Adaptive projected variational quantum dynamics

David Linteau,[1,2,*] Stefano Barison,[1,2,3] Netanel H. Lindner,[4] and Giuseppe Carleo[1,2,3]
[1]*Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland*
[2]*Center for Quantum Science and Engineering, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland*
[3]*National Centre for Computational Design and Discovery of Novel Materials MARVEL, EPFL, CH-1015 Lausanne, Switzerland*
[4]*Physics Department, Technion, 320003 Haifa, Israel*

(Received 8 August 2023; accepted 9 April 2024; published 6 May 2024)

We propose an adaptive quantum algorithm to prepare accurate variational time evolved wave functions. The method is based on the projected variational quantum dynamics (pVQD) algorithm, that performs a global optimization with linear scaling in the number of variational parameters. Instead of fixing a variational ansatz at the beginning of the simulation, the circuit is grown systematically during the time evolution. Moreover, the adaptive step does not require auxiliary qubits and the gate search can be performed in parallel on different quantum devices. We apply the algorithm, named adaptive pVQD, to the simulation of driven spin models and fermionic systems, where it shows an advantage when compared to both Trotterized circuits and nonadaptive variational methods. Finally, we use the shallower circuits prepared using the adaptive pVQD algorithm to obtain more accurate measurements of physical properties of quantum systems on hardware.

DOI: 10.1103/PhysRevResearch.6.023130

I. INTRODUCTION

Simulation of static and dynamic properties of quantum systems is a notoriously hard task for classical computers. While analytical solutions are available only for specific cases, the amount of time and computing resources required in general by exact numerical methods is exponential in the system size, making the calculations quickly unfeasible. While several approximated many-body numerical techniques have been proposed [1–4], the accurate description of important physical and chemical phenomena is a very active research problem [5–8].

In recent years, quantum computers have seen significant developments [9–11], opening potential opportunities for scientific discoveries. Hardware capabilities continue to advance steadily, and we can already create and manipulate complex many-body quantum systems [12–17]. However, large-scale fault-tolerant quantum computers remain far in the future, and contemporary devices show limitations in connectivity, size, and coherence times.

Accounting for these constraints, variational quantum algorithms (VQAs) have emerged as the leading strategy to take advantage of near-term quantum devices [18–21]. In this class of algorithms, the solution of a given problem (e.g., finding the ground state of a physical system) is encoded in a quantum circuit that depends on some parameters optimized with the aid of a classical device. VQAs have not only been proposed for quantum simulations but also for a variety of different applications, such as machine learning [22,23], combinatorial optimization [24,25], quantum error correction [26,27], and circuit compilation [28–32]. Variational schemes have also been introduced in quantum dynamics [33–44] as a more efficient alternative to Trotterization [45–49]. The accuracy of a variational quantum simulation is then tied to the ability of a parameterized circuit to describe time-evolved wave functions. Even if the initial wave function is well described by the chosen parameterized circuit, the complexity of the time-evolved wave functions varies with time and the chosen circuit may fail to describe them. The choice of the parameterized circuit is therefore crucial and it remains an open problem in variational simulations of quantum dynamics.

Adaptive schemes have been proposed in the context of variational ground-state search [50–53], especially to avoid committing to a particular parameterized circuit. The key idea is to construct the parameterized circuit during optimization. By systematically appending specific quantum gates to the parameterized circuit, adaptive methods have been shown to surpass standard approaches in the number of operations required and in the accuracy of the final results. Moreover, adaptive methods provide flexible circuits suited for dynamics simulations [35,54]. However, including an adaptive step for dynamics usually requires measurements of additional quantities, that might be difficult to perform, or auxiliary qubits.

In this paper, we introduce an adaptive variational algorithm for real-time evolution based on the projected-variational quantum dynamics (pVQD) algorithm [39], denoted "adaptive pVQD". The method inherits all the properties of the original pVQD algorithm and integrates the adaptive modification of the parameterized circuit without requiring auxiliary qubits. The structure of this paper is as

*david.linteau@epfl.ch

follows: in Sec. II we present the algorithm and describe how the adaptive routine is performed; in Sec. III we apply the method to study a time-dependent and a fermionic system, benchmarking the method against Trotter evolution and the original pVQD algorithm; Sec. IV concludes the paper with some considerations and outlooks on the proposed method.

II. METHOD

Consider a physical system governed by a Hamiltonian H. For clarity of exposition, we focus on time-independent Hamiltonians. However, this is not a requirement of the algorithm, as we explicitly show in Sec. III. To simulate the dynamics of quantum systems on a quantum computer, we have to prepare the time-evolved wave function $|\Psi(t)\rangle = U(t)|\psi_0\rangle$, where $|\psi_0\rangle = C_0|0\rangle^{\otimes N}$ is the initial state, N indicates the number of qubits representing the physical system and $U(t)$ is the unitary time-evolution operator.

The goal of variational algorithms for dynamics is to approximate $|\Psi(t)\rangle$ using parameterized states of the form $|\psi(\boldsymbol{\theta}(t))\rangle = C(\boldsymbol{\theta}(t))|\psi_0\rangle$, where $\boldsymbol{\theta}(t) \in \mathbb{R}^p$ is the vector of the p real parameters at time t. Different strategies have been proposed to prepare the variational approximation [33,34,36–39,41]. In particular, our adaptive scheme builds upon the projected-variational quantum dynamics algorithm (pVQD) [39].

A. Projected-variational quantum dynamics

The pVQD algorithm approximates the time evolution of a quantum system until t_f by discretizing it into N_t small time steps Δt. Given the variational state at time t, $|\psi(\boldsymbol{\theta}(t))\rangle$, we perform the small step time evolution and prepare the evolved state $|\phi(t+\Delta t)\rangle = U(\Delta t)|\psi(\boldsymbol{\theta}(t))\rangle$.

The new variational state $|\psi(\boldsymbol{\theta}(t+\Delta t))\rangle$ is prepared by finding the change of the parameters $\boldsymbol{d\theta}^* \in \mathbb{R}^p$ such that

$$\boldsymbol{d\theta}^* = \underset{\boldsymbol{d\theta}\in\mathbb{R}^p}{\arg\max}\, |\langle\psi(\boldsymbol{\theta}(t)+\boldsymbol{d\theta})|\phi(t+\Delta t)\rangle|^2 \tag{1}$$

and imposing $|\psi(\boldsymbol{\theta}(t+\Delta t))\rangle = |\psi(\boldsymbol{\theta}(t)+\boldsymbol{d\theta}^*)\rangle$. It has been shown that, in the infinitesimal time step limit $\Delta t \to 0$, the parameters update coincides with the one obtained using the McLachlan's variational principle [39]. From now on, we will indicate $\boldsymbol{\theta}(t)$ as $\boldsymbol{\theta}$ to simplify the notation, implying that the parameters we are referring to are the ones assigned to the ansatz at time t, except when explicitly indicated.

In order to execute the algorithm on a quantum device, the overlap optimization is performed minimizing with respect to $\boldsymbol{d\theta}$ the infidelity

$$\mathcal{I}(\boldsymbol{d\theta}, \Delta t) = 1 - \mathcal{F}(\boldsymbol{d\theta}, \Delta t), \tag{2}$$

where the fidelity is defined as

$$\mathcal{F}(\boldsymbol{d\theta}, \Delta t) = |\langle\psi(\boldsymbol{\theta}+\boldsymbol{d\theta})|U_{\mathrm{TS}}(\Delta t)|\psi(\boldsymbol{\theta})\rangle|^2, \tag{3}$$

and $U_{\mathrm{TS}}(\Delta t)$ is the Trotter-Suzuki approximation of the time-evolution operator [45,46]. In this paper, we use first- and second-order decomposition for spin systems and the Fermi-Hubbard model, respectively, but higher orders can be considered. Given the variational unitary $C(\boldsymbol{\theta})$ as a quantum circuit, the infidelity in Eq. (2) and its gradient can be computed efficiently on a quantum device [39].

The minimization of the infidelity is then performed in a gradient descent scheme

$$\boldsymbol{d\theta}_{\mathrm{new}} = \boldsymbol{d\theta}_{\mathrm{old}} - \eta\nabla_{\boldsymbol{d\theta}}\mathcal{I}(\boldsymbol{d\theta}, \Delta t), \tag{4}$$

where $\eta \in \mathbf{R}^+$ is the learning rate. This optimization goes on until a pre-determined convergence criteria is met. More details about the minimization routine used in this paper can be found in Appendix A. The procedure is repeated N_t times until the simulation reaches t_f. A sketch of the algorithm can be found in the upper box of Fig. 1, and a more detailed explanation in [39].

B. Adaptive step

The complexity of the wave function changes during the time evolution of the system. The variational circuit $C(\boldsymbol{\theta})$ might therefore not be expressive enough to accurately describe the time step evolution by only shifting the variational parameters. In this case, the parameter optimization fails and a bias is introduced in the representation of the wave function. In contrast to variational preparation of ground states, in the study of dynamics, this error propagates to the subsequent time steps, leading to poor accuracy on the final results. This is a shortcoming of any nonadaptive variational approach to dynamics, including the pVQD algorithm.

We aim to overcome this limitation by introducing the adaptive projected-variational quantum dynamics algorithm (adaptive pVQD). The method builds upon the original pVQD algorithm, by supplementing it with an adaptive step, which incrementally extends the variational ansatz. This is done by drawing new gates from a predefined pool of operators. However, the introduction of the adaptive step always requires an overhead in the number of circuits to be prepared and measured [35,54]. If this overhead is too high, it might hinder the application of the method, even if it is preferable from the perspective of circuit expressivity.

In the following, we show how the adaptive pVQD algorithm introduces the adaptive step in variational quantum dynamics while significantly reducing the overhead compared to other adaptive methods.

First, we consider a variational ansatz of the form

$$|\psi(\boldsymbol{\theta}, \boldsymbol{A})\rangle = C(\boldsymbol{\theta}, \boldsymbol{A})|\psi_0\rangle = \prod_i e^{-i\theta_i A_i}|\psi_0\rangle, \tag{5}$$

where each variational gate $e^{-i\theta_i A_i}$ with parameter $\theta_i \in \boldsymbol{\theta}$ is associated to a Hermitian generator $A_i \in \boldsymbol{A}$.

The parameterized state is now specified not only by the set of parameters $\{\boldsymbol{\theta}\}$, but also from the set of operators $\{\boldsymbol{A}\}$. However, for simplicity of notation, we will continue to adopt the notation $|\psi(\boldsymbol{\theta})\rangle \equiv |\psi(\boldsymbol{\theta}, \boldsymbol{A})\rangle$ and $C(\boldsymbol{\theta}) \equiv C(\boldsymbol{\theta}, \boldsymbol{A})$. We note that the original pVQD scheme [39] consists of fixing the set of operators $\boldsymbol{A}$ at the beginning of the simulation and keeping it constant until the final time t_f. Here, instead, we start with the initial state $|\psi_0\rangle$ represented by an empty set of operators.

When the circuit does not contain any parameter, or $|\psi(\boldsymbol{\theta})\rangle$ is not expressive enough to accurately describe the time step evolution by only shifting the variational parameters, new gates are added to it. The set of operators $\mathcal{A}$ from which we will draw the generators A_k to add to $\{\boldsymbol{A}\}$ is defined as the operator pool. In Sec. II C, two types of operation pools will

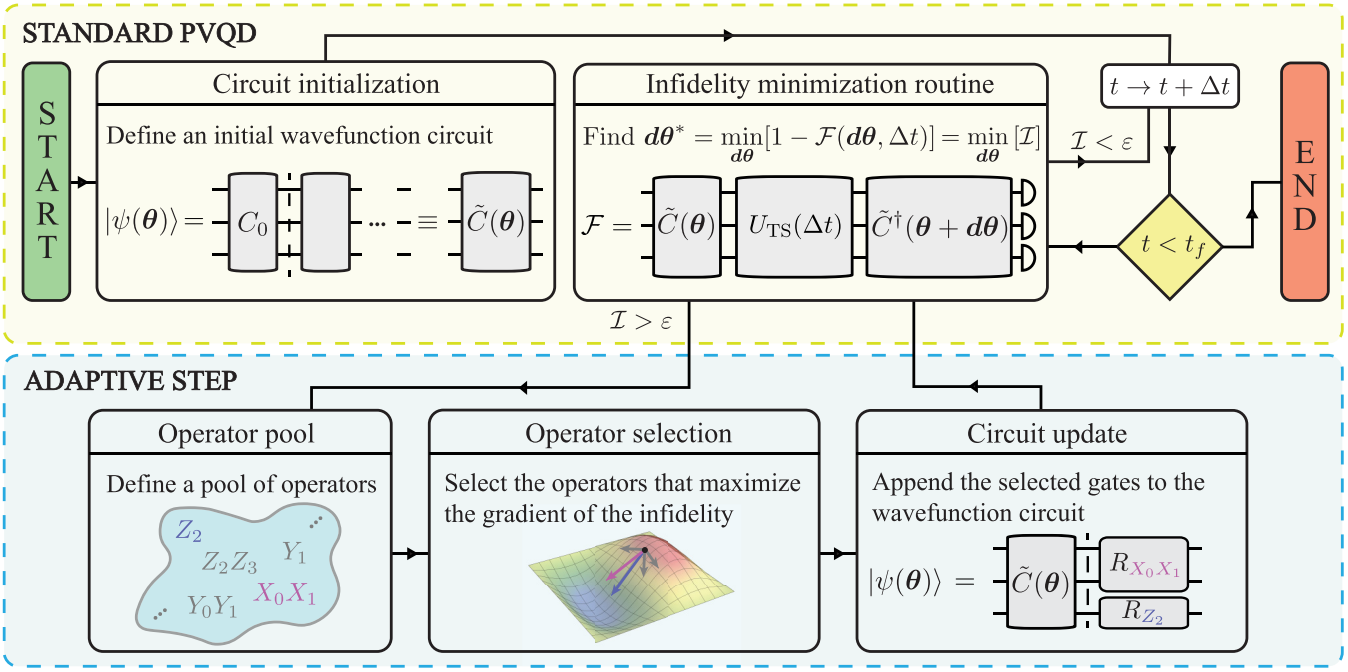

FIG. 1. Flowchart of the time evolution of the "adaptive pVQD" algorithm. Starting with a parameter-free circuit, we discretize the time evolution into multiple time steps. At each time step we optimize the parameters to approximate the real-time evolution of the quantum system. To simplify the notation, we define $\tilde{C}(\boldsymbol{\theta}) = C(\boldsymbol{\theta})C_0$. If the optimization does not converge to the required accuracy, or the ansatz does not contain any parameter, then rotations $\{R_{A_i^*}\}$ with associated the generators $\{A_i^*\}$ are appended to the circuit according to the adaptive step procedure described in Sec. II B. The algorithm stops once the final time t_f is reached.

be introduced. Here, we describe how to determine from a given pool the best gate to grow the quantum circuit and how to add it. As first proposed in [50], we look for the operator whose gate maximizes the derivative of the cost function with respect to its parameter. This is achieved by iterating over all the operators in the pool, a step that can be performed in parallel even on different quantum devices.

For ground-state methods, the cost function is the energy of the system, and the gradient is obtained by measuring the expectation value of the commutator between the trial operator and the Hamiltonian [50,55]. We must ensure that it is possible to apply a similar procedure when dynamics is considered. In the adaptive scheme proposed in [35], this step requires an additional measurement of the variance of the Hamiltonian with respect to the nonadaptive case. In the method presented here, the procedure changes depending on where the new gate is added. For instance, if the new gate is placed at the end of the circuit, the derivative with respect of the shift of the new parameter can be computed using the parameter shift rule [56], similarly as the minimization routine (see Appendix A for more details). If instead the new gate is placed at the beginning of the circuit, the number of circuits required can be reduced even more. Indeed, in this case the gradient of the fidelity with respect to the shift $d\theta_a$ of parameter θ_a associated with a trial operator A_a has the form

$$\frac{\partial \mathcal{F}}{\partial d\theta_a} = \langle \phi(\boldsymbol{\theta}, \Delta t)| e^{-id\theta_a A_a}[P_0, iA_a]e^{id\theta_a A_a}|\phi(\boldsymbol{\theta}, \Delta t)\rangle, \quad (6)$$

where we define the projector $P_0 = |\psi_0\rangle\langle\psi_0|$ and the state $|\phi(\boldsymbol{\theta}, \Delta t)\rangle = U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}(\Delta t)|\psi(\boldsymbol{\theta})\rangle$ (see Appendix B for the full derivation). Hence measuring the derivative of the fidelity corresponds to measuring the Hermitian operator $[P_0, iA_a]$ with respect to the state $e^{id\theta_a A_a}|\phi(\boldsymbol{\theta}, \Delta t)\rangle$. This reduces the number of circuits to evaluate to one per operator, compared to two for the parameter-shift rule. In both cases, to ensure continuity of time evolution, we initially set $\theta_a, d\theta_a = 0$. In this paper, we chose to add the new gates at the end and evaluate the derivative using the parameter-shift rule. Both procedures are parallelizable on multiple devices and do not require auxiliary qubits. A sketch of the adaptive step is shown in Fig. 1.

The adaptive step has been lately extended and optimized [51,53,55], with new protocols that greatly reduce the computational resources required with respect to the first proposal. In particular, we adopt the scheme presented in [53], which increases the depth of the parameterized circuit $|\psi(\boldsymbol{\theta})\rangle$ by 1 at every adaptive step, when no specific connectivity of the device is assumed. While the infidelity defined in Eq. (2) remains above a fixed threshold ε, additional adaptive steps are performed. For a detailed description, see Appendix C.

C. Operator pool

The choice of the operator pool is a key ingredient in the success and efficiency of adaptive variational algorithms. As an example, one could define a pool of operator that is able to prepare every state in the N-qubits Hilbert space, also called a complete pool of operators. However, finding such a pool is not only computationally intensive, but also not guaranteed to be the best choice, given the local nature of the updates to the ansatz. For these reasons, many different strategies have been proposed, such as the creation of a minimally complete pool

[51,57], the inclusion of symmetries directly in the operator pool [58], or the extension of a complete pool acting on a subsystem of the studied model [52].

In the study of dynamics, we can refer to the Trotterization of the time-evolution operator to select the pool. In particular, we consider local (L) and nonlocal (NL) operator pools, respectively, given by

$$\mathcal{A}_{\mathrm{L}} = \{X_i, Y_i, Z_i, X_iX_j, Y_iY_j, Z_iZ_j\}_{i,j\in[0,N-1]}^{\langle ij\rangle}, \tag{7}$$

$$\mathcal{A}_{\mathrm{NL}} = \{X_i, Y_i, Z_i, X_iX_j, Y_iY_j, Z_iZ_j\}_{i,j\in[0,N-1]}, \tag{8}$$

where X_i, Y_i, and Z_i are the Pauli gates acting on site i, and $\langle ij\rangle$ indicates nearest-neighbor sites. Once a physical system is mapped to a qubit system, we consider as local the gates that act on qubits that are neighbors in the indexing convention given by the mapping. This is not a requirement of the method. The local pool could alternatively be defined as the set of operators that are local in a given topology. A nonlocal pool, instead, contains operators that might act also on qubits that are not neighbors in the indexing convention. These nonlocal operators could be a product of the mapping, or introduced to improve the results. Indeed, given that $\mathcal{A}_{\mathrm{L}} \subseteq \mathcal{A}_{\mathrm{NL}}$, where the equality holds for devices with all-to-all connectivity, we expect that $\mathcal{A}_{\mathrm{NL}}$ will generate more flexible parameterized states. However, not only the choice of $\mathcal{A}_{\mathrm{NL}}$ leads to a measurement overhead, but the nonlocal nature of this pool may add long-range controlled-NOT (CNOT) gates to the circuit, according to the device connectivity. In Sec. III, we report the comparison of the two pools in the study of a fermionic system.

III. RESULTS

We apply the adaptive pVQD method to the study of the 1D Heisenberg XYZ model with an external driving field and the Fermi-Hubbard model. Both have nontrivial dynamics and open the pVQD method to the study of time-dependent and fermionic systems. In both cases, open boundary conditions were imposed.

A. Driven Heisenberg model

Given an open chain of L spins, the driven Heisenberg XYZ Hamiltonian can be written as

$$H(t) = \sum_{i=0}^{L-2}(J_xX_iX_{i+1} + J_yY_iY_{i+1} + J_zZ_iZ_{i+1}) + D(t) \tag{9}$$

where J_x, J_y, and J_z are coupling parameters and $D(t)$ is the time-dependent driving term. Many different driving terms can be applied to the system. Among those we choose

$$D(t) = \sum_{i=0}^{L-1}(-1)^i \sin(\omega t)Z_i, \tag{10}$$

where ω is the driving frequency.

First, we investigate the performance of the adaptive pVQD algorithm with a local pool on a perfect simulator and compare to Trotterized circuits and the original implementation of pVQD. We consider $J_x = 1$, $J_y = 0.8$, $J_z = 0.6$, an antiferromagnetic initial state $|\psi_0\rangle = |01010101\rangle$ and a final evolution time $t_f = 2$. In the classic version of the pVQD algorithm, we have to choose an ansatz for the time evolved wave function. We consider a circuit equivalent to a Trotter step where all the rotations are defined by variational parameters. The Trotter step circuit implementation for this model is shown in Appendix E. Both the Trotter and the pVQD full circuits are then obtained repeating this structure n_{TS} times. In particular, we fix $n_{\mathrm{TS}} = 10$ for the Trotter circuit and $n_{\mathrm{TS}} = 3$ for the pVQD ansatz.

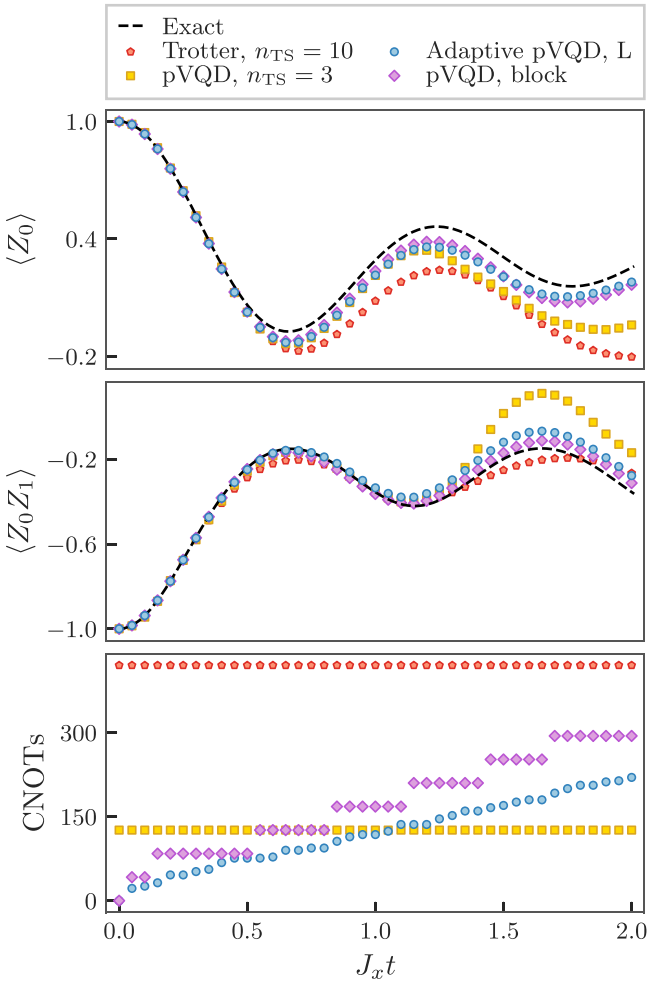

FIG. 2. Dynamics of the driven Heisenberg XYZ model studied with the adaptive pVQD algorithm with local pool (L), compared to standard Trotter evolution, pVQD and pVQD with block extensions. The plot shows the results for an open chain of $L = 8$ spins with $J_x = 1$, $J_y = 0.8$, and $J_z = 0.6$. The top and middle panels show the measurements of a single spin observable and a correlator, respectively. The bottom panel shows the number of CNOTs in the circuit describing the time-evolved wave function. The simulation started in the antiferromagnetic state $|\psi_0\rangle = |01010101\rangle$, and the infidelity threshold was set to $\varepsilon = 10^{-4}$ for all the variational methods.

After running the algorithms, we compare the different circuits obtained and use them to measure expectation values of single- and two-spin observables. The results are shown in Fig. 2. The Trotter circuit lags behind variational methods both in terms of accuracy and resource required. The pVQD method instead achieves accurate results up until $Jt = 1.0$, where the associated circuit becomes shallower than the one

of adaptive pVQD. From that time step onwards, the fixed representation power is the main source of error in the variational results.

In order to show the flexibility of the adaptive pVQD, we implement a naive modification of the pVQD algorithm that we indicate as pVQD with block extensions. In this case, a new step of the Trotterized variational ansatz is added to the circuit once the optimization procedure does not reach the desired accuracy. While this approach does improve the performance of the pVQD algorithm, we remark that it is not general, as it depends on the ansatz structure we have chosen. Furthermore, we can see from the bottom panel of Fig. 2 that the adaptive pVQD method always produces shallower circuits, with resources tailored to the needs of the specific time step.

Then, we extend the study to systems with different sizes. To this end, we define the integrated exact infidelity

$$\Delta_{\mathcal{I}}^{\mathrm{ex}}(t_f) = \int_0^{t_f} \left(1 - |\langle\Psi(t)|\psi(\boldsymbol{\theta})\rangle|^2\right) dt \tag{11}$$

with respect to the exact wave function $|\Psi(t)\rangle$ computed on a classical device. We again fix a final evolution time $t_f = 2$ and evaluate $\Delta_{\mathcal{I}}^{\mathrm{ex}}(t_f)$ for each method for systems of $L \in [3, 11]$ spins. In particular, we consider a Trotter circuit with a fixed depth of $n_{\mathrm{TS}} = 10$ and one with fixed Trotter step size $dt = J_x t_f / n_{\mathrm{TS}} = 0.05$—the same we use in the Trotter step of the pVQD algorithm. The results are shown in Fig. 3, together with the circuit depth at the end of the time evolution.

We note that the depth of the adaptive pVQD circuits increases with the system size and converges to the Trotter circuit with fixed depth, while having a lower integrated exact infidelity. We highlight that Fig. 3 only indicates the depth of the final circuit. In the case of adaptive pVQD, this corresponds to the deepest circuit prepared. The Trotterized circuits with a fixed Trotter step size yield the lowest values for $\Delta_{\mathcal{I}}^{\mathrm{ex}}$, but $n_{\mathrm{TS}} = 40$ Trotter steps are required to evolve the system to $t_f = 2$, resulting in circuits almost one order of magnitude deeper than any other. We performed multiple pVQD simulations with different variational ansätze equivalent to $n_{\mathrm{TS}} = 1, 2, 3, 8$ Trotter steps. We note that the integrated exact infidelities of pVQD with $n_{\mathrm{TS}} = 1, 2, 3$ all have a steep transition when the number of gates becomes smaller than the adaptive circuit. As mentioned above, these transitions are a result of the limited representation power of the ansatz. The adaptive scheme, on the contrary, is able to efficiently increase its expressivity. Although the standard pVQD calculation with $n_{\mathrm{TS}} = 8$ never undergoes this transition and $\Delta_{\mathcal{I}}^{\mathrm{ex}}$ is always lower than the one of the adaptive approach, we emphasize that the entire time evolution is performed with a deeper circuit. Finally, a plateau in the final circuit depth of the adaptive circuits can be seen for $L > 8$. This is similar to what was observed in [35], where the system size at which the number of gates required saturates depends on the evolution time.

A complete study of a quantum algorithm would require to run it on actual quantum hardware. Indeed, the adaptive method is able to produce circuits that are orders of magnitude shallower than Trotterization while keeping the accuracy comparable to it. However, it is crucial to fist assess the error rate of current hardware and determine how it affects the optimization of the variational parameters. To this end,

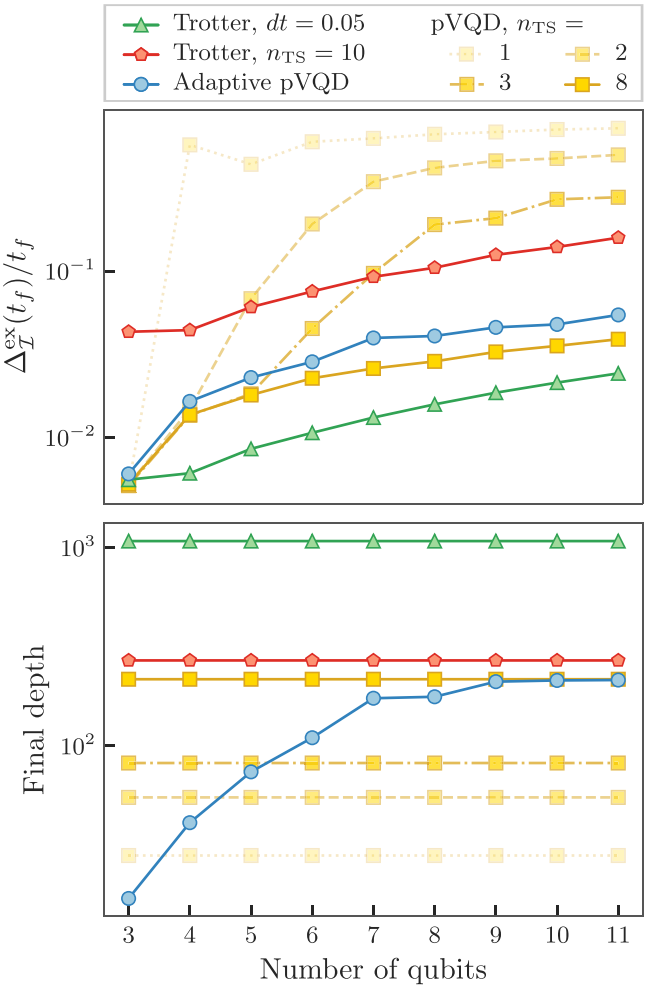

FIG. 3. Adaptive pVQD algorithm with local pool compared to standard Trotter evolution and pVQD for the driven Heisenberg XYZ model. We use the same settings as indicated in Fig. 2 for multiple systems of size $L \in [3, 11]$. The top panel shows the integrated exact infidelity of pVQD and Trotterization over an entire time evolution with final time $t_f = 2$ as a function of the system size. The bottom panel shows the circuit depth at the end of the time evolution.

we perform classical density matrix simulations with noise models imported from the IBM Brisbane superconducting device [59]. More details about the noise model can be found in Appendix F. From these noise simulations, to have qualitatively good results, we would need error rates on single- and two-qubit gates that are one order of magnitude lower than what we currently experience on these devices. Nonetheless, circuits produced with the adaptive pVQD scheme can already be used to improve the measurement of observables at long times on current quantum devices, which are otherwise limited by the depth of the Trotterization. For this reason, we first run the adaptive pVQD algorithm on the simulator, and then use the resulting sets of variational parameters to prepare quantum circuits on hardware for a system of $L = 4$ spins. In Fig. 4, we compare observables measured both on those variational wave functions and on Trotterized circuits with a fixed Trotter step size of $dt = 0.2$.

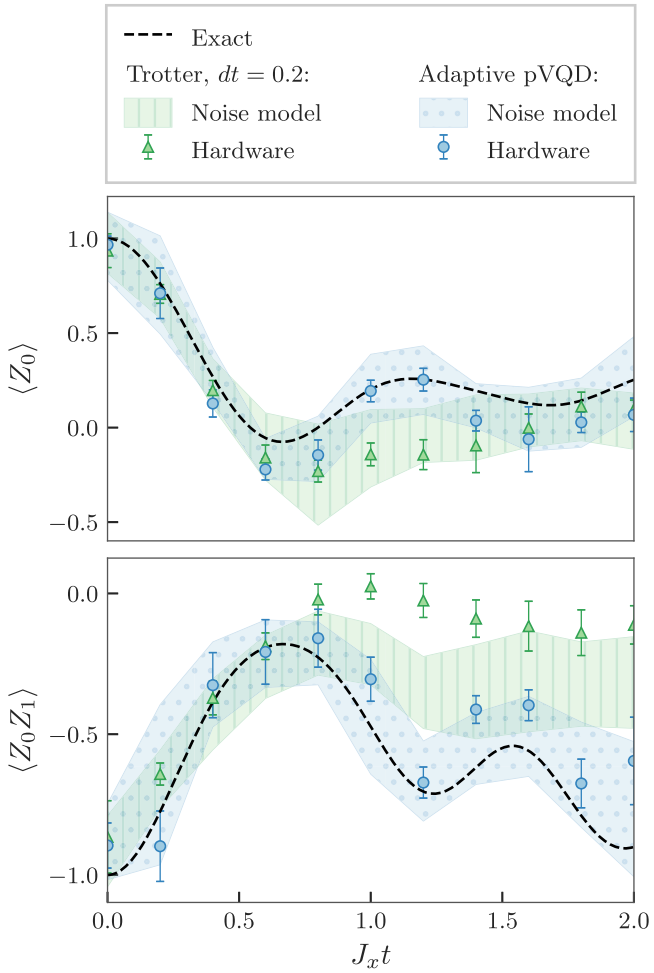

FIG. 4. Observables measured with the IBM Manila device for the driven Heisenberg XYZ model on an open chain with four sites, $J_x = 1, J_y = 0.8, J_z = 0.6$ and an antiferromagnetic initial state $|\psi_0\rangle = |0101\rangle$. The Trotter simulation is performed with a fixed Trotter step size of $dt = 0.2$. The adaptive pVQD circuits $|\psi(\boldsymbol{\theta})\rangle$ were obtained with a noiseless simulation that used a local operator pool. The shaded areas correspond to 50 noisy simulations using the noise model of IBM Manila. Each data point and error bar correspond to the mean and the standard deviation, respectively, of 50 experiments performed on hardware. Zero-noise extrapolation was applied to both noisy simulations and hardware experiments. Idle qubits were also dynamically decoupled from the active ones.

In this experiment, the final Trotter circuit has 180 CNOTs. This circuit is beyond what is currently accessible on quantum devices, so that the expectation value of the correlator settles close to 0 for $J_x t > 0.8$. On the other hand, the adaptive pVQD parameterized circuit $|\psi(\boldsymbol{\theta})\rangle$ has 28 CNOTs at the end of the evolution. This improvement in the number of gates is crucial for the application of error mitigation techniques, especially at longer times. In particular, zero-noise extrapolation (ZNE [33,60]) was applied both on the noisy simulations and hardware experiments. We choose a quadratic fit on values obtained with noise scaling factors [1, 2, 3]. Moreover, when running our algorithm on hardware, we dynamically decouple the idle qubits from the active ones using the standard procedure available in Qiskit [61]. We expect that more advanced noise mitigation techniques, such as the one presented in [62], will improve the results on the Trotter circuit. However, this is also true for the variational circuit prepared by the adaptive pVQD.

B. Fermi-Hubbard model

The Hamiltonian of the Fermi-Hubbard model on a $L_x \times L_y$ rectangular lattice is given by

$$H = -J \sum_{\langle ij \rangle, \sigma} (c^\dagger_{i\sigma} c_{j\sigma} + c^\dagger_{j\sigma} c_{i\sigma}) + U \sum_{i=0}^{L_x L_y - 1} n_{i\uparrow} n_{i\downarrow}, \tag{12}$$

where $c^\dagger_{i\sigma}$ ($c_{i\sigma}$) is the creation (annihilation) fermionic operator of spin $\sigma \in \{\uparrow, \downarrow\}$ at site i, $n_{i\sigma} = c^\dagger_{i\sigma} c_{i\sigma}$ counts the number of fermions with spin σ at site i and $\langle ij \rangle$ denotes nearest-neighbor sites on the lattice. The first term in the Hamiltonian accounts for the hopping between nearest-neighbor lattice sites, while the second term describes the on-site interactions.

There are several ways to encode fermionic Hamiltonians into qubit operators [63–69]. In this paper, we consider the Jordan-Wigner mapping [63] to encode each fermionic mode into a qubit. Since every lattice site can host two modes $(\uparrow, \downarrow)$, $N = 2L_x L_y$ qubits are required to simulate the Fermi-Hubbard model on a $L_x \times L_y$ grid. Before performing a fermionic encoding, we eliminate the spin index via $c_{i\uparrow} \to c_i$ and $c_{i\downarrow} \to c_{i+N/2}$ (and analogously for the number operator $n_{i\sigma}$). We then map each fermionic operator into a spin operator,

$$c_i \to Z^{\otimes i} \otimes \sigma^+ \otimes \mathbb{I}^{\otimes N-i-1}, \tag{13}$$

$$c^\dagger_i \to Z^{\otimes i} \otimes \sigma^- \otimes \mathbb{I}^{\otimes N-i-1}, \tag{14}$$

where $\sigma^\pm = (X \pm iY)/2$. The local occupation number can then be identified with the local spin number according to $n_i \in \{0, 1\} \mapsto Z_i \in \{\uparrow, \downarrow\}$. More details on the fermionic indexing convention and the Trotter step implementation can be found in Appendix E.

Given that the mapping requires an ordering of the fermionic modes, operators that are local in space might generate very long Pauli strings. For example, considering the snake-like pattern described in Appendix E, vertical hopping terms generate strings of Pauli Z with sizes up to $2L_x - 2$. This represents a bottleneck in studying fermionic systems on current quantum devices. We want to test the adaptive pVQD method on systems that show this type of long range terms. By restricting the operator pool, we investigate the possibility of describing time-evolved wave functions of the Hubbard model using only local gates. We perform noiseless simulations of a 2×2 square lattice, which can also be regarded as a 1D chain with periodic boundary conditions, comparing local and nonlocal operator pools. In particular, we study a hopping quench from an antiferromagnetic initial state at half-filling. We measure the expectation values of a local density operator and a density correlator and count the number of CNOTs in the circuits. We use a fixed-depth Trotter simulation and pVQD with block extension scheme as a benchmark. The results are shown in Fig. 5.

We do not restrict ourselves to specific quantum hardware to keep the comparison as general as possible. Instead, we count the number of CNOTs in a circuit by transpiling it into

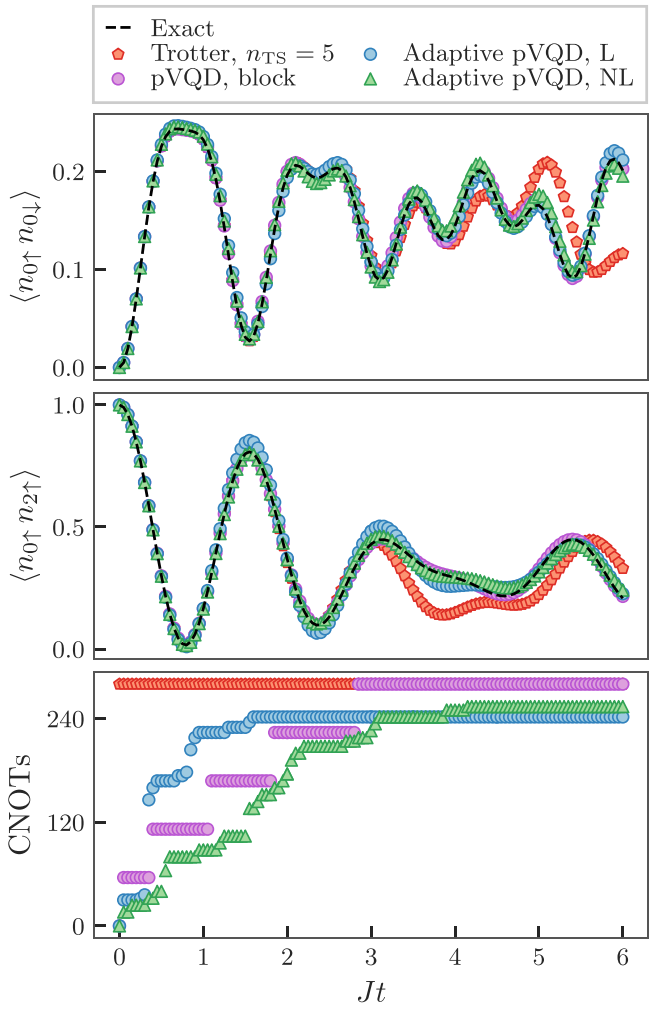

FIG. 5. Adaptive pVQD schemes for the Fermi-Hubbard model on a 2×2 open square lattice, also seen as a four-sites chain with periodic boundary conditions, (8 qubits) with $U/J = 0.8$. Local (L) and nonlocal (NL) operator pools are used to perform noiseless simulations and the results are compared to a Trotter evolution with $n_{TS} = 5$ Trotter steps and pVQD with block extensions. The system starts in the half-filled antiferromagnetic state $|\psi_0\rangle = |n_{0\uparrow} n_{1\uparrow} n_{2\uparrow} \cdots\rangle = |10100101\rangle$. We fixed the infidelity threshold to $\varepsilon = 10^{-4}$. The top and middle panels show the expectation values of an on-site number density operator and a number density correlator over time. The bottom panel shows the number of CNOTs in the circuit describing the time-evolved wave function.

an abstract device with all-to-all connectivity that is able to perform arbitrary single-qubit rotations and CNOTs. The local and nonlocal pool variants show different behavior over time in the count of CNOTs. We note that the nonlocal variant always requires fewer CNOTs than its local counterpart. However, some CNOTs are long range, and their implementation on an actual device can be challenging on hardware with fixed topology and limited connectivity. In contrast, the circuit structure produced by the local pool variant is already suited for current hardware implementation. More details about the adaptive pVQD output circuits can be found in Appendix D. Moreover, the plot highlights another limitation of the naive pVQD with block extensions approach. Indeed, it not only always prepares more expensive circuits than the nonlocal adaptive pVQD, but it has a similar CNOT count as the local variant while being restricted to use long-range gates (as defined in the Trotter step).

IV. CONCLUSIONS

We presented an adaptive version of pVQD, called adaptive pVQD, to simulate the real-time evolution of quantum systems. This algorithm importantly circumvents the need to choose a fixed ansatz from the beginning of the time evolution. The parameterized quantum circuits are grown adaptively to be both problem and hardware-tailored. This is obtained with a measurement overhead required to determine the best gates among those included in the operator pool.

However, the gate search can be operated in parallel and, in our scheme, it does not involve circuits with auxiliary qubits. This makes the adaptive pVQD algorithm more hardware efficient than standard methods, as exemplified in this paper with the driven Heisenberg model on the IBM quantum hardware. Finally, we have simulated the dynamics of the Hubbard model with only local gates, using the adaptive procedure to mitigate one of the bottlenecks that current quantum devices face in studying fermionic systems. Given the ease of introduction to the standard pVQD algorithm and its benefits, we believe that the adaptive procedure described here can be of great use in the simulation of dynamics both for current and future quantum devices.

The code used to run the simulations is open source and can be found at [70]. It was written in Python using Qiskit [61]. Exact classical simulations were performed using Qutip [71].

ACKNOWLEDGMENTS

We thank S. Economou for insightful discussions. This research was supported by the NCCR MARVEL, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (Grant No. 205602). N.H.L. and D.L. acknowledge financial support from the Israel Science Foundation within the ISF-Quantum program (Grant No. 2074/19).

APPENDIX A: MINIMIZATION ROUTINE

Here we present additional details on the minimization routine that we applied throughout the simulations we presented in the main text. In particular, we follow a gradient-based approach, with gradient computed using the parameter-shift rule. Gradient-based and nongradient-based optimization algorithms for dynamics were previously used for instance in [39,40], for both ideal and noisy quantum simulations. The parameter-shift rule readily applies here since every Pauli string A_i is involutive, i.e., $A_i^2 = \mathbb{I}$ [56]. For a fixed set of operators $\{\boldsymbol{A}\}$, the gradient of the infidelity was thus computed via the parameter-shift rule,

$$\frac{\partial \mathcal{I}}{\partial d\theta_i} = \frac{\mathcal{I}(\boldsymbol{\theta} + \boldsymbol{d\theta} + s\boldsymbol{e}_i) - \mathcal{I}(\boldsymbol{\theta} + \boldsymbol{d\theta} - s\boldsymbol{e}_i)}{2 \sin s}, \tag{A1}$$

where $\boldsymbol{e}_i$ is the standard unit vector, and we fixed $s = \pi/2$. The gradient was then fed to the Adam optimizer [72], implemented with the default hyperparameters and a learning rate $\alpha = 0.005$. The shift parameters $\boldsymbol{d\theta}^*$ were thus obtained using Adam.

Two stopping criteria for the optimizer were used: (1) the ℓ_∞ norm of the gradient of the infidelity is below a tolerance and (2) a maximum number of iterations is reached. Alternatively, as mentioned in [39], an optimization threshold independent from Δt can be used if the cost function $\mathcal{I}$ is replaced with $\mathcal{I}/\Delta t^2$.

APPENDIX B: GRADIENT OF THE FIDELITY

In this Appendix, we derive the expression for the gradient of the adaptive step presented in Eq. (6). Given the quantum circuit $U(\boldsymbol{\theta})$ that prepares the state $|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\psi_0\rangle$, we want to add the gate $e^{-id\theta_a A_a}$ to it, defining the new state $|\psi(\boldsymbol{\theta}+\boldsymbol{d\theta})\rangle = U(\boldsymbol{\theta})\, e^{-id\theta_a A_a}|\psi_0\rangle$. To obtain the gradient of the fidelity with respect to this added parameter $d\theta_a$, it is convenient to first rewrite the fidelity given in Eq. (3) as follows:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{d\theta}, \Delta t) &= |\langle\psi(\boldsymbol{\theta}+\boldsymbol{d\theta})|U_{\mathrm{TS}}(\Delta t)|\psi(\boldsymbol{\theta})\rangle|^2 \\
&= |\langle\psi_0|e^{id\theta_a A_a}U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}(\Delta t)U(\boldsymbol{\theta})|\psi_0\rangle|^2 \\
&= \langle\psi_0|e^{id\theta_a A_a}U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}(\Delta t)U(\boldsymbol{\theta})|\psi_0\rangle \\
&\quad * \langle\psi_0|U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}^\dagger(\Delta t)U(\boldsymbol{\theta})e^{-id\theta_a A_a}|\psi_0\rangle \\
&= \langle\psi_0|U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}^\dagger(\Delta t)U(\boldsymbol{\theta})e^{-id\theta_a A_a}|\psi_0\rangle \\
&\quad * \langle\psi_0|e^{id\theta_a A_a}U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}(\Delta t)U(\boldsymbol{\theta})|\psi_0\rangle \\
&= \langle\phi(\boldsymbol{\theta}, \Delta t)|e^{-id\theta_a A_a}P_0 e^{id\theta_a A_a}|\phi(\boldsymbol{\theta}, \Delta t)\rangle, \qquad \text{(B1)}
\end{aligned}$$

where we defined $|\phi(\boldsymbol{\theta}, \Delta t)\rangle = U^\dagger(\boldsymbol{\theta})U_{\mathrm{TS}}(\Delta t)U(\boldsymbol{\theta})|\psi_0\rangle$ and the projector $P_0 = |\psi_0\rangle\langle\psi_0|$. One can then readily differentiate with respect to $d\theta_a$ to obtain

$$\frac{\partial\mathcal{F}}{\partial d\theta_a} = \langle\phi(\boldsymbol{\theta}, \Delta t)|e^{-id\theta_a A_a}[P_0, iA_a]e^{id\theta_a A_a}|\phi(\boldsymbol{\theta}, \Delta t)\rangle, \qquad \text{(B2)}$$

which precisely corresponds to Eq. (6).

APPENDIX C: ADAPTIVE STEP IMPLEMENTATION

In this Appendix, we illustrate the adaptive procedure used in our simulations, based on what was initially proposed in [53]. The overall procedure can be divided in the following steps:

(1) *Compute the gradient of the infidelity for each operator in the pool*. To process the pool $\mathcal{A}$, the gate $e^{-i\theta_a A_a}$ associated to each trial operator $A_a \in \mathcal{A}$ is appended one at a time to the current parameterized circuit $\{\boldsymbol{\theta}, \boldsymbol{A}\}$, resulting in the trial circuit $\{(\boldsymbol{\theta}, 0), (\boldsymbol{A}, A_a)\}$. For the trajectory in parameter space to remain continuous, the new parameter θ_a and its shift $d\theta_a$ are set to 0. The gradient of the infidelity with respect to the new parameter is computed for each trial circuit using the parameter-shift rule, given explicitly in Eq. (A1).

(2) *Pick the operator in the pool that maximizes the gradient*. Update the parameters and operators to $\boldsymbol{\theta} \to (\boldsymbol{\theta}, 0)$ and $\boldsymbol{A} \to (\boldsymbol{A}, A^*)$, where A^* is the operator A_a that maximizes the infidelity gradient.

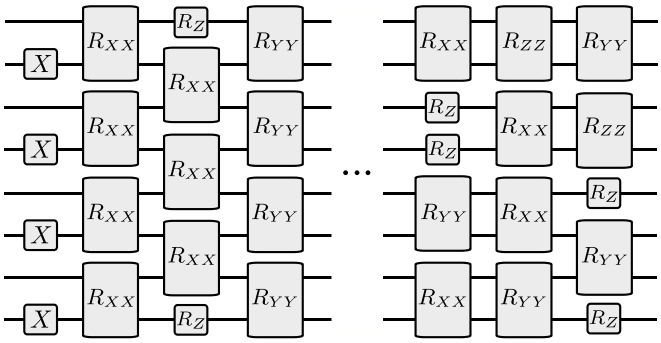

FIG. 6. Variational circuit obtained at $J_x t = 2$ in the simulation shown in Fig. 3, using the adaptive pVQD algorithm and local operator pool.

(3) *Remove the operators in the pool that act on qubit(s) already acted on*. Given that the operator A^* obtained in Step (2) acts on the qubits indices $\boldsymbol{\alpha}$, the subset of the operator pool that also acts on at least one index in $\boldsymbol{\alpha}$, namely

$$\mathcal{A}_{\boldsymbol{\alpha}} = \{A_a | A_a \in \mathcal{A} \text{ acts on } \boldsymbol{\beta}, \boldsymbol{\beta} \cap \boldsymbol{\alpha} \neq \emptyset\} \qquad \text{(C1)}$$

should be removed from the current operator pool. Hence the pool can be updated as follows: $\mathcal{A} \to \mathcal{A} \setminus \mathcal{A}_{\boldsymbol{\alpha}}$.

(4) *Go back to Step (2) until the operator pool is empty*.

(5) *Return the new circuit*. The new parameterized circuit is characterized by $\boldsymbol{\theta} \to (\boldsymbol{\theta}, 0, \cdots, 0)$ and $\boldsymbol{A} \to (\boldsymbol{A}, A_0^*, A_1^*, \cdots, A_k^*)$, assuming that k new operators were added.

As stated in the main text, this procedure guarantees that the depth of the parameterized circuit $|\psi(\boldsymbol{\theta})\rangle$ is increased by 1 in each adaptive step [53]. We note that this remains true even when no device connectivity is assumed. Indeed, the procedure guarantees that the operators added have disjoint support, which increases the depth only by 1. However, a depth increment larger than 1 might occur after the circuit is transpiled to a specific hardware with limited connectivity.

APPENDIX D: ADAPTIVE PVQD OUTPUT CIRCUITS

We illustrate in Figs. 6 and 7 examples of parameterized circuits obtained with the adaptive pVQD algorithm in

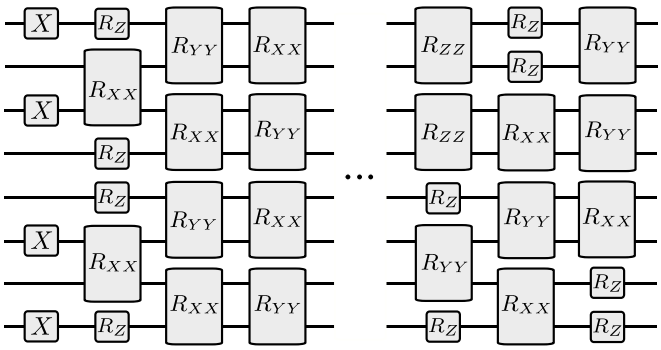

FIG. 7. Variational circuit obtained at $Jt = 4$ in the simulation shown in Fig. 5, using the adaptive pVQD algorithm and local operator pool.

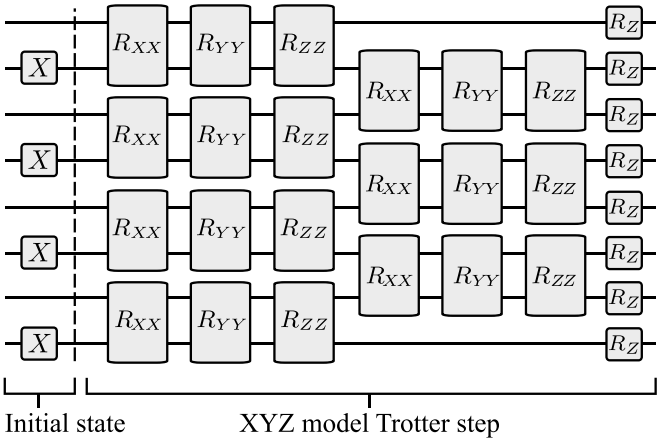

FIG. 8. Implementation of an antiferromagnetic initial state and a Trotter step for the driven Heisenberg model given in Eq. (9).

simulations shown in the main text. Each column of operators in the circuits corresponds to an adaptive step.

APPENDIX E: TROTTER STEP CIRCUIT ENCODINGS

In this Appendix, we provide the circuits used to implement a single Trotter step of the driven Heisenberg model and the Hubbard model. The Trotter step in the driven Heisenberg model is implemented with a checkerboard pattern of the two-qubit gates R_{XX}, R_{YY}, R_{ZZ}, with a layer of single-qubit R_Z at the end. We show a sketch in Fig. 8.

To realize the Trotter circuit for the Hubbard model, we first have to establish an ordering of the lattice sites and the fermionic modes. We number the sites using a snake-like pattern and, as indicated in the main text, we eliminate the spin index via $c_{i\uparrow} \to c_i$ and $c_{i\downarrow} \to c_{i+N/2}$. Under this ordering, the Jordan-Wigner transformation of the Hamiltonian terms reads

$$c_{i\uparrow}^\dagger c_{j\uparrow} + c_{j\uparrow}^\dagger c_{i\uparrow} \mapsto \frac{1}{2}\left[X_i \prod_{k=i+1}^{j-1} Z_k X_j + Y_i \prod_{k=i+1}^{j-1} Z_k Y_j\right], \quad \text{(E1)}$$

$$c_{i\downarrow}^\dagger c_{j\downarrow} + c_{j\downarrow}^\dagger c_{i\downarrow} \mapsto \frac{1}{2}\left[X_{i+N/2} \prod_{k=i+1}^{j-1} Z_{k+N/2} X_{j+N/2} + Y_{i+N/2} \prod_{k=i+1}^{j-1} Z_{k+N/2} Y_{j+N/2}\right], \quad \text{(E2)}$$

$$n_{i\uparrow} n_{i\downarrow} \mapsto \frac{1}{4}(\mathbb{I} - Z_i)(\mathbb{I} - Z_{i+N/2}), \quad \text{(E3)}$$

where we assumed $j > i$ without loss of generality. Given the mapped Hamiltonian, the Trotter step cannot be implemented only using R_{XX}, R_{YY}, R_{ZZ} and R_Z gates. Indeed, the nonlocality of the mapping requires some multiqubit rotation with size up to $2L_x$. The two multiqubit gates arising for $N = 8$ are the rotations generated by the Pauli strings $XZZX$ and $YZZY$, which can be decomposed as shown in [49]. Figure 9 presents our implementation.

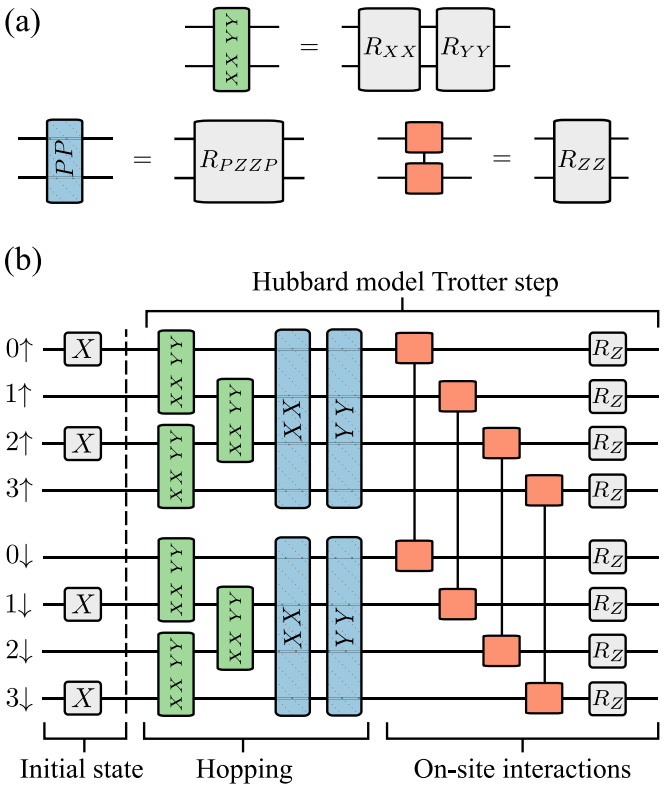

FIG. 9. (a) Gates used to define a Trotter step. (b) Quantum circuit encoding the first-order Trotter step of the Hubbard model with an half-filled antiferromagnetic initial state.

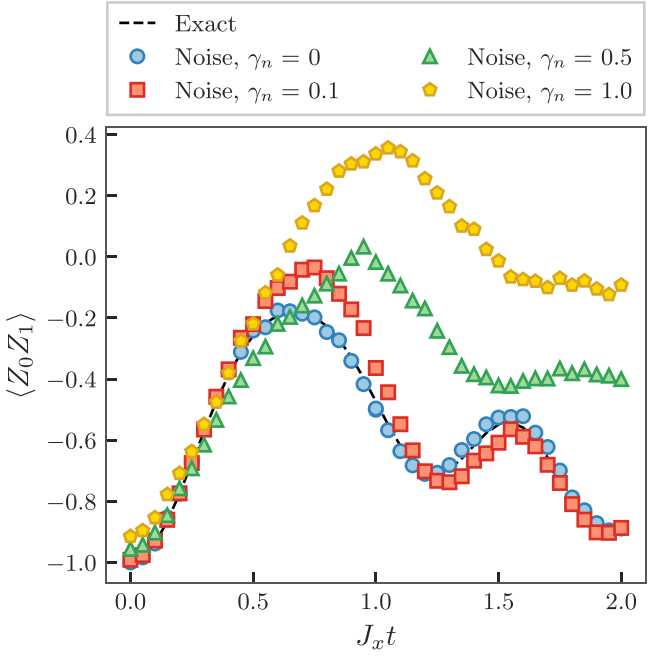

FIG. 10. Expectation value of the correlator Z_0Z_1 measured with the IBM Brisbane noise model for the driven Heisenberg XYZ model on an open chain with 4 sites, $J_x = 1, J_y = 0.8, J_z = 0.6$ and an antiferromagnetic initial state $|\psi_0\rangle = |0101\rangle$. The whole optimization is performed on the noisy simulator and every circuit is evaluated $n_{\text{shot}} = 10^4$ times. The simulations are performed using different noise scaling factor γ_n. $\gamma_n = 0$ refers to an ideal quantum device with only statistical shot noise.

APPENDIX F: EFFECTS OF HARDWARE NOISE ON VARIATIONAL OPTIMIZATION

In this Appendix, we analyze the effects of quantum hardware noise on the accuracy of the adaptive pVQD. To assess the possibility to run the adaptive pVQD algorithm on current quantum devices, we perform classical simulations of the density matrix of the quantum circuits using a noise model. These noise models can be obtained through the IBM Quantum platform [73] and include several types of error that can occur in a real device [59]. We note that even if accurate, the noise models represent a best case scenario with respect to hardware results. First, we import the noise model of IBM Brisbane, a 127-qubits superconducting quantum device, and we use it to run the adaptive pVQD algorithm on the driven Heisenberg model of Eq. (9) for $L = 4$. The results are shown in Fig. 10, and they indicate that even if the circuits are shallower than those obtained via Trotter decomposition, for $J_x t > 0.5$ they are too deep to provide qualitatively good results. Therefore, we introduce a noise scaling factor γ_n to rescale every error rate inside the noise model. The noise scaling factor γ_n is defined such that when $\gamma_n = 1$ the noise model is imported without rescaling, while $\gamma_n = 0$ indicates an ideal quantum device, where only shot noise affects the results. This factor acts on all error present in the noise model: the depolarizing channel, the preparation and measurement channels, thermal decoherence, and the dephasing channel [59]. Each channel can be rescaled independently, but for clarity of comparison we decided to use a single, uniform scaling factor. The code to implement the rescaled error models can be found on Github [70]. From Fig. 10, we note that a superconducting quantum device with error rates comparable to the model rescaled with $\gamma_n = 0.1$ would yield results in qualitative agreement with the exact solution. We conclude that current devices are still not able to run this type of variational algorithms, but we can use the circuits obtained in the classical simulations to measure observables at longer times more accurately than the Trotterization approach, as shown in the main text.

[1] A. W. Sandvik, A. Avella, and F. Mancini, Computational studies of quantum spin systems, AIP Conf. Proc. **1297**, 135 (2010).

[2] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[3] R. Orús, Tensor networks for complex quantum systems, Nat. Rev. Phys. **1**, 538 (2019).

[4] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[5] C. W. Cady, R. H. Crabtree, and G. W. Brudvig, Functional models for the oxygen-evolving complex of photosystem II, Coord. Chem. Rev. **252**, 444 (2008).

[6] L. Schimka, J. Harl, A. Stroppa, A. Grüneis, M. Marsman, F. Mittendorfer, and G. Kresse, Accurate surface and adsorption energies from many-body perturbation theory, Nat. Mater. **9**, 741 (2010).

[7] A. J. Leggett, What DO we know about high T_c? Nat. Phys. **2**, 134 (2006).

[8] L. Balents, Spin liquids in frustrated magnets, Nature (London) **464**, 199 (2010).

[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[10] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).

[11] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, J. Preskill, and J. R. McClean, Quantum advantage in learning from experiments, Science **376**, 1182 (2022).

[12] F. Dolde, V. Bergholm, Y. Wang, I. Jakobi, B. Naydenov, S. Pezzagna, J. Meijer, F. Jelezko, P. Neumann, T. Schulte-Herbrüggen *et al.*, High-fidelity spin entanglement using optimal control, Nat. Commun. **5**, 3371 (2014).

[13] G. Waldherr, Y. Wang, S. Zaiser, M. Jamali, T. Schulte-Herbrüggen, H. Abe, T. Ohshima, J. Isoya, J. F. Du, P. Neumann, and J. Wrachtrup, Quantum error correction in a solid-state hybrid spin register, Nature (London) **506**, 204 (2014).

[14] Y. Nam, J.-S. Chen, N. C. Pisenti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf *et al.*, Ground-state energy estimation of the water molecule on a trapped ion quantum computer, npj Quantum Inf. **6**, 33 (2020).

[15] Y. Wan, R. Jördens, S. D. Erickson, J. J. Wu, R. Bowler, T. R. Tan, P.-Y. Hou, D. J. Wineland, A. C. Wilson, and D. Leibfried, Ion transport and reordering in a 2D trap array, Adv. Quantum Technol. **3**, 2000028 (2020).

[16] A. C. Hughes, V. M. Schäfer, K. Thirumalai, D. P. Nadlinger, S. R. Woodrow, D. M. Lucas, and C. J. Ballance, Benchmarking a high-fidelity mixed-species entangling gate, Phys. Rev. Lett. **125**, 080504 (2020).

[17] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, Nature (London) **618**, 500 (2023).

[18] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[19] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[20] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3**, 191 (2019).

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

[22] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[23] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[24] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[25] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximate optimization algorithm for MaxCut: A fermionic view, Phys. Rev. A **97**, 022304 (2018).

[26] P. D. Johnson, J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik, QVECTOR: an algorithm for device-tailored quantum error correction, arXiv:1711.02249.

[27] X. Xu, S. C. Benjamin, and X. Yuan, Variational circuit compiler for quantum error correction, Phys. Rev. Appl. **15**, 034068 (2021).

[28] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, New J. Phys. **20**, 113022 (2018).

[29] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum-assisted quantum compiling, Quantum **3**, 140 (2019).

[30] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles, Noise resilience of variational quantum compiling, New J. Phys. **22**, 043006 (2020).

[31] T. Jones and S. C. Benjamin, Robust quantum compilation and circuit optimisation via energy minimisation, Quantum **6**, 628 (2022).

[32] Z. Lu, P.-X. Shen, and D.-L. Deng, Markovian quantum neuroevolution for machine learning, Phys. Rev. Appl. **16**, 044039 (2021).

[33] Y. Li and S. C. Benjamin, Efficient variational quantum simulator incorporating active error minimization, Phys. Rev. X **7**, 021050 (2017).

[34] C. Cîrstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, Variational fast forwarding for quantum simulation beyond the coherence time, npj Quantum Inf. **6**, 82 (2020).

[35] Y.-X. Yao, N. Gomes, F. Zhang, C.-Z. Wang, K.-M. Ho, T. Iadecola, and P. P. Orth, Adaptive variational quantum dynamics simulations, PRX Quantum **2**, 030307 (2021).

[36] S.-H. Lin, R. Dilip, A. G. Green, A. Smith, and F. Pollmann, Real- and imaginary-time evolution with compressed quantum circuits, PRX Quantum **2**, 010342 (2021).

[37] F. Barratt, J. Dborin, M. Bal, V. Stojevic, F. Pollmann, and A. G. Green, Parallel quantum simulation of large systems on small NISQ computers, npj Quantum Inf. **7**, 79 (2021).

[38] M. Benedetti, M. Fiorentini, and M. Lubasch, Hardware-efficient variational quantum algorithms for time evolution, Phys. Rev. Res. **3**, 033083 (2021).

[39] S. Barison, F. Vicentini, and G. Carleo, An efficient quantum algorithm for the time evolution of parameterized circuits, Quantum **5**, 512 (2021).

[40] N. F. Berthusen, T. V. Trevisan, T. Iadecola, and P. P. Orth, Quantum dynamics simulations beyond the coherence time on noisy intermediate-scale quantum hardware by variational trotter compression, Phys. Rev. Res. **4**, 023097 (2022).

[41] S. Barison, F. Vicentini, I. Cirac, and G. Carleo, Variational dynamics as a ground-state problem on a quantum computer, Phys. Rev. Res. **4**, 043161 (2022).

[42] A. Miessen, P. J. Ollitrault, F. Tacchino, and I. Tavernelli, Quantum algorithms for quantum dynamics, Nat. Comput. Sci. **3**, 25 (2023).

[43] C. Mc Keever and M. Lubasch, Classically optimized Hamiltonian simulation, Phys. Rev. Res. **5**, 023146 (2023).

[44] H. Zhao, M. Bukov, M. Heyl, and R. Moessner, Making Trotterization adaptive and energy-self-correcting for NISQ devices and beyond, PRX Quantum **4**, 030319 (2023).

[45] H. F. Trotter, On the product of semi-groups of operators, Proc. Am. Math. Soc. **10**, 545 (1959).

[46] M. Suzuki, Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems, Commun. Math. Phys. **51**, 183 (1976).

[47] D. S. Abrams and S. Lloyd, Simulation of many-body Fermi systems on a universal quantum computer, Phys. Rev. Lett. **79**, 2586 (1997).

[48] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, Quantum algorithms for fermionic simulations, Phys. Rev. A **64**, 022319 (2001).

[49] 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 (2020).

[50] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, Nat. Commun. **10**, 3007 (2019).

[51] H. L. Tang, V. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor, PRX Quantum **2**, 020310 (2021).

[52] J. S. Van Dyke, G. S. Barron, N. J. Mayhall, E. Barnes, and S. E. Economou, Scaling adaptive quantum simulation algorithms via operator pool tiling, Phys. Rev. Res. **6**, L012030 (2024).

[53] P. G. Anastasiou, Y. Chen, N. J. Mayhall, E. Barnes, and S. E. Economou, TETRIS-ADAPT-VQE: An adaptive algorithm that yields shallower, denser circuit Ansätze, Phys. Rev. Res. **6**, 013254 (2024).

[54] N. Gomes, D. B. Williams-Young, and W. A. de Jong, Computing the many-body Green's function with adaptive variational quantum dynamics, J. Chem. Theory Comput. **19**, 3313 (2023).

[55] P. G. Anastasiou, N. J. Mayhall, E. Barnes, and S. E. Economou, How to really measure operator gradients in ADAPT-VQE, arXiv:2306.03227.

[56] A. Mari, T. R. Bromley, and N. Killoran, Estimating the gradient and higher-order derivatives on quantum hardware, Phys. Rev. A **103**, 012405 (2021).

[57] V. O. Shkolnikov, N. J. Mayhall, S. E. Economou, and E. Barnes, Avoiding symmetry roadblocks and minimizing the measurement overhead of adaptive variational quantum eigensolvers, Quantum **7**, 1040 (2023).

[58] Y. S. Yordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, Qubit-excitation-based adaptive variational quantum eigensolver, Commun. Phys. **4**, 228 (2021).

[59] K. Georgopoulos, C. Emary, and P. Zuliani, Modeling and simulating the noisy behavior of near-term quantum computers, Phys. Rev. A **104**, 062432 (2021).

[60] K. Temme, S. Bravyi, and J. M. Gambetta, Error mitigation for short-depth quantum circuits, Phys. Rev. Lett. **119**, 180509 (2017).

[61] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. Jose Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen *et al.*, Qiskit: An open-source framework for quantum computing, Zenodo, https://doi.org/10.5281/zenodo.2562111 (2023).

[62] E. van den Berg, Z. K. Minev, A. Kandala, and K. Temme, Probabilistic error cancellation with sparse Pauli-Lindblad models on noisy quantum processors, Nat. Phys. **19**, 1116 (2023).

[63] E. W. E. P. Jornan, *The Collected Works of Eugene Paul Wigner* (Springer, Berlin, 1993).

[64] S. B. Bravyi and A. Y. Kitaev, Fermionic quantum computation, Ann. Phys. **298**, 210 (2002).

[65] F. Verstraete and J. I. Cirac, Mapping local Hamiltonians of fermions to local Hamiltonians of spins, J. Stat. Mech. (2005) P09012.

[66] J. D. Whitfield, V. Havlíček, and M. Troyer, Local spin operators for fermion simulations, Phys. Rev. A **94**, 030301(R) (2016).

[67] K. Setia, S. Bravyi, A. Mezzacapo, and J. D. Whitfield, Superfast encodings for fermionic quantum simulation, Phys. Rev. Res. **1**, 033033 (2019).

[68] Y.-A. Chen and Y. Xu, Equivalence between fermion-to-qubit mappings in two spatial dimensions, PRX Quantum **4**, 010326 (2023).

[69] J. Nys and G. Carleo, Quantum circuits for solving local fermion-to-qubit mappings, Quantum **7**, 930 (2023).

[70] D. Linteau, adaptive-pvqd, https://github.com/dalin27/adaptive-pvqd (2022).

[71] J. R. Johansson, P. D. Nation, and F. Nori, QuTiP: An open-source Python framework for the dynamics of open quantum systems, Comput. Phys. Commun. **183**, 1760 (2012).

[72] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[73] IBM, IBM Quantum, https://quantum.ibm.com/, accessed on April, 2024.