

Digital quantum simulation of an extended Agassi model: Using machine learning to disentangle its phase-diagram

Álvaro Sáiz ¹, José-Enrique García-Ramos ^{2,3}, José Miguel Arias ^{2,4}, Lucas Lamata ^{2,4} and Pedro Pérez-Fernández ^{1,2}

¹*Departamento de Física Aplicada III, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla, 41092 Sevilla, Spain*

²*Instituto Carlos I de Física Teórica y Computacional, Universidad de Granada, Fuentenueva s/n, 18071 Granada, Spain*

³*Departamento de Ciencias Integradas y Centro de Estudios Avanzados en Física, Matemática y Computación, Universidad de Huelva, 21071 Huelva, Spain*

⁴*Departamento de Física Atómica, Molecular y Nuclear, Facultad de Física, Universidad de Sevilla, Apartado 1065, E-41080 Sevilla, Spain*



(Received 14 June 2022; accepted 29 November 2022; published 21 December 2022)

A digital quantum simulation for the extended Agassi model is proposed using a quantum platform with eight trapped ions. The extended Agassi model is an analytically solvable model including both short range pairing and long range monopole-monopole interactions with applications in nuclear physics and in other many-body systems. In addition, it owns a rich phase diagram with different phases and the corresponding phase transition surfaces. The aim of this work is twofold: on one hand, to propose a quantum simulation of the model at the present limits of the trapped ions facilities and, on the other hand, to show how to use a machine learning algorithm on top of the quantum simulation to accurately determine the phase of the system. Concerning the quantum simulation, this proposal is scalable with polynomial resources to larger Agassi systems. Digital quantum simulations of nuclear physics models assisted by machine learning may enable one to outperform the fastest classical computers in determining fundamental aspects of nuclear matter.

DOI: [10.1103/PhysRevC.106.064322](https://doi.org/10.1103/PhysRevC.106.064322)

I. INTRODUCTION

Realistic physical systems, in general, and nuclear physics problems, in particular, are usually too complex to be solved exactly. Thus, many-body approximations have to be used to get reasonably approximate results. Frequently, the complexity of the problem is such that even the use of standard many-body approximations faces insuperable difficulties. To tackle this issue, algebraic models, either for fermions or bosons, having only few degrees of freedom have been developed. These can be solved analytically in some particular scenarios and numerically in general cases, even for very large systems. These facts make them very useful to check approximations and to provide solid references to test more complex and fundamental calculations. Among these types of algebraic models, with relevance in nuclear physics, one can mention the Elliot SU(3) model [1], which played a major role in the explanation of rotational structure emergence from single particle degrees of freedom in nuclei, the interacting boson model (IBM) for the study of low-lying collective states in medium and heavy nuclei [2], or the Lipkin-Meshkov-Glick (LMG) model [3] introduced first to check approximate many-body techniques in nuclear models with many nucleons interacting via a monopole-monopole force. Those are examples of simple models that have been able to permeate other fields of physics, e.g., condensed matter, quantum optics, or molecular physics, due to its simplicity and ability to capture the key aspects of a large variety of physical systems.

The Agassi model was proposed as a simple and exactly solvable version of the far more complex pairing-plus-quadrupole model of nuclear physics [4]. It includes a combination of a two-level pairing model and a monopole-monopole interaction such as the one of the LMG model. The extended Agassi model was introduced in Ref. [5] and its main difference with respect to the original model is the addition of a more general pairing interaction. The model presents a rich quantum phase diagram and, specifically, gives rise to quantum phase transitions of different nature. This makes the study of the extended Agassi model of much interest in the field of quantum phase transitions (QPT's), in particular, to test new techniques able to identify the phase of the system through the study of appropriate observables.

Quantum simulations provide a fast-developing and powerful tool to provide realizations of countless quantum systems of great interest, such as spin models, quantum chemistry, as well as QPT's [6]. In the near future, quantum simulators should be able to outperform classical computers and solve previously intractable problems. Consequently, many experimental setups are being proposed to validate the feasibility to deal with different physical models [7]. The usefulness of the Agassi model and its relevance in a variety of fields, including that of QPT's, motivates its quantum simulation. Furthermore, tools from quantum information have been recently used, as the quantum discord, to explore the phases in this model to gain insight about its structure [8].

The quantum simulation of the original Agassi model has been already proposed for a system of trapped ions with

four sites in [9], and the quantum phase diagram of the extended Agassi model has been extensively studied through a mean-field formalism [10]. In the present work, the quantum simulation of the extended Agassi model with eight sites in a system of trapped ions is worked out, as a natural followup to these previous studies. Note that the analysis is performed in a classical computer and the exact results are compared with the ones from a Trotter expansion which is used to simulate the quantum computer. In the case of the standard Agassi model with four sites, its quantum simulation only presents two phases: a symmetric and a broken symmetry phase. However, the extended Agassi model with eight sites presents a richer phase diagram with five different quantum phases. This makes the model specially issued to be studied for two reasons: (i) It is at the limit of what the current resources in facilities with trapped ions could do and (ii) its rich phase diagram allows evaluating and validating the use of machine learning methods to identify the phase of the system.

The phase/shape of a system is a property of its ground state. However, in Ref. [9] it was proved that it can also be determined through the time evolution study of selected operator matrix elements acting on a non eigenstate of the system. However, it is not always obvious how to determine the phase of the system through this procedure. We can find an alternative solution to this challenge using another technique that has seen a fast growth in the last decade, namely, machine learning. Thanks to its great versatility, it has been tested with large success in many and diverse fields of science, such as medicine [11], biology [12] and astrophysics [13]. Machine learning algorithms are specially suitable to identify features that are not immediately apparent and they work as ideal classifiers [14]. In this paper, two different machine learning approaches are presented to aid in the determination of the system phase: One simple approach using a multilayer perceptron (MLP) and another more complex and precise approach, using a convolutional neural network (CNN). Both are able to accurately predict the quantum phases of the system just taking as an input the quantum simulation of the time evolution of an appropriate operator.

The paper is organized as follows. In Sec. II, the extended Agassi model Hamiltonian is presented and then it is written in terms of the Pauli matrices using the Jordan-Wigner spin mapping approach. In Sec. III, the feasibility of a digital quantum simulation of the model is studied through its Trotterized dynamics that can be implemented efficiently with trapped-ion systems. In Sec. IV, a brief study of the QPT's of the model is presented. In Sec. V, the deep learning techniques, used for the determination of the quantum phase are presented and their results are reported. Finally, in Sec. VI, the conclusions of the paper are summarized.

II. SPIN MAPPING OF THE EXTENDED AGASSI MODEL

The Agassi model is a two-level system in which the upper level is labeled as $\sigma = 1$ and the lower level as $\sigma = -1$. Each level has Ω degeneracy, being Ω an even number [4]. One can then introduce j such that $\Omega = 2j$. Furthermore, in order to label the states within a given level, it is introduced a magnetic quantum number $m = \pm 1, \pm 2, \dots, \pm \Omega/2$. In Ref. [5], an

extended Agassi model was considered, which includes a more general pairing interaction with the additional term $-2hA_0^\dagger A_0$ [see Eq. (1)]. The Hamiltonian for the extended Agassi model is written as

$$H = \varepsilon J^0 - g \sum_{\sigma, \sigma' = -1, 1} A_\sigma^\dagger A_{\sigma'} - \frac{V}{2} [(J^+)^2 + (J^-)^2] - 2hA_0^\dagger A_0. \quad (1)$$

The operators of the Hamiltonian are all defined in terms of fermion creation and annihilation operators,

$$J^+ = \sum_{m=-j}^j c_{1,m}^\dagger c_{-1,m} = (J^-)^\dagger, \quad (2)$$

$$J^0 = \frac{1}{2} \sum_{m=-j}^j (c_{1,m}^\dagger c_{1,m} - c_{-1,m}^\dagger c_{-1,m}), \quad (3)$$

$$A_1^\dagger = \sum_{m=1}^j c_{1,m}^\dagger c_{1,-m}^\dagger = (A_1)^\dagger, \quad (4)$$

$$A_{-1}^\dagger = \sum_{m=1}^j c_{-1,m}^\dagger c_{-1,-m}^\dagger = (A_{-1})^\dagger, \quad (5)$$

$$A_0^\dagger = \sum_{m=1}^j (c_{-1,m}^\dagger c_{1,-m}^\dagger - c_{-1,-m}^\dagger c_{1,m}^\dagger) = (A_0)^\dagger, \quad (6)$$

where $c_{\pm 1,m}^\dagger$ and $c_{\pm 1,m}$ are the fermion creation and annihilation operators, respectively.

Using the Jordan-Wigner spin mapping approach [15,16], the Hamiltonian of the extended Agassi model can be mapped into Pauli matrices, which can later be experimentally implemented into a digital quantum simulator, as it will be discussed in the following section. In the following, a system with eight sites ($j = 2$) will be considered. Hence, to simplify the notation, the fermions are relabeled as

$$\begin{aligned} c_{1,2} &\rightarrow c_1, & c_{-1,2} &\rightarrow c_5, \\ c_{1,1} &\rightarrow c_2, & c_{-1,1} &\rightarrow c_6, \\ c_{1,-1} &\rightarrow c_3, & c_{-1,-1} &\rightarrow c_7, \\ c_{1,-2} &\rightarrow c_4, & c_{-1,-2} &\rightarrow c_8, \end{aligned} \quad (7)$$

and the corresponding relabeling for the creation operators holds. The spin image of the above fermions is

$$\begin{aligned} c_i^\dagger &= I_1 \otimes \dots \otimes I_{i-1} \otimes \sigma_i^+ \otimes \sigma_{i+1}^z \otimes \dots \otimes \sigma_N^z, \\ c_i &= I_1 \otimes \dots \otimes I_{i-1} \otimes \sigma_i^- \otimes \sigma_{i+1}^z \otimes \dots \otimes \sigma_N^z, \end{aligned} \quad (8)$$

where σ^\pm are written in terms of the Pauli matrices in the standard way

$$\sigma^+ = \frac{\sigma^x + i\sigma^y}{2}, \quad \sigma^- = \frac{\sigma^x - i\sigma^y}{2}, \quad (9)$$

$\sigma^{x,y,z}$ are, respectively,

$$\sigma^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (10)$$

\otimes stands for the Kronecker product and I for the identity operator.

The Hamiltonian can be written in terms of strings of tensor products of Pauli matrices. Indeed, there are terms which only depend on the Pauli matrix σ^z , and there are others that need to be written in terms of the Pauli matrices σ^x and σ^y , so that they can be mapped into the corresponding quantum gates. Once σ^+ and σ^- are decomposed in terms of σ^x and σ^y , each of the tensor product strings will give rise to 8 different terms. The total number of terms scales polynomially with the number of sites.

III. FEASIBILITY OF THE IMPLEMENTATION

Given a certain time-independent Hamiltonian, the evolution operator is written as

$$U(t) = e^{-itH}, \quad (11)$$

considering $\hbar = 1$.

For the case of a Hamiltonian written as the sum of k different noncommuting parts, the evolution operator can be approached using the so-called Trotter expansion as

$$U(t, n)_T = \left(\prod_k e^{-itH_k/n} \right)^n \approx U(t). \quad (12)$$

The study of the many Hamiltonian terms arising from the Jordan-Wigner mapping shows that many of them commute with each other. Hence, this can be exploited by adding these terms together in a total of eight noncommuting groups [$k = 8$ in Eq. (12)], saving computational resources. Each of these groups can then be implemented with trapped-ion systems using a combination of single and multiqubit (Mølmer-Sørensen) gates and simulation of the terms within each single group can be carried out sequentially without digital error [17–19].

The accuracy of this approximation can be measured by its fidelity, defined as

$$F(t, n_T) = |\langle \phi | U(t, n_T)_T^\dagger U(t) | \phi \rangle|^2. \quad (13)$$

This observable depends on time, t , and on the number of Trotter steps n_T . The considered initial state to which we apply the evolution will be along the rest of the paper $|\downarrow_1 \downarrow_2 \downarrow_3 \downarrow_4 \uparrow_5 \uparrow_6 \uparrow_7 \uparrow_8\rangle$, where $|\downarrow_i\rangle$ and $|\uparrow_i\rangle$ stand for the eigenvectors of σ_i^z with eigenvalues -1 and 1 , respectively. Hence, the trial state is not a Hamiltonian eigenstate, except for $V = g = h = 0$.

Figure 1 shows the fidelity as a function of both t and n_T . In a previously studied case with four sites ($j = 1$), it was seen that already $n_T = 5$ Trotter steps provided enough accuracy [9]. Here, the fidelity drops faster as time increases and for $n_T = 5$ the approximation only remains accurate at small times ($t < 1$) [panel (a)], but this drop is greatly reduced for $n_T = 15$ [panel (c)]. This is expected, since eight sites are considered and, therefore, there exist many more terms that need to be considered for the Trotterized Hamiltonian, which in turn means that a significantly higher amount of Trotter steps are required to attain the same accuracy in a given time range. In panels (b) and (d) the value of the fidelity is depicted

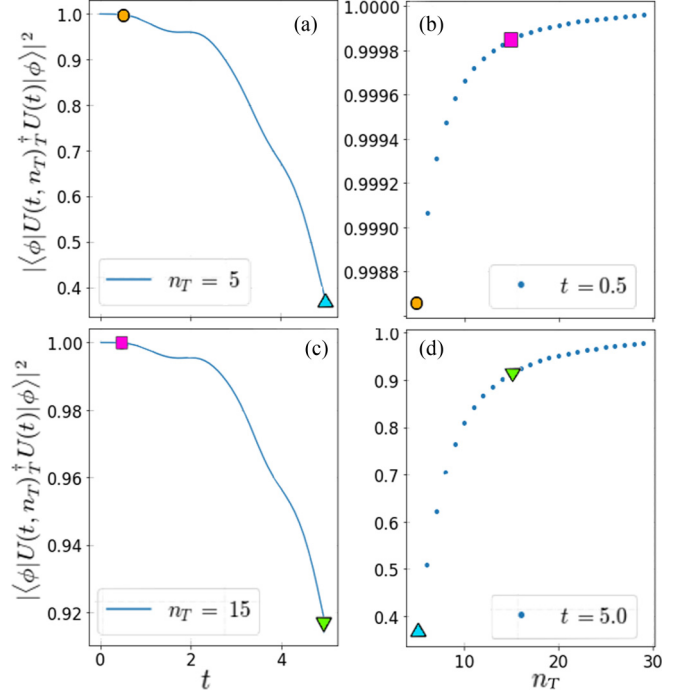


FIG. 1. Fidelity as a function of t for (a) $n_T = 5$ and (c) $n_T = 15$ and as a function of n_T for (b) $t = 0.5$ and (d) $t = 5$. The initial state in all cases is $|\downarrow_1 \downarrow_2 \downarrow_3 \downarrow_4 \uparrow_5 \uparrow_6 \uparrow_7 \uparrow_8\rangle$ and the Hamiltonian parameters are $\varepsilon = 1$, $g = 0.25$, $V = 0.25$, and $h = 0.25$. The highlighted points correspond to $n_T = 5, 15$ and $t = 0.5, 5$, respectively.

as a function of the Trotter steps, n_T , for two different values of time

Figure 2 shows as an example the survival probability, $|\langle \phi(t) | \phi(0) \rangle|^2$, computed for both the exact and the Trotterized evolution operator with different values of $V = 0.25$ (top row) and $V = 1.5$ (bottom row) and $n_T = 5, 15$ [panels (a) and (b)] and $n_T = 5, 30$ [panels (c) and (d)], keeping constant $g = 0.5$ and $h = 1.5$. For a small value of $V = 0.25$, a good fidelity can be achieved with only $n_T = 5$ Trotter steps. However, this is not true in general. There are situations where an accurate study require a larger number of Trotter steps as it is seen in Fig. 2, panels (c) and (d).

The implementation of the simulation is efficient, that is, it scales with polynomial resources. However, considering the need of two Mølmer-Sørensen gates for each tensorial product and two single-qubit gates for each σ^x and σ^y to be changed into a σ^z and back (assuming σ^z as the basis), around two thousand quantum gates would be needed to implement this model for $j = 2$. It is important to note that the aggregated error from all the gates will then increase with the number of Trotter steps needed for a precise approximation. Nevertheless, the estimations seem to agree that this proposal should be achievable in the near future as a good progress is being made in scaling trapped ion systems to the hundreds or thousands of qubits [20,21]. Some approaches with promising results are the implementation of high fidelity fast gates [22,23] and two-dimensional ion traps, since the fidelities in each dimension are independent from one another [24]. The use of a higher-order Trotter expansion would also help

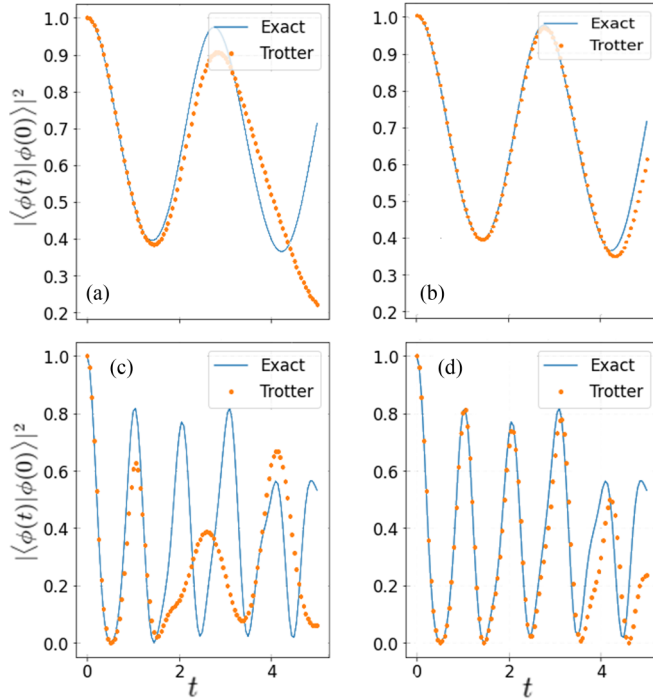


FIG. 2. Survival probability $|\langle\phi(t)|\phi(0)\rangle|^2$ as a function of t using the exact and the Trotterized evolution operator. In all cases, the initial state is $|\downarrow_1\downarrow_2\downarrow_3\downarrow_4\uparrow_5\uparrow_6\uparrow_7\uparrow_8\rangle$ and $\varepsilon = 1$, $g = 0.5$, and $h = 1.5$. Top row: $V = 0.25$ and (a) $n_T = 5$ and (b) $n_T = 15$. Bottom row: $V = 1.5$ and (c) $n_T = 5$ and (d) $n_T = 30$.

reducing the required number of Trotter steps for an accurate approximation [15].

IV. EXPLORING PHASE TRANSITIONS

The original Agassi model presents a rich quantum phase diagram with a symmetric, a parity broken phase and a superconducting one [25]. By adding the additional pairing interaction term of the extended Agassi model, the quantum phase diagram becomes even richer. In order to explore the phase diagram and the appearance of QPT's it is customary to rescale the parameters of the Hamiltonian (1) by introducing the new parameters χ , Σ , and Λ ,

$$V = \frac{\varepsilon\chi}{2j-1}, \quad g = \frac{\varepsilon\Sigma}{2j-1}, \quad h = \frac{\varepsilon\Lambda}{2j-1}. \quad (14)$$

The phase diagram of the extended Agassi model presents five different quantum phases, as shown in Fig. 3. They appear in the following areas of the phase diagram (see Ref. [10] for a detailed study of the quantum phases of the system):

- (i) Symmetric or spherical solution:
 $\chi < 1$, $\Sigma < 1$, $\Lambda < 1$.
- (ii) Hartree-Fock (HF) deformed solution:
 $\chi > 1$, $\Sigma < \chi$, $\Lambda < \frac{1+\chi^2}{2\chi}$.
- (iii) Bardeen-Cooper-Schrieffer (BCS) deformed solution:
 $\chi < \Sigma$, $\Sigma > 1$, $\Lambda < \frac{1+\Sigma^2}{2\Sigma}$.

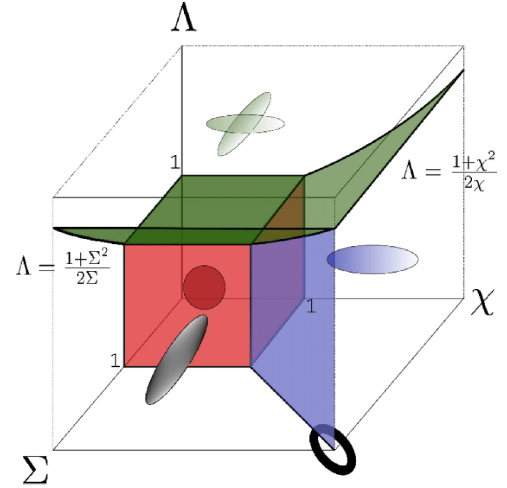


FIG. 3. Graphical representation of the phase diagram of the extended Agassi Hamiltonian. Red vertical planes represent second order QPT surfaces. The green surface ($\Lambda = 1$ for $\chi < 1$ and $\Sigma < 1$, $\Lambda = \frac{1+\chi^2}{2\chi}$ for $\chi > \Sigma$, and $\Lambda = \frac{1+\Sigma^2}{2\Sigma}$ for $\chi < \Sigma$) and the blue vertical one ($\chi = \Sigma$ and $\Lambda < \frac{1+\Sigma^2}{2\Sigma}$) correspond to first order critical surfaces. Red sphere, blue oval, black oval, black thick oval, and crossed green ovals correspond to the symmetric solution, the HF deformed solution, the BCS deformed solution, the closed valley solution, and the HF-BCS deformed solution, respectively. Figure adapted from [5].

- (iv) Combined HF-BCS deformed solution:

$$\Lambda > 1 \text{ for } \chi < 1 \text{ and } \Sigma < 1; \quad \Lambda > \frac{1+\chi^2}{2\chi} \text{ for } \chi > \Sigma > 1; \quad \Lambda > \frac{1+\Sigma^2}{2\Sigma} \text{ for } \Sigma > \chi > 1.$$

- (v) Closed valley solution: $\chi = \Sigma$, $\Lambda < \frac{1+\Sigma^2}{2\Sigma}$.

It is important to note that this phase diagram has been obtained using a mean-field approach, which is only valid for the large N limit, where N is the number of sites, and it allows to calculate the value of the order parameters as a function of the control parameters. Here, $N = 8$ and therefore, finite- N effects become relevant, causing the QPT's to be diffuse and harder to be distinguished.

In previous studies, time evolution of a correlation function has been successfully used as a tool to explore the QPT's of the standard Agassi model with $j = 1$ [9]. In the rest of this work, we will use precisely this observable to study the phase diagram of the extended Agassi model with $j = 2$. A correlation function between two given sites can be defined as

$$C_{\alpha,\beta}(i, j) = \langle\sigma_i^\alpha \otimes \sigma_j^\beta\rangle - \langle\sigma_i^\alpha\rangle\langle\sigma_j^\beta\rangle, \quad (15)$$

where $\alpha, \beta = x, y, \text{ or } z$. In our case, we will focus on a correlation function for sites 1 and 2 and $\alpha = \beta = z$,

$$C_z(1, 2) \equiv \langle\sigma_1^z \otimes \sigma_2^z\rangle - \langle\sigma_1^z\rangle\langle\sigma_2^z\rangle. \quad (16)$$

Figure 4(a) shows the systematics of the correlation function $C_z(1, 2)$ as we move through a line of fixed $\chi = 1.5$ and $\Lambda = 0$ from the HF phase ($\Sigma < \chi$) to the BCS phase ($\Sigma > \chi$) as a function of time. The correlation function oscillates with time and, despite the fact that the transition is not apparent, if we move away from the critical surface, e.g., at slices A

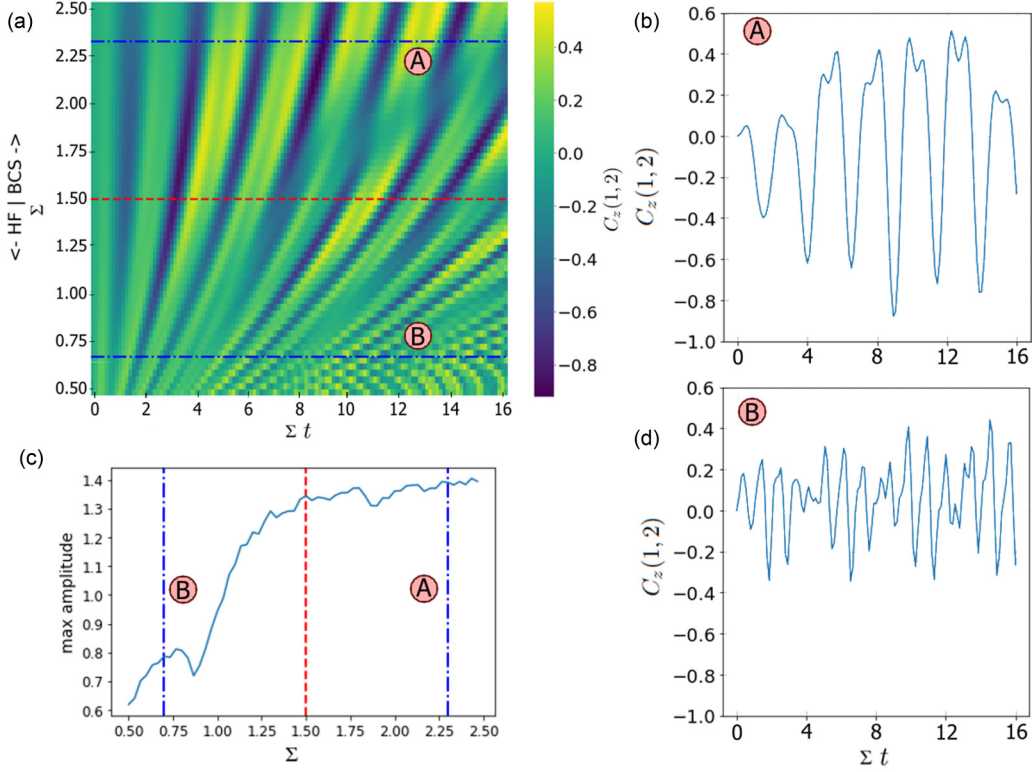


FIG. 4. (a) Time evolution of the correlation function $C_z(1, 2)$ from $\Sigma = 0.5$ to $\Sigma = 2.5$, with $\chi = 1.5$, $\Lambda = 0$, and an initial state $|\downarrow_1\downarrow_2\downarrow_3\downarrow_4\uparrow_5\uparrow_6\uparrow_7\uparrow_8\rangle$. The phase transition point $\Sigma = \chi = 1.5$ is marked by a dashed red line and the points $\chi = 1.5$, $\Sigma = 0.7$ (slice B) and $\chi = 1.5$, $\Sigma = 2.3$ (slice A) with blue dashed lines. (c) Maximum value of the amplitude of the oscillations of $C_z(1, 2)$ as a function of Σ value. (b) Time evolution of $C_z(1, 2)$ for $\Sigma = 2.3$ (slice A) and (d) for $\Sigma = 0.7$ (slice B).

and B, we can see a clear difference in the amplitude of these oscillations. Hence, it increases as one moves towards the BCS phase, as shown in panel (c) of Fig. 4. In panels (b) and (d), the time evolution of the correlation function for the points A and B are depicted. These two panels clearly show how the amplitude and the period of the oscillation greatly change depending on the value of the control parameter. In the Appendix, other examples are gathered showing the variety of behaviors that one can get depending on the position in the phase diagram. Other correlation functions between different sites, applied to the same and/or different initial states, yield similar results (not shown in this work), except when they vanish, which proves that correlation functions are good observables to obtain information about the phase of the system without the need of computing the ground state.

Even in the case where differences between quantum phases can be appreciated in the time evolution, it is hard, and sometimes impossible, to visually identify the quantum phase of the system, specially when looking at a single point in the phase space. Hence, two machine learning models will be trained for quantum phase classification using the time dynamics of the correlation function $C_z(1, 2)$. The benefit of using machine learning methods for classification is that, given the data to classify, the method can determine itself the optimal procedure and, moreover, it will be able to accurately label not previously considered cases.

V. MACHINE LEARNING METHODS FOR CLASSIFICATION

A. Multilayer perceptron

A multilayer perceptron (MLP) is a supervised machine learning algorithm that learns a nonlinear function which, given a set of features as input $X = x_1, x_2, \dots, x_m$, is able to approximately predict a target output y . The algorithm learns this function via gradient descent, modifying its weights by minimizing a loss function based on how accurate the predictions y' are in relation to the real target values y , which are known in the training process, as with any supervised method [26]. Once optimized, it can reliably predict the target $y' \approx y$ for inputs for which the real target is not already known.

The implementation of this model is very simple thanks to the existing standardized machine learning software packages, e.g., SCIKIT-LEARN [27], but first it is needed to define a set of training and test data for the model to learn with. For this purpose, we compute the exact as well as the Trotterized (using a number of Trotter steps $n_T = 6$) time evolution of the correlation operator $C_z(1, 2)$ for a large set of points throughout the phase diagram. We create a mesh of points with the parameters χ , Σ , and Λ in the interval $[0, 2]$ with 21 steps each for a total of 9261 points. For each of them, we compute the time dynamics for 100 time steps. This information together with the knowledge of the phase diagram (see Sec. IV) feeds the model as part of the training process.

Once the dataset is ready, it is randomly split into two sets. One set, the training set, will be used to train the model, while the other, the test set, will be used to test the accuracy of the model. It is compulsory for the model to be tested with data not used during the training process to be sure that the algorithm is not simply remembering the already used data. Therefore, it is important to have enough data points for testing and to reach an acceptable metric, while keeping as many points as possible for training. In most cases, to use from 10 to 20 % of data for testing is enough. In our case, we will use 10% of data points for testing.

SCIKIT-LEARN provides us with several optimization algorithms to perform training on the data. In this particular case, the default *Adam* solver gives optimal results. *Adam* solver is an optimization algorithm that is based on the stochastic gradient descent [28]. It combines the advantages of two other extensions of the stochastic gradient descent known as adaptive gradient algorithm (AdaGrad) and root mean square propagation (RMSProp). For a general overview on the theoretical and practical aspects of the learning algorithms based on stochastic gradient descent, see, e.g., Ref. [29].

The model actually computes the probability for each quantum phase and classifies the data based on which one is the most probable. The classification with the MLP correctly predicts the quantum phase in the 96.7% of the points in the test data when trained and tested with the exact time evolution. When the MLP is trained and tested with the Trotterized time evolution, it has an accuracy of 93.4% ($n_T = 6$). These results are very valuable, since it proves that the quantum phase can be determined using the correlation function without the need of computing the ground state, but it also shows that they can be identified despite using the Trotter approximation with a relative small number of steps, for which the agreement with the exact results is limited. Note that when reducing the number of Trotter steps, the aggregated error of the quantum gates is reduced too.

In Fig. 5, the results for the MLP method are presented through different trajectories in the parameter space of the model. Generally, the incorrect predictions occur near the critical points, which is expected, since the QPT's are not clearly defined because of finite- N effects ($N = 8$ in our case). Note that at the critical points (denoted by a dashed line), two or three quantum phases can coexist, therefore, it is considered that the model correctly classifies the quantum phase of the system when it predicts either of the two phases, except in the case of the closed valley solution, where that phase is considered the only valid answer. This is done to ensure that the model learns to identify this specific phase instead of choosing any of the other two phases with which it coexists.

Perhaps the most surprising case is, in fact, the one regarding this solution. In the results portrayed in Fig. 5(d), the model is able to predict the quantum phase of the system as it crosses the closed valley solution at $\chi = \Sigma = 1.5$ when trained with the exact solution. As mentioned, the model is not completely accurate in some cases [as can be seen from Fig. 5(c)-Trotter], but the QPT's are still fairly clear. In next section we will present a more accurate machine learning model.

B. Convolutional neural network

Convolutional Neural Networks (CNN) are types of deep learning algorithms that are well suited for image recognition. Their efficacy for extracting features and learning to classify images has been proven in multiple instances [30–32]. In our case, instead of feeding two-dimensional (2D) arrays of pixel values, one-dimensional (1D) arrays containing the time dynamics of the correlation function will be used. When working with 2D CNN, the data have actually the shape (x, y, z) , being z the value contained in the pixel, while x and y are its positions. 1D CNN work effectively in the same way, just with data of shape (x, y) , being y the value of the variable of interest at time x . The general structure of a CNN includes more than one convolution layer, so that one can extract higher level features. They also include additional layers, e.g., the pooling and the fully connected layers, each with a different purpose. For the present model the used layers are

- (i) **Convolution layer:** The layer responsible of performing the convolution operation.
- (ii) **Activation layer:** The layer that applies the activation function together with the filter of the convolution layer.
- (iii) **Pooling layer:** The pooling layer performs a dimension reduction of the data, collapsing data by connecting clusters of neurons to a single neuron each.
- (iv) **Dropout layer:** This optional layer temporarily deactivates, or *drops out*, randomly selected training parameters from the previous layer that has trainable parameters. Its goal is to avoid the “overfitting”.
- (v) **Fully connected layer:** Also known as *dense* layers, they connect every neuron of the input to every neuron of the output.
- (vi) **Softmax layer:** This layer is a fully connected or dense layer that applies a specific kind of activation function, called a *softmax* function, which is a normalized exponential function.

A fast and efficient way of implementing such a model is through the use of the *Keras* library [33], a deep learning application programming interface built on top of the machine learning platform *Tensorflow* [34]. *Keras* provides the ability to apply all the necessary layers sequentially, giving the model the final structure shown in Table I. Four cycles are applied, consisting on convolution, activation, pooling and dropout, allowing for a hierarchical decomposition of the input and the extraction of higher level features, as previously mentioned. Similarly, five additional fully connected (dense) layers are applied before the final output layer. The model includes a layer named “Flatten”. This layer simply reshapes the output of the previous layer into a single dimension array so that it can be passed as the input to the dense layer.

Table I shows, alongside the layers, the shape of the output and the number of trainable parameters of each layer. Note that only the convolution and dense layers have trainable parameters and the rest of the layers simply apply a chosen transformation to the data which is never changed throughout the training process. The convolution layers are applied with

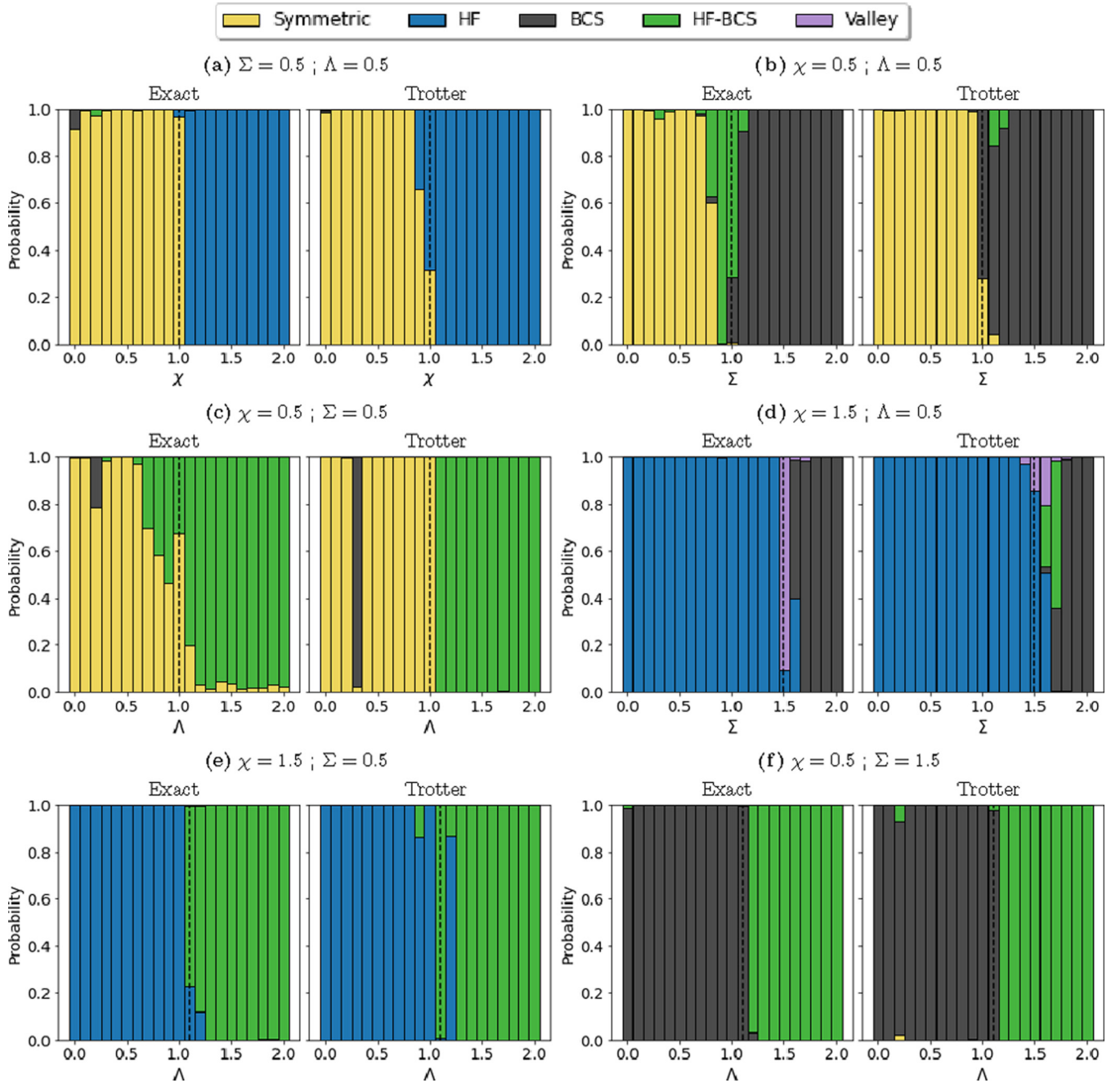


FIG. 5. Quantum phase prediction of the system via the MLP method applied to the $C_z(1, 2)$ correlation function. The different panels show the probability for the system of being in a given phase for specific values of the Hamiltonian parameters. Left panels correspond to the results using the exact time evolution, while right ones using the Trotter approach with $n_T = 6$. In all cases, the initial state $|\downarrow_1 \downarrow_2 \downarrow_3 \downarrow_4 \uparrow_5 \uparrow_6 \uparrow_7 \uparrow_8\rangle$ is used. (a) $\Sigma = 0.5$ and $\Lambda = 0.5$ moving from $\chi = 0$ (symmetric phase) to $\chi = 2$ (HF phase); (b) $\chi = 0.5$ and $\Lambda = 0.5$ moving from $\Sigma = 0$ (symmetric phase) to $\Sigma = 2$ (BCS phase); (c) $\chi = 0.5$ and $\Sigma = 0.5$ moving from $\Lambda = 0$ (symmetric phase) to $\Lambda = 2$ (combined HF-BCS phase); (d) $\chi = 1.5$ and $\Lambda = 0.5$ moving from $\Sigma = 0$ (HF phase) to $\Sigma = 2$ (BCS); (e) $\chi = 1.5$ and $\Sigma = 0.5$ moving from $\Lambda = 0$ (HF phase) to $\Lambda = 2$ (combined HF-BCS phase); (f) $\chi = 0.5$ and $\Sigma = 1.5$ moving from $\Lambda = 0$ (BCS phase) to $\Lambda = 2$ (combined HF-BCS phase). The dashed black line in each panel marks the theoretical critical point.

padding to match the first dimension of the output with the size of the corresponding input. The second dimension will be equal to the number of applied filters, 32, 64, 128, and 256 for the first, second, third, and fourth convolution layers, respectively. For convolution layers, the trainable parameters define each of the filters. Since the filters have a size 3, each

one will have three parameters or weights for each of the input channels, plus one bias weight. This number, multiplied by the number of applied filters, gives the total number of trainable parameters shown on the right column of Table I. In the case of the dense layers, each connection has its own weight and all input neurons are connected to all output ones, so the

TABLE I. Number of parameters and shape of the output of the used convolution layers.

Layer	Output shape	# of parameters
Convolution 1D (1)	100×32	128
Leaky ReLU (1)	100×32	0
Avg. pooling 1D (1)	34×32	0
Spatial dropout (1)	34×32	0
Convolution 1D (2)	34×64	6208
Leaky ReLU (2)	34×64	0
Avg. pooling 1D (2)	12×64	0
Spatial dropout (2)	12×64	0
Convolution 1D (3)	12×128	24704
Leaky ReLU (3)	12×128	0
Avg. pooling 1D (3)	4×128	0
Spatial dropout (3)	4×128	0
Convolution 1D (4)	4×256	98560
Leaky ReLU (4)	4×256	0
Avg. pooling 1D (4)	2×256	0
Spatial dropout (4)	2×256	0
Flatten	512	0
Dense (1)	512	262656
Leaky ReLU (1)	512	0
Dropout (1)	512	0
Dense (2)	512	262656
Leaky ReLU (2)	512	0
Dropout (2)	512	0
Dense (3)	512	262656
Leaky ReLU (3)	512	0
Dropout (3)	512	0
Dense (4)	512	262656
Leaky ReLU (4)	512	0
Dropout (4)	512	0
Dense (5)	512	262656
Leaky ReLU (5)	512	0
Dropout (5)	512	0
Softmax	5	2562
Total parameters		1445445

total number of parameters will be the size of the input times the size of the output, plus one bias weight for each output neuron.

Once the layer structure is defined, the model should be trained in a similar manner to the MLP case. The data are divided in three sets, namely, the test, the training and the validation set. The training one is used to get the trainable parameters, the validation one to check the consistency of the results and eventually to modify the network layer structure and hyperparameters (such as the number of neurons or filters), finally, the test dataset is used to report the prediction accuracy of the process. When working with neural networks, a common problem is the overfitting of the model, specially when the available data for training is limited. The model may “memorize” the training data and achieve good results for that input specifically, but becomes unable to generalize and predict correctly for data outside of the training set. In Fig. 7, training and validation accuracy are shown across the epochs of the training process. In the training case, the accuracy in predicting the phase is computed using the training data, while

in the validation accuracy a separate set of data is used, i.e., the validation set. Note that neither of these sets contain the test dataset. One can see that both training and validation accuracies steadily increase at a similar pace, which suggests that the model is not overfitted. In the case of overfitting, the training accuracy would increase faster than the validation accuracy [14] or the latter would stagnate at an earlier epoch, which is not the case. The total number of training epochs considered in this work was 200, as further cycles do not significantly increase the accuracy, assuring that the model has converged. In our case, we get an accuracy in the prediction of the phase, using the test dataset, of a 98.7% (corresponding to the exact evolution). Note that this reported accuracy do not use data neither from the validation dataset nor from the validation one.

The increase in accuracy from 96.7% using the MLP to 98.7% using the CNN model is small, but significant. In particular, now the model has a better performance in the most complicated regions, i.e., near the QPT’s. This is clearly seen in Fig. 6, especially for the cases b), c), and d) where the QPT becomes much more clear. The main reason is that not only the model is more precise, but also that inaccuracies are mainly limited to points with values of χ and/or Σ equal to or near zero, with only a few cases occurring at the critical points. The reason for these inaccuracies is that the amplitude of the correlation function is very small when both χ and Σ are small, and as such, it presents very little variation as one moves across Λ . Hence, this inaccuracy is mostly present between the symmetric and the combined HF-BCS phases. In both cases, the time evolution is quite similar but the model seems to slightly favor the combined HF-BCS phase as the amplitude is generally lower in this case.

It is worth to mention once again, that the results are robust to the error introduced from the Trotter approximation even in the case of a relatively small number of Trotter steps. As a matter of fact, the model was trained and tested using the Trotterized time dynamics of the correlation function $C_c(1, 2)$ computed for $n_T = 6$ and $n_T = 20$ with accuracies of 99.2% and 98.9%, respectively, to be compared with 98.7% of the exact evolution.

As a conclusion of this section, we can say that the accuracy in determining the phase is larger for the CNN than for the MLP method, which is expected, specially in regions nearby the QPT’s. Moreover, a really appealing fact is that the accuracy of the procedure is even larger when using the Trotter approximation with a small value of n_T , which has clear practical advantages. A possible explanation is that the larger oscillations observed in the approximate evolution, compared with the exact one, somehow helps the CNN to better predict the most problematic situations (see Appendix). The use of a relatively small number of Trotter steps reduce the number of needed quantum gates and, therefore, will reduce the experimental error. Because the above reasons, the practical implementation of this algorithm in a real quantum computer seems to be feasible and potentially valid to determine the phase of the system. As a matter of fact, the impact of noise on the learning accuracy, assuming it is small, it is not expected to affect the phase prediction.

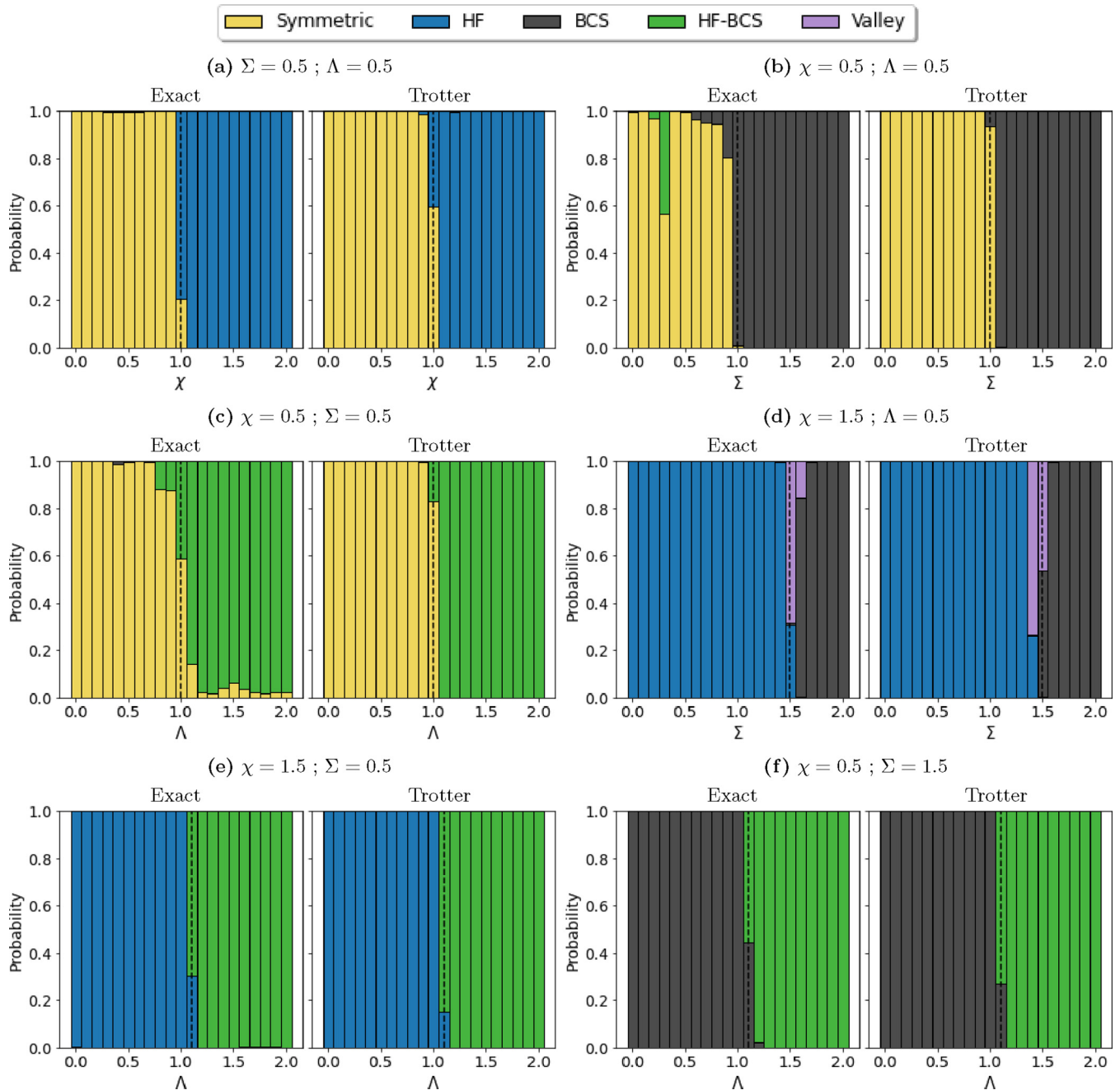


FIG. 6. Same caption as Fig 5 but using the CNN.

C. Further exploration

Other correlation functions, as well as different initial states were studied. There is a lot of combinations of different possible correlation functions and different initial states to evolve. On the one hand, it is obvious that equivalent correlation functions will give the same results, in this case $C_z(1, 2) = C_z(1, 3) = C_z(2, 4) = C_z(3, 4) = C_z(5, 6) = C_z(5, 8) = C_z(6, 8) = C_z(7, 8)$ and their conjugates for the initial state $|\downarrow_1 \downarrow_2 \downarrow_3 \downarrow_4 \uparrow_5 \uparrow_6 \uparrow_7 \uparrow_8\rangle$. On the other hand, the rest of the studied cases give comparable, although less accurate results. The CNN trained and tested on $C_z(1, 4)$ for the initial state $|\downarrow_1 \downarrow_2 \downarrow_3 \downarrow_4 \uparrow_5 \uparrow_6 \uparrow_7 \uparrow_8\rangle$ obtained an accuracy of 94.2%,

while using the correlation function $C_z(1, 2)$ for a different initial state $|\uparrow_1 \uparrow_2 \uparrow_3 \downarrow_4 \uparrow_5 \downarrow_6 \downarrow_7 \downarrow_8\rangle$ achieved only an accuracy of 86.5%. Initial states that are eigenstates of one or more of the terms of the Hamiltonian are unsuitable as the information of the time evolution is partially (or totally) lost. Similarly, some correlation functions may vanish, for instance, σ_x or σ_y when using initial states in the z basis. Even when computing the time dynamics of the correlation function $C_x(1, 2)$ for an initial state $|\downarrow_1^x \downarrow_2^x \downarrow_3^x \downarrow_4^x \uparrow_5^x \uparrow_6^x \uparrow_7^x \uparrow_8^x\rangle$ (where $|\downarrow_i^x\rangle$ and $|\uparrow_i^x\rangle$ stand for the eigenvectors of σ_i^x with eigenvalues -1 and 1 , respectively) which does not vanish, the CNN was unable of providing valuable results, i.e., it fails in finding the phases of the system.

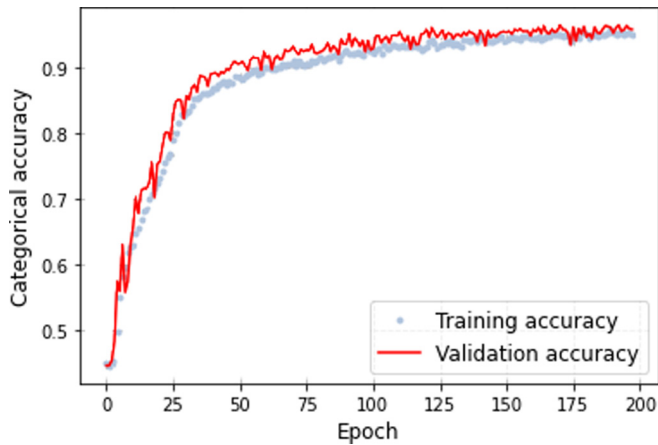


FIG. 7. Training and test accuracy of the CNN as a function of the number of epochs used in the training process.

At the moment, it is not evident how to discern which correlation function would be optimal to train the model and to classify the quantum phases of the system, apart of the visual inspection of the correlation function dynamics at different points in the phase space. Those with the most obvious differences to the naked eye seem to be also the ones that the CNN is able to better classify.

VI. SUMMARY AND CONCLUSIONS

In this work, an experimental setup for the quantum simulation of the extended Agassi model with $N = 8$ sites was proposed. The feasibility of this implementation with a Jordan-Wigner spin-mapping approach using trapped ions was studied. The estimations suggest that this protocol should be achievable in the near future as the fidelity of quantum gates will increase and new technologies will be developed.

The rich quantum phase diagram of the extended Agassi model has been studied, showing that the time dynamics of a quantum correlation function, calculated with a particular initial state, can give information about the quantum phase of the system even though the quantum phase transitions regions are diffuse due to the finite size of the system. Two supervised machine learning models were trained to extract this information and work as classifiers of the quantum phase of the system, namely, a simple MLP and a more complex CNN. The MLP has been proven to be an appropriate tool to explore the QPT's of the system, with an accuracy in determining the phase of the system of a 96.7%. The CNN achieved a significantly better result with an accuracy of a 98.4% in determining the phase and its performance is much better than the one of the MLP case to correctly classify the quantum phase near the critical points, giving a cleaner and more precise view of the quantum phase diagram. As previously mentioned, for $j = 2$, finite N effects make the differences between phases less apparent, but by increasing j (and therefore N), the previously blurred QPT's become better defined. For that reason, it is remarkable that the machine learning procedure is capable of classifying the quantum phase of the system for a small size of the system N , as well as for a low value of n_T Trotter steps. From an

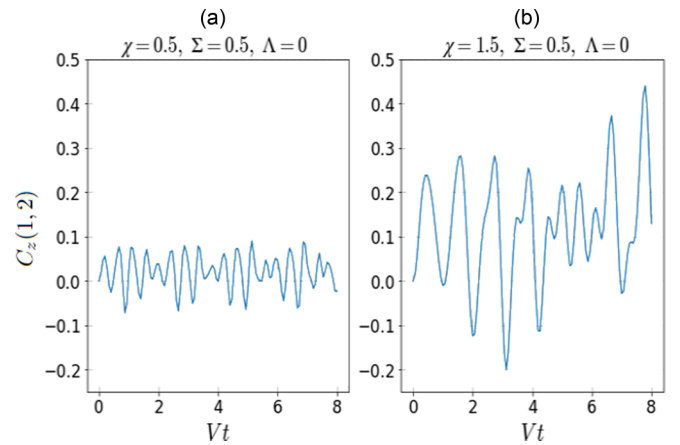


FIG. 8. Time evolution of the correlation function $C_z(1, 2)$ for (a) $\chi = 0.5$ and (b) $\chi = 1.5$ with $\Sigma = 0.5$ and $\Lambda = 0$ and an initial state $|\downarrow_1 \downarrow_2 \downarrow_3 \downarrow_4 \uparrow_5 \uparrow_6 \uparrow_7 \uparrow_8\rangle$.

experimental point of view, keeping both quantities low makes the quantum simulation of the system much feasible with current technology.

One must note that supervised machine learning algorithms require previous knowledge of the target system. In this case, the quantum phase diagram of the extended Agassi model is known thanks to mean field theory, and therefore, one can compute the necessary training data for the employed algorithms through an exact diagonalization in a classical computer. Naturally, cases where previous knowledge of the quantum phases is not available would be of special interest, for which different methods would be necessary. Despite limited by this fact, the presented approach serves as a proof of concept for future development, showing that machine learning algorithms can be powerful tools to extract quantum phase information from the time evolution of certain observables. It is then reasonable to think that, inspired by this idea, one could be able to obtain partial information of the phase diagram of an unexplored Hamiltonian using training data, in a more general manner, from known cases. Another option is

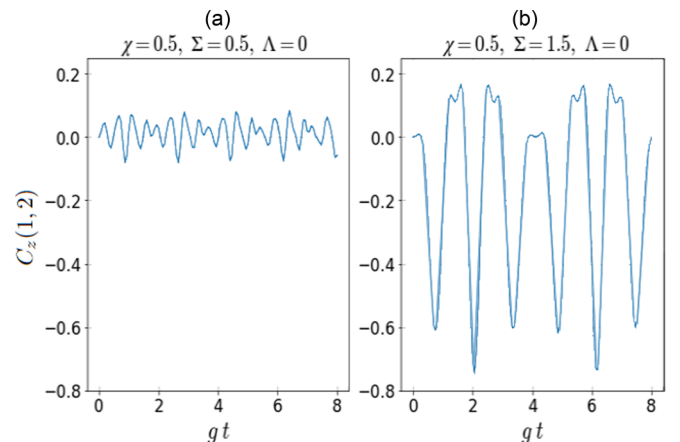


FIG. 9. Same caption as Fig. 8 but with $\chi = 0.5$, $\Lambda = 0$ and (a) $\Sigma = 0.5$ and (b) $\Sigma = 1.5$.

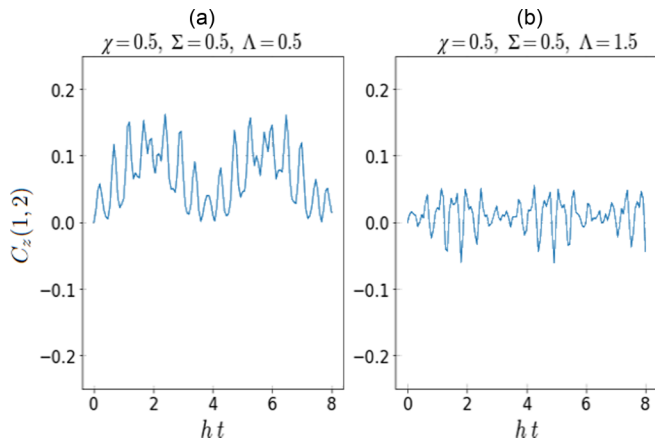


FIG. 10. Same caption as Fig. 8 but with (a) $\Lambda = 0.5$ and (b) $\Lambda = 1.5$, $\chi = 0.5$, and $\Sigma = 0.5$.

to make use of unsupervised machine learning algorithms to deduce the quantum phases of a generic Hamiltonian without any prior knowledge of such phases, though this supposes a new set of challenges. Namely, unsupervised classification algorithms rely somehow on grouping the data accordingly on how similar they are, but cases near critical points are expected to present a similar behavior. Therefore, it is unlikely that popular unsupervised algorithms would be able to give a precise prediction of the position of the QPT's, but could still be useful to get a general qualitative idea of the quantum phase diagram of a model.

The use of the time evolution of a correlation function to determine the phase of the Agassi model was already introduced in Ref. [9], but in the present work, its use has been improved through the implementation of machine learning models. It is worthy to mention that the phase of a system is a property of its ground state, however the time evolution of a given observable depends on the whole Hilbert space and on its energy spectrum. The quantum phase of the system strongly determine its spectrum and, therefore, will control the time evolution of any operator. An extreme case is the presence of excited state quantum phase transitions (ESQPTs) [35] in the spectrum which can be understood as a prolongation of the ground-state phase transition to the excited states and suppose a pileup of states at a particular energy, corresponding to a singularity in the density of states in the thermodynamic limit. The presence of a ESQPT in the spectrum has deep consequences on the time evolution of a non-eigenstate, as can be readily observed in works on the evolution of the fidelity [36] or the out-of-time-order correlators [37].

ACKNOWLEDGMENTS

This work was partially supported by the Consejería de Economía, Conocimiento, Empresas y Universidad de la Junta de Andalucía (Spain) under Groups No. FQM-160,

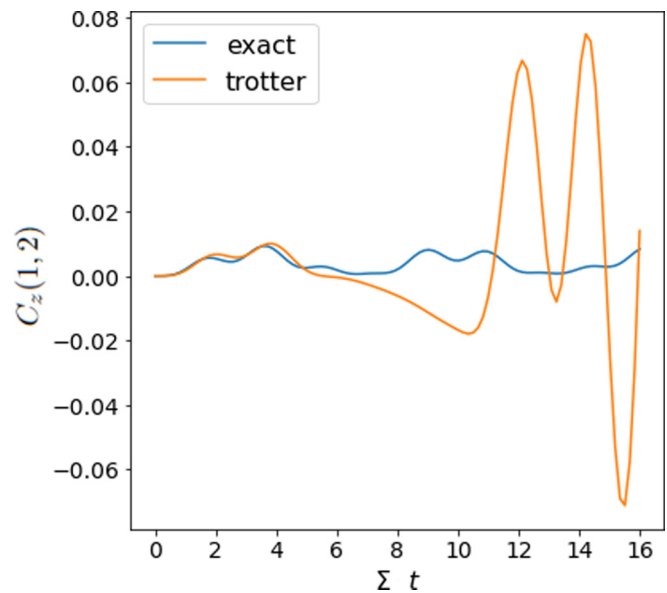


FIG. 11. Time evolution of the correlation function $C_z(1, 2)$ with parameters $\chi = 0.0$, $\Sigma = 0.1$ and $\Lambda = 0.6$ using the exact evolution (in blue) and the Trotter expansion (in orange).

FQM-177, and FQM-370, and under projects no. P20-00617, P20-00764, P20-01247, UHU-1262561, and US-1380840; by Grants No. PGC2018-095113-B-I00, PID2019-104002GB-C21, PID2019-104002GB-C22, and PID2020-114687GB-I00 funded by MCIN/AEI/10.13039/50110001103 and “ERDF A way of making Europe” and by ERDF, ref. SOMM17/6105/UGR. Resources supporting this work were provided by the CEAFCM and Universidad de Huelva High Performance Computer (HPC@UHU) funded by ERDF/MINECO Project No. UNHU-15CE-2848.

APPENDIX: TIME EVOLUTION OF $C_z(1, 2)$ FOR SELECTED CASES

Time evolution of the correlation function $C_z(1, 2)$ Eq. (16) for selected points of the phase diagram of the extended Agassi model. See Figs. 8–10. It can be seen that the time dynamics of this correlation function is significantly different for each quantum phase of the system, even though the differences are difficult to be described quantitatively.

In Fig. 11, it can be seen how the Trotter expansion differs from the exact evolution. The larger oscillations at higher times are a probable explanation for the enhanced performance of the CNN when trained and tested on the Trotterized data. Note that the error introduced from using the Trotter expansion is not arbitrary noise, it explicitly depends on the various terms of the Hamiltonian, so it still carries the information about the phase of the system. This error likely makes the features of the input data easier to recognize by the CNN than the much smoother exact evolution.

[1] J. P. Elliott and J. D. Cockcroft, Collective motion in the nuclear shell model. I. Classification schemes for states of mixed configurations, *Proc. R. Soc. London A* **245**, 128 (1958).

[2] F. Iachello, in *The Interacting Boson Model*, edited by R. E. Chrien and W. R. Kane (Springer US, Boston, MA, 1979).

- [3] H. Lipkin, N. Meshkov, and A. Glick, Validity of many-body approximation methods for a solvable model: (I). Exact solutions and perturbation theory, *Nucl. Phys.* **62**, 188 (1965).
- [4] D. Agassi, Validity of the BCS and RPA approximations in the pairing-plus-monopole solvable model, *Nucl. Phys. A* **116**, 49 (1968).
- [5] J. E. García-Ramos, J. Dukelsky, P. Pérez-Fernández, and J. M. Arias, Phase diagram of an extended Agassi model, *Phys. Rev. C* **97**, 054303 (2018).
- [6] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, *Phys. Lett. A* **146**, 319 (1990).
- [7] F. Tacchino, A. Chiesa, S. Carretta, and D. Gerace, Quantum computers as universal quantum simulators: State-of-the-art and perspectives, *Adv. Quantum Technol.* **3**, 1900052 (2019).
- [8] J. Faba, V. Martín, and L. Robledo, Two-orbital quantum discord in fermion systems, *Phys. Rev. A* **103**, 032426 (2021).
- [9] P. Pérez-Fernández, J. M. Arias, J. E. García-Ramos, and L. Lamata, A digital quantum simulation of the Agassi model, *Phys. Lett. B* **829**, 137133 (2022).
- [10] J. E. García-Ramos, J. Dukelsky, P. Pérez-Fernández, and J. M. Arias, An extended Agassi model: Algebraic structure, phase diagram, and large size limit, *Phys. Scr.* **94**, 044003 (2019).
- [11] Y. Gordienko, P. Gang, J. Hui, W. Zeng, Y. Kochura, O. Alienin, O. Rokovyi, and S. Stirenko, Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer, *Advances in Computer Science for Engineering and Education* (Springer Cham, Switzerland, 2019), p. 638.
- [12] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen *et al.*, Highly accurate protein structure prediction with AlphaFold, *Nature (London)* **596**, 583 (2021).
- [13] R. Arjona, H.-N. Lin, S. Nesseris, and L. Tang, Machine learning forecasts of the cosmic distance duality relation with strongly lensed gravitational wave events, *Phys. Rev. D* **103**, 103513 (2021).
- [14] K. O'Shea and R. Nash, An Introduction to Convolutional Neural Networks, [arXiv:1511.08458](https://arxiv.org/abs/1511.08458).
- [15] C. D. Batista and G. Ortiz, Generalized Jordan-Wigner Transformations, *Phys. Rev. Lett.* **86**, 1082 (2001).
- [16] P. Jordan and E. Wigner, Über das Paulische Äquivalenzverbot, *Z. Phys.* **47**, 631 (1928).
- [17] T. P. Harty, D. T. C. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas, High-Fidelity Preparation, Gates, Memory, and Readout of a Trapped-Ion Quantum Bit, *Phys. Rev. Lett.* **113**, 220501 (2014).
- [18] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas, High-Fidelity Quantum Logic Gates using Trapped-Ion Hyperfine Qubits, *Phys. Rev. Lett.* **117**, 060504 (2016).
- [19] J. Casanova, A. Mezzacapo, L. Lamata, and E. Solano, Quantum Simulation of Interacting Fermion Lattice Models in Trapped Ions, *Phys. Rev. Lett.* **108**, 190502 (2012).
- [20] C. Monroe and J. Kim, Scaling the ion trap quantum processor, *Science* **339**, 1164 (2013).
- [21] A. K. Ratcliffe, R. L. Taylor, J. J. Hope, and A. R. R. Carvalho, Scaling Trapped Ion Quantum Computers using Fast Gates and Microtraps, *Phys. Rev. Lett.* **120**, 220501 (2018).
- [22] R. L. Taylor, C. D. B. Bentley, J. S. Pedernales, L. Lamata, E. Solano, A. R. R. Carvalho, and J. J. Hope, A study on fast gates for large-scale quantum simulation with trapped ions, *Sci. Rep.* **7**, 46197 (2017).
- [23] C. Zhang, F. Pokorny, W. Li, G. Higgins, A. Pschl, I. Lesanovsky, and M. Hennrich, Submicrosecond entangling gate between trapped ions via Rydberg interaction, *Nature (London)* **580**, 345 (2020).
- [24] Z. Mehdi, A. K. Ratcliffe, and J. J. Hope, Scalable quantum computation with fast gates in two-dimensional Microtrap arrays of trapped ions, *Phys. Rev. A* **102**, 012618 (2020).
- [25] E. D. Davis and W. D. Heiss, Random-phase approximation and broken symmetry, *J. Phys. G: Nucl. Phys.* **12**, 805 (1986).
- [26] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learning Internal Representations by Error Propagation* (MIT Press, Cambridge, MA, 1986), pp. 318–362.
- [27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**, 2825 (2011).
- [28] D. P. Kingma and J. Ba, Adam: A Method for Stochastic Optimization, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [29] L. Bottou, *Stochastic Learning* (Springer, Berlin/Heidelberg, 2004), pp. 146–168.
- [30] M. Szarvas, U. Sakai, and J. Ogata, Real-time Pedestrian Detection Using LIDAR and Convolutional Neural Networks, in 2006 IEEE Intelligent Vehicles Symposium, 2006, pp. 213–218.
- [31] C. Nebauer, Evaluation of convolutional neural networks for visual recognition, *IEEE Transactions on Neural Networks* **9**, 685 (1998).
- [32] H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification* (Springer Publishing Co., Inc., Switzerland, 2017).
- [33] F. Chollet *et al.*, Ke (2015)ras. online), available at: <https://github.com/fchollet/keras>.
- [34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, software available from tensorflow.org.
- [35] M. Caprio, P. Cejnar, and F. Iachello, Excited state quantum phase transitions in many-body systems, *Ann. Phys. (NY)* **323**, 1106 (2008).
- [36] A. Relaño, J. M. Arias, J. Dukelsky, J. E. García-Ramos, and P. Pérez-Fernández, Decoherence as a signature of an excited-state quantum phase transition, *Phys. Rev. A* **78**, 060102(R) (2008).
- [37] Q. Wang and F. Pérez-Bernal, Probing an excited-state quantum phase transition in a quantum many-body system via an out-of-time-order correlator, *Phys. Rev. A* **100**, 062113 (2019).