

Learning Interactions between Rydberg Atoms

Olivier Simard^{1,2,*†}, Anna Dawid^{3,4,*}, Joseph Tindall³, Michel Ferrero^{1,2},
Anirvan M. Sengupta^{3,5,6} and Antoine Georges^{1,2,3,7}

¹*Collège de France, Paris Sciences & Lettres (PSL) University, 11 place Marcelin Berthelot, 75005 Paris, France*

²*Centre de physique théorique (CPHT), CNRS, École Polytechnique, Institut Polytechnique de Paris, F-91128 Palaiseau, France*

³*Center for Computational Quantum Physics, Flatiron Institute, 162 Fifth Avenue, New York, New York 10010, USA*

⁴*{aQa^L} Applied Quantum Algorithms – Leiden Institute of Advanced Computer Science & Leiden Institute of Physics, Universiteit Leiden, 2333 CA Leiden, The Netherlands*

⁵*Center for Computational Mathematics, Flatiron Institute, 162 5th Avenue, New York, New York 10010, USA*

⁶*Department of Physics and Astronomy, Rutgers University, Piscataway, New Jersey 08854, USA*

⁷*Department of Quantum Matter Physics (DQMP), Université de Genève, 24 quai Ernest Ansermet, CH-1211 Genève, Switzerland*

 (Received 7 January 2025; revised 18 April 2025; accepted 8 July 2025; published 8 August 2025)

Quantum simulators have the potential to solve quantum many-body problems that are beyond the reach of classical computers, especially when they feature long-range entanglement. To fulfill their prospects, quantum simulators must be fully controllable, allowing for precise tuning of the microscopic physical parameters that define their implementation. We consider Rydberg-atom arrays, a promising platform for quantum simulations. Experimental control of such arrays is limited by the imprecision on the optical-tweezer positions when assembling the array, hence introducing uncertainties in the simulated Hamiltonian. In this work, we introduce a scalable approach to Hamiltonian learning using graph neural networks (GNNs). We employ the density-matrix renormalization group to generate ground-state snapshots of the transverse-field Ising model realized by the array, for many realizations of the Hamiltonian parameters. Correlation functions reconstructed from these snapshots serve as input data to carry out the training. We demonstrate that our GNN model has a remarkable capacity to extrapolate beyond its training domain, regarding both the size and the shape of the system, yielding an accurate determination of the Hamiltonian parameters with a minimal set of measurements. We prove a theorem establishing a bijective correspondence between the correlation functions and the interaction parameters in the Hamiltonian, which provides a theoretical foundation for our learning algorithm. Our work could open the road to feedback control of the positions of the optical tweezers, hence providing a decisive improvement of analog quantum simulators.

DOI: [10.1103/f58h-zxs3](https://doi.org/10.1103/f58h-zxs3)

I. INTRODUCTION

Analog quantum simulators aim at answering fundamental questions about physics and solve problems that are not accessible to classical computers, such as quantum many-body problems with many degrees of freedom

and large-scale entanglement [1–7]. Recently, this promise is becoming a reality due to great experimental progress in controlling the individual quantum degrees of freedom of quantum systems with optical tweezers [8,9] and single-atom-resolved detection techniques such as quantum gas microscopy [10–13], among others. Nevertheless, experimental quantum platforms are noisy and imperfect, and this impacts quantum computing [14–16] and sensing [17,18]. To recognize and counter these imperfections, the field needs fast and scalable methods to characterize and verify prepared quantum states and Hamiltonians [19,20].

One of the frontiers of quantum characterization and verification is Hamiltonian learning, i.e., inferring the experimentally realized Hamiltonian through

*These two authors contributed equally to this work.

†Contact author: olivier.simard@polytechnique.edu

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

measurements. In realizable cases, the Hamiltonian can be directly reconstructed from knowledge of the steady states of the system [21–24], the wave function following a quantum quench [25,26] or following the application of single-qubit pulses [27]. However, the realization of such methods is not always feasible in an arbitrary quantum system due to their reliance on knowledge of the amplitudes of the quantum state.

The Hamiltonian can also be learned without full knowledge of the prepared quantum state. For instance, in Bayesian techniques, the Hamiltonian parameters are estimated by iteratively updating a probability distribution over possible Hamiltonians based on the measurement outcomes of the system [22,28,29]. Meanwhile, variational and gradient-based methods rely on optimizing a cost function based on how well the candidate Hamiltonian predicts the measurement outcomes. These methods can be combined with a trusted quantum simulator if available [30,31]. The gradient-based approach has been implemented using tensor networks [32], neural differential equations [33], and by estimating the time derivatives of some observables [34,35].

Despite these methodological advances, Hamiltonian learning is especially challenging and remains impractical in regimes where (i) comparison with classical techniques, such as tensor networks, is no longer possible [24,32,33], (ii) the Hamiltonian has many parameters and terms, some of which may be unknown [25,26,32,36], (iii) the experimental setup imposes constraints over the accessible measurement set [25–27], and (iv) the task needs to be realized fast [36] using a viable number of measurements.

Given the ongoing success of machine learning (ML) in quantum physics [37,38], especially in quantum state tomography [39–44] and phase classification [45–51], it is promising to address the outstanding challenges of Hamiltonian learning with neural networks. This direction has already been fruitful [24,52–58] and has outperformed Bayesian methods in a direct comparison [56]. While these neural-network-based methods are able to quickly predict many Hamiltonian parameters at once with low error based on a user-defined form of measurements, they require accurate numerical simulations of the quantum system under consideration and, therefore, do not scale up to large two-dimensional (2D) systems.

Here, we present a Hamiltonian-learning approach that addresses the scalability challenge of neural-network-based approaches and is driven by the needs and limitations of analog quantum simulators based on Rydberg atoms (Fig. 1). We demonstrate how a graph neural network (GNN), designed to be invariant with respect to the size of the quantum system, learns Hamiltonian parameters of a quantum transverse-field Ising model (TFIM) defined on small computationally tractable 2D arrays and successfully extrapolates to larger 2D arrays with only small impacts on the predictive accuracy of the GNN. This opens the way to tackling experimental challenges such as learning the noise sources or correcting errors in experimentally relevant regimes, with the training data obtained via numerical simulations of smaller physical systems.

In experiments, the sources of imprecision fall into two different classes. The first is of a technical nature and originates from the difficulty of making a homogeneously

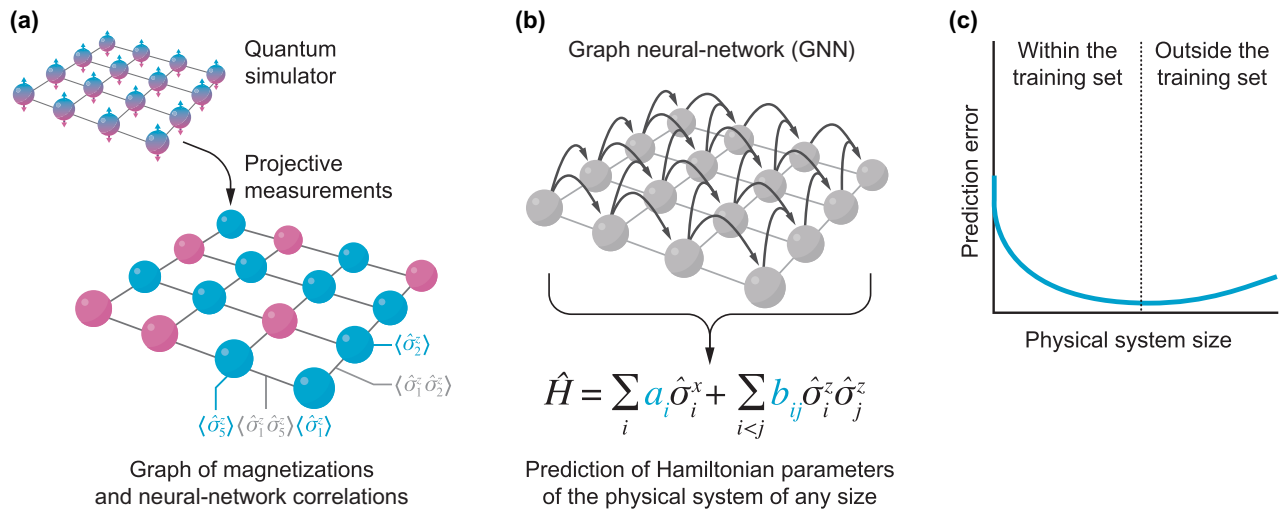


FIG. 1. Graph neural networks (GNNs) for Hamiltonian learning. (a) We represent the information about a quantum simulator in the form of a graph, where nodes contain on-site information and edges represent two-body correlators. The sketched graph includes only nearest-neighbor (NN) two-body correlators but can generally include correlators of arbitrary range and number of bodies. (b) A GNN takes in a graph of any size and learns to update its values into a useful representation, which then forms a basis for predicting an arbitrarily large number of parameters of predefined Hamiltonian terms. (c) We demonstrate that the GNN is able to predict the Hamiltonian parameters of quantum systems of large sizes while being trained only on small sizes.

spaced 2D arrangement of Rydberg atoms. The second class encompasses fundamental imprecision sources, which are the quantum zero-point motion of the atoms sitting in the finite potential well and the thermal fluctuations associated with the millikelvin temperatures. Here, we focus on the learning task that addresses the technical uncertainty leading to random optical-tweezer displacements in the system. The GNN learns to predict tweezer displacements based on observables such as the local magnetization and nearest-neighbor (NN) and next-nearest-neighbor (NNN) spin correlators, which can be obtained from projective measurements performed on quantum simulators. Due to the GNN prediction speed and accuracy, it could be used in future setups to counteract the tweezer displacements, hence providing feedback control on the experiment. This would pave the way toward the study of error-free Hamiltonians, such as those featuring quantum spin liquids [59], which would be sensitive to disorder generated by random tweezer displacements due to the narrow parameter ranges in which they take place [60–62].

In Sec. II, we describe in more detail the ultracold Rydberg platform, which inspired the learning task. There, we also introduce a bijective correspondence between the correlation functions and the interaction parameters of the Hamiltonian, which provides a theoretical foundation for our learning algorithm. This proof is laid out in Appendix A. Finally, we describe the learning algorithm itself and introduce the GNN architecture and its training procedure. In Sec. III, we demonstrate the scalability of the GNN and the dependence of its prediction error on data representation. We also show how the GNN prediction error depends on the number of snapshots used to estimate correlators. We furthermore compare the GNN performance to that of a multilayer perceptron (MLP). In Sec. IV, we conclude by discussing open problems and the next directions unlocked by our Hamiltonian-learning approach.

II. METHODS

We start this section by describing briefly, in Sec. II A, the Rydberg-atom platform for quantum simulation, which inspired our learning task, as well as the Hamiltonian studied in this work. In Sec. II B, we outline a proof, detailed in Appendix A, that demonstrates a bijective relation between spin exchange couplings and spin-spin correlation functions. This bijective relation ensures that, e.g., NN correlation functions suffice to uniquely determine NN couplings defining the Hamiltonian, and therefore the relative distances of all the NN Rydberg atoms on the lattice. Then, in Sec. II C, we introduce the key ML techniques used to predict the atomic displacements from the DMRG local magnetization and spin correlators, as well as the metrics used to judge the predictive power of the neural network.

A. Rydberg platform and the studied Hamiltonian

Among the promising quantum simulation platforms stand Rydberg-atom arrays [63–65]. By means of optical tweezers, the atoms are trapped and act like pseudo-spin- $\frac{1}{2}$ qubits, which can be addressed individually. On these platforms, the TFIM can be natively implemented, as well as other spin models, such as the quantum XY [66] or quantum XXZ models [67], by introducing anisotropy in the spin exchange energy. In Rydberg-based platforms, quantum entanglement is generated via the Rydberg-blockade mechanism, which arises from the electric dipole-dipole interaction, a second-order process that prevents the simultaneous electronic excitation of nearby atoms into Rydberg states.

In this work, we solve the Hamiltonian-learning task for the 2D TFIM

$$\hat{H} = \sum_{i < j} \underbrace{\frac{C_6}{R_{ij}^6}}_{\equiv J_{ij}} \hat{\sigma}_i^z \hat{\sigma}_j^z + \hbar \Omega \sum_i \hat{\sigma}_i^x + \hbar \delta \sum_i \hat{\sigma}_i^z, \quad (1)$$

where $i < j$ runs over all atomic positions while avoiding double counting of pairs and $C_6/\hbar \simeq 5.42 \times 10^6 \text{ rad } \mu\text{s}^{-1} \mu\text{m}^6$ is the dipole-dipole interaction term originating from the van der Waals interaction for rubidium with principal quantum number $n = 70$ [64], in which $\hbar$ the reduced Planck's constant. The parameter Ω stands for the transverse Ising field (Rabi frequency) and δ for the detuning field, while $\hat{\sigma}^z$ and $\hat{\sigma}^x$ denote the corresponding Pauli matrices. The distance R_{ij} describes the pairwise distance between any two atoms i and j sitting in the array. Hence, the interaction term [first term in Eq. (1)] covers all pairs of Rydberg atoms over the whole array and is in essence a long-range interaction that is responsible for the Rydberg blockade whenever the Rabi frequency $\hbar \Omega \ll C_6/R_{ij}^6 \equiv J_{ij}$. The two-level system is encoded in the Rydberg states, typically s orbitals. In the following, we will set $\hbar = 1$.

To emulate the TFIM on a Rydberg-atom platform, we employ the density-matrix renormalization group (DMRG) algorithm [68,69] to solve the TFIM. We provide more details on the DMRG calculations in Appendix E.

B. Bijection between interactions and spin-spin correlators

This section outlines a general theorem for Hamiltonian reconstruction, established in detail in Appendix A. This theorem states that any two-body correlation function c_{ij} that involves an arbitrary pair of lattice (orbital) degrees of freedom is uniquely (one-to-one) mapped to a corresponding Hamiltonian-interaction term J_{ij} . This assumes that the correlation matrices c_{ij} are J -representable, meaning that they correspond to correlation functions for some choice of J_{ij} , in analogy with the concept of V -representability

[70,71]. This theorem generalizes the Hohenberg-Kohn (HK) theorem from density-functional theory [72,73] and stands as a quantum generalization of the Henderson theorem [74]. In Appendix A, by means of a lemma, we specialize the proof of the bijection theorem to the TFIM, which holds if and only if $\Omega \neq 0$.

Theorem (Hohenberg-Kohn-Henderson Theorem for the TFIM). For the TFIM as defined in Eq. (1), for fixed nonzero Ω and fixed δ , there exists a bijection between the J -representable correlation functions $c_{ij} = \langle \hat{S}_i^z \hat{S}_j^z \rangle$ and the set of interactions $J_{ij}, J_{ij} \Leftrightarrow c_{ij}$.

The proof is given in Appendix A, and follows two key steps. First, we establish that, for nonzero Ω , two different interaction matrices, $J_{ij}^{(1)}$ and $J_{ij}^{(2)}$, cannot yield identical ground states Ψ , so that $\Psi^{(1)} \neq \Psi^{(2)}$ if $J_{ij}^{(1)} \neq J_{ij}^{(2)}$. This then allows us to use the variational principle for the energy as a strict inequality to prove, through a *reductio ad absurdum* argument, that different interaction matrices, $J_{ij}^{(1)}$ and $J_{ij}^{(2)}$, imply different sets of correlation matrices, $c_{ij}^{(1)} \neq c_{ij}^{(2)}$, respectively. This result serves as a theoretical foundation for reconstructing Hamiltonians from experimentally measured correlation functions in quantum systems, such as cold-atom quantum simulators.

In the proof, the key constraints come from Eq. (A6), which provides a constraint on the interaction differences

$\{J_{ij}^{(1)} - J_{ij}^{(2)}\}$ for each conventional $\hat{\sigma}^z$ -basis spin configuration $|\sigma\rangle$, for which the amplitude is nonzero. If we were to randomly pick such a $|\sigma\rangle$ configuration, we would need more than $N(N-1)/2 + 1$ such configurations to show that these differences are zero. In the context of TFIM with $\Omega \neq 0$, we have an overabundance of such constraints, namely, one for each 2^N basis vectors, due to the lemma detailed in the Appendix A. We therefore think that, beyond the TFIM, a similar theorem could be proven for systems without strong conditions, such as stoquasticity [75] or irreducibility [76] for the density matrix (properties enjoyed by the $\Omega \neq 0$ TFIM), provided that two distinct sets of interactions cannot yield the same ground state.

C. Machine-learning techniques

As previously mentioned, the physical setup of interest consists of Rydberg atoms arranged on a 2D surface and kept in place with optical tweezers. The measurement outcomes of this system, in terms of magnetization and spin-spin correlators, can be natively represented as a graph [see Fig. 2(a)]. For such a data representation, a well-suited ML architecture comes from the wide class of GNNs. The advantage that is especially important in the context of this work is that they preserve the graph structure of the processed data and can be made size invariant. In our case, we achieve size invariance by combining the GNN with properly designed *task* networks. Specifically, we employ the

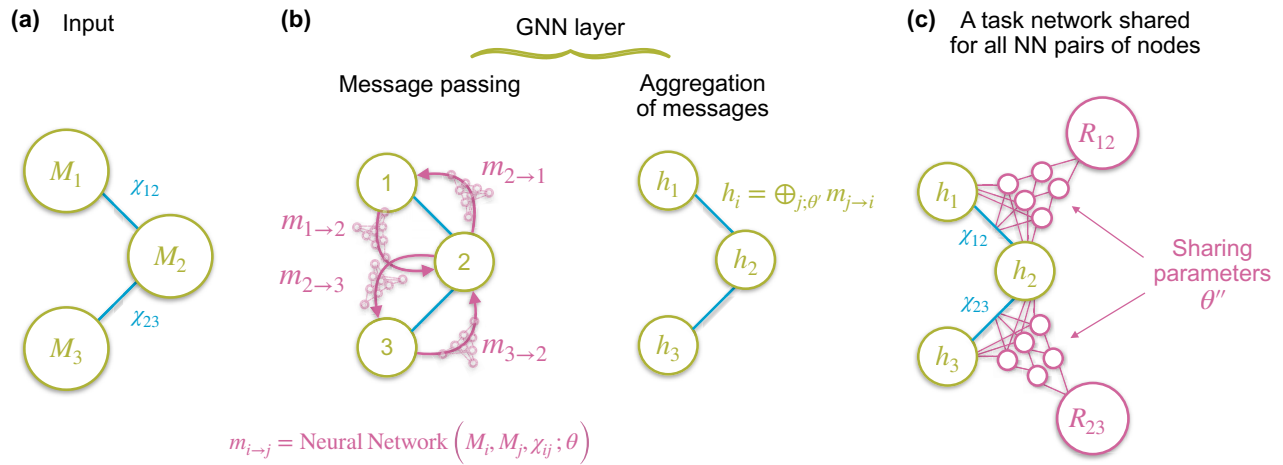


FIG. 2. An illustration of the elements of the GNN architecture on an example of a three-atom one-dimensional chain. (a) The input is a graph containing spin correlators measured in a quantum system. For example, here the nodes contain on-site magnetization, M_i , where i indicates a site and nearest-neighbor edges connecting nodes i and j contain NN spin-spin correlations, χ_{ij} . Graphs can contain higher-order many-body and longer-range correlators. (b) A GNN layer updates each node value with a new hidden value h_i , preserving the graph structure. The update is based on messages sent between the neighboring nodes, $m_{i \rightarrow j}$, which is predicted by the same message-passing network that takes in the values of neighboring nodes and edges. Then, the messages are aggregated via a learned combination of aggregators such as min, max, and mean, and form a new hidden value of each node. (c) The same task network takes in every pair of neighboring nodes along with their spin-spin correlation and predicts the distance between them. With such an architecture, the GNN is invariant to the physical size input and bases its predictions on the atom neighborhood. When comparing to the MLP model, we solely use the very same task networks to read off the NN correlators and infer the NN relative distances.

principal-neighbor-aggregation (PNA) variant with multiple aggregators [77] as a GNN layer and MLPs as task networks. We present some elements of the GNN in Fig. 2 and explain the key elements defining the GNN and how it processes the physical data in Sec. II C 1. To train the GNN, the L^2 norm between targets $\mathbf{y}$ and predictions $\hat{\mathbf{y}}$ is evaluated, where the targets and predictions correspond to relative NN distances ($R_{ij} \rightarrow R_{(ij)}$) and relative NNN distances ($R_{ij} \rightarrow R_{\langle ij \rangle}$) between atoms

$$\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2, \quad (2)$$

where $\|\cdot\|$ denotes the L^2 modulus of the vector. Thus, Eq. (2) is the cost function to be minimized via gradient-descent techniques. The optimizer used during training is AdamW [78].

1. Graph neural network

A GNN is a neural-network architecture that processes data represented as a graph [79]. This architecture is especially attractive for tackling physical problems, as the GNN processes the data in a physics-inspired way, emulating interactions between nodes in a graph. Moreover, the GNN layer is invariant to the input size and has more inherent interpretability than other types of neural network [80]. It has been widely used in classical physics [81–83] and is slowly entering quantum physics as well [84].

The input of the GNN layer is a graph and its output is a processed graph of the same structure but with modified node (and possibly edge) values. In Fig. 2(a), we show an exemplary input graph that is relevant to this work and that describes the outcomes of measurements on a three-spin chain, as discussed in Secs. II C 2 and II C 3. The GNN layer processes the input graph by sending learnable messages $m_{i \rightarrow j}$ between all pairs of directly connected nodes i and j , as shown in Fig. 2(b). All messages within one GNN layer are outputs of the same neural network, whose parameters Θ are optimized to send messages that are relevant to the task at hand. The input of this message-passing neural network is the values of the node pair between which the message is sent (here, M_i and M_j) and the value of the edge connecting the node pair (here, χ_{ij}). The same neural network with the same θ can send different messages between various pairs of nodes because its input (M_i , M_j , and χ_{ij}) varies between them. In our case, the message-passing network is a fully connected neural network. The architectural details are in Appendix D.

After the messages are sent, they are aggregated in each node to form a new updated hidden value of a node, $h_i = \oplus_{j, \theta'} m_{j \rightarrow i}$. In this work, we employ PNA [77] as the GNN layer architecture. Instead of using one aggregation operation, PNA uses a learnable linear combination $\oplus_{\theta'}$ of multiple aggregators such as summation, multiplication, max, min, and average. Note that it is at the aggregation

stage that edge effects in the system can be taken into account. The described GNN layer, composed of message passing and aggregation, can be repeated several times in a network architecture, effectively allowing for “communication” between nodes that are further apart from each other. In this work, we use four consecutive GNN layers, each composed of message passing and aggregation of messages.

We design the full neural-network architecture to be independent of the size of the physical system. While the GNN layer natively has this property, ultimately, we also need a *task* network that makes predictions based on the hidden representation of the graph learned by the GNN layer(s). Therefore, we combine the GNN layers with a neural network parametrized by θ'' that takes as an input hidden values of only two NN nodes, h_i and h_j , along with their spin-spin correlation function χ_{ij} and predicts the distance between NN spins i and j , $R_{(ij)}$ (or $R_{\langle ij \rangle}$ in case of NNN neighbor spins), as presented in Fig. 2(c). The input size independence is achieved here in a similar way as in the GNN layer. One task network learns a mapping between an NN pair and a distance between them and then makes predictions for all possible NN pairs. It can give different predictions for various NN pairs because the hidden values of participating nodes serving as the neural-network input are different. In our case, the task network is a fully connected neural network with two hidden layers. More generally, task networks can be adjusted to the form of Hamiltonian terms the parameters of which they need to learn. For one-body Hamiltonian terms, the task network can take one h_i as an input. For n -body terms, the input can include hidden values of n nodes.

Therefore, the training of the full neural network aims at finding such parameters $\Theta \in \{\theta, \theta', \theta''\}$ that allow the sending of useful messages between the graph nodes, an optimal message aggregation that takes into account edge effects in the system, and, finally, a correct prediction of the distances between NN spins, based on the developed hidden representation of the graph.

In Sec. III C 2, we compare the GNN performance to that of an MLP. When doing so, we employ the same task-network architecture but without the GNN processing of the graph—each task network directly takes the NN spin-spin correlation function of an atom pair as input and predicts the corresponding NN distance as output.

2. Physical inputs

The input layer to the GNN is fed with samples built in the following way. On the one hand, the node features consist of local-magnetization expectations $M_i = \langle \hat{\sigma}_i^z \rangle_{0; \vec{\Omega}}$, where $\hat{\sigma}_i^z$ is the Pauli diagonal operator at site i (in the Z basis), for a sequence of transverse field $\vec{\Omega}$ values [85]. On the other hand, the edge features consist of NN spin-spin correlation functions (χ^{NN}) and NNN spin-spin correlation

functions (χ^{NNN}), defined as

$$\chi_{ij;\vec{\Omega}}^{z(x)} = \langle \hat{\sigma}_i^{z(x)} \hat{\sigma}_j^{z(x)} \rangle_{0;\vec{\Omega}} \propto c_{ij;\vec{\Omega}}, \quad (3)$$

where i and j are NN and NNN connections between Rydberg atoms, respectively, and $\langle \cdot \rangle_0$ means that the expectation value is carried out in the ground state of the system. The vector notation $\vec{\Omega}$ used as subscript means that samples are arrays of such correlation functions across different equally spaced values of the transverse field (we assume that the field Ω is spatially homogeneous, i.e., the same for all atomic sites).

3. Schemes to estimate correlation functions

All the physical observables considered, namely, the magnetization and spin-spin correlation functions, are computed using DMRG [68,69]. There are two distinct ways to evaluate observables from a converged matrix-product-state (MPS) wave function obtained via DMRG: exactly via direct contraction of the tensor network $\langle \psi | O | \psi \rangle$ or with sampling by taking N_{sample} bit-string samples from the MPS distribution $p(x) \sim |\psi(x)|^2$, which can be done efficiently and perfectly [86,87]. We refer to the data sets obtained with these two methods as *exact* or *snapshot* data sets, respectively. The former exact data sets are used unless specified otherwise.

When computing the snapshot observables, one can sample the MPS expressed in the Z or X basis and then measure in the Z basis to compute χ^z or χ^x , respectively. Indeed, from Eq. (3),

$$\chi_{ij;\vec{\Omega}}^x = \sum_{\sigma} \left| \langle \sigma | \hat{H}_{\text{Hd}} | \Psi \rangle \right|^2 \langle \sigma | \hat{\sigma}_i^z \hat{\sigma}_j^z | \sigma \rangle, \quad (4)$$

where $\hat{H}_{\text{Hd}}$ is the involutive Hadamard operator defined in the Z basis,

$$\hat{H}_{\text{Hd}} \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Again, in Eq. (4), the set $\{|\sigma\rangle\}$ denotes the eigenstates of the $\hat{\sigma}_k^z$'s.

In the snapshot scenario in which we sample the MPS to compute observables, we resort to Eq. (4) to compute χ^x . We combine the Z observable (χ^z) and X observable (χ^x) in the GNN training on some explicit occasions (the specific training scenarios are covered in Sec. II C 4). Moreover, by default, if not explicitly stated otherwise, the inputted observables are Z observables. The technical details related to the training and testing of the GNN are discussed further in Appendix D.

4. Training scenarios

We have devised several training scenarios in this work to compare different types of input data and architectures

TABLE I. The table of training scenarios: a summary of the training cases used in this work. If not explicitly mentioned in a particular scenario, all correlation functions are measured in the Z basis. As a reminder, M represents the local magnetization.

Case 1	M and Ω history
Case 2	M , χ^{NN} , and Ω history
Case 3	M , χ^{NN} , χ^{NNN} , and Ω history
Case 4	M , χ^{NN} , NNN edges, and Ω history
Case 5	χ^{NN} , NNN edges, and Ω history
Case 6	M , χ^{NN} , χ^{NNN} , and Ω history ($Z + X$)

and select the best learning procedure. These training scenarios are listed in Table I and described in the following, in order:

- (i) To start with, the local magnetization $\langle \hat{\sigma}_i^z \rangle_{0;\vec{\Omega}}$, piled up across $\vec{\Omega}$, is inputted to the GNN without providing any physical information to the edges of the graph that stores the information about the cold-atom array. Only the disorderless (ordered) edge distances are provided as edge features.
- (ii) Then, in addition to the local magnetization, the NN spin correlation functions [Eq. (3)] stacked up along $\vec{\Omega}$ are inputted to the GNN input layer to dress up the NN graph edges. This scenario should substantially improve the performance of the neural network according to the bijective relation between connected spin correlators c_{ij} and spin couplings J_{ij} (for the proof, see Sec. A).
- (iii) Next, we add the NNN spin-correlation functions χ^{NNN} to the input data, thereby supplementing the NNN graph edges with physical information. In order to evaluate and discriminate the physical benefits from the architectural ones, we also consider scenarios in which the features of NNN graph edges are set to unity to help the flow of information during optimization. In this way, the NNN edges do not carry any information about χ^{NNN} .
- (iv) Hence, the features of the NNN graph edges are set to unity across $\vec{\Omega}$ to check out the effect of including NNN edges in the GNN architecture without assigning them any physical information (no values of χ^{NNN}). The NNN edges *per se* could act as “skip connections” in the back-propagation of the gradients.
- (v) To continue the comparisons, in light of the bijection relation covered in Sec. A, we remove the physical content from the NNN graph edges (no values of χ^{NNN}) and graph nodes (no values of local magnetization M) and set the features to unity across $\vec{\Omega}$.
- (vi) To finish the comparisons, we combine the Z - and X -observable measurements of the NN and NNN

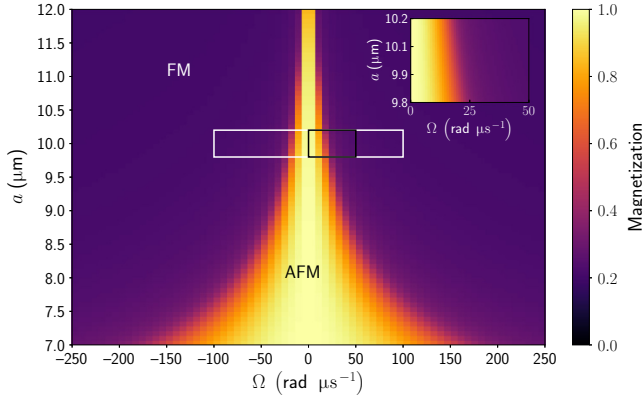


FIG. 3. The phase diagram of the Rydberg TFIM for a 5×5 square lattice. The white box encloses the parameter sets that can possibly be drawn during the training of the GNN. Along the y axis, a represents the NN distance between atoms. The inset is an enlargement of the region of the phase diagram enclosed within the black box. Note that the phase diagram has been calculated for an equally spaced ordered (disorderless) lattice.

correlators [Eq. (3)] in the training data set. When computing X observables by sampling the MPS, we resort to Eq. (4). The magnetization along z is also considered in the training and an Ω history is used. All the training scenarios hitherto mentioned are summarized in order in Table I and they all use the same Ω history (see further down below).

To generate the graph data set fed into the GNN, we choose a nominal NN distance between atomic sites $\bar{R}_{(ij)}$ of $10 \mu\text{m}$, corresponding to the array spacing a [5]—the notations a and $\bar{R}_{(ij)}$ will be used interchangeably to denote the nominal NN distance. The positions of the Rydberg atoms are slightly shifted about their nominal values, with perturbations comprised within the bracket $\Delta S \in [-0.1, 0.1] \mu\text{m}$, corresponding to the width of the white box shown in the phase diagram in Fig. 3. ΔS is drawn from a uniform distribution. Hence, the NN relative atomic distance $R_{(ij)} = \bar{R}_{(ij)} + \Delta R_{(ij)}$, where the perturbation in the NN relative distance between atoms lies within $\Delta R_{(ij)} \in [-0.2, 0.2] \mu\text{m}$. The white box in Fig. 3 spans the allowed values of the transverse field Ω selected during the training phase.

Component-wise, the GNN edge-feature space is made of copies of the exchange interactions J_{ij} at different equally spaced values of Ω . The explicit array of ten Ω 's, used by all the scenarios laid out in Table I, expands as

$$\vec{\Omega} = \left[-\frac{900}{9}, -\frac{700}{9}, \dots, \frac{700}{9}, \frac{900}{9} \right] \text{ rad } \mu\text{s}^{-1}$$

and is enclosed in the white box of Fig. 3. In the definition of $\vec{\Omega}$, an arbitrary increment of $200/9 \text{ rad } \mu\text{s}^{-1}$ separates

each consecutive element, so as to span both phases of the phase diagram displayed in Fig. 3.

5. Metrics

While investigating the scenarios listed above in Sec. II C 4, Table I, we use three main metrics to evaluate performance. Throughout this work, the figure layout orders the metrics from top to bottom. Hence, we first plot in the figures the coefficient of determination R^2 , defined as

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}, \quad (5)$$

which qualifies the predictive power of the statistical ML process. In Eq. (5), $\bar{y}_i$ stands for the mean of the images y_i of a function to be modeled and $\hat{y}_i$ represents the prediction of the model, associated with y_i . When R^2 approaches 1 from below, the model is deemed to effectively capture the key characteristics of an unknown underlying function. Next, we consider the mean absolute error (MAE), defined as

$$\text{MAE} = \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{N}, \quad (6)$$

whose dimension is in nanometers. In Eq. (6), the sample size is denoted by N and it grows larger as the cluster size increases, since the number of graph edges and nodes increases as well. Recall that the extrema of the possible variations (perturbations) of the NN relative distance is $\Delta R = \pm 0.2 \mu\text{m}$. Then, at last, the median of the absolute error, denoted MEDAE for short, is plotted. The combined estimation of the MEDAE and MAE allows us to evaluate the symmetry of the learned probability distribution: if it is skewed, the MEDAE and MAE differ significantly, whereas if it is symmetric, both the MAE and MEDAE look very similar.

III. RESULTS

As pointed out in Sec. II C 3, there are two distinct ways to estimate the physical observables from the DMRG ground state. One way is to work out the expectation values having access to the full wave-function content, thereby generating exact training data sets, while the other is to “sample” the wave function, thereby producing measurement snapshots constituting snapshot data sets. The former is useful to test thoroughly and faithfully the limitations and strengths of the GNN architecture employed, leaving out extra sources of statistical noises that would come from the projective measurements—only the numerical round-off error would be the source of “noise”. Therefore, the extent to which the bijection relation proven in Appendix A is respected by the PNA GNN can be

assessed. The performance of the GNN training using this exact scheme is presented in Sec. III B. The GNN training scheme utilizing snapshot observables, in closer relation to experiments, is presented in Sec. III C. These two ways of measuring observables should coincide perfectly with each other in the limit where the number of snapshots tends to infinity. Before delving into GNN Hamiltonian-learning results of Secs. III B and III C, we ought to study more closely the phase diagram of the TFIM at the constant zero-detuning field [$\delta = 0$ in Eq. (1)]. This is done in Sec. III A.

A. Phase diagram

In Fig. 3, we illustrate the phase diagram of the TFIM for a window of nominal NN distances $a = [7, 12] \mu\text{m}$. Two distinct magnetic orders feature on the phase diagram, namely, the antiferromagnetic phase (AFM) and the ferromagnetic phase (FM). The color plot displays the degree to which spins are ordered successively in an antiparallel fashion, namely, the magnetization M :

$$M = \frac{1}{N} \sum_i (-1)^{x_i + y_i} \langle \hat{\sigma}_i^z \rangle_0, \quad (7)$$

where N represents the total number of array sites, while x_i and y_i are integers accounting for the Cartesian components of the orthonormal Bravais lattice vectors of a square lattice. The white box appearing on Fig. 3 encloses the parameter region of the Hamiltonian given in Eq. (1), from which are drawn the physical parameters, mapped to graphs, that will constitute the training data set of the GNN. The region has been selected to span over domains of distances and transverse fields representative of the experimental setups. To give ideas of orders of magnitude, considering that $\bar{R}_{(ij)} \simeq 10 \mu\text{m}$, the interaction coefficient showing up in Eq. (1) is $C_6/\bar{R}_{(ij)}^6 \sim 10 \text{ rad } \mu\text{s}^{-1}$, hinting at the whereabouts of the phase transition. By defining Ω_c as the critical Rabi frequency above which the magnetization given in Eq. (7) falls below 10^{-1} , for $\bar{R}_{(ij)} \simeq 10 \mu\text{m}$, we obtain $\Omega_c \simeq 17 \text{ rad } \mu\text{s}^{-1}$ (the phase diagram is symmetric about $\Omega = 0$).

We then enlarge the plot in the vicinity of the phase transition in Fig. 3 (inset) to obtain a better view of the variation of the magnetization. The range of nominal NN distances a corresponds to the height of the white box appearing in Fig. 3. Note that its height encompasses all the possible a 's given that $\Delta R_{(ij)} \in [-0.2, 0.2] \mu\text{m}$. In the region with a significant color gradient, the variance of the NN spin correlation functions is highest, as will be seen later in Fig. 6. $\Omega \gtrsim 10 \text{ rad } \mu\text{s}^{-1}$ corresponds to the point at which the Rabi frequency and interaction terms are of the same order of magnitude and, thus, there is more variability in the ground state.

B. Predictions from the exact wave function

In Sec. III B 1, we show the scalability power of the GNN in the Hamiltonian-learning task. We also resort to the metrics introduced in Sec. II C 5 to compare against each other the various GNN training scenarios laid out in Table I. In particular, we dress up the graph inputs to the GNN with exact physical correlation functions estimated from the MPS, encoding the full ground-state distribution function via its amplitudes; thus, throughout this section, the GNN is trained over exact training data sets. Furthermore, in Sec. III B 2, we study how the choice of the Ω impacts the predictive and extrapolation power of the GNN. Finally, in Sec. III B 3, we summarize the results of additional studies from Appendixes B–C.

1. GNN scalability and comparison between training scenarios

We illustrate in Fig. 4 the indicative metrics for all training scenarios covered in Sec. II C (see Table I), following through the figure layout described in Sec. II C 5. The error bars show the standard error on the GNN metric results, based on a set of five independently trained GNNs within each case scenario. The main takeaway is that in every scenario, the GNN exhibits an impressive extrapolation ability to larger physical system sizes than the ones included in the training data set. In particular, here we have trained the GNN on data coming from physical system sizes of $\{4 \times 4, 5 \times 5, 6 \times 6\}$ and the GNN has successfully extrapolated to sizes up to 9×9 (which constitutes a numerical limit for our DMRG calculations) with only a small impact on its predictive error. Its successful extrapolation to rectangular arrays is also notable.

From looking at Fig. 4, it seems that the GNN does not benefit from physical information provided by the various correlation functions, namely, the NNN spin correlation function and local magnetization, as much as from the architectural add-ons such as the NNN edges; this is even more obvious when the local magnetization is set to 1 (case 5 in Table I). Nonetheless, including the physical information coming from χ^{NNN} is not without benefits, as can be seen from comparing case 3 (extra χ^{NNN}) with case 2, since the former outperforms the latter as far as the MAE [see Eq. (6)] and the MEDAE are concerned. By far, the worst GNN training implementation is case 1, where absolutely no physical-edge information—neither an NN nor an NNN correlation function—is included in the graphs of the data set. When adding the X correlators [Eq. (4)] as done in case 6, the predictive power of the model can be increased further, amassing more physical input through the training, at the cost of a higher standard error on the extrapolation of large cluster sizes.

Moreover, it looks as though the additional physical information to χ^{NN} is detrimental to the predictions of the GNN for the cluster sizes part of the training data

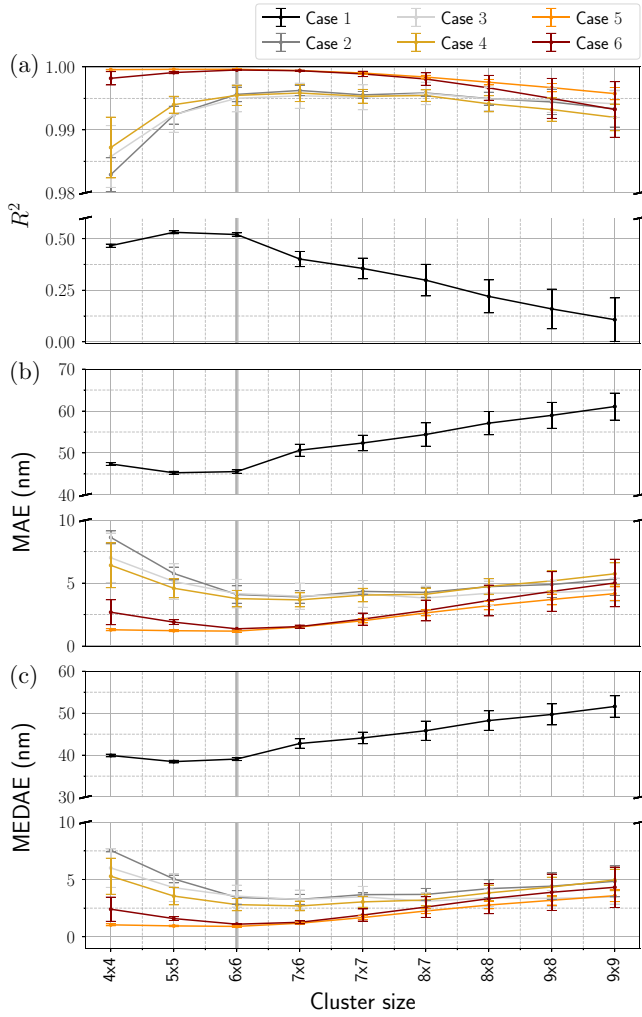


FIG. 4. The performance of the GNN training scenarios. (a) R^2 for various cluster sizes in the test data set. (b) The mean absolute error of the predictions. (c) The median of the absolute error of the predictions versus the targets. The vertical gray line indicates the cluster size beyond which the GNN extrapolates from its training data set. Note that all cases use the same Ω histories $\bar{\Omega}$. The cases are detailed in Table I. The error bars show the standard error on five distinct and independent trainings.

set (before the gray vertical line); this can be seen in all three measures shown in Fig. 4. This extra information also seems to negatively impact the error for large cluster sizes: the smallest error is obtained using case 5, while the largest is obtained using case 6. The GNN is also more faithful to the bijection relation described in Sec. A, when no physical observables are supplemented to NNN graph edges or nodes. Indeed, compared to case 5, both case 4, corresponding to the addition of local magnetizations to nodes, and case 3, corresponding to the additions of local magnetizations to nodes and χ^{NNN} to NNN graph edges, diminish the performance.

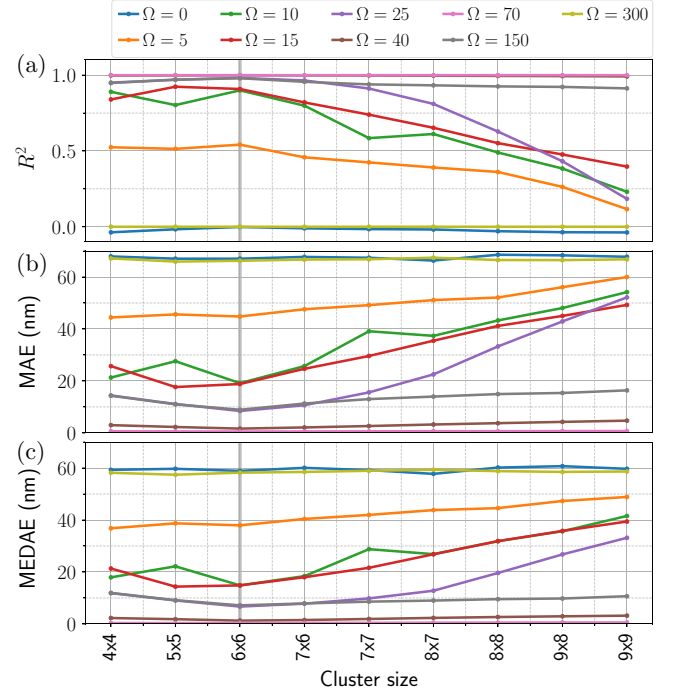


FIG. 5. The Ω -resolved performance of training case 3. (a) R^2 as a function of the Rydberg cluster sizes at various values of Ω (see legend). (b) The mean absolute error of the predicted NN relative displacements. (c) The median of the mean absolute error of the relative NN displacements. Ω is shown in units of $\text{rad } \mu\text{s}^{-1}$.

2. Effect of $\bar{\Omega}$ histories

In the training scenarios covered in Fig. 4, we have only considered cases in which histories of Ω are used to enrich both the edge and node features in the graphs. Of course, one can wonder what happens if one selects solely one value of Ω and if some Ω 's have higher information content for the Hamiltonian-learning task than the others. For that matter, we show in Fig. 5 a collection of single Ω value used to train the GNN following case 3, in that the physical observables such as M_i , χ^{NN} , and χ^{NNN} are used as node and edge attributes, but the data set is limited to only one Ω instead of an Ω history. From looking at Fig. 5, it is striking to see how the training performance hinges on the selected value of Ω . We can nevertheless state, with confidence, that the values of Ω chosen above, $\Omega \simeq C_6/\bar{R}_{(ij)}^6 \sim 10 \text{ rad } \mu\text{s}^{-1}$, although not too far away, give significantly better results. Furthermore, above that threshold, in the inset of Fig. 3, for a fixed value of Ω , the magnetization M given in Eq. (7) varies more within the same set of a 's.

To better understand why some values of Ω perform better than others in Fig. 5, we plot in Fig. 6 the standard deviation (STD) of χ^{NN} , denoted $\text{STD}[\chi^{\text{NN}}]$, as a function of the cluster sizes for all Ω 's considered. In the range $\Omega \simeq [10, 25] \text{ rad } \mu\text{s}^{-1}$, fluctuations in the correlation function

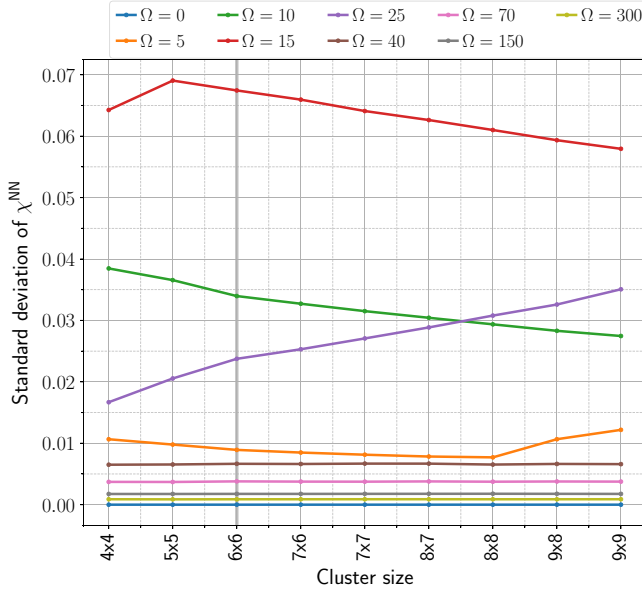


FIG. 6. The standard deviation of the χ^{NN} as a function of the cluster size for a set of transverse fields Ω : (a) R^2 ; (b) the mean absolute error; (c) the median of the absolute error. The gray vertical bar pins down the size above which the GNN extrapolates. The largest variances, across all cluster sizes, are obtained when $J_{(ij)} \approx \Omega$. Ω is shown in units of $\text{rad } \mu\text{s}^{-1}$.

surpass those of all other Ω 's by an order of magnitude at least, signaling the magnetic phase transition in the vicinity. Moreover, in that regime, the trend of $\text{STD}[\chi^{\text{NN}}]$ is far from being constant with the cluster size. Indeed, spin interactions in the critical regime become more long range, which renders the training more size dependent and greatly reduces the extrapolation performance at larger cluster sizes—the physics is different between cluster sizes. Even for $\Omega = 25 \text{ rad } \mu\text{s}^{-1}$, the fluctuations increase for all values of the cluster sizes, in line with the critical behavior of correlation functions when passing a quantum phase critical point. When the quantum critical point is passed but the standard variation of χ^{NN} remains roughly within $1 \times 10^{-2} \gtrsim \text{STD}[\chi^{\text{NN}}] \gtrsim 1 \times 10^{-3}$, this is where we see an optimal gain in performance. Hence, when $\Omega > J_{(ij)}$, but not by too much, one finds a sweet spot. However, one could generally take an Ω history for $\Omega > J_{(ij)}$. The values of $\Omega = \{0, 5, 300\} \text{ rad } \mu\text{s}^{-1}$ all feature a $\text{STD}[\chi^{\text{NN}}]$ that falls below 1×10^{-3} , too small for the GNN to pick up useful signals.

3. Additional studies

We also study the GNN extrapolation dependence on the physical system sizes included in the training data set. In Appendix B, we show that, unsurprisingly, the GNN trained on a training data set including a larger variety of system sizes performs and extrapolates better than when trained on smaller physical systems. Surprisingly, the

smaller training data sets diminish the GNN performance the strongest in the regime of small physical systems, indicating the difficulty of learning the edge effects of physical systems. Moreover, in Appendix C, we show that it is also possible to predict NNN distances between atoms, but it requires larger physical systems included in the training set to keep the same GNN performance. This hints at the existence of a critical physical system size that needs to be included in the data set for the GNN to extrapolate properly and at its dependence on the locality of the Hamiltonian.

C. Predictions from snapshots

In this section, we ask how the GNN performs when the input correlation functions are estimated from a collection of successive projective measurements of the wave function, i.e., trained from snapshot training data sets instead of exact training data sets. Note that all the assertions made in the previous sections carry over to this training since the GNN does not change; nor does the physics. Here, we still train the GNN with a training data set whose samples are made up of correlation functions (local and nonlocal) estimated from a number of snapshots (projective measurements of the wave function), nicknamed snapshot data. One could naively think that predicting the NN relative distances of snapshot data with the GNNs trained with exact correlators (directly measured in the ground state) would lead to a better outcome as opposed to the case in which the GNNs are trained with snapshot data. We have tried predicting the relative distances from snapshot data using a GNN model trained with exact correlators but it produced worse results compared to a GNN model trained on the snapshot data (not shown). In fact, when measuring directly from the wave function, no errors coming from the finite number of projective measurements spoil the training data set and the GNN can therefore dedicate all its variational parameters to “learning” the bijection presented in Sec. A. In Sec. III C 1, we show how the GNN performance depends on the number of snapshots used to estimate the correlation functions. In Sec. III C 2, we compare the performance of the GNN and the MLP on the snapshot data to determine the advantage of the graph pre-processing. Finally, in Sec. III C 3, we study whether the GNN error on snapshot data scales according to the central limit theorem.

1. Comparison between best training scenarios

The two best cases computed using Z-observable measurements, presented in Sec. III, are compared against each other in Fig. 7, namely, cases 3 and 5 (see Table I). Furthermore, since when dealing with snapshots to estimate correlation functions as in experiments one should be able to carry out measurements in the X basis as well [see Eq. (4)], we check the GNN performance when combining Z and X correlators in Fig. 8. The colors used in Figs. 7

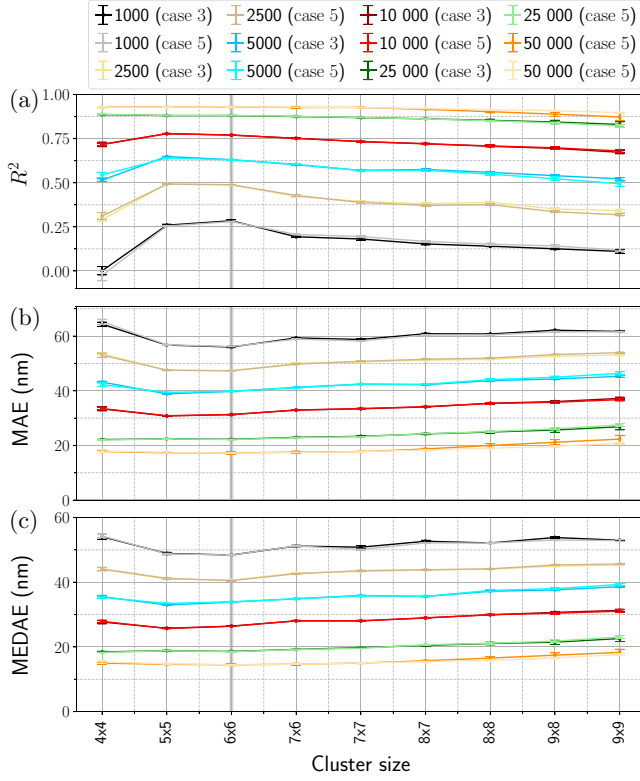


FIG. 7. The performance of training cases 3 and 5 using snapshot data. A view of the metrics considered so far as a function of the cluster size: (a) R^2 ; (b) the mean absolute error; (c) the median of the absolute error. The two best cases—namely, cases 3 and 5—have been retained. The data set has been generated using a finite number of snapshots to estimate the observables, namely, $N_{\text{smp}} = 1000, 2500, 5000, 10\,000, 25\,000$, and $50\,000$ snapshots. The error bars show the standard error on five distinct and independent trainings. The vertical gray lines indicate the point below which cluster sizes are considered in the snapshot training data set.

and 8 have been chosen carefully to match the snapshot sample size.

Looking first at Fig. 7, overall, one can note that case 5 slightly underperforms compared with case 3, especially for smaller N_{smp} 's. The architectural benefits seem to be washed out when training with snapshot data, as is clearly apparent when looking back at case 5 in Fig. 4. Evidently, increasing the number of snapshots is very beneficial to the training. When comparing Fig. 4 to Fig. 7, having in mind the bijection proof detailed in Appendix A, it seems obvious that the almost perfect rendering of the GNN (see cases 3, 5, and 6 in Fig. 4) is unlocked due to the exact and precise correlators provided, and that the main limitation seen in Figs. 7 and 8 relates to the statistically diluted physical content in snapshots.

To remedy the poorer information content in a snapshot data set, we show in Fig. 8 how the GNN performs if we include both X - and Z -observable measurements. Just as

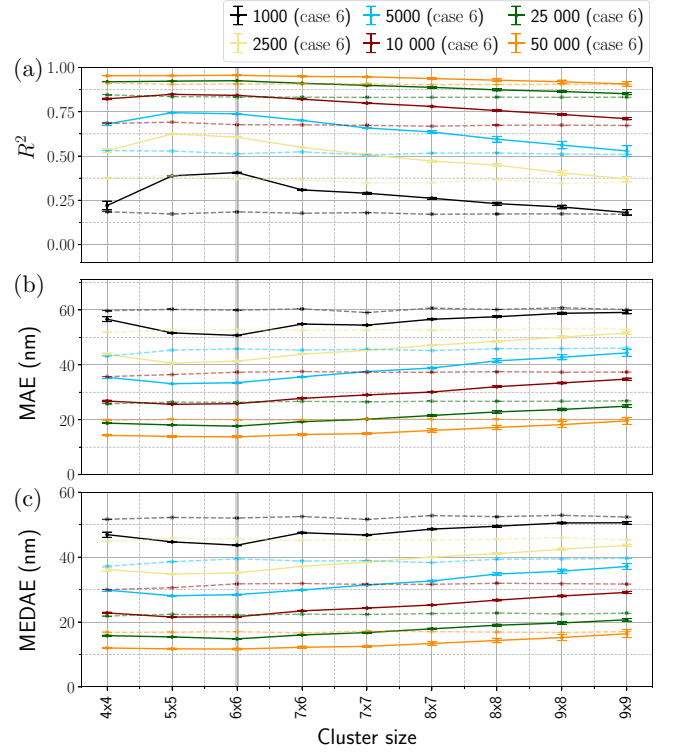


FIG. 8. The performance of training case 6 using snapshot data. A view of the metrics considered so far as a function of the cluster size. Case 6 is considered. The data set has been generated using the same finite number of snapshots as in Fig. 7. The error bars show the standard error on five distinct and independent trainings. Again, the gray vertical line indicates the cluster-size threshold above which neural-network models extrapolate. The dotted curves with corresponding colors are predictions from an MLP (including error bars).

observed in Fig. 4, we get a substantial improvement on the metrics considered, especially the MAE and the MEDAE, when the χ^{NN} and χ^{NNN} correlators are computed with the Z and X Pauli strings. When doing so, the MAE and the MEDAE hover around a discrepancy of 10% on snapshot data utilizing $10\,000$ projective measurements. However, the combination of the Z and X observables turns out not to be as beneficial as the doubling of N_{smp} .

2. Comparison between the GNN and the MLP performance

In Fig. 8, we compare the predictive errors and scaling of the GNN (full lines) and a simple MLP of the same architecture as the task network of the GNN (dashed lines). The same MLP is trained to predict an NN distance between a pair of atoms from the NN spin-spin correlation for all the atoms, disorder realizations, and cluster sizes included in the training set. The resulting MLP has a low predictive error, which is likely due to the bijection proved in Appendix A. Moreover, the MLP performance

is almost constant across the cluster sizes and is comparable to the performance of the GNN at the largest cluster sizes.

At the same time, the GNN outperforms the MLP over the range of cluster sizes considered in this study. This gain shows the advantage of graph preprocessing, which can lead to learning the edge effects and more accurate predictions for cluster sizes, where edge effects play a role.

3. Snapshot-size effect

To check if the GNN performance on snapshot data correlates with the data-set size according to the central limit theorem, in Fig. 9 we plot the standard deviation of the difference between the network predictions and targets for various snapshot sample sizes (N_{smp1}) for all cluster sizes (solid curves). Additionally, we use the snapshot data with $N_{\text{smp1}} = \{5000, 10\,000, 25\,000, 50\,000\}$ to fit the power-law scaling $\propto 1/\sqrt{N_{\text{smp1}}}$ of the central limit theorem (dotted curves with corresponding colors). Since the validity of the central limit theorem holds for large sample sizes, the profile of the standard deviation of the snapshot data with $N_{\text{smp1}} = \{1000, 2500\}$ deviates from the central limit theorem. Without much surprise, the standard error between the predictions and truths diminishes with the

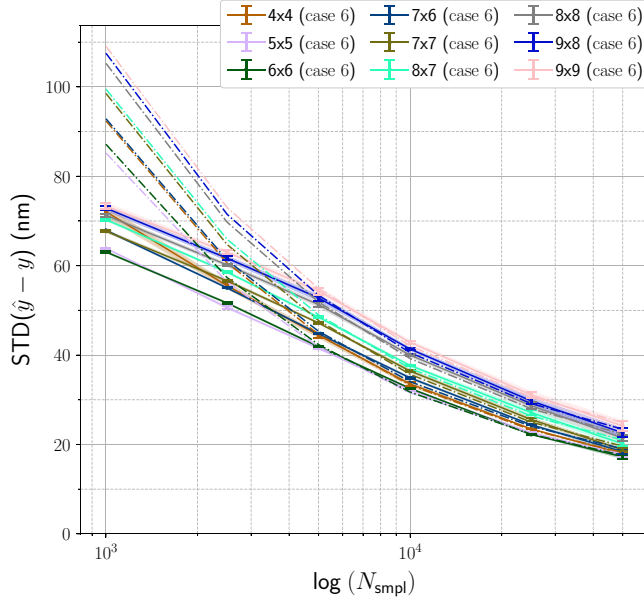


FIG. 9. The effect of the snapshot sample size N_{smp1} . The standard deviations (STDs) of the difference between NN relative distance predictions and targets for case 6 using snapshots (solid curves) for all cluster sizes considered. The corresponding dotted curves, the color of which matches that of the solid curve, show how the STD would evolve with the sample size according to the central limit theorem, taking the snapshot data with $N_{\text{smp1}} = \{5000, 10\,000, 25\,000, 50\,000\}$ as reference. The error bars show the standard error on five distinct and independent trainings of the PNA GNN.

number of snapshots N_{smp1} used to calculate the observables. For smaller values of N_{smp1} , the discrepancy between the solid and dotted curves signals that the GNN is trying to learn something from the statistical uncertainty arising from snapshot data, leading to noise overfitting.

Estimating large covariance matrices from finite samples is an area that has received a lot of attention in the past few decades [88]. Truncating the eigenvalue decomposition of the empirical covariance matrix (connected correlators), based on random matrix theory, is one obvious procedure that has been in use in applications from image denoising to portfolio construction [89–91]. More recently, further progress has been made on shrinkage-based methods [92]. Furthermore, one could draw snapshots from an MPS state in random computational bases, i.e., randomly applying a unitary rotation on each site before doing a measurement in the Z basis, since this can help the estimation of observables to convergence [86]. In the future, we will investigate whether these methods improve the performance of our network on snapshot data.

IV. DISCUSSION AND CONCLUSIONS

In this work, we have made use of the PNA algorithm to predict the positions of Rydberg atomic arrays in 2D cold-atom arrays. The training data set has been produced by simulating the ground state of a TFIM using the DMRG algorithm. Overall, the GNN employed has demonstrated a clear capacity to extrapolate outside its training domain—and also the MLP considered at the end of the work—be it in terms of system sizes or shape. The GNN and MLP scalability is probably enabled, from the side of the quantum system, by the effective locality of the studied Hamiltonian [93] (the decay of the spin coupling as $1/R_{ij}^6$). The training input data have been the local magnetization and the NN and NNN correlation functions, and the targets have been the NN and NNN relative distances between atoms in the array, depending on the training-case scenario (see Table I). To narrow down the best training conditions for the GNN, various training procedures have been followed, each with its own efficiency. The correlators have all been computed in the Z measurement basis and in some cases Z and X correlators have been used.

Depending on the number of experimental snapshots that can be generated, if one does not have knowledge of the best transverse field Ω with which to train the GNN, it turns out that generating samples with the same array disorder across different values of Ω helps tremendously. The values of Ω making up this array should be selected roughly above, but not too far from $\Omega \simeq C_6/\bar{R}_{(ij)}^6$, where C_6 represents the van der Waals interaction and $\bar{R}_{(ij)}$ represents the nominal NN distance between atoms. Alternatively, active learning could be employed to identify optimal Ω 's to train the GNN [94].

The GNN performance depends on the physical data that it utilized during the training. In high-quality numerical and/or experimental setups, only the NN spin correlation function, measured in one single measurement basis, should suffice to predict, in a one-to-one mapping, the NN relative distances between the Rydberg atoms. This is what is seen when exact correlators are used (directly computed from the full wave function). However, when using snapshot data, uncertainties occur and significantly reduce the performance of the GNN. To reduce this unwanted spoiling of the data, a combination of both Z and X correlators is used. In doing so, to get average predictions over the NN relative distances well below 10% discrepancy, more than 10 000 projective measurements need to be carried out to estimate the spin-spin correlators. However, increasing the number of snapshots is essential to improve the performance. We have also shown, by comparing the GNN performance to that of the MLP, that the graph preprocessing done by the GNN layer allows the inclusion of edge effects and provides more accurate predictions for cluster sizes, where edge effects play a role.

On the technical side, some architectural properties of the GNN can be exploited to boost the performance of the model. For instance, enabling NNN edges in input graphs without assigning physical values to their features is of great help. In this way, “skip connections” are created through which the neural network can back-propagate and help the flow of information to get through the neural layers. The local magnetization provides little information to the training and should also be set to identity. Including more cluster sizes in the training data sets significantly improves the extrapolation capabilities of the GNN, although three small cluster sizes seem to suffice to predict much larger cluster sizes of atoms with very high fidelity (with the exact correlators).

Based on this systematic study of the PNA GNN, we can easily identify various avenues to fine tune the neural network and tailor it to specific needs. The points brought up in what follows should further enhance the GNN performance in predicting the NN relative distances. As a first point, one could add more cluster sizes to the training data set $\{4 \times 4, 5 \times 5, 6 \times 6\}$ considered here, such as sizes $4 \times 5, 5 \times 6$, etc. Second, one could increase the number of correlators to include longer-range ones, thereby amplifying the over-completion of the task of inferring NN relative distances. Third, one could enable higher-order graph edges that are not associated with any correlator in the data sets, since this has proven to be beneficial (cases 4 and 5) to the performance; these “weightless” edges would act like “skip connections” in the back-propagation of the gradients during the GNN optimization. Lastly, the $\bar{\Omega}$ history used to produce the training data set could be more carefully chosen to remain near a quantum phase transition, with an increased number of $\bar{\Omega}$ slices.

We are currently exploring applications of the method described in this paper to time-dependent experimental data on Rydberg atomic arrays. Looking forward, the GNN (and MLP) could be trained on various types of Hamiltonians [24] and, therefore, could predict many possible Hamiltonian terms and parameters, e.g., identifying unknown sources of noise from the measurement data instead of only predicting the parameters of a predefined Hamiltonian. To achieve a larger generality of the GNN architecture described in this work, one needs to include many-body edges in the GNN layers and task networks, the inputs of which are multiple nodes. It is also interesting to consider the limitations of the GNN scalability to larger system sizes and whether it depends on the Hamiltonian locality (the power scaling of the interaction decay). Eventually, one could envision feedback control of the experimental setup to reduce uncertainties on the simulated Hamiltonian.

The DMRG code used to generate the data sets is available from the authors upon request and was written using the library ITensors.jl [95]. The PNA GNN codes to train and test upon the data sets are available at Ref. [96]. The PYTHON scripts used to produce the figures are also available on Zenodo [97]. The GNN code was written using utilities from PyTorch Geometric [98].

ACKNOWLEDGMENTS

We are grateful to Antoine Browaeys, Thierry Lahaye, and their group at Institut d’Optique Graduate School for getting us interested in this problem and for useful discussions. We thank Krzysztof Maziarz for suggesting PNA as the GNN architecture. We also thank Lucy Reading-Ikkanda for the graphical design of Figs. 1 and 2. O.S. thanks Sékou-Oumar Kaba for useful discussions regarding the GNN implementations. A.D. thanks Liubov Markovich for the discussions on Hamiltonian-learning techniques. O.S. acknowledges support from the Swiss National Science Foundation through the Postdoc.Mobility fellowship. A.D. acknowledges support from the Dutch National Growth Fund (NGF), as part of the Quantum Delta NL programme. This work was granted access to the high-performance computing (HPC) resources of the Très Grand Centre de Calcul (TGCC) and l’Institut du Développement et des Ressources en Informatique Scientifique (IDRIS) under the allocation A0170510609, attributed by Grand Equipement National de Calcul Intensif (GENCI). It has also used the HPC resources of Infrastructure, Données, Calcul Scientifique (IDCS), under the allocation CPHI 2024. This work is part of the France Hybrid HPC Quantum Initiative (HQI) initiative [99] and is supported by France 2030 under the French

National Research Agency Award No. ANR-22-PNCQ-0002. The Flatiron Institute is a division of the Simons Foundation.

DATA AVAILABILITY

The data that support the findings of this paper are openly available [97].

APPENDIX A: PROOF OF BIJECTION BETWEEN INTERACTIONS AND SPIN-SPIN CORRELATORS

In this appendix, we prove a general result that provides a theoretical foundation for the Hamiltonian reconstruction approach implemented in this work. In part, we show that it suffices to measure NN spin correlation functions to determine uniquely the NN relative distances between atoms and, therefore, the full set of interactions. We show that there is, in general, a bijective correspondence relating the matrix of spatially dependent correlation functions c_{ij} to the matrix of interactions J_{ij} between Ising spins in the TFIM. This result can be viewed as a generalization of the Hohenberg-Kohn (HK) theorem of density-functional theory [72,73], which establishes a bijection between the local one-body potential and the local density of a system of interacting quantum particles, and as a quantum many-body generalization of the Henderson theorem [74]. We note that previous work has demonstrated that a neural network can indeed be trained to learn the HK correspondence in interacting fermion models [100].

We write the TFIM Hamiltonian in the form

$$\hat{H} = \Omega \sum_i \hat{\sigma}_i^x + \delta \sum_i \hat{\sigma}_i^z + \sum_{i < j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z, \quad (\text{A1})$$

where $i < j$ indicates that each link is counted only once. We define the correlation functions as $c_{ij} = \langle \hat{S}_i^z \hat{S}_j^z \rangle$, where $\langle \cdot \rangle$ indicates the expectation value and $\hat{S}_i^z$ is the spin operator in the Z basis acting on the i th site, related to the Z Pauli matrix via $\hat{S}_i^z = \frac{\hbar}{2} \hat{\sigma}_i^z$. We prove that, for fixed $\Omega \neq 0$ and δ , there is a unique set of interactions J_{ij} that yields a set c_{ij} and, conversely, $J_{ij} \Leftrightarrow c_{ij}$. The condition $\Omega \neq 0$ is important since, e.g., when $\Omega = \delta = 0$, for all choices of $\{J_{ij}\}$ with each individual $J_{ij} < 0$, the ground state is the fully polarized ferromagnet and the c_{ij} 's are obviously the same, irrespective of the specific values of the interactions.

First, we need two properties of the ground states of the Hamiltonian when $\Omega \neq 0$. We use $|\sigma\rangle \equiv |\sigma_1, \dots, \sigma_N\rangle$ to denote the eigenstates of all the $\hat{S}_i^z$'s. These properties are consequences of the stoquasticity [75] of the TFIM and the irreducibility [76] of the canonical ensemble density matrix in the standard basis. We give an explicit proof for completeness.

Lemma. For the TFIM Hamiltonian $\hat{H}$ defined by Eq. (A1) with $\Omega \neq 0$, we have the following properties:

- (1) the ground state $|\Psi\rangle$ of $\hat{H}$ is unique
- (2) $\langle \sigma | \Psi \rangle \neq 0$ for all $|\sigma\rangle$

Proof. First, it is sufficient to prove the result for $\Omega < 0$, since we can find a unitary transformation $U := \prod_j (i\hat{\sigma}_j^y)$ that changes the sign of Ω and δ but leaves $\{J_{ij}\}$ intact. Since each $i\hat{\sigma}_j^y$ locally maps $|\uparrow\rangle$ to $-\downarrow\rangle$ and $|\downarrow\rangle$ to $|\uparrow\rangle$ at the j th site, the $|\sigma\rangle$'s are permuted among each other modulo an extra sign. Hence, the statement proved for the transformed Hamiltonian is true for the original sign.

Second, it is sufficient to prove that $V = e^{-\beta\hat{H}}$, $\beta > 0$, is a positive matrix; namely, it has all positive entries. Then, the Perron-Frobenius theorem [101–103] tells us that the dominant (Perron) eigenvalue is simple (nondegenerate) and the corresponding (Perron) eigenvector has all positive components. Note that V is Hermitian with positive eigenvalues, with the Perron eigenvalue being $\exp(-\beta E)$ when E is the ground-state energy of $\hat{H}$ and the Perron eigenvector is the ground state $|\Psi\rangle$. Now, we break up $\hat{H} = \hat{H}_0 + \Omega \sum_i \hat{\sigma}_i^x$, where

$$\hat{H}_0 = \delta \sum_i \hat{\sigma}_i^z + \sum_{i < j} J_{ij} \hat{\sigma}_i^z \hat{\sigma}_j^z. \quad (\text{A2})$$

We are ready to use the series expansion in the interaction picture (see, e.g., Ref. [104]) for V entries:

$$\begin{aligned} \langle \sigma' | V | \sigma \rangle &= \langle \sigma' | e^{-\tau\hat{H}} | \sigma \rangle \\ &= \sum_{n=0}^{\infty} (-\Omega)^n \sum_{i_1, i_2, \dots, i_n} \int_{\tau_{n-1}}^{\beta} d\tau_n \cdots \int_{\tau_1}^{\beta} d\tau_2 \int_0^{\beta} d\tau_n \\ &\quad \langle \sigma' | e^{-\beta\hat{H}_0} \hat{\sigma}_{i_n}^x(\tau_n) \cdots \hat{\sigma}_{i_2}^x(\tau_2) \hat{\sigma}_{i_1}^x(\tau_1) | \sigma \rangle, \end{aligned} \quad (\text{A3})$$

where $\hat{\sigma}_i^x(\tau) := e^{\tau\hat{H}_0} \hat{\sigma}_i^x e^{-\tau\hat{H}_0}$. This operator causes a spin flip at site i . All of the terms of this convergent series are non-negative. Moreover, between any $|\sigma\rangle$ and $|\sigma'\rangle$, there is at least one term with at most N $\hat{\sigma}_{i_k}^x$'s that is nonzero. Hence V has all-positive entries. ■

Now we come to the main theorem, adapted to the TFIM, which assumes the J -representability of the connected correlation functions [70,71].

Theorem (Hohenberg-Kohn-Henderson Theorem for the TFIM). For the TFIM as defined in Eq. (A1), for fixed nonzero Ω and fixed δ , there exists a bijection between the J -representable correlation functions $c_{ij} = \langle \hat{S}_i^z \hat{S}_j^z \rangle$ and the set of interactions J_{ij} , $J_{ij} \Leftrightarrow c_{ij}$.

Proof. The direction $J_{ij} \Rightarrow c_{ij}$ is obvious. We now prove that $c_{ij} \Rightarrow J_{ij}$ by *reductio ad absurdum*, following the same logic for the classic HK theorem [72]. By the aforementioned lemma, the ground state of $\hat{H}$ is nondegenerate.

For fixed values of Ω ($\neq 0$) and δ , we therefore start by assuming that one can construct two ground-state wave functions, Ψ_1 and Ψ_2 , with identical c_{ij} 's but different J_{ij} 's, i.e., $J_{ij}^{(1)}$ and $J_{ij}^{(2)}$. Therefore, the Hamiltonians $\hat{H}_1$ and $\hat{H}_2$, the ground states of which are Ψ_1 and Ψ_2 , with ground-state energies E_1 and E_2 , are identical besides the J_{ij} term:

$$\hat{H}_1 = \Omega \sum_i \hat{\sigma}_i^x + \delta \sum_i \hat{\sigma}_i^z + \sum_{i<j} J_{ij}^{(1)} \hat{\sigma}_i^z \hat{\sigma}_j^z, \quad (\text{A4})$$

$$\hat{H}_2 = \Omega \sum_i \hat{\sigma}_i^x + \delta \sum_i \hat{\sigma}_i^z + \sum_{i<j} J_{ij}^{(2)} \hat{\sigma}_i^z \hat{\sigma}_j^z. \quad (\text{A5})$$

We first show that $\Psi_1 \neq \Psi_2$. Assuming that, instead, $\Psi_1 = \Psi_2 \equiv \Psi$, one then has $(\hat{H}_1 - \hat{H}_2)|\Psi\rangle = (E_1 - E_2)|\Psi\rangle$. Expanding $|\Psi\rangle$ on a basis of eigenstates $|\sigma\rangle \equiv |\sigma_1, \dots, \sigma_N\rangle$ of the $\hat{S}_i^z$'s, one then obtains

$$\sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) \sigma_i \sigma_j = (E_1 - E_2), \quad (\text{A6})$$

which must be satisfied for *all* configurations σ for which $\langle \sigma | \Psi \rangle \neq 0$. But $\langle \sigma | \Psi \rangle \neq 0$ is true for all possible σ by the previous lemma. So Eq. (A6) is satisfied for all 2^N configurations of σ . If we sum this equation over all possible configurations, we obtain $0 = 2^N (E_1 - E_2)$, as $\sum_{\sigma} \sigma_i \sigma_j = 0$ for any $i \neq j$. Thus, $E_1 = E_2$.

We now have the equation

$$\sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) \sigma_i \sigma_j = 0. \quad (\text{A7})$$

To arrive at a contradiction, we will show that $J_{ij}^{(1)} = J_{ij}^{(2)}$ for every i, j pair with $i < j$. After picking one such specific i, j pair, choose all configurations in which $\sigma_i = 1, \sigma_j = 1$. Call this set s_{ij} . When $i' < j'$, $\sum_{\sigma \in s_{ij}} \sigma_{i'} \sigma_{j'} = 2^{N-2} \delta_{i'i'} \delta_{j'j'}$, δ_{ij} being the Kronecker delta. Summing Eq. (A7) over $\sigma \in s_{ij}$, we obtain $2^{N-2} (J_{ij}^{(1)} - J_{ij}^{(2)}) = 0$. This is in contradiction to the couplings being distinct. Hence, $\Psi_1 \neq \Psi_2$.

Having established that, we can now use the variational principle in the form of a *strict* inequality for both Hamiltonians:

$$E_1 = \langle \Psi_1 | \hat{H}_1 | \Psi_1 \rangle < \langle \Psi_2 | \hat{H}_1 | \Psi_2 \rangle, \quad (\text{A8})$$

$$E_2 = \langle \Psi_2 | \hat{H}_2 | \Psi_2 \rangle < \langle \Psi_1 | \hat{H}_2 | \Psi_1 \rangle. \quad (\text{A9})$$

Let us expand the right-hand sides of the inequalities:

$$\begin{aligned} \langle \Psi_2 | \hat{H}_1 | \Psi_2 \rangle &= \langle \Psi_2 | (\hat{H}_1 - \hat{H}_2) | \Psi_2 \rangle + \langle \Psi_2 | \hat{H}_2 | \Psi_2 \rangle \\ &= \langle \Psi_2 | \sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) \hat{\sigma}_i^z \hat{\sigma}_j^z | \Psi_2 \rangle + E_2 \\ &= \sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) c_{ij} + E_2, \end{aligned}$$

$$\begin{aligned} \langle \Psi_1 | \hat{H}_2 | \Psi_1 \rangle &= \langle \Psi_1 | \sum_{i<j} (J_{ij}^{(2)} - J_{ij}^{(1)}) \hat{\sigma}_i^z \hat{\sigma}_j^z | \Psi_1 \rangle + E_1 \\ &= - \sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) c_{ij} + E_1. \end{aligned}$$

Now, let us add Eqs. (A8) and (A9):

$$\begin{aligned} E_1 + E_2 &< E_1 + E_2 + \sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) c_{ij} \\ &\quad - \sum_{i<j} (J_{ij}^{(1)} - J_{ij}^{(2)}) c_{ij}; \end{aligned} \quad (\text{A10})$$

hence

$$E_1 + E_2 < E_1 + E_2 \quad (\text{A11})$$

is a clear contradiction.

We thus conclude by *reductio ad absurdum* that one cannot have two ground-state wave functions with identical $c_{ij} = \langle \hat{S}_i^z \hat{S}_j^z \rangle$ and different J_{ij} 's, and hence that $\langle \hat{S}_i^z \hat{S}_j^z \rangle \implies J_{ij}$. ■

APPENDIX B: NUMBER OF PHYSICAL SYSTEM SIZES IN TRAINING DATA SETS

In this appendix, the effect of the number of physical system sizes included in the training data sets is studied. Here, we utilize exact training data sets.

In Fig. 10, for the set of sizes $\{4 \times 4, 5 \times 5\}$ and $\{4 \times 4, 5 \times 5, 6 \times 6\}$, we plot the coefficient of determination R^2 [see Eq. (5)] in Fig. 10(a), the MAE [see Eq. (6)] in Fig. 10(b), and the median of the MAE (MEDAE) in Fig. 10(c). Case 3 is utilized. Without much surprise, the GNN performs much better at extrapolating relative NN distances $\Delta R_{(ij)}$ for clusters whose sizes lie outside the training set, because of system-size effects. It may, however, come as a surprise that the smaller system sizes, i.e., 4×4 and 5×5 , perform worse than that 6×6 within the training-size domain, but it seems that this effect mostly comes from the inclusion of the χ^{NNN} , as seen in Fig. 4. Since smaller clusters are subparts of larger clusters, the information of the smaller clusters aggregates and makes it possible to predict with high precision the ΔR 's of clusters with sizes beyond the training set.

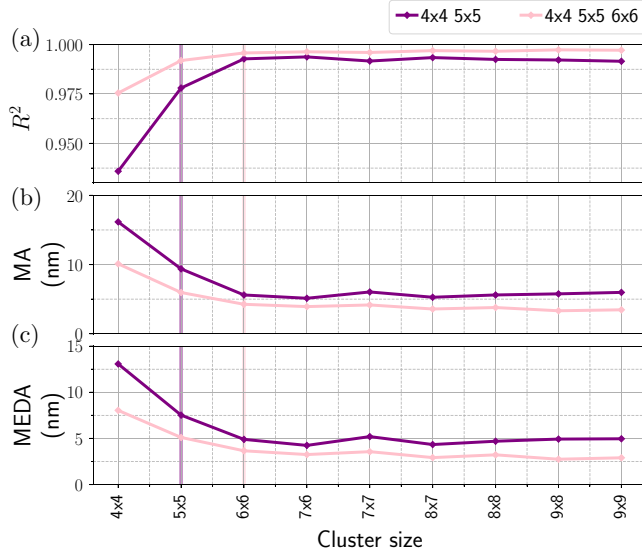


FIG. 10. The training data-set cluster-size effect. The incidence of the cluster sizes included in the exact training data set on various metrics: (a) R^2 ; (b) the mean absolute error; (c) the median of the absolute error. In violet, clusters of sizes 4×4 and 5×5 have been considered, whereas in light pink, one extra cluster size, 6×6 , has been considered. The vertical lines refer to the point beyond which the sizes lie outside of the training range, i.e., these sizes did not form part of the training data sets. The colors of the vertical lines match with those of the curves.

APPENDIX C: PREDICTING NN AND NNN DISTANCES

In this appendix, we check the performance of the GNN in predicting the NN and NNN relative distances simultaneously. Again, we utilize exact training data sets.

Regarding the cost function in Eq. (2), one could ask whether one should use both the relative NN and NNN distances whenever both χ^{NN} and χ^{NNN} are provided, or if one should use an overcomplete set of physical short-range correlators to predict only the relative NN distances, i.e., χ^{NN} and χ^{NNN} . As proven in Appendix A, with arbitrary precision on the correlation functions, one would theoretically only need χ^{NN} to predict relative NN distances $\Delta R_{(i,j)}$ between Rydberg atoms. However, we may think that giving some extra information to the GNN could be beneficial to overcome numerical or experimental imprecision. To that effect, in Fig. 11, we compare two GNN training cases 3 where, on one hand, only relative NN distances are used as targets and, on the other hand, both NN and NNN relative distances are used as targets, for training sizes $\{4 \times 4, 5 \times 5, 6 \times 6\}$ and $\{4 \times 4, 5 \times 5, 6 \times 6, 7 \times 7\}$. It is very clear that the overcompletion of information helps the GNN palliate the numerical and/or experimental inaccuracies in determining the relative distances, meaning that predicting solely NN relative distances with χ^{NN} and χ^{NNN} at hand is better than predicting both NN and NNN relative distances with the

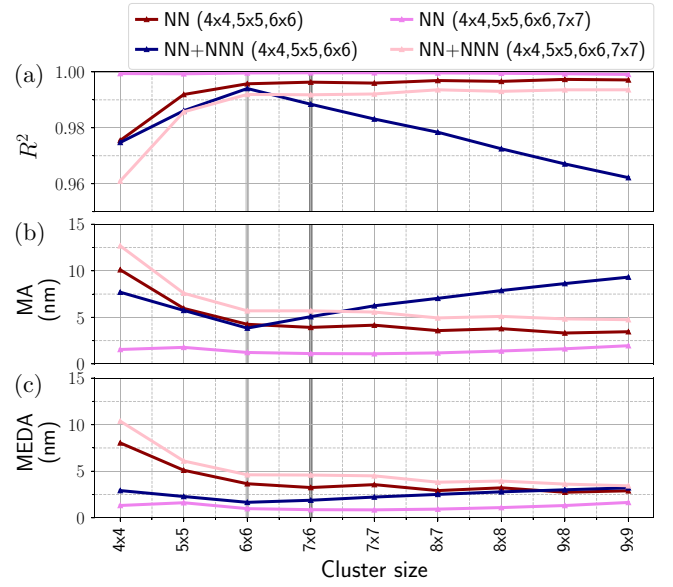


FIG. 11. Target optimization schemes. The incidence on various metrics of including the NNN relative distance as a target in the cost-function minimization: (a) R^2 ; (b) the mean absolute error; (c) the median of the absolute error. In blue and light pink, both the NN and NNN relative distances are predicted, while in red and pink, only the NN ones are predicted. The training sizes used are indicated in parentheses. The vertical gray lines refer to the point beyond which the sizes lie outside of the training range.

same information throughput. It also seems that predictions of larger relative distances (NNN and above) require larger cluster sizes in the training set because edge effects prevail for longer. Therefore, if one seeks to also predict NNN relative distances between atoms, one should consider training the GNN with larger cluster sizes to substantially improve the predictive power of the model (compare the navy blue curve in Fig. 11 with the pink one). Interestingly, adding another task network specializing in predicting NNN distances worsened the GNN performance compared to using the same task network for predicting NN and NNN distances.

APPENDIX D: DETAILS ON THE NUMERICAL IMPLEMENTATION AND HYPERPARAMETERS USED

Some technical details are set out in this appendix. The GNN was implemented using PyTorch [98] and the data making up the mutually exclusive training and testing data sets were churned out using the DMRG library ITensors.jl [95].

For all cases laid out in Table I, the same GNN hyperparameters were used. The PNA GNN consisted in 4 layers, and the number of node and edge features corresponded to the size of the Ω history (10 Ω values). The node-embedding dimension used was 32. The edge-feature

dimension corresponds to the combined number of NN edges and, if applicable, NNN edges embedded into a higher-dimensional space through a three-layer perceptron. 550 epochs were sufficient to train the GNN and obtain a validation loss comparable to the training one. A histogram of the degree of connectedness of the training data-set graphs was provided to the GNN during the training phase. The number of graphs in the training data set was 2000 per physical system size, times the Ω history, making it 20 000 samples per physical system size. The total number of samples was the same whether the snapshots were used to calculate the correlation function or direct evaluation was employed. The testing data set comprised 200 graphs per physical system size times the Ω history.

Regarding the learning-rate profile, the same linearly decreasing learning rate was employed to optimize all the GNNs trained. The starting learning-rate value was 5×10^{-3} while the ending value was 2.5×10^{-4} , for a total of 550 epochs. The learning-rate profile plays a big role in the training process: its ramp and extrema should be carefully tuned. The linearly decreasing ramp just mentioned was a good ramp profile.

APPENDIX E: TECHNICAL DETAILS OF THE DMRG CALCULATIONS

To ensure that the MPSs were converged after calculation, the maximum truncation error in the singular-value decomposition was capped to single-precision floating-point accuracy (1×10^{-7}) and we made sure that the energy estimations had converged to seven decimal places with increasing bond dimension through the sweeps. The energy converges within each sweep and the bond dimension increases from one sweep to another. We monitored how the maximum truncation error and energy estimation changed through 20 DMRG sweeps. To achieve that, we used a maximal bond dimension of $\chi_{\text{DMRG}} = 80$ for a cluster size 4×4 and then increased χ_{DMRG} by 10 for larger cluster sizes, up to 9×9 , to obtain $\chi_{\text{DMRG}} = 160$.

-
- [1] I. Bloch, J. Dalibard, and W. Zwerger, Many-body physics with ultracold gases, *Rev. Mod. Phys.* **80**, 885 (2008).
 - [2] L. Lamata, A. Parra-Rodriguez, M. Sanz, and E. Solano, Digital-analog quantum simulations with superconducting circuits, *Adv. Phys.: X* **3**, 1457981 (2018).
 - [3] M. Tomza, K. Jachymski, R. Gerritsma, A. Negretti, T. Calarco, Z. Idziaszek, and P. S. Julienne, Cold hybrid ion-atom systems, *Rev. Mod. Phys.* **91**, 035001 (2019).
 - [4] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Superconducting qubits: Current state of play, *Annu. Rev. Condens. Matter Phys.* **11**, 369 (2020).
 - [5] A. Browaeys and T. Lahaye, Many-body physics with individually controlled Rydberg atoms, *Nat. Phys.* **16**, 132 (2020).
 - [6] C. Monroe, W. C. Campbell, L.-M. Duan, Z.-X. Gong, A. V. Gorshkov, P. W. Hess, R. Islam, K. Kim, N. M. Linke, G. Pagano, P. Richerme, C. Senko, and N. Y. Yao, Programmable quantum simulations of spin systems with trapped ions, *Rev. Mod. Phys.* **93**, 025001 (2021).
 - [7] M. A. Baranov, M. Dalmonte, G. Pupillo, and P. Zoller, Condensed matter theory of dipolar quantum gases, *Chem. Rev.* **112**, 5012 (2012).
 - [8] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, *Science* **354**, 1024 (2016).
 - [9] D. Barredo, S. de Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays, *Science* **354**, 1021 (2016).
 - [10] W. S. Bakr, J. I. Gillen, A. Peng, S. Fölling, and M. Greiner, A quantum gas microscope for detecting single atoms in a Hubbard-regime optical lattice, *Nature* **462**, 74 (2009).
 - [11] J. F. Sherson, C. Weitenberg, M. Endres, M. Cheneau, I. Bloch, and S. Kuhr, Single-atom-resolved fluorescence imaging of an atomic Mott insulator, *Nature* **467**, 68 (2010).
 - [12] L. W. Cheuk, M. A. Nichols, M. Okan, T. Gersdorf, V. V. Ramasesh, W. S. Bakr, T. Lompe, and M. W. Zwierlein, Quantum-gas microscope for fermionic atoms, *Phys. Rev. Lett.* **114**, 193001 (2015).
 - [13] E. Haller, J. Hudson, A. Kelly, D. A. Cotta, B. Peaudecerf, G. D. Bruce, and S. Kuhr, Single-atom imaging of fermions in a quantum-gas microscope, *Nat. Phys.* **11**, 738 (2015).
 - [14] D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Côté, and M. D. Lukin, Fast quantum gates for neutral atoms, *Phys. Rev. Lett.* **85**, 2208 (2000).
 - [15] D. DeMille, Quantum computation with trapped polar molecules, *Phys. Rev. Lett.* **88**, 4 (2002).
 - [16] M. Morgado and S. Whitlock, Quantum simulation and computing with Rydberg-interacting qubits, *AVS Quantum Sci.* **3**, 023501 (2021).
 - [17] A. D. Ludlow, M. M. Boyd, J. Ye, E. Peik, and P. O. Schmidt, Optical atomic clocks, *Rev. Mod. Phys.* **87**, 637 (2015).
 - [18] K. Bongs, M. Holynski, J. Vovrosh, P. Bouyer, G. Condon, E. Rasel, C. Schubert, W. P. Schleich, and A. Roura, Taking atom interferometric quantum sensors from the laboratory to real-world applications, *Nat. Rev. Phys.* **1**, 731 (2019).
 - [19] J. Eisert, D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, Quantum certification and benchmarking, *Nat. Rev. Phys.* **2**, 382 (2020).
 - [20] J. Carrasco, A. Elben, C. Kokail, B. Kraus, and P. Zoller, Theoretical and experimental perspectives of quantum verification, *PRX Quantum* **2**, 010102 (2021).
 - [21] X.-L. Qi and D. Ranard, Determining a local Hamiltonian from a single eigenstate, *Quantum* **3**, 159 (2019).
 - [22] E. Bairey, I. Arad, and N. H. Lindner, Learning a local Hamiltonian from local measurements, *Phys. Rev. Lett.* **122**, 020504 (2019).

- [23] E. Bairey, C. Guo, D. Poletti, N. H. Lindner, and I. Arad, Learning the dynamics of open quantum systems from their steady states, *New J. Phys.* **22**, 032001 (2020).
- [24] S. Nandy, M. Schmitt, M. Bukov, and Z. Lenarčič, Reconstructing effective Hamiltonians from nonequilibrium thermal and prethermal steady states, *Phys. Rev. Res.* **6**, 023160 (2024).
- [25] Z. Li, L. Zou, and T. H. Hsieh, Hamiltonian tomography via quantum quench, *Phys. Rev. Lett.* **124**, 160502 (2020).
- [26] T. Olsacher, T. Kraft, C. Kokail, B. Kraus, and P. Zoller, Hamiltonian and Liouvillian learning in weakly-dissipative quantum many-body systems, *Quantum Sci. Technol.* **10**, 015065 (2025).
- [27] S.-T. Wang, D.-L. Deng, and L.-M. Duan, Hamiltonian tomography for quantum many-body systems with arbitrary couplings, *New J. Phys.* **17**, 093017 (2015).
- [28] T. J. Evans, R. Harper, and S. T. Flammia, Scalable Bayesian Hamiltonian learning, *ArXiv:1912.07636*.
- [29] H. Landa, D. Meirom, N. Kanazawa, M. Fitzpatrick, and C. J. Wood, Experimental Bayesian estimation of quantum state preparation, measurement, and gate errors in multiqubit devices, *Phys. Rev. Res.* **4**, 013199 (2022).
- [30] N. Wiebe, C. Granade, C. Ferrie, and D. G. Cory, Hamiltonian learning and certification using quantum resources, *Phys. Rev. Lett.* **112**, 190501 (2014).
- [31] J. Wang, S. Paesani, R. Santagati, S. Knauer, A. A. Gentile, N. Wiebe, M. Petruzzella, J. L. O'Brien, J. G. Rarity, A. Laing, and M. G. Thompson, Experimental quantum Hamiltonian learning, *Nat. Phys.* **13**, 551 (2017).
- [32] F. Wilde, A. Kshetrimayum, I. Roth, D. Hangleiter, R. Sweke, and J. Eisert, Scalably learning quantum many-body Hamiltonians from dynamical data, *ArXiv:2209.14328*.
- [33] T. Heightman, E. Jiang, and A. Acín, Solving the quantum many-body Hamiltonian learning problem with neural differential equations, *ArXiv:2408.08639*.
- [34] D. Stilck França, L. A. Markovich, V. V. Dobrovitski, A. H. Werner, and J. Borregaard, Efficient and robust estimation of many-qubit Hamiltonians, *Nat. Commun.* **15**, 311 (2024).
- [35] A. Gu, L. Cincio, and P. J. Coles, Practical Hamiltonian learning with unitary dynamics and Gibbs states, *Nat. Commun.* **15**, 312 (2024).
- [36] C. Kokail, B. Sundar, T. V. Zache, A. Elben, B. Vermersch, M. Dalmonte, R. van Bijnen, and P. Zoller, Quantum variational learning of the entanglement Hamiltonian, *Phys. Rev. Lett.* **127**, 170501 (2021).
- [37] A. Dawid *et al.*, *Machine Learning in Quantum Sciences* (Cambridge University Press, Cambridge, England, 2025), ISBN 9781009504935.
- [38] V. Gebhart, R. Santagati, A. A. Gentile, E. M. Gauger, D. Craig, N. Ares, L. Bianchi, F. Marquardt, L. Pezzè, and C. Bonato, Learning quantum systems, *Nat. Rev. Phys.* **5**, 141 (2023).
- [39] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Neural-network quantum state tomography, *Nat. Phys.* **14**, 447 (2018).
- [40] J. Carrasquilla, G. Torlai, and R. G. Melko, Reconstructing quantum states with generative models, *Nat. Mach. Intell.* **1**, 155 (2019).
- [41] G. Torlai, B. Timar, E. P. Van Nieuwenburg, H. Levine, A. Omran, A. Keesling, H. Bernien, M. Greiner, V. Vuletić, M. D. Lukin, R. G. Melko, and M. Endres, Integrating neural networks with a quantum simulator for state reconstruction, *Phys. Rev. Lett.* **123**, 19 (2019).
- [42] S. Ahmed, C. Sánchez Muñoz, F. Nori, and A. F. Kockum, Quantum state tomography with conditional generative adversarial networks, *Phys. Rev. Lett.* **127**, 140502 (2021).
- [43] H. Lange, M. Kebrič, M. Buser, U. Schollwöck, F. Grusdt, and A. Bohrdt, Adaptive quantum state tomography with active learning, *Quantum* **7**, 1129 (2023).
- [44] Z. An, J. Wu, M. Yang, D. L. Zhou, and B. Zeng, Unified quantum state tomography and Hamiltonian learning: A language-translation-like approach for quantum systems, *Phys. Rev. Appl.* **21**, 014037 (2024).
- [45] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, *Nat. Phys.* **13**, 435 (2017).
- [46] Y.-H. Liu and E. P. L. van Nieuwenburg, Discriminative cooperative networks for detecting phase transitions, *Phys. Rev. Lett.* **120**, 176401 (2018).
- [47] E. Greplova, A. Valenti, G. Boschung, F. Schäfer, N. Lörch, and S. D. Huber, Unsupervised identification of topological phase transitions using predictive models, *New J. Phys.* **22**, 045003 (2020).
- [48] N. Käming, A. Dawid, K. Kottmann, M. Lewenstein, K. Sengstock, A. Dauphin, and C. Weitenberg, Unsupervised machine learning of topological phase transitions from experimental data, *Mach. Learn.: Sci. Technol.* **2**, 035037 (2021).
- [49] Z. Patel, E. Merali, and S. J. Wetzel, Unsupervised learning of Rydberg atom array phase diagram with Siamese neural networks, *New J. Phys.* **24**, 113021 (2022).
- [50] C. Miles, R. Samajdar, S. Ebadi, T. T. Wang, H. Pichler, S. Sachdev, M. D. Lukin, M. Greiner, K. Q. Weinberger, and E.-A. Kim, Machine learning discovery of new phases in programmable quantum simulator snapshots, *Phys. Rev. Res.* **5**, 013026 (2023).
- [51] K. Cybiński, J. Enouen, A. Georges, and A. Dawid, in *Machine Learning and Physical Sciences Workshop at the 38th Conference on Neural Information Processing Systems (NeurIPS)* (Curran Associates, Vancouver, BC, Canada, 2024).
- [52] T. Xin, S. Lu, N. Cao, G. Anikeeva, D. Lu, J. Li, G. Long, and B. Zeng, Local-measurement-based quantum state tomography via neural networks, *npj Quantum Inf.* **5**, 1 (2019).
- [53] A. Valenti, E. van Nieuwenburg, S. Huber, and E. Greplova, Hamiltonian learning for quantum error correction, *Phys. Rev. Res.* **1**, 033092 (2019).
- [54] L. Che, C. Wei, Y. Huang, D. Zhao, S. Xue, X. Nie, J. Li, D. Lu, and T. Xin, Learning quantum Hamiltonians from single-qubit measurements, *Phys. Rev. Res.* **3**, 023246 (2021).
- [55] É. Genois, J. A. Gross, A. D. Paolo, N. J. Stevenson, G. Koolstra, A. Hashim, I. Siddiqi, and A. Blais, Quantum-tailored machine-learning characterization of a superconducting qubit, *PRX Quantum* **2**, 040355 (2021).
- [56] A. Valenti, G. Jin, J. Léonard, S. D. Huber, and E. Greplova, Scalable Hamiltonian learning for large-scale out-of-equilibrium quantum dynamics, *Phys. Rev. A* **105**, 023302 (2022).

- [57] R. Koch, D. van Driel, A. Bordin, J. L. Lado, and E. Grepova, Adversarial Hamiltonian learning of quantum dots in a minimal Kitaev chain, *Phys. Rev. Appl.* **20**, 044081 (2023).
- [58] N. Karjalainen, Z. Lippo, G. Chen, R. Koch, A. O. Fumega, and J. L. Lado, Hamiltonian inference from dynamical excitations in confined quantum magnets, *Phys. Rev. Appl.* **20**, 024054 (2023).
- [59] K. Slagle, Y. Liu, D. Aasen, H. Pichler, R. S. K. Mong, X. Chen, M. Endres, and J. Alicea, Quantum spin liquids bootstrapped from Ising criticality in Rydberg arrays, *Phys. Rev. B* **106**, 115122 (2022).
- [60] S. Morita, R. Kaneko, and M. Imada, Quantum spin liquid in spin 1/2 J1–J2 Heisenberg model on square lattice: Many-variable variational Monte Carlo study combined with quantum-number projections, *J. Phys. Soc. Jpn.* **84**, 024720 (2015).
- [61] W.-J. Hu, S.-S. Gong, W. Zhu, and D. N. Sheng, Competing spin-liquid states in the spin- $\frac{1}{2}$ Heisenberg model on the triangular lattice, *Phys. Rev. B* **92**, 140403 (2015).
- [62] A. Wietek and A. M. Läuchli, Chiral spin liquid and quantum criticality in extended $S = \frac{1}{2}$ Heisenberg models on the triangular lattice, *Phys. Rev. B* **95**, 035141 (2017).
- [63] H. Labuhn, D. Barredo, S. Ravets, S. de Léséleuc, T. Macrì, T. Lahaye, and A. Browaeys, Tunable two-dimensional arrays of single Rydberg atoms for realizing quantum Ising models, *Nature* **534**, 667 (2016).
- [64] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, Quantum simulation of 2D antiferromagnets with hundreds of Rydberg atoms, *Nature* **595**, 233 (2021).
- [65] L. Henriët, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Raymond, and C. Jurczak, Quantum computing with neutral atoms, *Quantum* **4**, 327 (2020).
- [66] D. Barredo, H. Labuhn, S. Ravets, T. Lahaye, A. Browaeys, and C. S. Adams, Coherent excitation transfer in a spin chain of three Rydberg atoms, *Phys. Rev. Lett.* **114**, 113002 (2015).
- [67] P. Scholl, H. J. Williams, G. Bornet, F. Wallner, D. Barredo, L. Henriët, A. Signoles, C. Hainaut, T. Franz, S. Geier, A. Tebben, A. Salzinger, G. Zürn, T. Lahaye, M. Weidemüller, and A. Browaeys, Microwave engineering of programmable XXZ Hamiltonians in arrays of Rydberg atoms, *PRX Quantum* **3**, 020303 (2022).
- [68] S. R. White, Density matrix formulation for quantum renormalization groups, *Phys. Rev. Lett.* **69**, 2863 (1992).
- [69] S. R. White, Density-matrix algorithms for quantum renormalization groups, *Phys. Rev. B* **48**, 10345 (1993).
- [70] M. Levy, Electron densities in search of Hamiltonians, *Phys. Rev. A* **26**, 1200 (1982).
- [71] H. Englisch and R. Englisch, Hohenberg-Kohn theorem and non- V -representable densities, *Physica A: Statistical Mechanics and its Applications* **121**, 253 (1983).
- [72] P. Hohenberg and W. Kohn, Inhomogeneous electron gas, *Phys. Rev.* **136**, B864 (1964).
- [73] L. Garrigue, Hohenberg-Kohn theorems for interactions, spin and temperature, *J. Stat. Phys.* **177**, 415 (2019).
- [74] R. Henderson, A uniqueness theorem for fluid pair correlation functions, *Phys. Lett. A* **49**, 197 (1974).
- [75] S. Bravyi, D. P. Divincenzo, R. Oliveira, and B. M. Terhal, The complexity of stoquastic local Hamiltonian problems, *Quantum Inf. Comput.* **8**, 361 (2008).
- [76] E. Crosson and J. Bowen, Quantum ground state isoperimetric inequalities for the energy spectrum of local Hamiltonians, *ArXiv:1703.10133*.
- [77] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, Principal neighbourhood aggregation for graph nets, *ArXiv:2004.05718*.
- [78] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, *ArXiv:1711.05101*.
- [79] P. W. Battaglia *et al.*, Relational inductive biases, deep learning, and graph networks, *ArXiv:1806.01261*.
- [80] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, Discovering symbolic models from deep learning with inductive biases, *Adv. Neural Inf. Process. Syst.* **2020-December**, 1 (2020).
- [81] J. Shlomi, P. Battaglia, and J.-R. Vlimant, Graph neural networks in particle physics, *Mach. Learn.: Sci. Technol.* **2**, 021001 (2021).
- [82] M. Cranmer, C. Cui, D. B. Fielding, S. Ho, A. Sanchez-Gonzalez, K. Stachenfeld, T. Pfaff, J. Godwin, P. Battaglia, and D. Kochkov, in *ICLR Workshop on Sparse Neural Networks* (OpenReview, Vienna, Austria (virtual), 2021), Vol. 7.
- [83] P. Lemos, N. Jeffrey, M. Cranmer, S. Ho, and P. Battaglia, Rediscovering orbital mechanics with machine learning, *Mach. Learn.: Sci. Technol.* **4**, 045002 (2023).
- [84] V. T. Tran, L. Lewis, H.-Y. Huang, J. Kofler, R. Kueng, S. Hochreiter, and S. Lehner, in *Machine Learning and Physical Sciences Workshop at the 36th Conference on Neural Information Processing Systems (NeurIPS)* (Curran Associates, New Orleans, LA, USA, 2022), https://ml4physicalsciences.github.io/2022/files/NeurIPS_ML4PS_2022_184.pdf.
- [85] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, Benchmarking graph neural networks, *ArXiv:2003.00982*.
- [86] E. M. Stoudenmire and S. R. White, Minimally entangled typical thermal state algorithms, *New J. Phys.* **12**, 055026 (2010).
- [87] A. J. Ferris and G. Vidal, Perfect sampling with unitary tensor networks, *Phys. Rev. B* **85**, 165146 (2012).
- [88] C. Lam, High-dimensional covariance matrix estimation, *Wiley Interdisciplinary Reviews: Computational Statistics* **12**, e1485 (2020).
- [89] A. M. Sengupta and P. P. Mitra, Distributions of singular values for some random matrices, *Phys. Rev. E* **60**, 3389 (1999).
- [90] L. Laloux, P. Cizeau, J.-P. Bouchaud, and M. Potters, Noise dressing of financial correlation matrices, *Phys. Rev. Lett.* **83**, 1467 (1999).
- [91] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud, Random matrix theory and financial correlations, *Int. J. Theor. Appl. Finance* **3**, 391 (2000).
- [92] O. Ledoit and M. Wolf, The power of (non-)linear shrinkage: A review and guide to covariance matrix estimation, *J. Financ. Econom.* **20**, 187 (2022).

- [93] H. Buhrman, S. Gharibian, Z. Landau, F. L. Gall, N. Schuch, and S. Tamaki, Beating Grover search for low-energy estimation and state preparation, *Phys. Rev. Lett.* **135**, 030601 (2025).
- [94] A. Dutt, E. Pednault, C. W. Wu, S. Sheldon, J. Smolin, L. Bishop, and I. L. Chuang, Active learning of quantum system Hamiltonians yields query advantage, *Phys. Rev. Res.* **5**, 033060 (2023).
- [95] M. Fishman, S. R. White, and E. M. Stoudenmire, The ITensor software library for tensor network calculations, *SciPost Phys. Codebases*, **4** (2022).
- [96] https://github.com/oliviersimard/Hamiltonian_Learning_GNN.git.
- [97] <https://doi:10.5281/zenodo.15525077>.
- [98] A. Paszke *et al.*, PyTorch: An imperative style, high-performance deep learning library, *ArXiv:1912.01703*.
- [99] www.hqi.fr.
- [100] J. Robledo-Moreno, G. Carleo, and A. Georges, Deep learning the Hohenberg-Kohn maps of density functional theory, *Phys. Rev. Lett.* **125**, 076402 (2020).
- [101] O. Perron, Zur Theorie der Matrices, *Math. Ann.* **64**, 248 (1907).
- [102] F. G. Frobenius, Über Matrizen aus Positiven Elementen, *Sitzungsber. Kgl. Preuss. Akad. Wiss.* **1908**, 471 (1908).
- [103] F. G. Frobenius, Über Matrizen aus nicht negativen Elementen, *Sitzungsber. Kgl. Preuss. Akad. Wiss.* **26**, 456 (1912).
- [104] A. W. Sandvik, Stochastic series expansion methods, *ArXiv:1909.10591*.