Scalable variational Monte Carlo
with graph neural ansatz

Li Yang
Google Research
Mountain View, CA 94043, USA
`lyliyang@google.com`

Wenjun Hu
Department of Physics and Astronomy
University of Tennessee
Knoxville, TN 37996, USA
`whu11@utk.edu`

Li Li
Google Research
Mountain View, CA 94043, USA
`leeley@google.com`

Abstract

Deep neural networks have been shown as a potentially powerful ansatz in variational Monte Carlo for solving quantum many-body problems. We propose two improvements in this direction. The first is graph neural ansatz (GNA), which is a variational wavefunction universal to arbitrary geometry. GNA results in accurate ground-state energies on 2D Kagome lattices, triangular lattices, and randomly connected graphs. Secondly, we design a distributed workflow on multiple accelerators to scale up the computation. We compute Kagome lattices with sizes up to 432 sites on 128 TPU cores. The parameter sharing nature of the GNA also leads to transferability across different system sizes and geometries.

1 Introduction

Quantum many-body systems have been studied for a century but still with many unsolved problems. Many numerical algorithms have been developed. For example, the tensor network [1–3] based methods have been successfully used for many low-dimensional systems, but their applications in two-dimensional and above are still under development. The quantum Monte Carlo (QMC) methods based on probabilistic sampling typically require a positive-semidefinite wavefunction [4, 5]. Variational Monte Carlo (VMC) [6] works well for arbitrary dimension and wavefunction signs. It requires a variational ansatz for the many-body wavefunction, which is a multi-dimensional complex-valued function whose input is a many-body state configuration and output is its amplitude. Previously, physicists designed them with the physical insight of the system [7], and in a form with much fewer parameters comparing with today's deep neural networks (DNN). Pioneer work by Carleo and Troyer [8] proposed to use a restricted Boltzmann machine (RBM) as the variational ansatz. Following this direction, the RBM and a few other shallow networks have been applied to study several quantum many-body systems with good accuracy [9–14]. Recently, DNNs has been used as variational ansatzes [15–18] and optimized with algorithms tailored to deep learning.

In this work, we incorporate two recent advances in deep learning into VMC, making it universal to the geometry of the system and scalable in system size. The first component is a graph neural ansatz (GNA) wavefunction, i.e. a graph neural network (GNN) [19] based variational ansatz. It can encode arbitrary geometries directly, while previously used convolutional neural networks (CNN) require additional artifacts to fit non-square lattice into grids. GNNs have been receiving increasing attention and have been successfully applied to many areas including social networks [20, 21],
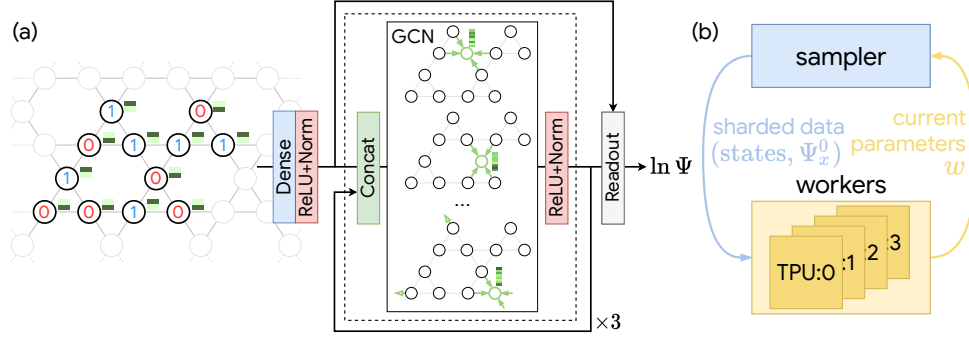
Figure 1: (a) Illustration of graph neural ansatz. (b) Distributed workflow of VMC using ISGO.

computer vision [22], combinatorial optimization [23], and more recently applications in physical sciences such as quantum chemistry [24–26], classical physics engines [27–29], glassy systems [30] and protein interactions [31]. As a demonstration of GNA, we use graph convolutional networks (GCN) [21] for various kinds of geometries including the 2D Kagome lattices, triangular lattices, and randomly connected graphs. The second component is a scalable implementation of importance sampling gradient optimization (ISGO) [18] on multiple accelerators (GPUs, TPUs). This makes the computation easily scaling up to $12 \times 12 \times 3$ Kagome lattices, which is typically impossible to fit into a single accelerator device.

Major contributions: 1. Proposed GNA as a universal ansatz wavefunction for VMC. 2. A design of the scalable implementation of VMC on accelerators, demonstrated on 2D Kagome lattice up to 432 sites with 128 TPU cores. 3. Showed transferability of GNA across different system sizes and geometries.

2 Graph neural ansatz for quantum many-body system

Existing studies of quantum many-body systems using VMC usually constrain on systems with structured geometries, e.g. with translational or rotational symmetries. Certain symmetry consideration is essential to the novel property of the system and brings intuition on physicists' design of ansatz. Recently, there is a trend to use DNNs as variational ansatzes since neural networks are universal approximators. Most of the applications are on 1D chains or 2D square lattices using RBMs or CNNs. Libraries such as NetKet [32] support non-square geometries, but do not generalize to arbitrary geometries nor run on accelerators. CNN has been shown to work well for approximating many-body ground states [18, 12, 17]. We attribute the success of CNNs to its inductive bias of capturing local connections and sharing of parameters. On the other hand, GNN can contain the inductive bias of capturing arbitrary connections, and also sharing of parameters among nodes and edges. In this paper, we use a special kind of GNN, GCN [21], as the GNA for various system geometries. In a GCN, each node has an embedding vector. Node embeddings at layer l are updated by

$$H^{(l+1)} = f([D^{-\frac{1}{2}}AD^{-\frac{1}{2}}H^{(l)}; H^{(l)}]W + b), \qquad (1)$$

where $H^{(l)} \in \mathbb{R}^{N_{\text{site}} \times F}$ is the node embeddings at layer l, N_{site} the total number of sites in the system, F the embedding size, A the adjacency matrix and D is a node degree diagonal matrix for normalizing the adjacency matrix. We also concatenate the messages from neighboring nodes with current embeddings along feature dimension and then linearly transform using $W \in \mathbb{R}^{2F \times F}$ and $b \in \mathbb{R}^F$. Note W and b are shared across GCN layers. Finally we apply nonlinear transformation $f(\cdot)$ using ReLU [33] followed by a layer normalization [34] to get the node embeddings at layer $l + 1$.

Fig 1(a) illustrates GNA on a Kagome lattice of size $2 \times 2 \times 3$. The periodic boundary conditions are enforced while constructing A. The sites on one edge of the system connect to the sites on the opposite edge. The state x, occupation of Bosons on each site $\{0, 1\}$, is one-hot encoded and then nonlinearly transformed to embeddings with size $F = 16$. Graph convolution in Eq. (1) is recurrently applied three times. Finally, we concatenate the output with input, sum up the embeddings overall sites to a single embedding vector, and map to a scalar $\ln \Psi$ using a dense layer of one unit.

3 Scaling up VMC to multiple accelerators

Consider a neural network $\Psi(x; w)$ as the variational ansatz with a set of trainable parameters w. The ansatz with optimal parameters $w^* = \arg\min_w E_v(w)$ approximates the ground state wavefunction, where the $E_v(w) = \langle \Psi(x; w) | H | \Psi(x; w) \rangle / \langle \Psi(x; w) | \Psi(x; w) \rangle$ is the variational energy. However, direct computation of $E_v(w)$ is infeasible due to the exponential size of the Hilbert space. A stochastic optimization approach is used. First N_{sample} quantum states following distribution $P_x^0 \propto |\Psi_x^0|^2$ are sampled using the Markov-chain Monte Carlo (MCMC) method. Here Ψ^0 is the wavefunction under the current parameter w, and x indexes the sampled states. Then the gradients $\partial_w E_v$ can be approximated stochastically,

$$\partial_w E_v \approx \sum_x I^0 E_x^0 \partial_w \ln \Psi_x^0 - \sum_x I^0 E_x^0 \sum_x I^0 \partial_w \ln \Psi_x^0, \tag{2}$$

where $E_x^0 = \sum_{x'} H_{x,x'} \Psi_{x'}^0 / \Psi_x^0$ is defined as the local energy under Ψ^0 and $I^0 = 2/N_{\text{sample}}$. Then the parameters are updated by $w \leftarrow w - \alpha \partial_w E_v$. In conventional VMC, after the parameters are updated once, another iteration of sampling begins. This process makes VMC highly inefficient while using DNNs as the variational ansatz. Since the MCMC sampling must be long enough to obtain a faithful sample distribution following P_x^0. The ISGO method [18] has been developed to overcome this issue by reusing samples for parameters updating. Note that it is incorrect to simply use the same set of samples to perform multiple updates with Eq. (2), since after each update, the samples following distribution of old wavefunction will not follow the distribution of the new wavefunction. The key to resolve this is to renormalize the distribution of those mismatched samples to P_x^0 by multiplying the local energies and derivatives in Eq. (2) with importance sampling:

$$\partial_w E_v \approx \sum_x I_x E_x \partial_w \ln \Psi_x - \sum_x I_x E_x \sum_x I_x \partial_w \ln \Psi_x, \tag{3}$$

where E_x is the local energy under the new wavefunction Ψ, and $I_x/I^0 = P_x/P_x^0 = \mathcal{C}|\Psi_x|^2/|\Psi_x^0|^2$ with $\mathcal{C}$ the normalization factor is approximated by using $\sum_x I_x/I^0 = 1$. This makes VMC sample efficient with DNN ansatzes.

For large systems, the number of samples from MCMC needs to be large for reliable numerical results. The memory of a single accelerator is the bottleneck. Therefore, we scale up Eq. (3) on multiple accelerator workers as shown in Fig 1(b). One cycle from sampler to workers and back is one *iteration*. In each iteration, the MCMC sampler receives parameters w in the current checkpoint and uses it to sample states and compute their corresponding $\ln \Psi$. Duplicated states are grouped into unique states with counts. The data are sharded and distributed back to the workers. Workers are single-program multiple-data (SPMD) programs and each worker governs a set of accelerator devices. Each worker computes local values I_x, E_x, $\partial_w \ln \Psi_x$ in parallel. And global values $\sum_x I_x E_x$ and $\sum_x I_x$ are computed by the efficient all-reduce sum supported by many deep learning frameworks and hardware, e.g. `psum` in JAX [35]. Finally, the local gradients are also averaged over workers by the all-reduce mean and parameters on each work are updated identically. After a certain number of updating steps, a checkpoint is saved, and the sampler uses this new checkpoint to start the next iteration.

4 Numerical Experiments

We demonstrate scalable VMC with GNA on hard-core Boson systems. The Hamiltonian is

$$H = -t \sum_{\langle ij \rangle} (b_i^\dagger b_j + b_j^\dagger b_i) + V \sum_{\langle ij \rangle} n_i n_j, \tag{4}$$

where $b_i^\dagger$ (b_i) are creation (annihilation) operators of hard-core Bosons on site i. Two hard-core Bosons can not occupy the same site. $n_i = b_i^\dagger b_i$ are Boson number operators on site i. The first term in Eq. (4) is the tunneling, and the second term is the nearest neighbor interaction. The tunneling parameter $t > 0$ varies and the interaction strength $V = 1$ is fixed. Since the off-diagonal elements in Eq. (4) are always negative, it can be proved that the ground state wavefunction is always positive semi-definite. So in this paper, we use GNAs with real-valued parameters only. To show the representation power of GNA, we consider the Hamiltonian on three different geometries with interesting physics properties,

differing by the nearest neighbors $\langle ij \rangle$. *Kagome lattice:* as t/V increases, the system undergoes a valence bond solid to superfluid phase transition. Whether the phase transition is weakly first-order or a continuous superfluid-solid transition is still under investigation [36, 37]. *Triangular lattice:* as t/V increases, the system undergoes a first-order supersolid to superfluid transition [38–41]. *randomly connected graph:* recently, disordered models such as Sachdev-Ye-Kitaev model [42, 43] received much attention. We consider a simple model of hard-core Bosons on a randomly connected graph. The probability of arbitrary two sites being connected is 0.5. These geometries are shown in Fig 2. Throughout the experiments, we use 5 Markov chains each with length 10^4, 500 sampling iterations, and 100 parameter updating steps within each sampling iteration. In practice, we find the energy often stops changing too much after 100 updating steps within one iteration. And the energy converges after a few sampling iterations (on the order of 10). Using more parameter updating steps or fewer samples per iteration may lead to gradient explosion, because if the initial sampling distribution is too far from the instantaneous wavefunction, reweighting will become catastrophically bad. However, the exploration of such hyperparameters is not the focus of this paper. Yang et al. [18] presents a comparison of the ISGO and conventional VMC algorithm on computational speed.

4.1 Benchmarking ground state energies with exact diagonalization

In Fig 2, we compute the ground state energies using VMC with GNA for (a) a 2D Kagome lattice, (b) a triangular lattice, (c) randomly connected graphs. The mean and standard deviation of the ground state energy is calculated from the variational energies of the last 50 sampling iterations. Since they are small systems, the exact results can be computed via exact diagonalization. As shown in Fig 2(d), VMC with GNA consistently produces accurate ground-state energies comparing to the exact solutions among different geometries and t/V.
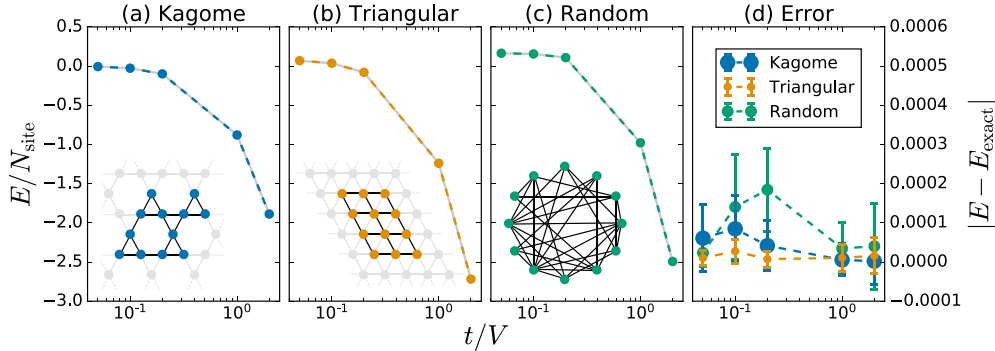

Figure 2: Comparison of ground state energies between the VMC with GNA (color) and exact diagonalization (gray) for (a) the $2 \times 2 \times 3$ Kagome lattice, (b) the 3×4 triangular lattice and (c) randomly connected graph with 12 sites. For Kagome and triangular lattices, the periodic boundary condition is enforced along with two primitive directions, so the lattice form a torus.
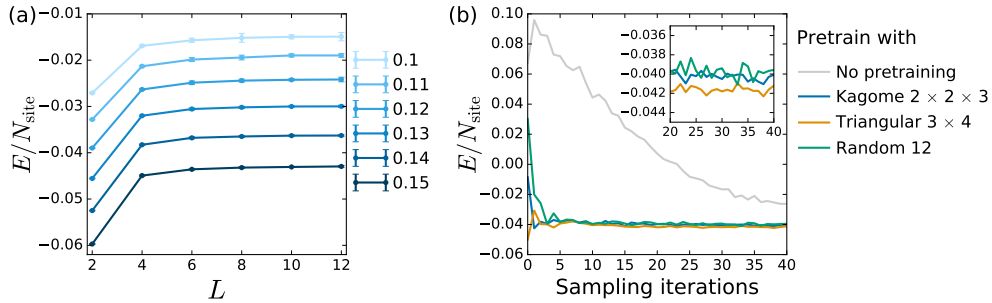

Figure 3: (a) Ground state energies of Kagome lattice verses system sizes L. The total number of sites is $L \times L \times 3$. Different curves denote various t/V. We use 8 TPU cores for $L < 8$ and up to 128 TPU cores for L up to 12. (b) $12 \times 12 \times 3$ Kagome lattice with different pretraining from 12 sites.

4

4.2 Solving Kagome lattice with large system sizes

Fig 3(a) shows the scalability of our approach, we compute the 2D Kagome lattice up to $12 \times 12 \times 3$, which is 432 sites in total. In each iteration, 5×10^4 samples are generated. Since the memory of a typical GPU is about 16G, the previous ISGO method fails on this system as the sampled states and their intermediate activations in the GNA can not fit into a single device.

Another interesting property of GNA is that the trained parameters are transferable to different system sizes or even different geometries. Note that the second point is hard for CNN. It allows us to pretrain a GNA on small systems and apply it to larger systems or even other geometries. In Fig 3(b) we show that GNA pretrained on small systems with 12 sites significantly improves the convergence on the large system with 432 sites. Without pretraining, we observe that the GNA with randomly initialized parameters often leads to more unique states, or sometimes leads to gradient explosion. With the GNA pretrained on a Kagome lattice with only $1/36$ the size of the targeted large Kagome lattice, the computation converges within 5 sampling iterations while GNA without pretraining does not even at 40. Interestingly, this effect stands even when the pretraining geometries, e.g. triangular and randomly connected graph, are different from the target geometries.

5 Conclusion and future work

We have combined GCN, a specific form of GNN, with a scalable implementation of the ISGO algorithm for calculating the ground state energies of hard-core Bosons on various geometries. As a universal graph encoder, GCN allows us to extend the previous neural network ansatz to systems with arbitrary geometry. Moreover, the parameter sharing nature of GCN enables fine-tuning of the model on a large system from the model pretrained on small systems. With a scalable ISGO implementation, we compute ground state energies of Kagome lattices with size up to $12 \times 12 \times 3 = 432$ sites on 128 TPU cores. Given a trained GCN, we expect other physical quantities, such as correlation functions, can be computed for much larger system sizes. This will be helpful for exploring the quantum phase transition properties of the system. In this paper, we study positive semi-definite ground state wavefunctions, thus GCN with real-valued parameters is enough. Ongoing work includes extending the method for complex-valued wavefunctions to resolve the wavefunction sign problem, which will be an unified framework that can be used for computing ground states of strongly interacting Fermions, frustrated spins, or computing unitary dynamics. This could be achieved by various approaches, for example, complex neural networks [44–47] or two-head networks outputting the absolute value and angle.

Broader Impact

This research develops a scalable variational Monte Carlo (VMC) algorithm and a universal graph neural ansatz (GNA). The combination of them is applied to a fundamental scientific field — quantum many-body physics. Most of the ethical aspects and societal consequences are not applicable due to the fundamentality of this research. However, we would like to address one possible negative outcome if this method received much attention and further developed. Research organizations with rich computation resources, especially machine learning hardware accelerators, would benefit most from the idea proposed by this work. By contrast, small organizations and individual researchers would have less or even no advantage to taking part in the development of this field. However, there may be other alternatives available, e.g. more affordable commercial cloud services for research or national supercomputer centers.

References

[1] Steven R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863–2866, Nov 1992. doi: 10.1103/PhysRevLett.69.2863. URL https://link.aps.org/doi/10.1103/PhysRevLett.69.2863.

[2] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96 – 192, 2011. ISSN 0003-4916. doi: https://doi.org/10.1016/j.aop.2010.09.012. URL http://www.sciencedirect.com/science/article/pii/S0003491610001752. January 2011 Special Issue.

[3] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117 – 158, 2014. ISSN 0003-4916. doi: https://doi.org/10.1016/j.aop.2014.06.013. URL http://www.sciencedirect.com/science/article/pii/S0003491614001596.

[4] David Ceperley and Berni Alder. Quantum monte carlo. *Science*, 231(4738):555–560, 1986. ISSN 0036-8075. doi: 10.1126/science.231.4738.555. URL https://science.sciencemag.org/content/231/4738/555.

[5] W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum monte carlo simulations of solids. *Rev. Mod. Phys.*, 73:33–83, Jan 2001. doi: 10.1103/RevModPhys.73.33. URL https://link.aps.org/doi/10.1103/RevModPhys.73.33.

[6] W. L. McMillan. Ground state of liquid he^4. *Phys. Rev.*, 138:A442–A451, Apr 1965. doi: 10.1103/PhysRev.138.A442. URL https://link.aps.org/doi/10.1103/PhysRev.138.A442.

[7] Federico Becca and Sandro Sorella. *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press, 2017.

[8] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017. ISSN 0036-8075. doi: 10.1126/science.aag2302. URL https://science.sciencemag.org/content/355/6325/602.

[9] Yusuke Nomura, Andrew S. Darmawan, Youhei Yamaji, and Masatoshi Imada. Restricted boltzmann machine learning for solving strongly correlated quantum systems. *Phys. Rev. B*, 96:205152, Nov 2017. doi: 10.1103/PhysRevB.96.205152. URL https://link.aps.org/doi/10.1103/PhysRevB.96.205152.

[10] Hiroki Saito and Masaya Kato. Machine learning technique to find quantum many-body ground states of bosons on a lattice. *Journal of the Physical Society of Japan*, 87(1):014001, Jan 2018.

[11] Zi Cai and Jinguo Liu. Approximating quantum many-body wave functions using artificial neural networks. *Phys. Rev. B*, 97:035116, Jan 2018. doi: 10.1103/PhysRevB.97.035116. URL https://link.aps.org/doi/10.1103/PhysRevB.97.035116.

[12] Xiao Liang, Wen-Yuan Liu, Pei-Ze Lin, Guang-Can Guo, Yong-Sheng Zhang, and Lixin He. Solving frustrated quantum many-particle models with convolutional neural networks. *Phys. Rev. B*, 98:104426, Sep 2018. doi: 10.1103/PhysRevB.98.104426. URL https://link.aps.org/doi/10.1103/PhysRevB.98.104426.

[13] Kenny Choo, Giuseppe Carleo, Nicolas Regnault, and Titus Neupert. Symmetries and many-body excitations with neural-network quantum states. *Phys. Rev. Lett.*, 121:167204, Oct 2018. doi: 10.1103/PhysRevLett.121.167204. URL https://link.aps.org/doi/10.1103/PhysRevLett.121.167204.

[14] Raphael Kaubruegger, Lorenzo Pastori, and Jan Carl Budich. Chiral topological phases from artificial neural networks. *Phys. Rev. B*, 97:195136, May 2018. doi: 10.1103/PhysRevB.97.195136. URL https://link.aps.org/doi/10.1103/PhysRevB.97.195136.

[15] Dmitrii Kochkov and Bryan K. Clark. Variational optimization in the ai era: Computational graph states and supervised wave-function optimization. *arXiv preprint arXiv:1811.12423*, 2018.

[16] Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.*, 124:020503, Jan 2020. doi: 10.1103/PhysRevLett.124.020503. URL `https://link.aps.org/doi/10.1103/PhysRevLett.124.020503`.

[17] Kenny Choo, Titus Neupert, and Giuseppe Carleo. Two-dimensional frustrated $J_1 - J_2$ model studied with neural network quantum states. *Phys. Rev. B*, 100:125124, Sep 2019. doi: 10.1103/PhysRevB.100.125124. URL `https://link.aps.org/doi/10.1103/PhysRevB.100.125124`.

[18] Li Yang, Zhaoqi Leng, Guangyuan Yu, Ankit Patel, Wen-Jun Hu, and Han Pu. Deep learning-enhanced variational monte carlo method for quantum many-body physics. *Phys. Rev. Research*, 2:012039, Feb 2020. doi: 10.1103/PhysRevResearch.2.012039. URL `https://link.aps.org/doi/10.1103/PhysRevResearch.2.012039`.

[19] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[20] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30*, pages 1024–1034. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf`.

[21] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[22] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[23] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6348–6358. Curran Associates, Inc., 2017.

[24] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[25] Steven Kearnes, Li Li, and Patrick Riley. Decoding molecular graph embeddings with reinforcement learning. *arXiv preprint arXiv:1904.08915*, 2019.

[26] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.

[27] Alvaro Sanchez-Gonzalez, Nicolas Heess, Jost Tobias Springenberg, Josh Merel, Martin Riedmiller, Raia Hadsell, and Peter Battaglia. Graph networks as learnable physics engines for inference and control. volume 80 of *Proceedings of Machine Learning Research*, pages 4470–4479. PMLR, 10–15 Jul 2018. URL `http://proceedings.mlr.press/v80/sanchez-gonzalez18a.html`.

[28] P. Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and K. Kavukcuoglu. Interaction networks for learning about objects, relations and physics. In *NIPS*, 2016.

[29] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter W Battaglia. Learning to simulate complex physics with graph networks. *arXiv preprint arXiv:2002.09405*, 2020.

[30] V. Bapst, T. Keck, A. Grabska-Barwińska, C. Donner, E. D. Cubuk, S. S. Schoenholz, A. Obika, A. W. R. Nelson, T. Back, D. Hassabis, and P. Kohli. Unveiling the predictive power of static structure in glassy systems. *Nature Physics*, 16(4):448–454, 2020. doi: 10.1038/s41567-020-0842-8. URL https://doi.org/10.1038/s41567-020-0842-8.

[31] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6530–6539. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7231-protein-interface-prediction-using-graph-convolutional-networks.pdf.

[32] Giuseppe Carleo, Kenny Choo, Damian Hofmann, James E.T. Smith, Tom Westerhout, Fabien Alet, Emily J. Davis, Stavros Efthymiou, Ivan Glasser, Sheng-Hsuan Lin, Marta Mauri, Guglielmo Mazzola, Christian B. Mendl, Evert van Nieuwenburg, Ossian O'Reilly, Hugo Théveniaut, Giacomo Torlai, Filippo Vicentini, and Alexander Wietek. Netket: A machine learning toolkit for many-body quantum systems. *SoftwareX*, 10:100311, 2019. ISSN 2352-7110. doi: https://doi.org/10.1016/j.softx.2019.100311. URL http://www.sciencedirect.com/science/article/pii/S2352711019300974.

[33] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 807–814. Omnipress, 2010. URL https://icml.cc/Conferences/2010/papers/432.pdf.

[34] Jimmy Ba, J. Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

[35] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

[36] S. V. Isakov, S. Wessel, R. G. Melko, K. Sengupta, and Yong Baek Kim. Hard-core bosons on the kagome lattice: Valence-bond solids and their quantum melting. *Phys. Rev. Lett.*, 97: 147202, Oct 2006. doi: 10.1103/PhysRevLett.97.147202. URL https://link.aps.org/doi/10.1103/PhysRevLett.97.147202.

[37] Xue-Feng Zhang, Yin-Chen He, Sebastian Eggert, Roderich Moessner, and Frank Pollmann. Continuous easy-plane deconfined phase transition on the kagome lattice. *Phys. Rev. Lett.*, 120: 115702, Mar 2018. doi: 10.1103/PhysRevLett.120.115702. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.115702.

[38] Dariush Heidarian and Kedar Damle. Persistent supersolid phase of hard-core bosons on the triangular lattice. *Phys. Rev. Lett.*, 95:127206, Sep 2005. doi: 10.1103/PhysRevLett.95.127206. URL https://link.aps.org/doi/10.1103/PhysRevLett.95.127206.

[39] R. G. Melko, A. Paramekanti, A. A. Burkov, A. Vishwanath, D. N. Sheng, and L. Balents. Supersolid order from disorder: Hard-core bosons on the triangular lattice. *Phys. Rev. Lett.*, 95:127207, Sep 2005. doi: 10.1103/PhysRevLett.95.127207. URL https://link.aps.org/doi/10.1103/PhysRevLett.95.127207.

[40] Stefan Wessel and Matthias Troyer. Supersolid hard-core bosons on the triangular lattice. *Phys. Rev. Lett.*, 95:127205, Sep 2005. doi: 10.1103/PhysRevLett.95.127205. URL https://link.aps.org/doi/10.1103/PhysRevLett.95.127205.

[41] Massimo Boninsegni and Nikolay Prokof'ev. Supersolid phase of hard-core bosons on a triangular lattice. *Phys. Rev. Lett.*, 95:237204, Nov 2005. doi: 10.1103/PhysRevLett.95.237204. URL https://link.aps.org/doi/10.1103/PhysRevLett.95.237204.

[42] Subir Sachdev and Jinwu Ye. Gapless spin-fluid ground state in a random quantum heisenberg magnet. *Phys. Rev. Lett.*, 70:3339–3342, May 1993. doi: 10.1103/PhysRevLett.70.3339. URL https://link.aps.org/doi/10.1103/PhysRevLett.70.3339.

[43] A. Kitaev. A simple model of quantum holography. In *KITP strings seminar and Entanglement 2015 program (Feb. 12, April 7, and May 27, 2015)*.

[44] Kenny Choo, Titus Neupert, and Giuseppe Carleo. Two-dimensional frustrated $J_1 - J_2$ model studied with neural network quantum states. *Phys. Rev. B*, 100:125124, Sep 2019. doi: 10.1103/PhysRevB.100.125124. URL https://link.aps.org/doi/10.1103/PhysRevB.100.125124.

[45] H. Zimmermann, A. Minin, and V. Kusherbaeva. Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms. In *ESANN*, 2011.

[46] T. Kim and T. Adalı. Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7):1641–1666, 2003. doi: 10.1162/089976603321891846.

[47] Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, Joao Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1T2hmZAb.