Hardware-efficient variational quantum algorithms for time evolution

Marcello Benedetti,[*] Mattia Fiorentini, and Michael Lubasch
Cambridge Quantum Computing Limited, SW1E 6DR London, United Kingdom

(Dated: September 29, 2020)

Parameterized quantum circuits are a promising technology for achieving a quantum advantage. An important application is the variational simulation of time evolution of quantum systems. To make the most of quantum hardware, variational algorithms need to be as hardware-efficient as possible. Here we present alternatives to the time-dependent variational principle that are hardware-efficient and do not require matrix inversion. In relation to imaginary time evolution, our approach significantly reduces the hardware requirements. With regards to real time evolution, where high precision can be important, we present algorithms of systematically increasing accuracy and hardware requirements. We numerically analyze the performance of our algorithms using quantum Hamiltonians with local interactions.

I. INTRODUCTION

Small quantum computers are available today and offer the exciting opportunity to explore classically difficult problems for which a quantum advantage may be achievable. The simulation of time evolution of quantum systems is an example of such problems where the advantage of using a quantum computer over a classical one is well understood [1–5]. This simulation is also important for our understanding of quantum chemistry and materials science which are key application areas for future quantum computers [6–9]. One of the main challenges in the design of time evolution algorithms is to reduce their experimental requirements without sacrificing their accuracy.

Although significant progress has been made based on the original quantum algorithm for simulating time evolution [2], this algorithm faces obstacles on current quantum hardware which lacks quantum error correction [10–12]. Promising alternatives are variational hybrid quantum-classical algorithms [13, 14] and variational quantum simulation [15, 16]. In these approaches, a parameterized quantum circuit (PQC) is prepared on a quantum computer and variationally optimized to solve the problem of interest. Promising examples of quantum advantage obtained with PQCs have already been identified for time evolution [17, 18] and in other contexts such as for nonlinear partial differential equations [19, 20], dynamical mean field theory [21, 22], and machine learning [23].

In this article, we focus on making the simulation of time evolution as hardware-efficient as possible. Existing proposals are based on the time-dependent variational principle or variants thereof [15, 16]. These approaches require matrix inversion which poses computational challenges for ill-conditioned matrices. Our contribution is a set of alternative techniques that do not need matrix inversion and that allow one to systematically increase the simulation accuracy, together with the hardware requirements, according to experimental capabilities. We analyze the hardware/accuracy trade-off and show that for imaginary time evolution our algorithm significantly reduces the hardware requirements over existing methods and produces accurate ground state approximations. For real time evolution, the accuracy per time step is essential. We present a hierarchy of algorithms where the accuracy can be systematically improved by utilizing more hardware resources.

Our strategy was inspired by several tensor network concepts. Firstly, we apply the Trotter product formula to the time evolution operator and optimize the ansatz one Trotter term after another. A similar procedure is used in the time-evolving block decimation algorithm [24–27] as well as with projected entangled pair states [28–30]. Secondly, for a given Trotter term, we restrict the optimization to its causal cone. This is an important concept for the multiscale entanglement renormalization ansatz [31, 32] as well as for matrix product states [33]. The same concept was used to simulate infinite matrix product states on a quantum computer [34]. Thirdly, we adapt the Rotosolve algorithm [35] to perform block-wise optimization, i.e., we optimize a set of parameters at a time while keeping all the others fixed. A similar approach is widely used in tensor network optimization where tensors are optimized one after another [36, 37].

The article has the following structure. Section II summarizes our methods, Section III presents our mathematical and numerical results, and Section IV contains our concluding discussion. The technical details are in the appendices.

[*] marcello.benedetti@cambridgequantum.com

II. METHODS

We present our methods in four stages. Firstly, we break time evolution into a sequence of variational problems. Secondly, we use PQC ansätze and causal cones to reduce the number of required qubits. Thirdly, we formulate the Rotosolve equations for optimizing the variational parameters. Finally we introduce approximations leading to hardware-efficient implementations.

A. Taking apart time evolution

In the following, we focus on real time evolution. To obtain imaginary time evolution one simply needs to substitute t by $-it$.

The simulation of real time evolution consists of approximating the action of the operator $e^{-it\hat{H}}$ on an initial state $|\psi_0\rangle$ of n qubits. The Hamiltonian is assumed to have the general form $\hat{H} = \sum_{k=1}^{K} h_k \hat{H}_k$ where $\hat{H}_k$ are tensor products of Pauli operators, h_k are real numbers, and $K \sim O(\text{poly}(n))$. We approximate the evolution by a sequence of short-time evolutions using the well-known Trotter product formula. With N time steps of size $\tau = t/N$ one obtains $e^{-it\hat{H}} \approx U(\tau)^N$, where $U(\tau) = e^{-i\tau h_K \hat{H}_K} \cdots e^{-i\tau h_1 \hat{H}_1}$. Notice that the accuracy of our Trotter approximation can be improved using higher-order formulas [38].

We variationally simulate the sequence of Trotter terms one term at a time. Consider the k-th term, $e^{-i\tau h_k \hat{H}_k}$, and let $|\psi_{k-1}\rangle$ be the variational approximation to the previous step. As shown in Appendix A, the current step can be simulated by maximizing the objective function:

$$\mathcal{F}_k(\boldsymbol{\theta}) = \text{Re}\left(\langle\psi_{k-1}| \, e^{i\tau h_k \hat{H}_k} \, |\psi(\boldsymbol{\theta})\rangle \right), \qquad (1)$$

with respect to the variational parameters $\boldsymbol{\theta}$, where $\text{Re}(\cdot)$ denotes the real part of a complex number. This optimization is carried out for all K terms in $U(\tau)$. The process is repeated N times, after which the simulation of time evolution is completed.

Figure 1 illustrates a few steps of the method. The variational state consists of a PQC (light blue rectangle) acting on the $|\mathbf{0}\rangle \equiv |0\rangle^{\otimes n}$ state of the computational basis. At each step, a term of the Trotter formula is selected (blue rectangle) and simulated by the variational method.

B. Parameterized quantum circuits and causal cones

We consider PQC ansätze composed of generic 2-qubit unitaries acting on nearest neighbors, as shown in Fig. 2 (a). The required 1d qubit-to-qubit connectivity matches that of many existing quantum computers. Yet, these ansätze are universal for quantum computation if we adjust their depth accordingly. We refer to the unitaries $U[i,j]$ (light blue rectangles) as *blocks*. In practice, they are made of gates from the hardware's gate set.

An interesting property is that the expectation of an operator acting on a few qubits depends only on the blocks inside its causal cone. Figure 2 (b) illustrates an example of causal cone for a 2-qubit operator. The expectation can be estimated with a quantum circuit of 6 qubits and 5 blocks, regardless of the overall size of the PQC ansätze.

This simple idea allows us to work with variational states that are significantly larger and more general than what is allowed by the available quantum computers. For example, periodic boundary conditions can be included with no additional hardware requirements. In Fig. 2 (a), block $U[\frac{n}{2}, 2]$ operates on the first and last qubits. It is easy to see that such a physical qubit-to-qubit connectivity is not required when using causal cones. The number of required physical qubits depends only on the depth of the ansatz and on the operator. For long-range operators or operators acting on more than 2 qubits one can obtain several causal cones.

Alternative qubit-efficient schemes based on measuring and resetting qubits during computation were proposed in Ref. [39]. Alternative approaches for periodic quantum systems were developed in Refs. [40, 41].

C. The update rule for Rotosolve

Let us specialize our discussion to PQCs of the form $U(\boldsymbol{\theta}) = U_D \cdots U_1$ where each gate is either fixed, e.g., a CNOT, or parameterized as $U_d = \exp(-i\theta_d G_d)$, where $\theta_d \in (-\pi, \pi]$ and G_d is a Hermitian and unitary ~~matrix such that~~ $G_d^2 = I$. This standard parameterization has nice properties that we exploit to design optimization algorithms. In Ref. [35], the authors showed that when all parameters but one are fixed, expectations have sinusoidal form. Therefore, there is an analytic expression for the extrema.

Let us now consider our objective in Eq. (1). We define the coordinate-wise objective for the d-th parameter as $f_{k,d}(x) \equiv \mathcal{F}_k(\theta_1, \cdots, \theta_{d-1}, x, \theta_{d+1}, \cdots, \theta_D)$, where

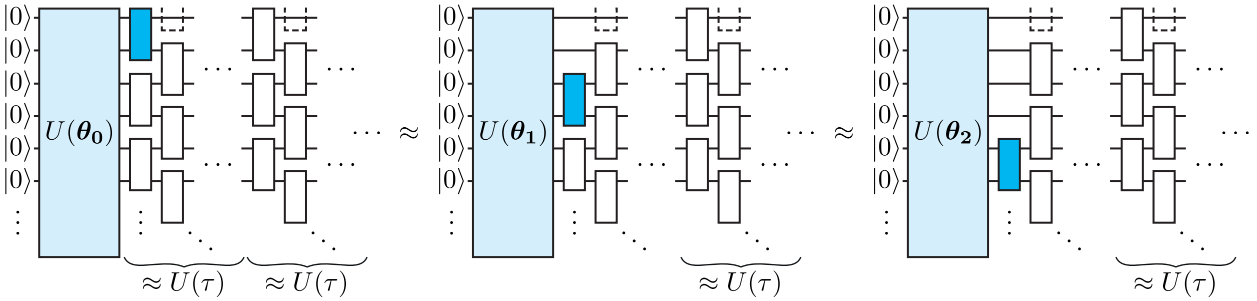
|0⟩
$U(\boldsymbol{\theta_0})$
$U(\boldsymbol{\theta_1})$
$U(\boldsymbol{\theta_2})$
≈
$\approx U(\tau)$
$\approx U(\tau)$
$\approx U(\tau)$
$\approx U(\tau)$

FIG. 1. The Trotter product formula applied to the time evolution operator results in repeated products of $U(\tau)$, where τ is the time step, acting on the initial state $|\psi_0\rangle = U(\boldsymbol{\theta_0})\,|\mathbf{0}\rangle$. We approximate time evolution one term after another, each time finding the optimal variational parameters $\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \ldots$

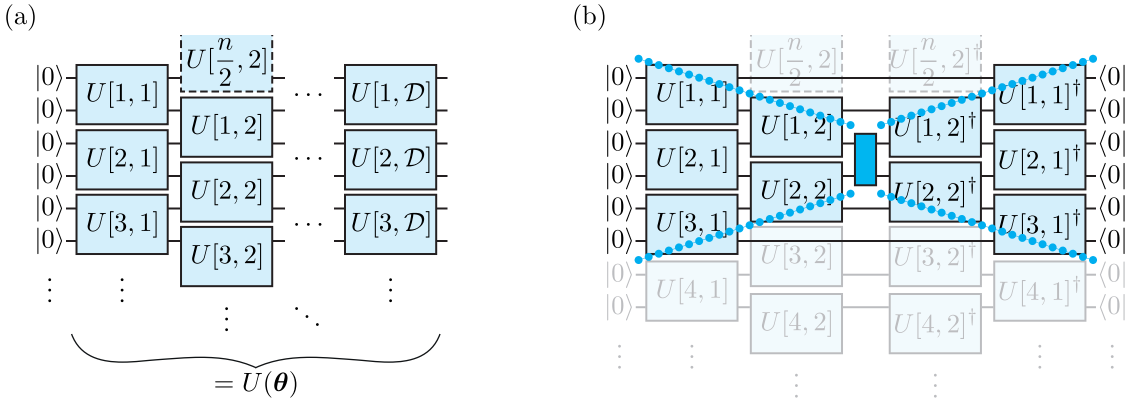

FIG. 2. (a) PQC ansatz for n qubits and depth $\mathcal{D}$. We consider open or periodic boundary conditions. For periodic boundary conditions the unitary block $U[\frac{n}{2},2]$ connects the first and the last qubit. (b) The computation of an expectation simplifies if one considers only the blocks inside the causal cone. We use this technique to attack problems of size larger than that of the available quantum hardware.

all parameters but one are fixed to their current value. In Appendix B we show that this is a sinusoidal function $f_{k,d}(x) = A_{k,d}\sin(x + B_{k,d})$ with amplitude $A_{k,d}$ and phase $B_{k,d}$. The result allows us to use the Rotosolve algorithm [35]. Rotosolve is a coordinate-wise optimization algorithm that neither requires the gradient nor the Hessian. It sweeps through all parameters and sets each of them to their locally optimal value $x^* = \frac{\pi}{2} - B_{k,d}$. At x^* the coordinate-wise objective attains ~~its maximum~~ value of $A_{k,d}$.

The estimation of $A_{k,d}$ and $B_{k,d}$ is efficient and leads to a simple update rule. For the d-th parameter and for the k-th term we have:

$$\theta_d \leftarrow \frac{\pi}{2} - \arctan\left(\frac{f_{k,d}(\theta_d)}{f_{k,d}(\theta_d + \frac{\pi}{2})}\right) + \theta_d, \qquad (2)$$

where θ_d in the right side is the current value of the parameter.

This formula requires evaluating the objective at $x = \theta_d$ and $x = \theta_d + \frac{\pi}{2}$. However, in Appendix C we show that $f_{k,d}(\theta_d) = A_{k,d-1}$ is known from the previous step. Thus, the method finds the maxima with a single evaluation, $f_{k,d}(\theta_d + \frac{\pi}{2})$.

D. Hardware-efficient implementation

In general, our method optimizes a new PQC for each Trotter term. Let us denote the $(k-1)$-th quantum state as $|\psi_{k-1}\rangle = V\,|\mathbf{0}\rangle$, and the k-th state as $|\psi(\boldsymbol{\theta})\rangle = U\,|\mathbf{0}\rangle$, where V and U are PQCs. The objective in Eq. (1) and the update rule in Eq. (2) can be estimated using a well-known primitive called the Hadamard test.

The Hadamard test can be challenging to execute on hardware when U and V are unrelated quantum circuits due to the potentially large number of controlled operations. The process can be largely simplified if U and V differ only locally, e.g., in few gates or circuit regions. For instance, if one circuit can be efficiently transformed into the other using local adjoints of gates, then the Hadamard test consists of a rather simple quantum circuit. Figure 3 (a) shows

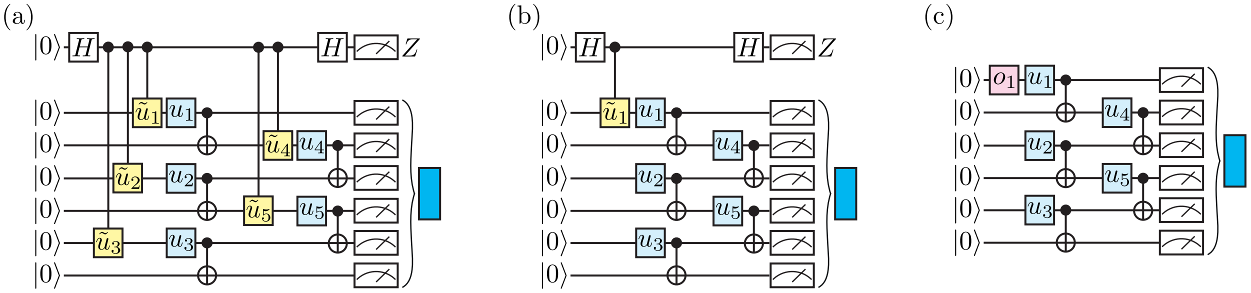

FIG. 3. Hadamard tests required for our algorithms. For illustrative purposes, we show a simple PQC ansatz made of 5 CNOTs and 5 parameterized gates u_i (light blue). (a) In Cone Update the previous and current PQCs differ by at most 5 gates. The test requires only local transformations such as the controlled-$\tilde{u}_i$ (yellow). (b) In Angle Update the two PQCs differ by 1 gate. (c) Angle Update can be made hardware-efficient by removing the ancilla qubit and the controlled operation, and introducing an operation o_i (pink). This further simplification is presented in Appendix E.

an example with 5 variational parameters. Gates u_i represent U. Gates controlled-$\tilde{u}_i$ represent the local transformations taking U to V. The subspaces where the ancilla (top qubit) is $|0\rangle$ and $|1\rangle$ contain $U|\mathbf{0}\rangle$ and $V|\mathbf{0}\rangle$, respectively. Finally, a measurement yields the quantity in Eq. (1). This is discussed in detail in Appendix D.

Clearly, we only need to optimize the blocks in the causal cone of the Trotter term, see Fig. 2 (b). We call this approach the Cone Update. In Cone Update, each Hadamard test requires $O(N_b N_p)$ controlled gates where N_b is the number of blocks in the causal cone, and N_p is the number of parameters in a block.

The closer U and V remain during the execution of the algorithm, the fewer controlled gates are required for the Hadamard test. This suggests a systematic way to reduce hardware requirements by introducing approximations to the objective. The first level of approximation consists of replacing $|\psi_{k-1}\rangle$ in Eq. (1) with the current variational state $|\psi(\boldsymbol{\theta})\rangle$ after a block has been updated. This way, U and V differ by $O(N_p)$ parameters, greatly simplifying the Hadamard test. Note that the replacement is performed once for each of the N_b blocks in the causal cone. Hence, to effectively simulate time step τ we use a time step τ/N_b in the objective. We call this approach the Block Update.

The second level of approximation consists of replacing $|\psi_{k-1}\rangle$ in the objective with $|\psi(\boldsymbol{\theta})\rangle$ after an angle has been updated. This guarantees that U and V differ by one parameter at all times. Since the replacement is done $N_b N_p$ times, we use a time step $\tau/(N_b N_p)$ in the objective. We call this approach the Angle Update. Figure 3 (b) shows an example with 5 parameters and where the Hadamard test requires a single controlled gate. Angle Update can be further simplified by replacing the indirect measurement with direct ones [42, 43]. This removes the need for the ancillary qubit and the controlled gate resulting in the circuit shown in Fig. 3 (c). The Rotosolve equations for the hardware-efficient algorithm are derived in Appendix E.

III. RESULTS

In this Section, we first present an analysis of the hardware requirements and errors of our algorithms. Then we benchmark their performance using numerical experiments.

A. Analysis of resources and errors

We compare the hardware requirements of our update methods in Table I. Here N_b denotes the number of blocks inside the causal cone of one Trotter term, N_p denotes the number of parameters per block, and we assume that every block has the same number of parameters. We conclude that the hardware-efficient Angle Update requires the smallest amount of resources and no matrix inversion which makes this a stable and efficient algorithm for simulating time evolution. Block Update interpolates between Angle and Cone Update in a natural way. If we fix a maximum possible block size $N_p^{\max}$ beforehand – for example to a value related to given hardware constraints – we consider the corresponding Block Update for $N_p \leq N_p^{\max}$ to be hardware-efficient. In general, however, the technique used to achieve hardware-efficiency on N_p controlled gates requires $O(e^{\alpha N_p})$ circuits for some $\alpha > 0$ [43]. Therefore, Block Update with arbitrary block size, as well as Cone Update, are not hardware-efficient.

Table I also shows the hardware requirements for

Update method	Circuits per sweep	Matrix inversion	Controlled gates per circuit	Hardware-efficient
Cone	$O(N_b N_p)$	No	$O(N_b N_p)$	No
Block	$O(N_b N_p)$	No	$O(N_p)$	Yes/No
Angle	$O(N_b N_p)$	No	$O(1)$	Yes
TDVP	$O(N_b^2 N_p^2)$	Yes	$O(1)$	Yes

TABLE I. Characteristics of Cone, Block, and Angle Update along with the TDVP methods of Ref. [16]. A *sweep* consists of updating all parameters inside the causal cone of a Trotter term exactly once.

the TDVP methods of Ref. [16]. Compared with these TDVP methods, our update procedures have the advantage that they do not need matrix inversion. Matrix inversion is numerically unstable when the condition number of the matrix is large and small errors in the matrix can become large errors in the matrix inverse [44]. The TDVP methods of Ref. [16] compute the matrix elements from mean values over several measurements. For a desired accuracy ϵ per matrix element this procedure requires $O(1/\epsilon^2)$ measurements. To determine the accuracy of the time-evolved parameters after a TDVP udpate, we need to take into account the condition number κ of the matrix because the TDVP update needs the matrix inverse. We show in Appendix F that for a desired accuracy ϵ in the time-evolved parameters the required number of measurements scales as $O(\kappa^2/\epsilon^2)$ in the worst case. Therefore, for ill-conditioned matrices, the TDVP methods of Ref. [16] may need to compute matrix elements very accurately and require many measurements.

We obtained some numerical evidence by computing the median condition number $\tilde{\kappa}$ of 100 random initializations of our ansatz. To avoid instabilities, we ignored singular values smaller than 10^{-7}. With 15 parameters we had $\tilde{\kappa} \approx 16$, with 45 parameters we had $\tilde{\kappa} \approx 5001$, and with 75 parameters we had $\tilde{\kappa} \approx 17085$. This rapid increase of the condition number as a function of the number of variational parameters indicates a challenging scaling of TDVP cost for our choice of ansatz.

In our methods there exist two sources of error. Firstly, there is a Trotter error resulting from the Trotter product formula that was used to split the time evolution operator. Secondly, there is an approximation error per Trotter term resulting from the limitations of our PQC ansatz and optimization methods. Both errors can be quantified and systematically decreased. The Trotter error is known and can be decreased by using higher-order Trotter product formulas. The approximation error is computed during PQC optimization and can be systematically decreased by using Block Update with successively increasing block sizes and adding new variational parameters to the PQC. Therefore the complete error per time step scales like $O((\tau\|\hat{H}\|)^{\alpha_{\text{Trotter}}} + K\epsilon_{\text{approx}})$ where α_{Trotter} is determined by the Trotter product formula, K is the number of terms in the Hamiltonian, and ϵ_{approx} represents the largest approximation error encountered during the optimization.

B. Numerical experiments

We performed two numerical experiments to validate our methods. Further details are provided in Appendix G. First, we looked into the accuracy of Cone, Block, and Angle Update. This is important, in particular, for real time evolution where the variational state should closely track the true evolved state at all times. In other words, the objective in Eq. (1) should be very close to 1 for any Trotter term. It is interesting to analyze the accuracy as a function of the number of optimization sweeps.

We chose the ansatz in Fig. 2 (a) with depth $\mathcal{D} = 2$ and with periodic boundary conditions. Regardless of the number of qubits n, there are only two possible causal cones: a 4-qubit cone enclosing $N_b = 3$ blocks, if the term is located in front of a block, and a 6-qubit cone enclosing $N_b = 5$ blocks, if the term is located in front of two blocks (the latter case is shown in Fig. 2 (b)). We selected Trotter terms $\exp(-i\tau\hat{\sigma}_j \otimes \hat{\sigma}_{j+1})$ at random and used a rather large time step $\tau = 0.1$. For Block Update we use $\tau/(N_s N_b)$ and for Angle Update we use $\tau/(N_s N_b N_p)$, where the extra factor of N_s is the number of sweeps.

Figure 4 (a) shows the mean objective and standard deviation for the case of 4-qubit cones. Cone Update outperforms and converges to the optimal value of 1, while Block and Angle Update do not benefit from an increased number of sweeps. The 6-qubit case is shown in Fig. 4 (b), where a similar behavior is observed. However, none of the methods converge to the optimal value of 1, reflecting the limitations of the shallow PQC ansatz.

We also verified whether Cone, Block, and Angle Update can find ground state energies via imaginary time evolution. We chose the 1d

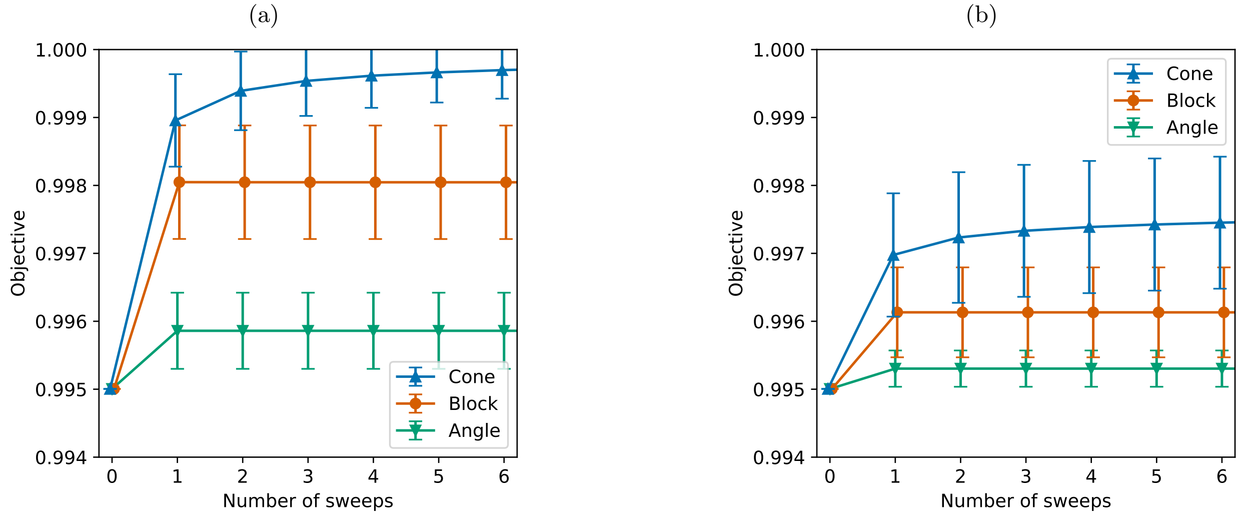

FIG. 4. Mean objective function and standard deviation of 25 random initializations for real time evolution with $\tau = 0.1$. (a) The Trotter term is located in front of a block in our PQC ansatz. Its causal cone encloses 4 qubits and 3 blocks, for a total of 45 parameters. (b) The Trotter term is located in front of two blocks in our PQC ansatz, which is a less favorable position. Its causal cone encloses 6 qubits and 5 blocks, for a total of 75 parameters.

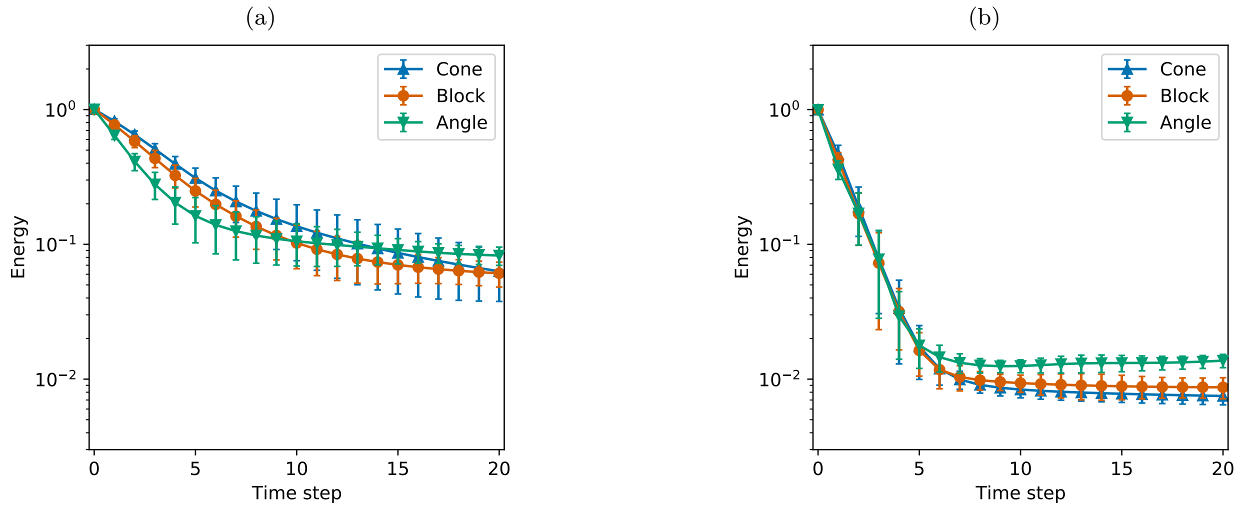

FIG. 5. Mean energy and standard deviation of 8 random initializations for imaginary time evolution on the 1d quantum Ising model. The energy is relative to the ground state, thus 0 energy represents the ideal solution to this problem. Here the time step is a hyper-parameter. We chose $\tau = 0.1$ for Cone Update, $\tau/N_b = 0.1$ for Block Update, $\tau/(N_b N_p) = 0.1$ for Angle Update. (a) The transverse field is set to $h = 1$ where the corresponding infinite system is critical. (b) The transverse field is set to $h = 4$ where the corresponding infinite system is non-critical.

quantum Ising model Hamiltonian for 8 qubits $\hat{H} = J\sum_{j=1}^{7} \hat{\sigma}_j^x \hat{\sigma}_{j+1}^x + h\sum_{j=1}^{8} \hat{\sigma}_j^z$. We also chose the PQC ansatz in Fig. 2 (a) with depth $\mathcal{D} = 2$ and open boundary conditions. In this experiment, the time step τ plays the role of an hyper-parameter since we are not interested in accurately simulating the evolution, but rather in finding the ground state as quickly as possible.

Figure 5 (a) shows the mean energy and standard deviation obtained in 20 time steps for $J = h = 1$ in the Hamiltonian, i.e., for the critical point of the corresponding infinite system. For our choices of hyper-parameter τ, all methods reached similar low energies. Figure 5 (b) shows the results for $J = 1$ and $h = 4$, i.e., where the corresponding infinite system is far from the critical point. All methods converged rapidly, producing states that are very close to the ground state. We emphasize that the circuits for Angle Update are much simpler than those for Cone Update.

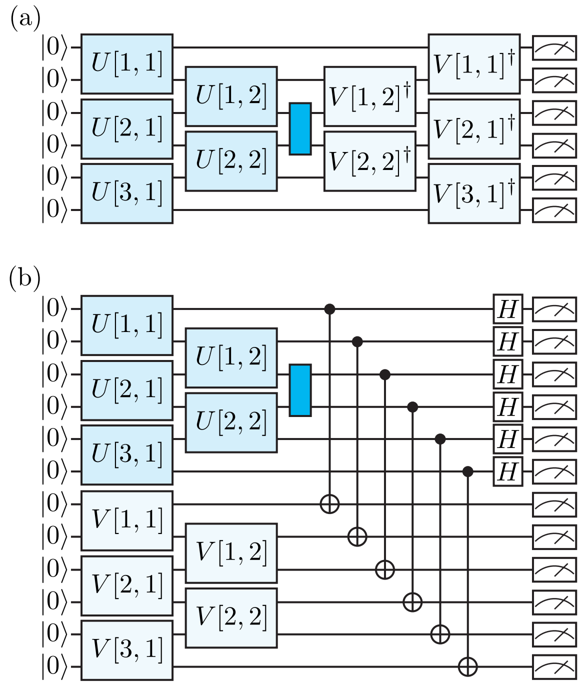

FIG. 6. A hardware-efficient Cone Update is obtained when the objective function is the state overlap $\left|\langle\psi_{k-1}|\, e^{i\tau h_k \hat{H}_k} \,|\psi(\boldsymbol{\theta})\rangle\right|^2 = \left|\langle\mathbf{0}|\, V^\dagger e^{i\tau h_k \hat{H}_k} U(\boldsymbol{\theta})\, |\mathbf{0}\rangle\right|^2$, where the V unitaries are fixed and the U unitaries are varied. (a) We maximize the probability of measuring $|\mathbf{0}\rangle$, which is equivalent to the state overlap. (b) The state overlap can alternatively be computed using this shorter-depth circuit representing the destructive Swap test [45, 46].

IV. DISCUSSION

Variational simulations of time evolution can be performed without inverting a possibly ill-conditioned matrix at each time step. To this end, we derived suitable algorithms whose hardware requirements can be adjusted to match the experimental capabilities.

One of the main applications of imaginary time evolution is to find ground states. Our most hardware-efficient algorithm, Angle Update, performed remarkably well at this task. In practice, once Angle Update converges one could switch to more demanding algorithms, such as Block or Cone Update, in order to fine-tune the result.

For real time evolution, the task is to simulate the time-dependent Schrödinger equation. We presented numerical evidence that Block and Cone Update achieve the high accuracy required. We also expect our Angle Update to be useful for specific applications, such as the computation of steady states, where accuracy per time step is not crucial.

A recent publication [21] contains simulations of real time evolution based on the maximization of the state overlap. As illustrated in Fig. 6, combining this method with our cone strategy leads to a promising hardware-efficient algorithm that can additionally make use of hardware-efficient overlap computation [45, 46]. Here the Rotosolve equations are already known [35] as the overlap maximization is equivalent to the expectation minimization for the Hermitian operator $M = -|\mathbf{0}\rangle\langle\mathbf{0}|$. Therefore this choice of objective function gives rise to a hardware-efficient Cone Update.

A number of recent articles implement tensor network techniques via parameterized quantum circuits (PQCs) (see Refs. [20, 39, 47] for examples). Our work contributes to this line of research, bringing PQC and tensor network optimization closer together.

The last key aspect of this work is the use of causal cones to enable simulations of finite systems larger than the size of the underlying quantum hardware. We envision that such techniques that detach the logical model from some of the hardware limits will ultimately enable us to attack large problems and obtain a quantum advantage.

V. ACKNOWLEDGEMENTS

We thank Nathan Fitzpatrick, David Amaro, and Carlo Modica for helpful discussions.

Appendix A: Variational simulation of time evolution

In this Appendix, we detail our method for the variational simulation of time evolution. To keep the discussion general, we consider an arbitrary complex time $z \in \mathbb{C}$. Later we will specialize to purely real time and purely imaginary time evolution.

We want to simulate the time evolution operator $e^{-iz\hat{H}}$ applied to an initial state $|\psi\rangle$ of n qubits. We assume that the Hamiltonian is given in general form $\hat{H} = \sum_{k=1}^{K} h_k \hat{H}_k$, where $\hat{H}_k \in \{\mathbb{1}, \hat{Z}, \hat{X}, \hat{Y}\}^{\otimes n}$ is a tensor product of Pauli operators, h_k is a real number, and $K \sim O(\text{poly}(n))$. There can be terms $\hat{H}_k$ in $\hat{H}$ that do not commute with each other.

A commonly used technique to simplify the problem consists of expanding the time evolution operator into a product. A product formula, such as the Trotter formula, produces a sequence of short-time evolutions which approximates the full evolution. Let us apply the product formula for $N \sim O(\mathrm{poly}(n))$ time steps of size $\zeta = z/N$:

$$\begin{aligned} e^{-iz\hat{H}} |\psi\rangle &\approx \left(e^{-i\zeta h_K \hat{H}_K} \dots e^{-i\zeta h_1 \hat{H}_1} \right)^N |\psi\rangle \\ &= e^{-i\zeta h_K \hat{H}_{K,N}} \dots e^{-i\zeta h_1 \hat{H}_{1,N}} \dots e^{-i\zeta h_K \hat{H}_{K,1}} \dots e^{-i\zeta h_1 \hat{H}_{1,1}} |\psi\rangle \, . \end{aligned} \tag{A1}$$

In the second line, $\hat{H}_{k,m}$ has an additional subscript indicating the m-th application of the k-th term.

Now assume we were able to obtain a variational approximation to the first operation:

$$e^{-i\zeta h_1 \hat{H}_{1,1}} |\psi\rangle \approx \left| \psi(\boldsymbol{\theta}^*_{1,1}) \right\rangle , \tag{A2}$$

where $\boldsymbol{\theta}_{1,1}$ is the vector of variational parameters and $\boldsymbol{\theta}^*_{1,1}$ indicates their optimal value. Then we can substitute this in Eq. (A1) and proceed with a variational approximation to the second operation:

$$e^{-i\zeta h_2 \hat{H}_{2,1}} e^{-i\zeta h_1 \hat{H}_{1,1}} |\psi\rangle \approx e^{-i\zeta h_2 \hat{H}_{2,1}} \left| \psi(\boldsymbol{\theta}^*_{1,1}) \right\rangle \approx \left| \psi(\boldsymbol{\theta}^*_{2,1}) \right\rangle . \tag{A3}$$

where again we assumed the optimal value for the variational parameters. Iterating the above procedure for all the NK terms we obtain an approximation to the full time evolution $e^{-iz\hat{H}} |\psi\rangle \approx \left| \psi(\boldsymbol{\theta}^*_{K,N}) \right\rangle$. To simplify the notation let us condense indexes k and n into a single index l, and let us write $|\psi(\boldsymbol{\theta}^*_l)\rangle = |\psi_l\rangle$ whenever the parameters have been optimized.

In order to find the variational approximation at step l we define the following cost function:

$$\begin{aligned} \mathcal{C}_l(\boldsymbol{\theta}_l) &= \left\| |\psi(\boldsymbol{\theta}_l)\rangle - e^{-i\zeta h_l \hat{H}_l} |\psi_{l-1}\rangle \right\|^2 \\ &= \langle \psi(\boldsymbol{\theta}_l) | \psi(\boldsymbol{\theta}_l) \rangle + \langle \psi_{l-1} | e^{i(\bar{\zeta}-\zeta) h_l \hat{H}_l} |\psi_{l-1}\rangle - \langle \psi(\boldsymbol{\theta}_l) | e^{-i\zeta h_l \hat{H}_l} |\psi_{l-1}\rangle - \langle \psi_{l-1} | e^{i\bar{\zeta} h_l \hat{H}_l} |\psi(\boldsymbol{\theta}_l)\rangle \\ &= \mathrm{const.} - 2\,\mathrm{Re}\left(\langle \psi_{l-1} | e^{i\bar{\zeta} h_l \hat{H}_l} |\psi(\boldsymbol{\theta}_l)\rangle \right) . \end{aligned} \tag{A4}$$

Here $\mathrm{Re}(\cdot)$ denotes the real part of a complex number, and $\bar{\zeta}$ denotes the complex conjugate of ζ. We have assumed an ansatz for which $\langle \psi(\boldsymbol{\theta}_l) | \psi(\boldsymbol{\theta}_l) \rangle$ is constant. This is the case, for example, if the ansatz is implemented by a PQC. We have also used that $\hat{H}_l$ is Hermitian.

The minimization of $\mathcal{C}_l(\boldsymbol{\theta}_l)$ is equivalent to the maximization of the following objective function:

$$\mathcal{F}_l(\boldsymbol{\theta}_l) = \mathrm{Re}\left(\langle \psi_{l-1} | e^{i\bar{\zeta} h_l \hat{H}_l} |\psi(\boldsymbol{\theta}_l)\rangle \right) . \tag{A5}$$

Let us now specialize to the two cases of interest. For real time evolution we write $z \equiv t$ where $t \in \mathbb{R}$ is the total time, and $\zeta \equiv \tau = t/N$ is the time step. Since the terms $\hat{H}_l$ are tensor products of Pauli operators, we have that $\hat{H}_l^2 = \mathbb{1}$. Using this property and the definition of matrix exponential $e^A = \sum_{n=0}^{\infty} A^n/n!$, it can be verified that $e^{i\tau h_l H_l} = \cos(\tau h_l)\mathbb{1} + i \sin(\tau h_l)\hat{H}_l$. Plugging this in the objective function we obtain:

$$\mathcal{F}_{l,\mathrm{real}}(\boldsymbol{\theta}_l) = \cos(\tau h_l)\,\mathrm{Re}\left(\langle \psi_{l-1} | \psi(\boldsymbol{\theta}_l) \rangle \right) - \sin(\tau h_l)\,\mathrm{Im}\left(\langle \psi_{l-1} | \hat{H}_l |\psi(\boldsymbol{\theta}_l)\rangle \right) . \tag{A6}$$

For imaginary time we write $z \equiv -it$ where $t \in \mathbb{R}$ is the total time, and $\zeta \equiv -i\tau = -it/N$ is the time step. Following the same argument above, it can be verified that $e^{-\tau h_l H_l} = \cosh(\tau h_l)\mathbb{1} - \sinh(\tau h_l)\hat{H}_l$. Plugging this in the objective function we obtain:

$$\mathcal{F}_{l,\mathrm{imag}}(\boldsymbol{\theta}_l) = \cosh(\tau h_l)\,\mathrm{Re}\left(\langle \psi_{l-1} | \psi(\boldsymbol{\theta}_l) \rangle \right) - \sinh(\tau h_l)\,\mathrm{Re}\left(\langle \psi_{l-1} | \hat{H}_l |\psi(\boldsymbol{\theta}_l)\rangle \right) . \tag{A7}$$

In Appendix B, we show that the coordinate-wise version of Eq. (A5) has a sinusoidal form. This fact is inherited by Eqs. (A6) and (A7), and is exploited to design the optimization algorithm in Appendix C. In Appendix D we present quantum circuits for the estimation of the objectives.

Appendix B: Sinusoidal form of the coordinate-wise objective function

In Ref. [35], the authors showed that for certain standard parameterized quantum circuits (PQCs) the expectation $\mathrm{tr}\big(MU\rho U^\dagger\big)$ as a function of a single parameter has sinusoidal form. This yields an efficient coordinate-wise optimization algorithm called Rotosolve that does not require explicit computation of the gradient or the Hessian. Here we present a similar derivation for objective functions of the form $\mathrm{Re}(\mathrm{tr}\big(MU\rho V^\dagger\big))$ where U is a parameterized quantum circuit and V is a fixed circuit. Note that there is nothing preventing us from parameterizing V and carrying out the same derivation.

Let us consider a PQC of the form $U(\boldsymbol{\theta}) = U_D \cdots U_1$ where each gate is either fixed, e.g., a CNOT, or parameterized as $U_d = \exp(-i\theta_d G_d)$, where $\theta_d \in (-\pi, \pi]$ and G_d is a Hermitian and unitary matrix such that $G_d^2 = \mathbb{1}$. For example, tensor products of Pauli matrices are suitable choices for $G_d \in \{\mathbb{1}, \hat{X}, \hat{Y}, \hat{Z}\}^{\otimes n}$. For the parameterized gates we use the definition of matrix exponential to get $U_d = \cos(\theta_d)\mathbb{1} - i\sin(\theta_d)G_d$. Without loss of generality, let us consider a pure initial state $\rho = |\mathbf{0}\rangle\langle\mathbf{0}|$ where $|\mathbf{0}\rangle \equiv |0\rangle^{\otimes n}$.

Expanding the objective function we have $\mathrm{Re}(\langle\mathbf{0}|\, V^\dagger M U_D \cdots U_d \cdots U_1 \,|\mathbf{0}\rangle)$. To express this as a function of a single parameter θ_d we simplify the notation absorbing all gates before U_d in a unitary which we call U_B, and we absorb all gates after U_d in a unitary which we call U_A. Using this notation we write:

$$\begin{aligned} f_d(x) &= \mathrm{Re}\left(\langle\mathbf{0}|\, V^\dagger M U_A U_d U_B \,|\mathbf{0}\rangle\right) \\ &= \mathrm{Re}\left(\langle\mathbf{0}|\, V^\dagger M U_A \left(\cos(x) I - i\sin(x) G_d\right) U_B \,|\mathbf{0}\rangle\right) \\ &= \mathrm{Re}\left(\langle\mathbf{0}|\, V^\dagger M U_A U_B \,|\mathbf{0}\rangle\right)\cos(x) + \mathrm{Re}\left(\langle\mathbf{0}|\, V^\dagger M U_A (-iG_d) U_B \,|\mathbf{0}\rangle\right)\sin(x). \end{aligned} \tag{B1}$$

Noting that $U_d(0) = I$ and $U_d(\frac{\pi}{2}) = -iG_d$, we rewrite the above as:

$$\begin{aligned} f_d(x) &= \mathrm{Re}\left(\langle\mathbf{0}|\, V^\dagger M U_A U_d(0) U_B \,|\mathbf{0}\rangle\right)\cos(x) + \mathrm{Re}\left(\langle\mathbf{0}|\, V^\dagger M U_A U_d(\tfrac{\pi}{2}) U_B \,|\mathbf{0}\rangle\right)\sin(x) \\ &= f(0)\cos(x) + f(\tfrac{\pi}{2})\sin(x). \end{aligned} \tag{B2}$$

We now use the harmonic addition theorem $a\cos(x) + b\sin(x) = \sqrt{a^2+b^2}\sin\big(x + \arctan\big(\frac{a}{b}\big)\big)$ to obtain:

$$\begin{aligned} f_d(x) &= A\sin(x + B), \\ A &= \sqrt{f_d(0)^2 + f_d(\tfrac{\pi}{2})^2}, \\ B &= \mathrm{arctan2}\left(f_d(0), f_d(\tfrac{\pi}{2})\right). \end{aligned} \tag{B3}$$

The objective function as a function of a single parameter has sinusoidal form with amplitude A, phase B, and period 2π. Note that we must use the arctan2 function in order to correctly handle the sign of numerator and denominator, as well as the case where the denominator is zero.

There is nothing special about the evaluations at 0 and $\frac{\pi}{2}$ in Eq. (B3). Indeed we can estimate A and B from any pair of points that are $\frac{\pi}{2}$ apart:

$$\sqrt{f_d(\phi)^2 + f_d(\phi+\tfrac{\pi}{2})^2} = \sqrt{A^2\sin^2(\phi+B) + A^2\sin^2(\phi+\tfrac{\pi}{2}+B)} = |A|\sqrt{\sin^2(\phi+B)+\cos^2(\phi+B)} = A, \tag{B4}$$

$$\frac{f_d(\phi)}{f_d(\phi+\frac{\pi}{2})} = \frac{\sin(\phi+B)}{\sin\big(\phi+\frac{\pi}{2}+B\big)} = \tan(\phi+B). \tag{B5}$$

Now, from the graph of the sine function, it is easy to locate the maxima at $\theta_d^* = \frac{\pi}{2} - B + 2\pi k$ for all $k \in \mathbb{Z}$. Using the above expression for B we obtain

$$\begin{aligned} \theta_d^* &= \arg\max_x f_d(x) \\ &= \tfrac{\pi}{2} - \mathrm{arctan2}\left(f_d(\phi), f_d(\phi+\tfrac{\pi}{2})\right) + \phi + 2\pi k, \end{aligned} \tag{B6}$$

where $\phi \in (-\pi, \pi]$, and we choose k such that $\theta_d^* \in (-\pi, \pi]$.

A similar derivation can be done for objective functions of the form $\mathrm{Im}(\mathrm{tr}\big(MU\rho V^\dagger\big))$. For the d-th parameter, we write $f_d(x) = \mathrm{Im}(\langle\mathbf{0}|\, V^\dagger M U_A U_d U_B \,|\mathbf{0}\rangle)$ and we obtain the maxima exactly as in Eq. (B6). Figure 7 shows the sinusoidal forms for a random choice of U, V and M on $n = 4$ qubits.

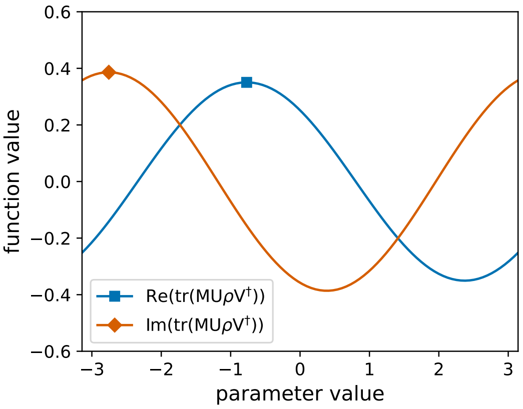

FIG. 7. Sinusoidal form of the objective functions used in this work. This example is for a random choice of U, V, M, and ρ on $n = 4$ qubits. The maxima (square and diamond) can be found in closed form. This requires the evaluation of the objective function at two arbitrary parameter values spaced $\frac{\pi}{2}$ apart.

Appendix C: The update rule with a single evaluation

In this Appendix, we take a deeper look at the quantity of interest, Eq. A5, and derive the update rule for the Rotosolve optimizer [35]. Consider the l-th term in the Trotter product formula and let $\boldsymbol{\theta}$ be the current value of the parameter vector. Then for the d-th parameter we have:

$$\begin{aligned} f_{l,d}(x) &= \mathcal{F}_l(\theta_1, \cdots, \theta_{d-1}, x, \theta_{d+1}, \cdots, \theta_D) \\ &= \mathrm{Re}\left(\langle \mathbf{0}|\, V^\dagger e^{i\bar{\zeta} h_l \hat{H}_l} U_D \cdots U_d(x) \cdots U_1 \,|\mathbf{0}\rangle\right) \\ &= \underbrace{\sqrt{f_{l,d}(\phi)^2 + f_{l,d}(\phi + \tfrac{\pi}{2})^2}}_{A_{l,d}} \sin\left(x + \underbrace{\arctan\left(\frac{f_{l,d}(\phi)}{f_{l,d}(\phi+\frac{\pi}{2})}\right) - \phi}_{B_{l,d}}\right), \end{aligned} \tag{C1}$$

where $A_{l,d}$ is the amplitude, $B_{l,d}$ is the phase, and $\phi \in (-\pi, \pi]$ can be chosen at will. The third line is obtained by applying the result from Appendix B.

The estimation of $B_{l,d}$ requires the evaluation of the objective at two points spaced $\frac{\pi}{2}$ apart. However, we can recycle information from previous steps in order to require a single evaluation. The approach is as follows. Say we have found the maximum θ^*_{d-1} for the $(d-1)$-th parameter. At no additional cost we calculate $f_{l,d-1}(\theta^*_{d-1}) = A_{l,d-1}$. Now we move to the d-th parameter. Setting ϕ in Eq. (C1) to the current parameter value, $\phi = \theta_d$, we happen to know that $f_{l,d}(\theta_d) = f_{l,d-1}(\theta^*_{d-1}) = A_{l,d-1}$. Hence, we only need to evaluate $f_{l,d}(\phi + \frac{\pi}{2})$.

In summary, we obtain the update rule:

$$\begin{aligned} \theta^*_d &= \arg\max_x f_{l,d}(x) \\ &= \frac{\pi}{2} - \arctan\left(\frac{f_{l,d}(\theta_d)}{f_{l,d}(\theta_d + \frac{\pi}{2})}\right) + \theta_d, \end{aligned} \tag{C2}$$

where θ_d is the current parameter value, and $f_{l,d}(\theta_d)$ is known from the previous parameter update. With this equation we can use Rotosolve for the variational simulation of time evolution.

Appendix D: Hadamard test for Cone Update

In this Appendix, we discuss the Hadamard test used in Cone Update. Recall that the proposed method initializes a PQC at each step and trains it to simulate the action of a short-time evolution operator on the previous variational state. Let us denote the state obtained at the $(l-1)$-th step as $|\psi_{l-1}\rangle = V|\mathbf{0}\rangle$ and the state for the l-th step as $|\psi(\boldsymbol{\theta}_l)\rangle = U|\mathbf{0}\rangle$. Here V and U are PQCs acting on the easy-to-prepare state $|\mathbf{0}\rangle \equiv |0\rangle^{\otimes n}$.

The Hadamard test can be challenging to execute on existing hardware when U and V are unrelated quantum circuits due to the potentially large number of controlled operation. Cone Update simplifies the test, exploiting the fact that U and V differ only locally. One circuit can be efficiently transformed into the other using adjoints of gates. To show this we start from $U|\mathbf{0}\rangle$, add one ancilla qubit in the state $|0\rangle$ which is then acted upon by a Hadamard gate, so that we get $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes U|\mathbf{0}\rangle$. Now we include the local transformations from U to V as gates controlled by the ancilla qubit. As an example, if U contains a rotation gate $R_z(a)$ and V contains $R_z(b)$ at the same location, then we only need to attach a controlled-$R_z(b-a)$ rotation. The subspaces where the ancilla is $|0\rangle$ and $|1\rangle$ will contain $U|\mathbf{0}\rangle$ and $V|\mathbf{0}\rangle$, respectively. In other words, the result is $\frac{1}{\sqrt{2}}(|0\rangle \otimes U|\mathbf{0}\rangle + |1\rangle \otimes V|\mathbf{0}\rangle)$. Having another Hadamard gate acting upon the ancilla qubit gives $\frac{1}{2}\left(|0\rangle \otimes (U+V)|\mathbf{0}\rangle + |1\rangle \otimes (U-V)|\mathbf{0}\rangle\right)$. Measuring the expectation of say $\hat{Z} \otimes \hat{H}_l$ one obtains the real part $\langle \hat{Z} \otimes \hat{H}_l \rangle = \mathrm{Re}(\langle \mathbf{0}| V^\dagger \hat{H}_l U |\mathbf{0}\rangle)$.

For the imaginary part, we follow the same procedure, but we include a phase gate after the first Hadamard gate. This yields $\langle \hat{Z} \otimes \hat{H}_l \rangle = \mathrm{Im}(\langle \mathbf{0}| V^\dagger \hat{H}_l U |\mathbf{0}\rangle)$. With the two circuits just described we can estimate the objective function in Eq. (A6) for real time evolution, and in Eq. (A7) for imaginary time evolution. If a causal cone contains N_b blocks, each with N_p parameterized gates, the Hadamard test requires $O(N_b N_p)$ controlled operations.

Appendix E: The hardware-efficient Angle Update

In this Appendix, we present the implementation of our method that has the lowest hardware requirements. Angle Update is efficient in terms of circuit depth, but still uses an ancilla qubit and a controlled operation which may be challenging to realize on existing hardware. We can avoid the use of those while requiring the execution of additional circuits. This approach uses the methods presented in Refs. [42] and [43] to replace indirect measurements with direct ones.

Recall that Angle Update consists of replacing the variational state $|\psi_{l-1}\rangle = V|\mathbf{0}\rangle$ in the objective function with the variational state $|\psi(\boldsymbol{\theta})\rangle = U|\mathbf{0}\rangle$ after each parameter update. This approximation guarantees that the two PQCs V and U differ by just one parameter at all times. Thus, we can drop V and express everything in terms of U. Let us consider the l-th term of the Trotter product formula. The coordinate-wise objective for the d-th parameter is:

$$f_{l,d}(x) = \mathrm{Re}\left(\langle \mathbf{0}| U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger e^{i\bar{\zeta} h_l \hat{H}_l} U_D \cdots U_d(x) \cdots U_1 |\mathbf{0}\rangle\right). \tag{E1}$$

Here the variable x appears only in the gate denoted by $U_d(x)$ and, with a slight abuse of notation, U_d is fixed and uses the current parameter value θ_d.

In the following, it is useful to consider the two cases of real and imaginary time evolution separately. For real time ($\zeta \equiv \tau \in \mathbb{R}$) the coordinate-wise version of Eq. (A6) is:

$$\begin{aligned} f_{l,d,\mathrm{real}}(x) = &\cos(\tau h_l)\,\mathrm{Re}\left(\langle \mathbf{0}| U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger \mathbb{1} U_D \cdots U_d(x) \cdots U_1 |\mathbf{0}\rangle\right) - \\ &\sin(\tau h_l)\,\mathrm{Im}\left(\langle \mathbf{0}| U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger \hat{H}_l U_D \cdots U_d(x) \cdots U_1 |\mathbf{0}\rangle\right). \end{aligned} \tag{E2}$$

Recall from Eq. (C1) that this function is of sinusoidal form and is maximized by $x^* = \frac{\pi}{2} - B_{l,d}$, Also recall that $B_{l,d}$ is estimated from two evaluation of the objective carried out at any two points spaced $\frac{\pi}{2}$ apart. For the first evaluation, we choose the current parameter value obtaining $f_{l,d,\mathrm{real}}(\theta_d) = \cos(\tau h_l)$. The second evaluation is slightly more involved. Using $U_d(\theta_d + \frac{\pi}{2}) = U_d(\theta_d) U_d(\frac{\pi}{2}) = -iU_d(\theta_d) G_d$, we have that:

$$f_{l,d,\mathrm{real}}(\theta_d + \tfrac{\pi}{2}) = -\sin(\tau h_l)\,\mathrm{Im}\left(\langle \mathbf{0}| U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger \hat{H}_l U_D \cdots U_d(-iG_d) \cdots U_1 |\mathbf{0}\rangle\right). \tag{E3}$$

Now we can use the technique from Ref. [43] to write the above as the difference of two expectations. Since G_d is a Pauli operator, we can define projective measurement operators $\frac{1}{\sqrt{2}}(\mathbb{1} \pm G_d)$. In turn these can be used to define new observables $\hat{H}_{l\pm} = \frac{1}{2}(\mathbb{1} \pm G_d)U_d^\dagger \cdots U_D^\dagger \hat{H}_l U_D \cdots U_d(\mathbb{1} \pm G_d)$ which are easily verified to be Hermitian. With these, a direct calculation shows that:

$$f_{l,d,\text{real}}(\theta + \tfrac{\pi}{2}) = -\frac{\sin(\tau h_l)}{2}\left(\langle \hat{H}_{l+}\rangle - \langle \hat{H}_{l-}\rangle\right). \tag{E4}$$

In practice, the projective measurement operators $\frac{1}{\sqrt{2}}(\mathbb{1} \pm G_d)$ correspond to measuring the qubit when these operators act in the eigenbasis of G_d [43]. Putting everything together, Eq. (E2) is maximized in closed form using two expectations:

$$\begin{aligned} \theta_d^* &= \arg\max_x f_{l,d,\text{real}}(x) \\ &= \frac{\pi}{2} - \arctan\left(\frac{2\cot(\tau h_l)}{\langle \hat{H}_{l-}\rangle - \langle \hat{H}_{l+}\rangle}\right) + \theta_d. \end{aligned} \tag{E5}$$

Clearly, Angle Update is much more hardware-efficient than Cone Update for real time evolution. All quantities are estimated by direct measurements, without using ancilla qubits or controlled gates.

Let us now consider imaginary time evolution. For imaginary time ($\zeta \equiv -i\tau$ with $\tau \in \mathbb{R}$) the coordinate-wise version of Eq. (A7) is:

$$\begin{aligned} f_{l,d,\text{imag}}(x) = {} & \cosh(\tau h_l)\,\text{Re}\left(\langle \mathbf{0}|\, U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger \mathbb{1} U_D \cdots U_d(x) \cdots U_1 \,|\mathbf{0}\rangle\right) - \\ & \sinh(\tau h_l)\,\text{Re}\left(\langle \mathbf{0}|\, U_1^\dagger \cdots U_d^\dagger \cdots U_D^\dagger \hat{H}_l U_D \cdots U_d(x) \cdots U_1 \,|\mathbf{0}\rangle\right). \end{aligned} \tag{E6}$$

Following the same argument above, we need to evaluate the objective at two points spaced $\frac{\pi}{2}$ apart in order to estimate $B_{l,d}$ and the maximum. Again we perform the first evaluation at the current parameter value obtaining $f_{l,d,\text{imag}}(\theta_d) = \cosh(\tau h_l) - \sinh(\tau h_l)\,\langle \hat{H}\rangle_{\theta_d}$. The subscript is used to stress that the expectation is computed using the current parameter value. For the second evaluation, we use the method introduced in Ref. [42] to obtain the difference of two expectations:

$$f_{l,d,\text{imag}}(\theta_d + \tfrac{\pi}{2}) = -\frac{\sinh(\tau h_l)}{2}\left(\langle \hat{H}_l\rangle_{\theta_d + \frac{\pi}{4}} - \langle \hat{H}_l\rangle_{\theta_d - \frac{\pi}{4}}\right). \tag{E7}$$

Here the subscripts are used to stress the use of a shifted value for the d-the parameter.

Plugging these expressions in the equation for the optimum we obtain:

$$\begin{aligned} \theta_d^* &= \arg\max_x f_{l,d,\text{imag}}(x) \\ &= \frac{\pi}{2} - \arctan\left(\frac{2\coth(\tau h_l) - 2\,\langle \hat{H}_l\rangle_{\theta_d}}{\langle \hat{H}_l\rangle_{\theta_d - \frac{\pi}{4}} - \langle \hat{H}_l\rangle_{\theta_d + \frac{\pi}{4}}}\right) + \theta_d. \end{aligned} \tag{E8}$$

In other words, Eq. (E6) can be maximized in closed form using the expectation of $\hat{H}_l$ for three slightly different circuits. Angle Update is therefore hardware-efficient for imaginary time evolution. All quantities are estimated by direct measurements, without using ancilla qubits or controlled gates.

In Cone Update we were able to recycle information from previous steps and reduce the circuit count. In Angle Update this cannot be done exactly as the objective function changes after each parameter update. For imaginary time, assuming the change in objective function value is small, we can approximately recycle information from the previous steps as $f_{l,d,\text{imag}}(\theta_d) \approx f_{l,d-1,\text{imag}}(\theta_{d-1}^*)$, bringing the total number of circuits to two. For small values of τ, the error of this approximation is proportional to the modification $\delta\theta_{d-1}$ of the previous parameter. A smaller value of τ produces a smaller value of $\delta\theta_{d-1}$, hence the error of this approximation can be systematically reduced by decreasing τ.

Appendix F: Time-dependent variational principle and matrix inversion

The time-dependent variational principle (TDVP) can be derived in the following way. Our goal is to time-evolve an ansatz $|\psi(\boldsymbol{\theta})\rangle$ via the Schrödinger equation:

$$i\frac{\mathrm{d}}{\mathrm{d}t}\,|\psi(\boldsymbol{\theta})\rangle = \hat{H}\,|\psi(\boldsymbol{\theta})\rangle\,, \tag{F1}$$

where we have set $\hbar = 1$. The right-hand side of equation (F1) may leave the variational space of $|\psi(\boldsymbol{\theta})\rangle$ which is created by all possible choices for $\boldsymbol{\theta}$. To stay in the variational space we minimize the distance:

$$\begin{aligned}\mathrm{dist}_{\mathrm{TDVP}} &= \left\| i\frac{\mathrm{d}}{\mathrm{d}t} |\psi(\boldsymbol{\theta})\rangle - \hat{H}|\psi(\boldsymbol{\theta})\rangle \right\|^2 \\ &= \left(\frac{\mathrm{d}}{\mathrm{d}t}\langle\psi(\boldsymbol{\theta})|\right)\left(\frac{\mathrm{d}}{\mathrm{d}t}|\psi(\boldsymbol{\theta})\rangle\right) + i\left(\frac{\mathrm{d}}{\mathrm{d}t}\langle\psi(\boldsymbol{\theta})|\right)\hat{H}|\psi(\boldsymbol{\theta})\rangle - i\langle\psi(\boldsymbol{\theta})|\hat{H}\left(\frac{\mathrm{d}}{\mathrm{d}t}|\psi(\boldsymbol{\theta})\rangle\right) + \langle\psi(\boldsymbol{\theta})|\hat{H}^2|\psi(\boldsymbol{\theta})\rangle .\end{aligned} \tag{F2}$$

Using the chain rule:

$$\frac{\mathrm{d}}{\mathrm{d}t}|\psi(\boldsymbol{\theta})\rangle = \sum_k \frac{\partial|\psi(\boldsymbol{\theta})\rangle}{\partial\theta_k}\frac{\mathrm{d}\theta_k}{\mathrm{d}t}, \tag{F3}$$

and the definitions:

$$\begin{aligned} B_k &:= \frac{\mathrm{d}\theta_k}{\mathrm{d}t} \\ A_{j,k} &:= \left(\frac{\partial\langle\psi(\boldsymbol{\theta})|}{\partial\theta_j}\right)\left(\frac{\partial|\psi(\boldsymbol{\theta})\rangle}{\partial\theta_k}\right) \\ C_j &:= \left(\frac{\partial\langle\psi(\boldsymbol{\theta})|}{\partial\theta_j}\right)\hat{H}|\psi(\theta)\rangle , \end{aligned} \tag{F4}$$

we rewrite Eq. (F2) as:

$$\mathrm{dist}_{\mathrm{TDVP}} = \sum_{j,k}\bar{B}_j A_{j,k} B_k + i\sum_j \bar{B}_j C_j - i\sum_j \bar{C}_j B_j + \mathrm{const}. \tag{F5}$$

In a PQC ansatz the variational parameters are rotation angles. Thus, we now restrict θ_j to be real and obtain B_j that are also real. That is, $\bar{B}_j = B_j$ in the equation above. The minimum of this equation in terms of the B_j can be determined by taking the derivatives and equating them to zero:

$$\frac{\partial \mathrm{dist}_{\mathrm{TDVP}}}{\partial B_j} = \sum_k 2\,\mathrm{Re}(A_{j,k})B_k - 2\,\mathrm{Im}(C_j) = 0. \tag{F6}$$

This is equivalent to:

$$\sum_k \mathrm{Re}(A_{j,k})B_k = \mathrm{Im}(C_j). \tag{F7}$$

Notice that Eq. (F7) can be written as a matrix vector equation by defining $\mathrm{Re}(A_{j,k})$ to be the matrix elements inside a matrix A, and defining B_k to be the vector elements inside a vector $\boldsymbol{B}$, and defining $\mathrm{Im}(C_j)$ to be the vector elements inside a vector $\boldsymbol{C}$. This leads to the final TDVP equation:

$$A\boldsymbol{B} = \boldsymbol{C}. \tag{F8}$$

This is an equation for the time-dependence of the parameters in our PQC ansatz since $\boldsymbol{B} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\theta}$. Therefore, TDVP replaces the original Schrödinger equation (F1) with a new equation (F8) that time-evolves the variational parameters directly. Although we focused here on real time evolution, other applications such as imaginary time evolution lead to similar systems of linear equations [16].

To quantify the accuracy of our solution vector $\boldsymbol{B}$ in equation (F8) it is important to emphasize that the matrix A and the vector $\boldsymbol{C}$ are constructed from a finite number of measurements on a quantum computer and therefore have a finite precision [15, 16]. The TDVP algorithm determines the individual elements in A and $\boldsymbol{C}$ as mean values over m measurements of specific quantum circuits and this mean value computation has the error ϵ_{MC} of classical Monte Carlo sampling scaling like $O(1/\sqrt{m})$. Therefore, using a number of measurements m for each element in the matrix A and vector $\boldsymbol{C}$, both are accurate only up to an error

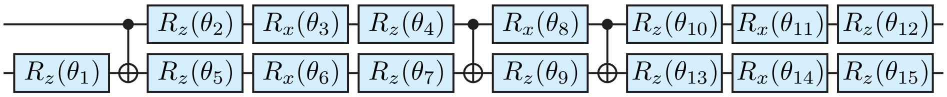

FIG. 8. The 2-qubit block used in the simulations. Each block has 3 CNOTs and $N_p = 15$ adjustable parameters consisting of angles of rotation about the canonical x and z axes.

scaling like $O(1/\sqrt{m})$. To analyze the effect of this error in A on the solution of equation (F8) we replace A by $A + \delta A$ where now A represents the exact A and δA its error. Similarly we replace $\boldsymbol{B}$ by $\boldsymbol{B} + \delta\boldsymbol{B}$:

$$\begin{aligned}(A+\delta A)(\boldsymbol{B}+\delta\boldsymbol{B}) &= \boldsymbol{C}\\ A\delta\boldsymbol{B} + \delta A\boldsymbol{B} &= 0\\ \delta\boldsymbol{B} &= -A^{-1}\delta A\boldsymbol{B}\\ \|\delta\boldsymbol{B}\| &\leq \left\|A^{-1}\right\|\|\delta A\|\|\boldsymbol{B}\|\\ \frac{\|\delta\boldsymbol{B}\|}{\|\boldsymbol{B}\|} &\leq \|A\|\left\|A^{-1}\right\|\frac{\|\delta A\|}{\|A\|},\end{aligned} \tag{F9}$$

where we have used $\boldsymbol{AB} = \boldsymbol{C}$, ignored $\delta A\delta\boldsymbol{B}$, and where $\|\cdot\|$ is a norm. We observe that $\|A\|\left\|A^{-1}\right\|$ is the condition number. Thus, $\kappa = \|A\|\left\|A^{-1}\right\| = \sigma_{\max}/\sigma_{\min}$ where $\sigma_{\max}$ denotes the largest and $\sigma_{\min}$ the smallest singular value of A. We also observe that $\|\delta A\|/\|A\|$ is the relative error of A which scales like $O(1/\sqrt{m})$. Therefore, we have obtained an upper bound for the relative accuracy of $\boldsymbol{B}$:

$$\epsilon_B^{\max} \propto \frac{\kappa}{\sqrt{m}}. \tag{F10}$$

Equivalently, Eq. (F10) states that for a desired accuracy ϵ in $\boldsymbol{B}$ we need the number of measurements m to scale like $O(\kappa^2/\epsilon^2)$. The same scaling is obtained for finite precision $\boldsymbol{C}$ by repeating the above calculation using $\boldsymbol{C} + \delta\boldsymbol{C}$. We conclude that the matrix inversion required in the original TDVP methods [15, 16] leads to a computational cost scaling like $O(\kappa^2/\epsilon^2)$ for accuracy ϵ. Compared with the standard measurement error $O(1/\epsilon^2)$ the additional factor of κ^2 can be computationally challenging especially for ill-conditioned matrices A.

Appendix G: Numerical simulations

For the numerical simulations we used the PQC ansatz shown in Fig. 2 (a), with depth $\mathcal{D} = 2$ and number of qubits $n \in \{4, 6, 8\}$ depending on the experiment. For expectations and Hadamard tests, we always used causal cones to reduce the size of the simulation. For 1-qubit and 2-qubit nearest-neighbor Trotter terms, the causal cones involve at most 6 qubits. This remains true if we include periodic boundary conditions in the ansatz, such as the $U[\frac{n}{2}, 2]$ block shown in Fig. 2 (a).

Each block in the ansatz was implemented using a 2-qubit minimal construction proposed in Ref. [48] and shown in Fig. 8. This construction has $N_p = 15$ parameters and is universal for 2-qubit unitaries up to a global phase. Note however that in all our experiments the phase was relevant since we used $n > 2$.

Expectations and Hadamard tests were calculated exactly. We did not include finite-sampling noise or hardware noise. All numerical simulations were programmed in Python using the QuTiP library [49].

[1] S. Wiesner, "Simulations of many-body quantum systems by a quantum computer," (1996), arXiv:quant-ph/9603028 [quant-ph].

[2] S. Lloyd, "Universal quantum simulators," Science **273**, 1073–1078 (1996).

[3] D. S. Abrams and S. Lloyd, "Simulation of many-body fermi systems on a universal quantum computer," Phys. Rev. Lett. **79**, 2586–2589 (1997).

[4] C. Zalka, "Simulating quantum systems on a quantum computer," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**, 313–322 (1998).

[5] I. Kassal, S. P. Jordan, P. J. Love, M. Mohseni, and A. Aspuru-Guzik, “Polynomial-time quantum algorithm for the simulation of chemical dynamics,” Proceedings of the National Academy of Sciences **105**, 18681–18686 (2008).

[6] I. Kassal, J. D. Whitfield, A. Perdomo-Ortiz, M.-H. Yung, and A. Aspuru-Guzik, “Simulating chemistry using quantum computers,” Annual Review of Physical Chemistry **62**, 185–207 (2011).

[7] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferov, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, “Quantum chemistry in the age of quantum computing,” Chemical Reviews **119**, 10856–10915 (2019).

[8] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, “Quantum computational chemistry,” Rev. Mod. Phys. **92**, 015003 (2020).

[9] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, “Quantum algorithms for quantum chemistry and quantum materials science,” (2020), arXiv:2001.03685 [quant-ph].

[10] A. Smith, M. Kim, F. Pollmann, and J. Knolle, “Simulating quantum many-body dynamics on a current digital quantum computer,” npj Quantum Information **5**, 106 (2019).

[11] E. Gustafson, P. Dreher, Z. Hang, and Y. Meurice, “Benchmarking quantum computers for real-time evolution of a $(1 + 1)$ field theory with error mitigation,” (2019), arXiv:1910.09478 [hep-lat].

[12] B. Fauseweh and J.-X. Zhu, “Digital quantum simulation of non-equilibrium quantum many-body systems,” (2020), arXiv:2009.07375 [quant-ph].

[13] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. OBrien, “A variational eigenvalue solver on a photonic quantum processor,” Nature Communications **5**, 4213 (2014).

[14] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, “The theory of variational hybrid quantum-classical algorithms,” New Journal of Physics **18**, 023023 (2016).

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

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

[17] S.-H. Lin, R. Dilip, A. G. Green, A. Smith, and F. Pollmann, “Real- and imaginary-time evolution with compressed quantum circuits,” (2020), arXiv:2008.10322 [quant-ph].

[18] C. Cirstoiu, Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, “Variational fast forwarding for quantum simulation beyond the coherence time,” npj Quantum Information **6**, 1–10 (2020).

[19] M. Lubasch, P. Moinier, and D. Jaksch, “Multigrid renormalization,” Journal of Computational Physics **372**, 587 – 602 (2018).

[20] M. Lubasch, J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, “Variational quantum algorithms for nonlinear problems,” Phys. Rev. A **101**, 010301 (2020).

[21] B. Jaderberg, A. Agarwal, K. Leonhardt, M. Kiffner, and D. Jaksch, “Minimum hardware requirements for hybrid quantum–classical DMFT,” Quantum Science and Technology **5**, 034015 (2020).

[22] I. Rungger, N. Fitzpatrick, H. Chen, C. H. Alderete, H. Apel, A. Cowtan, A. Patterson, D. M. Ramo, Y. Zhu, N. H. Nguyen, E. Grant, S. Chretien, L. Wossnig, N. M. Linke, and R. Duncan, “Dynamical mean field theory algorithm and experiment on quantum computers,” (2020), arXiv:1910.04735 [quant-ph].

[23] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, “Parameterized quantum circuits as machine learning models,” Quantum Science and Technology **4**, 043001 (2019).

[24] G. Vidal, “Efficient classical simulation of slightly entangled quantum computations,” Phys. Rev. Lett. **91**, 147902 (2003).

[25] G. Vidal, “Efficient simulation of one-dimensional quantum many-body systems,” Phys. Rev. Lett. **93**, 040502 (2004).

[26] A. J. Daley, C. Kollath, U. Schollwck, and G. Vidal, “Time-dependent density-matrix renormalization-group using adaptive effective hilbert spaces,” Journal of Statistical Mechanics: Theory and Experiment **2004**, P04005 (2004).

[27] G. Vidal, “Classical simulation of infinite-size quantum lattice systems in one spatial dimension,” Phys. Rev. Lett. **98**, 070201 (2007).

[28] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac, “Classical simulation of infinite-size quantum lattice systems in two spatial dimensions,” Phys. Rev. Lett. **101**, 250602 (2008).

[29] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, “Unifying projected entangled pair state contractions,” New Journal of Physics **16**, 033014 (2014).

[30] M. Lubasch, J. I. Cirac, and M.-C. Bañuls, “Algorithms for finite projected entangled pair states,” Phys. Rev. B **90**, 064425 (2014).

[31] G. Vidal, “Class of quantum many-body states that can be efficiently simulated,” Phys. Rev. Lett. **101**, 110501 (2008).

[32] G. Evenbly and G. Vidal, “Algorithms for entanglement renormalization,” Phys. Rev. B **79**, 144108 (2009).

[33] M. B. Hastings, “Light-cone matrix product,” Journal of Mathematical Physics **50**, 095207 (2009).

[34] F. Barratt, J. Dborin, M. Bal, V. Stojevic, F. Pollmann, and A. G. Green, “Parallel quantum simulation of

large systems on small quantum computers," (2020), arXiv:2003.12087 [quant-ph].

[35] M. Ostaszewski, E. Grant, and M. Benedetti, "Quantum circuit structure learning," (2019), arXiv:1905.09692 [quant-ph].

[36] F. Verstraete, V. Murg, and J. Cirac, "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems," Advances in Physics **57**, 143–224 (2008).

[37] R. Ors, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," Annals of Physics **349**, 117 – 158 (2014).

[38] N. Hatano and M. Suzuki, "Finding exponential product formulas of higher orders," in *Quantum Annealing and Other Optimization Methods* (Springer Berlin Heidelberg, 2005) pp. 37–68.

[39] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, "Towards quantum machine learning with tensor networks," Quantum Science and Technology **4**, 024001 (2019).

[40] J. Liu, L. Wan, Z. Li, and J. Yang, "Simulating periodic systems on quantum computer," (2020), arXiv:2008.02946 [quant-ph].

[41] D. Z. Manrique, I. T. Khan, K. Yamamoto, V. Wichitwechkarn, and D. M. Ramo, "Momentum-space unitary couple cluster and translational quantum subspace expansion for periodic systems on quantum computers," (2020), arXiv:2008.08694 [quant-ph].

[42] J. Li, X. Yang, X. Peng, and C.-P. Sun, "Hybrid quantum-classical approach to quantum optimal control," Phys. Rev. Lett. **118**, 150503 (2017).

[43] K. Mitarai and K. Fujii, "Methodology for replacing indirect measurements with direct measurements," Phys. Rev. Research **1**, 013006 (2019).

[44] G. H. Golub and C. F. V. Loan, *Matrix Computations* (1996) p. 3rd edition.

[45] J. C. Garcia-Escartin and P. Chamorro-Posada, "Swap test and hong-ou-mandel effect are equivalent," Phys. Rev. A **87**, 052330 (2013).

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

[47] E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. G. Green, and S. Severini, "Hierarchical quantum classifiers," npj Quantum Information **4**, 1–8 (2018).

[48] V. V. Shende, I. L. Markov, and S. S. Bullock, "Minimal universal two-qubit controlled-not-based circuits," Phys. Rev. A **69**, 062321 (2004).

[49] J. Johansson, P. Nation, and F. Nori, "Qutip: An open-source python framework for the dynamics of open quantum systems," Computer Physics Communications **183**, 1760 – 1772 (2012).