Scalable approach to many-body localization via quantum data

Alexander Gresch, Lennart Bittel, and Martin Kliesch
Quantum Technology Research Group, Heinrich Heine University Düsseldorf, Germany

We are interested in how quantum data can allow for practical solutions to otherwise difficult computational problems. A notoriously difficult phenomenon from quantum many-body physics is the emergence of many-body localization (MBL). So far, it has evaded a comprehensive analysis. In particular, numerical studies are challenged by the exponential growth of the Hilbert space dimension. As many of these studies rely on exact diagonalization of the system’s Hamiltonian, only small system sizes are accessible.

In this work, we propose a highly flexible neural network based learning approach that, once given training data, circumvents any computationally expensive step. In this way, we can efficiently estimate common indicators of MBL such as the adjacent gap ratio or entropic quantities. Our estimator can be trained on data from various system sizes at once which grants the ability to extrapolate from smaller to larger ones. Moreover, using transfer learning we show that already a two-dimensional feature vector is sufficient to obtain several different indicators at various energy densities at once. We hope that our approach can be applied to large-scale quantum experiments to provide new insights into quantum many-body physics.

I. INTRODUCTION

The goal of quantum computing is to efficiently solve practically relevant problems that are intractable on classical computers. Many those problems require a fault-tolerant, universal quantum computer. This requirement, in turn, comes in conjunction with the need for quantum error correction which yields a daunting overhead in the qubit numbers. Both requirements exceed the current available quantum hardware substantially. Hence, in the meantime, the potential of hybrid quantum algorithms is explored. They aim to optimally use the few dozens of available qubits with no or little error mitigation schemes. Most of their pragmatic approaches are centered around variational quantum algorithms (VQAs) [1, 2]. These algorithms provide heuristics for problems such as finding the ground-state energy in the field of quantum chemistry [3] or solving combinatorial problems [4]. Even though the encountered practical constraints impose a tall hurdle, those efforts appear promising for near-future applications. Such hopes are furthermore fueled by the achievements in the field of deep learning, especially during the last decade.

Despite the absence of rigorous performance guarantees, there has been a tremendous success of deep learning methods in diverse fields ranging from computer vision, natural language processing to finance and beyond [5].

Over the last year, rigorous performance guarantees for machine-learning-based approaches to quantum many-body physics have been found [6–8]. These findings suggest that machine learning algorithms are well suitable to generalize efficiently on quantum data that is obtained by quantum experiments or a quantum simulation. In particular, with the recent development in hybrid quantum algorithms such as the variational quantum eigensolver (VQE) [3, 9], variational methods become interesting, viable experimental alternatives. Alterations to the originally proposed scheme allow for the study of a few eigenvalues and -states around a target energy [10] which does not need to be the ground state [11]. The VQE’s setting suits the study of MBL quite well [12].

To demonstrate the importance of the quantum data, difficult problems from quantum physics are needed. These problems are rendered as such because of their evasive behavior under analytical or numerical analyses. One of such notoriously difficult problems is the phenomenon of localization in interacting quantum many-body systems, known as MBL [13–15], see e.g. Refs. [16–18] for reviews. It originates from the well-known Anderson model of non-interacting fermions in a disordered potential where localization occurs above a certain disorder threshold [19]. The seminal works [13, 20] proved the sur-

FIG. 1. Workflow for training our model architecture to predict indicator values $\hat{y}$ from the system’s disorder vector $h$. We pass the latter into a recurrent neural network as in Fig. 2 which extracts general features of $h$ in a scalable fashion. These features can be augmented by the respective energy density $\epsilon$ we are considering. Together, they are fed into a fully-connected neural network that maps them to $\hat{y}$. They are compared to the results $y$ obtained from exactly diagonalizing the system’s Hamiltonian in the corresponding energy density $\epsilon$. 

* alexander.gresch@hhu.de
vival of the localization under the introduction of a weak interaction in terms of a perturbation. This localization can be pinpointed to the emergence of macroscopically many conserved quantities [16, 21–24] that suppress the flow of correlations through the system. In the regime of strong interactions (or conversely, a negligible disordered potential), MBL does not occur which indicates a phase transition between the MBL phase and the delocalized one. The latter can be explored deploying e.g. classically motivated ergodic arguments [25]. However, little is known about the transition region between the two phases and its underlying mechanism. The emergence of MBL connects to the fundamental question of thermalization in quantum mechanics [26–28], possibly bridged by the eigenstate thermalization hypothesis (ETH) [16, 29]. Numerical studies of the transition either apply exact diagonalization [14] or approximate methods using either shift-invert diagonalization [30] or renormalization group techniques [31]. Around the presumed transition region between the two phases, the numerical methods suffer from the curse of dimensionality because the Hilbert space dimension grows exponentially with the chain length \( L \). Moreover, a numerical extrapolation to the thermodynamic limit at which the transition is expected to be characterized by a single value for the critical disorder parameter \( h_c \) is hampered by finite-size effects [32].

### A. Related works

The idea of applying neural networks (NNs) to physical problems and, in particular, phase classification, arises as a consequence of its success with feature extraction e.g. for conventional image classification, where the classifiers could achieve a higher prediction accuracy than human test groups [33]. It has led to a surge of explorations in applying similar methods to difficult problems in (quantum) many-body physics [34–37]. The phenomenon of MBL, in particular, has attracted many numerical approaches using machine learning [38–40] or deep learning [41–44]. The previous attempts typically utilized NNs for the phase classification in order to extract a phase diagram of the transition in an energy-density- and disorder-parameter-resolved way. Employing a recurrent neural network (RNN) to study the behavior of MBL was – to the best of our knowledge – first accomplished by Ref. [42] who trace the temporal evolution of an observable as a phase classification task. In variation to those approaches, we propose to employ an RNN to characterize a given instance of the Hamiltonian’s components in terms of quantum data. For the characterization, there has been an explorative work done by Nieuwenburg, Baum, and Refael [14] in the same direction. They show the learnability of the adjacent gap ratio by means of convolutional NNs from the disorder vector joined with the corresponding disorder parameter, i.e. from \( \mathbf{h} \oplus h \) [44, Appendix]. Their efforts, however, resort to a proof-of-principle demonstration and use it for data augmentation. Moreover, their architecture is not scalable in the system size \( L \) because the output size of the convolutional layers grows linearly with \( L \). Such convolutional layers can be made scalable with the input size as demonstrated by Saraceni, Cantori, and Pilati [45]. They propose an architecture where the number of extracted features does not grow with the input size and can thus be mapped to a fixed output size. Apart from this last instance, all the previous methods are restricted to a given, fixed chain length and therefore not applicable to data from a larger system. Another bottleneck is the fact that the typical input for these approaches consists of heavily preprocessed data such as the entanglement spectrum [41] or even a whole eigenvector of the Hamiltonian [43]. Both are obtained by exact diagonalization and thus lack a feasible source of training data from the transition regime for system sizes \( L \gtrsim 20 \).

### B. Our contribution

In this work, we propose an NN-architecture that is both applicable to data from different system sizes and not necessitating any computationally costly preprocessing of the input data. We accomplish this by directly presenting the local disorder values \( \mathbf{h} = (h_1, \ldots, h_L) \) to an RNN. This step lifts the system size constraint by treating \( \mathbf{h} \) as a sequence of inputs such that the sequence length corresponds to the system size. The output of the RNN serves as the extracted feature vector from the disorder sequence. Typically, such features do not yet resemble the indicators. Rather, they are global properties of the input which are not tied to a specific regression task. This view is adapted from results in computer vision where the first layers of image classifying networks merely detect edges and corners, independent of the underlying classification problem [46]. Hence, we use a final fully-connected NN as sketched in Fig. 1 that maps the extracted features to the indicator estimates. With this choice for our architecture, we can investigate in the features further by means of transfer learning [47]. To this end, we show that a set of features extracted from some indicators can be generalized to other previously unseen indicators. Moreover, we show that we can achieve this goal with only two features of the input without a significant drop in performance. Finally, we demonstrate the efficiency of our architecture to enhance the resolution of the phase diagram of the test data set. We achieve this because our trained network is capable of predicting the indicator values for various choices of the energy density \( \epsilon \) and disorder parameter \( h \) at once.

We emphasize that this NN-based approach to the phenomenon of MBL differs from previous attempts drastically. Previously, NNs have been used for the classification task of preprocessed inputs [41–44]. Such an ansatz depends completely on the availability of the preprocessed input. We take a step further and demonstrate that distinctive signatures of MBL, encoded in the indicator values, are directly learnable from a given disorder
realization in a spin chain. That is, we only enter the defining values of the Hamiltonian and regard the processed indicators as targets, not as inputs to our NN. We obtain these estimates for each disorder realization and for various energy densities at once, i.e. we do not require any averages beforehand.

C. Outline

In the next Section II A, we introduce artificial NNs and in particular our model architecture that is based on a recurrent variant. We proceed by introducing the quantum many-body system of interest for the study of MBL in Section II B. As a test bed for our set-up, this will be the disordered Heisenberg spin chain. To this end, we present prominent indicators of MBL and their behavior in each of the two phases. In Section III A, we demonstrate the scalability of our architecture to predict data for system sizes beyond the training set. This includes a quantitative benchmark of the quality of the network’s output. As the next step, we emphasize in Section III B by the means of transfer learning that the relevant global features of the input are recognized. Moreover, this hints towards a compatibility between the various indicators which is understood in the study of Anderson localization but remains unclear for MBL. Lastly, we show the numerical efficiency of our method in Section III C to obtain a high-resolution phase diagram of the MBL-transition. We complement our work with a summary and an outlook for future directions in Section IV.

II. PRELIMINARIES

In the following, we start with providing the required background of RNNs, accompanied by a physical model featuring MBL, the Heisenberg spin chain.

A. Recurrent artificial neural networks

We use artificial NNs and in particular their recurrent variant (RNN). NNs are loosely inspired by their biological counterpart in the human brain. Effectively, they serve as a black-box approach to a universal function approximator. They are modularly built by so-called parameterized layers, usually of the form $y_t = \sigma(W_i y_{t-1} + b_i)$ where the parameters of the $l$-th layer $(W_i, b_i)$ are called weights and biases, respectively. The linearity is broken by a so-called activation function $\sigma$ which is a non-linear function, usually applied element-wise to its argument. This way, a predefined type of input $x =: y_0$ is processed layer by layer. This is referred to as the feed-forward pass of the NN. As a consequence, we can consider the NN as a parameterized black-box function $f_\theta(x) = \hat{y}$ with parameters $\theta$ given by the weights and biases. In the supervised learning setting, the input $x$ is tied to a target value $y$ of which $\hat{y}$ is an estimation. The quality of the estimation is quantifiable by the so-called loss function. Its gradient with respect to the network’s parameters $\theta$ can be computed efficiently by the method of backpropagation. It is used in an update rule, such as gradient descent, for the parameters to iteratively find a set of parameters that minimizes the loss [46].

The key limitation of the plain-vanilla NN is the restriction in the fixed input shape. RNNs have a special architecture that allows e.g. for an arbitrary input and output length. This feature is heavily utilized in the field of natural language processing. The recurrent behavior of a layer is achieved by the introduction of a hidden state $\mathcal{H}$. To this end, we regard the input $x = (x_1, x_2, \ldots, x_T)$ as a sequence of $T$ individual inputs. The hidden state can be repeatedly updated according to the network’s parameters $\theta$ and the current input, i.e. $\mathcal{H}_t = \mathcal{H}_t(\theta, \mathcal{H}_{t-1})$ with $t = 1, \ldots, T$. Importantly, the same parameters $\theta$ are used for every update of the hidden state. The final hidden state $\mathcal{H}_T$ serves as the output of the recurrent layer. A schematic is shown in Fig. 2.

B. The model for MBL

A common model often consulted on for the study of MBL is the one-dimensional Heisenberg spin chain of length $L$ whose Hamiltonian reads as

$$H = J \sum_{i=1}^{L} \sum_{\alpha \in \{x,y,z\}} \sigma^{(i)}_{\alpha} \sigma^{(i+1)}_{\alpha} + \sum_{i=1}^{L} h_i \sigma^{(i)}_z, \quad (1)$$

where $\sigma^{(i)}_{x/y/z}$ denotes the respective Pauli matrix acting on the $i$-th site. We work with periodic boundary conditions, i.e. $\sigma^{(L+1)}_{x/y/z} \equiv \sigma^{(1)}_{x/y/z}$. The parameters $h = (h_1, \ldots, h_L)$ are the local disorder strengths which are sampled independently from a uniform distribution over the interval $h_i \in [-h, h]$ for each site $i$. The variable $h$ is called

\begin{equation}
\begin{array}{c}
\mathcal{H}_T \\
\uparrow \\
\{x_t\}
\end{array}
\quad \equiv \quad
\begin{array}{c}
\mathcal{H}_0 \\
\uparrow \\
x_1
\end{array}
\quad \rightarrow 
\begin{array}{c}
\mathcal{H}_1 \\
\uparrow \\
x_2
\end{array}
\quad \rightarrow \cdots
\begin{array}{c}
\mathcal{H}_T \\
\uparrow \\
x_T
\end{array}
\end{equation}

FIG. 2. Scheme of an RNN cell as used in Fig. 1. On the left, the cell is shown as a black-box that iterates over an input sequence $\{x_t\}$ and produces an output state $\mathcal{H}_T$. Unfolding the cell results in the scheme on the right. An initial hidden state $\mathcal{H}_0$ is evolved over $T$ time steps during which the sequence elements are fed into the network one after another. The final evolved hidden state is released as the network’s output. Each box on the right corresponds to the same cell architecture, i.e. having the same weights and biases for each time step. The recurrent cell can process inputs of arbitrary sequence lengths $T$. 
Another prominent measure is the averaged over all eigenvalues close to a target energy variant in this limit. On the other hand, i.e. for \( h \rightarrow \infty \) the interaction term is negligible, and we recover the localization behavior of the Anderson model. In between these limits, a phase transition from the delocalized phase to the many-body localized one is therefore assumed. Numerical studies report an estimation of the critical disorder parameter \( h_c \) of \( h_c \approx 6^1 \), which has an additional slight dependence on the considered energy density \( \epsilon(E) := (E - E_{\text{min}})/(E_{\text{max}} - E_{\text{min}}) \) [15]. This numerically observed so-called mobility edge is debated from theoretical grounds and attributed to finite-size effects [24].

There are several properties of the two phases which are shared with the Anderson metal-insulator transition. Such properties like the system’s entanglement or its spectral statistics are typically aimed to be summarized by a single real number. Since it varies in its numerical value from one phase to the other, it is referred to as an indicator for many-body localization. This is not an order parameter as there exists no mean-field theory for MBL [18]. Indicators can be divided into three groups of origin: (i) spectral indicators (function of the eigenvalues), (ii) functions of the eigenvectors (e.g. entanglement entropies), and (iii) time-averaged observables after a quench. As one example for a spectral indicator, it is known that the distribution of the spectral gaps of the Hamiltonian varies between the two phases. In particular, for \( h \rightarrow 0 \) the gaps are distributed according to the Wigner-Dyson distribution whereas the distribution is Poissonian in the MBL phase [13]. These two limiting cases are incorporated by the adjacent gap ratio \( (\ell) \). This ratio can be computed for the \( i \)-th spectral gap \( \delta_i = E_{i+1} - E_i \geq 0 \) as

\[
\ell_i := \frac{\min\{\delta_{i+1}, \delta_i\}}{\max\{\delta_{i+1}, \delta_i\}}.
\]

Averaging over all eigenvalues close to a target energy density and over different disorder realizations yields \( \langle \ell \rangle_{\text{deloc}} \approx 0.53 \) in the delocalized limit and \( \langle \ell \rangle_{\text{MBL}} = 2 \ln(2) - 1 \approx 0.39 \) in the MBL phase when \( h \rightarrow \infty \).

Localization is not only traceable by spectral statistics. Another prominent measure is the half-chain entanglement entropy [14]. To this end, we split the chain in half and calculate the reduced density matrix of the first half \( \rho_A := \text{Tr}_B[\rho_{AB}] \) by tracing out the second half of the joint density matrix \( \rho_{AB} \). The density operator is constructed for each eigenstate \( |n\rangle \) of the Hamiltonian, i.e. \( \rho_{AB} = |n\rangle\langle n| \). The entanglement entropy \( (S_A) \) is given by computing

\[
S_A := \text{Tr}[\rho_A \ln(\rho_A)]
\]

and averaging again over eigenstates and disorder realizations. We normalize this quantity with the expected maximal half-chain entropy which is the Page entropy [48]. In this way, the indicator varies from 1 in the delocalized regime to approaching 0 in the MBL phase as entanglement is suppressed by the local disorder. Moreover, we note a volume-law scaling of the entanglement entropy with respect to the system size in the delocalized phase but only an area-law scaling in the localized regime [49].

In addition, the eigenstates carry information about the transport behavior of the spin which is a global conserved quantity. The dynamical spin fraction \( (\mathcal{F}) \) quantifies the degree of relaxation of an initial inhomogeneous spin density [14]. It is given as

\[
\mathcal{F} := 1 - \frac{\langle M^\dagger M \rangle}{\langle M^\dagger \rangle \langle M \rangle}
\]

with \( M = \sum_{j=1}^L \sigma_j^{(j)} \exp\left(2\pi i \frac{j - 1}{L}\right) \)

where the expectation value is taken for all eigenstates close to a target energy. Again, we average \( \mathcal{F} \) over many disorder realizations. The persistent spin inhomogeneity in the MBL phase means that \( \langle \mathcal{F} \rangle \rightarrow 0 \) whereas in the delocalized regime \( \langle \mathcal{F} \rangle \rightarrow 1 \).

III. RESULTS

In this work, we report on a highly flexible deep learning architecture whose workflow we depict in Fig. 1 that learns the quantum data obtained from an experiment or a numerical study. In this way, predictions can be made for single instances at various energy levels at once, and we do not need any averages over input configurations. Moreover, the set-up lifts the restriction of a fixed system size for the available quantum data and only requires the relevant parameters of the underlying Hamiltonian. We demonstrate that the set-up extracts global, i.e. task-independent features from the input which makes it applicable to predicting a broad class of quantum data. Thus, our approximation scheme serves as a computationally cheap alternative to demanding numerical methods such as exact diagonalization. We emphasize that, in a broader sense, our method is not limited to the study of MBL but applicable to many more problems in quantum many-body physics.

---

1 Due to our definition of Eq. (1) via Pauli matrices, the critical value is twice as large as typically reported in the literature.
A. Scalable indicator approximation

Over the last two decades of approaching Anderson localization analytically and subsequently MBL mostly numerically, several properties of the phenomenon have been demonstrated to be summarized by the aid of the aforementioned indicators. We demonstrate that they can be approximated efficiently by an NN. Intuitively, this comes as no surprise for the indicator values are functions of the Hamiltonian’s parameters which are taken as the input of the NN. The defining parameters of the Hamiltonian (1) are the local disorder values \( h = (h_1, \ldots, h_L) \) as we consider isotropic nearest-neighbor interactions of relative unit strength. As we explain later in Section III C, our architecture is capable of estimating the indicators for various values of the energy density \( \epsilon \) at once. For now, however, we restrict ourselves to the infinite temperature regime, i.e. with \( \epsilon = 0.5 \) fixed. In order to accommodate disorder vectors of different lengths, we use an RNN architecture that treats the disorder vector as a sequence of the local disorder values. RNNs have specifically been designed to handle variable sequence lengths by virtue of their recursive design, see Fig. 2 and further details in Appendix A. As loss function, we choose the mean-squared-error (MSE) between the obtained estimations of the RNN and the actual values obtained by exact diagonalization of the Hamiltonian. As a framework for setting up the NNs and its training, we rely on PyTorch [50]. We publish our data and the code for performing the training of the NNs and for creating all here presented plots online [51].

Figure 3 shows a plot of the learned indicator statistics for \( L = 14 \) where the network has been trained on data from chain lengths \( L = 10, 12 \). We interleave the plotting of the underlying target data with the corresponding output from the NN. For various values of the disorder parameter \( h \), we sampled disorder vectors that make up different Hamiltonians. For each of these, we obtained the vector of indicator values \( y \) from Section II B via exact diagonalization. Each of the disorder vectors was fed into our NN to output an estimation \( \hat{y} \) of \( y \). In the plot, we show the mean and the standard deviation (that results from different realizations of the disorder vector sampled with the same disorder parameter \( h \)) of \( y \) and \( \hat{y} \), respectively. Especially the entanglement entropy \( S_A \) (3) and the dynamical spin fraction \( F \) (4) show a good agreement up to the second moment of the data distribution. For the adjacent gap ratio \( r \) (2), only the mean is well-approximated which indicates that the dependence of \( r \) on the level of the particular disorder realization may be harder to learn. Importantly, we demonstrate that our NN-architecture can be queried on data belonging to an arbitrary chain length \( L \). Here, we have trained on smaller system sizes and find a qualitative agreement for the larger system size, \( L = 14 \), in the plot.

Additionally, we can quantitatively benchmark the performance of our network using the coefficient of determination \( R^2 \). It is used as a benchmarking tool in linear regression and is defined as

\[
R^2 := 1 - \frac{\sum_i (f(x_i) - y_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{\text{MSE}[f(X), Y]}{\text{Var}[Y]},
\]

where the sum runs over all data point pairs \( \{(x_i, y_i)\} \) in the test set, the mean over the targets \( \bar{y} \) is denoted by \( \hat{y} \), \( f \) represents the NN and \( \text{Var}[Y] \) denotes the variance of \( Y \). So, it essentially compares the MSE of the network outputs with the variance in the data. For a non-linear function \( f \) the second term on the right-hand-side is unbounded from above and the corresponding \( R^2 \) value will lie in the interval \((-\infty, 1]\) which is unwanted for a squared expression. The coefficient of determination (5) can be transformed to a non-negative number by introducing \( R_{\text{norm}}^2 := 1/(2 - R^2) \in [0, 1] \) [52]. Here, \( R_{\text{norm}}^2 = 1 \) means an approximation being exact and \( 1/2 \) constitutes a baseline value, which is attained for \( f \) being the constant function that outputs the target mean. We calculate the normalized coefficient indicator-wise for each value of the disorder parameter \( h \).

The result for the same energy density as in Fig. 3 is presented in Fig. 4. We emphasize that the network has not encountered any training data from the largest system size, \( L = 14 \). Yet, it is qualitatively able to estimate values beyond its training set system sizes. This quantitative observation corroborates our first qualitative one in Fig. 3. Since the entanglement entropy and the dynamical spin fraction have been well-matched, we see a large value of \( R_{\text{norm}}^2 \), for values \( h \gtrsim 3 \) accordingly. The breakdown for disorder parameter values below that can be attributed to the vanishing variance in the test set for \( h \to 0 \) due to the vanishing disorder in the Hamiltonian. As a consequence, it does not pose a threat to our set-up as it could easily be circumvented by weighting the corresponding training data accordingly. As we have seen already, the adjacent
Disorder parameter $h$

Wigner-Dyson distribution from random matrix theory, which is a consequence of the vanishing disorder in the system, and which we call the multitask network. A quantitative comparison using the normalized coefficient of determination (5) is given in Fig. 5. The transferred features lead to features of the input that are task-independent, the prediction quality should be comparable to the case where the RNN actually extracts general features. Since such tasks are detached from the actual classification task, the first layers are said to detect task-unspecific, general features of the input and thus regarded as feature extractors. Only the last layers of a (deep) NN map these extracted features to the specific problem at hand.

In this section, we inspect whether such a behavior is exhibited by our proposed model. We approach this question with the aid of transfer learning [47]. The idea is, assuming that the RNN actually extracts general features of the disorder vector $h$, to keep the RNN fixed after we have trained it on a set of MBL indicators. We can now switch the targets in the training set, i.e. exchange the target indicators with some new indicators which the network has not encountered before. As the RNN-output is detached from the choice of the target indicators, we only retrain the NN that maps the features to the newly chosen indicators. If the output of the RNN corresponds to features of the input that are task-independent, the prediction quality should be comparable to the case where we retrain the full model from scratch on the new data.

We select the dynamical spin fraction $F$ (4) as the transfer target indicator. To this end, we train our model on the adjacent gap ratio $r$ (2) and on the entanglement entropy $S_A$ (3) for system sizes $L = 10, 12$. Thus, we exclude $F$ explicitly from the training set. Once the training succeeds, we keep the RNN’s parameters fixed and only retrain the subsequent NN to predict the spin fraction given the output of the RNN. We benchmark the prediction quality with a model of the same architecture that is trained to predict only $F$ from scratch. Furthermore, we compare both predictions with the previous model from Fig. 4 that has been trained on all three indicators at once and which we call the multitask network. A quantitative comparison using the normalized coefficient of determination (5) is given in Fig. 5. The transferred features lead to a comparable performance as a model that is retrained of benchmarking our approach is the generation of the training set (due to the cost intensity of the exact diagonalization), we are interested in how the network copes with a shrunken training data set. We refer to Appendix B for the analysis and plots. In essence, we find that we can shrink the training data set if we allow for more training epochs in return. This way, we can reduce the training data set down to a number close to the number of trainable parameters in the network. These observations are crucial for obtaining a data set from an actual experiment in the future where determining indicator values for even a single realization might be expensive.

**B. Transfer learning**

The common notion in deep learning is that there exists a hierarchy of abstraction in what the different layers of an NN are capable of identifying. This view has been corroborated by inspecting the first layers of state-of-the-art image classifiers which correspond to edge and corner detection [46]. We select the dynamical spin fraction $F$ (4) as the transfer target indicator. To this end, we train our model on the adjacent gap ratio $r$ (2) and on the entanglement entropy $S_A$ (3) for system sizes $L = 10, 12$. Thus, we exclude $F$ explicitly from the training set. Once the training succeeds, we keep the RNN’s parameters fixed and only retrain the subsequent NN to predict the spin fraction given the output of the RNN. We benchmark the prediction quality with a model of the same architecture that is trained to predict only $F$ from scratch. Furthermore, we compare both predictions with the previous model from Fig. 4 that has been trained on all three indicators at once and which we call the multitask network. A quantitative comparison using the normalized coefficient of determination (5) is given in Fig. 5. The transferred features lead to a comparable performance as a model that is retrained.

![Graph of $R^2_{\text{norm.}}$ for each MBL indicator as a function of the disorder parameter $h$ at an energy density $\epsilon = 0.5$. The graph shows the normalized coefficient of determination for different system sizes $L = 10, 12, 14$.](image)

![Graph of the dynamical spin fraction $F$ and the adjacent gap ratio $r$ as functions of the disorder parameter $h$.](image)
Additionally, the performance of the these two networks is very similar to the multitask network. The differences between any two curves is due to statistical errors. We find a similar situation when selecting the adjacent gap ratio or the entanglement entropy as the transfer target indicator, respectively (data not shown). We can attribute the congruence of all three different types of training to the following two reasons. First, there appears no qualitative difference in the learnability of each of the indicators. Moreover, they seem to be compatible with each other in the sense that they can all be obtained from the same features. In our case, we are able to apply the transfer learning scheme using only two features. We provide more details in Appendix A. This indicates that the extracted features are general enough to allow for the estimation of a variety of indicators which, in turn, do not rely on a specific set of features produced during a specific training procedure.

C. Energy dependency

Lastly, we demonstrate that predictions from our trained estimator recover the results from previous numerical studies of MBL in the limit of averaging over many disorder realizations. Namely, we recover the phase diagram of the transition for various chain lengths \( L \) that show the indicator values in dependence of the considered disorder parameter \( h \) and energy density \( \epsilon \). To this end, we can generate predictions of unseen trial disorder realizations, i.e., random instances of disorder vectors for a given chain length and disorder parameter. These instances are fed into our NN to accumulate a trial data set for various energy densities \( \epsilon \) at once. The latter is straight-forwardly incorporated by augmenting the output of the RNN by the corresponding value for \( \epsilon \). Since we solely focus on the network’s prediction, we do not need to perform the exact diagonalization procedure for these new instances. Therefore, generating this large data set is efficient in the system size. The resulting phase diagram for the dynamical spin fraction \( F \) is presented in Fig. 6.

Most importantly, we are now able to generate images of the phase diagram to an arbitrary resolution with numerical efficacy. Moreover, we are not limited by the initial resolution in the training data. This is because we only require forward passes through the NN which scales both linearly in the number of queried values for both the disorder parameter and the energy density. We provide further insights in Appendix C.

IV. CONCLUSION AND OUTLOOK

We have constructed a RNN architecture that approximates values for certain indicators for MBL directly from the variable part of the Hamiltonian, i.e., the local disorder strengths. The recurrent set-up ensures that the network can process data for an arbitrary system size \( L \) and produce a good estimation output provided the trial system size is not too far off the training set. Moreover, our approach does not require any further computationally expensive preprocessing of the input data. In this way, we are able to characterize single disorder realizations by providing the corresponding indicator values. By inspecting
the intermediate features of the RNN by means of transfer learning, we observe that all considered indicators can be derived from two features alone. Furthermore, they serve as an archetype for various indicators at arbitrary energy densities at once. This enables us to study the transition region by means of phase diagrams that can be rendered to an arbitrary resolution.

**Outlook**

With a training set that consists of indicator sets from different system sizes, we envision an interplay between an actual experiment and our architecture. The experiment can address systems consisting of dozens of spins or qubits. Thus, it delivers the training set for the architecture beyond what is reachable by exact diagonalization studies. As we demonstrated, our architecture is not inclined to a specific data type. Thus, the experiment is not restricted to a certain indicator but can provide the most amenable one (such as the growth of the entanglement entropy [53] or the imbalance after a quench [53–56]) for the training set. Motivated by our findings in Section III A, we conjecture that only a few realizations per disorder parameter are sufficient as to merely guide the extrapolation. In addition, the indicators are expected to become more and more pronounced in their respective shape. Therefore, we do not expect large deviations from the case of smaller system sizes up to finite-size effects. The whole premise of transfer learning relies on the assumption that the additional data for a larger system size only serves as a guidance for the overall learned structure on the training set. This boosts training the NN significantly [47]. Given experimental training input, the network can in turn provide estimates for data outside of or in between gaps in the training set which can be benchmarked by the experiment in return [57, 58]. Other possibilities of enriching the training set is to resort to numerical approximations, for example by tensor networks methods which are well-suited deep within the MBL phase [59] or yet another NN architecture to even speed up those methods [60]. With the data at hand, a more detailed examination of the compability of different indicators allows to shed some light on their yet unknown coaction towards MBL. Diving deeper into the interpretation of the archetypical feature and the compatibility of various indicators is an interesting research direction for future works.

Our proposed scheme aims to bring together the often independent advances in experiments and numerics, and we see possible research directions in the now scalable phase classification task and a better understanding of the learning process of the recurrent feature extractor. Furthermore, the connection of our method with a VQA is of broader interest ranging from applications in condensed matter and statistical physics to the field of (hybrid) quantum computation or quantum machine learning. Compared to the existing traditional numerical methods, the interplay of a quantum experiment or its simulation with our method may constitute a new type of quantum advantage in the sense that we can obtain an efficient classical method only via accessing a quantum data set. Such a pairing provides a potentially powerful computational tool that is yet to be augmented with experimental data in the future.

**ACKNOWLEDGMENTS**

We thank Christian Gogolin for fruitful discussions. Computational support and infrastructure was provided by the “Centre for Information and Media Technology” (ZIM) at the University of Düsseldorf (Germany). This work has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the grant number 441423094 within the Emmy Noether Program.
APPENDIX

In this appendix, we provide more details on our network architecture and the training procedure. Starting with Appendix A, we describe the generation of the data sets and detail the architecture of our approach. In Appendix B, we examine the network’s performance under a shrinking data set size. Finally, we give some more comments on the obtained phase diagram in Section III C and its analysis in Appendix C.

A. Details on the network architecture and the training procedure

We briefly describe how we set up the training and the test set as well as the network architecture used for the results in the main text. We set up a grid for the disorder parameter \( h \), i.e. we chose 30 values \( h = 0.5, 1, 1.5, \ldots, 15 \) which lie well around the assumed critical disorder parameter value of \( h_c \approx 6 \). For each chain length \( L = 10, 12, 14 \) we have sampled disorder vectors \( h \) with entries \( h_i \) independently and identically distributed from the uniform distribution, such that \( h_i \in [-h, h] \) for a given disorder parameter value \( h \). For each \( h \) and \( L \), this was done \( N_{\text{train}} = 1000 \) and \( N_{\text{test}} = 100 \) times for the two data sets, respectively. Each of these disorder vectors yields a realization of the Hamiltonian \( (1) \). Its eigenvalues and eigenvectors were found via exact diagonalization. We have chosen a grid of \( N_e = 19 \) energy densities \( \epsilon = 0.05, 0.1, 0.15, \ldots, 0.95 \) and have kept the 100 next closest eigenvalues and their corresponding eigenvectors for calculating the three indicators from Section II B.

The architecture of our proposed network scheme is summarized in Fig. 7 and we explain its choice in the following. The first part consists of an RNN-cell that serves as a feature extractor of the input. The RNN is presented each disorder parameter \( h_i \), successively and updates its hidden state according to its parameters and the value of \( h_i \). The hidden state was initialized as zero. After having fed in \( h_{L} \), the final updated hidden state is released as the output of the RNN. We treat this output as the feature vector of the disorder vector. Due to this recursive procedure, RNNs can be unstable during training because of exploding or vanishing gradients in the optimization procedure. In order to circumvent this problem, the long short-term memory (LSTM) cell [61] and the gated recurrent unit (GRU) [62] have been proposed with competing performance-efficiency trade-offs [63]. We find the latter to be slightly better in performance during training. Concerning the number of output features of the RNN, we find qualitative good results when choosing a feature dimension of 2. A larger dimensionality does increase the performance of the indicator approximation, however, we observe a severely decreased performance when applying the transfer learning scheme from Fig. 5. We have only used a single RNN cell of depth one. Lastly, we have performed a computationally inexpensive preprocessing of the disorder vector. We regroup the elements of the disorder vector in pairs of two, i.e. transform according to \( \{h_1, h_2, h_3, \ldots, h_L\} \mapsto \{(h_1, h_2), (h_2, h_3), \ldots, (h_{L-1}, h_L), (h_L, h_1)\} \). Regroupings into even larger tuples are also possible. The pairing in two, however, fits in well with the nearest-neighbor interactions and the periodic boundary condition and, furthermore, leads to the best performance. Afterwards, the feature vector is augmented by the value for the energy density \( \epsilon \) under consideration. Together, we map them to the three indicator values by a fully-connected NN of hidden size 10. As the loss we choose the mean-squared-error (MSE) and train the model for \( N_{\text{epochs}} = 15 \) on the training data. We use the Adam optimizer with default values [64], a batch-size of 128 and a learning rate \( \eta = 10^{-3} \).

For the transfer learning scheme of Section III B and for creating the model that is capable of dealing with an arbitrary energy density \( \epsilon \) in Section III C, the training consists of two stages: we first proceed as outlined above. This pretraining is necessary to facilitate an easier focussed training of the RNN to extract meaningful features which we show in Fig. 8. Then, we fix the parameters of the RNN and thus the intermediate features, and train the subsequent fully-connected NN on the full training data for 30 more epochs with a decreased learning rate of \( 10^{-4} \) following the Adam optimizer routine. This fine-tuning of the NN yields a greater performance compared to training the two components of the model jointly. The choice for the hyperparameters (architecture of the two individual components, feature size, number of hidden neurons and the optimizer parameters) above has been determined on a held-out validation data set.
we expect a better performance with a larger training data set for each training reinitialization. We done so by always picking a random subset of the full $N_{\text{train}}$ and the test set after convergence. For each value of $N_{\text{train}}$, we reinitialize and train the model ten times. In particular, this has resulted in very long training loops for a small $N_{\text{train}}$ as we have trained for several hundreds of epochs. Due to the mini-batching during training, we track the actual number of received update steps during training for various values of $N_{\text{epoch}}$ and exclude the system size of $L = 14$ from the training set. We set a value of $N_{\text{epoch}} = 30$ for training on the largest data set size with $N_{\text{train}} = 100$ and adjusted that value accordingly for smaller sizes. In all considered cases, this leads to a convergence of the models and we extract the remaining average MSE on both the training and the test set after convergence. For each value of $N_{\text{train}}$, we reinitialize and train the model ten times. In all cases, when we decreased the training set, we have done so by always picking a random subset of the full training data set for each training reinitialization.

In particular, this has resulted in very long training loops for a small $N_{\text{train}}$ as we have trained for several hundreds of epochs. Due to the mini-batching during training, we track the actual number of received update steps during training for various values of $N_{\text{epoch}}$ and exclude the system size of $L = 14$ from the training set. We set a value of $N_{\text{epoch}} = 30$ for training on the largest data set size with $N_{\text{train}} = 100$ and adjusted that value accordingly for smaller sizes. In all considered cases, this leads to a convergence of the models and we extract the remaining average MSE on both the training and the test set after convergence. For each value of $N_{\text{train}}$, we reinitialize and train the model ten times. In all cases, when we decreased the training set, we have done so by always picking a random subset of the full training data set for each training reinitialization. We show the two averaged losses in Fig. 9. This reveals that shrinking the training set down to $N_{\text{train}} \approx 3$ (this corresponds to a total number of training points of around 180) yields no qualitative increase of neither of the two losses after training. This threshold is of the order or trainable parameters of the model (cf. Fig. 7). Below it, we observe a decreased training loss while the test loss is increased. In this limit of scarce data, the model begins to overfit the training data at the expense of a larger loss on the test set. This small number is encouraging for the model application to data that stems from an actual experiment as we have to repeat the same experiment only a handful of times for each point in the phase diagram we are interested in. This highlights the feasibility of our approach to actual data stemming from a quantum experiment.

In this section, we provide details on the results of Section IIIA. In particular, we investigate the performance dependence on the size of the training data set. We can test this quantitatively by decreasing the number of samples per disorder parameter $N_{\text{train}}$. In this setting, half a value in $N_{\text{train}}$ corresponds to a two-fold reduction in the training set size. If we were to train now for a fixed number of epochs $N_{\text{epoch}}$, that is, until the network encountered each data point $N_{\text{epoch}}$ times during training, we expect a better performance with a larger $N_{\text{train}}$. In this case, the network receives more update iterations to minimize the MSE objective, hence the performance gain. For a fairer comparison, we track both the training and the test loss during training after each update step. Hence, the total number of iteration steps is to be made a constant, i.e. on a training set of twice the size we allow the network to train for half the epochs. In this setting, each training run allows the NN the same total amount of update steps.

In particular, this has resulted in very long training loops for a small $N_{\text{train}}$ as we have trained for several hundreds of epochs. Due to the mini-batching during training, we track the actual number of received update steps during training for various values of $N_{\text{epoch}}$ and exclude the system size of $L = 14$ from the training set. We set a value of $N_{\text{epoch}} = 30$ for training on the largest data set size with $N_{\text{train}} = 100$ and adjusted that value accordingly for smaller sizes. In all considered cases, this leads to a convergence of the models and we extract the remaining average MSE on both the training and the test set after convergence. For each value of $N_{\text{train}}$, we reinitialize and train the model ten times. In all cases, when we decreased the training set, we have done so by always picking a random subset of the full training data set for each training reinitialization. We show the two averaged losses in Fig. 9. This reveals that shrinking the training set down to $N_{\text{train}} \approx 3$ (this corresponds to a total number of training points of around 180) yields no qualitative increase of neither of the two losses after training. This threshold is of the order or trainable parameters of the model (cf. Fig. 7). Below it, we observe a decreased training loss while the test loss is increased. In this limit of scarce data, the model begins to overfit the training data at the expense of a larger loss on the test set. This small number is encouraging for the model application to data that stems from an actual experiment as we have to repeat the same experiment only a handful of times for each point in the phase diagram we are interested in. This highlights the feasibility of our approach to actual data stemming from a quantum experiment.

In particular, this has resulted in very long training loops for a small $N_{\text{train}}$ as we have trained for several hundreds of epochs. Due to the mini-batching during training, we track the actual number of received update steps during training for various values of $N_{\text{epoch}}$ and exclude the system size of $L = 14$ from the training set. We set a value of $N_{\text{epoch}} = 30$ for training on the largest data set size with $N_{\text{train}} = 100$ and adjusted that value accordingly for smaller sizes. In all considered cases, this leads to a convergence of the models and we extract the remaining average MSE on both the training and the test set after convergence. For each value of $N_{\text{train}}$, we reinitialize and train the model ten times. In all cases, when we decreased the training set, we have done so by always picking a random subset of the full training data set for each training reinitialization. We show the two averaged losses in Fig. 9. This reveals that shrinking the training set down to $N_{\text{train}} \approx 3$ (this corresponds to a total number of training points of around 180) yields no qualitative increase of neither of the two losses after training. This threshold is of the order or trainable parameters of the model (cf. Fig. 7). Below it, we observe a decreased training loss while the test loss is increased. In this limit of scarce data, the model begins to overfit the training data at the expense of a larger loss on the test set. This small number is encouraging for the model application to data that stems from an actual experiment as we have to repeat the same experiment only a handful of times for each point in the phase diagram we are interested in. This highlights the feasibility of our approach to actual data stemming from a quantum experiment.
values for $\epsilon$ to our model. One initial alternative consists of various fully-connected NNs that are individually trained to predict the indicator values at a single $\epsilon$ each. While this, at first, has appeared beneficial with respect to the validation loss, there are a few drawbacks of this approach. The first one is the increased model complexity opposed to our scheme now. Here, we only require one single NN whereas the naive approach would require an NN for every $\epsilon$ of interest. Secondly, this approach limits the resolution of the prediction when in comes to obtaining the phase diagram in Fig. 6 as we require a data set for every $\epsilon$ of interest. Our approach circumvents both issues by the introduction of $\epsilon$ as an intermediate feature. This way, we can set up a much tighter grid for both $\epsilon$ as well as the disorder parameter $h$ and make predictions for each possible combination. To this end, we sample $N = 100$ new samples of disorder vectors $h$ for each $h$ and obtain the feature value by feeding it to the RNN. Then, we augment this value with every value of $\epsilon$ of interest and parse everything to the NN. Lastly, we average over $N$ and show this mean in dependence of $\epsilon$ and $h$ in the phase diagram. Since we only require forward passes through our model, this procedure is highly efficient: the run time is proportional to the chain length $L$ and to the number of queried values for both $h$ and $\epsilon$ and in that sense optimal.

We have also experimented with analysing the model’s predictions with a more quantitative measure such as the finite-size scaling analysis (FSSA) [65]. This method is aimed at mitigating the finite-size effects in the data and to obtain quantitative estimates of the critical disorder parameter $h_c$ and the critical exponent of the transition $\nu$. To this end, data from various chain lengths is given to the FSSA and fitted around the assumed value for $h_c$. We have tried to query our model at chain lengths beyond those in the training set, i.e. $L > 14$ but failed to reproduce previous approaches [15] as we have not observed signs of the $\epsilon$-dependent mobility edge in the transition. We attribute this observation to two different origins. First, we observe that the approximation is of higher quality around the transition region (cf. Fig. 4) and significantly so in the middle of the spectrum (at $\epsilon \approx 0.5$). The latter might leave a bias in the data at either side of the spectrum which is observed in the phase diagram. The second reason is due to our choice of the RNN architecture as feature extractor. In Fig. 8, we have shown the typical feature vector produced by the RNN after training. One important aspect is that there exists a cross-over point that is independent of the chain length $L$ of the input data but whose position depends on the initialization of the network parameters. This introduces a bias in the indicators since this cross-over is not apparent in the training data. We have tried to average the output over multiple retrainings (and therefore feature vectors) and by increasing the number of features but failed to lift this bias. However, we conjecture that with a more careful design of the RNN architecture, this is possible. In any case, the investigation of finding the right feature architecture is both interesting from a numerical and a theoretical perspective as it helps to shine some light on the nature of the MBL transition.

**ACRONYMS**

| Acronym     | Description                                      |
|-------------|--------------------------------------------------|
| FSSA        | finite-size scaling analysis                     |
| GRU         | gated recurrent unit                             |
| LSTM        | long short-term unit                              |
| MBL         | many-body localization                           |
| MSE         | mean-squared-error                               |
| NN          | neural network                                    |
| RNN         | recurrent neural network                         |
| VQA         | variational quantum algorithm                    |
| VQE         | variational quantum eigensolver                  |

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics 3, 625 (2021), arXiv:2012.08448 [quant-ph].

[2] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heinonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. 94, 015004 (2022), arXiv:2101.08448 [quant-ph].

[3] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. 5, 4213 (2014), arXiv:1304.3061.

[4] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028 [quant-ph].

[5] M. I. Jordan and T. M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349, 255 (2015).

[6] H. Y. Huang, R. Kueng, and J. Preskill, Information-Theoretic Bounds on Quantum Advantage in Machine Learning, Phys. Rev. Lett. 126, 190505 (2021), arXiv:2101.02464.

[7] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nat. Commun. 2021 12:1 12:1 (2021), arXiv:2011.01938 [quant-ph].

[8] H.-Y. Huang, R. Kueng, G. Torlai, V. V. Albert, and J. Preskill, Provably efficient machine learning for quantum many-body problems, arXiv:2106.12627.

[9] P. J. O’Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tran-
M. Srednicki, D. J. Luitz and Y. B. Lev, J. Z. Imbrie, V. Ros, and A. Scardicchio, I. H. Kim, A. Chandran, and D. A. Abanin, A. Chandran, I. H. Kim, G. Vidal, and D. A. Abanin, F. Alet and N. Laflorencie, J. Eisert, M. Friesdorf, and C. Gogolin, R. Nandkishore and D. A. Huse, V. Oganesyan and D. A. Huse, S. Liu, S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, J. M. Deutsch, L. Rademaker, M. Ortuño, and A. M. Somoza, O. Higgott, D. Wang, and S. Brierley, A. Pal and D. A. Huse, N. Saraceni, S. Cantori, and S. Pilati, P. Huembeli, A. Dauphin, P. Wittek, and C. Gogolin, F. Alet and N. Laflorencie, and F. Alet, Many-body localization phase transition, Phys. Rev. B 91, 081103 (2015), arXiv:1411.0660 [cond-mat-dis-nn].

R. Nandkishore and D. A. Huse, Many-body localization and thermalization in quantum statistical mechanics, Annu. Rev. Condens. Matter Phys. 6, 15 (2015), arXiv:1404.0686 [cond-mat.stat-mech].

J. Eisert, M. Friesdorf, and C. Gogolin, Quantum many-body systems out of equilibrium, Nat. Phys. 11, 124 (2015), arXiv:1408.5148 [quant-ph].

F. Alet and N. Laflorencie, Many-body localization: An introduction and selected topics, Comptes Rendus Physique 19, 498 (2018), arXiv:1711.03145 [cond-mat.str-el].

P. W. Anderson, Absence of diffusion in certain random lattices, Phys. Rev. 109, 1492 (1958).

D. Basko, I. Aleiner, and B. Altshuler, Metal-insulator transition in a weakly interacting many-electron system with localized single-particle states, Annals of Physics 321, 1126 (2006), arXiv:cond-mat/0506617 [cond-mat.mes-hall].

A. Chandran, I. H. Kim, G. Vidal, and D. A. Abanin, Constructing local integrals of motion in the many-body localized phase, Phys. Rev. B 91, 085425 (2015), arXiv:1407.8480 [cond-mat-dis-nn].

I. H. Kim, A. Chandran, and D. A. Abanin, Local integrals of motion and the logarithmic lightcone in many-body localized systems, arXiv:1412.3073 [cond-mat-dis-nn].

L. Rademaker, M. Ortuño, and A. M. Somoza, Many-body localization from the perspective of integrals of motion, Ann. Phys. 529, 1600322 (2017), arXiv:1610.06238 [cond-mat.str-el].

J. Z. Imbrie, V. Ros, and A. Scardicchio, Local integrals of motion in many-body localized systems, Ann. Phys. 529, 1600278 (2017), arXiv:1609.08076 [cond-mat-dis-nn].

D. J. Luitz and Y. B. Lev, The ergodic side of the many-body localization transition, Ann. Phys. 529, 1600350 (2017), arXiv:1610.08993 [cond-mat-dis-nn].

J. M. Deutsch, Quantum statistical mechanics in a closed system, Phys. Rev. A 43, 2046 (1991).

M. Srednicki, Chaos and quantum thermalization, Phys. Rev. E 50, 888 (1994), arXiv:cond-mat/9403051 [cond-mat].

M. Rigol, V. Dunjko, and M. Olshanii, Thermalization and its mechanisms for generic isolated quantum systems, Nature 452, 854 (2008), arXiv:0708.1324 [cond-mat.stat-mech].

L. D’Alessio, Y. Kafri, A. Polkovnikov, and M. Rigol, From quantum chaos and eigenstate thermalization to statistical mechanics and thermodynamics, Adv. Phys. 65, 239 (2016), arXiv:1509.09434.

F. Petracaprina, N. Macé, D. J. Luitz, and F. Alet, Shift-invert diagonalization of large many-body localizing spin chains, SciPost Phys. 5, 45 (2018), arXiv:1803.05395 [cond-mat.dis-nn].

S. P. Lim and D. N. Sheng, Many-body localization and transition by density matrix renormalization group and exact diagonalization studies, Phys. Rev. B 94, 045111 (2016), arXiv:1510.08145 [cond-mat.str-el].

V. Khemani, S. P. Lim, D. N. Sheng, and D. A. Huse, Critical properties of the many-body localization transition, Phys. Rev. X 7, 021013 (2017), arXiv:1607.05756 [cond-mat.dis-nn].

K. He, X. Zhang, S. Ren, and J. Sun, Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, arXiv:1502.01852.

E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, Learning phase transitions by confusion, Nat. Phys. 13, 435 (2017), arXiv:1610.02948 [cond-mat-dis-nn].

J. Carrasquilla and R. G. Melko, Machine learning phases of matter, Nat. Phys. 13, 431 EP (2017), arXiv:1605.01735 [cond-mat.str-el].

Y.-H. Liu and E. P. L. van Nieuwenburg, Discriminative cooperative networks for detecting phase transitions, Phys. Rev. Lett. 120, 176401 (2018), arXiv:1706.08111 [cond-mat.str-el].

R. G. Melko, G. Carleo, J. Carrasquilla, and J. I. Cirac, Restricted Boltzmann machines in quantum physics, Nat. Phys. 15, 887 (2019).

Y.-T. Hsu, X. Li, D.-L. Deng, and S. Das Sarma, Machine learning many-body localization: Search for the elusive nonergodic metal, Phys. Rev. Lett. 121, 245701 (2018), arXiv:1805.12138.

J. Venderley, V. Khemani, and E.-A. Kim, Machine learning out-of-equilibrium phases of matter, Phys. Rev. Lett. 120, 257204 (2018), arXiv:1711.00020 [cond-mat.dis-nn].

W. Zhang, L. Wang, and Z. Wang, Interpretable machine learning study of the many-body localization transition in disordered quantum Ising spin chains, Phys. Rev. B 99, 054208 (2019), arXiv:1807.02954.

F. Schindler, N. Regnault, and T. Neupert, Probing many-body localization with neural networks, Phys. Rev. B 95, 245134 (2017), arXiv:1704.01578 [cond-mat-dis-nn].

E. van Nieuwenburg, E. Bairey, and G. Refael, Learning phase transitions from dynamics, Phys. Rev. B 98, 060301 (2018), arXiv:1712.00450 [cond-mat-dis-nn].

P. Huembeli, A. Dauphin, P. Witteck, and C. Gogolin, Automated discovery of characteristic features of phase transitions in many-body localization, Phys. Rev. B 99, 104106 (2019), arXiv:1806.00419 [quant-ph].

E. van Nieuwenburg, Y. Baum, and G. Refael, From Bloch oscillations to many-body localization in clean interacting systems, Proc. Nat. Acad. Sci. 116, 9269 (2019), arXiv:1808.00471 [cond-mat.dis-nn].

N. Saraceni, S. Cantori, and S. Pilati, Scalable neural net-
