


From tensor-network quantum states to tensorial recurrent neural networks

Dian Wu¹,* Riccardo Rossi¹,† Filippo Vicentini¹,‡ and Giuseppe Carleo¹,§

Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

 (Received 20 March 2023; accepted 9 May 2023; published 5 July 2023)

We show that any matrix product state (MPS) can be exactly represented by a recurrent neural network (RNN) with a linear memory update. We generalize this RNN architecture to two-dimensional lattices using a multilinear memory update. It supports perfect sampling and wave-function evaluation in polynomial time, and can represent an area law of entanglement entropy. Numerical evidence shows that it can encode the wave function using a bond dimension lower by orders of magnitude when compared to MPS, with an accuracy that can be systematically improved by increasing the bond dimension.

DOI: [10.1103/PhysRevResearch.5.L032001](https://doi.org/10.1103/PhysRevResearch.5.L032001)

Introduction. Tensor networks (TNs) have been extensively used to represent the states of quantum many-body physical systems [1–3]. Matrix product states (MPS) are possibly the simplest family of TNs, and are suitable to capture the ground state of one-dimensional (1D) gapped Hamiltonians [4,5]. They can be contracted in polynomial time to compute physical quantities exactly, and optimized by density matrix renormalization group (DMRG) [6] when used as variational *Ansätze*. More powerful TN architectures that cannot be efficiently contracted in general have been proposed later, notably projected entangled pair states (PEPS) [7]. They are of interest mainly because they can describe area-law states in dimensions $D > 1$ [8]. In practice, they are usually approximately contracted and optimized with low-rank truncation [9–11]. They can also be optimized in the variational Monte Carlo (VMC) framework [12–15] using stochastic sampling rather than exact contraction.

In recent years, a growing trend has been to use TNs in machine learning tasks such as supervised [16,17] and unsupervised [18,19] learning. It is of particular interest to use them as generative models [20,21], where it is possible to draw perfect samples with tractable probability densities. Considering that many generative models are based on recurrent neural networks (RNNs), there have been attempts to use tensor operations in RNNs to improve their expressivity [22,23]. This kind of architecture has shown leading performance when applied to quantum physical systems [24].

Soon after the original proposal of neural quantum states (NQS) [25], several attempts have been made to link them to

ground states of well-known Hamiltonians and TN quantum states. While an arbitrary Hamiltonian might require an NQS with a depth growing exponentially with the system size [26], several Hamiltonians with desirable topological properties have compact representations in NQS [27–30].

Considering the relation between neural networks (NNs) and TNs, the first works focused on the restricted Boltzmann machines (RBMs), which are one of the simplest classes of NNs. It is impossible to efficiently map an RBM onto a TN, as they correspond to string-bond states with an arbitrary nonlocal geometry [28]. This result was later refined to show that an RBM may correspond to an MPS with an exponentially large bond dimension, and only short-range RBMs can be mapped onto efficiently computable entangled plaquette states [31]. Similar results have been obtained that deep Boltzmann machines with proper constraints can be mapped onto TNs that are efficiently computable through transfer matrix methods [32].

Among other classes of NNs, it has been shown that simple cases of convolutional neural networks (CNNs) and RNNs, which are also instances of arithmetic circuits or sum-product networks [33], can be efficiently mapped onto TNs. Such efficient mapping is no longer possible with modifications to those simple cases, including overlapped convolution kernels, stacked recurrent layers [34], or nonlocal score function [35], which suggests that the reuse of information in general NN architectures is an extensive source of expressivity. In the opposite direction, a mapping from arbitrary TNs to feedforward NNs can be constructed using NN layers that contract a set of tensor indices at a time [36].

In this Letter, we propose another family of variational *Ansätze* that shares characteristics with NNs and TNs. We start by establishing an exact mapping from MPS to an RNN with a linear update rule for the memory and a quadratic output layer, which we refer to as 1D MPS-RNN. We then propose a generalization of this architecture to two-dimensional (2D) lattices using a linear combination of memory components at previous neighboring sites in each memory update, which we call 2D MPS-RNN. After a theoretical discussion of the limitation on entanglement entropy due to the linear memory

*dian.wu@epfl.ch

†riccardo.rossi@epfl.ch

‡filippo.vicentini@epfl.ch

§giuseppe.carleo@epfl.ch

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

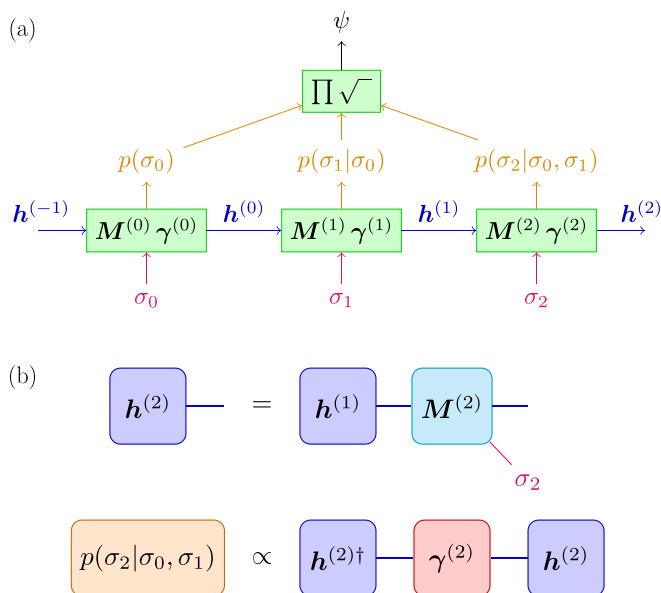


FIG. 1. (a) Computational graph of the vanilla MPS-RNN quantum state. The phase factor $\phi(\sigma)$ is omitted in the figure. (b) TN diagrams of the memory $\mathbf{h}^{(i)}$ and the conditional probability $p(\sigma_i|\sigma_{<i})$ for the vanilla MPS-RNN.

update, we propose a multilinear update with inspiration from the tensorial nature of the PEPS architecture. This *Ansatz* can represent states with an area law of entanglement entropy, which is particularly interesting in quantum physics. Unlike PEPS, our *Ansatz* is still a generative model, which supports perfect sampling and wave-function evaluation in polynomial time. Finally, we conduct numerical experiments to compare the performances of MPS and our *Ansätze* on the antiferromagnetic Heisenberg model (AFHM) on a square lattice and triangular lattice. The results show that our *Ansätze* produce lower variational energy with bond dimensions smaller by orders of magnitude compared to MPS, even in the presence of frustration, and the trade-off between computation cost and accuracy can be systematically controlled by the bond dimension, which is the only hyperparameter of our architectures.

RNN quantum states. We can write any wave function ψ of a quantum spin-1/2 system consisting of V sites in the form

$$\psi(\sigma) = \left(\prod_i \sqrt{p(\sigma_i|\sigma_{<i})} \right) e^{i\phi(\sigma)}, \quad (1)$$

where $\sigma = (\sigma_0, \dots, \sigma_{V-1})$ is the spin configuration in the z basis, and $p(\sigma_i|\sigma_{<i})$ is the conditional probability of measuring the spin i given the measurements of previous spins $\sigma_{<i} = (\sigma_0, \dots, \sigma_{i-1})$. An RNN quantum state [37] uses a memory vector $\mathbf{h}^{(i)}$ at each step to summarize the information about the previous spins $\sigma_{\leq i}$, which is updated by a function $\mathbf{h}^{(i)} = f_{\text{update}}^{(i)}(\sigma_i, \mathbf{h}^{(i-1)})$. Using this information, we compute each conditional probability from an output function $p(\sigma_i|\sigma_{<i}) = f_{\text{out}}^{(i)}(\mathbf{h}^{(i)})$. Therefore, the RNN quantum state is defined by the update function $f_{\text{update}}^{(i)}$, the output function $f_{\text{out}}^{(i)}$, and the phase function ϕ , as illustrated in Fig. 1(a). The purpose of introducing $\mathbf{h}^{(i)}$ instead of working directly with the spin values is the compression of the exponential amount of information in the

wave function while keeping a rich expressivity. The trade-off between tractability and expressivity is usually controlled by the complexity of $f_{\text{update}}^{(i)}$ and $f_{\text{out}}^{(i)}$, and the size of $\mathbf{h}^{(i)}$. A notable feature of RNN quantum states is the perfect sampling from the probability distribution of the spins, without the need of Markov chains and therefore avoiding the problem of autocorrelation, which is particularly advantageous when stochastically optimizing the wave function by VMC.

Mapping MPS to RNN. In the following, we present an exact mapping from an MPS of bond dimension χ to a specific RNN architecture of memory dimension χ . Accordingly, in this Letter, we use the term ‘‘bond dimension’’ when referring to the dimension of the memory of our RNN architectures to underline the fact that they are conceptually similar. It is perhaps not completely surprising that a mapping from MPS to RNN exists as MPS has been shown to allow perfect sampling [38] and as it achieves an efficient compression of the information of the wave function for many quantum systems.

The MPS *Ansatz* for a wave function ψ is defined by

$$\psi(\sigma) = \sum_{s_0, \dots, s_V=0}^{\chi-1} \prod_{i=0}^{V-1} M_{\sigma_i; s_{i+1}, s_i}^{(i)}, \quad (2)$$

where $M_{\sigma_i}^{(i)}$ is a complex $\chi \times \chi$ matrix that depends on the spin σ_i at the site i . Note that we keep the indices s_0 and s_V at both ends to simplify the discussion. To rewrite the MPS as an RNN, we identify the intermediate result of the tensor contraction as the memory $\mathbf{h}^{(i)}$, which satisfies a local update rule

$$\mathbf{h}^{(i)} = M_{\sigma_i}^{(i)} \mathbf{h}^{(i-1)}, \quad (3)$$

where $\mathbf{h}^{(i)} \in \mathbb{C}^\chi$ is a vector for each site i , and implicitly depends on the previous spins $\sigma_{\leq i} = (\sigma_0, \dots, \sigma_i)$. The boundary condition is $\mathbf{h}^{(-1)} = (1, \dots, 1)$. This is useful as the conditional probability of the MPS is then proportional to a positive semidefinite quadratic form of the memory

$$p(\sigma_i|\sigma_{<i}) \propto [\mathbf{h}^{(i)}]^\dagger \boldsymbol{\gamma}^{(i)} \mathbf{h}^{(i)}, \quad (4)$$

where the explicit form of $\boldsymbol{\gamma}^{(i)}$ as a contraction of the $M_{\sigma}^{(i)}$ matrices is given for completeness in the Supplemental Material [39], Sec. S1. The phase of ψ can be obtained from the memory at the last site:

$$\phi(\sigma) = \arg \sum_s h_s^{(V-1)}. \quad (5)$$

We now have all the elements to introduce the *Ansatz* that we refer to as the vanilla MPS-RNN: Eq. (3) defines the memory update in terms of the variational parameters $M_{\sigma}^{(i)}$, the conditional probability is obtained from Eq. (4), and the phase is obtained from Eq. (5). See Fig. 1(b) for TN diagrammatic illustrations of the memory update and the conditional probability. We elevate each $\boldsymbol{\gamma}^{(i)}$ to be a free variational parameter, independent of $M_{\sigma}^{(i)}$. Therefore, while any MPS can be exactly mapped to a vanilla MPS-RNN with the same bond dimension, the opposite is not true, and the freedom of $\boldsymbol{\gamma}^{(i)}$ brings additional expressivity. As $M_{\sigma}^{(i)}$ and $\boldsymbol{\gamma}^{(i)}$ depend on the spatial position i , we can say that we are encoding the spatial dimension into the ‘‘time’’ dimension of the RNN.

We slightly generalize the vanilla MPS-RNN architecture to improve its numerical performance by adding the

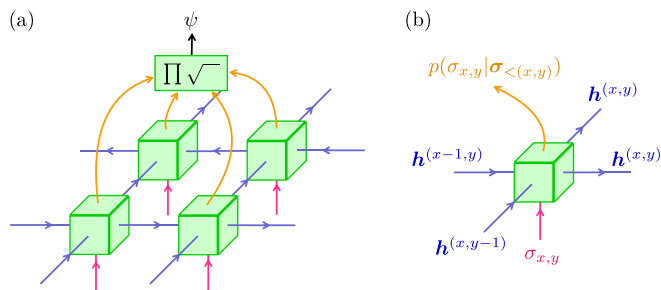


FIG. 2. (a) Computational graph of 2D MPS-RNN and tensor-RNN architectures for a square lattice with snake ordering. (b) Inputs and outputs at each site of 2D MPS-RNN and tensor-RNN architectures. For each site of the square lattice (x, y) , the RNN takes as input the memory vectors $\mathbf{h}^{(x-1, y)}$ and $\mathbf{h}^{(x, y-1)}$ from previous neighboring sites and the spin value $\sigma_{x, y}$ at the current site, and outputs the updated memory vector $\mathbf{h}^{(x, y)}$ to be used in future neighboring sites and the conditional probability $p(\sigma_{x, y} | \sigma_{<(x, y)})$.

normalization of the memory at each step and including a vectorial term in the memory update, as described in the Supplemental Material [39], Sec. S2. We call the resulting ansatz 1D MPS-RNN, which is the one we use in numerical experiments.

2D MPS-RNN. From the known success of MPS and its exact mapping to 1D MPS-RNN, we conclude that the latter can represent the ground states of gapped 1D systems. Therefore, the nonlinearity of conventional RNN is not necessary to efficiently represent short-range entanglement in one dimension, as 1D MPS-RNN only uses a linear memory update. By taking inspiration from the PEPS architecture, we consider a minimal generalization of the architecture to 2D systems, where we seek to efficiently approximate their ground states using linear or multilinear memory updates while keeping the computation time and memory scaling polynomially with the system size.

In the following, for simplicity, we limit our discussion to a square lattice of size $V = L \times L$. As the RNN sequentially outputs the conditional probability of a given spin at each step, we need to define a 1D ordering for the sites on the 2D lattice. We use the “snake” ordering as commonly used in MPS [40] and RNN [37] for 2D inputs, which is illustrated in Fig. 2. A minimal generalization of Eq. (3) to 2D consists in using a linear combination of the memory components at two previous neighboring sites, and we call the resulting architecture 2D MPS-RNN. We provide the TN diagram for the memory update of 2D MPS-RNN in Fig. 3, which is a generalization of Fig. 1(b). The equations are detailed in the Supplemental Material [39], Sec. S3.

A notable difference between 2D MPS-RNN and TN is that in the former each memory vector is used in two future neighboring sites. In a TN, on the contrary, each tensor only appears once in the contraction. This reuse of information can lead to a more efficient state compression using fewer parameters [34]. We remark that while the shallow recurrent arithmetic circuit (RAC) TN in Ref. [41] is only a subset of MPS, our vanilla MPS-RNN is a superset of MPS.

The direct information flow between vertically neighboring sites is an apparent advantage of 2D MPS-RNN over 1D architectures. Owing to this, the memory at a given site no

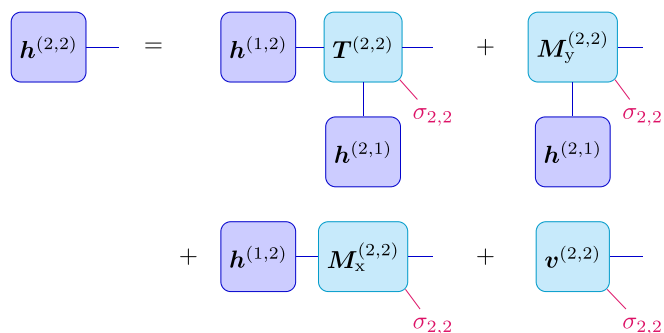


FIG. 3. TN diagram of the memory update for tensor-RNN. The memory update for 2D MPS-RNN is obtained for $\mathbf{T}^{(2,2)} = \mathbf{0}$, and the memory update for 1D MPS-RNN can be derived by further setting $\mathbf{M}_y^{(2,2)} = \mathbf{0}$.

longer needs to be carried along the $O(L)$ memory updates to influence the memory of a vertical neighbor of the 2D lattice. However, because of the linearity of the memory update proposed above, a 2D MPS-RNN with bond dimension χ can be exactly simulated by a 1D MPS-RNN with bond dimension $L\chi$, as proved in the Supplemental Material [39], Sec. S4. Therefore, 2D MPS-RNN can reduce the bond dimension χ at most linearly in L compared to 1D MPS-RNN rather than exponentially. In particular, its entanglement entropy cannot have an area law because it can scale at most logarithmically with the length L of a horizontal cut at a fixed χ . **Tensor-RNN.** We are therefore led to consider a nonlinear generalization of the memory update Eq. (3), and a minimal choice is a multilinear function of the memories at previous neighboring sites. The resulting memory update is sketched in Fig. 3 with the help of a TN diagram and detailed in the Supplemental Material [39], Sec. S3. We call this architecture tensor-RNN. It can represent states with an area law of entanglement entropy, and an example of this is constructed in the Supplemental Material [39], Sec. S5.

Compared to other methods employing 2D RNN as NQS, such as Refs. [24,37], our *Ansätze* are developed by generalizing TNs and therefore retain some advantages of TNs, namely (1) their expressivity is controlled by a single hyperparameter, the bond dimension, which avoids the laborious architecture search of RNN; (2) their linear or multilinear architectures are simple enough to enable theoretical analysis tools inspired from TNs, such as entanglement entropy; and (3) they provide an effective way of initialization when used in VMC, which we discuss below.

The *Ansätze* we have discussed form a hierarchy: $\text{MPS} \subsetneq \text{1D MPS-RNN} \subsetneq \text{2D MPS-RNN} \subsetneq \text{tensor-RNN}$, where an architecture on the right can exactly simulate an architecture on the left using the same bond dimension. We refer to this procedure as *hierarchical initialization*. During the variational optimization, we first use the DMRG algorithm to optimize a MPS, then use the optimized parameters to initialize a 1D MPS-RNN. After the gradient-based optimization of the 1D MPS-RNN *Ansatz*, we use it to initialize a 2D MPS-RNN, which after optimization can be used to initialize a tensor-RNN. This procedure provides reasonable starting points for each optimization and avoids being stuck early in a local minimum with high energy.

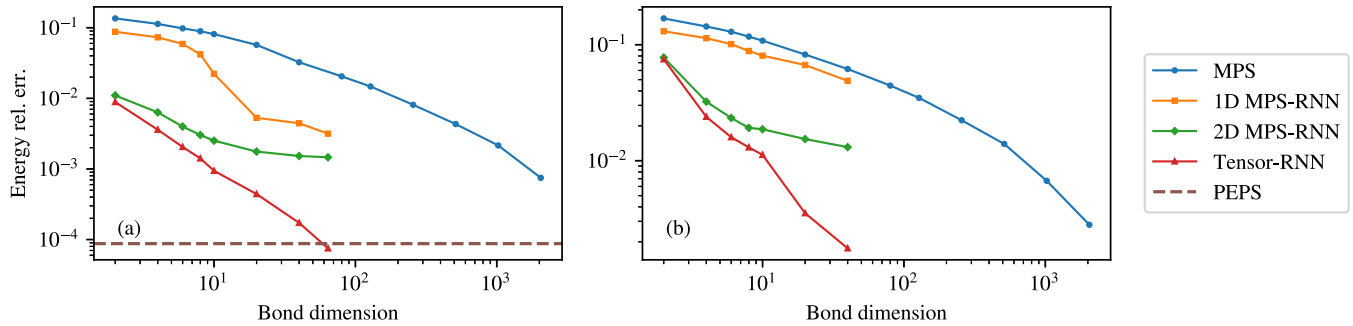


FIG. 4. Energy relative error of MPS (optimized by DMRG) and our *Ansätze* (optimized by VMC) as a function of the bond dimension χ . The Hamiltonian is AFHM on (a) 10×10 square lattice and (b) 10×10 triangular lattice, and the energy is compared to (a) QMC [42] and (b) DMRG with $\chi = 4096$. The PEPS result in (a) is taken from Ref. [43] with $\chi = 10$.

Numerical experiments. To numerically evaluate the performances of our proposed *Ansätze*, we start with the AFHM, defined by the Hamiltonian $\hat{H} = \sum_{\langle i,j \rangle} \hat{S}^{(i)} \cdot \hat{S}^{(j)}$, where $\langle i, j \rangle$ denotes a pair of nearest neighbors in the lattice. We perform numerical experiments on a 10×10 square lattice with open boundary conditions (OBCs). We apply the Marshall sign rule [44] to the Hamiltonian, which makes the ground-state wave function positive. We use the standard VMC method to optimize our *Ansätze*, with automatic differentiation (AD) [45] to compute the gradients, perfect sampling [46], Adam optimizer [47], and the hierarchical initialization described above. The variational energies are compared to the quantum Monte Carlo (QMC) result [42]. More information on the numerical implementation can be found in the Supplemental Material [39], Sec. S7. Figure 4(a) shows that we can systematically obtain lower variational energies by increasing the bond dimension χ . We remark that the memory updates of 2D MPS-RNN and tensor-RNN can reduce the bond dimension by one to two orders of magnitude, which confirms our expectation for 2D systems. The significant compression achieved by tensor-RNN over 2D MPS-RNN shows that the tensorial part of the memory update is crucial to capturing 2D quantum entanglement, as expected from the TN analogy.

Finally, we apply our *Ansätze* on a frustrated system, namely the AFHM on a 10×10 triangular lattice with OBC. The ground-state wave function can no longer be made positive by the Marshall sign rule, and the hierarchical initialization is particularly useful in this case, as it alleviates the difficulty of learning the sign structure of the wave function for frustrated systems [48,49]. Figure 4(c) shows that the reduction of bond dimension in 2D MPS-RNN and tensor-RNN is unaffected by the frustration. Moreover, a comparison of square and triangular lattices provides evidence that the multilinear memory update of tensor-RNN is important to capture the sign structure.

Conclusions. We have shown that an RNN can exactly represent an MPS of bond dimension χ with a linear

memory update, which we call 1D MPS-RNN. Using analogies with TNs and analytical arguments, we have proposed two minimal generalizations of the RNN architecture better suited to capturing 2D quantum entanglement while still using linear or multilinear memory updates. The tensorial version, tensor-RNN, can represent states with the area law of entanglement entropy while keeping the convenient properties of efficient, perfect sampling and wave-function evaluation. We have provided numerical evidence for the square AFHM that our *Ansätze* can achieve performance comparable to an MPS of bond dimension higher by orders of magnitude. These results have also been confirmed in the presence of frustration in the case of the triangular AFHM, where we have found that the hierarchical initialization from MPS to tensor-RNN is particularly useful to learn the sign structure of the wave function, which is generally a challenge for NQS.

Several possible future directions can be envisaged. In this work, we have not used one of the essential features of neural networks: nonlinear activation. Future work is needed to understand to what extent nonlinearities help increase the expressivity of representing quantum states and how they affect the optimization of tensorial RNNs. It has also been shown that deep architectures in modern neural networks can more efficiently produce entanglement [36], and our tensorial architectures can be generalized in a multilayer fashion. Another promising research direction for the architectures concerns physical symmetries, and we may incorporate the quantum number conservation and the $SU(2)$ symmetry of MPS [50] into our *Ansätze*. With those techniques, we expect tensorial RNNs to find broader applications.

Our code is available online [51].

Acknowledgments. We thank Jannes Nys and Or Sharir for their valuable remarks. Support from the Swiss National Science Foundation is acknowledged under Grant No. 200021_200336.

- [1] R. J. Baxter, Dimers on a rectangular lattice, *J. Math. Phys.* **9**, 650 (1968).
 [2] I. Affleck, T. Kennedy, E. H. Lieb, and H. Tasaki, Rigorous Results on Valence-Bond Ground States in Antiferromagnets, *Phys. Rev. Lett.* **59**, 799 (1987).

- [3] H. Takasaki, T. Hikihara, and T. Nishino, Fixed point of the finite system DMRG, *J. Phys. Soc. Jpn.* **68**, 1537 (1999).
 [4] F. Verstraete and J. I. Cirac, Matrix product states represent ground states faithfully, *Phys. Rev. B* **73**, 094423 (2006).

- [5] M. B. Hastings, An area law for one-dimensional quantum systems, *J. Stat. Mech.: Theory Exp.* (2007) P08024.
- [6] S. R. White, Density Matrix Formulation for Quantum Renormalization Groups, *Phys. Rev. Lett.* **69**, 2863 (1992).
- [7] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, Computational Complexity of Projected Entangled Pair States, *Phys. Rev. Lett.* **98**, 140506 (2007).
- [8] G. Lami, G. Carleo, and M. Collura, Matrix product states with backflow correlations, *Phys. Rev. B* **106**, L081111 (2022).
- [9] G. Evenbly, Algorithms for tensor network renormalization, *Phys. Rev. B* **95**, 045117 (2017).
- [10] F. Pan, P. Zhou, S. Li, and P. Zhang, Contracting Arbitrary Tensor Networks: General Approximate Algorithm and Applications in Graphical Models and Quantum Circuit Simulations, *Phys. Rev. Lett.* **125**, 060503 (2020).
- [11] L. Vanderstraeten, L. Burgelman, B. Ponsioen, M. Van Damme, B. Vanhecke, P. Corboz, J. Haegeman, and F. Verstraete, Variational methods for contracting projected entangled-pair states, *Phys. Rev. B* **105**, 195140 (2022).
- [12] A. W. Sandvik and G. Vidal, Variational Quantum Monte Carlo Simulations with Tensor-Network States, *Phys. Rev. Lett.* **99**, 220602 (2007).
- [13] A. Sfondrini, J. Cerrillo, N. Schuch, and J. I. Cirac, Simulating two- and three-dimensional frustrated quantum systems with string-bond states, *Phys. Rev. B* **81**, 214426 (2010).
- [14] W.-Y. Liu, Y.-Z. Huang, S.-S. Gong, and Z.-C. Gu, Accurate simulation for finite projected entangled pair states in two dimensions, *Phys. Rev. B* **103**, 235155 (2021).
- [15] T. Vieijra, J. Haegeman, F. Verstraete, and L. Vanderstraeten, Direct sampling of projected entangled-pair states, *Phys. Rev. B* **104**, 235141 (2021).
- [16] E. Stoudenmire and D. J. Schwab, Supervised learning with tensor networks, in *Advances in Neural Information Processing Systems*, Vol. 29 (Curran Associates, Red Hook, NY, 2016), pp. 4799–4807.
- [17] S. Cheng, L. Wang, and P. Zhang, Supervised learning with projected entangled pair states, *Phys. Rev. B* **103**, 125117 (2021).
- [18] Z.-Y. Han, J. Wang, H. Fan, L. Wang, and P. Zhang, Unsupervised Generative Modeling Using Matrix Product States, *Phys. Rev. X* **8**, 031012 (2018).
- [19] J. Liu, S. Li, J. Zhang, and P. Zhang, Tensor networks for unsupervised machine learning, *Phys. Rev. E* **107**, L012103 (2023).
- [20] S. Cheng, L. Wang, T. Xiang, and P. Zhang, Tree tensor networks for generative modeling, *Phys. Rev. B* **99**, 155131 (2019).
- [21] T. Vieijra, L. Vanderstraeten, and F. Verstraete, Generative modeling with projected entangled-pair states, [arXiv:2202.08177](https://arxiv.org/abs/2202.08177).
- [22] I. Sutskever, J. Martens, and G. E. Hinton, Generating text with recurrent neural networks, *Proceedings of the 28th International Conference on Machine Learning* (Omnipress, Madison, WI, 2011), pp. 1017–1024.
- [23] O. Irsoy and C. Cardie, Modeling compositionality with multiplicative recurrent neural networks, *3rd International Conference on Learning Representations* (ACM, New York, 2015).
- [24] M. Hibat-Allah, E. M. Inack, R. Wiersema, R. G. Melko, and J. Carrasquilla, Variational neural annealing, *Nat. Mach. Intell.* **3**, 952 (2021).
- [25] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [26] G. Carleo, Y. Nomura, and M. Imada, Constructing exact representations of quantum many-body systems with deep neural networks, *Nat. Commun.* **9**, 5322 (2018).
- [27] D.-L. Deng, X. Li, and S. Das Sarma, Machine learning topological states, *Phys. Rev. B* **96**, 195145 (2017).
- [28] I. Glasser, N. Pancotti, M. August, I. D. Rodriguez, and J. I. Cirac, Neural-Network Quantum States, String-Bond States, and Chiral Topological States, *Phys. Rev. X* **8**, 011006 (2018).
- [29] R. Kaubruegger, L. Pastori, and J. C. Budich, Chiral topological phases from artificial neural networks, *Phys. Rev. B* **97**, 195136 (2018).
- [30] S. Lu, X. Gao, and L.-M. Duan, Efficient representation of topologically ordered states with restricted Boltzmann machines, *Phys. Rev. B* **99**, 155136 (2019).
- [31] J. Chen, S. Cheng, H. Xie, L. Wang, and T. Xiang, Equivalence of restricted Boltzmann machines and tensor network states, *Phys. Rev. B* **97**, 085104 (2018).
- [32] L. Pastori, R. Kaubruegger, and J. C. Budich, Generalized transfer matrix states from artificial neural networks, *Phys. Rev. B* **99**, 165123 (2019).
- [33] H. Poon and P. Domingos, Sum-product networks: A new deep architecture, *IEEE International Conference on Computer Vision Workshops* (IEEE Computer Society, Washington, DC, 2011), pp. 689–690.
- [34] Y. Levine, O. Sharir, N. Cohen, and A. Shashua, Quantum Entanglement in Deep Learning Architectures, *Phys. Rev. Lett.* **122**, 065301 (2019).
- [35] V. Khrulkov, A. Novikov, and I. Oseledets, Expressive power of recurrent neural networks, *6th International Conference on Learning Representations* (ACM, New York, 2018).
- [36] O. Sharir, A. Shashua, and G. Carleo, Neural tensor contractions and the expressive power of deep neural quantum states, *Phys. Rev. B* **106**, 205136 (2022).
- [37] M. Hibat-Allah, M. Ganahl, L. E. Hayward, R. G. Melko, and J. Carrasquilla, Recurrent neural network wave functions, *Phys. Rev. Res.* **2**, 023358 (2020).
- [38] A. J. Ferris and G. Vidal, Perfect sampling with unitary tensor networks, *Phys. Rev. B* **85**, 165146 (2012).
- [39] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevResearch.5.L032001> for mathematical definitions of our tensorial recurrent neural network architectures, proofs for the computational complexity and the entanglement entropy, details of numerical experiments, and additional figures visualizing the properties of variational states, which includes Refs. [52–57].
- [40] E. M. Stoudenmire and S. R. White, Studying two-dimensional systems with the density matrix renormalization group, *Annu. Rev. Condens. Matter Phys.* **3**, 111 (2012).
- [41] Y. Levine, O. Sharir, A. Ziv, and A. Shashua, On the long-term memory of deep recurrent networks, [arXiv:1710.09431](https://arxiv.org/abs/1710.09431).
- [42] A. W. Sandvik, Finite-size scaling of the ground-state parameters of the two-dimensional Heisenberg model, *Phys. Rev. B* **56**, 11678 (1997).
- [43] W.-Y. Liu, S.-J. Dong, Y.-J. Han, G.-C. Guo, and L. He, Gradient optimization of finite projected entangled pair states, *Phys. Rev. B* **95**, 195154 (2017).
- [44] W. Marshall, Antiferromagnetism, *Proc. R. Soc. London, Ser. A* **232**, 48 (1955).

- [45] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, Automatic differentiation in machine learning: A survey, *J. Mach. Learn. Res.* **18**, 1 (2018).
- [46] O. Sharir, Y. Levine, N. Wies, G. Carleo, and A. Shashua, Deep Autoregressive Models for the Efficient Variational Simulation of Many-Body Quantum Systems, *Phys. Rev. Lett.* **124**, 020503 (2020).
- [47] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, *3rd International Conference on Learning Representations* (Ref. [23]).
- [48] T. Westerhout, N. Astrakhantsev, K. S. Tikhonov, M. I. Katsnelson, and A. A. Bagrov, Generalization properties of neural network approximations to frustrated magnet ground states, *Nat. Commun.* **11**, 1593 (2020).
- [49] M. Bukov, M. Schmitt, and M. Dupont, Learning the ground state of a non-stoquastic quantum Hamiltonian in a rugged neural network landscape, *SciPost Phys.* **10**, 147 (2021).
- [50] S.-S. Gong, W. Zhu, D. N. Sheng, O. I. Motrunich, and M. P. A. Fisher, Plaquette Ordered Phase and Quantum Phase Diagram in the Spin-1/2 J_1 - J_2 Square Heisenberg Model, *Phys. Rev. Lett.* **113**, 027201 (2014).
- [51] <https://github.com/cqsl/mps-rnn>.
- [52] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**, 1735 (1997).
- [53] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (Association for Computational Linguistics, Stroudsburg, PA, 2014), pp. 103–111.
- [54] L. R. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika* **31**, 279 (1966).
- [55] 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. Th eveniaut, G. Torlai, F. Vicentini, and A. Wietek, NetKet: A machine learning toolkit for many-body quantum systems, *SoftwareX* **10**, 100311 (2019).
- [56] F. Vicentini, D. Hofmann, A. Szab o, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderon, N. Astrakhantsev, and G. Carleo, NetKet 3: Machine learning toolbox for many-body quantum systems, *SciPost Phys. Codebases*, **7** (2022), doi:10.21468/SciPostPhysCodeb.7.
- [57] M. Fishman, S. R. White, and E. M. Stoudenmire, The ITensor software library for tensor network calculations, *SciPost Phys. Codebases*, **4** (2022), doi:10.21468/SciPostPhysCodeb.4.