arXiv:2409.16361v2 [quant-ph] 25 Jun 2025

Deep Circuit Compression for Quantum Dynamics via Tensor Networks

Joe Gibbs[1,2] and Lukasz Cincio[3]

[1]School of Mathematics and Physics, University of Surrey, Guildford, GU2 7XH, UK
[2]AWE, Aldermaston, Reading, RG7 4PR, UK
[3]Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Dynamic quantum simulation is a leading application for achieving quantum advantage. However, high circuit depths remain a limiting factor on near-term quantum hardware. We present a compilation algorithm based on Matrix Product Operators for generating compressed circuits enabling real-time simulation on digital quantum computers, that for a given depth are more accurate than all Trotterizations of the same depth. By the efficient use of environment tensors, the algorithm is scalable in depth far beyond prior work, and we present circuit compilations of up to 64 layers of $SU(4)$ gates. Surpassing only 1D circuits, our approach can flexibly target a particular quasi-2D gate topology. We demonstrate this by compiling a 52-qubit 2D Transverse-Field Ising propagator onto the IBM Heavy-Hex topology. For all circuit depths and widths tested, we produce circuits with smaller errors than all equivalent depth Trotter unitaries, corresponding to reductions in error by up to 4 orders of magnitude and circuit depth compressions with a factor of over 6.

1 Introduction

The dynamic simulation of quantum many-body systems was an initial motivation for the use of quantum computers [1], and is still considered a leading application for achieving quantum advantage [2]. This can be attributed to a range of factors, including a compact mapping onto qubits, a repeating circuit structure aiding noise model characterization [3], and a rapid growth in entanglement limiting classical simulations [4].

Currently, the most popular approach for performing real-time simulation of quantum dynamics is through the use of product formulas [5]. The propagator e^{-iHt} for a Hamiltonian H cannot be directly applied on digital quantum computers, so it is approximated via Trotterization into time steps composed of lower body operators. Trotter decompositions come in a range of orders that vary in circuit depth and associated error scalings [6]. To reach a tolerable simulation

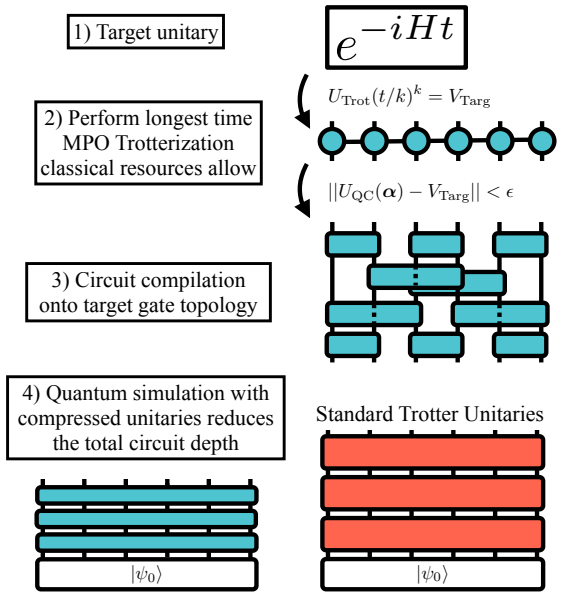


Figure 1: **Overview of Algorithm.** 1) The input of the algorithm is a Hamiltonian H and time step t. The algorithm finds a shallow circuit that approximates the propagator e^{-iHt}. 2) The propagator is represented as an MPO by Trotterization with a fine time step for a negligible Trotter error, for the longest time that results in an MPO with a tractable bond dimension. 3) For the gate topology of a target quantum computer, an optimization is performed to maximize the overlap between the target unitary V_{Targ} and the variational quantum circuit $U_{\text{QC}}(\boldsymbol{\alpha})$. 4) The compressed circuit for the propagator can enable real-time simulation on a quantum computer with a shorter circuit depth than possible using standard Trotterization methods.

error with product formulas, large scale experiments are currently limited by the deep required circuit depths beyond the coherence time of NISQ devices available quantum hardware.

The long term goal of quantum simulation will be enabled by error-corrected fault-tolerant quantum computers. Today's noisy intermediate scale quantum devices (NISQ) [7] are however characterized by an intermediate number of physical qubits with restrictions on circuit depth due to hardware noise. Reducing the number of noisy operations in a quantum circuit, notably the 2-qubit gate count, is critical for achieving a near-term quantum advantage, and remains the primary concern for NISQ algorithm designers.

The need for automated methods of generating short-depth circuits motivated the development of Variational Quantum Algorithms (VQAs) [8]. These are hybrid quantum-classical algorithms that construct a cost function to measure and improve the quality of a parameterized quantum circuit for performing a given task, while constraining the overall quantum computational resources used.

VQAs have been developed to allow a search for compressed circuits generating a target unitary transformation [9, 10, 11]. Khatri *et al* [12] proposed the Hilbert-Schmidt Test in their Quantum-Assisted Quantum Compiling (QAQC) algorithm, a circuit to allow the evaluation of the global overlap between two unitaries with a fixed depth overhead. Given a target unitary, QAQC can then variationally learn an equivalent unitary with smaller gate counts, and compile to a particular quantum computers gateset. Following work used novel out-of-distribution generalization properties of quantum neural networks [13, 14, 15] to allow unitary compilation with significantly reduced circuit resources. VQAs however have well known trainability issues, primarily caused by the infamous Barren Plateau phenomenon [16, 17], and complex cost function landscapes with many local minima [18], limiting large-scale implementations so far. Motivated by the numerous challenges of NISQ computations, researchers seek to maximally utilize the advanced classical computers available to support the quantum computer.

Tensor networks [19] are the leading framework for simulating quantum circuits [20, 21], and we use Matrix Product Operators (MPO) [22] to express and compress our circuits. Prior work has investigated the classical compilation of quantum states represented as Matrix Product States (MPS), into a quantum circuit to be uploaded onto the quantum computer ready for further study [23, 24, 25, 26, 27, 28, 29, 30]. Specifically, we target the common task of performing dynamic Hamiltonian simulation. Refs. [31, 32] proposed variational algorithms for classical optimization of translationally-invariant circuits to reduce errors compared to equivalent depth Trotterizations of the evolution operator. Recent work has also represented quantum circuits as Matrix Product Operators to classically find short-depth unitaries for approximating Trotter evolution [33, 34, 30, 28] .

Recent work has explored the use of MPO-based compilations to compress the time evolution operator e^{-iHt}. Causer *et al* [33] provided a local optimization based on the polar decomposition, however the method presented for contracting gate environments had an exponential scaling in the number of layers. McKeever and Lubasch [34, 35] similarly demonstrated unitary compilations for shallow circuit depths. Tepaske *et al* [36, 37] gave an MPO-based compilation for this task with simulations on systems sizes of up to 16 qubits.

Our work extends the literature by enabling compilation of large scale, deep circuits, with flexibility to target arbitrary topology NISQ devices. Our algorithm allows deeper circuits to be optimized than typically expected to be representable by an MPO, because the gates in our circuit are primarily used to disentangle the target unitary. We demonstrate compressions on large scale and deep parameterized circuits, on up to 200 qubits and 64 layers of $SU(4)$ gates.

An overview of our algorithm is given in Fig. 1. For a target Hamiltonian, H, we compute the propagator e^{-iHt} for the longest time the unitary can be represented as an MPO with a tractable bond dimension. Then we perform a classical circuit compilation onto the gate topology of a target quantum computer, where an optimization of the gates minimizes the error compared to the target time evolution unitary. The optimization of the parameterized quantum circuit is always able to reduce the approximation error to the target propagator compared to equivalent depth Trotter unitaries. Then, this compressed unitary

may be uploaded to the quantum computer and applied multiple times, enabling a real-time simulation of quantum dynamics with a shorter depth than possible with standard Trotterization methods.

After initializing the quantum circuit with an equivalent depth Trotterization, we are always able to find a reduction in the approximation error while maintaining the same circuit depth. For the range of Hamiltonians we test on, including a quasi-2D Hamiltonian expressed on the IBM Heavy-Hex gate topology, we show that our compressions result in up to 4 orders of magnitude reduction in the error, and depth reductions by over a factor of 6.

The rest of the paper is organized as follows. Section 2 introduces the error metric used to compare approximation errors of unitaries, the efficient use of environment tensors, and the full optimization procedure used. Section 3 presents the results of our compression algorithm applied to a range of Hamiltonians, and gives numerical evidence supporting our ability to optimize deep circuits. Finally, Section 4 gives a discussion of our work.

2 Methods

2.1 Error Metric

To evaluate the performance of our compilations, we require a measure of distance between n-qubit unitaries. The metric we use is the Hilbert-Schmidt Test (HST), given by

$$C_{\mathrm{HST}}(U,V) = 1 - \frac{1}{2^{2n}}|\mathrm{Tr}(U^\dagger V)|^2. \quad (1)$$

This is an operationally meaningful quantity as it is directly related to the average state infidelity when applying the two unitaries to a random state [14] e.g.

$$C_{\mathrm{HST}}(U,V) = \frac{2^n+1}{2^n}(1-\mathbb{E}_{|\psi\rangle\sim\mathcal{S}_{\mathrm{Haar}_n}}|\langle\psi|U^\dagger V|\psi\rangle|^2), \quad (2)$$

where $\mathcal{S}_{\mathrm{Haar}_n}$ is the n-qubit Haar random distribution. For our compilations, we have the target unitary V_{Targ} and a variational quantum circuit $U_{\mathrm{QC}}(\boldsymbol{\alpha})$, with these a more explicit version of the cost function we optimize over reads

$$C_{\mathrm{HST}}(\boldsymbol{\alpha}) = 1 - \frac{1}{2^{2n}}|\mathrm{Tr}(U_{\mathrm{QC}}(\boldsymbol{\alpha})^\dagger V_{\mathrm{Targ}})|^2. \quad (3)$$

$C_{\mathrm{HST}}(\boldsymbol{\alpha})$ could be directly computed from the inner product $\mathrm{Tr}(U_{\mathrm{QC}}(\boldsymbol{\alpha})^\dagger V_{\mathrm{Targ}})$, however for n-qubit operators this is an exponentially large quantity. To avoid potential floating-point overflow errors, we instead convert our MPOs into MPS on a doubled physical space, normalize, and then take their inner product. This allows the HST to be computed faithfully, without needing to store exponentially large numerical values. The normalization of this MPS is equivalent to rescaling the original MPO to have a norm of 2^N as expected for a unitary. If this step was not performed, our cost function may become unfaithful and record negative values from to the deviation of the norm of the MPO due to truncation errors.

2.2 Constructing the target MPO

Our algorithm begins by constructing the propagator e^{-iHt} for a target Hamiltonian H, represented as an MPO. The bond dimension of this operator is governed by the total evolution time, the type of interactions, and their range when mapped onto a 1D chain of qubits.

The propagator is constructed by decomposing the operator into a very deep circuit using a high order Trotterization, which is then contracted into an MPO. Specifically we use

$$e^{-iHt} \approx U_{4\mathrm{th}}(t/k)^k, \quad (4)$$

where $U_{4\mathrm{th}}(t)$ is a 4th-order Trotterization, and we set $k = 10$ which has a negligible Trotter error for the Hamiltonians and time steps we study.

We seek to compress unitaries enabling dynamic simulation for the longest time where the target MPO is still representable by an MPO with tractable bond dimension. To determine this longest time, we do a parameter sweep across a range of timescales. For each total evolution time, we iteratively perform the MPO Trotterization for an increasing maximum bond dimension, and compute C_{HST} between the consecutive MPOs. We consider that quantity to be converged when it is below 10^{-10}. We then select the propagator to train on as the largest total evolution time where the iterative MPO Trotterization converged below a final bond dimension threshold. We set this bond dimension to be 128 for our numerical results.

When performing time evolution on a NISQ device, compiling the propagator with an arbitrarily small truncation error will be redundant

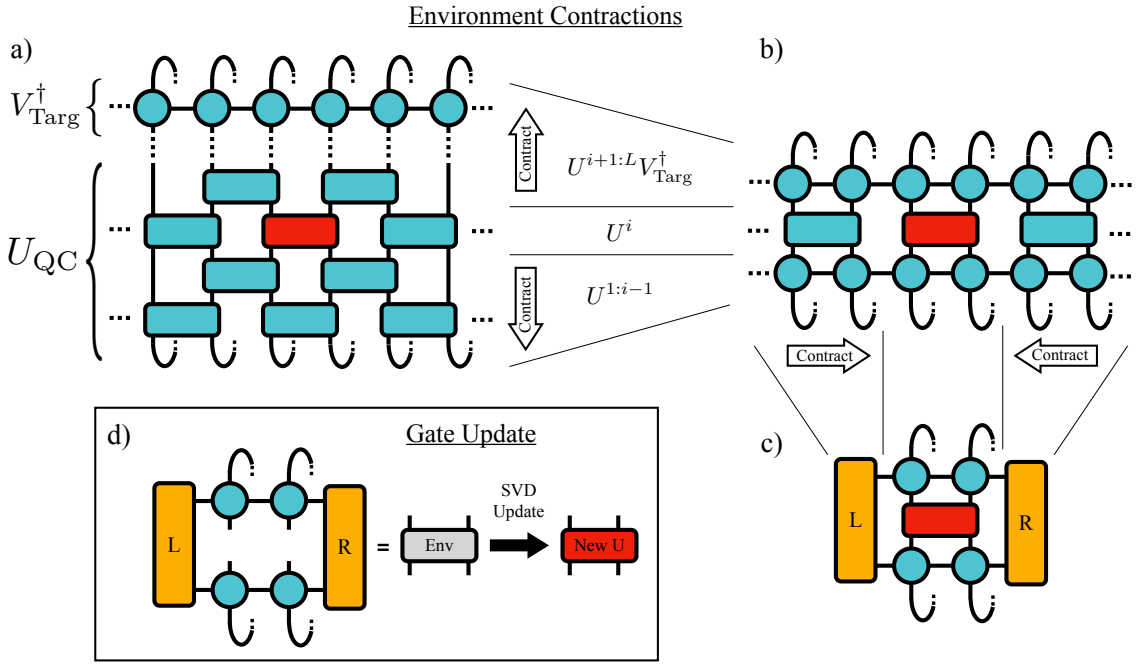


Figure 2: **Efficiencies in Environment Contractions.** a) The contraction for the inner product of the quantum circuit U_{QC} and the target unitary V_{targ}. We iterate over the gates, updating their state to increase the inner product. Starting at the top layer of the circuit, we move down to the first layer and back up to the top. This is one 'sweep'. The curved lines indicate the top and bottom physical legs are connected. b) When optimizing a particular layer of the circuit, the tensor network outside the row of gates is unchanged. To avoid unnecessary re-contraction, we store the state of the network above and below the current layer with environment MPOs. c) When moving along one gate in the layer, the network is fixed to the left and the right. The state outside the current gate is represented by contracted left and right environment tensors. We can therefore cheaply move back and forth updating gates in the layer by reusing the environment tensors L and R. d) The environment of the gate is computed, which is SVD'd to find the locally optimal gate and updated in the circuit.

compared to the compilation error of the outputted quantum circuit, and errors due to hardware noise. For example, if our final compilation error is $C_{\text{HST}}(U_{\text{QC}}, V_{\text{Targ}}) \approx 10^{-4}$, then our total truncation error does not need to be lower than $C_{\text{HST}}(V_{\text{Targ}}, e^{-iHt}) \approx 10^{-5}$. Therefore, before beginning the optimization, we compress the MPO to an error threshold below the target compilation error, which can significantly reduce the bond dimension of the target MPO and the computational burden during training. As a result, this observation enables us to extend the simulation time that is attainable with our method.

Throughout our work we denote this compressed MPO as V_{Targ}. The MPO compression procedure used is a variational compression [4]. The MPO is transformed to an MPS on a doubled physical space, truncated to a lower bond dimension, and then variationally optimized until the fidelity with the uncompressed MPS has converged, and then remapped back into an MPO. We use a single singular value decomposition (SVD) compression sweep as an initialization. If the cost C_{HST} between the target and variationally compressed MPO is above the target error fidelity, we repeat this procedure for a higher bond dimension until the desired error condition is satisfied.

2.3 Optimization

We use an optimization scheme where all gates are iteratively swept through and updated to reduce the cost function given in Eq. (3). For each individual gate, we find their environment in the tensor network, which is singular value decomposed to find the optimal unitary to maximize the overlap. This is a standard optimization procedure in tensor network algorithms [38], and has been used for the classical optimization of circuits represented by state vectors [39, 40, 41], MPS [23, 28] and MPO [33, 28].

To minimize the total circuit depth, we choose our ansatz structure to be the most dense pack-

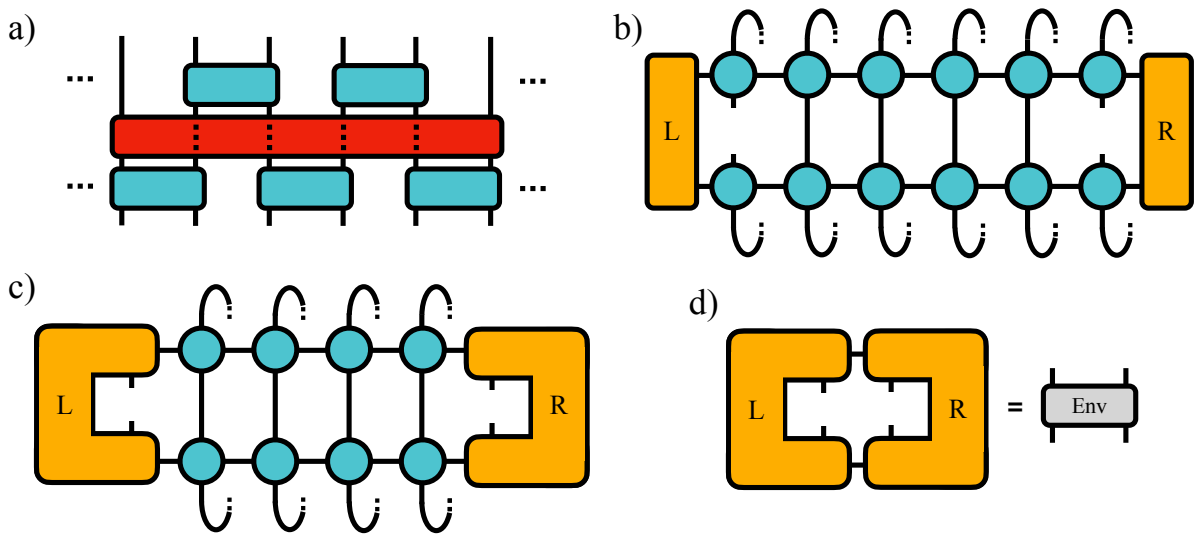


Figure 3: **Environments of non-local gates.** a) A flexible topology circuit compilation may involve local 2D gates mapping onto non-local 1D gates. b) The tensor network giving the environment for a non-local gate acting on qubit indices $\{i, i+5\}$. c) The contraction can maintain a $\mathcal{O}(\chi^2)$ memory and $\mathcal{O}(\chi^3)$ runtime by iteratively contracting neighboring MPO tensors into the left/right environment tensors. d) After all intermediate MPO site tensors have been absorbed, the left and right environment tensors are contracted to give the final gate environment.

ing of commuting layers of $SU(4)$ gates, which by the KAK decomposition are known to require at most 3 entangling gates [42]. Our circuits can be tailored to the specific gate topology of the target quantum computer, rather than assuming access to a 1D chain of qubits with nearest-neighbor connectivity. We detail this further in Section 2.5.

A circuit composed of L layers of commuting gates can be denoted $U^{1:L} = \prod_{i=1}^{L} U^i$, where U^i is the i^{th} layer of commuting gates. At the i^{th} layer we optimize over the gates contained in U^i to minimize $C_{\text{HST}}(\boldsymbol{\alpha})$. We update the circuit in "sweeps", where we start at U^L, move down to U^1, and back up to U^L.

The network is contracted into top/bottom environment MPOs representing $U^{1:i-1}$ and $U^{i+1:L}V^{\dagger}_{\text{Targ}}$ respectively, see Fig. 2b). For a deep and wide circuit, contracting these MPOs is a costly operation. Here we take advantage of the update order to more efficiently reuse and update these environments. When moving sequentially through the circuit, the tensor network is only updated locally within a layer. When moving to the neighboring layer, we can cheaply update the environment MPOs by absorbing the new layer of gates, rather than recontracting the whole network. All circuit templates in this work are derived from Trotterizations, and consecutive layers within the circuits do not commute. However the gates within a particular layer do mutually commute, even for the circuits compiled onto a 2D topology, as each gate acts on disjoint qubits.

When optimizing a particular gate in a layer, we find its local environment by contracting the tensor network shown in Fig. 2b) from the left and the right. This gives the gate's environment tensor by the contraction shown in Fig. 2d). With these left/right environment tensors, denoted by L/R in Fig. 2c), we can cheaply move left/right optimizing the gates within the layer multiple times by locally updating the environment tensors.

While the computation of the required quantities can be made efficient, the navigation during the optimization over the manifold of unitaries can remain challenging, and a good initialization is critical. We find that for a target circuit depth, initializing with an equivalent depth Trotterization gives excellent results with immediate decreases in the cost function, and we use this initialization technique to obtain all our results described in Sec. 3.

2.4 MPO Resets

We note that over the course of many optimization sweeps, truncation errors will cause the MPO representation of the tensor network environment above and below the current row to deviate from faithfully representing the current state of the quantum circuit, causing the computation of incorrect gate environments and degrading the optimization. However, consider the environment

MPOs for the outer layers U^1 and U^L. For U^1, the top and bottom environment MPOs represent $U^{2:L}V^\dagger_{\text{Targ}}$ and $\mathbb{1}$ (the identity) respectively, and for U^L the top and bottom environment MPOs represent $V^\dagger_{\text{Targ}}$ and $U^{1:L-1}$. In both of these cases, one of the environment MPOs represents a unitary that remains fixed throughout the optimization, $\mathbb{1}$ and $V^\dagger_{\text{Targ}}$. As a result, at these outer layers one of the environment MPOs can be reset to their exact value, allowing a simple and cheap method for entirely removing these accumulating truncation errors.

2.5 Targeting a 2D gate topology

Prior works on MPO-based circuit compilations have assumed access to a 1D chain of qubits with nearest-neighbor gates. We extend this to enable a flexible compilation onto the 2D qubit connectivity of a target quantum computer. MPS are a fundamentally 1D tensor network, and a result will be exponentially impeded in one dimension as the size of a target 2D system increases. However, due to their favorable runtime scaling as a function of bond dimension, MPS have been successfully applied to model restricted-width 2D systems, e.g., on infinite cylinders, where local 2D interactions are transformed into non-local 1D interactions. For MPS simulations of a 2D lattice with dimensions $L_x \times L_y$, the computational cost is linear in L_x and exponential in L_y. A calculation scalable in both dimensions must be based on a corresponding 2D tensor network, which is beyond the scope of this work. However, the extension to 2D systems of limited width still provides utility for hardware demonstrations on available quantum hardware, extending the literature of tensor network circuit compilations and exploring the limits of MPS-based compilation of quantum dynamics.

For the compression of unitaries on a 2D gate topology, similarly we will encounter non-local gates acting on the MPO and need an extended contraction scheme for computing gate environments. The gates present in a Trotterization of a 2D Hamiltonian will inevitably result in non-local gates when mapped onto the MPO, requiring an extended contraction scheme for computing gate environments. Fig. 3 shows the required contraction when a gate acts on MPO sites $\{i, i+5\}$. To maintain a scaling of $\mathcal{O}(d^2\chi^2)$ for memory and $\mathcal{O}(d^3\chi^3)$ for runtime (where d is the physical site dimension, and χ is the MPO virtual bond dimension), the contraction is performed by iteratively absorbing the intermediate MPO site tensors into the left/right environment tensors and normalizing, until these are directly connected and can be contracted to give the final gate environment. As described earlier, the environment is SVD'd to find the new locally optimal gate. When finished optimizing a given circuit layer containing these non-local gates, and we wish to apply this layer into the top/bottom environment MPOs, the non-local gates are decomposed into a 2-body MPO and applied using an MPO-MPO zip-up, similar to the MPO-MPS zip-up algorithm proposed in [43].

While the computation of the target MPO was not the dominant cost in our numerical results, this subroutine would become more expensive and limiting for Hamiltonians with longer-range interactions when mapped onto a chain of qubits, for example molecular systems in quantum chemistry. Our compilation only requires access to an MPO representing e^{-iHt} to optimize a variational circuit against. As a result, we do not explicitly require a circuit to compress. More efficient methods of constructing high-order MPOs for time evolution [44, 45] could be used instead, which may extend the range of systems and simulations times our compilations can be applied to.

3 Results

We test our compression algorithm against a range of Hamiltonians: (i) the 1D Transverse-Field Ising model, on 200 qubits for an evolution time of $t = 2.0$; (ii) the 1D Hubbard model, on 100 qubits for an evolution time of $t = 0.2$; (iii) the 1D $J_1 - J_2$ model, on 40 qubits for an evolution time of $t = 1.0$; and (iv) a 2D Transverse-Field Ising model, with the same connectivity graph as the 52 qubit subset of the IBM Heavy-Hex topology shown in Fig. A1, for an evolution time of $t = 0.5$. These Hamiltonians are specified in more detail in Appendix A. We highlight that for the 2D Transverse-Field Ising model demonstration, we perform the compilation assuming access to a quantum computer with the same nearest-neighbor interactions of the gate topology. When mapped onto the MPO, gates that are local on Heavy-Hex lattice will be transformed

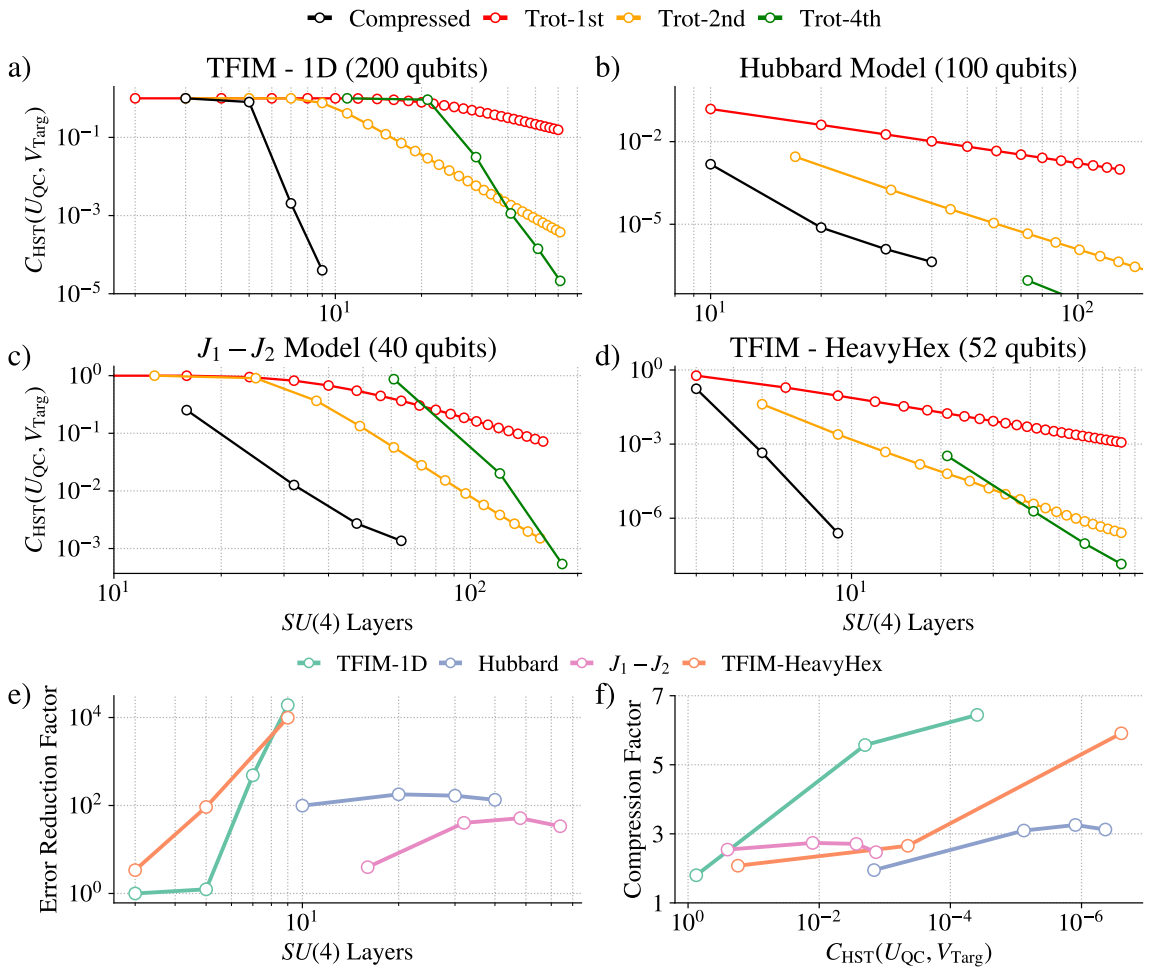


Figure 4: **Compression of Trotter Unitaries** a)-d) For a range of target Hamiltonians, we compile circuits to minimize the approximation error with the propagator for a range of Hamiltonians, specified in Appendix A. The final error upon completion of the optimization is measured in comparison to increasing depth Trotterizations. We compare against Trotterized unitaries of the form $U_r(t/k)^k$ for r = 1st, 2nd and 4th order Trotterizations, shown by the red, orange and green curves. Our numerical results present compressions of Trotter unitaries of up to 200 qubits, and 64 layers of commuting $SU(4)$ gates. e) For each of the compressed circuits, we measure factor by which the cost is reduced compared to the equivalent depth Trotter unitary, with reductions presented up to 4 orders of magnitude. When comparing against Trotter, we use the interpolation of the data points. f) We also display the depth compression factor, which for a particular compressed circuit is the ratio of the circuit depth compared to the shortest Trotter unitary with the same error versus V_{Targ}.

into non-local 1D gates, and we detail the extension this requires in Sec. 2.5. Our compressed circuits are tested against a range of Trotter decompositions. Specifically, for a given parameterized circuit $U_{\mathrm{QC}}(\boldsymbol{\alpha})$, we are interested in the error $C_{\mathrm{HST}}(U_{\mathrm{QC}}(\boldsymbol{\alpha}), e^{-iHt})$, and we express e^{-iHt} by the highly accurate Trotter decomposition given in Eq. (4). We test this against Trotter unitaries of the form $U_r(t/k)^k$ for r = 1st, 2nd and 4th order Trotterizations, shown by the red, orange and green curves in Fig. 4a)-d).

In computing a Trotter unitary for a given Hamiltonian, terms can be reordered to allow neighboring layers of gates acting on the same qubit indices to be absorbed into a single layer. We maximally exploit this to compare our compressed circuits to the shortest depth comparable Trotterization.

For all circuit depths tested, our compressed circuits are able to achieve a smaller approximation error to the target unitary compared to all equivalent depth Trotterizations. During the optimization, we train with a constant maximum bond dimension of 128. However to faithfully determine the progress of the optimization, all data points for shown in Fig. 4 has been obtained by testing $C_{\mathrm{HST}}(U_{\mathrm{QC}}, V_{\mathrm{Targ}})$ with a doubled bond dimension when computing the MPOs U_{QC} and V_{Targ}.

For all Hamiltonians tested, we find at least a

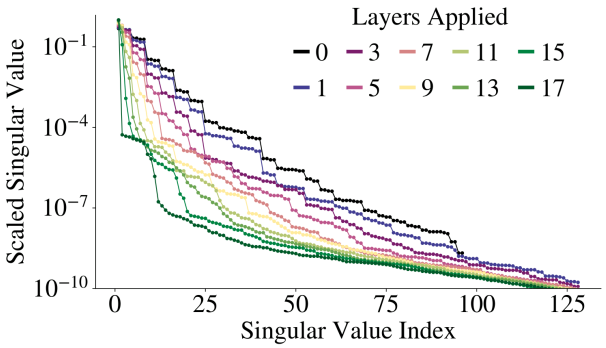


Figure 5: **Faster decay of singular values with increasing layers.** Here we provide numerical results motivating that the compilation is not directly impeded by circuit depth. The target unitary (V_{Targ}) considered is the same MPO trained on in Fig. 4a) for the propagator of the 1D Transverse-Field Ising model. We optimize an $L = 17$ layer circuit to maximize the overlap with V_{Targ} then extract the operator-Schmidt coefficients of the resulting MPO after inverse layers of the optimized ansatz are progressively applied, see text for details. The singular value vector is scaled by dividing by the norm. Specifically, we consider the MPOs of the form $(U^{L-i:L})^\dagger V_{\text{Targ}}$ for increasing values of i. As the data clearly shows, the more layers are applied, the faster the singular values decay, indicating that the dominant cost of the algorithm depends on the bond dimension of the target unitary, rather than the number of layers in the parameterized circuit.

factor of 10 reduction in the cost compared to the best Trotterization, and up to a factor of 10^4 for the Transverse-Field Ising model. Conversely, to achieve a particular compilation error compared to the target unitary, our compressed circuits are typically at least 2 times shallower than the Trotterization with the same error, and up to 6.4 times shallower. We are generous to Trotterization by comparing against the polynomial interpolation of the data points, even if a Trotterization does not exist at that exact depth.

In Sec. 2.2 we described how the longest time propagator is determined for a maximum allowed bond dimension before the compression of the MPO (the bond dimension is set to 128 in this study). We found that for the majority of our results, using the same maximum bond dimension was sufficient for the whole optimization to proceed successfully. Only for the most extreme depth circuits, e.g. greater than 60 layers, we found that the optimization would proceed for a large number of iterations, but later may reach a point where the cost function fails to monotonically decrease. This is due to accumulating truncation errors too great in comparison to the small cost function values, causing the gate environments to become unfaithful. For these restricted cases, we found that restarting the optimization with a modest increase in training bond dimension e.g. from 128 to 156, was sufficient to return the optimization to a smooth decrease of the cost function.

The simulation of quantum circuits with tensor networks is typically expected to require an exponentially growing bond dimension as a function of circuit depth. We stress that for our application, by the efficient use of environment tensors, and the circuit layers effectively acting as disentanglers, the computational cost is instead dominated by the bond dimension of the target unitary. We observe that training on a comparable bond dimension to the target MPO is sufficient to achieve accurate compilation. Those results are corroborated by an observation that optimizing over deep circuit does not significantly increase the required bond dimension to represent the environment MPOs.

We demonstrate this point with numerical results in Fig 5. We take the target MPO for the 1D Transverse-Field Ising model in Fig. 4a) and optimize an $L = 17$ layer circuit, U, to maximize the overlap with V_{Targ}. After progressively applying inverse layers of the parameterized circuit to V_{Targ}, we study the rate of decay of the operator-Schmidt coefficients. Specifically, for the MPO $(U^{L-i:L})^\dagger V_{\text{Targ}}$, we move the orthogonality center onto the central MPO site tensor, contract this with its neighboring tensor, and SVD the joint tensor to inspect the singular values after normalization. In Fig. 5, we show these normalized singular values, and it is clear that as more inverse layers are applied with increasing i, the faster the singular values decay. This behavior can be expected, because as the optimization progresses and the compilation error decreases, the product $(U^{L-i:L})^\dagger V_{\text{Targ}}$ moves closer to the identity MPO. This behavior of our circuit layers acting as disentanglers is key for our compilation being able to perform deep circuit optimizations.

4 Discussion

Simulating quantum dynamics is a leading application for achieving a quantum advantage on near-term quantum computers, however the high circuit depths remain prohibitive on current NISQ devices. To advance the time to productive use of quantum computers, it is desirable to maximally use classical computation to optimize quantum circuits so the algorithms are run more efficiently. In this direction, we build upon work performing classical pre-computation and optimization of quantum circuits, and present a compression of Trotter unitaries for quantum dynamics. Our MPO-based optimization found error reductions by up to a factor of 10^4 compared to equivalent depth Trotterizations, or alternately for a target error rate a depth compression up to a factor of 6.4. By the efficient use of environment tensors, our work extends the literature by enabling far deeper circuit compilations than previously possible, optimizing our results and we show the successful optimization of circuits with up to 64 layers of SU(4) gates. These deep circuit optimizations are enabled by the gate layers acting as disentanglers to the target MPO, allowing deep circuits to be targeted beyond prior works. To demonstrate that our approach can flexibly compile to the connectivity of a target quantum computer, we show the compression of a unitary directly onto the 2D IBM Heavy-Hex topology.

While the computation of gate environments can be made efficient in the limit of large qubit numbers and circuit depths, the success of the algorithm remains dependent on a challenging optimization. We do not claim that the compiled circuits we present are optimal in error versus the propagator for a given circuit depth. Future work could explore more advanced global optimization algorithms e.g. using Riemannian optimization over the unitary manifold [46], however the simple and local gate update used was sufficient to reduce errors compared to Trotterization for every circuit depth tested. This demonstrates practical utility for reducing the resource requirements in the quantum simulation of dynamics. for achieving up to 4 orders of magnitude reduction in approximation errors and over a factor of 6 reduction in circuit depth.

Acknowledgments

We acknowledge the use of the TensorOperations package [47] for our tensor contractions. We thank Faisal Alam for helpful suggestions and contributions to the technical discussion. J.G. was supported with funding and computational resources provided by AWE. L.C. was supported by Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20230049DR as well as by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC05-00OR22725 through the Accelerated Research in Quantum Computing Program MACH-Q project.

References

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

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

[3] Ewout Van Den Berg, Zlatko K Minev, Abhinav Kandala, and Kristan Temme. "Probabilistic error cancellation with sparse pauli–lindblad models on noisy quantum processors". Nature PhysicsPages 1–6 (2023).

[4] Ulrich Schollwöck. "The density-matrix renormalization group in the age of matrix product states". Annals of physics **326**, 96–192 (2011). url: https://www.sciencedirect.com/science/article/pii/S0003491610001752?via%3Dihub.

[5] Seth Lloyd. "Universal quantum simulators". SciencePages 1073–1078 (1996).

[6] Andrew M Childs, Yuan Su, Minh C Tran, Nathan Wiebe, and Shuchen Zhu. "Theory of trotter error with commutator scaling". Physical Review X **11**, 011020 (2021).

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

[8] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. "Variational quantum algorithms". Nature Reviews Physics **3**, 625–644 (2021).

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

[10] Kaoru Mizuta, Yuya O Nakagawa, Kosuke Mitarai, and Keisuke Fujii. "Local variational quantum compilation of large-scale Hamiltonian dynamics". PRX Quantum **3**, 040302 (2022).

[11] Matias Bilkis, M Cerezo, Guillaume Verdon, Patrick J Coles, and Lukasz Cincio. "A semi-agnostic ansatz with variable structure for variational quantum algorithms". Quantum Machine Intelligence **5**, 43 (2023).

[12] Sumeet Khatri, Ryan LaRose, Alexander Poremba, Lukasz Cincio, Andrew T Sornborger, and Patrick J Coles. "Quantum-assisted quantum compiling". Quantum **3**, 140 (2019).

[13] Matthias C Caro, Hsin-Yuan Huang, M Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J Coles. "Generalization in quantum machine learning from few training data". Nature Communications**13** (2022).

[14] Matthias C Caro, Hsin-Yuan Huang, Nicholas Ezzell, Joe Gibbs, Andrew T Sornborger, Lukasz Cincio, Patrick J Coles, and Zoë Holmes. "Out-of-distribution generalization for learning quantum dynamics". Nature Communications **14**, 3751 (2023).

[15] Joe Gibbs, Zoe Holmes, Matthias C Caro, Nicholas Ezzell, Hsin-Yuan Huang, Lukasz Cincio, Andrew T Sornborger, and Patrick J Coles. "Dynamical simulation via quantum machine learning with provable generalization". Physical Review Research **6**, 013241 (2024).

[16] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. "Barren plateaus in quantum neural network training landscapes". Nature Communications **9**, 1–6 (2018).

[17] Martin Larocca, Supanut Thanasilp, Samson Wang, Kunal Sharma, Jacob Biamonte, Patrick J Coles, Lukasz Cincio, Jarrod R McClean, Zoë Holmes, and M Cerezo. "A review of barren plateaus in variational quantum computing". Nature Reviews Physics **3**, 625–644 (2025).

[18] Eric R Anschuetz and Bobak T Kiani. "Quantum variational algorithms are swamped with traps". Nature Communications **13**, 7760 (2022).

[19] Román Orús. "A practical introduction to tensor networks: Matrix product states and

projected entangled pair states". Annals of Physics **349**, 117–158 (2014).

[20] Feng Pan and Pan Zhang. "Simulation of quantum circuits using the big-batch tensor network method". Physical Review Letters **128**, 030501 (2022).

[21] Joseph Tindall, Matthew Fishman, E. Miles Stoudenmire, and Dries Sels. "Efficient tensor network simulation of IBM's Eagle kicked Ising experiment". PRX Quantum **5**, 010308 (2024).

[22] Ian P McCulloch. "From density-matrix renormalization group to matrix product states". Journal of Statistical Mechanics: Theory and Experiment **2007**, P10014 (2007).

[23] Manuel S Rudolph, Jing Chen, Jacob Miller, Atithi Acharya, and Alejandro Perdomo-Ortiz. "Decomposition of matrix product states into shallow quantum circuits". Quantum Science and Technology **9**, 015012 (2023). url: https://iopscience.iop.org/article/10.1088/2058-9565/ad04e6.

[24] Manuel S Rudolph, Jacob Miller, Danial Motlagh, Jing Chen, Atithi Acharya, and Alejandro Perdomo-Ortiz. "Synergistic pretraining of parametrized quantum circuits via tensor networks". Nature Communications **14**, 8367 (2023).

[25] James Dborin, Fergus Barratt, Vinul Wimalaweera, Lewis Wright, and Andrew G Green. "Matrix product state pre-training for quantum machine learning". Quantum Science and Technology **7**, 035014 (2022).

[26] Francois Jamet, Connor Lenihan, Lachlan P Lindoy, Abhishek Agarwal, Enrico Fontana, Baptiste Anselme Martin, and Ivan Rungger. "Anderson impurity solver integrating tensor network methods with quantum computing" (2023). url: https://arxiv.org/abs/2304.06587.

[27] David Rogerson and Ananda Roy. "Quantum circuit optimization using differentiable programming of tensor network states" (2024). url: https://arxiv.org/abs/2408.12583.

[28] Baptiste Anselme Martin, Thomas Ayral, François Jamet, Marko J Rančić, and Pascal Simon. "Combining matrix product states and noisy quantum computers for quantum simulation". Physical Review A **109**, 062437 (2024).

[29] Sheng-Hsuan Lin, Rohit Dilip, Andrew G Green, Adam Smith, and Frank Pollmann. "Real-and imaginary-time evolution with compressed quantum circuits". PRX Quantum **2**, 010342 (2021).

[30] Niall F Robertson, Bibek Pokharel, Bryce Fuller, Eric Switzer, Oles Shtanko, Mirko Amico, Adam Byrne, Andrea D'Urbano, Salome Hayes-Shuptar, Albert Akhriev, et al. "Tensor network enhanced dynamic multiproduct formulas" (2024). url: https://arxiv.org/abs/2407.17405.

[31] Refik Mansuroglu, Timo Eckstein, Ludwig Nützel, Samuel A Wilkinson, and Michael J Hartmann. "Variational Hamiltonian simulation for translational invariant systems via classical pre-processing". Quantum Science and Technology **8**, 025006 (2023).

[32] Ayse Kotil, Rahul Banerjee, Qunsheng Huang, and Christian B Mendl. "Riemannian quantum circuit optimization for hamiltonian simulation". Journal of Physics A: Mathematical and Theoretical **57**, 135303 (2024).

[33] Luke Causer, Felix Jung, Asimpunya Mitra, Frank Pollmann, and Adam Gammon-Smith. "Scalable simulation of nonequilibrium quantum dynamics via classically optimized unitary circuits". Physical Review Research **6**, 033062 (2024).

[34] Conor Mc Keever and Michael Lubasch. "Classically optimized Hamiltonian simulation". Physical review research **5**, 023146 (2023).

[35] Conor Mc Keever and Michael Lubasch. "Towards adiabatic quantum computing using compressed quantum circuits". PRX Quantum **5**, 020362 (2024).

[36] Maurits SJ Tepaske, Dominik Hahn, and David J Luitz. "Optimal compression of quantum many-body time evolution operators into brickwall circuits". SciPost Physics **14**, 073 (2023).

[37] Maurits SJ Tepaske, David J Luitz, and Dominik Hahn. "Optimal compression of con-

strained quantum time evolution". Physical Review B **109**, 205134 (2024).

[38] Glen Evenbly and Guifré Vidal. "Algorithms for entanglement renormalization". Physical Review B—Condensed Matter and Materials Physics **79**, 144108 (2009).

[39] Alon Kukliansky, Ed Younis, Lukasz Cincio, and Costin Iancu. "Qfactor: A domain-specific optimizer for quantum circuit instantiation". In 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). Volume 1, pages 814–824. IEEE (2023).

[40] Tomonori Shirakawa, Hiroshi Ueda, and Seiji Yunoki. "Automatic quantum circuit encoding of a given arbitrary quantum state" (2021). url: https://arxiv.org/abs/2112.14524.

[41] Alon Kukliansky, Lukasz Cincio, Ed Younis, and Costin Iancu. "Leveraging quantum machine learning generalization to significantly speed-up quantum compilation" (2024). url: https://arxiv.org/abs/2405.12866.

[42] Robert R. Tucci. "An Introduction to Cartan's KAK Decomposition for QC Programmers" (2005). url: http://arxiv.org/abs/quant-ph/0507171.

[43] EM Stoudenmire and Steven R White. "Minimally entangled typical thermal state algorithms". New Journal of Physics **12**, 055026 (2010).

[44] Maarten Van Damme, Jutho Haegeman, Ian McCulloch, and Laurens Vanderstraeten. "Efficient higher-order matrix product operators for time evolution". SciPost Physics **17**, 135 (2024).

[45] Michael P Zaletel, Roger SK Mong, Christoph Karrasch, Joel E Moore, and Frank Pollmann. "Time-evolving a matrix product state with long-ranged interactions". Physical Review B **91**, 165112 (2015).

[46] Ilia A Luchnikov, Mikhail E Krechetov, and Sergey N Filippov. "Riemannian geometry and automatic differentiation for optimization problems of quantum physics and quantum technologies". New Journal of Physics **23**, 073006 (2021).

[47] L. Devos, M. Van Damme, J. Haegeman, and contributors. "Tensoroperations.jl" (2023).

Supplementary Material for
"Deep Circuit Compression for Quantum Dynamics via Tensor Networks"

A Hamiltonians

Here we further detail the Hamiltonians studied in our numerical results in Sec. 3.

1) 1D Transverse-Field Ising model

$$H = \sum_{i=1}^{n-1} Z_i Z_{i+1} + h \sum_{i=1}^{n} X_i \tag{5}$$

We target a 200 qubit chain, with a field strength of $h = 1.0$ for a total evolution time of $t = 2.0$.

2) 1D Hubbard model

$$H = -t_{\text{hop}} \sum_{i,\sigma} \left(c^\dagger_{i,\sigma} c_{i+1,\sigma} + c^\dagger_{i+1,\sigma} c_{i,\sigma} \right) + U \sum_i n_{i,\uparrow} n_{i,\downarrow} \tag{6}$$

We use a staggered representation where the spin up/down orbitals are neighboring on a 1D chain. We target the Hamiltonian with 50 sites (corresponding to 100 qubits), with $U = 4.0$, $t_{\text{hop}} = 1.0$, and total evolution time $t = 0.2$. We assume access to a 1D chain of qubits with nearest-neighbor interactions; the next-nearest neighbor interaction for the hopping terms therefore requires fermionic SWAP gates in the Trotter unitaries.

3) 1D $J_1 - J_2$ Model

$$H = J_1 \sum_{i=1}^{n-1} \bar{\sigma}_i \cdot \bar{\sigma}_{i+1} + J_2 \sum_{i=1}^{n-2} \bar{\sigma}_i \cdot \bar{\sigma}_{i+2} \tag{7}$$

We target a chain of 40 qubits for a total evolution time of $t = 1.0$, with $J_1 = 1.0$ and $J_2 = 0.25$. We assume access to a 1D chain of qubits with nearest-neighbor interactions; the next-nearest neighbor interaction therefore requires SWAP gates in the Trotter unitaries.

4) Transverse-Field Ising model on the Heavy Hex topology

We also compile circuits for the Transverse-Field Ising model on the 2D Heavy-Hex topology found on IBM superconducting chips. For the 2D compilation, we compile the circuit directly onto the same topology, generating a circuit that has the same local 2D interactions as the real quantum device. We target the 52 qubit subset of qubits shown in Fig. A1. We set $h = 0.75$, and $t = 0.5$.

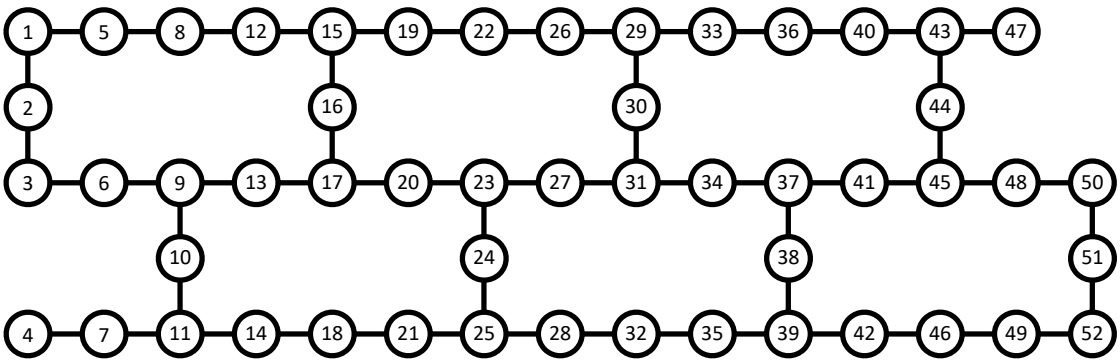


Figure A1: **Heavy Hex Topology.** For the 2D compilation demonstration, the Hamiltonian chosen is the Transverse-Field Ising model with the same connectivity graph as the shown 52 qubit subset of the IBM Heavy-Hex topology. To map the local 2D gates onto the MPO, we use a winding across the shortest dimension, as indicated by the qubit numbering.