Overhead-constrained circuit knitting for variational quantum dynamics

Gian Gentinetta, Friederike Metz, and Giuseppe Carleo

Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
Center for Quantum Science and Engineering, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland

Simulating the dynamics of large quantum systems is a formidable yet vital pursuit for obtaining a deeper understanding of quantum mechanical phenomena. While quantum computers hold great promise for speeding up such simulations, their practical application remains hindered by limited scale and pervasive noise. In this work, we propose an approach that addresses these challenges by employing circuit knitting to partition a large quantum system into smaller subsystems that can each be simulated on a separate device. The evolution of the system is governed by the projected variational quantum dynamics (PVQD) algorithm, supplemented with constraints on the parameters of the variational quantum circuit, ensuring that the sampling overhead imposed by the circuit knitting scheme remains controllable. We test our method on quantum spin systems with multiple weakly entangled blocks each consisting of strongly correlated spins, where we are able to accurately simulate the dynamics while keeping the sampling overhead manageable. Further, we show that the same method can be used to reduce the circuit depth by cutting long-ranged gates.

1 Introduction

Quantum computers are promising tools for simulating quantum systems [1–6]. Particularly, the efficient simulation of quantum dynamics can provide insightful information about the nature of physical phenomena at the microscopic scale [7–12]. However, the practical utility of quantum devices is currently constrained by limitations in scale and the effects of noise [13–16]. While the size of available quantum computers is steadily growing [17], most publicly available devices are still very limited in size. In order to extend the capabilities of Noisy Intermediate-Scale Quantum (NISQ) devices [18], several schemes have been proposed to partition large systems into small clusters that can be solved individually on smaller quantum hardware [19–34]. To combine the solutions and recover the entanglement between the subsystems, classical resources are usually employed. Hence, ultimately, these hybrid quantum-classical computing approaches allow for quantum simulations on a larger scale.

Developing strategies for efficiently partitioning quantum computations is especially timely, as one of the focuses of the next generation of quantum processors lies in connecting mul-

Gian Gentinetta: gian.gentinetta@epfl.ch

tiple medium-size quantum chips, allowing for parallelization of quantum simulations with real-time classical communication [17]. This strategy is of particular utility if each subsystem that is simulated on a separate device is itself highly entangled and, hence, difficult to simulate classically. On the other hand, the entanglement between the partitions should be weak such that classical methods can be efficiently employed to recombine the subsystems. The idea of splitting a quantum system into subsystems can also be motivated by the underlying physical or chemical processes. Several interesting physical systems naturally allow for partitioning into weakly-entangled subsystems such as ground and low-energy eigenstates of local lattice Hamiltonians [35–37] and molecules [38], as well as quantum impurities immersed in a bath [39, 40].

Two prominent hybrid quantum-classical schemes that combine multiple quantum circuits using classical post-processing are entanglement forging [28–30] and circuit knitting [19–23]. Entanglement forging relies on the fact that a bipartite quantum state can always be written in the Schmidt decomposition. This enables a classical computer to combine the states of two systems implemented on separate quantum devices. If the two systems are weakly entangled with each other, a small number of Schmidt coefficients suffices for a good approximation of the full solution. Crucially, entanglement forging is limited to two subsystems, as the Schmidt decomposition cannot be applied to general multipartite states. Circuit knitting, on the other hand, employs quasi-probability distributions to cut gates that span across different systems into locally realizable quantum channels. This allows to arbitrarily cut a quantum circuit into multiple subsystems. However, this technique imposes a sampling overhead that scales exponentially in the number of gates cut.

In this work, we propose a method for quantum time evolution that splits a quantum circuit ansatz into multiple subsystems using circuit knitting while keeping the sampling overhead controlled. This is achieved by imposing a constraint on the circuit parameters during the optimization of the variational quantum circuit.

We employ this method to simulate the dynamics of quantum systems using the projected variational quantum dynamics (PVQD) algorithm [41]. While there have been implementations of quantum-classical hybrid schemes to quantum dynamics using perturbation theory [34] or by leveraging mean-field corrections and auxiliary qubits [31], an application to variational quantum dynamics is largely missing in the literature. The task is non-trivial, as evolving a parameterized quantum state in time either requires measuring (complex) matrix elements of the geometric tensor [42–45] or fidelities between quantum states [41, 46]. This poses a challenge to entanglement forging, where the ansatz is given by a superposition of quantum circuits. There, measuring overlaps is expensive and usually requires non-local circuits such as Hadamard-tests [47]. Instead, in the framework of circuit knitting, fidelities can be straightforwardly computed using, for example, the compute-uncompute method [48] without introducing any ancilla qubits or long-ranged gates.

We test our method on spin systems in a transverse field Ising model, where we weakly couple multiple blocks of strongly correlated spins. We show that with a realistic sampling overhead, we can significantly improve the accuracy of the simulation compared to a pure block product approximation, which does not consider any entanglement between different blocks. Furthermore, the trade-off between the sampling overhead and the accuracy of the variational state can be tuned in a controlled way via a single hyperparameter of the optimization. Finally, we demonstrate that our scheme can also reduce the required circuit depth when simulating models containing long-range interactions.

The structure of this paper is as follows: In Section 2, we explain how we use PVQD and circuit knitting techniques to evolve a quantum circuit ansatz in time while keeping the sampling overhead controlled. In Section 3, we test our method on quantum spin systems in a transverse field Ising model for different setups. Finally, in Section 4, we discuss the results and provide an outlook on possible future applications of the method.

2 Methods

We consider the dynamics of a quantum system represented by a Hilbert space partitioned into N individual subsystems (called blocks) $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_N$, where the blocks are simulated either in parallel on separate quantum devices or sequentially on the same machine. While the qubits within one block can be highly entangled, we impose that the entanglement between blocks is weak, such that it can be recovered efficiently using classical resources.

2.1 Projected variational quantum dynamics

We perform the dynamics of the system governed by a Hamiltonian H using the projected variational quantum dynamics (PVQD) algorithm [41]. While traditional trotterized time evolution requires circuits that grow in depth with increasing evolution time t, the advantage of variational algorithms such as PVQD is that the circuit depth remains constant over the whole evolution. PVQD evolves the parameters θ of a quantum circuit ansatz $|\psi(\theta)\rangle$ in time, by minimizing the infidelity

$$\theta_t = \arg\min_{\theta} \left[1 - \left| \langle \psi(\theta) | \, e^{-i\Delta t H} \, |\psi(\theta_{t-1})\rangle \right|^2\right] \tag{1}$$

at every time step t. This ensures that $|\psi(\theta_t)\rangle$ is the state within the manifold defined by the ansatz that is closest to the true time-evolved state $e^{-i\Delta t H} |\psi(\theta_{t-1})\rangle$. Here, the time evolution unitary $e^{-i\Delta t H}$ can be expanded into gates using the Trotter-Suzuki decomposition of the first order for which the introduced error scales as $\mathcal{O}(\Delta t^2)$. In our case, the time step Δt is chosen to be small to keep the error negligible.

Crucially, PVQD only requires measuring fidelities between two quantum states. This can be achieved by sampling from hardware efficient circuits, in contrast to other variational methods such as the time-dependent variational principle (TDVP) [42–45], where complex-valued state overlaps need to be measured using for example Hadamard-tests.

The fidelity between two quantum circuits is usually obtained using the compute-uncompute method [48], in which one measures the probability of retrieving the all-zero bit string after evolving the circuit in Eq. (1). The optimization of this global loss function is known to be prone to cost function-dependent barren plateaus [49], i.e. the gradients vanish exponentially fast in the number of qubits n. It has been shown that for small enough time steps Δt, PVQD is not affected by this problem as the initial guess $|\psi(\theta_{t-1})\rangle$ has a non-zero overlap with the target state $e^{-i\Delta t H} |\psi(\theta_{t-1})\rangle$ [41, 50]. In addition, in the following experiments, we further increase the variance of the gradient by measuring a local observable with the same maximum as the global fidelity. The observable is defined as averaging over the local $|0\rangle\langle 0|$ projectors

$$\mathcal{O}_{\text{loc}} = \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}^{\otimes k-1} \otimes |0\rangle\langle 0| \otimes \mathbb{1}^{\otimes n-k}. \tag{2}$$

2.2 Circuit knitting

Performing any measurements on the variational state defined on the composite Hilbert space $\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_N$ requires running circuits spanning across all blocks. To realize measurements on circuits of smaller sizes, we utilize circuit knitting techniques [19–23] to cut cross-block gates and recover the entanglement using additional circuit evaluations and classical post-processing. Circuit knitting allows decomposing a global quantum channel $\mathcal{U}$ acting on a quantum state ρ into locally realizable quantum channels $\mathcal{E}_k^i$ according to a quasi-probability decomposition (QPD)

$$\mathcal{U}[\rho] = \sum_{k=1}^{K} \alpha_k \mathcal{E}_k^1 \otimes \mathcal{E}_k^2 \otimes \cdots \otimes \mathcal{E}_k^N[\rho], \tag{3}$$

for $K \in \mathbb{N}$ and $\alpha_k \in \mathbb{R}$. In our specific case, $\mathcal{U}$ will be the channel defined by a unitary gate acting on qubits of separate blocks $\mathcal{H}_i \otimes \mathcal{H}_j$, $\rho = |\psi\rangle\langle\psi|$ is the pure state defined by the circuit prior to applying this gate, and $\{\mathcal{E}_k^i, \mathcal{E}_k^j\}$ are the corresponding set of channels that act locally only within each subsystem $\mathcal{H}_i$ or $\mathcal{H}_j$.

In practice, for every circuit evaluation, the global channel $\mathcal{U}$ is replaced by some locally realizable channel $\mathcal{E}_k = \mathcal{E}_k^1 \otimes \cdots \otimes \mathcal{E}_k^N$ sampled according to the probability distribution defined by $p_k \propto |\alpha_k|$. While the QPD provides an unbiased estimator of the true expectation value of the measurement, the sampling cost required to achieve the same precision increases. Crucially, some of the α_k can be negative, which leads to a sampling overhead of

$$\omega(\mathcal{U}, \{\mathcal{E}_k^i\}_{k,i}) = \left(\sum_k |\alpha_k|\right)^2. \tag{4}$$

This overhead is multiplicative[1] and, hence, scales exponentially in the number of gates that are cut.

2.3 Overhead constrained PVQD

The circuit that needs to be run to evaluate the fidelity in Eq. (1) is composed of gates arising from the Trotter step unitary $e^{-i\Delta t H}$ and gates in the variational ansatz state $|\psi(\theta)\rangle = U(\theta)|0\rangle$ that potentially span across multiple blocks and thus have to be cut (see Fig. 1). For the Trotter gates, we restrict the analysis to 2-local Hamiltonians, such that the multiqubit gates appearing in the Trotter expansion are given by two-qubit rotations defined as $e^{-i\Delta t J_{ij}\sigma_i \otimes \sigma_j}$, for Pauli operators $\sigma_i, \sigma_j \in \{X, Y, Z\}$ and coupling coefficients $J_{ij} \in \mathbb{R}$[2]. The sampling overhead imposed by cutting a single instance of this gate with the optimal decomposition is given as $\omega_{J_{ij}} = \left(1 + 2|\sin(2\Delta t J_{ij})|\right)^2$ [22, 23]. For the time evolution to be accurate, we require Δt to be small. Moreover, we consider only cases in which the coupling J_{ij} between qubits of different blocks is weak. Hence, we can assume $\Delta t J_{ij} \ll 1$, and thus, $\omega_{J_{ij}}$ is close to 1. If the Trotter step requires a total of L such gates to be cut, the overhead scales as $\omega_{\Delta t} = \omega_J^L$, where for simplicity, we take $J_{ij} = J\ \forall ij$. While this scales exponentially in the number of gates, the base is small, and for a finite number of blocks, the overhead remains manageable.

[1] In general, the overhead is sub-multiplicative as, for the combination of multiple gates, a more efficient QPD can be found [23,51,52]. In order to allow for a straightforward implementation of the circuit knitting scheme, in the following, we nevertheless assume a multiplicative sampling overhead.

[2] This case includes widely studied spin-1/2 Hamiltonians like the Ising or Heisenberg models. However, our framework can be extended to Hamiltonians with k-local interactions.

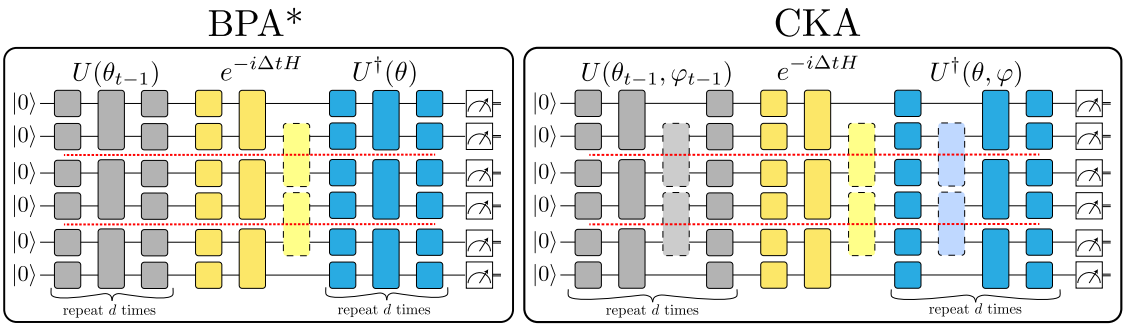

Figure 1: Circuits used to measure the (local) fidelity between the time evolved state $e^{-i\Delta t H}U(\theta_{t-1},\varphi_{t-1})\,|0\rangle = e^{-i\Delta t H}\,|\psi(\theta_{t-1},\varphi_{t-1})\rangle$ and the ansatz $|\psi(\theta,\varphi)\rangle$ required for a PVQD optimization step. We cut the circuits into distinct blocks (indicated by the red dashed lines), which can each be simulated on a separate quantum device. In the experiments considered in this work, the single-qubit gates are realized using R_X rotations, while the two-qubit gates correspond to R_{ZZ} rotations. Gray-shaded gates are fixed by the parameters of the last time step t, while the parameters of the blue-shaded gates are varied to optimize Eq. (6). This structure is repeated d times to increase the expressibility of the ansatz.The trotterized time evolution unitaries are colored yellow. **Left panel:** Block product ansatz (BPA*) where the only entangling gates between different blocks appear in the Trotter step. (In a pure block product approximation (BPA) all inter-block gates are omitted, including the ones arising in the time evolution step.) **Right panel:** Circuit knitting ansatz (CKA). Here, additional entangling gates between the different blocks are introduced into the ansatz. For clarity, the parameters of these dashed, light-colored gates are labeled φ, whereas θ denotes the angles of all other gates that do need to be cut.

For the cross-block gates introduced by the variational state $U(\theta)$, the analysis is less straightforward, as generally, the ansatz can be constructed from an arbitrary gate set. Many commonly used ansatzes consist of parameterized single-qubit rotations followed by CNOT gates that impact the entanglement. Cutting a CNOT gate, however, comes at a fixed cost of $\omega_{\mathrm{CNOT}} = 9$. Even when employing more intricate cutting schemes that reduce the overhead of cutting n CNOT gates simultaneously, the sampling overhead grows as $\omega_{\mathrm{CNOT}^{\otimes n}} = (2^{n+1}-1)^2$ [51]. An alternative class of two-qubit gates that allow more control over the sampling overhead when being cut are parameterized two-qubit rotations such as those appearing in the Trotter decomposition. If one cuts M two-qubit rotations with angles $\varphi_1,\dots\varphi_M$, the multiplicative sampling overhead needed to evaluate the PVQD loss function with the circuit knitting scheme is given as

$$\omega(\varphi) = \omega_{\Delta t}\cdot\left(\prod_{i=1}^{M}\left(1+2|\sin(\varphi_i)|\right)^2\right)^2, \tag{5}$$

where $\omega_{\Delta t}$ is the overhead due to cutting the Trotter step, and the additional square appears due to doubling the circuit (see Fig. 1). The total overhead can become extremely large if the angles φ_i are unbound. A way to circumvent this issue is to employ a block product ansatz that does not introduce any entangling gates between different blocks. This ansatz is shown in Fig. 1 on the left and labeled as BPA* to distinguish it from a pure block product approximation (BPA) where also the entangling Trotter gates are omitted. While the BPA* comes at a minimal sampling overhead, it is not able to capture any entanglement between different blocks. Even for weakly entangled systems, the ansatz is thus expected to fail after evolving the system for a long enough time. Hence, it becomes necessary to add parameterized entangling gates between different blocks of the ansatz state (see right panel of Fig. 1). We refer to this type of ansatz as circuit knitting approximation (CKA).

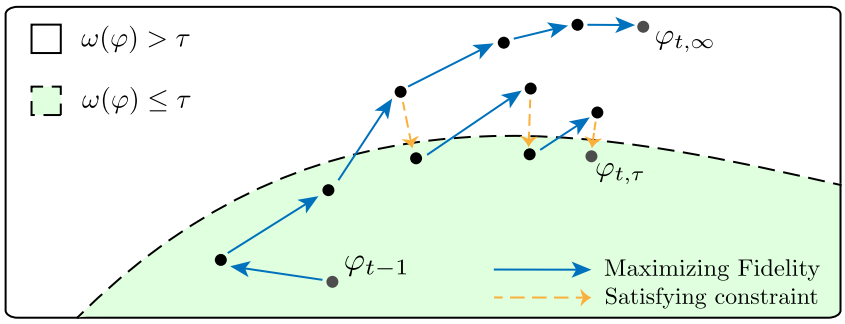

Figure 2: Solving the constrained optimization problem defined in Eq. (6) to evolve the ansatz state by one-time increment. The optimization starts at the parameters of the last time step $t-1$. In every iteration, the parameters are first updated to maximize the fidelity with respect to the true time evolved state using an ADAM [53] update step (blue arrow). After the update, the multiplicative sampling overhead $\omega(\varphi)$ is computed according to Eq. (5) and compared against the threshold τ. In case $\omega(\varphi) > \tau$, the parameters are projected onto the manifold of $\omega(\varphi) \leq \tau$ (orange dashed arrow). This procedure is repeated until the parameters converge; the final point is labeled as $\varphi_{t,\tau}$. In contrast, the path on the top represents the usual, unconstrained optimization with no predefined threshold that converges to different parameters $\varphi_{t,\infty}$ which, however, incur an uncontrolled sampling overhead.

In order to keep the overhead ω controllable throughout the optimization of the CKA, we add a constraint to the optimization of Eq. (1) such that ω is always bound by a threshold $\tau > 1$

$$\begin{aligned} \theta_t, \varphi_t = &\arg\min_{\theta,\varphi} \left[1 - \left| \langle \psi(\theta,\varphi)| \, e^{-i\Delta t H} \, |\psi(\theta_{t-1},\varphi_{t-1})\rangle \right|^2\right] \\ &\text{s.t.}\, \omega(\varphi) \leq \tau, \end{aligned} \tag{6}$$

where we denote by θ the parameters of gates acting within a single block and by φ the parameters of gates that are being cut, i.e. two-qubit gates stretching across two blocks. We satisfy the constraint throughout the optimization by projecting the parameters φ back into the allowed subspace defined by $\omega(\varphi) \leq \tau$ (see Fig. 2). This projection is performed after every PVQD update step, which would result in circuits exceeding the predefined overhead threshold. We note that the number of entangling gates is fixed by the ansatz structure and the overhead is controlled by tuning the values of the parameter in those gates. In some cases, this might lead to gates being effectively removed from the circuit when the rotation angles are set to 0 during the optimization. This behavior is analyzed in detail in Appendix B. The steps of the algorithm are outlined in Algorithm 1, and an in-depth description is provided in Appendix A.

3 Results

As an example application of our method, we consider the transverse field Ising model (TFIM) spin system

$$H = \sum_{\langle ij \rangle} J_{ij} Z_i Z_j + \sum_i X_i, \tag{7}$$

where we assume that the coupling between neighboring spins J_{ij} is large for i, j in the same block and small for i, j in different blocks. In all subsequent simulations, we start the time evolution from the product state $|0\rangle^{\otimes n}$ of all n spins pointing up. We compare our circuit knitting ansatz (CKA) with different thresholds τ to a pure block product

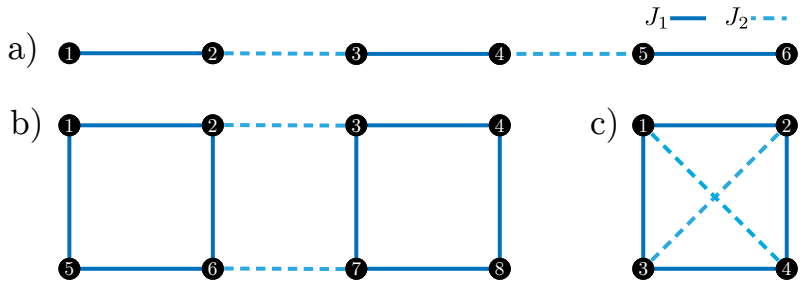

Figure 3: Spins in a transverse field Ising model. In (a) and (b), the system is split into blocks, where the coupling between two blocks J_2 is much smaller than the coupling within a block J_1. In (c), the strong coupling corresponds to nearest-neighbor interactions, whereas the next-nearest neighbor interactions are weak.

approximation (BPA) and to the BPA*, where the full Trotter step, including all cross-block interactions, is implemented.

3.1 Spin chain

In the first experiment, we consider a spin chain of $N = 3$ blocks of 2 spins each, as shown in Fig. 3 (a). The coupling within one block is chosen to be at the critical point $J_{ij} = J_1 = 1$, whereas the coupling between two blocks is set to $J_{ij} = J_2 = 1/4$. The ansatz follows the structure of the Trotter decomposition of $e^{-i\Delta t H}$ with $d = 3$ repetitions of alternating layers of R_X and R_{ZZ} rotations (see Fig. 1). The total number of R_{ZZ} gates that need to be cut is $N - 1$ for the BPA* and $(2d + 1)(N - 1)$ for the CKA.

In Fig. 4 (a) we plot the fidelity of time-evolved states obtained through PVQD state vector simulations with respect to the exact solution. We observe that the pure block product ansatz optimized with block product Trotter gates (BPA) has the poorest performance as the fidelity quickly drops and reaches a value of only 0.87 at time $t = 2$. This behavior is, however, expected since neither the ansatz nor the optimization takes into account any interactions or entanglement between different blocks of the systems. Adding (and cutting) the Trotter gates involving cross-block interactions while keeping the same block product ansatz (BPA*) slightly increases the fidelity. Finally, we expand the ansatz itself by adding parameterized gates between the blocks which are cut (CKA), and employ the overhead-constrained PVQD algorithm for the evolution. We are able to control the fidelity by tuning the threshold hyper-parameter τ that constrains the allowed sampling overhead for the ansatz. Ultimately, our optimization scheme gives us the means to naturally interpolate between the results obtained with a block product ansatz which incurs only a minimal sampling overhead, and the unconstrained PVQD evolved state, which gives rise to an unbounded overhead.

In Fig. 4 (b), we show the evolution of a correlated observable acting on all three blocks. The behavior of long-ranged observables is typically more difficult to capture in hardware-efficient variational simulations, as their support grows faster compared to purely local observables. The BPA(*) is expected to fail in representing correlations spanning across different blocks, as becomes evident at times $t > 1$. In contrast, the CKA with the particular thresholds chosen here can capture the inter-block correlations accurately also for long times.

To explicitly see how the fidelity obtained with the overhead-constrained optimization

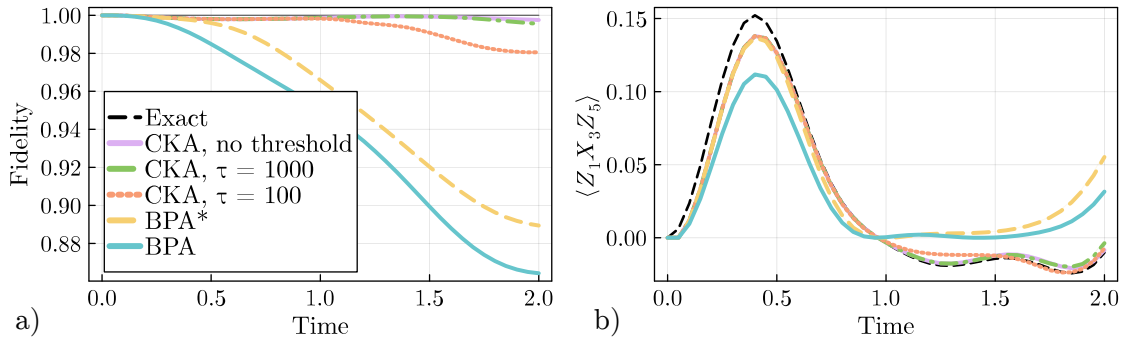

Figure 4: Simulating the dynamics of a TFIM spin chain consisting of 3 blocks with 2 spins each. **(a)** Fidelity of our time evolved ansatz with respect to the exact solution. **(b)** Expectation value of the observable acting as Z on spins 1 and 5 and as X on spin 3. We compare an ansatz involving parameterized two-qubit gates between blocks that are cut (CKA) while keeping the sampling overhead controlled under a threshold τ and a block-product state ansatz without entangling gates between the blocks (BPA). In the case of the latter, we further differentiate between optimizing with a block-product Trotter gate (i.e, no inter-block interactions) or the full Trotter gate, including the exact inter-block interactions (BPA*). We find that CKA can reach higher fidelities at longer times while the exact accuracy can be controlled by changing the overhead threshold.

increases with a higher threshold, we plot the mean infidelity

$$I = \frac{1}{T}\sum_{t=1}^{T}\left[1 - |\,\langle\psi(\theta_t,\varphi_t)|\psi_t\rangle\,|^2\right] \tag{8}$$

of the simulations with respect to the exact solution $|\psi_t\rangle$ in Fig. 5 (a). A larger overhead threshold improves the expressibility of the ansatz as the inter-block gate parameters are less constrained. As a result, the mean infidelity decreases. In order to fully quantify the computational cost required to achieve a certain fidelity, we further include a shot-based simulation, taking into consideration finite sampling noise. Fig. 5 (b) shows how the mean infidelity decreases as the total number of shots is increased. For every point, 10 simulations were performed with a fixed number of shots R per circuit evaluation. The total number of shots for every run is calculated as

$$R_{\text{tot}} = R \cdot n_{\text{iter}} \cdot 2\, n_{\text{params}} \cdot \sum_{t=1}^{T} \omega(\varphi_t), \tag{9}$$

where $n_{\text{iter}} = 200$ is the number of iterations per time step[3], $2 \cdot n_{\text{params}}$ the cost of calculating the gradient using the parameter shift rule for n_{params} parameters, and $T = 40$ the number of time steps in the simulation. The overhead is set to 1 for the BPA. While Fig. 5 (a) suggests that increasing the threshold improves the expressibility of the ansatz, leading to decreasing infidelities, Fig. 5 (b) demonstrates that shot noise limits the simulation from reaching the ideal infidelity. Given a fixed budget of total shots R_{tot}, choosing the optimal threshold τ and the number of shots R per circuit evaluation is a nontrivial constrained optimization problem. In Fig. 5 (a), this balance is illustrated as there is a regime around 10^{11} total shots where having a lower threshold (but larger R) results in lower infidelity than a high threshold (but smaller R). On the other hand, for a higher budget of around 10^{12} total shots, choosing the larger threshold is advantageous.

[3]This number has been chosen high to ensure the optimization converges. Exploiting smart termination criteria should allow to significantly reduce the number of iterations.

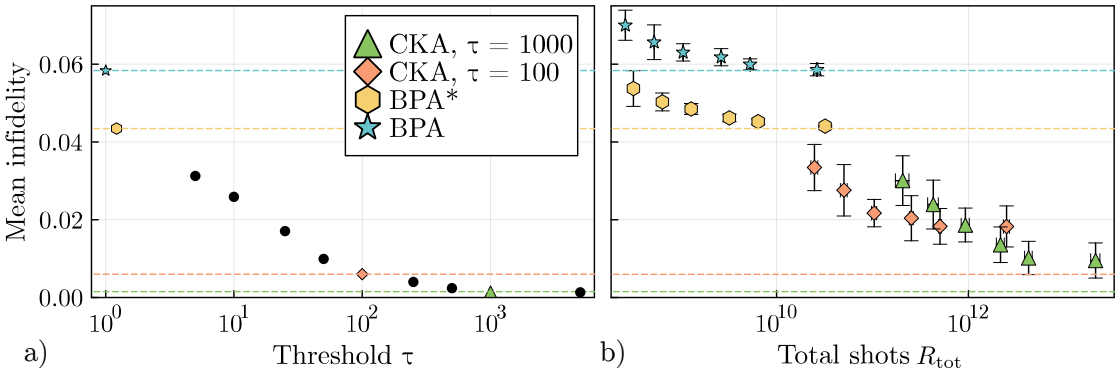

Figure 5: Mean infidelity over the time evolution. In **(a)** as a function of the threshold τ constraining the sampling overhead in the statevector simulations presented in Fig. 4. The black points correspond to additional simulations performed (for thresholds 5, 10, 25, 50, 250, 500, 5000) to interpolate between the points shown in the other plots. In **(b)** as a function of the total shots required for the simulation, with dashed lines indicating the result achieved by the statevector simulation. Note that here we do not plot the CKA without a threshold, as the first point of this method would start at 10^{17} total shots and is, thus, unfeasible in reality.

We further investigate how entanglement between different blocks can be captured with the CKA and how the entanglement correlates to the sampling overhead. We generally expect that imposing a threshold on the sampling overhead required for the circuit knitting scheme limits the entanglement that can arise between the subsystems. In order to quantify the entanglement in our ansatz state, we split the system shown in Fig. 3 (a) into a bipartite system. We call A the subsystem containing the center block and B the subsystem containing the outer two blocks. We write the pure state $|\psi\rangle = U(\theta, \varphi)\,|0\rangle$ defined by the quantum circuit in its Schmidt decomposition

$$|\psi\rangle = \sum_{k=1}^{\dim(A)} \lambda_k\,|a_k\rangle\,|b_k\rangle\,, \tag{10}$$

where $\lambda_k \geq 0$ are the Schmidt coefficients, $|a_k\rangle\,, |b_k\rangle$ the Schmidt basis states in systems A and B, respectively. From this decomposition, the von Neumann entanglement entropy can be easily computed as [54]

$$E(|\psi\rangle) = -\sum_{k=1}^{\dim(A)} \lambda_k^2 \log\left(\lambda_k^2\right). \tag{11}$$

In Fig. 6 (a), we show how the entanglement entropy grows in time for different ansatzes. As expected, the BPA(*) captures no entanglement between the distinct blocks, while the CKA without a threshold recovers the full entanglement of the exact solution. For the CKA with $\tau = 100, 1000$, we observe that the entanglement entropy eventually starts deviating and stays below its exact value as expected. To understand whether the errors in the entanglement entropy arise due to the constrained optimization problem, we also show how the sampling overhead increases over time and, if applicable, caps at the threshold (see Fig. 6 (b)). Interestingly, the entanglement entropy is growing even after the sampling overhead saturates (indicated by the vertical lines) and does not plateau to a specific value. In this case, the optimization learns that due to the multiplicative overhead it is more efficient to have few entangling gates with large angles compared to many gates with small angles. Once the threshold is reached, the entanglement generation thus starts to

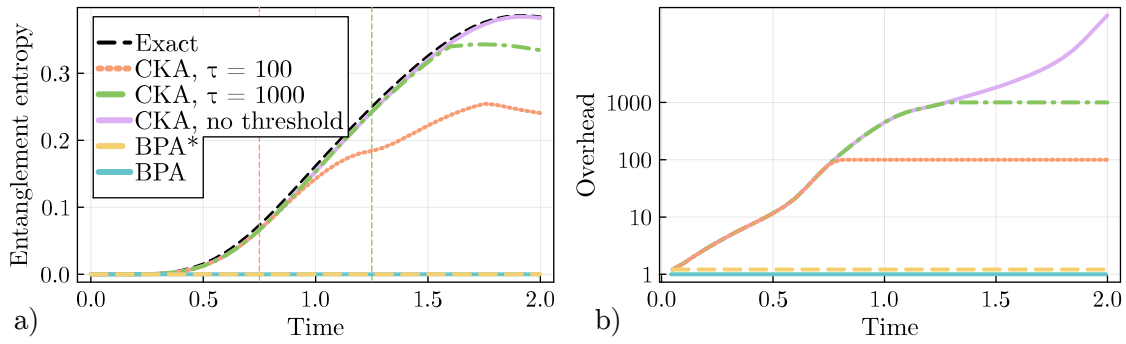

Figure 6: **(a)** The entanglement entropy between the center block and the outer two blocks is calculated as a function of time for different ansatzes. The overhead threshold in the CKA determines the time window for which the entanglement growth can be captured accurately. In contrast, the BPA(*) is not able to account for any inter-block entanglement by construction. **(b)** Required sampling overhead versus simulation time for different thresholds. We indicate the exact time at which the overhead reaches the threshold as vertical dashed lines in (a).

be concentrated on a few gates which allows the entanglement entropy to grow further. A detailed explanation for this behavior is provided in Appendix B.

3.2 Two-leg ladder

Next, we demonstrate that our scheme can also be applied to lattice geometries beyond the simple 1d spin chain. To that end, we consider the Ising model on a two-dimensional extension of the chain as shown in Fig. 3 (b). Each of the two blocks is comprised of 4 spins and coupled to the other block via weak nearest-neighbor interactions. To simulate the dynamics of this system, we choose an ansatz layout reflecting the corresponding trotterized time evolution operator. Specifically, we use alternating layers of single-qubit R_X rotations and R_{ZZ} rotations, repeated $d = 5$ times. The total number of R_{ZZ} that need to be cut is 2 for the BPA* and $2(2d + 1)$ for the CKA.

In Fig. 7, we show how the fidelity of the different ansatzes with respect to the exact solution evolves in time. Additionally, we plot the expectation value of the observable that acts as X on the four outer qubits (the two qubits on the left of the first block and the two qubits on the right of the second block). While the block product ansatz (BPA*) initially tracks the qualitative behavior of the dynamics, it fails in the second half of the simulation period. Here, adding the cross-block entangling unitaries to the ansatz is necessary to accurately approximate the time-evolved state. The CKA with a threshold of $\tau = 100$ captures the qualitative dynamics of the observable plotted in Fig. 7 until $t \approx 1.5$. In order to accurately simulate the dynamics until $t = 2$, the threshold has to be increased to $\tau = 1000$.

3.3 Reducing circuit depth

Many state-of-the-art quantum computing platforms such as those based on superconducting qubits feature only a limited qubit connectivity. Gates acting on qubits that are not adjacent in the device layout have to be implemented via additional two-qubit SWAP operations. However, these extra gates increase the amount of noise in a computation. In the era of NISQ devices, it is therefore crucial to find ways of reducing the circuit depth while keeping the simulations as accurate as possible. To that end, circuit knitting can be

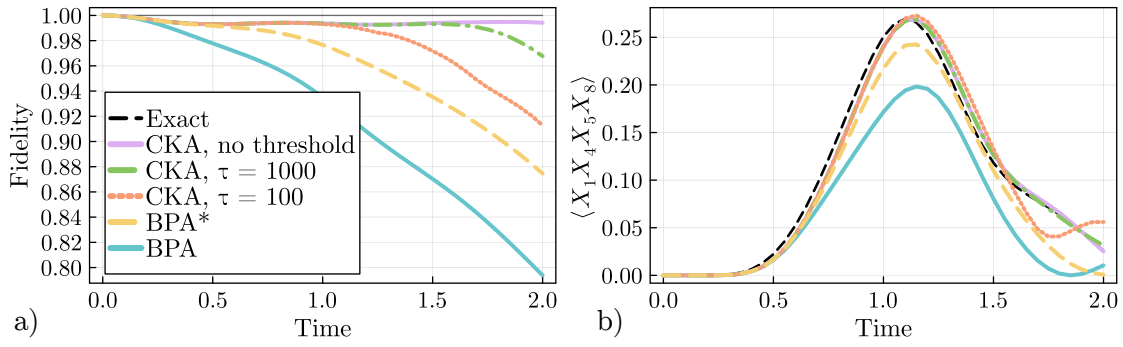

Figure 7: Simulating the dynamics of the two-leg ladder TFIM. **(a)** Fidelity of the simulation with respect to the exact solution for the BPA, BPA* and CKA with different thresholds. **(b)** Expected value of the observable acting as X on the four outer qubits (cf. in Fig. 3 (b)). The accuracy of the simulation improves as the threshold is increased.

employed to cut long-range acting gates.

Here, we demonstrate the use of circuit knitting to effectively reduce the circuit depth in the variational simulation of the dynamics of the J1-J2 transverse field Ising model depicted in Fig. 3 (c) and defined by

$$H = J_1 \sum_{\langle ij \rangle} Z_i Z_j + J_2 \sum_{\langle\langle ij \rangle\rangle} Z_i Z_j + \sum_i X_i, \tag{12}$$

where we again choose $J_1 = 1$, $J_2 = 1/4$. $\langle ij \rangle$ indicates nearest-neighbors, whereas $\langle\langle ij \rangle\rangle$ corresponds to next-nearest-neighbors. Instead of cutting the system into blocks, we here cut the long-range gates induced by the next-nearest-neighbor interactions. We compare the PVQD dynamics for similar ansatzes as in the previous experiments with $d = 4$ repeated layers. Specifically, we consider an ansatz that is composed only of hardware-efficient gates, i.e, gates acting only on nearest-neighbor spins/qubits. For consistency, we refer to this ansatz as BPA(*), even though we are not cutting the system into blocks in this case. In contrast, the CKA ansatz reflects the full interaction graph of the model and contains additional $4(2d + 1)$ long-range entangling gates that are cut using circuit knitting.

The results of our simulations are provided in Fig. 8, where we show both the time dependence of the fidelity to the exact state and of an observable acting on two non-adjacent spins. In the BPA, all gates acting on next-nearest-neighbors are omitted from the circuit, including the Trotter step. As a result, the fidelity quickly deteriorates as we effectively evolve with a slightly different model where $J_2 = 0$. In contrast, for BPA*, the finite next-nearest-neighbor interactions are included in the Trotter step while the ansatz is kept hardware-efficient. In this case, the fidelities stay high throughout the time evolution interval. Hence, the hardware-efficient ansatz comprised of $d = 4$ repeated layers is already able to accurately represent the long-range correlations and entanglement generated by the next-nearest-neighbor interactions of the model. However, we can improve on these fidelities even further by using the CKA with a comparatively small overhead threshold of $\tau = 10$.

In order to quantify the depth reduction enabled by cutting long-ranged gates in this example, we count the number of SWAP gates required to run PVQD on this system without cutting any gates. Given a quantum device where the connectivity coincides with

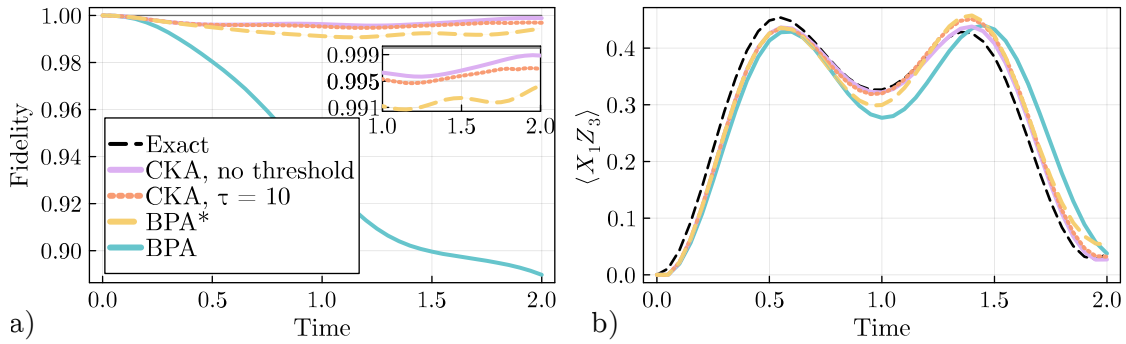

Figure 8: Simulating the dynamics of J1-J2 Ising model sketched in Fig. 3 (c). We show **(a)** the fidelity of the simulation with respect to the exact solution as well as **(b)** the expectation value of the observable acting as X on the upper left qubit and as Z on the lower right qubit. BPA here indicates a hardware efficient circuit, whereas CKA includes next-nearest-neighbor 2-qubit gates that are cut using circuit knitting while the overhead is constrained by a threshold τ. The main improvement over the BPA is given by BPA*, where next-nearest-neighbor interactions are considered in the Trotter decomposition of $e^{-i\Delta t H}$.

this geometry (i.e. nearest-neighbor spins/qubits are connected), every Trotter layer would require 2 SWAP gates. For the ansatz with 4 repeated layers, this results in $2{+}2{\cdot}4{\cdot}2 = 18$ SWAP gates that can be saved using circuit knitting (2 for the Trotter step, $4 \cdot 2$ for the ansatz, and the extra factor of 2 comes from doubling the circuit in the compute-uncompute method).

Overall, in this example, circuit knitting enables us to trade-off a larger circuit depth for an increased sampling overhead.

4 Discussion & Outlook

Circuit knitting allows for the simulation of larger quantum systems using small quantum devices. While in general, cutting circuits can be expensive due to the sampling overhead, we show that this overhead can be controlled by constraining the parameters in the variational circuit optimization. Applying this technique to the dynamics simulation of spin systems with PVQD, we are able to achieve the optimal fidelity given a fixed budget of samples. A change in the threshold hyper-parameter τ leads to a trade-off between the accuracy of the simulation and the sampling overhead. The optimal threshold therefore depends on the quantum computing resources available, the desired accuracy, and the total evolution time.

In the examples considered in this work, we show that with a realistic sampling overhead, the accuracy of the dynamics simulations can be drastically improved compared to a simple block product ansatz. Classical resources can thus effectively be used to recover entanglement between the different subsystems. Our framework opens the door to simulating the dynamics of quantum systems with a large number of qubits that are otherwise not reachable with current hardware. Possible systems of interest are, for example, quantum impurities immersed in a bath [39, 40] or low-energy eigenstates of local lattice Hamiltonians and molecules [35–38]. Furthermore, we show that our technique can also be used to reduce the circuit depth if, instead of cutting the system into blocks, we cut long-ranged but weak interactions. Here, we observed that with a controlled sampling overhead, the dynamics can be accurately simulated with hardware-efficient circuits.

Overhead-constrained circuit knitting allows for combining multiple quantum circuits that are individually already difficult to simulate classically, enabling the simulation of even larger systems given that the entanglement between subcircuits remains weak. This is of particular importance as the current trend in quantum hardware is to combine several smaller devices for distributed computing applications [17]. Hence, our method presents another step towards the overarching goal of quantum utility [12, 55].

The direct application of circuit knitting to a trotterized simulation of dynamics, as performed in the recent utility experiments by IBM [12] is prohibitively expensive. The reason is the accumulated sampling overhead when cutting a Trotter time evolution into subsystems. The sampling overhead is fixed by the coupling strength and scales exponentially with the simulation time and thus cannot be controlled to remain below a manageable threshold. This issue is addressed with our overhead-constrained circuit knitting approach applied to PVQD where we provide a way to control the total budget of circuit evaluations.

An expansion of this work would encompass a hardware experiment of the overhead-constrained PVQD. In this regard, it will be interesting to see whether current error mitigation techniques are powerful enough to mitigate the hardware noise to a level where the (local) fidelities can be measured to sufficient precision for the optimization to be successful.

Moreover, the constrained optimization presented in this work is not limited to PVQD but can be extended to arbitrary loss functions. As such, it could, for example, be applied to simulate ground states using circuit knitting and VQE [56, 57] while keeping the sampling overhead controlled. More general, the overhead-constrained circuit knitting can be extended to any variational quantum algorithms where the total system can be split into weakly entangled blocks. In cases where the optimal partitioning of the system into subsystems cannot be physically motivated, heuristic methods might be applied to find the optimal placement of cuts [58]. Overall, the question of where to optimally place the cuts is a non-trivial optimization problem on its own and represents an interesting direction for future research.

Finally, we remark that the calculations of the sampling overhead throughout this work are based on the worst-case scenario, where the total overhead is the product of the overheads required to cut individual gates. It has recently been proposed that this overhead can be further reduced by using more intricate decompositions that cut multiple gates simultaneously [51, 52]. Alternatively, we could also take into consideration quantum communication between the different devices [59]. In a recently appeared manuscript [31], it has been shown that under similar conditions this can significantly increase the fidelity in distributed simulations of quantum dynamics. However, the hardware to implement a knitting scheme with quantum communication is currently missing and the additional computational costs of such a method will highly depend on how efficient and flexible these quantum links will be.

Code availability Simulations presented in this work were performed in Julia [60] using the `Yao.jl` framework [61] and are available on Github [62].

Acknowledgments We thank Stefano Barison, Julien Gacon, and David Sutter for fruitful discussions on hybrid algorithms, optimization techniques, and circuit knitting. This research was supported by the NCCR MARVEL, a National Centre of Competence in

Research, funded by the Swiss National Science Foundation (grant number 205602).

References

[1] Richard P. Feynman. "Simulating physics with computers". International Journal of Theoretical Physics **21**, 467–488 (1982).

[2] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets". Nature **549**, 242–246 (2017).

[3] A. Chiesa, F. Tacchino, M. Grossi, P. Santini, I. Tavernelli, D. Gerace, and S. Carretta. "Quantum hardware simulating four-dimensional inelastic neutron scattering". Nature Physics **15**, 455–459 (2019).

[4] Frank Arute et al. "Hartree-fock on a superconducting qubit quantum computer". Science **369**, 1084–1089 (2020).

[5] Frank Arute et al. "Observation of separated dynamics of charge and spin in the fermi-hubbard model" (2020). arXiv:2010.07965.

[6] C. Neill et al. "Accurately computing the electronic properties of a quantum ring". Nature **594**, 508–512 (2021).

[7] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z. X. Gong, and C. Monroe. "Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator". Nature **551**, 601–604 (2017).

[8] James Dborin, Vinul Wimalaweera, F. Barratt, Eric Ostby, Thomas E. O'Brien, and A. G. Green. "Simulating groundstate and dynamical quantum phase transitions on a superconducting quantum computer". Nature Communications **13**, 5977 (2022).

[9] Sepehr Ebadi, Tout T. Wang, Harry Levine, Alexander Keesling, Giulia Semeghini, Ahmed Omran, Dolev Bluvstein, Rhine Samajdar, Hannes Pichler, Wen Wei Ho, Soonwon Choi, Subir Sachdev, Markus Greiner, Vladan Vuletić, and Mikhail D. Lukin. "Quantum phases of matter on a 256-atom programmable quantum simulator". Nature **595**, 227–232 (2021).

[10] Ehud Altman. "Many-body localization and quantum thermalization". Nature Physics **14**, 979–983 (2018).

[11] Wibe A. de Jong, Kyle Lee, James Mulligan, Mateusz Płoskoń, Felix Ringer, and Xiaojun Yao. "Quantum simulation of nonequilibrium dynamics and thermalization in the schwinger model". Phys. Rev. D **106**, 054508 (2022).

[12] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, and Abhinav Kandala. "Evidence for the utility of quantum computing before fault tolerance". Nature **618**, 500–505 (2023).

[13] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. "Toward the first quantum simulation with quantum speedup". Proceedings of the National Academy of Sciences **115**, 9456–9461 (2018).

[14] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. "Encoding electronic spectra in quantum circuits with linear t complexity". Phys. Rev. X **8**, 041015 (2018).

[15] Yunseong Nam and Dmitri Maslov. "Low-cost quantum circuits for classically intractable instances of the hamiltonian dynamics simulation problem". npj Quantum Information **5**, 44 (2019).

[16] Mario Motta, Erika Ye, Jarrod R. McClean, Zhendong Li, Austin J. Minnich, Ryan Babbush, and Garnet Kin-Lic Chan. "Low rank representations for quantum simulation of electronic structure". npj Quantum Information **7**, 83 (2021).

[17] Jay Gambetta. "Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing". url: https://research.ibm.com/blog/ibm-quantum-roadmap-2025.

[18] John Preskill. "Quantum Computing in the NISQ era and beyond". Quantum **2**, 79 (2018).

[19] Sergey Bravyi, Graeme Smith, and John A. Smolin. "Trading classical and quantum computational resources". Phys. Rev. X **6**, 021043 (2016).

[20] Tianyi Peng, Aram W. Harrow, Maris Ozols, and Xiaodi Wu. "Simulating large quantum circuits on a small quantum computer". Phys. Rev. Lett. **125**, 150504 (2020).

[21] Kosuke Mitarai and Keisuke Fujii. "Constructing a virtual two-qubit gate by sampling single-qubit operations". New Journal of Physics **23**, 023021 (2021).

[22] Kosuke Mitarai and Keisuke Fujii. "Overhead for simulating a non-local channel with local channels by quasiprobability sampling". Quantum **5**, 388 (2021).

[23] Christophe Piveteau and David Sutter. "Circuit knitting with classical communication". IEEE Transactions on Information TheoryPage 1–1 (2024).

[24] Zhuo Fan and Quan-lin Jie. "Cluster density matrix embedding theory for quantum spin systems". Phys. Rev. B **91**, 195118 (2015).

[25] Klaas Gunst, Sebastian Wouters, Stijn De Baerdemacker, and Dimitri Van Neck. "Block product density matrix embedding theory for strongly correlated spin systems". Phys. Rev. B **95**, 195127 (2017).

[26] Takeshi Yamazaki, Shunji Matsuura, Ali Narimani, Anushervon Saidmuradov, and Arman Zaribafiyan. "Towards the practical application of near-term quantum computers in quantum chemistry simulations: A problem decomposition approach" (2018). arXiv:1806.01305.

[27] Max Rossmannek, Panagiotis Kl. Barkoutsos, Pauline J. Ollitrault, and Ivano Tavernelli. "Quantum HF/DFT-embedding algorithms for electronic structure calculations: Scaling up to complex molecular systems". The Journal of Chemical Physics **154**, 114105 (2021).

[28] Andrew Eddins, Mario Motta, Tanvi P. Gujarati, Sergey Bravyi, Antonio Mezzacapo, Charles Hadfield, and Sarah Sheldon. "Doubling the size of quantum simulators by entanglement forging". PRX Quantum **3**, 010309 (2022).

[29] Patrick Huembeli, Giuseppe Carleo, and Antonio Mezzacapo. "Entanglement forging with generative neural network models" (2022). arXiv:2205.00933.

[30] Paulin de Schoulepnikoff, Oriel Kiss, Sofia Vallecorsa, Giuseppe Carleo, and Michele Grossi. "Hybrid ground-state quantum algorithms based on neural schrödinger forging" (2023). arXiv:2307.02633.

[31] Abigail McClain Gomez, Taylor L. Patti, Anima Anandkumar, and Susanne F. Yelin. "Near-term distributed quantum computation using mean-field corrections and auxiliary qubits" (2023). arXiv:2309.05693.

[32] Stefano Barison, Filippo Vicentini, and Giuseppe Carleo. "Embedding classical variational methods in quantum circuits" (2023). arXiv:2309.08666.

[33] Xiao Yuan, Jinzhao Sun, Junyu Liu, Qi Zhao, and You Zhou. "Quantum simulation with hybrid tensor networks". Phys. Rev. Lett. **127**, 040501 (2021).

[34] Jinzhao Sun, Suguru Endo, Huiping Lin, Patrick Hayden, Vlatko Vedral, and Xiao Yuan. "Perturbative quantum simulation". Phys. Rev. Lett. **129**, 120505 (2022).

[35] J. Eisert, M. Cramer, and M. B. Plenio. "Colloquium: Area laws for the entanglement entropy". Rev. Mod. Phys. **82**, 277–306 (2010).

[36] Ulrich Schollwöck. "The density-matrix renormalization group in the age of matrix product states". Annals of Physics **326**, 96–192 (2011).

[37] Jin-Guo Liu, Yi-Hong Zhang, Yuan Wan, and Lei Wang. "Variational quantum eigensolver with fewer qubits". Phys. Rev. Res. **1**, 023025 (2019).

[38] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. "Quantum computational chemistry". Rev. Mod. Phys. **92**, 015003 (2020).

[39] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti. "Electronic structure calculations with dynamical mean-field theory". Reviews of Modern Physics **78**, 865–951 (2006).

[40] Qiming Sun and Garnet Kin-Lic Chan. "Quantum embedding theories". Accounts of Chemical Research **49**, 2705–2712 (2016).

[41] Stefano Barison, Filippo Vicentini, and Giuseppe Carleo. "An efficient quantum algorithm for the time evolution of parameterized circuits". Quantum **5**, 512 (2021).

[42] P. A. M. Dirac. "Note on exchange phenomena in the thomas atom". Mathematical Proceedings of the Cambridge Philosophical Society **26**, 376–385 (1930).

[43] Jacov Frenkel. "Wave mechanics: Advanced general theory". London: Oxford University Press. (1934).

[44] A.D. McLachlan. "A variational solution of the time-dependent schrodinger equation". Molecular Physics **8**, 39–44 (1964).

[45] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. "Theory of variational quantum simulation". Quantum **3**, 191 (2019).

[46] Julien Gacon, Jannes Nys, Riccardo Rossi, Stefan Woerner, and Giuseppe Carleo. "Variational quantum time evolution without the quantum geometric tensor". Physical Review Research**6** (2024).

[47] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. "Quantum algorithms revisited". Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **454**, 339–354 (1998).

[48] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. "Supervised learning with quantum-enhanced feature spaces". Nature **567**, 209–212 (2019).

[49] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. "Cost function dependent barren plateaus in shallow parametrized quantum circuits". Nature Communications **12**, 1791 (2021).

[50] Tobias Haug and M. S. Kim. "Optimal training of variational quantum algorithms without barren plateaus" (2021). arXiv:2104.14543.

[51] Lukas Schmitt, Christophe Piveteau, and David Sutter. "Cutting circuits with multiple two-qubit unitaries" (2023). arXiv:2312.11638.

[52] Christian Ufrecht, Laura S. Herzog, Daniel D. Scherer, Maniraman Periyasamy, Sebastian Rietsch, Axel Plinge, and Christopher Mutschler. "Optimal joint cutting of two-qubit rotation gates" (2023). arXiv:2312.09679.

[53] Diederik P. Kingma and Jimmy Ba. "Adam: A method for stochastic optimization" (2017). arXiv:1412.6980.

[54] Michael A. Nielsen and Isaac L. Chuang. "Quantum computation and quantum information: 10th anniversary edition". Cambridge University Press. (2010).

[55] Sajant Anand, Kristan Temme, Abhinav Kandala, and Michael Zaletel. "Classical benchmarking of zero noise extrapolation beyond the exactly-verifiable regime" (2023). arXiv:2306.17839.

[56] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. "A variational eigenvalue solver on a photonic quantum processor". Nature Communications **5**, 4213 (2014).

[57] Tuhin Khare, Ritajit Majumdar, Rajiv Sangle, Anupama Ray, Padmanabha Venkatagiri Seshadri, and Yogesh Simmhan. "Parallelizing quantum-classical workloads: Profiling the impact of splitting techniques" (2023). arXiv:2305.06585.

[58] Sebastian Brandhofer, Ilia Polian, and Kevin Krsulich. "Optimal partitioning of quantum circuits using gate cuts and wire cuts" (2023). arXiv:2308.09567.

[59] Daniele Cuomo, Marcello Caleffi, and Angela Sara Cacciapuoti. "Towards a distributed quantum computing ecosystem". IET Quantum Communication **1**, 3–8 (2020).

[60] Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. "Julia: A fresh approach to numerical computing". SIAM Review **59**, 65–98 (2017).

[61] Xiu-Zhe Luo, Jin-Guo Liu, Pan Zhang, and Lei Wang. "Yao.jl: Extensible, Efficient Framework for Quantum Algorithm Design". Quantum **4**, 341 (2020).

[62] Gian Gentinetta, Friederike Metz, and Giuseppe Carleo. "Code for manuscript *Overhead-constrained circuit knitting for variational quantum dynamics*". Github (2024).

A Detailed description of the optimization

In this appendix, we provide a detailed description of the algorithm applied to optimize Eq. (6), including numerical values of hyper-parameters. An overview of the algorithm is given in the main text and in Algorithm 1. The initial guess for the new parameters for time step $t+1$ is given by $\theta^{k=0} = \theta_t + \Delta\theta$, where $\Delta\theta = \theta_t - \theta_{t-1}$ (the same procedure holds for the parameters φ). For the first time step $t = 1$, we set $\Delta\theta = 0$. Starting from this initial guess, we make an ADAM [53] update on θ^k, φ^k according to the gradient of the objective function in Eq. (6), where the gradient is calculated using auto-differentiation for statevector simulations and using the parameter shift rule for shot based simulations. We further modify the ADAM algorithm slightly by keeping the momentum for the inter-block parameters φ at 0. This has been observed to speed up the convergence. Otherwise, we use standard hyper-parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is set to 10^{-3} for statevector simulations and to 10^{-2} for shot-based simulations. This update step yields the next single-block parameters θ^{k+1} and a candidate for the cross-block parameters $\tilde{\varphi}^{k+1}$. The total sampling overhead is computed according to Eq. (5) and compared against the threshold τ. If $\omega(\tilde{\varphi}^{k+1}) \leq \tau$, the parameters are accepted and we continue with the next iteration. If, however, the overhead is larger than the threshold, the cross-block parameters φ are projected back to the region where $\omega(\varphi) \leq \tau$ using the following procedure. The gradient $g = \nabla_\varphi \omega(\tilde{\varphi}^{k+1})$ is calculated using auto-differentiation. Using a step size of $\mu = 10^{-3}$ (for the 1d statevector experiments we set $\mu = 10^{-5}$), we perform $m \in \mathbb{N}$ steps from $\tilde{\varphi}^{k+1}$ along g until

$$\omega(\tilde{\varphi}^{k+1} - m \cdot \mu \cdot \frac{g}{\|g\|}) \leq \tau \tag{13}$$

is fulfilled. The new cross-block parameters are then defined as $\varphi^{k+1} = \tilde{\varphi}^{k+1} - m \cdot \mu \cdot \frac{g}{\|g\|}$, ensuring that the next optimization step starts in the constraint satisfying region. This procedure is repeated until convergence; in our simulations we ran the optimizations for 200 iterations. The number of (purely classical) steps required to project the parameters back to the constraint satisfying region is in the order of $m \approx 1$ for $\mu = 10^{-3}$ and $m \approx 50$ for $\mu = 10^{-5}$. An example of a learning curve resulting from this optimization procedure is given in Fig. 9.

Algorithm 1 Algorithm employed to solve the constrained optimization problem defined in Eq. (6) in order to perform a time step of the overhead-constrained PVQD.

$\theta^0, \varphi^0 \leftarrow \theta_{t-1}, \varphi_{t-1}$
$k \leftarrow 0$
while algorithm not converged **do**
  $k \leftarrow k + 1$
  $\theta^k, \tilde{\varphi}^k \leftarrow$ ADAM [53] update step on the objective function in Eq. (6)
  if $\omega(\tilde{\varphi}_k) > \tau$ **then**
    $g \leftarrow \nabla_\varphi \omega(\tilde{\varphi}^k)$
    $\mu^* \leftarrow \min\{\mu > 0 \,|\, \omega(\tilde{\varphi}^k - \mu g) \leq \tau\}$
    $\varphi^k \leftarrow \tilde{\varphi}^k - \mu^* g$
  else
    $\varphi^k \leftarrow \tilde{\varphi}^k$
  end if
end while
$\theta_t, \varphi_t \leftarrow \theta^k, \varphi^k$

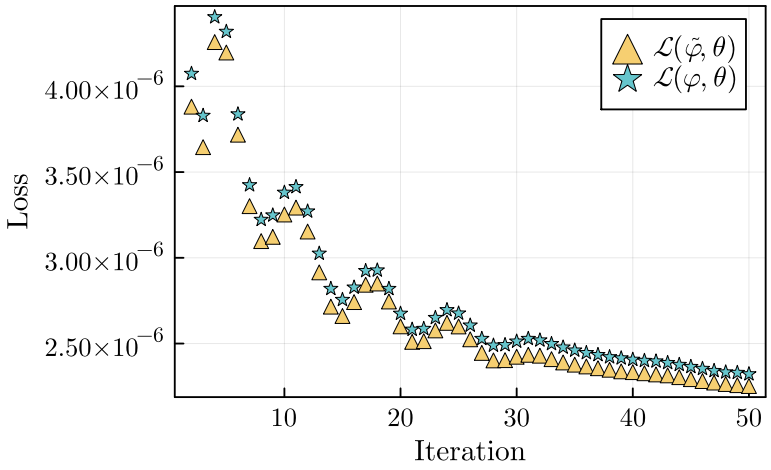

Figure 9: Example of a training curve given by the local infidelity loss function of Eq. (6), here for the 30th time step of the simulation with a threshold $\tau = 1000$ as shown in Fig. 4 of the main text. The blue stars indicate the final loss of each iteration, whereas the yellow triangles show the loss after the ADAM update but before projecting the parameters to the constraint-satisfying subspace. We only plot the first 50 of 200 iterations.

B Parameter behavior in overhead-constrained PVQD

In this appendix, we analyze how the circuit parameters evolve during the overhead-constrained PVQD, explaining how the entanglement between subsystems is able to grow even after the threshold on the sampling overhead has been reached. Fig. 10 shows the circuit parameters during the time evolution of the Ising chain discussed in Section 3.1. The three CKA simulations are identical until the sampling overhead saturates at the threshold τ (indicated by vertical, dashed lines). At this point, the parameters φ of the gates between blocks are no longer freely optimized but are constrained such that the overhead $\omega(\varphi)$ remains below the threshold.

Nevertheless, Fig. 6 in the main text shows that the entanglement entropy increases after the threshold has been reached. This is achieved by reducing the angles in the entangling gates for two of the three layers of the ansatz, allowing the angles in the remaining layer to increase further. The optimization algorithm thus learns that concentrating the generation of entanglement onto one layer reduces the sampling overhead. This is due to the fact that the overhead is multiplicative (see Eq. (5) in the main text).

Indeed, for $\tau = 100$ only one layer is parameterized by non-zero angles at the end of the evolution. For the $\tau = 1000$ case, a similar development is observed, albeit not as extreme (only for one layer the angles become zero). The single-qubit gates and two-qubit gates within a block are optimized to accommodate this transition, leading to a more intricate evolution of the parameters.

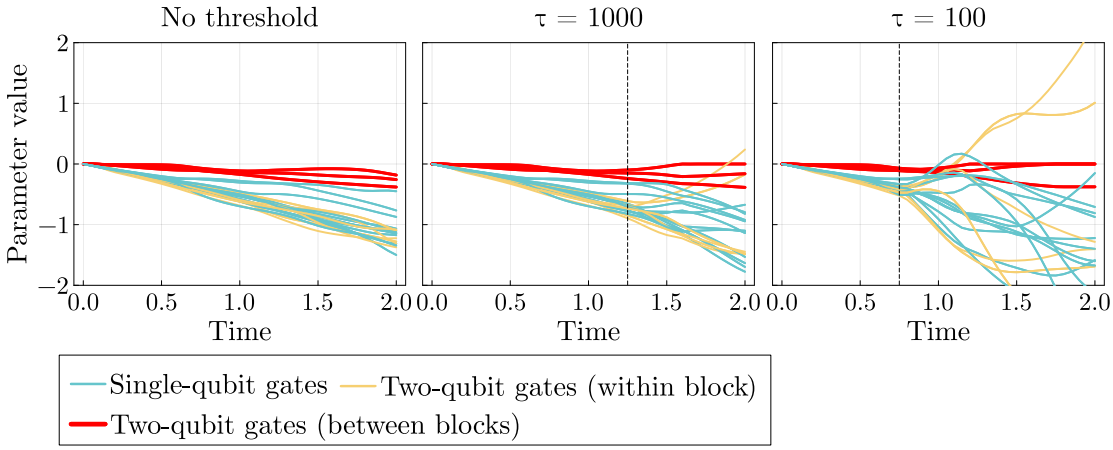

Figure 10: Parameter evolution during simulations of the Ising chain as shown in Fig. 4 of the main text for CKA runs without threshold and with thresholds $\tau = 100,\ 1000$. The vertical black dashed lines indicate the exact time when the sampling overhead reaches the imposed threshold. We differentiate between angles parameterizing the gates between blocks (referred to as φ in the main text) and parameters of the single-qubit and remaining two-qubit gates (referred to as θ). For the CKA simulation without a threshold, the parameters evolve smoothly throughout the evolution. When a threshold is imposed, the parameter evolution becomes more involved once the threshold has been reached. Furthermore, to limit the (multiplicative) overhead, the algorithm effectively removes some of the inter-block gates by reducing their parameters to zero.