

ARTICLE OPEN



Efficient quantum state tomography with convolutional neural networks

Tobias Schmale¹✉, Moritz Reh¹✉ and Martin Gärtner^{1,2,3}✉

Modern day quantum simulators can prepare a wide variety of quantum states but the accurate estimation of observables from tomographic measurement data often poses a challenge. We tackle this problem by developing a quantum state tomography scheme which relies on approximating the probability distribution over the outcomes of an informationally complete measurement in a variational manifold represented by a convolutional neural network. We show an excellent representability of prototypical ground- and steady states with this ansatz using a number of variational parameters that scales polynomially in system size. This compressed representation allows us to reconstruct states with high classical fidelities outperforming standard methods such as maximum likelihood estimation. Furthermore, it achieves a reduction of the estimation error of observables by up to an order of magnitude compared to their direct estimation from experimental data.

npj Quantum Information (2022)8:115; <https://doi.org/10.1038/s41534-022-00621-4>

INTRODUCTION

With modern day noisy intermediate scale quantum (NISQ)¹ simulators outperforming each other in terms of system size and complexity on a timescale of mere months, characterizing the physically prepared states becomes exceedingly difficult. Quantum state tomography (QST)² describes the reconstruction of a density matrix from experimental measurement data and might be considered one of the hardest feats, exhausting not only experimental, but also numerical resource limits. This is due to the curse of dimensionality, inherent to all quantum systems, from which QST suffers in two-fold form: Not only do standard tomography schemes require an exponential amount of experimental data, but also the classical post-processing is often of intractable nature. We consider four main properties to be desirable for QST-schemes:

- (i) Sub-exponential scaling in required experimental data.
- (ii) Sub-exponential scaling in classical post-processing.
- (iii) 'Observable universality', requiring that upon performing a successful tomography, any linear or non-linear quantum observable should be faithfully reconstructable, without requiring further experimental data.
- (iv) 'State universality' meaning that the algorithm should be indifferent to the (possibly mixed) target state that is prepared experimentally.

Obviously, no algorithm can exist that perfectly satisfies all these conditions. Most tomography schemes that have recently been developed or applied to experimental systems, give up one or more of these conditions in order to gain w.r.t. the remaining. Standard maximum likelihood estimation (MLE)³ scales exponentially, giving up on requirements (i) and (ii). Some Bayesian methods⁴ gain on (i) by giving up (ii). Many entanglement detection schemes^{5,6} as well as shadow tomography^{7,8} give up on requirement (iii). A multitude of variational approaches have been developed which restrict the state space in which they seek for an optimal solution, therefore giving up on property (iv) and possibly (iii). Examples are matrix-product state tomography^{9–11}, which

restricts its search space to weakly entangled states; compressed sensing (CS)^{12–14}, which restricts to low-rank density matrices, and permutationally invariant (PI) tomography^{15,16}, which restricts to PI states. It is this restriction of the search space, that allows these methods to obtain target state approximations from datasets of significantly reduced sizes, which is the feature that also our work aims to exploit.

A new class of variational ansatz functions, which has been employed for QST recently are neural network quantum states (NQS)^{17–19}. The exploration of NQS is motivated by universal approximation theorems^{20,21} and the observation that many NQS can efficiently encode volume-law entanglement and thereby have higher representational power compared to most tensor-network based approaches^{22–24} as well as a favorable generalization to higher dimensions. These emerging neural network QST (NN-QST) approaches^{18,19,25–30} are starting to receive attention from the experimental communities with applications to Rydberg-, trapped-ion and optical systems^{31–34}. We note that neural networks have also been applied to quantum state readout tasks without relying on NQS^{35–38}, which we, however, exclude from our definition of NN-QST, since our focus is on efficient variational methods.

A major challenge for the successful application of NN-QST schemes is the choice of the variational ansatz, i.e. the network architecture, and the understanding of its intrinsic limitations. Recently, it was shown that convolutional neural networks (CNNs) are capable of efficiently encoding volume-law entanglement³⁹ for pure states, motivating us to explore this architecture also for NN-QST tasks operating on mixed states. We use CNNs to learn a state's probabilistic representation from measured data, bridging the gap between quantum theory and neural networks by employing Positive Operator Valued Measures (POVMs). The resulting QST scheme scales subexponentially in the system size as it contains no exponentially large state representations and makes no assumption on the target state's purity, generalizing previous approaches^{18,19}. While this scheme has been demonstrated for specific classes of states and network architectures²⁵, applications to experimental systems have not exceeded the few

¹Kirchhoff-Institut für Physik, Universität Heidelberg, Heidelberg, Germany. ²Physikalisches Institut, Universität Heidelberg, Heidelberg, Germany. ³Institut für Theoretische Physik, Universität Heidelberg, Heidelberg, Germany. ✉email: tobias.schmale@itp.uni-hannover.de; moritz.reh@kip.uni-heidelberg.de; martin.gaertner@kip.uni-heidelberg.de

qubit regime^{27,33}. The main reason for this are poorly understood performance advantages. Making further progress crucially requires the strengths and limitations of neural network based tomography schemes to be evaluated in comparison to standard tomography methods. Here we perform quantitative comparisons between our NN-QST scheme and standard techniques like MLE. For a broad range of typical experimental scenarios, we see improvements compared to MLE for small datasets, i.e. few measurement samples, as well as a noise reduction for the estimation of local observables on larger systems, thus decreasing the necessary amount of experimental samples at a given error threshold.

RESULTS

POVM formalism

For efficient NN-QST a suitable variational ansatz is crucial. Here we adopt an approach pioneered in²⁵, where the quantum state is encoded by a probability distribution over the outcomes of experimental measurements. This probability distribution, in turn, is approximated by a neural network. This has the advantage that it allows for a tomography scheme that is directly compatible with experimentally measured data and allows for applying standard probabilistic machine learning models operating on real numbers. In the following we briefly summarize the employed learning rule and the POVM formalism, and direct to²⁵, as well as^{33,40,41} for further details on NQS in combination with POVMs.

In the probabilistic formulation of quantum mechanics, the state of a system is not represented by its density matrix ρ , but by the Born-rule probabilities

$$P(\mathbf{a}) = \text{Tr}[\rho M_{\mathbf{a}}] \quad (1)$$

of an informationally complete (IC) POVM, consisting of measurement operators $M_{\mathbf{a}}$, where \mathbf{a} labels possible measurement outcomes. ‘Informationally complete’ means that the $M_{\mathbf{a}}$ form a complete basis for the set of hermitian operators, implying that any density matrix or observable can be expanded using the POVM operators. Therefore, knowing $P(\mathbf{a})$ is equivalent to knowing ρ and in principle ρ can be inferred from $P(\mathbf{a})$ by inverting Eq. (1) to $\rho = P(\mathbf{a})T_{\mathbf{a}\mathbf{a}}^{-1}M_{\mathbf{a}}$ using the overlap matrix $T_{\mathbf{a}\mathbf{a}'} = \text{Tr}[M_{\mathbf{a}}M_{\mathbf{a}'}]$ and by summing over repeated indices. Any observable may be computed by sampling from the POVM distribution using

$$\text{Tr}[\rho O] = \sum_{\mathbf{a}} P(\mathbf{a})O_{\mathbf{a}} = \langle O_{\mathbf{a}} \rangle_{\mathbf{a} \sim P(\mathbf{a})}, \quad (2)$$

where $O_{\mathbf{a}} = \text{Tr}[M_{\mathbf{a}}O]T_{\mathbf{a}\mathbf{a}}^{-1}$ is the POVM representation of an observable O . We consider systems of N qubits where each qubit is read out individually and thus the POVM elements are product operators $M_{\mathbf{a}} = M_{a_1} \otimes \dots \otimes M_{a_N}$. With such a factorized POVM, the overlap matrix T also factorizes, which allows the computation of $O_{\mathbf{a}}$ to be efficient, given that O is a local observable, or can be expressed as a sum of few (polynomially many in N), possibly non-local, Pauli strings²⁵.

The experimentally most convenient IC measurement scheme consists of single qubit Pauli measurements. By randomly selecting the x , y or z basis for measuring each qubit in each experimental run, one effectively measures a POVM with 6 possible outcomes per qubit (3 bases with 2 outcomes each). Unfortunately, this results in an overcomplete POVM, with an overlap matrix T that is not invertible. This can easily be remedied by grouping three out of the six outcomes into one, resulting in four POVM operators such as e.g.

$$\left\{ M_{0/1/2} = \frac{1}{3} |\uparrow_{x/y/z}\rangle \langle \uparrow_{x/y/z}|, M_3 = \mathbb{1} - M_0 - M_1 - M_2 \right\}.$$

Here M_3 now incorporates the 3 $|\downarrow\rangle$ outcomes for all axes x , y and z . This POVM is typically referred to as the Pauli-4 POVM²⁵. In this

setting, a measurement outcome $\mathbf{a} = a_1 a_2 \dots a_N$ is a string of N single qubit outcomes for a N qubit system, over which the POVM distribution $P(\mathbf{a}) = P(a_1, a_2, \dots, a_N)$ is defined. A dataset of size N_s is a set of multiple such outcomes $D = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{N_s}\}$.

We use a neural network as a non-linear function that returns a single POVM probability $P(\mathbf{a})$ given a POVM outcome \mathbf{a} as input. We can thus write our neural network ansatz for the POVM distribution as $P_{\text{NN}}^{\theta}(\mathbf{a})$, where θ denotes the tuple of variational parameters describing the ansatz. Fitting the variational parameters of the network builds on standard machine learning methods: We use the established ADAM optimizer⁴² to find the parameters θ that have the highest likelihood of reproducing the data. This is equivalent to minimizing the cross-entropy between the dataset distribution and the NN distribution. The ADAM optimizer gave considerably better results than less advanced optimizers like the pure gradient-descent optimizer or a momentum optimizer.

We finally point out, that the positivity condition on the density matrix is not known to easily translate into the POVM formalism, without introducing exponential scaling⁴³. Therefore, one cannot guarantee that all sampled observables obey physical constraints (such as $\text{Tr}[\rho^2] < 1$). We have observed violations of such constraints very rarely, hence we argue that the lack of the positivity condition is not a limiting factor of this approach. While RBM purifications⁴⁴ are one means of achieving a positive neural network density matrix, these can currently not make use of other, more flexible network architectures such as CNNs.

CNN architectures

NN-QST in conjunction with POVMs has been successfully applied to recurrent neural networks (RNNs)²⁵ as well as attention based models²⁷. We expand on this knowledge, by employing two distinctly different versions of CNNs, motivated by recent theoretical developments, which frame CNNs as a generalization to matrix product states^{24,39}.

The first architecture we consider is the ‘standard’ CNN in Fig. 1a, which is fed a one-hot encoded vector of single-site outcomes, performs repeated convolutions on it and returns the corresponding (unnormalized) POVM probability $\tilde{P}_{\text{NN}}(\mathbf{a})$. Motivations for using this architecture are its established representational power when operating on pure states^{24,39}, the ability to represent 1D and 2D states, as well as the ability to encode symmetries such as translation invariance without increasing the computational complexity. Unfortunately, one has to resort to Markov Chains in order to draw samples from the network, which can lead to unintended correlations between generated samples⁴⁵.

The second architecture is a 1D autoregressive CNN (ARCNN)^{46,47}, illustrated in Fig. 1b. This architecture is very similar to the standard CNN, but makes use of the autoregressive property, which states that a probability of multiple variables can be partitioned into a product of conditionals, which are returned by the final layer of the ARCNN: $P(\mathbf{a}) = P(a_1)P(a_2|a_1)\dots P(a_N|a_1 \dots a_{N-1})$. Other network architectures that also make use of this property include recurrent neural networks²⁵ and Transformers²⁷. The autoregressive property of the ARCNN allows probabilities to be exactly normalized, and for samples to be drawn exactly, circumventing the need for Markov Chains. More details on both architectures can be found in the methods section.

Expressivity of CNNs

For both the CNN with product-output layer (see methods) and the ARCNN, one can write down a maximum (physical) distance d_{max} , beyond which correlations are typically not captured

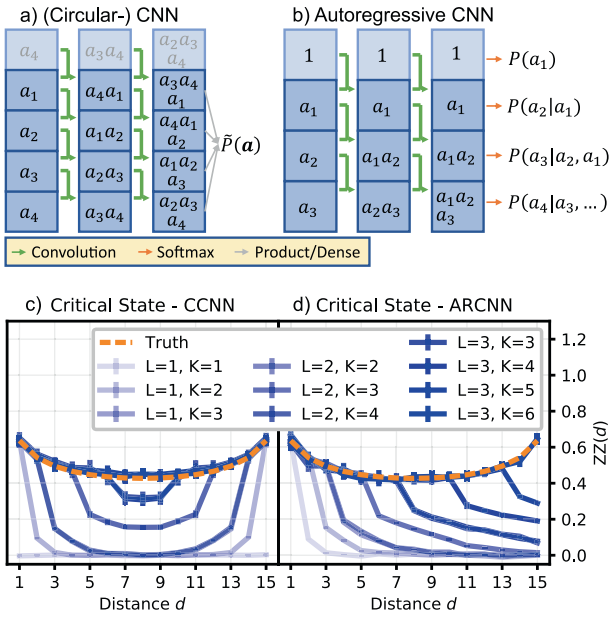


Fig. 1 CNN architectures and their expressivity. **a, b** Schematics of the CNN architectures. Entries of cells should be understood as a dependency, not as literal values, e.g. bottom right cell of standard CNN depends on a_2, a_3 and a_4 . Lighter cells are padding cells. The green arrows indicate the dot-product of the originating cells with a kernel. See the methods section for details. **a** Standard CNN, here with circular boundary (CCNN), **b** Autoregressive CNN (ARCNN). **c, d** $\langle \sigma_i^z \sigma_j^z \rangle$ correlator as a function of distance $d = |i - j|$, for varying network depth L and kernel size K , shown for 1D transverse-field Ising model (TFIM) ground states ($N = 16$, $J/B = 1$). Error bars are standard deviations across multiple independent tomography datasets and neural network training runs with randomly initialized parameters.

correctly anymore:

$$d_{\max}^{\text{CNN}} = (K - 1)L, \quad (3)$$

$$d_{\max}^{\text{ARCNN}} = (K - 1)L + 1, \quad (4)$$

where L is the network depth and K the size of the convolution kernels. Both can be understood intuitively from the schematics in Fig. 1a and b: With each layer (i.e. L times), the dependence of any single site outcome gets propagated by a distance of one less than the kernel size (i.e. $K - 1$). For the autoregressive CNN, an input of a_i results in the probability for a_{i+1} , explaining the fact, that correlations reach one site further as compared to the CNN. These cutoff points for correlations can be seen in Fig. 1c and d.

For the following results, we are mainly interested in systems with long-range correlations, hence we use architectures with $d_{\max} = l$, where l is the (side-) length of (2D) 1D systems. This gives a very compact ansatz, with the number of variational parameters scaling only cubically in the system size. For the benchmark cases studied in this work, we did not find a benefit in increasing d_{\max} beyond this point, as this increases the number of network parameters, artificially enlarging the search space and increasing the risk of overfitting, resulting in worse generalization. In the following it will be crucial to quantify how well the networks generalize, i.e. how many samples one needs to learn a state well, or equivalently, what errors one can expect for given dataset sizes.

NN-QST benchmarks

We benchmark the tomography scheme by comparing it to two conventional schemes: MLE, i.e. parametrizing the full density

matrix and maximizing the likelihood of this reproducing the dataset, as it presents the go-to choice for many small scale qubit systems and is commonly used in experiments^{3,48}, as well as direct estimation of observables from the dataset. To carry out this comparison we generate synthetic measurement data sets. We compute a target density matrix exactly and compute its POVM distribution, from which we draw samples ($1k - 100k$ for 16 qubit systems). We then use these samples to train the network, resulting in the optimal variational representation $P_{\text{NN}}(\mathbf{a})$. For small systems, i.e. those where it is feasible, we perform MLE to obtain an estimate for the density matrix following³, of which we compute the POVM distribution $P_{\text{MLE}}(\mathbf{a})$. We then ask, which of these two estimates is closer to the ground truth target distribution, as measured by the classical infidelity

$$D_{\text{NN/MLE}} = 1 - \sum_{\mathbf{a}} \sqrt{P_{\text{NN/MLE}}(\mathbf{a}) P_{\text{Truth}}(\mathbf{a})}. \quad (5)$$

This allows us to quantify, which of the two estimates is better, by using the quotient $D_{\text{NN}}/D_{\text{MLE}}$. If it is less than one, the network gives the better estimate for the target state compared to MLE, and vice versa.

For systems where MLE is infeasible, we instead consider the root mean square (RMS) error of observables

$$\text{RMS}_{\text{NN/Data}} = \sqrt{\langle (O_{\text{NN/Data}} - O_{\text{Truth}})^2 \rangle}, \quad (6)$$

inspired by²⁶. These can either be computed from the training dataset itself, or from the network-encoded distribution, from which we draw 500k samples for 16 qubit systems. In this situation the NN acts in a way of replacing the measured dataset with a larger, network-generated one, aiming to decrease statistical measurement noise. Here we can ask, if the NN gives an advantage for the estimation of observables, by looking at the quotient $\text{RMS}_{\text{NN}}/\text{RMS}_{\text{Data}}$.

Transverse field Ising model. We start by benchmarking this method on ground states of a translation-invariant TFIM

$$H = -J \sum_{\langle i,j \rangle} \sigma_i^z \sigma_j^z - B \sum_i \sigma_i^x \quad (7)$$

with coupling strength $J > 0$ using the standard CNN. This serves as a proof of concept in an idealized scenario, as the translation invariance is directly encoded in the neural network. Since the comparison of a symmetrized network to an unsymmetrized reference is somewhat unfair, we refrain from enforcing any symmetries for the later examples.

Figure 2 shows the method being applied to small, i.e. MLE-suitable 1D states at the Ising-critical point. Our method achieves a reduction of infidelity by a factor 2-5, depending on system and dataset size. The figure shows one main trend: the network advantage shrinks for increased dataset size. This is also an expected result, as MLE has to outperform any variational approach in the limit of infinite dataset size since the network is an approximation while MLE uses a full parameterization of the state. We generally see this behavior for all studied systems.

In Fig. 3, we show the method being applied to a 4×4 lattice, which is not feasible for MLE anymore. The histograms show how enhancing the dataset using the NN can lead to a reduced variance of observable estimates. For the network advantage, we see two trends: An increased advantage for small datasets, as well as an improved performance for smaller coupling strengths. The latter results in states closer to product states, which are easier for the network to learn. If J/B becomes too large, training the network becomes unreliable on such small datasets and the advantage disappears.

Noisy long-range interacting ion chain. For a more experimentally motivated⁴⁹ example, we look at ground states $|\psi_0\rangle$ of a 16-site,

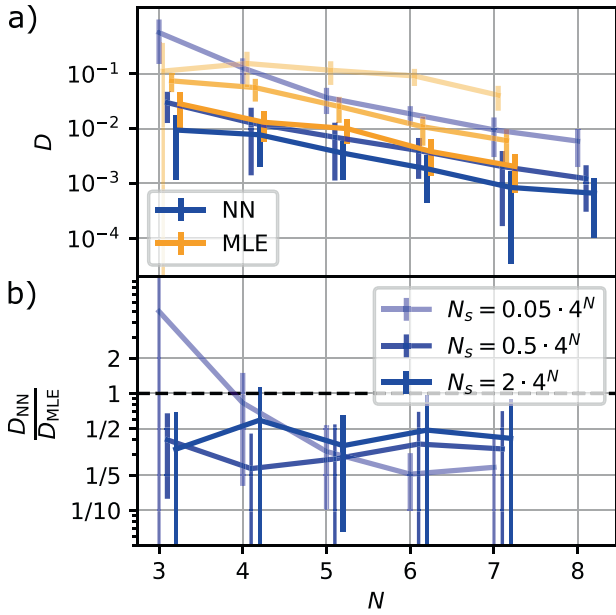


Fig. 2 Tomography of 1D Ising groundstates: NN vs MLE. **a** Residual classical infidelity of CCNN and MLE on 1D Ising ground state with periodic boundary conditions, $J/B = 1$. Shown for different dataset sizes N_s . **b** Residual network infidelity normalized to residual MLE infidelity. Error bars are standard deviations across multiple independent tomography datasets and neural network training runs with randomly initialized parameters.

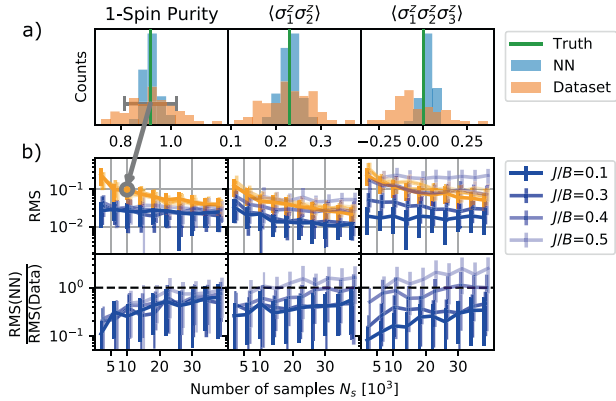


Fig. 3 Tomography of 2D Ising groundstates: NN vs plain data. Comparing the performance of the CCNN against the plain dataset in estimating local observables on a 4×4 Ising lattice with periodic boundaries. **a** Histogram of observable estimates from independent datasets and network initializations, with $J/B = 0.3$ and $N_s = 10k$. **b** RMS errors of observables for varying coupling strengths and dataset sizes, as well as comparisons of NN and plain dataset in terms of the quotient of their respective RMS errors. Error bars are standard deviations across multiple independent tomography datasets and neural network training runs with randomly initialized parameters.

long-range interacting ion chain Hamiltonian

$$H = - \sum_{ij} \frac{J}{|i-j|^{1.1}} \sigma_i^z \sigma_j^z - B \sum_i \sigma_i^x \quad (8)$$

with $J > 0$, open boundary conditions and small (3%) added dephasing noise. The target state is thus $\rho_{\text{Target}} = 0.97 |\psi_0\rangle\langle\psi_0| + \frac{0.03}{2} \mathbb{1}$. As this is naturally a 1D system, we use the ARCNN. Here we study the NN advantage for correlation functions of increasing order. This is interesting, as higher order correlators

are typically harder to estimate from samples, as the variance of the POVM-observable scales exponentially in correlation order. Thus, higher moments require a better approximation of the state, providing a sensitive benchmark for the quality of the state representation. Specifically, we look at powers n of the Pauli-Z operator

$$C_n := \frac{1}{16-n+1} \sum_{i=1}^{16-n+1} \langle \sigma_i^z \sigma_{i+1}^z \dots \sigma_{i+n-1}^z \rangle. \quad (9)$$

Local observables, like the terms in Eq. (9), only depend on the reduced density matrix of the subsystem they act on. Thus, these observables can in principle be estimated by performing MLE on this subsystem only. We show this ‘local MLE’ applied to the bare dataset and to the NN-enhanced dataset in Fig. 4 in addition to the previously employed benchmarks. Using the NN-generated dataset, we see a reduction in RMS to a degree, that allows sampling for correlators of three orders higher, than what is possible with the bare dataset. When applying the local MLE to both bare and NN-generated dataset, this advantage is reduced significantly, but does not disappear. However, we emphasize that the ARCNN is on par with MLE, at a greatly reduced computational complexity. We note that the computational cost depends on the specific implementation and that there exist more efficient MLE algorithms than the one used here (eg, ref.⁵⁰). However, the difference in scaling behaviour between MLE and NN-QST persists (cf. Methods section). Notice also the peak in RMS at a correlation order of 2 for MLE, leading to an increased RMS compared to plain sampling. We find this to be systematic, which is why we refrain from doing this comparison to local MLE in the remainder of this work, where second order correlation functions are of interest.

Close examination of the datapoint for C_6 in Fig. 4 reveals effects of the lack of a positivity constraint on the density matrix for direct estimation from samples and POVM based NN-QST. Here the dataset estimate lies outside the physical range $-1 \leq C_n \leq 1$, but the network is able to correct this due to the reduced statistical error. Only for C_8 (not shown) the network cannot cure the unphysical feature present in the dataset. Testing that observables only take physical values may serve as a useful sanity check for whether the network generalizes towards a physical state.

Steady states of a driven dissipative 2D-system. As a final system, we consider steady states of a 4×4 TFIM with spontaneous decay, motivated by ongoing research into phase diagrams of open quantum systems⁵¹, with potential applications to Rydberg systems. We switch back to the standard CNN with dense output layer, since the system is two-dimensional. However, now no symmetries are enforced in the network. We use the Monte-Carlo wave function approach⁵², to simulate the dynamics under the Lindblad master equation

$$\dot{\rho} = -i[H, \rho] + \gamma \sum_j \left(L_j \rho L_j^\dagger - \frac{1}{2} \{L_j^\dagger L_j, \rho\} \right) \quad (10)$$

with the Hamiltonian from Eq. (7) with antiferromagnetic coupling $J < 0$, $B < 0$ and

$$L_j = \sigma_j^- = \frac{1}{2} (\sigma_j^x - i\sigma_j^y), \quad (11)$$

until the steady state is reached. We consider the density matrix that is obtained using 1000 pure-state trajectories as our exact target. This system undergoes a dissipative phase transition⁵¹ which is visible as a peak in the correlation length

$$\xi_z^2 = \sum_{ij} |\mathbf{r}_i - \mathbf{r}_j|^2 (\langle \sigma_i^z \sigma_j^z \rangle - \langle \sigma_i^z \rangle \langle \sigma_j^z \rangle). \quad (12)$$

We show this phase transition as obtained by computing the correlation length on the 1D diagonal of the 2D lattice, once

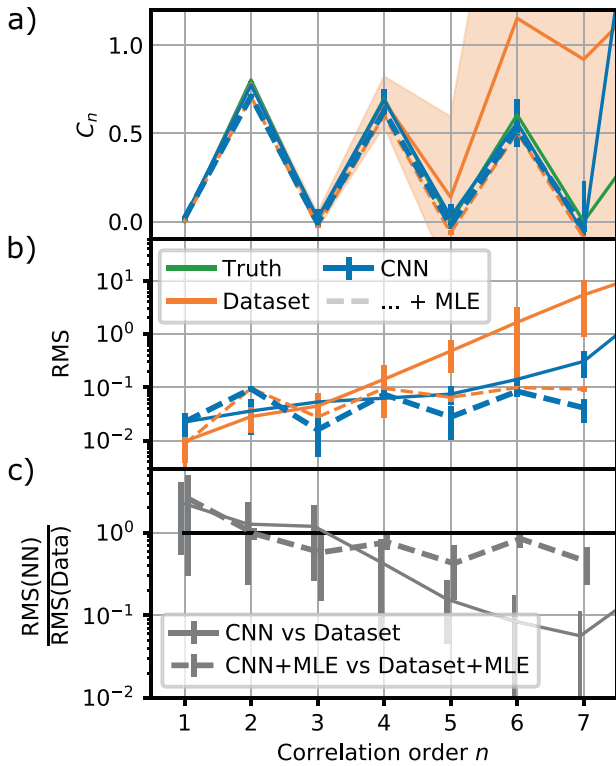


Fig. 4 Tomography of noisy long-range interacting groundstates: NN vs plain data. Evaluating the ARCNN by evaluating observables from Eq. (9) and comparing to those obtained from the plain dataset, on a length 16 ion chain with 3% dephasing noise, $J/B = 0.6$ and open boundaries, with a dataset size of $N_s = 10$ k. Dashed lines show results of applying local MLE directly to dataset (orange) and to a neural network enhanced dataset (blue). **a** Observables according to Eq. (9), **b** residual RMS error, **c** residual network RMS normalized to dataset RMS. Error bars are standard deviations across multiple independent tomography datasets and neural network training runs with randomly initialized parameters.

directly from the training data, as well as from a NN-enhanced dataset in Fig. 5. The network is able to capture the phase transition, as a peak in the correlation length at $|B/\gamma| \approx 2$ is clearly visible. At the dashed gray line, we exchange \uparrow and \downarrow in the POVM that the CNN uses (i.e. M_3 now groups all the $|\uparrow\rangle$ outcomes instead of the $|\downarrow\rangle$ outcomes), ensuring that the target state does not contain exact zeros in its POVM distribution. For small $|B|$, the steady state tends towards an eigenstate of the observable in question. Thus the variance of sampling this observable is significantly reduced and the network has no advantage here. For the limit of large $|B|$ we see a huge variance in the sampled correlation length, which the CNN trades for a small bias, i.e. systematic error. The overall effect is that the CNN bias and CNN variance lead to a significantly smaller RMS error as compared to the bare dataset. Notice that this bias also shrinks with increasing dataset size (Fig. 5, insets).

Depending on the observable of interest, the bias can have a more severe effect than depicted. When computing the correlation length over the 1D diagonal, as in Fig. 5, the corresponding sum in Eq. (12) is a weighted average of $\binom{4}{2} = 6$ connected correlators of the form $\langle \sigma_i^z \sigma_j^z \rangle - \langle \sigma_i^z \rangle \langle \sigma_j^z \rangle$. After sampling, one may consider each of these connected correlators as a random variable with a variance and a bias. For the plain dataset, this bias is of course zero. However, when evaluating Eq. (12) over the entire lattice, the sum contains $\binom{16}{2} = 120$ terms, with roughly similar variance and bias. For the sampled case, by simple addition of probability distributions, the variance of the latter observable is thus reduced by a factor of $\sqrt{120/6} \approx 4.5$ compared to the

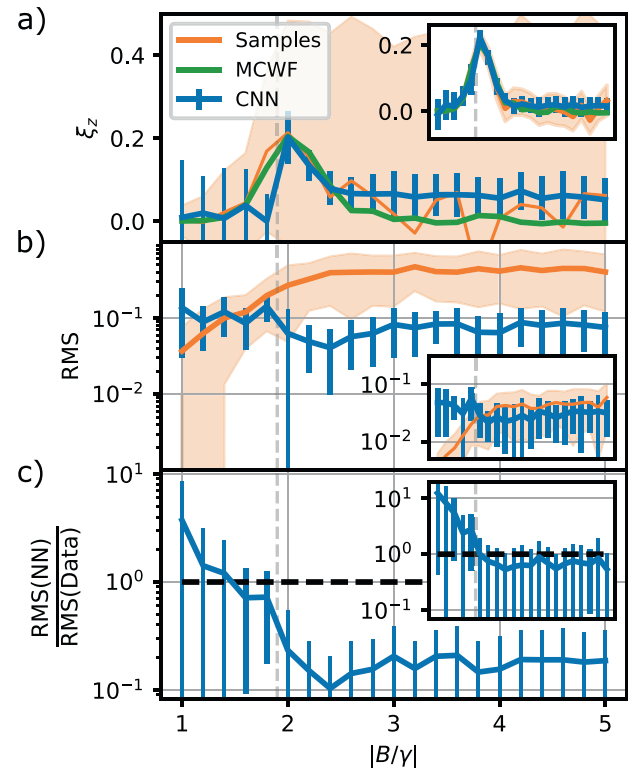


Fig. 5 Tomography of steady states: NN vs plain data. Dissipative phase transition of a 4×4 TFIM with spontaneous emission at $J = -1.25\gamma$ with open boundaries. $N_s = 1$ k (insets: 100 k). Gray vertical line: Switch \uparrow and \downarrow in POVM for CNN. **a** Observables according to Eq. (12) but summed only over system diagonal, **b** residual RMS error, **c** residual network RMS normalized to dataset RMS. Error bars are standard deviations across multiple independent tomography datasets and neural network training runs with randomly initialized parameters.

former one. Due to the bias, the network is not able to make use of this self averaging effect, resulting in a significantly reduced advantage. We show this full system correlation length in the methods section. Thus the network advantage is the largest, if one is interested in the expectation values of individual correlators, and might be smaller if large sums over many similarly distributed correlators are involved as the bias inherent to the variational approach becomes more statistically significant for the averaged results.

DISCUSSION

Motivated by the proven superior representation capabilities of CNNs³⁹, we explored the application of two different CNN architectures to NN-QST tasks, i.e. the reconstruction of pure and mixed quantum states. We especially found the autoregressive CNN to be extremely versatile, due to its great expressivity, exact sampling and stability during training. For a broad range of experimentally relevant scenarios, including pure and dephased ground states, as well as steady states, we presented quantitative comparisons to traditional schemes. This showed a significant advantage over MLE or, for system sizes inaccessible to MLE, direct sampling of local observables from an IC-POVM dataset. Although we demonstrated the described properties on synthetic data, the method may be readily applied to real quantum simulation experiments. Once the quantum state has been successfully learned from experimental data, any linear or non-

linear observable can be extracted from the resulting NQS representation.

Like any variational approach, which benefits from restricting the state space to a representable subspace, this scheme is subject to potential bias. The effect of such bias intimately depends on the estimated observable, the approximated state, as well as the sample size, as discussed above. This makes it necessary to a priori validate the method to ensure generalization of a given model to the situation at hand, as we have done in this work for a range of experimentally motivated cases.

We emphasize that the proposed method can harness the strengths of any variational function approximator, thus directly profiting from the rapid development of ever more expressive architectures in the machine learning community. New network architectures can only enhance the state space that is covered by NN-QST, motivating further research in this area. For future projects it would be interesting to compare many of the NN-QST schemes, to methods like shadow tomography^{7,8}, which explicitly claim superiority over NN-QST.

METHODS

Details on network architectures

To better explain the obtained parameter counts for the neural networks, we briefly explain the convolution operation in more detail, which the CNNs perform iteratively (here shown for the 1D case).

$$(\text{conv}(\mathbf{x}; \mathbf{k}, b))_i = f \left(b + \sum_{j=0}^{|\mathbf{k}|-1} k_j x_{j+i} \right), \quad (13)$$

for $i \in \{0, \dots, |\mathbf{x}| - |\mathbf{k}|\}$

of the input data \mathbf{x} with a so-called kernel \mathbf{k} denotes a single convolution. Here the length of a vector \mathbf{v} is denoted by $|\mathbf{v}|$, b is a bias and f is a non-linear activation function acting element-wise on the results of the convolution. This can be thought of as taking dot-products of the kernel and translated section of the input vector. For a 2D CNN, \mathbf{k} would be a matrix, and Eq. (13) would compute dot products between this matrix and translated submatrices of the then two-dimensional input \mathbf{x} .

Multiple kernels are used per layer l resulting in multiple intermediate representations $\mathbf{x}^{l,m}$. The latter is then computed via

$$\mathbf{x}^{l+1,m} = \sum_n \text{conv}(\mathbf{x}^{l,n}; \mathbf{k}^{l,m,n}, b^{l,m}). \quad (14)$$

This structure on kernel-level is *not* explicitly depicted in Fig. 1. The axis indexed by m here is often called the *feature* dimension.

Keeping the number of features per layer $f = N$ constant, as done throughout this work and using Eq. (3), the $\mathbf{k}^{l,m,n}$ tensor has $K \cdot L \cdot f \cdot f = \mathcal{O}(N) \cdot N \cdot N = \mathcal{O}(N^3)$ parameters.

We employ two distinctly different network architectures. The first is the ‘standard’ CNN in Fig. 1a, which is fed a one-hot encoded vector of single-site outcomes (leading to an input shape of (Batch-Size, N , 4)) and performs L convolutions with kernels of size K . Boundary conditions are either open or periodic, depending on the symmetries of the target state. In Eq. (13) a periodic

boundary condition implies that x_{i+j} wraps around to x_0 when $i+j$ reaches the length of \mathbf{x} . A final layer turns the network output into a single scalar. Two common options are a dense layer, or a product layer, the latter resulting in network outputs of the form $\tilde{P}_{\text{NN}}(a) = e^{\sum \text{last layer}}$. This results in an *un*-normalized probability distribution. Therefore, before each training step, a Monte Carlo estimate of the normalization constant has to be performed. For normalization, we generate as many uniform POVM samples as there are samples within each training batch. For translation invariant states, we use the product output layer and the dense output layer otherwise.

The second architecture is a 1D autoregressive CNN (ARCNN)^{46,47} in Fig. 1b. This architecture makes use of the autoregressive property, which states that a probability of multiple variables can be partitioned into a product of conditionals: $P(\mathbf{a}) = P(a_1)P(a_2|a_1) \dots P(a_N|a_1 \dots a_{N-1})$, see⁴⁷ for more details on the autoregressive structure. The autoregressive CNN only differs from the standard CNN in the boundary conditions and in the fact that the physical dimension is shifted by one site, so that the last site is not used as an input. The outputs are fed into a softmax layer, which results in four conditional probabilities to be returned from each site. The exactly normalized probabilities may be computed by passing a POVM outcome through the network and selecting the conditional probability according to the input at each site. Exact samples can be drawn from this distribution, by passing zeros through the network, sampling the first site, passing this outcome through the network and sampling the next site, etc. This results in N forward passes in order to generate one exact POVM sample, at the benefit of circumventing Markov chains.

In addition to exact normalization and sampling, we empirically find this network architecture to be substantially easier to train, and the training algorithm to converge for a wider variety of states. There is no simple generalization of this approach to higher dimensional systems, however approaches like⁴⁶ have been proposed.

The hyperparameter configurations used for producing the data shown in each figure are summarized in Table. 1.

All training runs throughout this project took less than 2000 epochs to complete while all MLE runs required less than 100 iterations. This shows that there is no prohibitive issue in training-complexity for the states tested. One training epoch using the NN takes a time of $\mathcal{O}(\text{num.Parameters}) = \mathcal{O}(N^3)$ ⁴², while one iteration in the MLE takes a time of $\mathcal{O}(4^N)$ ³.

Self-averaging can reduce network advantage

As explained in the main text, the effect of a bias, that the network might introduce for a given observable may be amplified when one is interested in observables which contain sums over many independent observables. In this scenario the network-bias inhibits any positive influence of possible self averaging effects. We demonstrate this, by computing the correlation length from Eq. (12) by summing over all pairs of lattice sites, as opposed to only summing over a diagonal as in Fig. 5. The result is shown in Fig. 6. The sampled observables self-average, resulting in a smaller statistical error at fixed sample size, while the network systematically

Table 1. Network architectures used for each figure. In all cases we used a tanh-activation function, a learning rate of 10^{-3} , N features per layer for N particles, and an exponential output function. $[x]$ denotes the nearest integer function and $\lceil x \rceil$ the ceiling function.

Figure	Network architecture	Network boundary conditions	Num. Layers L	Kernel size K	Num. Parameters
2 - Small 1D TFIM	CNN	Circular	$\lceil \sqrt{N} \rceil$	$\lceil \sqrt{N} \rceil + 1$	73–673
3 - Larger 2D TFIM	CNN	Circular	2	2	1329
4 - Ion Chain	ARCNN	Open	3	6	3572
5 - Steady States	CNN	Open	2	3	3169

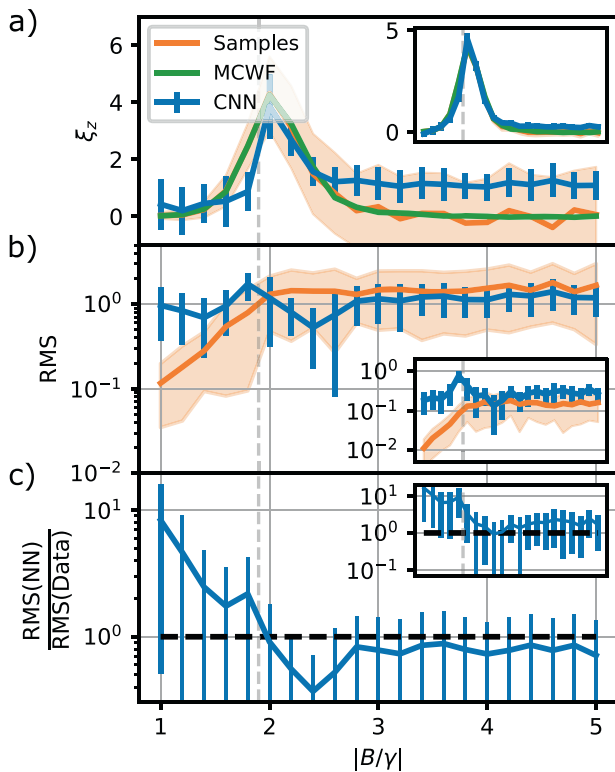


Fig. 6 Tomography of steady states: demonstration of a self-averaging effect. Dissipative phase transition of a 4×4 TFIM with spontaneous emission at $J = -1.25\gamma$ with open boundaries. $N_s = 1k$ (insets: $100k$). Gray vertical line: Point where \uparrow and \downarrow have been exchanged in POVM. **a** Observables according to Eq. (12) summed over entire system. **b** Residual RMS error. **c** Residual network RMS normalized to dataset RMS. Error bars are standard deviations across multiple independent tomography datasets and neural network training runs with randomly initialized parameters.

overestimates the correlation length for fields greater than the critical field.

DATA AVAILABILITY

The code developed for this project for generating test data is available at gitlab.com/ann-povm/qst-code.

CODE AVAILABILITY

The code developed for this project for performing the NN-QST benchmarks is available at gitlab.com/ann-povm/qst-code.

Received: 26 November 2021; Accepted: 17 August 2022;

Published online: 23 September 2022

REFERENCES

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
2. Paris, M. & Řeháček, J. (eds.) Quantum State Estimation, vol. 649 of *Lecture Notes in Physics* (Springer Berlin Heidelberg, 2004). <https://doi.org/10.1007/b98673>.
3. Lvovsky, A. I. Iterative maximum-likelihood reconstruction in quantum homodyne tomography. *J. Opt. B: Quantum Semiclass. Opt.* **6**, S556–S559 (2004).
4. Blume-Kohout, R. Optimal, reliable estimation of quantum states. *N. J. Phys.* **12**, 043034 (2010).
5. Harney, C., Pirandola, S., Ferraro, A. & Paternostro, M. Entanglement classification via neural network quantum states. *N. J. Phys.* **22**, 045001 (2020).

6. Harney, C., Paternostro, M. & Pirandola, S. Mixed state entanglement classification using artificial neural networks. *N. J. Phys.* **23**, 063033 (2021).
7. Huang, H.-Y., Kueng, R. & Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.* **16**, 1050–1057 (2020).
8. Struchalin, G., Zagorovskii, Y. A., Kovlakov, E., Straupe, S. & Kulik, S. Experimental estimation of quantum state properties from classical shadows. *PRX Quantum* **2**, 010307 (2021).
9. Cramer, M. et al. Efficient quantum state tomography. *Nat. Commun.* **1**, 149 (2010).
10. Baumgratz, T., Gross, D., Cramer, M. & Plenio, M. B. Scalable reconstruction of density matrices. *Phys. Rev. Lett.* **111**, 020401 (2013).
11. Lanyon, B. P. et al. Efficient tomography of a quantum many-body system. *Nat. Phys.* **13**, 1158–1162 (2017).
12. Gross, D., Liu, Y.-K., Flammia, S. T., Becker, S. & Eisert, J. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.* **105**, 150401 (2010).
13. Schwemmer, C. et al. Experimental comparison of efficient tomography schemes for a six-qubit state. *Phys. Rev. Lett.* **113**, 040503 (2014).
14. Riofrío, C. A. et al. Experimental quantum compressed sensing for a seven-qubit system. *Nat. Commun.* **8**, 15305 (2017).
15. Tóth, G. et al. Permutationally invariant quantum tomography. *Phys. Rev. Lett.* **105**, 250403 (2010).
16. Moroder, T. et al. Permutationally invariant state reconstruction. *N. J. Phys.* **14**, 105001 (2012).
17. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
18. Torlai, G. et al. Neural-network quantum state tomography. *Nat. Phys.* **14**, 447–450 (2018).
19. Torlai, G. & Melko, R. G. Machine-learning quantum states in the NISQ era. *Annu. Rev. Condens. Matter Phys.* **11**, 325–344 (2020).
20. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
21. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **4**, 251–257 (1991).
22. Deng, D.-L., Li, X. & Das Sarma, S. Quantum entanglement in neural network states. *Phys. Rev. X* **7**, 021021 (2017).
23. Huang, Y. & Moore, J. E. Neural network representation of tensor network and chiral states. *Phys. Rev. Lett.* **127**, 170601 (2021).
24. Sharir, O., Shashua, A. & Carleo, G. Neural tensor contractions and the expressive power of deep neural quantum states. *arXiv:2103.10293*. <http://arxiv.org/abs/2103.10293> (2021).
25. Carrasquilla, J., Torlai, G., Melko, R. G. & Aolita, L. Reconstructing quantum states with generative models. *Nat. Mach. Intell.* **1**, 155–161 (2019).
26. Torlai, G., Mazzola, G., Carleo, G. & Mezzacapo, A. Precise measurement of quantum observables with neural-network estimators. *Phys. Rev. Res.* **2**, 022060 (2020).
27. Cha, P. et al. Attention-based quantum tomography. *Mach. Learn.: Sci. Technol.* **3**, 01LT01 (2021).
28. Smith, A. W. R., Gray, J. & Kim, M. S. Efficient quantum state sample tomography with basis-dependent neural-networks. *PRX Quantum* **2**, 020348 (2021).
29. Melkani, A., Gneiting, C. & Nori, F. Eigenstate extraction with neural-network tomography. *Phys. Rev. A* **102**, 022412 (2020).
30. Huang, H. & Situ, H. Investigating reconstruction of quantum state distributions with neural networks. *Eur. Phys. J.* **136**, 204 (2021).
31. Palmieri, A. M. et al. Experimental neural network enhanced quantum tomography. *Npj Quantum Inf.* **6**, 20 (2020).
32. Torlai, G. et al. Integrating neural networks with a quantum simulator for state reconstruction. *Phys. Rev. Lett.* **123**, 230504 (2019).
33. Neugebauer, M. et al. Neural network quantum state tomography in a two-qubit experiment. *Phys. Rev. A* **102**, 042604 (2020).
34. Tiunov, E. S., Tiunova, V. V., Ulanov, A. E., Lvovsky, A. I. & Fedorov, A. K. Experimental quantum homodyne tomography via machine learning. *Optica* **7**, 448 (2020).
35. Ahmed, S., Sánchez Muñoz, C., Nori, F. & Kockum, A. F. Quantum state tomography with conditional generative adversarial networks. *Phys. Rev. Lett.* **127**, 140502 (2021).
36. Lohani, S., Kirby, B. T., Brodsky, M., Danaci, O. & Glasser, R. T. Machine learning assisted quantum state estimation. *Mach. Learn.: Sci. Technol.* **1**, 035007 (2020).
37. Quek, Y., Fort, S. & Ng, H. K. Adaptive quantum state tomography with neural networks. *npj Quantum Inf.* **7**, 105 (2021).
38. Lode, A. U. J. et al. Optimized observable readout from single-shot images of ultracold atoms via machine learning. *Phys. Rev. A* **104**, L041301 (2021).
39. Levine, Y., Sharir, O., Cohen, N. & Shashua, A. Quantum entanglement in deep learning architectures. *Phys. Rev. Lett.* **122**, 065301 (2019).
40. Carrasquilla, J. et al. Probabilistic simulation of quantum circuits using a deep-learning architecture. *Phys. Rev. A* **104**, 032610 (2021).

41. Reh, M., Schmitt, M. & Gärttner, M. Time-dependent variational principle for open quantum systems with artificial neural networks. *Phys. Rev. Lett.* **127**, 230501 (2021).
42. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv:1412.6980*. <http://arxiv.org/abs/1412.6980> (2017).
43. Fuchs, C. A. & Schack, R. Quantum-bayesian coherence. *Rev. Mod. Phys.* **85**, 1693–1715 (2013).
44. Torlai, G. & Melko, R. G. Latent space purification via neural density operators. *Phys. Rev. Lett.* **120**, 240503 (2018).
45. Liu, J. S. Monte Carlo Strategies in Scientific Computing. Springer Series in Statistics (Springer New York, 2004). <https://doi.org/10.1007/978-0-387-76371-2>.
46. Sharir, O., Levine, Y., Wies, N., Carleo, G. & Shashua, A. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.* **124**, 020503 (2020).
47. Lin, S.-H. & Pollmann, F. Scaling of neural-network quantum states for time evolution. *Phys. Status Solidi B* **259**, 2100172 (2022).
48. Lanyon, B. P. et al. Measurement-based quantum computation with trapped ions. *Phys. Rev. Lett.* **111**, 210501 (2013).
49. Tan, W. L. et al. Observation of domain wall confinement and dynamics in a quantum simulator. *Nat. Phys.* **17**, 742–747 (2021).
50. Shang, J., Zhang, Z. & Ng, H. K. Superfast maximum-likelihood reconstruction for quantum tomography. *Phys. Rev. A* **95**, 062336 (2017).
51. Jin, J. et al. Phase diagram of the dissipative quantum ising model on a square lattice. *Phys. Rev. B* **98**, 241108 (2018).
52. Plenio, M. B. & Knight, P. L. The quantum-jump approach to dissipative dynamics in quantum optics. *Rev. Mod. Phys.* **70**, 101–144 (1998).
53. Johansson, J., Nation, P. & Nori, F. QuTiP 2: A python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **184**, 1234–1240 (2013).
54. Bradbury, J. et al. JAX: composable transformations of Python+NumPy programs. github.com/google/jax (2018).
55. Heek, J. et al. Flax: A neural network library and ecosystem for JAX. github.com/google/flax (2020).
56. Jülich Supercomputing Centre. JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre. *JLSRF* **5**, A135 (2019).

ACKNOWLEDGEMENTS

We thank J. Carrasquilla and M. Schmitt for discussions. Monte-Carlo wave function trajectories were computed using the Qutip library⁵³. The JAX library⁵⁴ as well as the Flax framework⁵⁵ were used to build the neural network models and training algorithms.

AUTHOR CONTRIBUTIONS

T.S. developed the code base, M.R. performed MCWF simulations. All authors contributed equally to analysing the data and writing of the manuscript. M.G. supervised the project.

FUNDING

This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC2181/1-390900948 (the Heidelberg STRUCTURES Excellence Cluster) and within the Collaborative Research Center SFB1225 (ISOQUANT). This work was partially financed by the Baden-Württemberg Stiftung gGmbH. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant no INST 40/575-1 FUGG (JUSTUS 2 cluster). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS⁵⁶ at Jülich Supercomputing Centre (JSC). Open Access funding enabled and organized by Projekt DEAL. For the publication fee we acknowledge financial support by Deutsche Forschungsgemeinschaft within the funding programme, Open Access Publikationskosten" as well as by Heidelberg University.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to Tobias Schmale, Moritz Reh or Martin Gärttner.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022