D = 8$. (c) and (d) Effectiveness as reflected by the computed ground-state energy for (c) $D = 4$ and (d) $D = 8$, with the mean relative error $\Delta E/E_0$ achieved by the cold-start and different transfer learning protocols as compared to quantum Monte Carlo simulations. The blue open bars are for the cold-start protocol, the red closed bars are for the $(L, 2)$ -tiling protocol, the green closed bars are for the $(1, 2)$ -tiling protocol, and the black open bars are for the $(2, 2)$ -tiling protocol. (e) and (f) Effectiveness as reflected by computing the antiferromagnetic correlator $(-1)^l \langle \sigma_i^z \sigma_{i+l}^z \rangle$ for $D = 8$ as a function of l from lattice node $\mathbf{i} = (4, 5)$. Note that the correlator equals 1 for $l = 0$. (e) Correlations along the diagonal direction $\mathbf{e} = (1, 1)$. (f) Correlations along the row direction $\mathbf{e} = (1, 0)$. The cyan solid line corresponds to quantum Monte Carlo simulations, the blue solid line corresponds to the cold-start protocol, the red dashed line corresponds to the $(L, 2)$ -tiling protocol, the green dash-dotted line corresponds to the $(1, 2)$ -tiling protocol, and the black dotted line corresponds to the $(2, 2)$ -tiling protocol.

than in later ones. For instance, SCE:8-4 is more efficient and effective than SCE:4-8. These observations, made for the Ising model, might not be transferable to other Hamiltonian systems.

5. Two-dimensional systems

The previous transfer learning protocols are not confined to one-dimensional systems. In the following, we test them on the two-dimensional isotropic antiferromagnetic Heisenberg model ($\Delta = -1$ and $J_{XXZ} = 1$) for a square lattice of linear size D with open boundary conditions. The number of visible nodes of the restricted Boltzmann machines is thus $L_v = D^2$.

In Fig. 11 we assess the efficiency and effectiveness of the different transfer learning protocols for system sizes $D = 4, 8$. We show the computation times t (in seconds) in Figs. 11(a)

and 11(b) and the relative error on the energy in Figs. 11(c) and 11(d) as compared to quantum Monte Carlo simulations [2]. In Fig. 11(e) we plot the antiferromagnetic correlator $(-1)^l \langle \sigma_i^z \sigma_{i+l}^z \rangle$ as a function of l along the diagonal direction $\mathbf{e} = (1, 1)$ for $D = 8$. The same quantity is plotted in Fig. 11(f) but along the row direction $\mathbf{e} = (1, 0)$. In both plots, we have chosen the reference spin to be located at position $\mathbf{i} = (4, 5)$ in the center of the square lattice. Similarly to the one-dimensional case, we see that the $(L, 2)$ -tiling protocol performs the best for this isotropic antiferromagnetic Heisenberg model in terms of combined efficiency and effectiveness. For $D = 8$ in Figs. 11(d)–11(f), the cold-start protocol performed as well as $(L, 2)$ tiling in terms of effectiveness. It is followed by the $(2, 2)$ -tiling and $(1, 2)$ -tiling protocols. The $(1, 2)$ -tiling protocol in particular performs worst in terms of correlations as seen in Figs. 11(e) and 11(f). We also observe that all the transfer learning protocols are quite robust since the error bars are relatively small, except for the time required to reach the stopping criterion for the $(1, 2)$ -tiling protocol.

VII. CONCLUSION

We have proposed the idea of transfer learning for the scalability of neural-network quantum states. Among a multitude of candidates in the design space of transfer learning protocols for the problem at hand, we devised and presented a selected few that seem to be amenable to physical interpretation. We evaluated and compared the performances of these transfer learning protocols, designed for neural-network quantum states, in terms of efficiency (speed) and effectiveness (accuracy) for different physical models in different physical phases, in both one and two dimensions.

We have considered two prototypical physical models, one described by the transverse field Ising Hamiltonian and the other one by the Heisenberg XXZ one. If these two models exhibit similar magnetic phases, they differ on an important physical property. Indeed, contrary to the transverse field Ising model, the Heisenberg XXZ model conserves the total magnetization. This means that its configuration space can be partitioned in uncoupled magnetic sectors with fixed total magnetization. For the Ising model, all these sectors are coupled. Because of this important difference, each transfer learning protocol performs differently for the two models. In the case of the Heisenberg model, we can restrict the transfer learning protocols to a single magnetic sector (we chose the one with zero total magnetization) whereas the full configuration space must be considered for the Ising model.

The best performing protocol for the one-dimensional transverse field Ising model is what we call $(L, 2)$ tiling. It is also generally competitive for the one-dimensional and two-dimensional Heisenberg XXZ models except for the ferromagnetic phase where the $(1, 2)$ -tiling and $(2, 2)$ -tiling protocols perform better. This is due to the zero-magnetization constraint imposed on the system. We also investigated several other size-increasing scenarios for the one-dimensional Ising model, beyond mere consecutive size-doubling steps. We observed that there is a trade-off to make between efficiency and effectiveness between these different size-increasing scenarios. We found that, in general, larger size increases at the beginning of the transfer learning process and smaller

increases later are preferable, although a cold start for too large initial sizes may not return a good enough initial state for ensuing transfer learning protocols. Our findings and observations call for further studies to better characterize and understand the physical guiding principles behind the success of a given transfer learning protocol and its dependence on the physical system and phase considered.

In summary, our empirical results demonstrate that transfer learning for the scalability of neural-network quantum states can provide two advantages: (i) efficiency, which allows us to reach a good approximation of the ground state in shorter times compared to a cold start, and (ii) effectiveness, which reduces the chances that the optimization procedure remains trapped in a local minimum far from the ground state.

For the transfer learning to be productive, the chosen tiling protocol must be adapted to the physical model considered and its targeted ground-state phase. If the characteristic physical properties of the targeted ground-state phase are not preserved in the transfer protocol, then the method becomes rapidly inefficient and ineffective, as is the case for the (1,2)-tiling protocol in the antiferromagnetic phase. On the contrary, transfer learning protocols that preserve such characteristic physical properties generally provide good scalability.

Our exploration of different tiling methods is of course not exhaustive and other protocols could provide even better scalability. We leave for future work the exploration of other strategies to leverage transfer learning for neural-network quantum states, the investigation of transfer learning between neural-network quantum states for systems of equal sizes described by the same Hamiltonian in different parameter regimes [31] as well as between neural-network quantum states for systems described by different Hamiltonians, and the consideration of transfer learning on restricted Boltzmann machines with complex-valued weights and with symmetries such as translational invariance.

ACKNOWLEDGMENTS

We acknowledge C. Guo and Supremacy Future Technologies for support on the matrix product state simulations. This work was partially funded by the National University of Singapore, the French Ministry of European and Foreign Affairs, and the French Ministry of Higher Education, Research and Innovation under the Merlion program as Merlion Project “Deep Quantum.” Some of the experiments reported in this article were performed on the infrastructure of Singapore National Supercomputing Centre [32] and were funded under project “Computing the Deep Quantum.”

APPENDIX: CONVERGENCE OF THE ENERGY VS NUMBER OF ITERATIONS

Figures 12(a) and 12(b) show plots of the ground-state energy versus the number of iterations $N \geq 1$ for different transfer learning protocols run for one realization of the

FIG. 12. Ground-state energy E as a function of the number of iterations $N \geq 1$ for one realization of the one-dimensional Ising model with 128 spins: (a) the ferromagnetic ($J_l = 2$ and $h = 1$) Ising model and (b) the antiferromagnetic ($J_l = -2$ and $h = 1$) Ising model. To smooth out the computed curves, the energy value plotted at iteration N is the average of the energies obtained in the iteration window $[N_0, N]$, where $N_0 = \max(N - 100, 0)$. The blue solid line shows the cold-start protocol, the red dashed line the $(L, 2)$ -tiling protocol, the black dotted line the $(2,2)$ -tiling protocol, the green dash-dotted line the $(1,2)$ -tiling protocol, and the cyan solid line the matrix product state simulation.

one-dimensional ferromagnetic and antiferromagnetic Ising models with 128 spins, respectively. For each protocol, the initial weights are those obtained at the stopping criterion by the same protocol run for 64 spins except, naturally, for the cold-start protocol.

For the ferromagnetic case ($J_l = 2$ and $h = 1$), all transfer learning protocols start with an energy below the cold-start one. We also see that the $(L, 2)$ -tiling protocol (red dashed line) reaches the stopping criterion much earlier than the $(1,2)$ -tiling (green dash-dotted line) and $(2,2)$ -tiling (black dotted line) protocols even though its energy started above. Noticeably, the energy for the $(1,2)$ -tiling and $(2,2)$ -tiling protocols first increases before finally decreasing and converging to the value obtained with the matrix product state simulation. This behavior may indicate a trapping of their respective initial conditions in a local minimum.

For the antiferromagnetic case ($J_l = -2$ and $h = 1$), the $(1,2)$ tiling is the only transfer learning protocol starting with an energy above the cold start. We also note that the $(L, 2)$ -tiling protocol converges faster than the $(2,2)$ -tiling protocol even if it starts at a higher energy. This is an interesting point because, if the $(L, 2)$ -tiling protocol may achieve higher energy values than the other protocols right after weight transfer, we find that its energy at the stopping criterion is always the lowest. In practice, the weights generated by the $(L, 2)$ -tiling protocol allow one to reach a lower minimum of energy without getting trapped in some higher local minimum.

- [1] J. Gubernatis, N. Kawashima, and P. Werner, *Quantum Monte Carlo Methods: Algorithms for Lattice Models* (Cambridge University Press, Cambridge, 2016).
- [2] A. W. Sandvik, *Phys. Rev. B* **56**, 11678 (1997).

- [3] R. Orús, *Ann. Phys. (NY)* **349**, 117 (2014).
- [4] F. Verstraete, J. I. Cirac, and V. Murg, *Adv. Phys.* **57**, 143 (2008).
- [5] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).

- [6] U. Schollwöck, *Ann. Phys. (NY)* **326**, 96 (2011).
- [7] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
- [8] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, *Rev. Mod. Phys.* **68**, 13 (1996).
- [9] W. Metzner and D. Vollhardt, *Phys. Rev. Lett.* **62**, 324 (1989).
- [10] A. Georges and G. Kotliar, *Phys. Rev. B* **45**, 6479 (1992).
- [11] G. Carleo and M. Troyer, *Science* **355**, 602 (2017).
- [12] K. Choo, G. Carleo, N. Regnault, and T. Neupert, *Phys. Rev. Lett.* **121**, 167204 (2018).
- [13] S. Czischek, M. Gärtner, and T. Gasenzer, *Phys. Rev. B* **98**, 024311 (2018).
- [14] D.-L. Deng, X. Li, and S. Das Sarma, *Phys. Rev. X* **7**, 021021 (2017).
- [15] R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, *Nat. Phys.* **15**, 887 (2019).
- [16] S. Das Sarma, D.-L. Deng, and L.-M. Duan, *Phys. Today* **72** (3), 48 (2019).
- [17] G. Carleo, J. I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [18] K. Choo, T. Neupert, and G. Carleo, *Phys. Rev. B* **100**, 125124 (2019).
- [19] T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov, *Nat. Commun.* **11**, 1593 (2020).
- [20] G. Carleo, K. Choo, D. Hofmann, J. E. T. Smith, T. Westerhout, F. Alet, E. J. Davis, S. Efthymiou, I. Glasser, S.-H. Lin, M. Mauri, G. Mazzola, C. B. Mendl, E. van Nieuwenburg, O. O'Reilly, H. Théveniaut, G. Torlai, F. Vicentini, and A. Wietek, *SoftwareX* **10**, 100311 (2019).
- [21] R. Caruana, *Mach. Learn.* **28**, 41 (1997).
- [22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, 2016* (USENIX Association, Berkeley, 2016), Vol. 16, pp. 265–283.
- [23] Z.-A. Jia, B. Yi, R. Zhai, Y.-C. Wu, G.-C. Guo, and G.-P. Guo, *Adv. Quantum Technol.* **2**, 1800077 (2019).
- [24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (MIT Press, Cambridge, 2014), Vol. 2, pp. 3320–3328.
- [25] G. Hinton, N. Srivastava, and K. Swersky, Neural networks for machine learning, Lecture 6a: Overview of mini-batch gradient descent, University of Toronto, 2012 (unpublished), available at <https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf>.
- [26] D. P. Kingma and J. Ba, in *Proceedings of the 3rd International Conference on Learning Representations, San Diego, 2015* (unpublished).
- [27] G. E. Hinton, in *Neural Networks: Tricks of the Trade*, edited by G. Montavon, G. B. Orr, and K. R. Müller, Lecture Notes in Computer Science Vol. 7700 (Springer, Berlin, 2012), pp. 599–619.
- [28] <https://github.com/remmyzen/nqs-tensorflow>.
- [29] S. Sorella, M. Casula, and D. Rocca, *J. Chem. Phys.* **127**, 014105 (2007).
- [30] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, *Phys. Rev. Lett.* **124**, 020503 (2020).
- [31] R. Zen, L. My, R. Tan, F. Hebert, M. Gattobigio, C. Miniatura, D. Poletti, and S. Bressan, *arXiv:2002.02618*.
- [32] <https://www.nscg.sg>