Fermionic dynamics on a trapped-ion quantum computer beyond exact classical simulation

Phasecraft[†]

(Dated: October 31, 2025)

Simulation of the time-dynamics of fermionic many-body systems has long been predicted to be one of the key applications of quantum computers. Such simulations – for which classical methods are often inaccurate – are critical to advancing our knowledge and understanding of quantum chemistry and materials, underpinning a wide range of fields, from biochemistry to clean-energy technologies and chemical synthesis. However, the performance of all previous digital quantum simulations has been matched by classical methods, and it has thus far remained unclear whether near-term, intermediate-scale quantum hardware could offer any computational advantage in this area. Here, we implement an efficient quantum simulation algorithm on Quantinuum's System Model H2 trapped-ion quantum computer for the time dynamics of a 56-qubit system that is too complex for exact classical simulation. We focus on the periodic spinful 2D Fermi-Hubbard model and present evidence of spin-charge separation, where the elementary electron's charge and spin decouple. In the limited cases where ground truth is available through exact classical simulation, we find that it agrees with the results we obtain from the quantum device. Employing long-range Wilson operators to study deconfinement of the effective gauge field between spinons and the effective potential between charge carriers, we find behaviour that differs from predictions made by classical tensor network methods. Our results herald the use of quantum computing for simulating strongly correlated electronic systems beyond the capacity of classical computing.

The Fermi-Hubbard model exemplifies the key challenge in many-body physics: accurately modelling systems of interacting particles. Although a highly simplified model of interacting electrons, it contains a rich phase diagram [1], making it an ideal laboratory in which to study phenomena such as spin-charge separation [2–4], the metal-insulator transition [5], and magnetic ordering [6–8]. However, the straightforward definition of the model hides an inherent complexity. The most complex instance of the model whose ground state has been exactly computed numerically is just 17 electrons on 22 sites [9]. On the other hand, several state-of-the-art approximate methods can simulate low-energy states on hundreds of sites at different filling fractions [10], while special-purpose analogue simulators can address larger instances still (between hundreds and thousands of sites) [11, 12], serving as significant tools to probe equilibrium properties of interacting systems.

Simulating dynamical properties of the Fermi-Hubbard model appears to be significantly harder than equilibrium low-energy properties for classical computers. Indeed, the largest reported instances where exact time-dynamics have been simulated classically have 14 sites [13]; although approximate methods reaching 6×5 sites have been demonstrated for certain observables [14], the level of accuracy of these is unknown. Quantum computers are predicted to be able to simulate the Fermi-Hubbard model efficiently, yet all previous digital simulations of the dynamics of the Fermi-Hubbard model on quantum computing hardware are either 1D instances [3, 15, 16], or relatively small 2D instances [17], and can be simulated accurately using tensor network methods. Analogue quantum simulators can simulate dynamical properties for systems on thousands of sites [12, 18–20], but face significant limitations on the initial states they can prepare and the quantities that can be measured at the end of the simulation.

Here we implement an efficient quantum algorithm for simulating the time-dynamics of the 2D Fermi-Hubbard model with periodic boundary conditions on the Quantinuum H2-2 trapped-ion quantum computer. Our system is described by the Hamiltonian

$$H = -J \sum_{\langle i,j \rangle \in \mathcal{L},\sigma} (e^{i\phi_{ij}} c^\dagger_{i,\sigma} c_{j,\sigma} + \text{h.c.}) + U \sum_{i \in \mathcal{L}} n_{i,\uparrow} n_{i,\downarrow}, \tag{1}$$

where $c^\dagger_{i,\sigma}$, $c_{j,\sigma}$ are fermionic creation and annihilation operators at sites i, j, $n_{i,\sigma} = c^\dagger_{i,\sigma} c_{i,\sigma}$ and spin $\sigma = \{\uparrow, \downarrow\}$. The first term describes the hopping of electrons in the lattice $\mathcal{L}$, characterised by the hopping integral $-J$. The pattern of phases ϕ_{ij} corresponds to the insertion of magnetic flux in the system. We study this model on a doubly-periodic lattice (torus) with $|\mathcal{L}| = 7 \times 4$ sites and a π phase flux in the short direction (see Figure 1 top). This is beyond the reach of exact classical simulation in practice, because the Hilbert space explored by the system is $\geq 2^{43}$-dimensional, even taking symmetries into account.

Periodic boundary conditions reduce spurious boundary effects that do not contribute in the thermodynamic limit. This is necessary for approximately capturing translational invariance without using large model sizes, and we anticipate that this will remain relevant in the quantum simulation of such systems. Trapped-ion quantum computers are particularly well-suited to modelling this periodicity because of their connectivity model. The qubits (ions) are freely reconfigurable with a comparatively low cost in error and runtime compared to gate operations, yielding an effective all-to-all connectivity model. This affords the freedom of non-local interactions with minimal cost overhead, allowing for periodic boundaries.

We choose to start with a dimerised state at half-filling (see Figure 1 middle). The initial state corresponds to a dimer covering that is broken at two points, where a holon (i.e., a fully unoccupied site) and a doublon (a doubly occupied site) are located. Our experiment begins with this state, time-evolves for times $t \in [0.1, 2]$ in 0.1 increments (all times are in units of inverse hopping J^{-1}), and measures in the real space occupation basis (computational basis in the qubit representation),

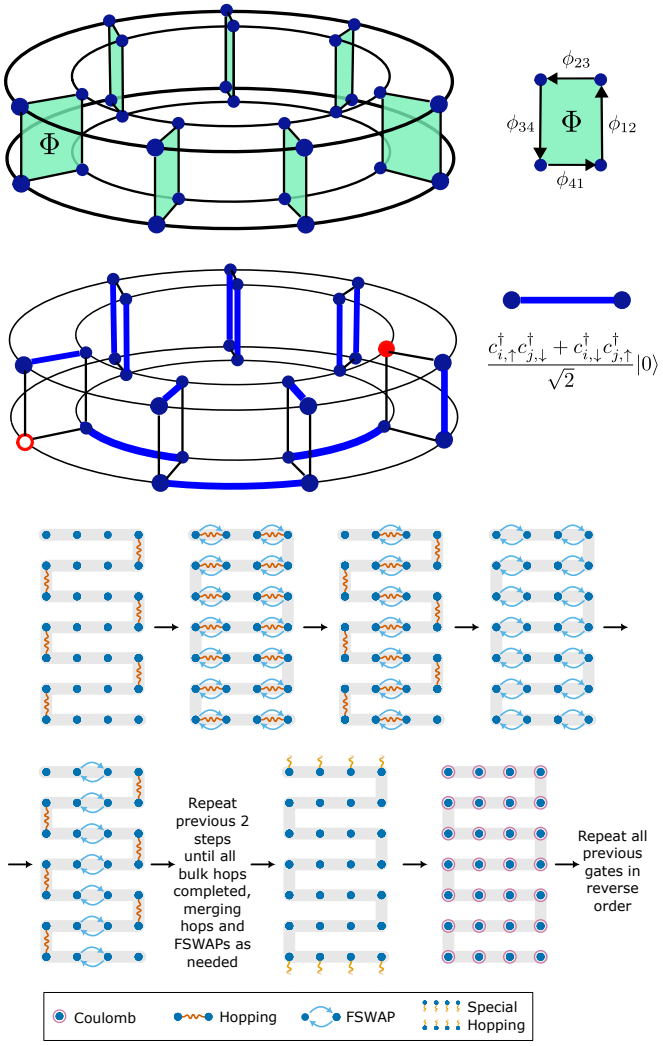


FIG. 1. **Fermi-Hubbard lattice, initial state and map of a spin sector on the quantum circuit. Top,** The Fermi-Hubbard model instance studied in this work is a double-periodic system of size $|\mathcal{L}| = L_x \times L_y = 7 \times 4$ (56 qubits), with $\Phi = \pi$ flux on the small direction. Around each plaquette $\phi_{41} = \pi$, with other phases being zero. Each site (blue circles) can accommodate spin-up and spin-down electrons. **Middle,** The initial state is a dimerised configuration where we place a maximally entangled state, shown as a blue link. Each link corresponds to the $S_z^{\text{total}} = 0$ triplet state. The empty and full red circles correspond to a holon and a doublon, respectively. **Bottom,** Structure of the quantum circuit for a single Trotter step on qubits (blue circles) representing a single spin-sector. The grey line shows the structure of the Jordan-Wigner encoding. "Special hopping" means a hopping term across the vertical boundary. Further details in Section C.

enabling any property constructed from spin-resolved densities to be determined.

We consider the non-interacting ($U/J = 0$) and interacting ($U/J = 4$) regimes. We set $J = 1$ and measure all energies in units of the hopping strength J. The non-interacting Fermi-Hubbard model is well-known to generally be exactly solvable classically, both for simulating time evolution and for ground-state properties. However, in our case, straightforward classical simulability does not apply, even for the non-interacting model. This is because the initial state we use is not a fermionic Gaussian state, i.e. it is not a single Slater determinant in some basis. Instead, it is an example of a fermionic magic state [21, 22]: that is, one of a family of states which promote fermionic linear optics to universal quantum computation. This means that, as one scales up the family of instances considered in this work, we should not expect non-interacting time-dynamics starting with these states to continue to be classically simulable. Nevertheless, for small operator weights, we are able to show that the $U = 0$ regime remains simulable in this setting (Section E).

In this work, we study several signatures of spin-charge separation, including local and global charge and spin correlations, together with extended correlators (Wilson loops and open lines) that diagnose the potential between spin carriers (spinons) and the potential between charge carriers (holons and doublons).

TECHNIQUES

The first step in representing a fermionic model on a quantum computer is to choose a fermionic encoding, which maps fermionic modes to qubits while preserving fermionic anti-symmetry. Here we use the well-known Jordan-Wigner transform, which represents the Fermi-Hubbard model space-optimally, using 2 qubits per site and hence 56 qubits to represent a 7×4 system. The price paid is that certain hopping terms in Equation (1) are mapped to high-weight Pauli strings of the form $(X_iX_j + Y_iY_j)Z_{i+1}\dots Z_{j-1}$. However, we can mitigate this cost using fermionic swap networks [23, 24], which enable sequences of long-range – and hence high-weight – operations in the Jordan-Wigner transform to be implemented efficiently.

We simulate time-dynamics using a second-order Trotter formula,

$$e^{-iHt} \approx \left(\prod_{h\in\mathcal{H}}^{\rightarrow} e^{-ih\frac{t}{2k}} \prod_{o\in\mathcal{O}} e^{-iot/k} \prod_{h\in\mathcal{H}}^{\leftarrow} e^{-ih\frac{t}{2k}} \right)^k, \quad (2)$$

where $\mathcal{H}$ denotes the set of hopping terms, $\mathcal{O}$ denotes the set of onsite terms, and the ordering of the terms in $\mathcal{H}$ is determined by the swap network. We execute $k = 4$ Trotter steps and time-evolve the initial state up to time $t = 2$. Exact classical simulation of low-weight observables for $U = 0$ shows that, in this instance, Trotterised time-dynamics approximates the true dynamics up to small errors – far beyond the regime where theoretical bounds hold (see Section D). As we cannot exactly simulate the interacting case or high-weight observables, we resort to smaller-scale experiments to estimate the level of accuracy of Trotterisation in these cases. Based on this (see Section C 3 d), and on comparison against ground truth in the $U = 0$ setting for low-weight observables, we estimate that (noiseless) Trotterised dynamics are accurate up to time $t \approx 1.5$.

The circuit that we implement for initial state preparation and for time evolution is illustrated in Figure 1. We obtain additional gate savings beyond a standard swap network by observing that there is no need to reorder the qubits at the end of the swap network to return to the initial ordering, given that the symmetric structure of a second-order Trotter step naturally undoes any shuffling. In addition, we use the structure of the second-order Trotter formula to merge each layer's final time evolution by hopping terms (which are executed in reverse order) with the next layer's first time evolution by hopping terms in forward order. Further details may be found in Section C.

Previous uses of fermionic swap networks for the Fermi-Hubbard model have focused on open boundary conditions [23–25]. Periodic boundary conditions in the short direction of the lattice can be implemented without any increase in circuit complexity, because all modes pass each other at some point during the fermionic swap network, and hopping terms can be merged with fermionic swaps without any additional cost. Furthermore, periodic boundary conditions in the long direction can also be implemented at little additional cost. This is because for states of fixed parity in each spin sector, the parity operator Z acts as a constant, implying that we can replace the long Z strings that would extend between two distant qubits with Z strings on the complementary set of qubits. Some remaining parity corrections need to be made, which can be implemented with a small number of additional controlled-Z gates.

The quantum circuits we execute contain at most 2,415 two-qubit gates and 4,627 one-qubit gates. The two-qubit gates are CPHASE gates with varying angles, which are native gates on the device, up to one-qubit gates. See Section C for a further description of gate decomposition, breakdown of gate counts and details of optimisations applied.

Error mitigation

We implement three main techniques to improve the quality of our experimental results (see Section D for further details and validation).

The first of these is an error suppression technique known as *Pauli pseudo-twirling* [26, 27]. Pseudo-twirling is a variant of the well-known twirling technique, which is based on the idea that the primary source of error on many quantum circuit platforms – and in particular on ion traps – is two-qubit gates. Systematic errors on these gates can be reduced and converted into incoherent errors by conjugating each gate with operators that commute with them.

The second technique is an error mitigation method known as *Training with Fermionic Linear Optics (TFLO)* [28]. This method uses the fact that quantum circuits consisting solely of so-called fermionic linear optics (FLO) operations can be efficiently simulated classically. This allows data sets of noisy and exact observable values to be prepared, enabling the inference of a map between exact and noisy data, thus allowing the effect of noise on a given observable to be inverted. This map can then be applied to experimental data, which is not classically simulable, expecting the error behaviour to be similar. Time evolution of the Fermi-Hubbard model is particularly well suited to the TFLO technique, since the quantum circuit for simulating time dynamics is FLO in the case $U = 0$. As the initial state we consider is not a Gaussian state, standard classical simulation techniques [29] do not apply. We are nevertheless able to develop efficient classical algorithms for computing low-weight observables for the output of our experiment (see Section E), allowing these to be computed exactly in seconds. We expect that high-weight observables (and also sampling from measurement outcomes) will require exponential cost, albeit lower than in the $U = 4$ case. Thus, to mitigate errors in high-weight observables, we use an alternative technique where training data is produced using tensor network techniques for short times, which are expected to be accurate in that regime.

The final post-processing technique we use is *Gaussian process regression (GPR)* [30, 31]. This technique enables us to obtain meaningful results from a very small number of shots per data point computed: only 160 shots per point, made up of 16 pseudo-twirled instances, with 10 shots for each. GPR produces estimates based on the assumption that each experimental value is sampled from a Gaussian distribution, with means that are correlated and the level of correlation depending on the distance between the parameters. Since the Gaussian assumption is not strictly satisfied, we cross-validate the GPR method using a particle filter (see Section D 7).

Classical simulation

Our quantum circuits on 56 qubits are beyond the capacity of direct classical statevector simulation. However, it is also necessary to consider more advanced classical simulation techniques, as these can sometimes simulate surprisingly large-scale and complex quantum computations. Here we considered multiple such techniques: direct tensor contraction; time-evolving a matrix product state (MPS) via the time-dependent variational principle (TDVP) method [32]; fermionic matrix product states [33]; and Majorana propagation [34]. We found that the computational resources required by direct tensor contraction scale poorly with the lattice size, with direct contraction already ruled out for a 6×4 lattice (see Section G 2). We evaluated several approximate tensor network techniques for simulating the Fermi-Hubbard model and found that using TDVP on an MPS ansatz performed most reliably, so we report its results below alongside the experimental results, together with those of Majorana propagation. See Section G for a description of our simulations and an overview of alternative implementations considered.

As an overall test of how well our error-mitigated experiment fared against TDVP, we use cross-entropy benchmarking to compare the outputs of each of the experiment and TDVP against the exactly simulable exact distribution for $U = 0$. We find (see Section H) that the experiment achieves higher accuracy than simulation through TDVP with respect to this metric. More concretely, we observe linear cross-entropy benchmarking fidelities between one and five percent after error mit-

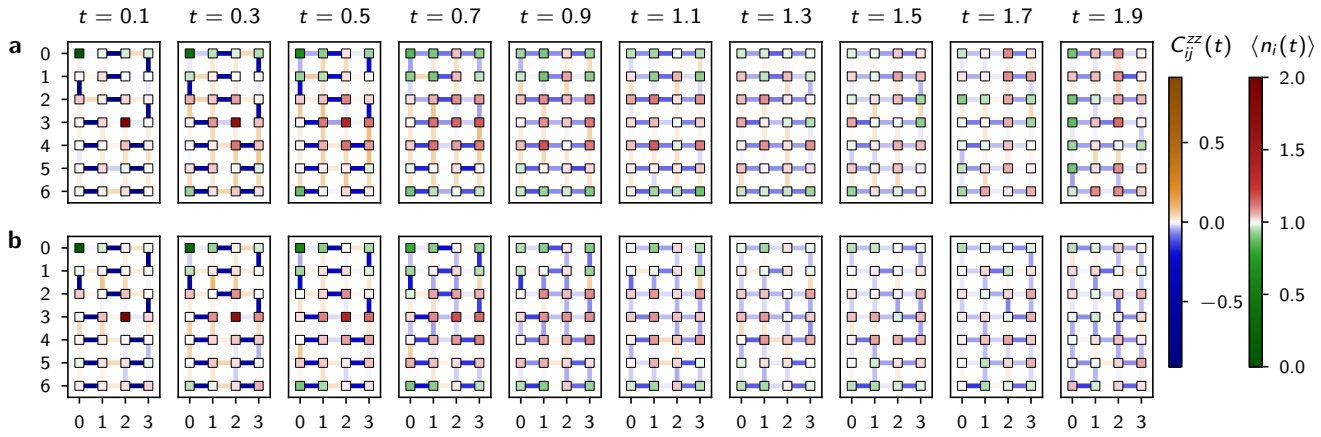


FIG. 2. **Evolution of local charge, and spin correlations.** Local charge density $\langle n_i(t)\rangle$ (squares as sites) and spin (connected) correlation function $C_{ij}^{zz}(t)$ between nearest-neighbours (represented by links) as a function of time for **a,** $U = 0.0$ and **b,** $U = 4.0$ for the error mitigated (TFLO + GPR) experimental data. In the charge sector, we see diffusion from the initial holon-doublon configuration towards the uniform state. In the free case **(a)**, the charges develop a charge-density profile oscillating in the y direction, while in the interacting case **(b)**, the charge profile is more disordered. In the spin sector, the initial triplet configuration takes $\sim t = 0.7$ to melt, leaving behind a residual antiferromagnetic correlation, greater for the interacting case (see also Figure 3).

igation, which includes Trotter errors, as compared with ideal, continuous FLO.

RESULTS

We begin by using our simulation algorithm to obtain an overall picture of the time evolution of our Fermi-Hubbard model instance. Figure 2 shows the evolution of charge densities $\langle n_i(t)\rangle = \langle n_{i\uparrow}(t)\rangle + \langle n_{i\downarrow}(t)\rangle$ and neighbouring spin (connected) correlation

$$C_{ij}^{zz}(t) = 4\left(\langle S_i^z(t)S_j^z(t)\rangle - \langle S_i^z(t)\rangle\langle S_j^z(t)\rangle\right), \qquad (3)$$

over time for $U = 0$ (panel a) and $U = 4$ (panel b). Here $S_i^z = (n_{i,\uparrow} - n_{i,\downarrow})/2$. We observe that the initial charge configuration diffuses radially, while the initial spin correlations vanish. While in both cases the charge tends towards the homogeneous state, several differences emerge. For $U = 0$ at large times ($t = 1.9$), the charge arranges into charge-density waves in the short direction of the lattice (see also Section B). Note as well a slight antiferromagnetic tendency in the y direction, which can be attributed to the initial triplet configuration (see Figure 1), while no such charge-density wave appears for $U = 4$. The presence of interactions also slightly favours overall antiferromagnetic order at late times, in a pattern that is completely disordered with respect to the initial template of correlations.

At half-filling, the charge carriers are doublons and holons, which can emerge without constraints when the interaction vanishes, as the spin sectors are totally decoupled in this regime. In contrast, for large interactions, the doublons have to overcome an energy $\sim U$ to appear. We can directly inspect the proliferation of doublons (and holons), captured by the observable

$$N_{\text{doublons}} \coloneqq \sum_{i\in\mathcal{L}} n_{i,\uparrow}n_{i,\downarrow} \qquad (4)$$

in Figure 3a, which equilibrate to a value that decreases as the interaction increases. Note that for $U = 4$, we observe that the creation of holon-doublon pairs is not accurately modeled by the t-J model [35], which assumes a constant number of pairs, in contrast to the proliferation seen in Figure 3. We also include the results from the Trotterised circuit (for $U = 0$) to show the accumulation of Trotter error at large times. Note that this error is smaller than the statistical uncertainty due to sampling measurement results.

We capture the magnetic ordering through the nearest-neighbour triplet density

$$n_{\text{triplets}}(t) \coloneqq \frac{2}{L_x L_y}\sum_{\langle i,j\rangle} C_{ij}^{zz}(t), \qquad (5)$$

shown in Figure 3b. Initially, the order melts, rapidly approaching zero. The late-time behaviour of the signal shows equilibration with slightly stronger antiferromagnetic tendency in the presence of interactions compared with the non-interacting case. For both the number of doublons and the triplet density, we note that the experimental results differ from the tensor network simulations at times $t \gtrsim 0.5$ for $U = 4$. For the doublon number, we observe that, while Majorana propagation predicts a smaller number of doublons than the experiment, the trends match, including the late-time increase in doublon population. Likewise, the number of triplets exhibits agreement between Majorana propagation and the experimental data, within small experimental errors. For $U = 4$, it is not possible to exactly determine the effect of Trotter error, and it is possible that the late time increase in

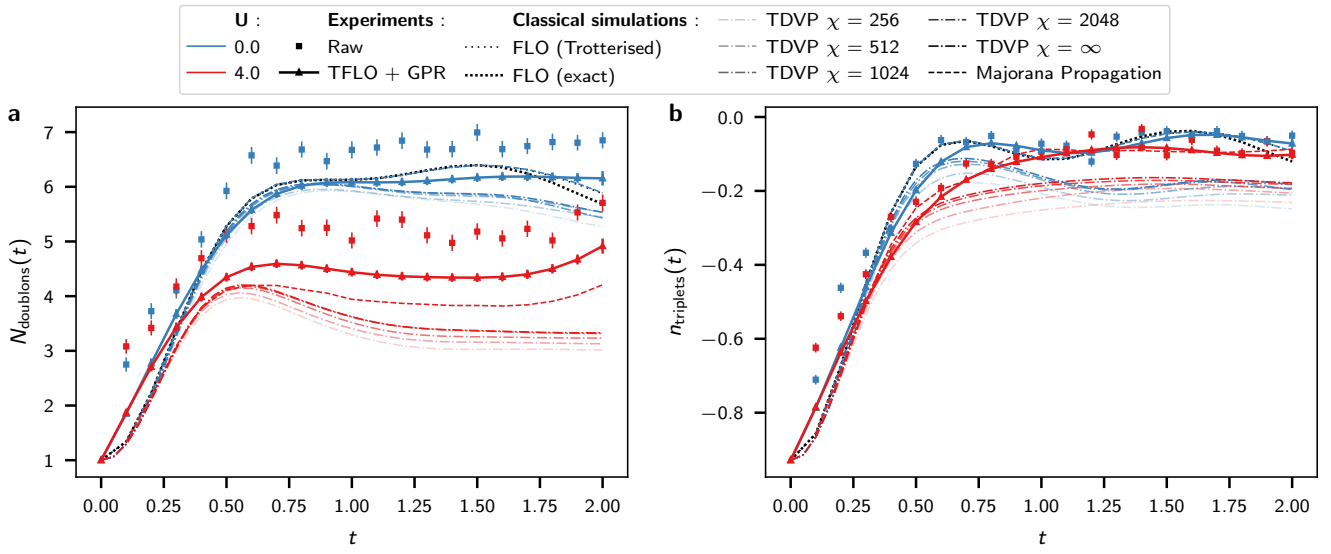


FIG. 3. **Evolution of global doublon charge and magnetic correlations. a,** Total number of doublons and **b,** triplet density as a function of time. While the initial state has a fixed energy density, it is not an equilibrium state. In particular, the number of doublons evolves nontrivially with time (panel **a**). By mitigating the raw data with TFLO + GPR, we recover a signal that aligns with the ground truth, here represented by the exact FLO simulation of the Trotterised circuit for $U = 0$ and by Majorana propagation at $U = 4$. The TFLO + GPR curve has been obtained by symmetrising over doublons and holons (since $N_{\text{doublons}}(t) = N_{\text{holons}}(t)$) via $N^{\text{sym}}_{\text{doublons}} = (N_{\text{doublons}} + N_{\text{holons}})/2$. Since N_{doublons} and N_{holons} are not independent, in computing the error bars of $N^{\text{sym}}_{\text{doublons}}$, we assumed perfectly correlated errors (i.e., $\sigma^{\text{sym}}_{\text{doublon}} = \sigma_{\text{doublon}} + \sigma_{\text{holon}}$). Note that for N_{doublons}, the results from Majorana propagation are equally distant from the TDVP simulations as from the experimental data, but the experiment is capturing the late increase in doublon number. From panel **b**, we can see that while the initial melting of the antiferromagnetic order is slower for the interacting system, the spin ordering is lost essentially at $t \gtrsim 0.5$. In n_{triplets}, the agreement between the mitigated experimental signal and the results from Majorana propagation is within the error bars, while the results from tensor network TDVP simulations converge to a smaller negative value, indicating more antiferromagnetic order in the tensor network simulations than expected for this state. In both panels, error bars indicate one standard deviation of the mean.

the doublon number is an artifact of Trotter error. However, small-scale simulations (Section C 3 d) suggest that Trotter error may be substantially lower than worst-case commutator bounds would indicate.

Spin-charge separation and deconfinement

The average energy of the initial state is comparable to an equilibrium state with temperature $T \sim 3J$ (see Section A). A Fermi-Hubbard system at half-filling with that temperature is expected to host fractionalised quasiparticles [36, 37]. We study the fractionalisation of the constituent electron into a fermionic particle carrying spin 1/2 (spinon) and a boson carrying the charge (holon/doublon). We investigate this by expressing the Fermi-Hubbard model in terms of a modified Kotliar-Ruckenstein representation [38] of the fermion operator, which ultimately leads to a dual $U(1)$ gauge theory description of the Fermi-Hubbard Hamiltonian (see Section B 1 for details). In this dual picture, spinons interact with an emergent gauge field described by the motion of the doublons/holons. Constructing gauge-invariant operators in terms of this gauge field, we find the following Wilson loop operators

$$\mathcal{W}_d(\mathcal{C}) \coloneqq \prod_{j\in\mathcal{C}} (1 - n_{j,\uparrow} n_{j,\downarrow}), \tag{6}$$

where $\mathcal{C}$ is a closed loop of sites in the lattice. The expectation value of these Wilson loops can be used to diagnose the confinement/deconfinement transition of the gauge field mediating the interaction between spinons. Usually, the Wilson loop that detects confinement/deconfinement is taken in a space-time loop (see also Figure 4a), which can be directly understood as the Euclidean action of a potential between two charges [39]. In our case, we measure the Wilson loop at fixed space-like regions (i.e., for a fixed time): see, e.g., Figure 4b. These two descriptions are related, but are not equal. It has been shown [40] that the fixed time (or horizontal) Wilson lines are not a faithful descriptor of confinement, as they can show an area law in the deconfined phase. As discussed in [40], if the system in spatial dimension $d-1$ is confined, then the horizontal Wilson lines will show an area law, regardless of the true nature of the system in d dimensions. In our setting, we can still use the horizontal Wilson lines as descriptors of the confinement/deconfinement transition because in one dimension, the Fermi-Hubbard model is deconfined for any interaction parameter, thus avoiding the constraint.

In Figure 5, we plot the behaviour of area vs perimeter in

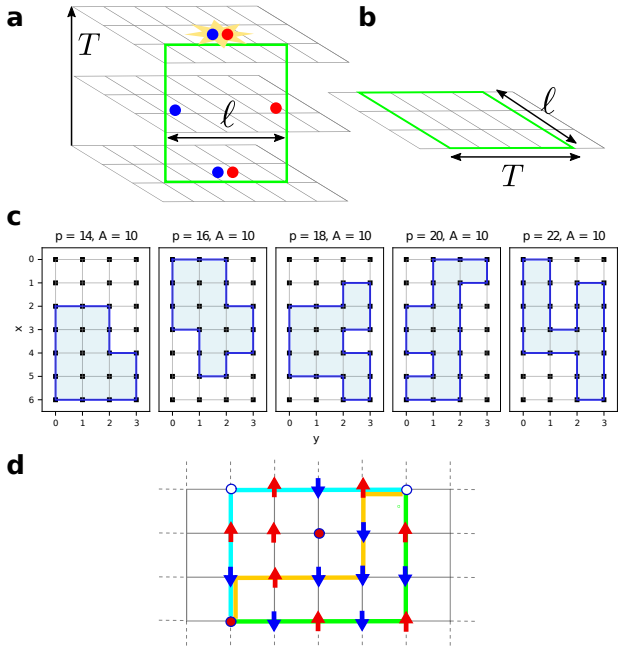


FIG. 4. **Wilson open lines.** **a,** In (lattice) gauge theory, the Wilson loops that serve as order parameters of the confinement/deconfinement transition are usually taken in space-time, here sketched as a green rectangle spanning a region of size ℓ in space and (Euclidean) time T. **b,** In our analysis, we fix the time slice and analyse the expectation of Wilson lines as a function of perimeter or area for different real times. **c,** Examples of some of the closed Wilson loops considered in the computation of $W_d(\mathcal{C})$ in Figure 5 below for area $A = 10$ and varying perimeters p. **d,** Different paths in the computation of holon-doublon Wilson lines, located in opposite corners and represented by a white and red circle, respectively. (Green) This path has a full Néel order of the spinons in between. (Yellow) where the Néel order is broken. (Blue) Path where holons are present.

Wilson loops. Here, we fix the area (perimeter) and study the expectation value of this observable as a function of the perimeter (area). An illustration of the type of loops chosen for fixed area and varying perimeter is shown in Figure 4c.

We show both raw data and data mitigated using the short-time result from the MPS-based TDVP simulation. The latter is discussed in more detail in Section D 3. The values reported by the error-mitigated experiment diverge from tensor network results, for the largest available times t. The observables corresponding to the largest Wilson loops we consider have over a million terms spanning all Majorana weight sectors up to weight 44. Given the failure of Majorana propagation to accurately simulate the open Wilson lines of Equation (7) below, which are much easier with respect to the number of terms involved, we do not expect it to be effective for the closed Wilson loops of Equation (6). This suggests that the confinement/deconfinement transition witnessed by $\mathcal{W}_d$ cannot be captured by Majorana propagation.

We observe that $\mathcal{W}_d$ scales with the perimeter of the loop (keeping the area fixed) for both the non-interacting and the interacting cases. As $\mathcal{W}_d$ is a Wilson loop constructed from *link* operators mediating the interaction between spinons, this signals their deconfinement. To probe spin-charge separation, we need to compare the effective potential between charge carriers with the potential between spinons. The operator that measures the potential between charge carriers is an open Wilson line motivated by the dual $U(1)$ lattice gauge theory description of the model (see Section B 1) and takes the form

$$V_{\alpha\beta}(M) \coloneqq \sum_{\substack{\text{sites } i,j:\\ |i-j|=M}} \frac{n_i^\alpha n_j^\beta}{N_{\text{pairs}}} \sum_{\substack{\text{paths } \gamma:\\ \text{from } i \text{ to } j}} \frac{\sum_{(m,n)\in\gamma} S_m^z S_n^z}{N_{\text{paths}}(i,j)} \tag{7}$$

where $\alpha, \beta = \{h, d\}$ and $n_i^d \coloneqq n_{i\uparrow} n_{i\downarrow}$, $n_i^h \coloneqq (1 - n_{i\uparrow})(1 - n_{i\downarrow})$. The outermost sum runs over all pairs of sites with distance M while the innermost sum runs over all pairs of sites (m, n) along a path γ from i to j. Here $N_{\text{paths}}(i, j)$ is the number of paths between a given pair (i, j). An illustration of the type of paths that enter in the computation of the potential $V_{\alpha\beta}$ is shown in Figure 4d. We extract this from the measured shots in the computational basis. In Figure 6 we show the value of this observable as a function of the (Manhattan) distance between a doublon and a holon, obtained from raw experimental data and from sampling the tensor network state at bond dimension $\chi = 2048$. We do not include Majorana propagation results because our simulations failed to converge as we decreased the truncation parameters, even for low times, as detailed in Section F.

In Figure 6 we observe some striking differences between different simulation methods, as well as between interacting and noninteracting models. As time progresses, the experimental values show a clear separation between the $U = 0$ and the $U = 4$ regimes, where $U = 0$ shows no signal of a growing confining potential between a holon and a doublon with distance, as opposed to the interacting case $U = 4$. In contrast, the classical simulation predicts a confining potential even for zero interaction. The difference between the experimental signal and the TDVP results could be attributed to Trotter error, although it appears at times where Trotter error is expected to be small (see Section C 3 d).

Another plausible source of difference between computational methods could be the effective energy density of the states produced by the different methods. The lower production of doublons and higher antiferromagnetic order for mid-to-late times in the classical tensor networks simulations observed in Figure 3 suggest the state produced through TDVP has a smaller energy density than the corresponding one explored by the quantum computer. To illuminate this, we also explore the behaviour of the closed and open Wilson lines as a function of their size for the ground-state of the system, obtained through Density Matrix Renormalisation Group (see Section G 5 for implementation details). This result is shown in Figure 11 in Section B, for both the closed and open Wilson lines. While we observe similar perimeter scaling of $\mathcal{W}_d$ for the ground-state, TDVP and quantum results, the expectation of the open line operators for the Wilson loops in the ground-state differ from the ones obtained experimentally for $U = 4$, while being similar to the results obtained by TDVP. We take this as another signature of the different energy density that the states can achieve. Taking the experimental re-

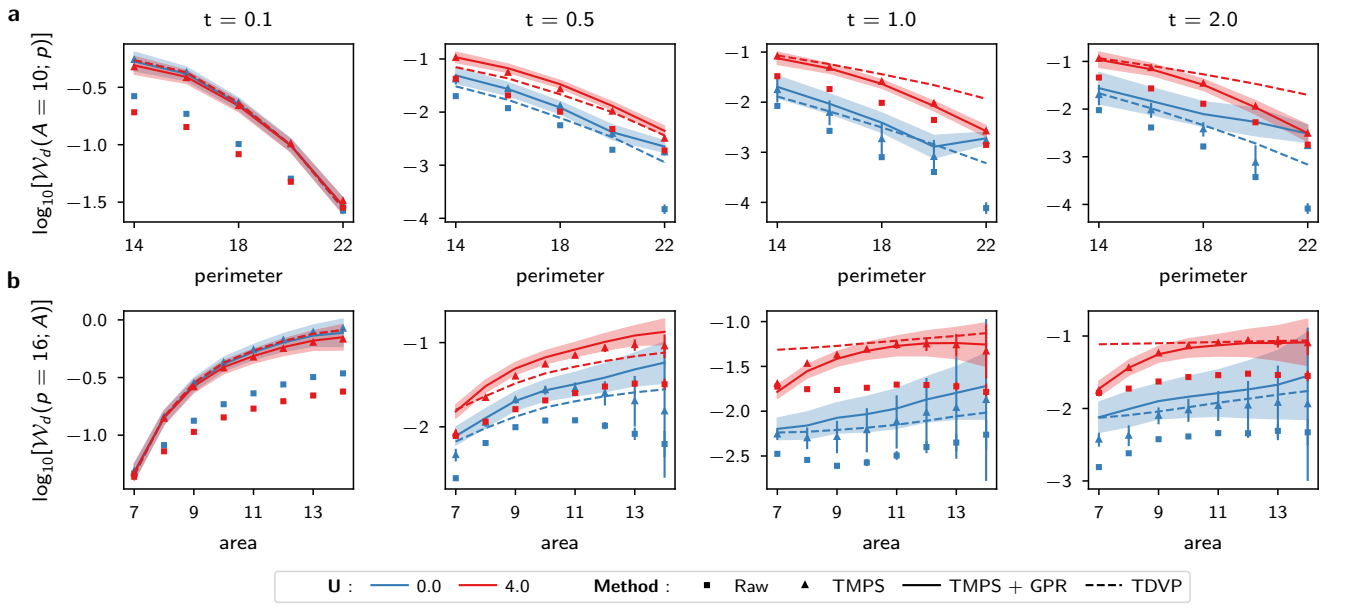


FIG. 5. **Wilson loops as function of area and perimeter.** Wilson loops as defined in Equation (6) with: **a,** fixed area, varying perimeter; **b,** fixed perimeter, varying area. The square points (Raw) indicate the raw device signal, triangular points (TMPS) indicate device signal corrected by the training with MPS (TMPS) procedure described in Section D 3, and the solid lines (TMPS+GPR) are the curves produced by GPR applied to the triangular points, in both time and perimeter/area directions. The dashed line is produced by sampling $10,000$ shots from the MPS obtained by TDVP with $\chi = 2048$. TMPS data have been obtained by training with TDVP data up to $t = 0.3$. Here, we observe the scaling of space-like Wilson lines with the perimeter. This operator is the order parameter of the $U(1)$ gauge field mediating the interactions between spinons. The nontrivial scaling with perimeter signals deconfinement.

sults at face value, the closed Wilson lines show deconfinement of the gauge field acting between spinons, while the open Wilson lines between holons and doublons show a signature of a growing confining potential between them, which are the charge carriers. This signals spin-charge separation for the state at this energy density.

OUTLOOK

We have shown that it is possible to analyse the time-dynamics of a periodic materials model that is beyond the capacity of exact classical simulation, using an algorithm running on a digital quantum computer and executing over 2000 two-qubit quantum gates. In some cases, the results we obtain diverge both quantitatively and qualitatively from those obtained through all the advanced approximate classical simulation techniques available to us. In particular, we provide evidence that – at the energy densities explored – the Fermi-Hubbard model exhibits spin-charge separation by examining the deconfinement of the field mediating the spinon interactions and the potential between charge carriers. We have validated our results by comparing them against exact classical simulations in the interaction-free case, and approximate classical simulations in the regimes where they are expected to be accurate. Beyond the observables we consider, we believe that many additional physical properties of the Fermi-Hubbard model can be obtained from the data we collected.

To determine whether a time-dynamics simulation outmatches the capabilities of classical computers, we require that the following desiderata are satisfied: (i) the results obtained from the quantum algorithm agree with exact classical methods (here, FLO) in a classically tractable regime (here, $U = 0$); (ii) approximate classical methods (here, MPS-based TDVP simulations) do not agree with exact classical methods in a classically tractable regime; (iii) the results from the quantum algorithm are not reproducible by approximate classical methods in a regime that is intractable for exact classical methods (here, $U = 4$). The Wilson lines/loops discussed here fulfil these criteria, with further evidence provided by cross-entropy benchmarking. This heralds the use of quantum computing for simulating strongly-correlated electronic systems beyond the capacity of classical computing. The boundary of classical tractability is not final and changes with the development of more specialised and efficient classical methods, so we invite the community to attempt to replicate our findings using classical techniques.

We used a quantum simulation algorithm based on a second-order Trotter formula, which was found to be effective in a recent study [41]. Future work could explore more efficient algorithms. Ongoing improvements in quantum gate fidelities will enable access to longer evolution times and larger system sizes, elucidating physics such as the role of charge fluctuations or external time-dependent pulses in the development of superconducting order [42]. We expect that the path from here to full fault tolerance will be full of discoveries,

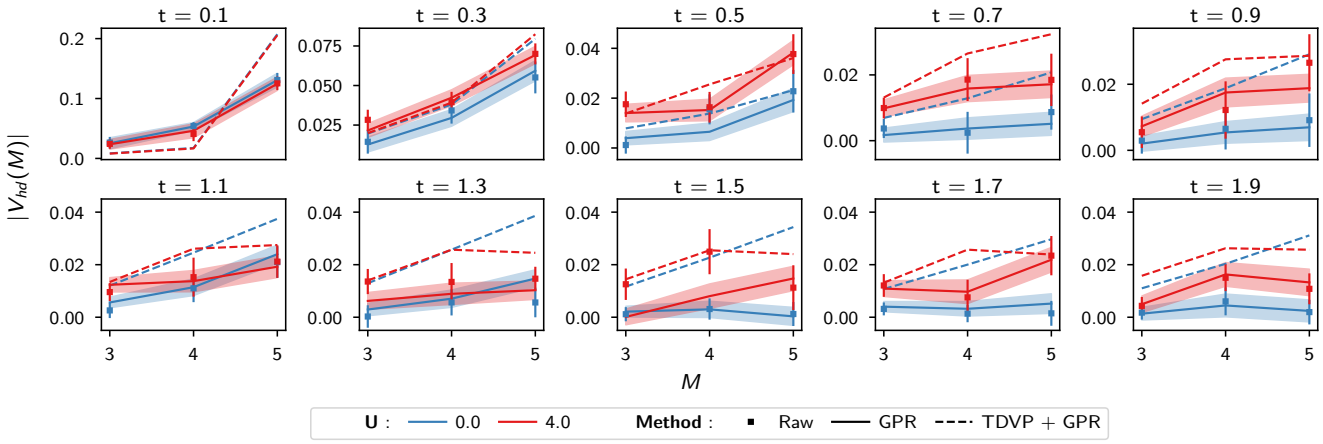


FIG. 6. **Wilson lines and their dependence with distance and time.** Expectation of the (absolute value of the) open Wilson line $V_{hd}(M)$ as a function of time for different Manhattan distances M, as obtained from 10000 samples of the MPS state produced by a TDVP simulation with $\chi = 2048$ (dashed line) and the experimental data (continuous line). The expectation value of this operator measures the effective interaction potential between a doublon and a holon. Increasing interaction with distance signals confinement. Both experimental and TDVP data have been smoothed using GPR.

with error correction playing an increasingly fundamental role as quantum gate fidelities improve and the sizes of simulated systems expand.

ACKNOWLEDGMENTS

We wish to acknowledge the contribution of Eli Chertkov, Henrik Dreyer, Michael Foss-Feig and the rest of the Quantinuum team for their guidance in developing this experiment. We would also like to thank Vanya Eccles, Callum MacPherson, Sam White and the TKET support team for technical assistance throughout the experiment. We thank Andrew Childs for helpful comments on a previous version. This work received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 817581).

† Phasecraft

Faisal Alam[1], Jan Lukas Bosse[1], Ieva Čepaitė[1], Adrian Chapman[2], Laura Clinton[1], Marcos Crichigno[2], Elizabeth Crosson[2], Toby Cubitt[1,3], Charles Derby[1], Oliver Dowinton[1], Paul K. Faehrmann[1,4], Steve Flammia[2,5], Brian Flynn[1], Filippo Maria Gambetta[1], Raúl García-Patrón[1,6], Max Hunter-Gordon[1], Glenn Jones[1], Abhishek Khedkar[1], Joel Klassen[1], Michael Kreshchuk[2], Edward Harry McMullan[1], Lana Mineh[1], Ashley Montanaro[1,7], Caterina Mora[1], John J. L. Morton[1,8], Dhrumil Patel[2,5], Pete Rolph[1], Raul A. Santos[1], James R. Seddon[1], Evan Sheridan[1], Wilfrid Somogyi[1], Marika Svensson[1], Niam Vaishnav[1], Sabrina Yue Wang[1], Gethin Wright[1]

[1] Phasecraft Ltd, London, UK
[2] Phasecraft Inc, Washington DC, USA
[3] Department of Computer Science, University College London, UK
[4] Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany
[5] Department of Computer Science, Virginia Tech, USA
[6] School of Informatics, QSL, University of Edinburgh, UK
[7] University of Bristol, UK
[8] Department of Electrical and Electronic Engineering, UCL, London, UK

Author contributions— JK, RAS, AM conceived of the project and provided leadership and oversight. JK, CD, RAS, BF, SYW designed the quantum circuits. BF, SYW, JLB, EHM implemented and executed the circuits on hardware. MH, JK, JLB, AM, SYW, EC, FMG developed and applied the error mitigation techniques. BF, SYW, RAS, FMG, AK, PR, LM performed the tensor network simulations. SYW, JLB, BF, AM performed the FLO simulations. JLB implemented the Majorana propagation. LC, JK performed the Trotter error analysis. EC, TC, SYW, MHG, MS, BF, NV performed the XEB analysis. RAS, SYW, FMG, MK, AM undertook the physics analysis. BF, MHG, EHM, JLB, SYW, FMG developed the underpinning algorithmic infrastructure used for data analysis. All authors wrote and revised the manuscript and the Supplementary Information.

Appendix A: The Fermi-Hubbard model

The Fermi-Hubbard (FH) model represents a paradigmatic system in the study of strongly correlated materials. It is described by the single-band Hamiltonian

$$H = -J \sum_{\langle i,j\rangle \in \mathcal{L},\sigma} (e^{i\phi_{ij}} c^\dagger_{i,\sigma} c_{j,\sigma} + \text{h.c.}) + U \sum_{i\in\mathcal{L}} n_{i,\uparrow} n_{i,\downarrow}, \tag{A1}$$

where $c_{j,\sigma}$ ($c^\dagger_{j,\sigma}$) is a fermionic destruction (creation) operator at site j and spin $\sigma = \{\uparrow,\downarrow\}$ satisfying the canonical anti-commutation rules $\{c_{j,\sigma}, c^\dagger_{k,\sigma'}\} = \delta_{ij}\delta_{\sigma\sigma'}$. The density operator is given by $n_{j,\sigma} = c^\dagger_{j,\sigma} c_{j,\sigma}$. The first term describes the hopping of electrons in the lattice $\mathcal{L}$, characterized by the hopping integral $-J$. The pattern of phases ϕ_{ij} corresponds to the insertion of magnetic flux in the system. In this model, interactions appear whenever a site is doubly occupied. This is modeled by the second term in Equation (A1). In our experimental setup, we considered a rectangular $L_x \times L_y = 7 \times 4$ lattice with double periodic boundary conditions and nearest neighbour hopping. As shown schematically in Figure 1 in the main text (top), we added a small π flux in the smaller direction (shown in light green), corresponding to magnetic field inside the tours. In bigger systems, the effect of this flux depends strongly on how the system is enlarged. Fixing L_y and increasing $L_x \to \infty$, the system corresponds to an L_y-legged cylinder with flux. Using DMRG and field theory techniques, the low-energy description of this model can be understood qualitatively [43–46]. At half filling, Umklapp terms open a charge gap in the presence of interactions. A spin gap also opens for even L_y, while the spin sector remains gapless for odd L_y. This is a direct manifestation of the Haldane conjecture [47, 48]. In contrast, extending the system by repeating its unit cell in three dimensions corresponds to studying a stack of Dirac semimetals, a system that is radically different to the standard Fermi-Hubbard model [49, 50]. In our particular scenario, adding a π magnetic flux changes the eigenstates and eigenenergies of Equation (A1) compared to the case of zero flux, without breaking time-reversal symmetry. At zero interaction and half-filling, the ground-state goes from being unique ($\Phi = 0$) to being 36-fold degenerate ($\Phi = \pi$). More importantly for the following analysis, the Aharonov-Bohm flux substantially changes the time-dynamics evolution of the electrons. At π-flux, the paths connecting opposite sites on a plaquette pierced by the magnetic field interfere destructively, as opposed to constructively in the zero flux case.

While at finite temperatures the Mermin-Wagner theorem [51] prevents the existence of antiferromagnetic long-range order (AFLRO) in the half-filled FH model, it is widely accepted that at exactly zero temperature and high interaction strengths the half-filled FH system possesses AFLRO [52–56].

The low-energy description of the half-filled FH model maps to a 2D Heisenberg antiferromagnet, with gapless spin excitations. On the other hand, neutron scattering [57] and ARPES [58] experiments reveal a systematic tendency of spin-gap formation above the critical superconducting temperature T_c in a wide range of cuprates. The formation of a spin-gap and the emergence of superconductivity are expected to be related. A spin-gap implies that low-energy spin fluctuations are suppressed, thus reducing the effect of scattering channels that can break Cooper pairs, strengthening $d-$wave unconventional superconductivity [59].

This opens the question of how this physics can appear in the FH model, where the spin excitations are gapless. One possibility is that the system spontaneously dimerizes. In this scenario, the reduction in magnetic exchange energy is larger than the increase in energy from an elastic distortion in the lattice.

We study the melting of this dimerized state by performing a quench from the dimerized state Figure 1 (bottom) by time evolving it with the Hamiltonian defined in Equation (A1). The dimerization pattern that we chose allows us to reduce the circuit complexity of preparing the initial state, as it follows the Jordan-Wigner line discussed in Section C 1. This dimerization is broken at two sites to put a holon and a doublon, shown as an open and full circle in Figure 1 (bottom).

The average energy of the initial state is $\langle\Psi|H|\Psi\rangle = U$, and its average density $n = N_{el}/L_xL_y = 1$, where N_{el} is the total number of electrons. Using the equation of state for the FH model [60], this energy density corresponds to that of a system with approximate temperature $T \sim 3J$. In the phase diagram, a system with this temperature and filling lies above the dome where antiferromagnetic order is dominant and right in the intersection of the strange metal and pseudogap regions.

At half filling, the charge carriers are doublons and holons, which can emerge without constraints when the interaction vanishes, as the spin sectors are totally decoupled in this regime. In contrast, for large interactions, the doublons have to overcome an energy $\sim U$ to appear. As we will further discuss in the next section, introducing a basis where the doublons, holons and spinons (the carriers of spin but no charge) are explicit, we map the FH model into an equivalent $U(1)$ lattice gauge theory.

Appendix B: Additional results

The initial state and the Hamiltonian are invariant under several symmetries. A particularly important one is spin-reflection $Rc_{i,\uparrow}R^\dagger = c_{i,\downarrow}$. The presence of this symmetry immediately separates the spin and charge responses. The spin operator $S_i^z = \frac{1}{2}(n_{i,\uparrow} - n_{i,\downarrow})$ is odd under the symmetry action as $RS_i^z(t)R^\dagger = -S_i^z(t)$, while the initial state is even under this transformation. This implies that the expectation value of any operator consisting of an odd number of spin operators vanishes identically for all times. We discuss the use of this symmetry for error mitigation of observables in Section D 2. On the other hand the charge density $n_i = n_{i,\uparrow} + n_{i,\downarrow}$ evolves non-trivially. As shown in Figure 2, the initial inhomogeneous charge configuration dilutes towards the uniform density state. To quantitatively capture this melting, and the spatial movement of the charges, in Figure 7 we include the results of the spatial Fourier transform of the charge density $\tilde{n}(\boldsymbol{k}) \coloneqq \sum_r e^{i\boldsymbol{k}\cdot\boldsymbol{r}} n_{\boldsymbol{r}} - L_x L_y$ at different time steps, where we have substracted the zero momentum component. Note that initially (up to $t \sim 0.8$), the evolution of the charge in the interacting and non-interacting cases is very similar, while later times show a divergence in the signal, with late-time dynamics being of charge-density wave type (in the short direction) for $U = 0$ and disordered for $U/J = 4$.

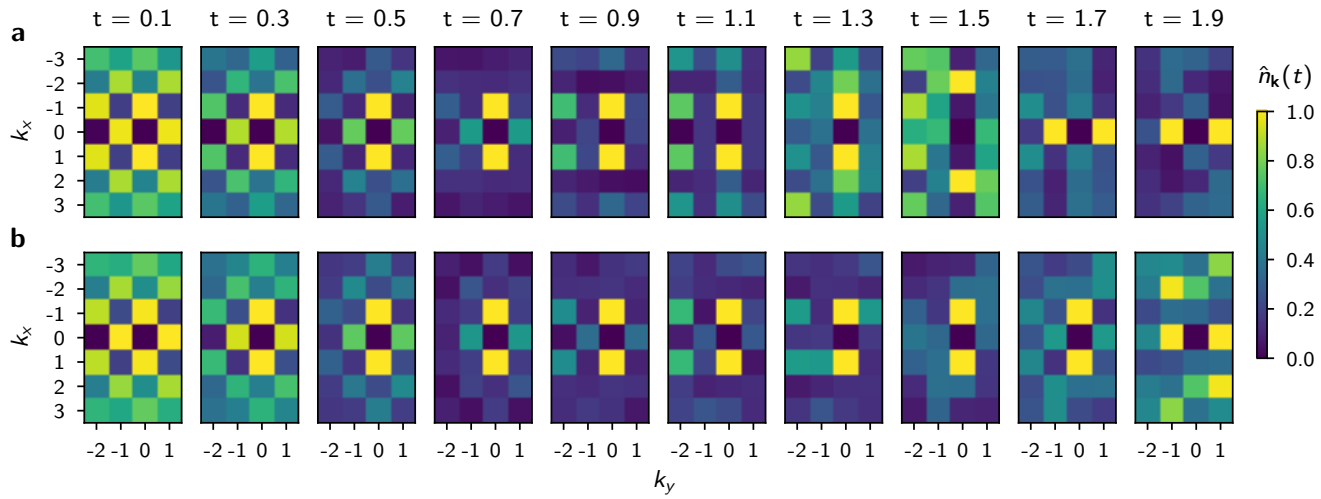


FIG. 7. Normalised space Fourier transform of the mitigated experimental density profiles shown in Figure 2 for a) $U = 0$ and b) $U = 4$. We set the central peak at $(k_x, k_y) = 0$. Note that the stripe charge density manifests here as a sharp peak at $k_x = 0$ for long times in the non-interacting case

The space-time evolution of the charges shows the expected equilibration towards the uniform density state in a way that resembles a fluid. We can gain insight about the melting of the original order, and the effect of interactions by analysing the pair correlation function for charges (see Figure 8 (left))

$$g_c(r,t) = \frac{1}{L_x L_y Z(r)} \sum_{\boldsymbol{x}} \langle n_{\boldsymbol{x}}(t) n_{\boldsymbol{x}+\boldsymbol{r}}(t) \rangle, \tag{B1}$$

and spins (see Figure 8 (right))

$$g_s(r,t) = \frac{1}{L_x L_y Z(r)} \sum_{\boldsymbol{x}} \langle S^z_{\boldsymbol{x}}(t) S^z_{\boldsymbol{x}+\boldsymbol{r}}(t) \rangle, \tag{B2}$$

where $Z(r)$ is the number of sites at a particular (Euclidean) distance r, so in the square lattice $Z(r = 1) = 4, Z(\sqrt{2}) = 4, Z(2) = 4$ and so on. As seen from Figure 8, the system displays a pairs correlations in the spin and charge sector that resemble a liquid at short times. As time progresses, the $g_{s,c}$ flatten, indicating a gas-like behavior. The peaks appearing at the largest Euclidean distances are a result of the radius r wrapping around the system.

At zero interaction, the movement of holon-doublons should distort the antiferromagnetic order of the initial state. As the interaction increases, the exchange interaction between local spin moments penalizes a disruption of the antiferromagnetic order. We see this effect in Figure 2 (right) where the connected spin-correlation function between nearest neighbor sites i, j $C^{zz}_{ij}(t)$, defined in Equation (3) is shown as a function of time. Note that the initial triplet ($m_z^{\text{total}} = 0$) state is an eigenstate with eigenvalue zero of the dimer Hamiltonian for any value of the interaction parameter. Since the total Hamiltonian can be viewed as the union of dimer Hamiltonians in all possible nearest-neighbour bonds, the initial order takes a considerable amount of time

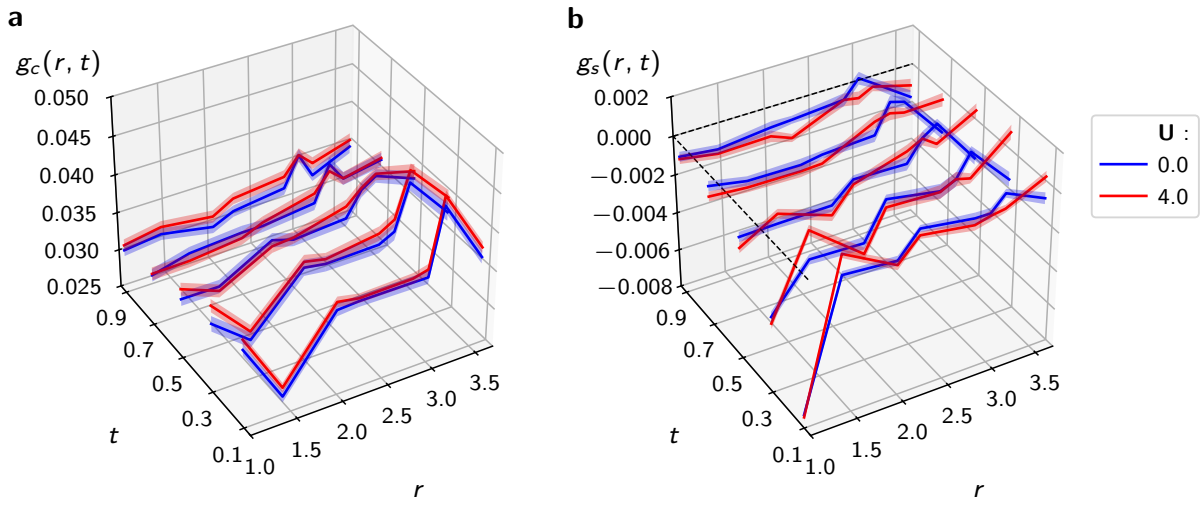


FIG. 8. a) Pair correlation function for charges. (b) Pair correlation function for spins. In both panels, data for TFLO + GPR are shown, with shaded regions corresponding to errors.

$t = 0.4J^{-1}$ to melt. Once the melting has happened, we observe that higher interaction leads to a higher antiferromagnetic order at final times. Summing over all possible links between nearest neighbors, we define the (triplet) density $n_{\text{triplets}}(t)$ in Equation (5). Its behavior as a function of time is shown in Figure 3.

The expectation of a considerably higher antiferromagnetic order in the interacting regime contrasts with the death of triplet correlations shown in Figure 3 (right). This seemingly puzzling behavior can be understood from a microscopic point of view. In all the results above, we observe an interplay between the spin and the charge degree of freedom, for non-zero interaction. In particular, the spin order is dampened by the presence of a non-zero density of mobile charge carriers, which can move and distort the spin alignment of the background. In order to disentangle the effect of the mobile charges from the spin background, we study the fractionalisation of the electron into spin and charge quasiparticles.

1. Fractionalisation

In the one-dimensional FH model, it is possible to show [61] that the fundamental electronic degree of freedom fractionalises into spin and charge modes that propagate independently. The situation in two dimensions is not so sharp. It is believed [36] that the electron fractionalises into spin and charge degrees of freedom above some temperature, while the ground-state is confining, but there is no conclusive evidence of this. One manifestation of this physics appears in the behaviour of extended Wilson operators. In this subsection, we explain this connection and present numerical and experimental results on the expectation values of these operators, which support the idea that for the state studied, the electronic degrees of freedom are indeed fractionalised.

We study this problem from the point of view of a dual-lattice gauge theory description of the FH model, where the electron is fractionalised into the (modified) Kotliar-Ruckenstein representation [38]

$$c^\dagger_{i\sigma} = f^\dagger_{i\sigma} h_i + \sum_{\sigma'} \epsilon_{\sigma\sigma'} f_{i\sigma'} d^\dagger_i, \tag{B3}$$

where $\epsilon_{\sigma\sigma'} = -\epsilon_{\sigma'\sigma}$ and $\epsilon_{\uparrow\downarrow} = 1$. Here $f_{i\sigma}$ are fermionic fields satisfying the usual anticommutation relation $\{f_{i\sigma}, f^\dagger_{j\sigma}\} = 0$, $\{f_{i\sigma}, f^\dagger_{j\sigma}\} = \delta_{ij}\delta_{\sigma'\sigma}$ while h_i, d_i are bosonic fields satisfying the $\mathfrak{sl}(2,\mathbb{C})$ algebra

$$[h_i, h^\dagger_j] = -\delta_{ij} Z^h_j, \quad [Z^h_j, h^\dagger_i] = \delta_{ij} h^\dagger_j, \tag{B4}$$

$$[d_i, d^\dagger_j] = -\delta_{ij} Z^d_j, \quad [Z^d_j, d^\dagger_i] = \delta_{ij} d^\dagger_j, \tag{B5}$$

and $[d_i, f_{j\sigma}] = [h_j, f_{j\sigma}] = 0$. The boson operators $h^\dagger_i, d^\dagger_i$ create a holon and a doublon, respectively. Since we do not want more than a doublon or a holon per site, we fix the representation of the h_j and d_j by the constraint $d^{\dagger 2}_j = h^{\dagger 2}_j = 0$. This fixes the operators h_i, d_i to be in the fundamental representation of $\mathfrak{sl}(2,\mathbb{C})$ and can be interpreted as the usual $\sigma^- = \frac{1}{2}(X - iY)$ matrices with $\{h_i, h^\dagger_i\} = \{d_i, d^\dagger_i\} = 1$. Representing the bosonic degrees of freedom as operators in a finite-dimensional Hilbert

space is the main difference between that representation [38] and the one that we use in this work. In this new enlarged basis, the anticommutation relation of the physical fermion operators becomes

$$\{c_{i\sigma}, c_{i\sigma}^\dagger\} = f_{i\sigma}^\dagger f_{i\sigma} + f_{i\bar{\sigma}}^\dagger f_{i\bar{\sigma}} + h_i^\dagger h_i + d_i^\dagger d_i - 2f_{i\sigma}^\dagger f_{i\sigma} h_i^\dagger h_i - 2f_{i\bar{\sigma}}^\dagger f_{i\bar{\sigma}} d_i^\dagger d_i, \tag{B6}$$

where $\bar{\uparrow} = \downarrow$ ($\bar{\downarrow} = \uparrow$). This means that to recover the physical states, we have to impose a constraint. The simplest one is

$$C_i = \sum_{\sigma=\uparrow,\downarrow} f_{i\sigma}^\dagger f_{i\sigma} + h_i^\dagger h_i + d_i^\dagger d_i = 1, \tag{B7}$$

such that the physical states $|\text{phys}\rangle$ satisfy $C_j|\text{phys}\rangle = |\text{phys}\rangle$ and in the physical states we have $\{c_{i\sigma}, c_{i\sigma}^\dagger\} = 1$. The operator C_j has integer eigenvalues $\lambda(C_j) = (0, 1, 2, 3, 4)$ with multiplicities $m(C_j) = (1, 4, 6, 4, 1)$. The projector onto the physical states can be written as $\mathcal{P} := \prod_j \mathcal{P}_j$ with the local projector

$$\mathcal{P}_j = \frac{C_j(2 - C_j)(3 - C_j)(4 - C_j)}{6}. \tag{B8}$$

Lastly, the fermion operator $c_{j,\sigma} = f_{i\sigma}^\dagger h_i + \epsilon_{\sigma\sigma'} f_{i\sigma'} d_i^\dagger$ commutes with the projector as $[c_{j\sigma}, C_j] = 0$ implies $[c_{j\sigma}, \mathcal{P}_j] = 0$. As a consequence, the physical operator does not create transitions between the physical states and the non-physical ones.

The physical Hamiltonian (as the enlarged Hamiltonian acting on the physical states) is given by $H = H_n + H_s + H_{\text{int}}$ with

$$H_n = -J \sum_{\sigma\langle ij\rangle} e^{i\phi_{ij}} (f_{i\sigma}^\dagger (h_i h_j^\dagger) f_{j\sigma} + f_{i\sigma} (d_i^\dagger d_j) f_{j\sigma}^\dagger) + \text{h.c.}, \tag{B9}$$

$$H_s = -J \sum_{\langle ij\rangle} e^{i\phi_{ij}} [(f_{i\uparrow}^\dagger f_{j\downarrow}^\dagger - f_{i\downarrow}^\dagger f_{j\uparrow}^\dagger) h_i d_j + (f_{i\downarrow} f_{j\uparrow} - f_{i\uparrow} f_{j\downarrow}) d_i^\dagger h_j^\dagger] + \text{h.c.}, \tag{B10}$$

$$H_{\text{int}} = U \sum_i d_i^\dagger d_i. \tag{B11}$$

This Hamiltonian has a local $U(1)$ gauge symmetry generated by the vertex operators $F_{j\sigma}^\theta := e^{i\theta f_{j\sigma}^\dagger f_{j\sigma}}$, $H_j^\theta := e^{i\theta h_j^\dagger h_j}$ and $D_j^\theta := e^{i\theta d_j^\dagger d_j}$ with $F_{j\sigma}^\theta f_{j\sigma} F_{j\sigma}^{\dagger\theta} = e^{i\theta} f_{j\sigma}^\dagger$, (with similar relations for the other operators). The local transformations $g_j^\theta := H_j^\theta F_{j\uparrow}^\theta D_j^\theta F_{j\downarrow}^\theta$ generates a $U(1)$ transformation that leaves the Hamiltonian invariant as

$$g_j^\theta c_{j\sigma} g_j^{\dagger\theta} = f_{j\sigma}^\dagger h_j + \epsilon_{\sigma\sigma'} f_{j\sigma'} d_j^\dagger. \tag{B12}$$

Motivated by the Hamiltonian H_n in Equation (B9), we can define the following gauge field mediating the interaction between the spinon $\mathcal{G}_{ij} := (d_i^\dagger d_j)$ which transforms as a gauge link operator $g_i^{\theta_j} g_j^{\theta_i} \mathcal{G}_{ij} g_j^{\dagger\theta_i} g_i^{\dagger\theta_j} = e^{-i(\theta_j - \theta_i)} \mathcal{G}_{ij}$. From this link operator, we can construct Wilson lines that are gauge invariants. For example, over a plaquette with sites from 1 to 4, we have

$$\mathcal{W}_d(\square) := \text{Tr}\left[\prod_{j\in\square} (d_{j-1}^\dagger d_j)\right] = \text{Tr}[(d_1^\dagger d_2)(d_2^\dagger d_3)(d_3^\dagger d_4)(d_4^\dagger d_1)] = \prod_{j\in\square} d_j d_j^\dagger. \tag{B13}$$

More generally, for any space-like loop $\mathcal{C}$ we define

$$\mathcal{W}_d(\mathcal{C}) := \prod_{j\in\mathcal{C}} d_j d_j^\dagger = \prod_{j\in\mathcal{C}} (1 - n_{j,\uparrow} n_{j,\downarrow}), \tag{B14}$$

where, in the last step, we used the map to the physical degrees of freedom.

In Figures 9 and 10 we observe the behaviour of the expectation of horizontal (doublon) Wilson loops $\langle \mathcal{W}_d(t) \rangle$ for different times, as we fix the area and vary the perimeter (Figure 9), or for fixed perimeter and varying the area (Figure 10). We observe that the $\mathcal{W}_d$ has a perimeter scaling, for any of the values of interaction U considered. Although these Wilson loops are horizontal (i.e, taken at a fixed time), they can still serve as order parameters for deconfinement (of spinons). Note that even at $U = 0$, we observe a perimeter law, with similar scaling but weaker strength for larger times. This is not surprising. As $\mathcal{W}_d$ only probes the interaction between spinons, we cannot claim spin-charge separation from this signature alone because the charge carriers could be deconfined in the same way without a clear separation between them, as would happen in the case of normal non-interacting fermions.

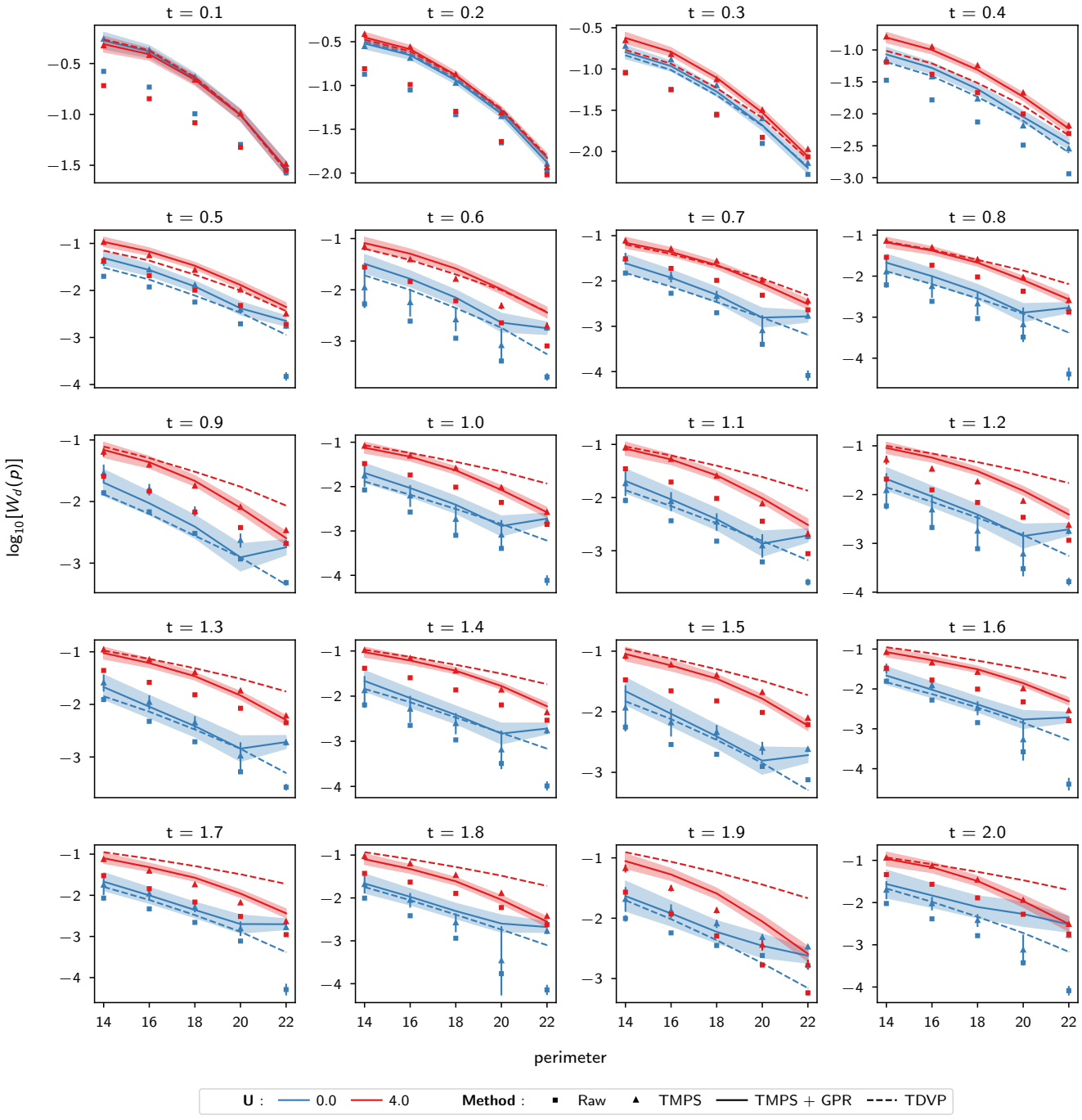


FIG. 9. Wilson loop with fixed area ($A = 10$) and varying perimeter. Same as in Figure 5 but all times shown.

To really compare the confinement/deconfinement between charge and spin carriers, we can define, inspired by H_n in Equation (B9), the complementary link operator between doublons given by

$$d_i^\dagger \left(\sum_\sigma f_{i\sigma} f_{j\sigma}^\dagger \right) d_j = d_i^\dagger \left(f_{i\uparrow} f_{j\uparrow}^\dagger + f_{i\downarrow} f_{j\downarrow}^\dagger \right) d_j = d_i^\dagger \begin{bmatrix} f_{i\uparrow} & f_{i\downarrow} \end{bmatrix} \begin{bmatrix} f_{j\uparrow}^\dagger \\ f_{j\downarrow}^\dagger \end{bmatrix} d_j \tag{B15}$$

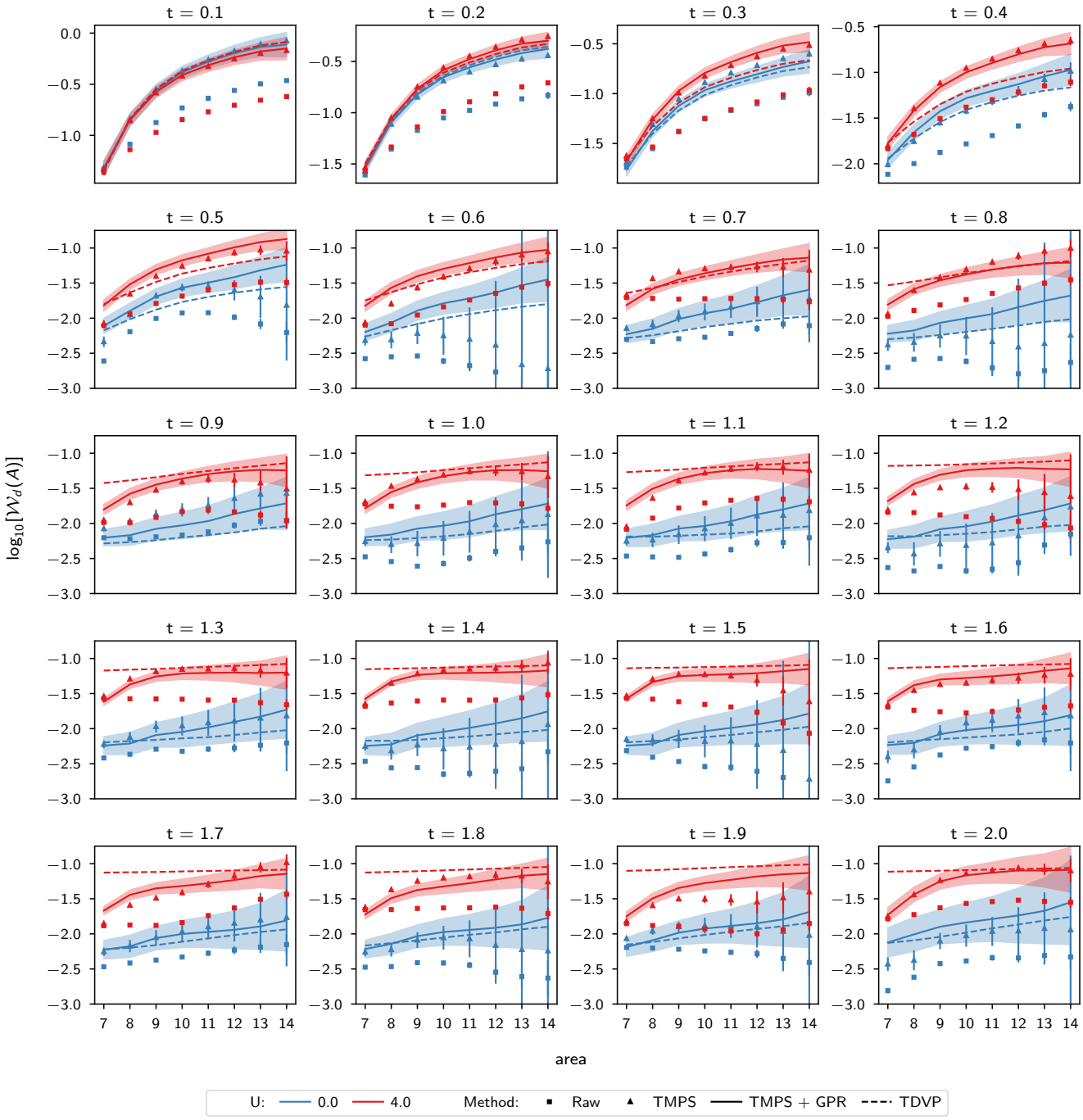


FIG. 10. Wilson loop with fixed perimeter ($p = 16$) and varying area. Same as in Figure 5, but all times shown.

The local gauge-invariant operator is then

$$W_{h,j} \coloneqq \begin{bmatrix} f^\dagger_{j\uparrow} \\ f^\dagger_{j\downarrow} \end{bmatrix} \begin{bmatrix} f_{j\uparrow} & f_{j\downarrow} \end{bmatrix} = \begin{bmatrix} f^\dagger_{j\uparrow} f_{j\uparrow} & f^\dagger_{j\uparrow} f_{j\downarrow} \\ f^\dagger_{j\downarrow} f_{j\uparrow} & f^\dagger_{j\downarrow} f_{j\downarrow} \end{bmatrix} = \begin{bmatrix} Q_j + S^3_j & S^+_j \\ S^-_j & Q_j - S^3_j \end{bmatrix}, \tag{B16}$$

where we have introduced the spin and charge operators $S^3_j \coloneqq \frac{1}{2}(f^\dagger_{j\uparrow} f_{j\uparrow} - f^\dagger_{j\downarrow} f_{j\downarrow})$, $S^+_j \coloneqq f^\dagger_{j\uparrow} f_{j\downarrow}$ and $Q_j \coloneqq \frac{1}{2}(f^\dagger_{j\uparrow} f_{j\uparrow} + f^\dagger_{j\downarrow} f_{j\downarrow})$. In the subspace of the physical states where the charge operator is fixed to $Q_j = 1/2$, W_h, j corresponds exactly to the Lax operator associated with the quantum inverse scattering method [62] for the 1D Heisenberg chain.

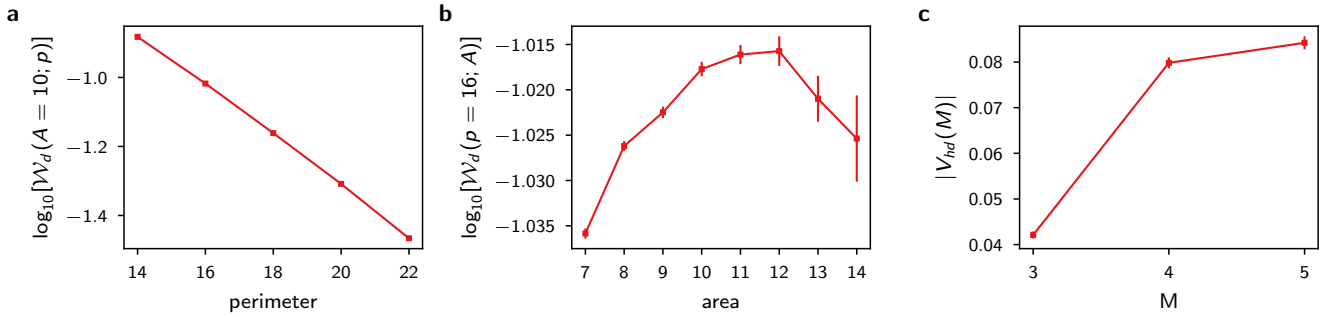


FIG. 11. Wilson loop with a) fixed area, varying perimeter; b) fixed perimeter, varying area; c) open Wilson line $V_{hd}(M)$ computed using 10000 shots from a DMRG simulation that approximates the ground-state of the FH model with $U = 4$.

The operator that is gauge invariant is given by the open Wilson line between doublons, and it takes the form

$$d_i^\dagger \mathcal{S}_{ij} d_j := d_i^\dagger \begin{bmatrix} f_{i\uparrow} & f_{i\downarrow} \end{bmatrix} \left(\prod_{k \in \mathcal{C}_{ij}} W_{h,k} \right) \begin{bmatrix} f_{j\uparrow}^\dagger \\ f_{j\downarrow}^\dagger \end{bmatrix} d_j, \tag{B17}$$

where $\mathcal{C}_{ij}$ is a path connecting the sites i and j. Remarkably, the product of gauge-invariant operators $\prod_{k\in\mathcal{C}} W_{k,h}$ along a closed path satisfying $Q_k = 1/2\ \forall k \in \mathcal{C}$ corresponds to the monodromy matrix of the Heisenberg model [62]. This monodromy matrix and the intertwining relation of the Lax operator can be used to prove the existence of N conserved quantities on the N-site Heisenberg model, where one of those is the Heisenberg Hamiltonian. We will discuss this connection in depth in a separate publication. We use this connection to isolate a meaningful operation associated with the quantum operator $\prod_{k\in\mathcal{C}_{ij}} W_{h,k}$ that can be extracted from measurements in the computational basis. The simplest non-trivial operator in the family generated by the product $\prod_{k\in\mathcal{C}_{ij}} W_{h,k}$ acting on the paths containing only spinons is the z component of the Heisenberg Hamiltonian. Using this, we *define* the line operator $H_{line} = \sum_{j\in line} S_j^z S_{j+1}^z$, that we measure in a line connecting two doublons or a doublon and a hole. Then we consider the expectation value of the operator

$$V_{\alpha\beta}(M) := \sum_{\substack{\text{sites } i,j: \\ |i-j|=M}} \frac{n_i^\alpha n_j^\beta}{N_{\text{pairs}}} \sum_{\substack{\text{paths } \gamma: \\ \text{from } i \text{ to } j}} \frac{\sum_{(m,n)\in\gamma} S_m^z S_n^z}{N_{\text{paths}}(i,j)} \tag{B18}$$

as a measure of the potential between the particles $\alpha, \beta = h, d$. Here $n_i^d := n_{i\uparrow} n_{i\downarrow}$, $n_i^h := (1 - n_{i\uparrow})(1 - n_{i\downarrow})$. We extract $V_{hd}(M)$ from the measured shots in the computational basis using the following algorithm

- Define a Manhattan distance M.
- Loop over the shots to find one doublon and one hole at distance M. This is a configuration, if no shots are found return zero, otherwise
- For path $\in$ the paths with Manhattan distance M that connect the two doublons, compute $R_{\text{path}} = \sum_{k\in\text{path}} S_k^z S_{k+1}^z$.
- Sum R_{path} over all paths, return this as $S(\text{config})$.
- Finally, return the sum over configurations and divide by the total number of paths. This is $V_{hd}(M)$.

In Figure 4 we sketch an example of this procedure for configurations starting with a doublon and ending in a holon.

As discussed in the main text, the growth with distance of the expectation of V^{hd} observed in Figure 6 signals a confining potential between the doublons and holons. This attraction is apparent from the signal that measures the expected charge difference between doublons and holons $\Delta_i = \langle n_i^d \rangle - \langle n_i^h \rangle$. Note that the sum over the lattice $\sum_i (\langle n_i^d \rangle - \langle n_i^h \rangle)$ is constant in time, due to total number conservation. In Figure 12 we observe the real-time dynamics of the signal $\langle n_i^d \rangle - \langle n_i^h \rangle$ for $U/J = 0$ (left) and $U/J = 4$ (right). Note that the expected local charge difference Δ_i for $U/J = 0$ seems to form charge density waves, the $U/J = 4$ signal tends faster to the uniform state $\Delta_i = 0$

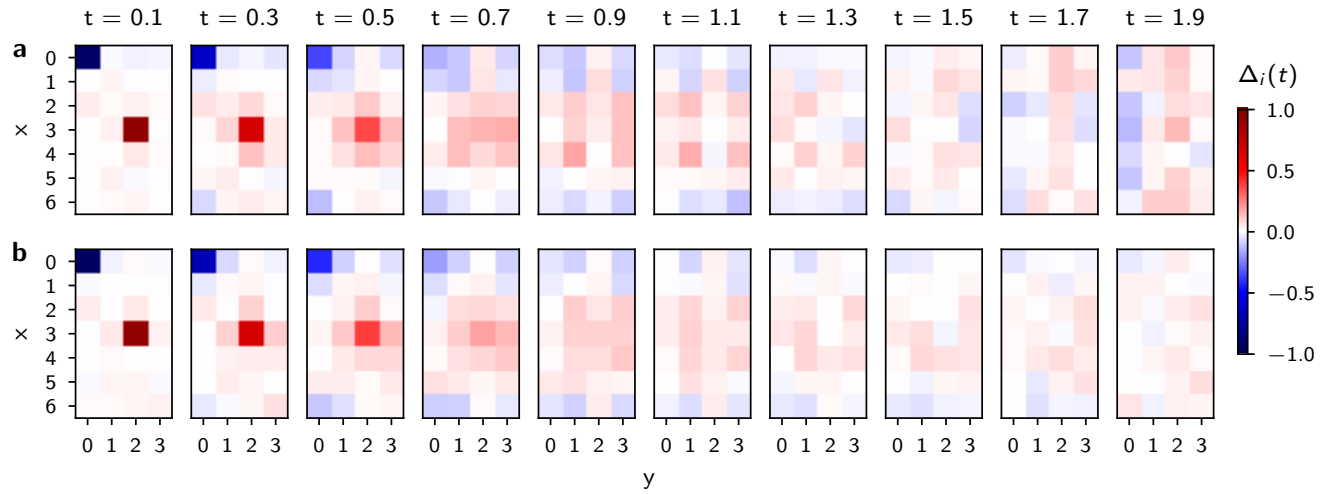


FIG. 12. Evolution of the charge difference $\Delta_i = \langle n_i^d \rangle - \langle n_i^h \rangle$ between doublons and holons. Note the similarities in the spread up to times $t \sim 0.5$ and the periodic nature of the system apparent from the wrapping around of the holon density. The total density Δ_i decays to zero faster in the interacting case compared with the non-interacting case.

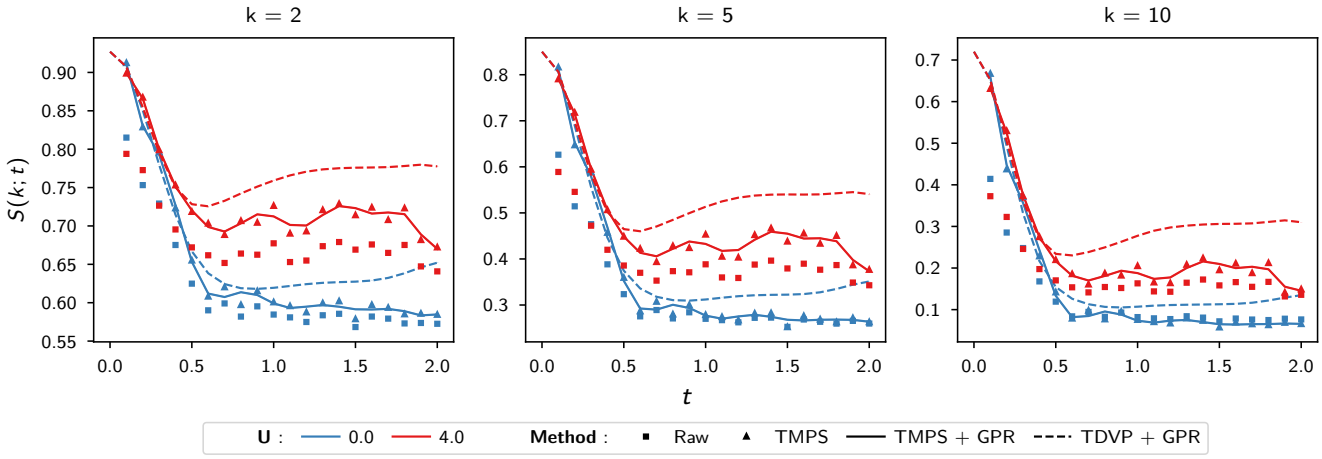


FIG. 13. Moments of the operator $|\mathcal{L}| - N_{\text{doublons}}$ that measures the absence of doublons. Each curve has been obtained by averaging over 1000 random subsets using 10000 samples from the TDVP simulation (dashed lines) and all the experimental shots (the continuous line is a guide for the eye for the mitigated results). Tata data have been produced using the TMPS map obtained from TDVP data up to $t = 0.5$.

Finally, to highlight the different states that the experiment and the TDVP algorithm produce, we study different moments of the (normalized) absence of doublons operator $X = 1 - \frac{N_{\text{doublons}}}{|\mathcal{L}|}$ given by

$$\langle X^k(t) \rangle = \sum_{\{j_m\}_{m=1}^k \in \mathcal{L}} \frac{1}{|\mathcal{L}|^k} \left\langle \prod_{m=1}^{k} (1 - n_{j_m,\uparrow}(t) n_{j_m,\downarrow}(t)) \right\rangle . \tag{B19}$$

We estimate these moments through a Monte Carlo estimator by sampling random subsets of k sites 1000 times. The results are shown in Figure 13 for $k = 2, 5, 10$. We observe divergence between the results of the tensor networks, mitigated experimental data and the results obtained by sampling the TDVP state after $t \sim 0.7$. This divergence is essentially the one observed for the number of doublons operator N_{doublons}, as the moment k of X is very close to the mean $\mu = \langle X \rangle$ to the power k as

$$\langle X^k \rangle = \langle [\mu + [X - \mu]]^k \rangle = \sum_{m=0}^{k} \binom{k}{m} \mu^m \langle [X - \mu]^{k-m} \rangle, \tag{B20}$$

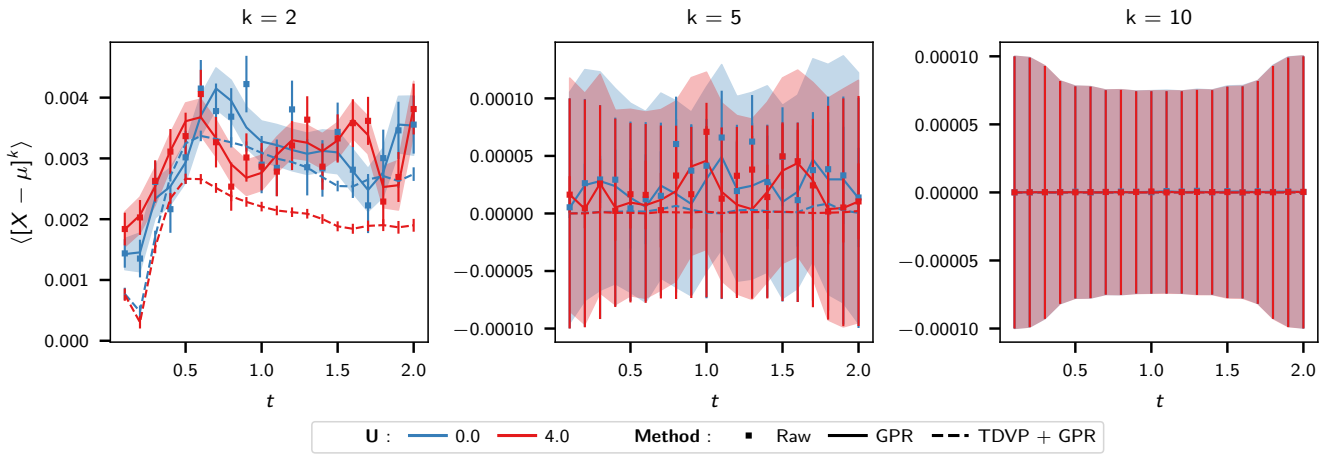


FIG. 14. Central moments $\langle [X-\mu]^k \rangle$ for different values of k.

and the centred moments $\langle [X-\mu]^k \rangle$ are very small. In order to distinguish the value of the centred moments from zero and from the moments produced by sampling the TDVP state, more shots would be required.

Appendix C: Quantum Circuits

1. Fermion to Qubit Mapping

We use the Jordan-Wigner (JW) mapping to represent the fermionic system on qubits. Every mode is assigned to a qubit and Fock states are mapped to computational basis states such that a mode is occupied (unoccupied) if its corresponding qubit is in the $|1\rangle$ ($|0\rangle$) state. An ordering is chosen for the modes (correspondingly for the qubits) and creation/annihilation operators are mapped as

$$c_j^\dagger = \sigma_j^+ \prod_{i<j} Z_i, \quad c_j = \sigma_j^- \prod_{i<j} Z_i, \quad n_j = \frac{1}{2}(1 - Z_j), \tag{C1}$$

where $\sigma_j^\pm = (X \mp iY)/2$. The $\sigma^\pm$ parts capture the creation/annihilation properties and the Z strings ensure the correct anticommutation relations. This work uses the "snake" ordering for modes on a square lattice, where modes are ordered along rows from left to right and right to left in an alternating manner. Both spin-up and spin-down sectors are snake-ordered, with the spin-down sector coming after spin-up, as shown in Figure 16.

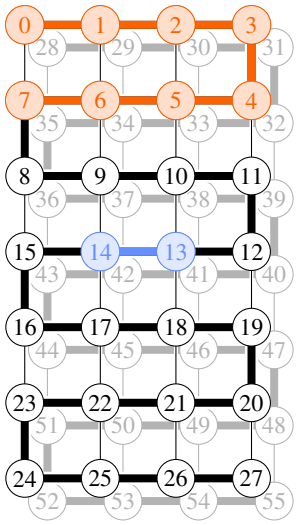


FIG. 15.

FIG. 16. Jordan-Wigner snake ordering on a spin-$\frac{1}{2}$, 4×7 grid. The spin-up sector is in the foreground and the spin-down sector is in the background. In the qubit picture, the support of a hopping term is on the qubits corresponding to the hopping pair and all modes in between them in the ordering. A horizontal hopping term is only supported on 2 qubits (blue) but a vertical hopping term can be supported on a longer chain covering the width of the lattice (red).

a. *Encoded Interactions*

Undressed hopping interactions are represented as

$$c_i^\dagger c_j + c_j^\dagger c_i = \frac{1}{2}(X_i X_j + Y_i Y_j) \prod_{i<k<j} Z_k. \tag{C2}$$

When a magnetic field is present these are represented as

$$e^{i\phi_{ij}} c_i^\dagger c_j + e^{-i\phi_{ij}} c_j^\dagger c_i = \frac{1}{2}(\cos\phi_{ij}(X_i X_j + Y_i Y_j) + \sin\phi_{ij}(Y_i X_j - X_i Y_j)) \prod_{i<k<j} Z_k, \tag{C3}$$

although in our case we only use $\phi_{ij} \in \{0, \pi\}$ where 0 gives the undressed hopping term and π gives the undressed term with a minus sign.

The onsite Coulomb terms of the Fermi-Hubbard Hamiltonian are represented on qubits as

$$n_{i,\uparrow}n_{i,\downarrow} = \frac{1}{4}(1 - Z_{i,\uparrow} - Z_{i,\downarrow} + Z_{i,\uparrow}Z_{i,\downarrow}). \tag{C4}$$

The part proportional to 1 may be ignored as it will only contribute an unobservable global phase to the evolution. Furthermore, since this term appears for every spin pair, the Hamiltonian contains a part proportional to $\sum_i (Z_{i,\uparrow} + Z_{i,\downarrow}) = \frac{1}{2}\sum_i (2 - n_{i,\uparrow} - n_{i,\uparrow})$ which may also be ignored because it will only contribute a global phase to states of fixed particle number. For our purposes it then suffices to represent the Coulomb term as

$$n_{i,\uparrow}n_{i,\downarrow} \approx \frac{1}{4}Z_{i,\uparrow}Z_{i,\downarrow}. \tag{C5}$$

b. Fixed Parity JW Loop

Since the Fermi-Hubbard model preserves parity in each spin sector, for states of fixed parity in spin sector σ, the parity operator $P_\sigma = \prod_i V_{i,\sigma}$ acts as a constant. In particular, for even (odd) parity states we have $P_\sigma = +1$ ($P_\sigma = -1$). This means that, for fixed parity states, the interactions can be multiplied by the $\pm$ parity operator with no effect on the physics. Specifically, one multiplies by $+P_\sigma$ when it is positive and $-P_\sigma$ when it is negative. This can be helpful for JW encoded systems as it allows the Z string involved in hopping interaction to be "flipped" like so

$$\pm P(c_i^\dagger c_j + c_j^\dagger c_i) = \mp\frac{1}{2}(X_iX_j + Y_iY_j) \prod_{\substack{0\le k<i \\ j<k\le N}} Z_k. \tag{C6}$$

where N is the number of modes in the sector and we have omitted the spin index. This flexibility of representation means that hopping terms between modes that are distant with respect to the ordering can be made much lower weight, providing alternative, more efficient avenues for their implementation. In this work, we only apply this move to undressed hopping terms due to our choice of magnetic field, but the effect on dressed terms is analogous.

As a result of this, one can consider the mode ordering within a spin-sector to be a loop where the first and final spin-σ are also adjacent to one another and hopping terms can be represented with a Z string between the modes in either direction (so long as the appropriate sign is applied). This is particularly useful when simulating lattices with periodic boundary conditions.

2. Fermionic Swap Networks

Encoded hopping terms between horizontal pairs are two-qubit operators as they are adjacent in the ordering, so their evolution can be implemented with a simple quantum circuit involving two two-qubit gates (see Section C 4). The hopping terms between vertical pairs are not adjacent in this sense: they involve long strings of Z operators and can be costly to implement (see Figure 16). An efficient way to implement evolution under these terms is via fermionic swap networks [23].

The fermionic swap gate (FSWAP) acts as

$$\text{FSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} = \text{SWAP} \cdot \text{CZ}, \tag{C7}$$

and transforms Paulis under conjugation as

$$XI \leftrightarrow ZX, \quad YI \leftrightarrow ZY, \quad ZI \leftrightarrow IZ. \tag{C8}$$

Under the JW encoding, when acting on qubits $(j, j+1)$ adjacent in the ordering, it transforms the encoded fermionic operators as

$$c_j \leftrightarrow c_{j+1}, \quad c_j^\dagger \leftrightarrow c_{j+1}^\dagger, \quad n_j \leftrightarrow n_{j+1}, \tag{C9}$$

effectively swapping the positions of the encoded modes, hence the name. FSWAP operations can be implemented on the Quantinuum hardware using only a single two-qubit gate (see Section C 4 for further details).

Networks of these operations can be used to rearrange the modes such that every interacting pair is adjacent at some point with respect to the ordering, at which time their interaction is implemented via the two-qubit hopping circuit (Section C 4).

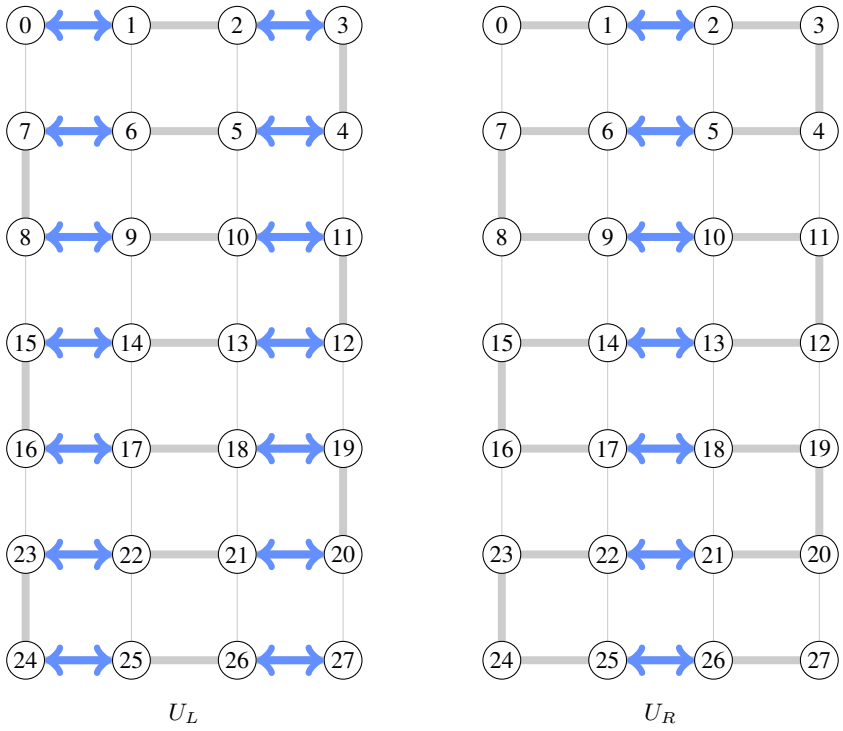


FIG. 17. Illustration of the action of U_L and U_R on a 4×7 grid, the blue arrows denote FSWAPs.

3. Second-Order Trotter Circuit

Our second-order Trotter step of time t has the basic structure

$$S_2(t) = \prod_{h\in\mathcal{H}}^{\rightarrow} e^{-ih\frac{t}{2}} \prod_{o\in\mathcal{O}} e^{-iot} \prod_{h\in\mathcal{H}}^{\leftarrow} e^{-ih\frac{t}{2}}, \tag{C10}$$

where $\mathcal{H}$ is the set of hopping interactions in the Hamiltonian and $\mathcal{O}$ is the set of Coulomb interactions. The arrows indicate that the second round of hopping interactions is applied in the reverse order to the first, the Coulomb interactions all commute and are applied in parallel directly via R_{zz} gates. The hopping parts are implemented via FSWAP networks, which specify the ordering of the terms. The second hopping part of the circuit is applied with the same gate schedule as the first, only in reverse order. Below, we present the instructions for applying the hops to a single spin sector, which are also applied to both the spin-up and spin-down sectors.

a. Swap Network

Let L_x and L_y be the horizontal and vertical lattice dimensions and U_L (U_R) be the circuit which FSWAPs even (odd) columns with those to their right, in cases where the leftmost column is even; see Figure 17. For some pairs, FSWAPs will be merged with a hopping interaction, which can be done with the same circuit cost as an FSWAP; see Section C 4.

The first round of hopping terms is implemented via one of the following swap network schedules. For a step size of t all hopping terms are applied for time $t/2$.

For even L_x:

1. Implement all vertical hopping terms between modes adjacent in the ordering.

2. Apply U_L, replacing any FSWAP with a merged FSWAP and hopping interaction if the pair of modes being swapped is involved in an unimplemented hopping term.

3. Implement all vertical hopping terms between modes adjacent in the ordering.

4. Apply U_R,

   replacing any FSWAP with a merged FSWAP and hopping interaction if the pair of modes being swapped is involved in an unimplemented hopping term.

5. Repeat from 2. terminating as soon as all hopping terms have been implemented.

For odd L_x:

1. Implement all hopping terms between pairs on the leftmost column that are adjacent in the ordering.
2. Implement all hopping terms between pairs on the rightmost column that are adjacent in the ordering.
3. Apply U_L, replacing any FSWAP with a merged FSWAP and hopping interaction if the pair of modes being swapped is involved in an unimplemented hopping term.
4. Implement all hopping terms between pairs on the leftmost column that are adjacent in the ordering.
5. Apply U_R, replacing any FSWAP with a merged FSWAP and hopping interaction if the pair of modes being swapped is involved in an unimplemented hopping term.
6. Repeat from 2. terminating as soon as all hopping terms have been implemented.

Note that the resulting swap network is shorter than that required for a first-order Trotter step. This is because for the first order, we would need to return the modes to their original positions in preparation for the next step, whereas here we are content to leave them in a jumbled state because the original positioning will be restored after the second swap network.

b. Periodic Boundary Conditions

In either case, the horizontal hops across the periodic boundary are implemented automatically by the swap networks above, as every pair of modes on a row is adjacent at some point in the schedule.

The vertical hops are only automatically implemented by the above networks in the case of an even *vertical* dimension L_y.

This is because of how the Jordan-Wigner ordering lines up between the top and bottom rows. In the even case, every vertical pair is adjacent w.r.t. the JW loop at some point in the swap network, but in the odd case, this never happens for the pairs connecting the top and bottom rows. These hopping terms are implemented by a bespoke circuit after the main swap network is completed. This circuit is detailed in Section C 4.

c. Trotter Step Merging

The symmetry of second-order Trotter circuits allows for further gate savings to be made. In cases of even and odd L_x, the Trotter step begins and ends with the application of all vertical hopping terms available in the initial configuration (step 1. for even, steps 1. and 2. for odd). This means that the final operation in step j can be merged with the first in step $j+1$ with the exception of the final Trotter step in the circuit. Specifically, for a circuit with N Trotter steps numbered 1 to N with step j being for time t_j, steps $j < N$ have these hopping terms omitted from the end of their circuit and steps $k > 1$ have the hopping terms at the start evolve for $(t_k + t_{k-1})/2$ instead of $t_k/2$. This reduces the number of gates required; the precise saving is discussed in Section C 5. A complete Trotter step for a 3×3 lattice is illustrated in Figure 18.

d. Trotter Error

We estimate the accuracy of the Trotterized dynamics using small-scale numerical simulations on the Fermi-Hubbard instance studied in this work. We simulate the error in the expectation values of all Pauli Z observables up to weight-3, using 4 Trotter steps and for times up to $T = 2$. That is, for each observable O and time t we compute all

$$|\langle\psi(t)|\, O\, |\psi(t)\rangle - \langle\psi'(t)|\, O\, |\psi'(t)\rangle\,|. \tag{C11}$$

Where $|\psi(t)\rangle = e^{-itH}|\psi_0\rangle$ and $|\psi'(t)\rangle = \prod_{j=1}^{4} S_2(t/4)\,|\psi_0\rangle$. For $|\psi_0\rangle$, we choose the dimerized configuration used in the experiments. We then examine the average and maximum values of these errors. We also compute infidelity between the exact time-evolved state $|\psi(t)\rangle$ and Trotterized time-evolved state $|\psi'(t)\rangle$, defined as

$$2\sqrt{1 - |\langle\psi(t)|\psi'(t)\rangle|^2} \tag{C12}$$

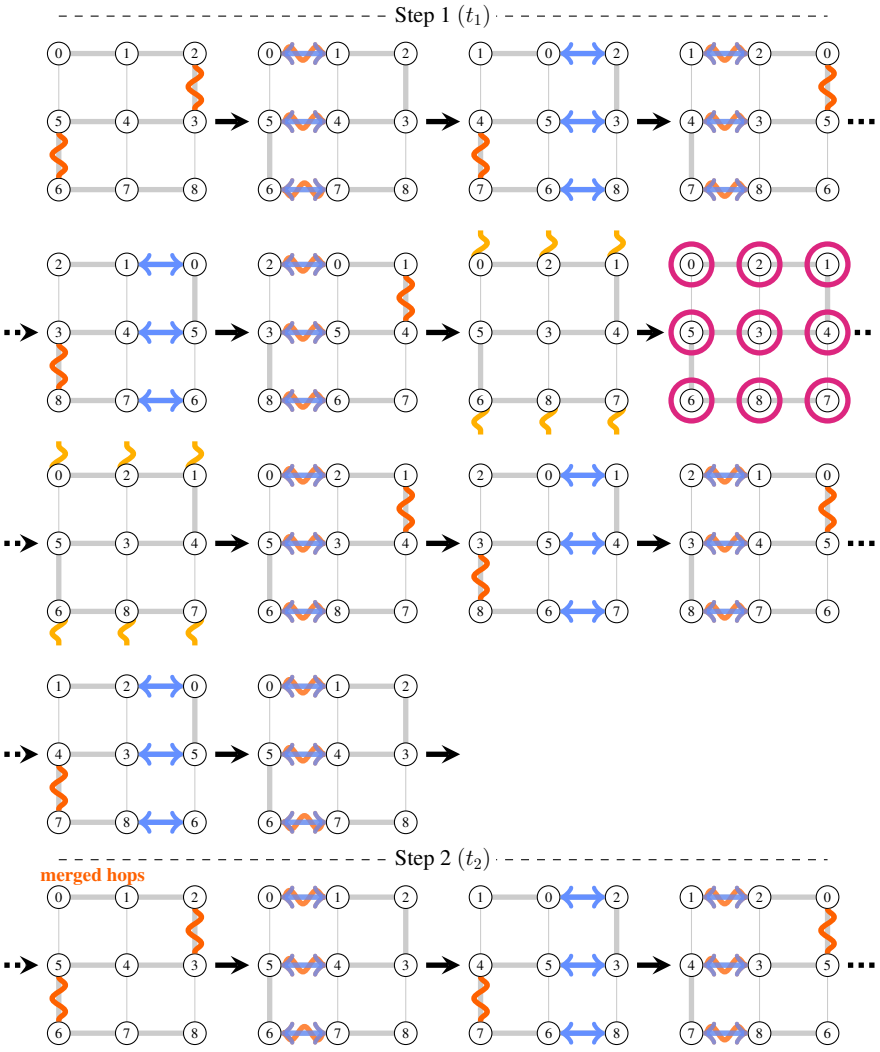
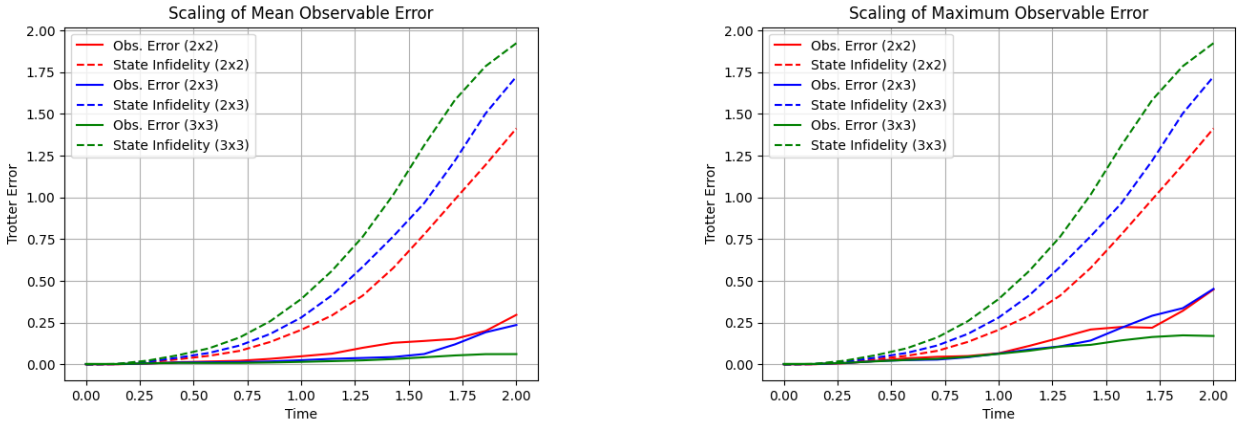


FIG. 18. Illustration of the first Trotter step and the start of the second for a 3×3 lattice. Identical operations are made on both spin sectors, so only spin-up is shown. Blue arrows indicate FSWAPs, red wavy lines indicate hopping interactions and orange wavy lines denote the special circuit used to implement hopping terms missed by the swap network in the odd L_x case. All hopping terms are for time $t_1/2$ ($t_2/2$) in step 1 (2), except in the case where merging is noted. Blue arrows and red wavy lines superimposed indicate a simultaneous hop and FSWAP and the magenta circles indicate Coulomb interactions for time t_1 (t_2) with the corresponding spin-down modes. Parallelizable actions are shown on the same grid and labels indicate the location of the encoded fermionic modes at each stage. The final round of hops of step 1 and the first round of step 2 are merged into hops of time $(t_1 + t_2)/2$.

in order to upper bound the Trotter error for all observables. The results of the simulation are shown in Figure 19. We observe that for late times and simulable sizes, although state infidelity increases with system size, the local observable error remains relatively stable. This error is significantly smaller than infidelity, with the average error being comparable to hardware noise. For local observables and local Hamiltonians, there are theoretical grounds for assuming that Trotter error can be independent of system size, especially for fixed times [63].

However, we note that between times 1.5 and 2, Trotter error begins to play a more significant role, and our simulations begin interpolating between true time dynamics and Floquet dynamics. This is borne out in our comparisons to exact $U = 0$ simulations, as can be seen in Figure 3.

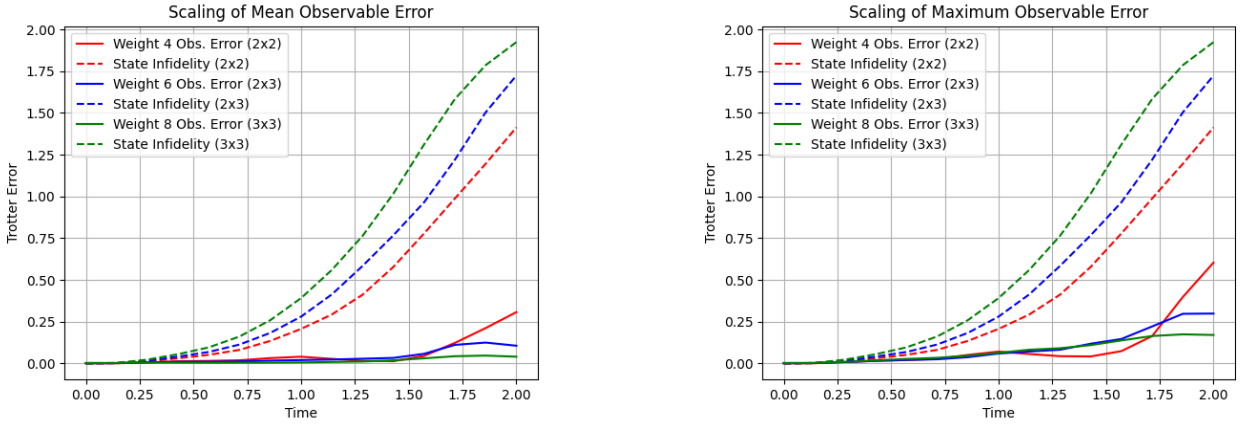
We also repeat this analysis for observables whose weight scales with system size. Again we look at system sizes $L_x \times L_y = 2 \times 2$, 2×3 and 3×3 but with observables with weight four, six and eight. The results of these calculations are shown in Figure 20.



(a) Mean error over observables.

(b) Maximum error over observables.

FIG. 19. Trotter error on the expectation values (Equation (C11), solid lines) of all one, two and three qubit Pauli Z observables for 4 second-order Trotter steps over 15 times between $t = 0$ and $t = 2$. As in our experiments, we take $U = 4$, apply a magnetic flux in the smaller direction and use a dimerized configuration as input state. We then examine the system sizes $L_x \times L_y = 2 \times 2$, 2×3 and 3×3 subject to double periodic boundary conditions. We look at the mean error over all observables at each time (a) and the maximum error over these observables at each time (b). We compute the state infidelity (Equation (C12), dashed lines) between the exact time evolved state at each time t and the state evolved using 4 Trotter steps for each time t.



(a) Mean error over observables.

(b) Maximum error over observables.

FIG. 20. Trotter error on the expectation values (Equation (C11), solid lines) of all weight-four Pauli-Z observables for $L_x \times L_y = 2 \times 2$, all weight-six observables for $L_x \times L_y = 2 \times 3$ and weight-eight observables for $L_x \times L_y = 3 \times 3$. We use 4 second-order Trotter steps over 15 times between $t = 0$ and $t = 2$. As in our experiments, we take $U = 4$, apply a magnetic flux in the smaller direction and use a dimerized configuration as the input state. We use double periodic boundary conditions. We look at the mean error over all observables at each time (a) and the maximum error over these observables at each time (b). We compute the state infidelity (Equation (C12), dashed lines) between the exact time evolved state at each time t and the state evolved using 4 Trotter steps for each time t.

4. Gate Decompositions

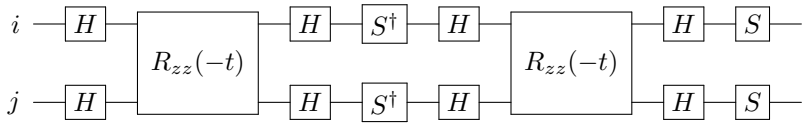
We use the notation for Pauli rotation

$$R_\sigma(t) = e^{-\frac{i}{2}\sigma t}. \tag{C13}$$

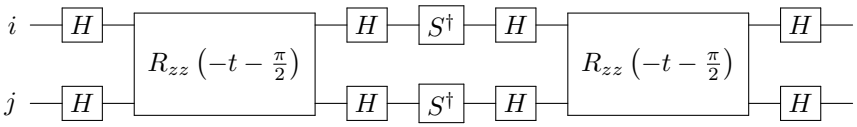
The native two-qubit gate on the device is the arbitrary R_{zz}. Due to the all-to-all connectivity of the device, SWAP operations can be applied implicitly by relabeling qubits. With this, one can implement an FSWAP by applying a CZ and an implicit swap to a pair of qubits (in any order), where the CZ gate can be applied with the gates

$$\mathrm{CZ} = R_{zz}\left(\frac{\pi}{2}\right) \cdot S \otimes S. \tag{C14}$$

An undressed ($\phi_{ij} = 0$) hopping interaction between modes adjacent in the JW ordering with $J = 1$ for time t – i.e. $e^{i(c_i^\dagger c_j + c_j^\dagger c_i)t} = e^{i\frac{1}{2}(X_iX_j+Y_iY_j)t}$ – is implemented with the circuit:



If the pair of modes is adjacent across the loop discussed in Section C 1 b, then the sign of the R_{zz} rotations changes according to Equation (C6). In the cases where a hopping interaction is immediately followed by or preceded by an FSWAP, we can apply both at once using the circuit:

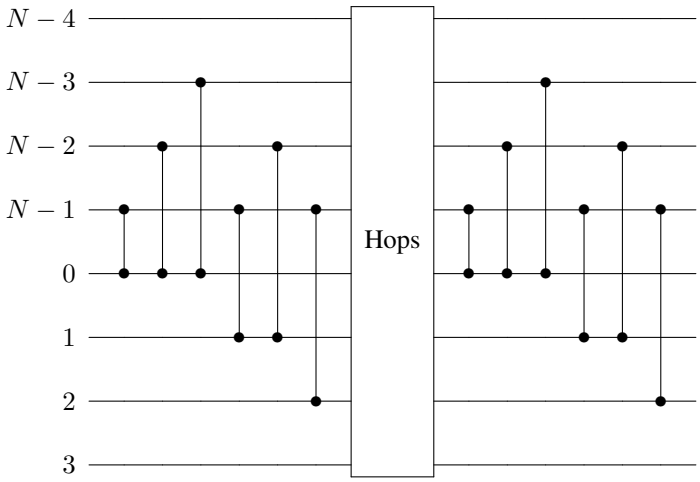


When the $\phi_{ij} = \pi$ replace t with $-t$.

For a lattice with odd vertical dimension L_y, the vertical terms across the vertical boundary (i.e., hops between pairs $\{(0, N - L_x), (1, N - L_x + 1), \ldots, (L_x - 1, N - 1)\}$) are implemented by the following circuit

$$\left(\prod_{i=0}^{L_x-2}\prod_{j=1}^{L_y-1-i} \mathrm{CZ}(i, N-j)\right) \cdot \prod_{i=0}^{L_x-1} \mathrm{HOP}(i, N-L_x+i, t) \cdot \left(\prod_{i=0}^{L_x-2}\prod_{j=1}^{L_y-1-i} \mathrm{CZ}(i, N-j)\right) \tag{C15}$$

where $\mathrm{HOP}(i, j, t)$ denotes the two-qubit hopping interaction circuit between qubits i and j for time t. The case of $L_x = 4$ is illustrated below in circuit diagram form.



The CZ gates in the circuit serve to cancel out the Z-strings in the encoded hopping terms. Consider the $L_x = 4$ example. The hopping terms in question are (up to factors)

$$\begin{aligned}
h_{0,N-4} &= (X_0X_{N-4} + Y_0Y_{N-4})Z_{N-3}Z_{N-2}Z_{N-1}, \\
h_{1,N-3} &= (X_1X_{N-3} + Y_1Y_{N-3})Z_{N-2}Z_{N-1}Z_0, \\
h_{2,N-2} &= (X_2X_{N-2} + Y_2Y_{N-2})Z_{N-1}Z_0Z_1, \\
h_{3,N-1} &= (X_3X_{N-1} + Y_3Y_{N-1})Z_0Z_1Z_2,
\end{aligned} \tag{C16}$$

where we are using the JW loop to get a lower weight representation of the terms as discussed in Section C 1 b and there is no magnetic field alteration. Defining the "stringless" versions of the hopping terms as

$$\begin{aligned}
\tilde{h}_{0,N-4} &= (X_0X_{N-4} + Y_0Y_{N-4}), \\
\tilde{h}_{1,N-3} &= (X_1X_{N-3} + Y_1Y_{N-3}), \\
\tilde{h}_{2,N-2} &= (X_2X_{N-2} + Y_2Y_{N-2}), \\
\tilde{h}_{3,N-1} &= (X_3X_{N-1} + Y_3Y_{N-1}),
\end{aligned} \tag{C17}$$

it is not hard to check that conjugating $\tilde{h}_{i,j}$ by the CZ circuit blocks above results in the corresponding $h_{i,j}$ for all pairs in question. Accordingly, evolution under the stringless hopping terms conjugated by these CZs is equivalent to evolution under the true, stringed terms.

The reader may recall that, as this is applied after the swap network, the modes will be shuffled from their original ordering. This does not affect this circuit, as at any point in the swap network (Equation (C16)) holds up to a permutation of the labels on the LHS of the equations and since the circuit applies an evolution under all the Paulis, this relabelling is irrelevant to the outcome.

The total two-qubit gate cost of this circuit for the whole system is

$$C_{\text{b}} = 4\frac{(L_x-1)L_x}{2} + 4L_x, \tag{C18}$$

where the first term is from the CZ gates, the second is from the hopping terms. As this circuit is always placed before and after the Coulomb interaction circuit, the set of CZ gates on either side of the Coulomb circuit can be commuted through to cancel each other out, making the effective cost of this circuit

$$\widetilde{C}_{\text{b}} = 2\frac{(L_x-1)L_x}{2} + 4L_x = L_x^2 + 3L_x. \tag{C19}$$

5. Trotter Circuit Gate Costs

The total two-qubit gate cost of a single forward pass of the swap network depends on the number of FSWAPs, vertical hopping terms and horizontal hopping terms implemented (n_f, n_v, n_h respectively) and is given by

$$C_{\text{nw}} = n_f + 2n_v + n_h. \tag{C20}$$

Each vertical hopping term costs two two-qubit gates but horizontal hopping terms effectively cost only 1 because they are merged with an FSWAP for a total of 2 gates, this is counted as 1 gate due to the FSWAP and 1 due to the hopping. The numbers of each type of hopping term implemented in the swap network are given by

$$n_h = 2L_xL_y, \quad n_v = \begin{cases} 2L_xL_y & : L_y \text{ even}, \\ 2L_x(L_y-1) & : L_y \text{ odd}. \end{cases} \tag{C21}$$

This is due to the fact that the vertical hopping terms across the periodic boundary are not caught by the swap network as discussed in Section C 3 b.

The total number of swaps can be found by counting how many applications of U_L and U_R are needed for each mode to reach both ends of its horizontal row. For even L_x this requires $L_x - 1$ applications of U_L and $L_x - 2$ of U_R, with U_L and U_R containing L_xL_y and $(L_x-2)L_y$ FSWAPs respectively (counting both spin sectors). For odd L_x this requires $L_x - 1$ of both U_L and U_R which both contain $(L_x-1)L_y$ FSWAPs. The usual swap network that returns the lattice to its original configuration on completion requires L_x applications of U_L and U_R in both cases.

The total number of FSWAPs in our swap network is then:

$$n_f = \begin{cases} (2L_x^2 - 5L_x + 4)L_y & : L_x \text{ even}, \\ (2L_x^2 - 4L_x + 2)L_y & : L_x \text{ odd}. \end{cases} \tag{C22}$$

Combining these, we get

$$C_{\text{nw}} = \begin{cases} (2L_x^2 + L_x + 4)L_y & : L_x \text{ even}, L_y \text{ even}, \\ (2L_x^2 + L_x + 4)L_y - 4L_x & : L_x \text{ even}, L_y \text{ odd}, \\ (2L_x^2 + 2L_x + 2)L_y & : L_x \text{ odd}, L_y \text{ even}, \\ (2L_x^2 + 2L_x + 2)L_y - 4L_x & : L_x \text{ odd}, L_y \text{ odd}. \end{cases} \tag{C23}$$

As discussed in Section C 3 c, all but the last Trotter step have their final round of vertical hopping terms omitted as they are merged into the start of the next step. For each step, this amounts to a two-qubit gate saving of

$$C_{\text{merge}} = \begin{cases} 4L_y & : L_y \text{ even}, \\ 4(L_y-1) & : L_y \text{ odd}. \end{cases} \tag{C24}$$

Lattice Size	2q gates (our network)	2q gates (standard network)
4×5	$428N + 16$	$484N + 40$
4×6	$486N + 24$	$552N + 48$
5×5	$674N + 16$	$717N + 48$
4×7	$588N + 24$	$668N + 56$

TABLE I. Comparison for various lattice sizes of two-qubit gate cost for an N step second order Trotter circuit using the optimised circuit described in Section C 3 and the standard swap network used on JW lattices. Note that all lattices have periodic boundary conditions and the costs for the standard network also include the same circuit for vertical boundary terms and account for merged hops/swaps between Trotter steps.

The merged hopping terms are all vertical hops adjacent in the ordering in the initial lattice configuration before any FSWAPs, L_y for even L_y and $L_y - 1$ for odd, due to the misalignment of the JW ordering. The factor of 4 comes from the 2 spin sectors and the two-qubit gate cost for a hopping term.

As noted above, the cost of the circuit to implement vertical hops missed by the swap network is

$$\widetilde{C}_{\mathrm{b}} = \begin{cases} 0 & : L_y \text{ even}, \\ L_x^2 + 3L_x & : L_y \text{ odd}, \end{cases} \tag{C25}$$

and the cost of the Coulomb interactions is

$$C_{\mathrm{c}} = L_x L_y. \tag{C26}$$

Putting these together, the total cost of an N step second-order Trotter circuit is

$$C_{\mathrm{tot}} = N(2C_{\mathrm{nw}} + 2\widetilde{C}_{\mathrm{b}} + C_{\mathrm{c}} - C_{\mathrm{merge}}) + C_{\mathrm{merge}}, \tag{C27}$$

where the final addition of C_{merge} is due to the final step not having anything to merge its last hops with. In terms of lattice dimensions, this is

$$C_{\mathrm{tot}} = \begin{cases} N(4L_x^2 + 3L_x + 5)L_y + 4L_y & : L_x \text{ even}, L_y \text{ even}, \\ N((4L_x^2 + 3L_x + 4)L_y + 2L_x^2 - 2L_x + 4) + 4(L_y - 1) & : L_x \text{ even}, L_y \text{ odd}, \\ N(4L_x^2 + 5L_x)L_y + 4L_y & : L_x \text{ odd}, L_y \text{ even}, \\ N((4L_x^2 + 5L_x)L_y + 2L_x^2 - L_x + 4) + 4(L_y - 1) & : L_x \text{ odd}, L_y \text{ odd}. \end{cases} \tag{C28}$$

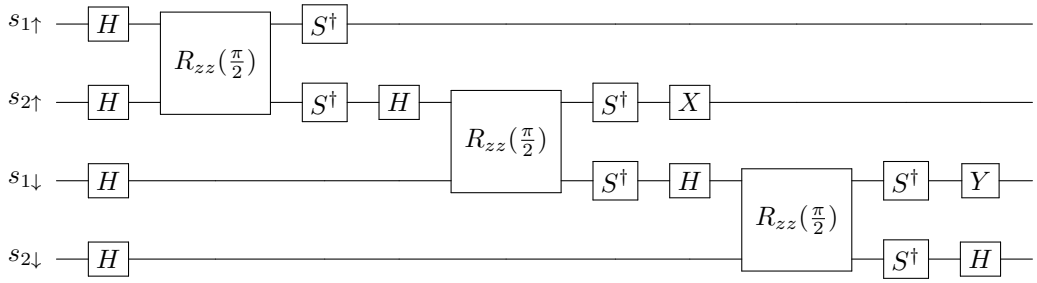
Table I shows a comparison between the gate cost of a second-order Trotter circuit using our optimisations vs a standard swap network [23].

6. Initial state preparation

We prepare a series of triplets across our lattice. A triplet is prepared over two sites, s_1, s_2, given by

$$|\psi_{\mathrm{triplet}}\rangle = \frac{1}{\sqrt{2}}(c^\dagger_{s_1,\uparrow} c^\dagger_{s_2,\downarrow} + c^\dagger_{s_1,\downarrow} c^\dagger_{s_2,\uparrow}) |\Omega\rangle = \frac{1}{\sqrt{2}}(|1_{s_1,\uparrow} 0_{s_2,\uparrow} 0_{s_1,\downarrow} 1_{s_2,\downarrow}\rangle - |0_{s_1,\uparrow} 1_{s_2,\uparrow} 1_{s_1,\downarrow} 0_{s_2,\downarrow}\rangle) \tag{C29}$$

This state is produced using the following circuit, which is a modified Greenberger–Horne–Zeilinger state creation circuit.



(C30)

This state preparation circuit incurs a cost of three two-qubit gates per triplet pair. In our system of 28 sites, Figure 21, all sites are involved in a triplet, except for sites 0 and 13, which encode a holon and a doublon, respectively. Therefore, we have 13 triplet pairs, so state preparation accounts for 39 two-qubit gates in Table II.

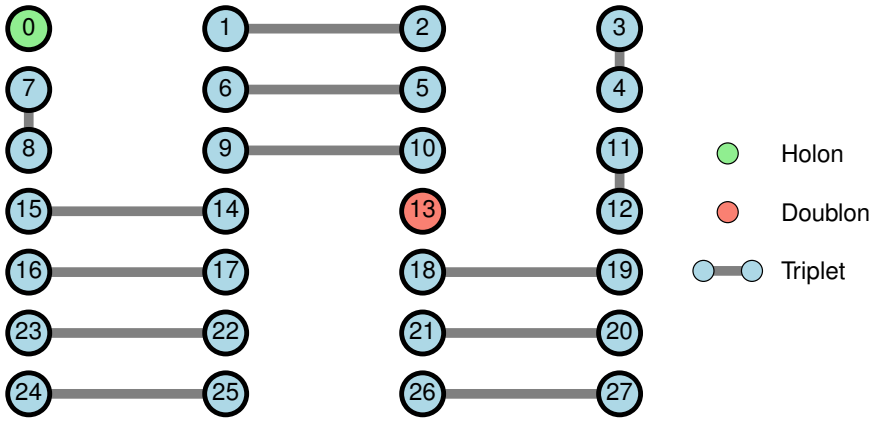


FIG. 21. Initial state layout. A holon (green) and a doublon (red) are prepared, with all other sites prepared into triplet pairs (blue), where each triplet connects sites adjacent in the JW encoding.

7. Compiler optimisations

We construct our circuits as described in Section C 3, and then pass them through the compiler provided by `pytket` [64]. In particular, we apply `optimisation_level = 2` of the compiler [65], the main effect of which is captured by a pass of `FullPeepholeOptimise()`, which synthesises one-, two- and three-qubit subcircuits. The compiler also recognises and removes redundant gates and gate sequences, such as self-inverse gates and diagonal gates immediately before measurements, and performs some optimisations specific to Clifford circuits [66]. The main effect of this compilation on our circuits is to remove one-qubit gates: as described in Section C 3, we optimise for two-qubit gate count such that there are no superfluous two-qubit gates, and therefore the compiler cannot find any further gates to remove. However, the subcircuits for each component, i.e., hopping, swap, and onsite terms, include numerous one-qubit gates that can be merged in between rounds of two-qubit gates, and so the compilation meaningfully reduces the one-qubit gate count.

In Table II we report the number of gates included in our circuits before and after compilation. Note that our circuits include `SWAP` gates, but on the ion-trap platform, these are performed in software by relabelling qubits, i.e., no quantum operation occurs, so we omit these from gate counts. Table I states that for N Trotter steps, our swap network construction admits a cost of $588N + 24$; we use $N = 4$ and Section C 6 shows that the initial state preparation incurs a further 39 two-qubit gates, giving $(588 \times 4) + 24 + 39 = 2415$ two-qubit gates overall.

	one-qubit	two-qubit
Uncompiled	11605	2415
Compiled	4576 – 4627	2410 – 2415

TABLE II. Number of gates in circuits before and after optimisation through pytket's compiler. There are minor variations due to the number of gates which can be compiled together following the pseudo-twirling routine.

	Error magnitude
Single-qubit gate error	3E-05
Two-qubit gate error	1E-03
Single-qubit leakage	1E-05
Two-qubit leakage	2E-04
Memory error	2E-04
Measurement crosstalk error	3E-06
SPAM error	1E-03

TABLE III. Hardware specifications of Quantinuum's H2-2 ion-trap device at the time of simulation.

8. Hardware specification

We ran our circuits on Quantinuum's `H2-2` ion-trap quantum processor in two batches: first between 19-23 June 2025, and second between 18-22 July 2025. The specified device performance at the time of running is listed in Table III, taken from [67]. We prepared time evolution circuits in intervals $\Delta t = 0.1$, from $T = 0.1$ to $T = 2.0$, i.e., 20 circuits per timeseries. With two values of onsite term, $U = 0, 4$, and 8 pseudo-twirling instances per circuit (Section D 1), in total we ran $20 \times 2 \times 8 = 320$ distinct circuits, each for 160 shots. Each pseudo-twirling instance was executed in a random order. The list of pseudo-twirling instances was split into two halves and each half was executed on a different day. Given the depth of our circuits, and the gatespeed of ion-trap processors, each shot took ~ 5 seconds, for a total runtime on the quantum device of $5 \times 20 \times 160 = 16000$ seconds ≈ 4.5 hours. Additionally, there were queues to access the device, as well as compilation time, so that the experiment took several days to run in practice.

Appendix D: Error suppression, error mitigation, and experimental analysis

Error suppression and mitigation are important to improve the accuracy of results obtained from a noisy quantum computer. The extremely restrictive constraints on the number of shots available in an experiment on ion-trap hardware mean that many techniques are not available or appropriate. Here we use a technique known as training with fermionic linear optics (TFLO) [28], implemented using a variant which uses very few additional shots, together with Gaussian process regression (GPR) for further cleanup of the results. In addition, we implement a technique known as Pauli pseudotwirling to suppress systematic errors and facilitate error mitigation.

We compare the performance of these methods with raw data in Figure 22, Figure 23 and Figure 24. We see that usually the combination of TFLO and GPR significantly reduces the error, compared with the unmitigated results. Further, we see that the error achieved by TDVP increases over time, as expected, and that this error cannot be explained by Trotter error alone. By contrast, the error achieved by our experimental results does not significantly increase with time, until the latest times (which we attribute to Trotter error). We expect that a significant portion of the error remaining in our results after error mitigation with TFLO can be explained by statistical noise, which is mitigated by GPR. Next, we describe these techniques in detail.

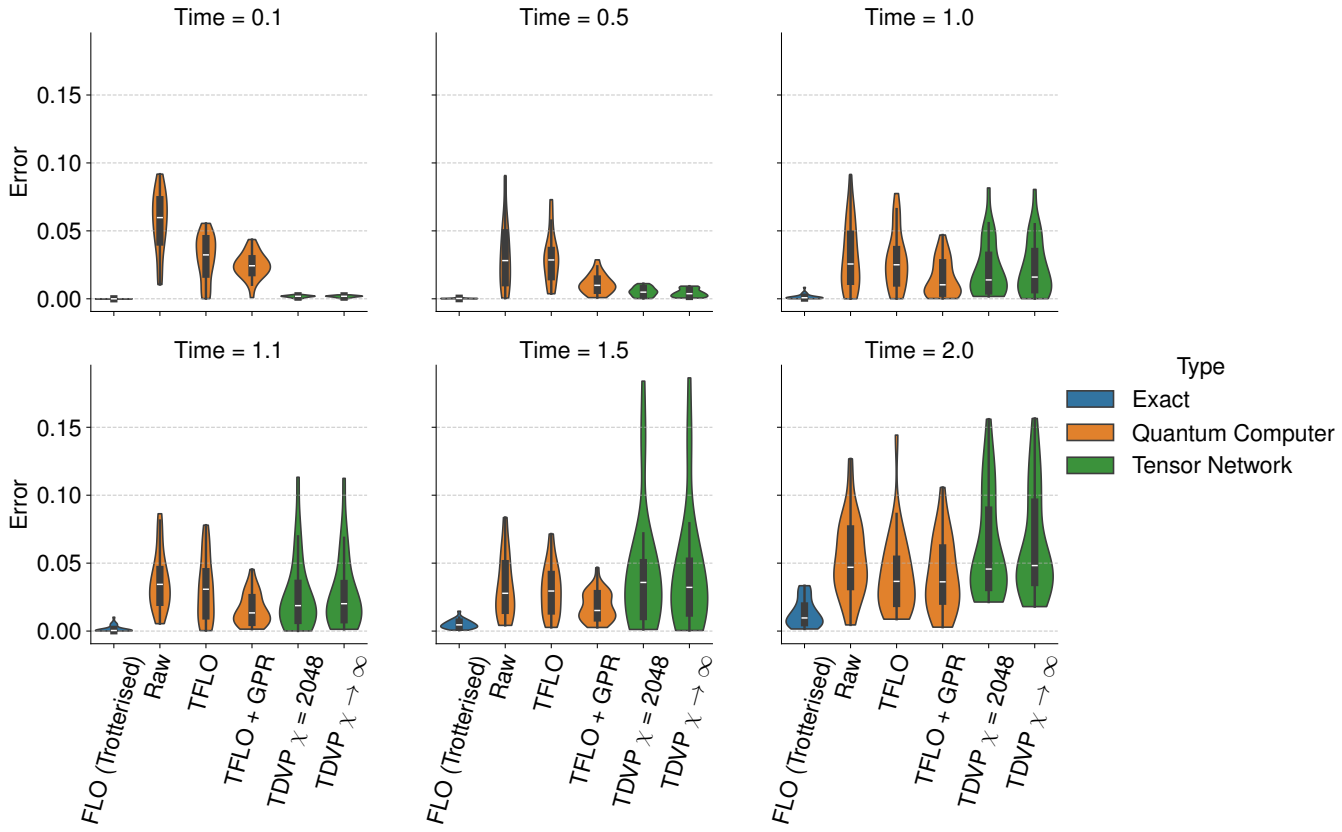


FIG. 22. Error profile for the doublon observable compared with exact dynamics over time, computed through raw hardware data, mitigated hardware data (FLO, GPR), and TDVP simulation along with extrapolated TDVP results.

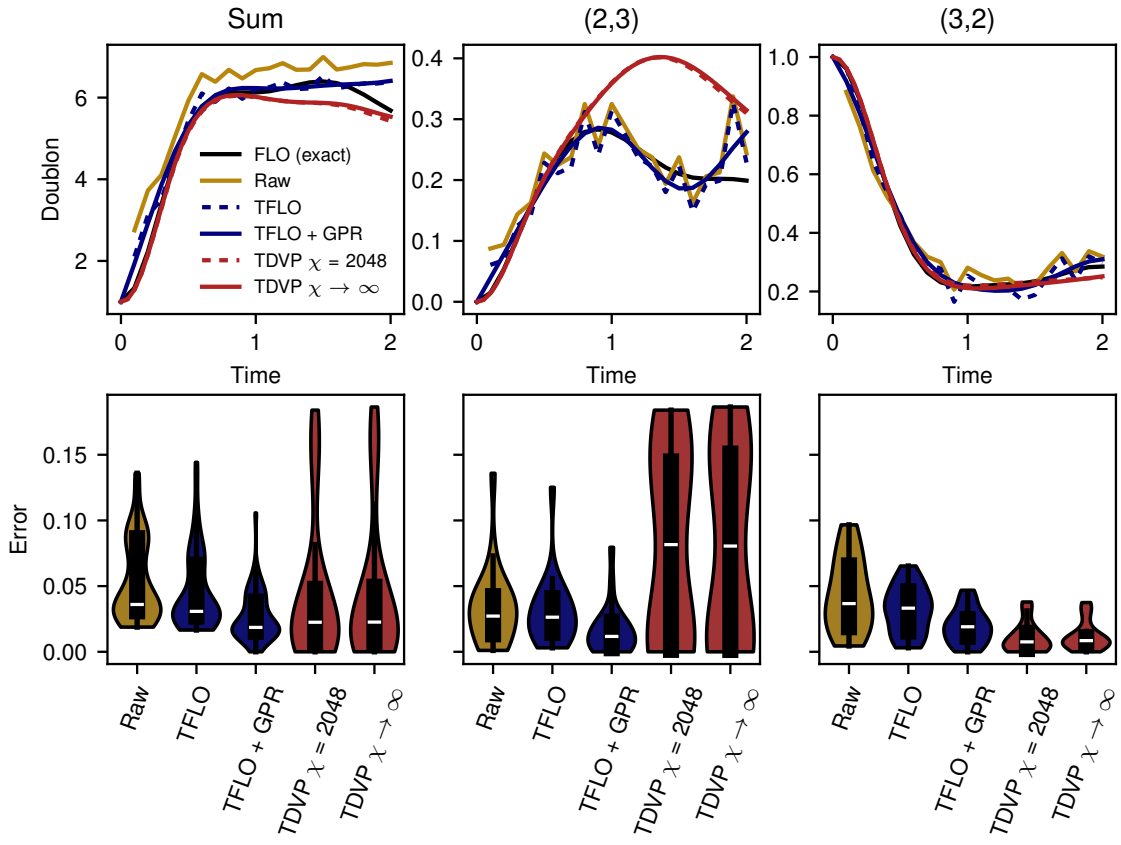


FIG. 23. Dynamics of doublon observable for the raw, mitigated and TDVP-simulated data at $U = 0$, along with corresponding errors with respect to exact FLO simulation. The left panel sums the doublon over the entire lattice. The central (right) panel highlights a particular site where we see the worst (best) performance for TDVP.

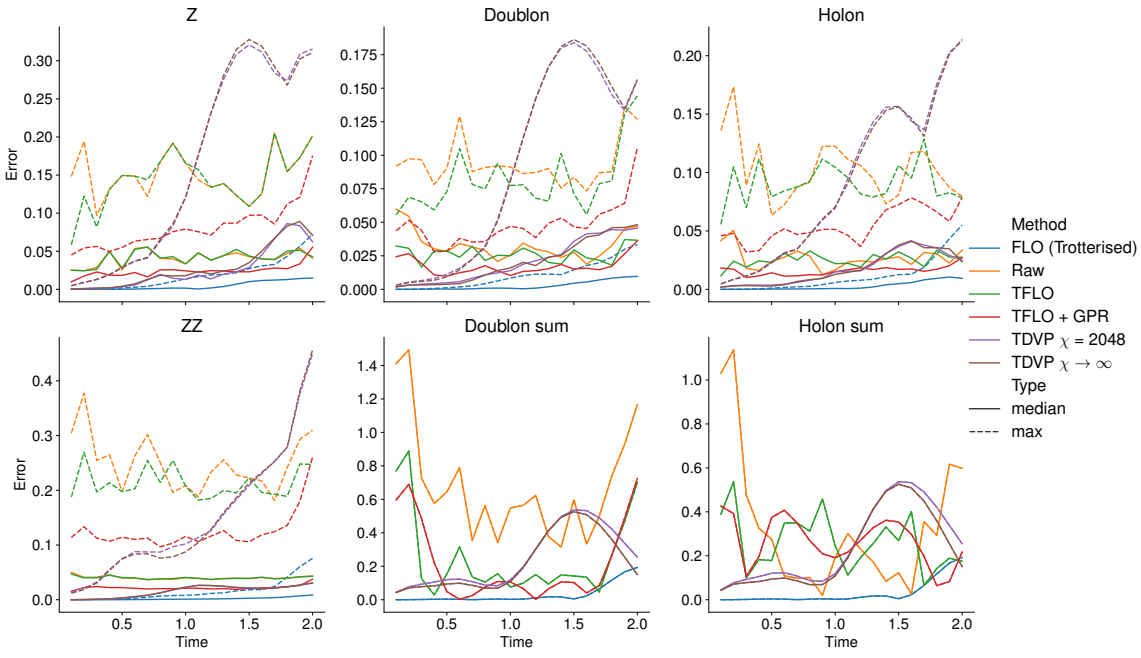


FIG. 24. Median and maximum errors for all times for a range of types of observables.

1. Pauli pseudo-twirling

Pauli twirling [68] is a well-known and effective method to suppress errors in quantum circuits [69]. The theoretical underpinning of this method is that in many realistic quantum hardware platforms, 2-qubit gates are the primary source of error, and the native 2-qubit gate is a Clifford gate (such as a CNOT or CZ gate). Each 2-qubit gate is sandwiched by a random choice of Pauli matrices picked from those for which, in principle, the ideal gate would be left unchanged. However, the approximate gate actually implemented by the hardware will not remain unchanged. Averaged over a number of instantiations, this tends to suppress systematic errors. Indeed, one can show that this process replaces coherent errors with incoherent errors, which facilitates certain error-mitigation procedures. Further, this procedure will have a dynamical decoupling effect and reduce other systematic errors occurring in the circuit.

Here, the native 2-qubit gate we execute is not a Clifford gate, but a R_{zz} gate. Nevertheless, we can still carry out a more limited version of this procedure, known as Pauli pseudo-twirling [26]. Instead of wrapping the R_{zz} gate between Pauli matrices that leave it unchanged, it is conjugated by random two-qubit Paulis and the sign of the rotation angle in $R_{zz}(t) = e^{-\frac{i}{2}ZZt}$ is either left unchanged or flipped, depending on whether the two-qubit Pauli commutes or anti-commutes with ZZ. Again, this procedure leaves the ideal circuit unchanged but helps shape the noise channel in the presence of noise. Similar procedures have been found to be effective on superconducting qubit [26] and ion-trap [27] quantum hardware platforms.

To implement this in our experiment, we simply traversed each circuit, for each $R_{zz}(t)$ gate sampled a random two-qubit Pauli P, and replaced it with $PR_{zz}(t)P$ (or $PR_{zz}(-t)P$ if ZZ and P anti-commute). We sampled 16 such twirling instances for each logical circuit (corresponding to different values of t and U). Note that the introduction of the additional two-qubit Paulis does not increase the gate count of the final compiled circuits, because runs of subsequent single-qubit gates get compiled into a single U_3 gate.

2. Symmetry averaging

As explained in Section B, the initial state and Hamiltonian are invariant under the spin-reflection symmetry $Rc_{i,\uparrow}R^\dagger = c_{i,\downarrow}$. This symmetry acts analogously on the Z operators, i.e. $RZ_{i_1,\sigma_1}\cdots Z_{i_n,\sigma_n}R^\dagger = Z_{i_1,\overline{\sigma}_1}\cdots Z_{i_n,\overline{\sigma}_n}$. This allows us to increase the effective shot-count for the expectation values of all multi-qubit Z strings and expectation values derived from them by averaging each multi-qubit Z expectation value with its spin-reflected partner.

This is also the only obvious, simple symmetry of the initial state and Hamiltonian, because the irregular arrangement of triplets in the initial state breaks the translation and reflection symmetries of the lattice.

3. Training with fermionic linear optics (TFLO)

TFLO [28] is a method which takes advantage of the well-known fact that quantum circuits that contain only fermionic linear optics (FLO) operations can be efficiently simulated classically [29]. In some quantum algorithms, such as ours, many of the gates in the quantum circuits to be executed are FLO operations, so it is natural to expect that the behaviour of the "true" circuit in terms of errors should be similar to the quantum circuit that one would produce by removing all FLO operations from that circuit. Therefore, for any given observable, one can prepare a set of pairs of noisy and exact expectation values for that observable and use these to infer a map from noisy to exact expectation values. The hope is that applying this map to the corresponding quantum circuit, which is not FLO, will achieve similar results in terms of reducing the error in this observable.

Here, there is a particularly direct connection between the circuit that we wish to execute and an exactly simulable model. Every gate in the quantum circuit for simulating time-dynamics of the Fermi-Hubbard model for $U = 0$ is FLO, and further, the structure of the circuit is exactly the same as the structure of the quantum circuit for $U \neq 0$, with the only difference being that for $U = 0$, there is no time evolution by onsite terms.

This connection has previously been used in the context of VQE with the Hamiltonian variational ansatz to reduce errors in ground state simulation of the Fermi-Hubbard model [25]. In that work, training data was produced by choosing random parameters for the variational ansatz for many choices of parameters. Here, we do not have the shot budget to do this. So instead, we use the time-dynamics experimental results themselves as training data. We produce a set of samples for $U = 0$ for the same set of time-points as for $U = 4$, and aim to use this to infer a function for mapping noisy to exact observables. It is important that the quantum circuits for these two cases look as alike as possible, so that errors behave as similarly as possible. We therefore kept the same gate and fermionic swap network structure in the $U = 0$ circuit as the $U = 4$ case, by including the onsite terms with an evolution parameter of 0.

An additional complication we face is that the full quantum circuit we execute for $U = 0$ is not exactly simulable classically using standard FLO methods due to the initial spin triplet state preparation step. Thus, we refer to circuits of this form as "NearFLO" throughout. Nevertheless, we can implement an efficient method for simulating NearFLO circuits classically in the case where one wishes to compute a low-weight observable (see Section E). To mitigate errors in high-weight observables, we

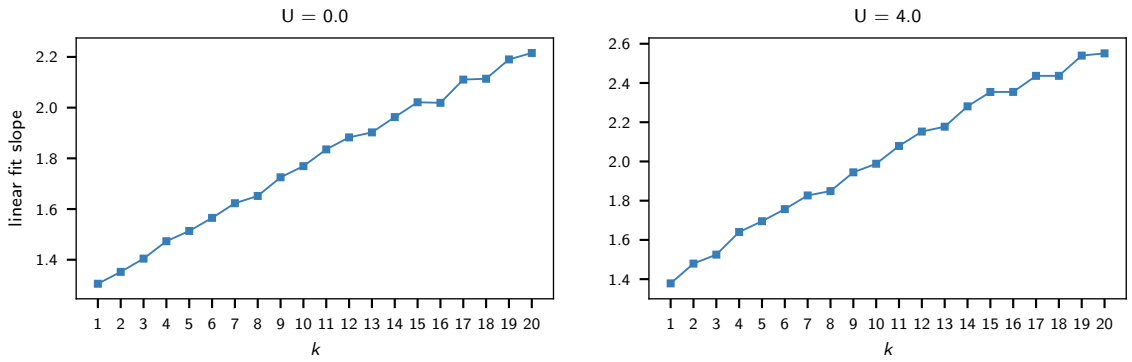


FIG. 25. Slope of TMPS fit for $S(k)$.

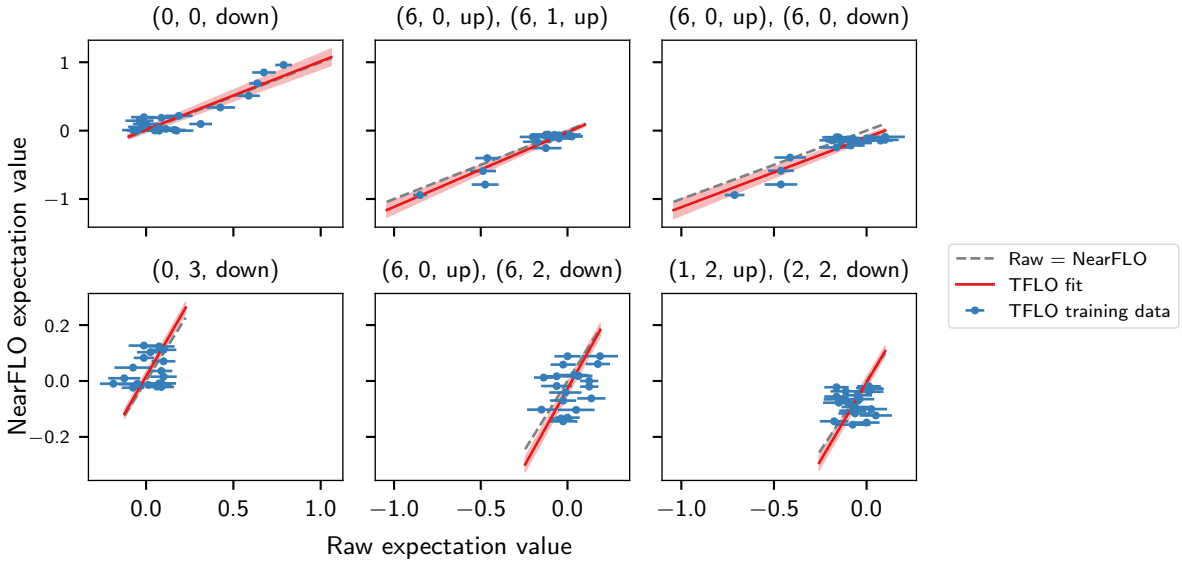


FIG. 26. A few examples of the linear fits performed for TFLO error mitigation. The top row shows single- and two-qubit Z expectation values between qubits that start out at ± 1 and are thus easy to fit to, while the bottom row shows examples of single- and two-qubit Z expectation values that start out at 0 and thus require a regularization before performing a successful fit.

use an alternative technique. Tensor network (TDVP) simulations are expected to achieve a high level of accuracy for short evolution times, and can be used to produce samples in the computational basis, and hence to estimate arbitrary observables. Therefore, we can use TDVP at short times to produce training data, on which the same fitting process as standard TFLO can be used. This is additionally justified by our observation that experimental errors for $U = 0$ do not get significantly worse for long times (Figure 22). We called this technique training with MPS (TMPS). For a given observable, we choose the time interval to be used to produce training data such that a good convergence in the bond dimension is observed. We can use the slope of the fit to get a sense of the extent to which errors affect high-weight observables. We observe (see Figure 25) that, as expected, the slope increases exponentially with the weight of the observable, but at a slow exponential rate. Moreover, Figure 27 shows that mitigating the same observable via TFLO and TMPS produces compatible results.

We find (see Figure 26) that a simple linear fit is often, though not always, a good model for the map between noisy and exact data. We use ordinary least-squares to perform this linear fit. We experimented with a more complex fitting function, which included dependence on the evolution time t, but we did not find that this significantly improved the quality of our results, as measured by the accuracy at $U = 0$. We attribute this to the hardware error model not significantly depending on t. Given the risk of overfitting, we chose not to use this model.

Due to the triplets in the initial states, many observables vary little over time. Additionally, due to the relatively low shot count, the statistical uncertainty on all observables is relatively large (on average ± 0.08). This means that for many observables, the signal-to-noise ratio (SNR) is too small to perform a well-controlled least-squares fit. To still be able to perform TFLO on these

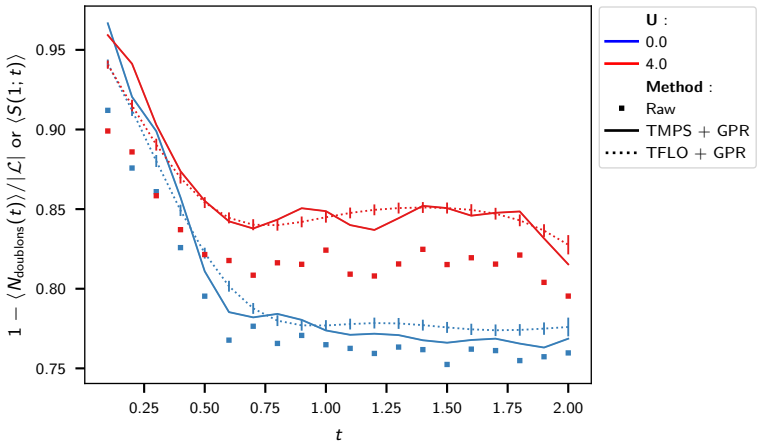


FIG. 27. Comparison between $1 - \langle N_{\mathrm{doublons}}(t)\rangle/|\mathcal{L}|$ and $\langle X(t)\rangle$. The agreement between the solid lines, obtained using TMPS, and the dotted lines, obtained using TFLO, shows that TMPS produces results compatible with TFLO.

observables, we employed a two-stage fitting procedure. In the first stage, we identified all observables with a sufficiently large SNR and performed an ordinary least-squares fit to find the best TFLO rescaling parameters. In the second stage, we grouped all observables by the number of qubits they act on and whether they act on multiple Fermi-Hubbard sites. For each group, we computed a prior distribution over the optimal fit parameters. This prior distribution was then used (per group) to perform a linear least squares fit with regularisation to obtain well-behaved fit parameters even when the SNR of the time series was low.

4. Postselection and using other occupation number sectors

Every gate in the quantum circuits that we execute preserves fermionic occupation number within each spin sector, which implies that we can detect errors which alter these quantities and, if we choose, postselect to only retain samples for which these quantities are correct [25]. However, in our experiments, the number of shots taken is heavily constrained, so it is not obvious whether postselection will be advantageous, because restricting the samples to a smaller subset tends to increase statistical error. Indeed, for $U = 0$ we observe that postselection on occupation number makes errors worse due to the low number of shots retained and increases statistical noise (see Figure 28). On the other hand, we see that the Hamming weight in each spin sector tends to vary very little from the expected values (see Figure 29). This provides a simple metric for the quality of our results, and suggests that local observables should tend to vary only slightly from their expected values. For example, local densities (Z operators) would be expected to vary by at most $\sim 1/56 \approx 0.018$ compared with their true values.

5. Gaussian process regression

As a final stage in our postprocessing of experimental data, we implement Gaussian process regression (GPR) smoothing for all time series. Briefly, a Gaussian process (GP) [30] is a probability distribution over functions. It is formulated as a marginal distribution over any finite set of function values such that the joint probability distribution for this finite set of function values is a multivariate Gaussian whose mean and covariance matrix only depend on the evaluation points and some hyperparameters encoded in what is called the kernel of a GP. This kernel can encode various things about the sampled functions, such as (n-times) differentiability, periodicity, or the correlation length l on which it varies. A good overview of what can be encoded into the kernel and how is presented in [30, Chap. 4].

To use this probability distribution over functions for smoothing and fitting, a simple Bayesian procedure can be used. The kernel of the GP defines a prior distribution over functions, the noisy observations of the time series play the role of new evidence, and the application of Bayes' theorem then yields a posterior distribution over functions. The mean of this posterior distribution is then the (n-times) differentiable / periodic / function with correlation length l that is most likely given the observed data. And the diagonal of the covariance matrix of the posterior distribution corresponds to the one-sigma confidence interval of the posterior time series.

We used the Python implementation of GPR regression in `scikit-learn` [70] with a squared-exponential kernel with length scale $l = 0.6$ and variance $\sigma = 1.5$. These hyperparameters were determined by maximizing the marginal likelihood of l and σ given the observed time series for a variety of variables. We then found that the optimal hyperparameters were narrowly

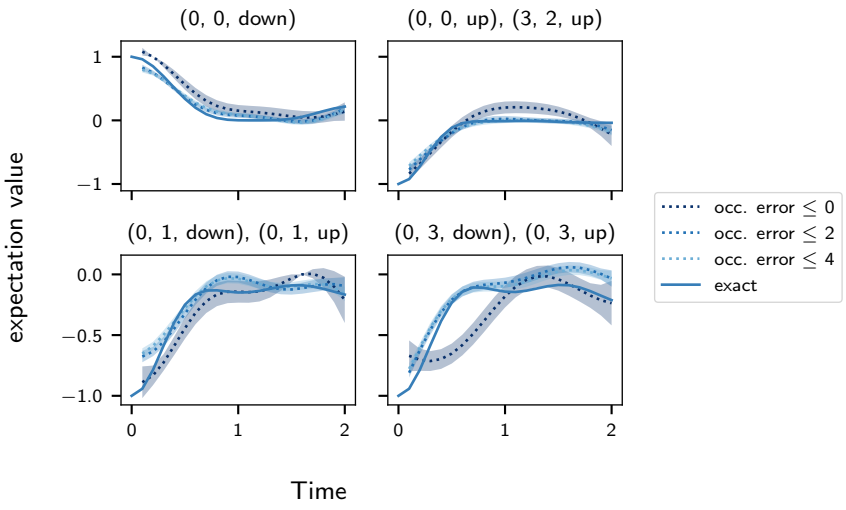


FIG. 28. The effect of post-selection on expectation values for a few examples of single and two-qubit Z observables. This is done using the $U = 0$ time series for exact comparison data and with GPR smoothing to decrease statistical noise. We observe that the most aggressive postselection on only the correct occupation numbers is the noisiest because of the small number of samples retained. When retaining samples with occupation error ≥ 2, the time series don't change anymore due to the fact that the vast majority of samples have 1 or 2 occupation errors.

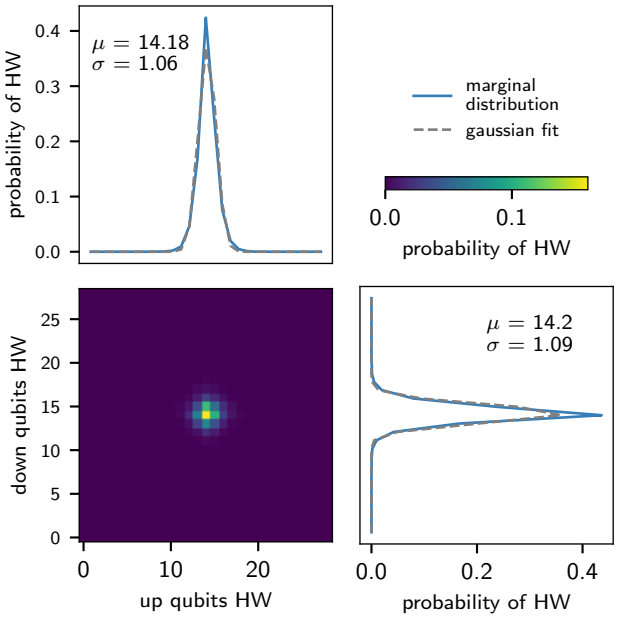


FIG. 29. The distribution of Hamming weights (HW) on the up and down qubits is well described by a Gaussian centered at $(14, 14)$, i.e., the correct Hamming weight in the absence of hardware errors. 19% of shots have exactly Hamming weight $(14, 14)$.

clustered around $l = 0.6$ and $\sigma = 1.5$ and thus fixed these values throughout all later analyses. Fixing the hyperparameters to the same values for all observables instead of determining them per observable also helped to avoid outliers due to sampling noise.

The error propagation procedure reported in Section D 6 gives error bars on the input data to the GPR smoothing procedure. But, by adding a Gaussian white noise kernel to the overall kernel, the average noise strength also becomes a parameter that is learnable through the hyperparameter optimisation procedure described in the previous paragraph. Fortunately, the error bars obtained through error propagation broadly agreed with those learned through hyperparameter optimisation in the GPR process. This gives further confidence that the approximations necessary for the error propagation are valid.

6. Error bars

The GPR error bars shown in the figures in the main text were all computed using Gaussian error propagation and error bars of the initial raw values computed from the shots were computed using

$$\sigma^2_{\text{raw}} = \frac{1}{N_t(N_t-1)}\sum_{i=1}^{N_t}(\hat{\mu}_i - \mu)^2 + \frac{1}{N_t^2 N_s}\sum_{i=1}^{N_t}\sigma_i^2 \tag{D1}$$

where N_t is the number of twirling instances, N_s the number of shots taken per twirling instance, μ_i and σ_i^2 are the per twirling instance sample mean and sample variance and $\mu = \frac{1}{N_t}\sum_{i=1}^{N_t}\mu_i$ is the total mean.

a. *Pseudo-twirling and error bars*

When computing error bars in conjunction with (pseudo)-twirling, some care must be taken to account for the iterated expectations correctly. One might be tempted to simply collect all $N_t \times N_s$ shots into one set, compute the expectation of the observable O as

$$\hat{\mu}_O = \frac{1}{N_t \times N_s}\sum_{i,j=1}^{N_t,N_s} O(x_{i,j}) \tag{D2}$$

and the variance of $\hat{\mu}_O$ in the usual way as

$$\hat{\sigma}^2_{\hat{\mu}_O} = \frac{1}{N_t \times N_s(N_t \times N_s - 1)}\sum_{i,j=1}^{N_t,N_s}(O(x_{i,j}) - \hat{\mu}_O)^2. \tag{D3}$$

But unless $N_s = 1$ or all twirling instances are actually the same, this will underestimate the error. An easy example to see this is to assume that we have very few twirling instances N_t from which we take many shots N_s, and that the different twirling instances produce very different distributions. Then the naive Equation (D3) will be very small, even though we have only seen the data from very few twirling instances and they all give very different estimates for μ_O.

To get an accurate estimate $\hat{\sigma}^2_{\hat{\mu}_O}$ for the variance of $\hat{\mu}_O$, we have to correctly account for the contribution to the variance coming from sampling different twirling instances—the first term in Equation (D1)—and the contribution coming from shot noise for each circuit—the second term in Equation (D1). This can be done by employing the law of total variance. Carefully going through the derivations then gives Equation (D1).

b. *Verification via bootstrapping*

For error propagation, we employed the `uncertainties` Python package [71], which implements automatic Gaussian error propagation for many basic mathematical functions. However, for Gaussian error propagation to be valid, we have to assume that (i) the input distributions are all Gaussian; (ii) all data transformation functions are (or at least well-approximated by) affine linear functions. In addition to these constraints, the `uncertainties` package only tracks and propagates the individual uncertainty of each variable, but not correlations. The most straightforward alternative to Gaussian error propagation is bootstrapping [72, Chap. 8]. Very briefly, bootstrapping is a procedure for estimating the distribution (or properties of the distribution) of an estimator (here, our mitigated and processed observable values) by replacing the full distribution of the ingoing random variable (that we don't know) with the empirical distribution that we obtained from the samples we actually have.

Similar to the discussion in Section D 6 a, one has to be careful when implementing the bootstrap method in conjunction with pseudo-twirling. The correct scheme is to first sample N_t circuits with replacement from the pool of twirling instances and then sample N_s shots with replacement from each of the sampled circuits. This dataset can then be piped through the whole error mitigation and data analysis pipeline. Repeating this procedure sufficiently many times allows us to estimate the standard deviations of the final mitigated observables. In Figure 30 we show that the error bars obtained via bootstrapping agreed remarkably well with those obtained via the computationally much cheaper Gaussian error propagation. We thus decided to use Gaussian error propagation for all error bars reported here.

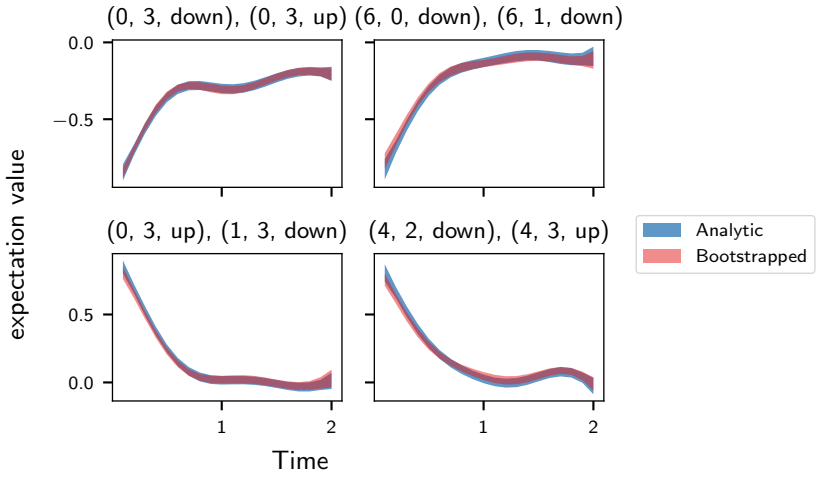


FIG. 30. Comparison of error bars computed via bootstrapping vs Gaussian error propagation for a random choice of 4 observables.

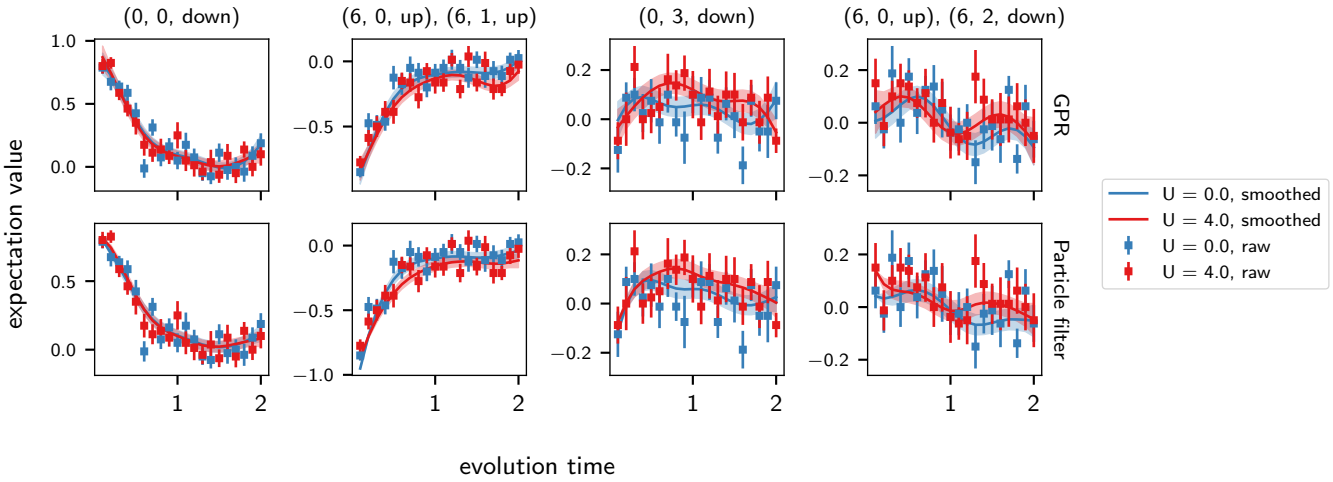


FIG. 31. Comparison of smoothing with Gaussian process regression (top) and particle filters (bottom) for a few single and two-qubit Z timeseries. We find that the smoothed time series are very similar, even though the assumptions behind GPR (Gaussianity of input error bars) are not met.

7. Crossvalidation via particle filters

In addition to GPR smoothing, we also experimented with a particle filter (PF)-based smoothing method to postprocess all time series. While GPR assumes Gaussian noise in the observations, the measurement noise in our setting is not strictly Gaussian. To account for this, we employ particle filtering as an additional validation method, which can accommodate non-Gaussian noise models. A particle filter [73] is a Bayesian sequential Monte Carlo method that represents probability distributions over hidden states with a large ensemble of weighted samples, called particles. Each particle evolves forward in time under a stochastic state–space model and is reweighted by the likelihood of the observed data. Through repeated propagation and resampling, the PF generates full latent trajectories consistent with all observations, providing a smoothed estimate of the underlying latent signal. As shown in Figure 31, the smoothed time series obtained using PF are in close agreement with those obtained using GPR, despite the difference in the underlying noise assumptions. For a more in-depth review on particle filter with backward smoothing, refer to Sections 4 and 5 of Ref. [74].

We implemented the PF from scratch in Python, using two passes with ancestry-based backward smoothing. At each time step, the hidden state consists of the signal value and its rate of change (velocity), which evolve under a stochastic update with

Gaussian noise and exponential velocity decay. Each particle is then reweighted according to the likelihood of the observed data at that time step, and then the resampling step systematically prunes unlikely particles and replicates likely ones, maintaining a representative ensemble. After this forward run, we perform backward smoothing using the particle ancestry. Each resampled particle at a given time step stores the index of its parent from the previous time step, which defines an ancestry tree. By tracing these links backward from a randomly chosen particle at the final time step, we generate full state trajectories consistent with all observations. Averaging over many such trajectories yields the smoothed time series, while the standard deviation across them quantifies the uncertainty.

Similar to GPR, we fixed the particle filter hyperparameters across all observables to maintain consistency and avoid overfitting to sampling noise in individual series. We used 6000 particles, while the remaining parameters (e.g., noise scales, initial uncertainties, velocity decay) were chosen based on test signals with known values. At the same time, they were kept sufficiently general to provide reliable smoothing across all observables.

Appendix E: Classical simulation of non-Gaussian circuits

The initial state considered in this work is a triplet covering alongside two lonely sites (doublon and holon). We therefore refer to it as a non-Gaussian or magic state [21, 75–80] because it cannot be generated by a FLO unitary (more details in Section E 1), and thus in general has to be treated as a linear combination of the constituent Fock states, the number of which grow exponentially with the number of triplets present.

Recall that fermionic Gaussian states are those of non-interacting fermions and hence correspond to the $U = 0$ regime, which we will work in for the remainder of this section.

1. Non-Gaussian observables

Recall that the initial state consists of lonely sites, which are either holes or doublons, and physical triplets across pairs of sites (a, b). The triplet creation operator $\frac{c^\dagger_{a,\uparrow}c^\dagger_{b,\downarrow}+c^\dagger_{a,\downarrow}c^\dagger_{b,\uparrow}}{\sqrt{2}}$ is not a fermionic Gaussian state and cannot be generate by a FLO unitary [29]. Thus, the simulation of observables in theory will scale exponentially with the number of triplets. However, we note that the initial state can be written as a product state across N_{triplets} triplets and N_{lonely} lonely sites according to Equation (E1)

$$|\psi\rangle_{\text{init}} = \bigotimes_{k=1}^{N_{\text{triplets}}} |\text{triplet}\rangle_{s_k} \bigotimes_{l=1}^{N_{\text{lonely}}} |\text{lonely}\rangle_{s_l} \tag{E1}$$

where s_k denote the pair of sites participating in the k^{th} triplet and s_l is a single lonely site index. The observables required for TFLO are the single and two-point Pauli Z correlators, $\langle Z_i\rangle$ and $\langle Z_iZ_j\rangle$ $\forall i, j$ fermionic mode indices (i.e,. site and spin indices are conglomerated). Here we work instead with the number operators $\langle \hat{n}_i\rangle$ and $\langle \hat{n}_i\hat{n}_j\rangle$ but there is a trivial conversion between $\hat{n} \leftrightarrow Z$ via $\hat{n} = \frac{1-Z}{2}$. This set of densities and correlators is enough to perform TFLO corrections for all observables considered in the main text except for the Wilson lines. As we will discuss later in this section, arbitrary weight observables in theory would scale exponentially with N_{triplets}, leading to an algorithm which is cumbersome and hence not implemented for the Wilson loops.

a. *Densities*

The observables $\langle \hat{n}_i\rangle, \forall i$ where i denotes a fermionic mode can be computed as follows

$$\langle \psi_{\text{init}}(t)|\, c_i^\dagger c_i\, |\psi_{\text{init}}(t)\rangle = \langle \psi_{\text{init}}|\, e^{iHt} c_i^\dagger c_i e^{-iHt}\, |\psi_{\text{init}}\rangle \tag{E2}$$

$$= \sum_{\alpha,\beta} R_{\alpha,i}R^*_{\beta,i}\, \langle \psi_{\text{init}}|\, c_\alpha^\dagger c_\beta\, |\psi_{\text{init}}\rangle \tag{E3}$$

$$= \sum_{\alpha} |R_{\alpha,i}|^2 \tag{E4}$$

where H is the single band Hamiltonian defined in Equation (A1) and $R = e^{-iht}$ where h is the adjacency matrix of the lattice $\mathcal{L}$. Due to the structure of the initial state, the coefficients α and β must be equal for a non-vanishing value of $\langle \psi_{\text{init}}|\, c_\alpha^\dagger c_\beta\, |\psi_{\text{init}}\rangle$, since flipping two bits within or across sites result in an orthogonal state.

b. *Density-Density correlators*

The observable of interest is

$$\langle \psi_{\text{init}}(t)|\, c_i^\dagger c_i c_j^\dagger c_j\, |\psi_{\text{init}}(t)\rangle = \langle \psi_{\text{init}}|\, e^{iHt} c_i^\dagger c_i c_j^\dagger c_j e^{-iHt}\, |\psi_{\text{init}}\rangle \tag{E5}$$

$$= \sum_{\alpha,\beta,\gamma,\eta} R_{\alpha,i}R^*_{\beta,i}R_{\gamma,j}R^*_{\eta,j}\, \langle \psi_{\text{init}}|\, c_\alpha^\dagger c_\beta c_\gamma^\dagger c_\eta\, |\psi_{\text{init}}\rangle \tag{E6}$$

Note that since ψ_{init} is not a fermionic Gaussian state, one cannot simply compute this with a determinant/pfaffian. Hence, we need to consider the exponential $(2L)^P$ mode combinations in the sum. Here, $P = 4$ is the fermionic operator weight. However, for the special case of $P = 4$, the computational complexity can be reduced to scale as $O(L^2)$ due to the structure of the initial state.

The term $c^\dagger_\alpha c_\beta c^\dagger_\gamma c_\eta$ flips four qubits, turning $|\psi_{\text{init}}\rangle \to |\psi'_{\text{init}}\rangle$. The structure of a triplet state over sites a and b is given by

$$\frac{1}{\sqrt{2}}(|1_{i,\uparrow}0_{i,\downarrow}0_{j,\uparrow}1_{j,\downarrow}\rangle - |0_{i,\uparrow}1_{i,\downarrow}1_{j,\uparrow}0_{j,\downarrow}\rangle) \tag{E7}$$

and the only way to have a non-vanishing overlap with itself is to flip all four qubits. For a lonely site, the only non-vanishing overlap with itself is to flip nothing. Thus, out of all the possible combinations of $1, 2, 3$ and 4 distinct/unique values of α, β, γ and η, for a non-trivial overlap one only needs to consider the cases where the indices land on a single or a pair of physical sites.

a. Four distinct indices All non-trivial combinations of distinct α, β, γ and η must be within a single triplet. The contribution to the total correlator becomes

$$\begin{aligned}\langle n_i n_j\rangle_{\text{4 distinct}}(t) = \sum_{(i,j)}^{N_{\text{triplets}}} \Bigg(&\sum_{\alpha,\beta,\gamma,\eta\in\text{perm}[i\uparrow,i\downarrow,j\uparrow,j\downarrow]} R_{\alpha,i}R^*_{\beta,i}R_{\gamma,j}R^*_{\eta,j} \\ &\times \frac{1}{2}\langle\Omega|\,(c^\dagger_{i,\uparrow}c^\dagger_{j,\downarrow} + c^\dagger_{i,\downarrow}c^\dagger_{j,\uparrow})^\dagger c^\dagger_\alpha c_\beta c^\dagger_\gamma c_\eta (c^\dagger_{i,\uparrow}c^\dagger_{j,\downarrow} + c^\dagger_{i,\downarrow}c^\dagger_{j,\uparrow})\,|\Omega\rangle \Bigg)\end{aligned} \tag{E8}$$

where $|\Omega\rangle$ is the vacuum Fock state. Perm denotes the permutations of the list and (i, j) denote the pairs of **site** indices forming the triplet covering.

b. Two distinct indices There are two unique cases here, $\alpha = \beta, \gamma = \eta$ or $\alpha = \eta, \beta = \gamma$, as only number operators are non-trivial. For both conditions, the two distinct mode indices can be

1. Within the same triplet.
2. Across two triplets.
3. Within a lonely site.
4. Across two lonely sites.
5. Between a triplet and a lonely site.

(E9)

The total correlator becomes

$$\begin{aligned}\langle n_i n_j\rangle_{\text{2 distinct}}(t) = \sum_{(i,j)} \Bigg(&\sum_{\alpha,\gamma\in\text{perm}[i\uparrow,i\downarrow,j\uparrow,j\downarrow]} |R_{\alpha,i}|^2|R_{\gamma,j}|^2\,\langle\Omega|\,\mathcal{C}^\dagger_{i,j} n_\alpha n_\gamma \mathcal{C}_{i,j}\,|\Omega\rangle \\ &+ \sum_{\alpha,\beta\in\text{perm}[i\uparrow,i\downarrow,j\uparrow,j\downarrow]} R_{\alpha,i}R^*_{\beta,i}R_{\beta,j}R^*_{\alpha,j}\,\langle\Omega|\,\mathcal{C}^\dagger_{i,j} n_\alpha(1-n_\beta)\mathcal{C}_{i,j}\,|\Omega\rangle \Bigg)\end{aligned} \tag{E10}$$

where (i, j) denote the pairs of **site** indices across all the conditions outlined in Equation (E9). $\mathcal{C}$ is the creation operator of the initial state type across sites i and j.

c. One distinct index This is similar to the case of the number operator discussed in Section E 1 a, i.e.

$$\langle n_i n_j\rangle_{\text{1 distinct}}(t) = \sum_\alpha \sum_i^{2L} |R_{\alpha,i}|^2|R_{\alpha,j}|^2\,\langle\psi_{\text{init}}|\,n_i\,|\psi_{\text{init}}\rangle \tag{E11}$$

where i now runs over all **modes**.

The case of three distinct indices is trivially zero because in $c^\dagger_\alpha c_\beta c^\dagger_\gamma c_\eta$, there is either a double creation or annihilation acting on the same mode. Finally,

$$\langle n_i n_j\rangle(t) = \langle n_i n_j\rangle_{\text{1 distinct}}(t) + \langle n_i n_j\rangle_{\text{2 distinct}}(t) + \langle n_i n_j\rangle_{\text{4 distinct}}(t) \tag{E12}$$

which scales as $O(L^2)$ from the combination of two sites.

c. *Alternative observables not implemented: arbitrary weight Pauli Z observables*

The speedups discussed in Section E 1 allow for fast computations of $\langle Z\rangle(t)$ and $\langle ZZ\rangle(t)$ correlators that do not scale exponentially with system size. In general, it will be helpful to compute arbitrary weight P observables of the form

$$\langle\prod_k Z_k\rangle(t) = \langle\psi_{\text{init}}|\,e^{iHt}\left(\prod_k^P Z_k\right)e^{-iHt}\,|\psi_{\text{init}}\rangle\,, \tag{E13}$$

should they be needed for more error mitigation, such as with TFLO. Here, we leverage the fact that the Pauli Z_k operator on mode k is a FLO unitary, i.e.

$$Z_k = -ie^{i\frac{\pi}{2}(1-2c_k^\dagger c_k)}, \tag{E14}$$

and expand the initial state into its Fock state decomposition,

$$|\psi_{\text{init}}\rangle = \sum_m \alpha_m \left(\prod_{i=1}^{W} c_{m_i}^\dagger\right)|\Omega\rangle \tag{E15}$$

where m_i runs over the modes making up the Fock state f with complex coefficient α_m and $|\Omega\rangle$ is the vacuum state. W is the hamming weight of all the Fock states making up $|\psi_{\text{init}}\rangle$

$$\langle\prod_k Z_k\rangle(t) = \sum_{m,m'} \alpha_{m'}^* \alpha_m \langle\Omega|\left(\prod_{i=1}^{W} c_{m_i'}^\dagger\right)^\dagger e^{iHt}\left(\prod_k^P Z_k\right)e^{-iHt}\left(\prod_{i=1}^{W} c_{m_i}^\dagger\right)|\Omega\rangle \tag{E16}$$

$$= \sum_{m,m'} \alpha_{m'}^* \alpha_m \left(\sum_{q_1,\ldots,q_W} (\prod_{i=W}^{1} \mathcal{R}_{q_i,m_i'}^*)\right)\langle\Omega| c_{qW}\ldots c_{q1} c_{m_1}^\dagger \ldots c_{m_W}^\dagger |\Omega\rangle \tag{E17}$$

$$= \sum_{m,m'} \alpha_{m'}^* \alpha_m \left(\sum_{q_1,\ldots,q_W} (\prod_{i=W}^{1} \mathcal{R}_{q_i,m_i'}^*)\right)\text{sgn}\left(\sigma_{q1,\ldots,q_W}^{m_1,\ldots,m_W}\right) \tag{E18}$$

$$= \sum_{m,m'} \alpha_{m'}^* \alpha_m \det\begin{pmatrix} (m_1) & \mathcal{R}_{m_1,m_1'}^* & \mathcal{R}_{m_1,m_2'}^* & \ldots & \mathcal{R}_{m_1,m_W'}^* \\ \vdots & & \ldots & \ldots & \vdots \\ (m_W) & \mathcal{R}_{m_W,m_1'}^* & \mathcal{R}_{m_W,m_2'}^* & \ldots & \mathcal{R}_{m_W,m_W'}^* \end{pmatrix} \tag{E19}$$

where $\mathcal{R} = e^{iht}\left(\prod_k^P -ie^{i\frac{\pi}{2}\text{diag}(-1)^k}\right)e^{-iht}$ is the $2L \times 2L$ matrix from the combination of the FLO time dynamics unitaries and Pauli Z tensor products. The values m_1 and m_W marked in grey denote the row indices; they are relative to each other and do not participate in the determinant. For example, if $m_2 - m_1 > 1$ but $m_1 + 1$ does not exist as an index, then the m_2 row is directly below the m_1 row. Therefore, the absolute values of $m_1, \ldots, m_W$ are not the actual row indices but give their relative position to each other so as not to wrongfully pick up an additional -1. This factor will not be a problem if the initial state was FLO, whereby the superposition is no longer necessary. Note that if the fixed values of $m_1, \ldots, m_W$ are monotonically increasing, this becomes the conventional definition of a determinant. $\text{sgn}(\sigma_A^B)$ is equal to $(-1)^{P_{AB}}$, where P_{AB} is the number of swaps to permute the set of indices A to match B.

For the triplet covering considered in the main text, $|\alpha_m| = (\frac{1}{\sqrt{2}})^{N_{\text{triplets}}}$. Thus, as equation Equation (E19) suggests, the cost of simulating any weight P Pauli Z observable is bottlenecked by the summation over $(4^{N_{\text{triplets}}})$ combinations of Fock states from the initial state, and not P. The total cost of performing TFLO corrections for Wilson loops such as that in Figure 10 will practically scale as $O(4^{N_{\text{triplets}}}) \times N_T \times N_{\text{loops}}$ for the $N_T = 20$ time steps taking in the experiment and there are $N_{\text{loops}} = 4182$ Wilson loops considered with Area = 12. Not only is this numerically demanding, but TFLO is not guaranteed to be successful since each Wilson loop's time dynamics signal may vary too little for a good linear map fitting. As a result, we omit attempting TFLO on the Wilson observables discussed in the main text.

2. Sampling from the triplet Magic state

The ability to sample bitstrings from the triplet state can also be beneficial. For example, one can compare the sampled distribution against hardware to understand noise. Alternatively, it can be used to compute observables such as those discussed in Section E 1 c in the non-Gaussian context, to be used as TFLO training data.

Several existing algorithms would be applicable to sample from the magic state. To simulate measurements from a pure FLO system, one can employ the qubit-by-qubit sampling scheme proposed by Terhal and DiVincenzo [29] with a naive time complexity upper bound of $O(N^4)$ for an N-mode measurement. This was subsequently improved upon by Bravyi and König [81, 82] to an upper bound of $O(N^3)$. Since the triplet covering initial state is a linear combination of FLO states but not FLO itself, naively applying the qubit-by-qubit sampling algorithm requires computations of overlaps of the form

$$\langle\psi_{\text{FLO}}(t)|\,\hat{O}\,|\phi_{\text{FLO}}(t)\rangle \tag{E20}$$

where keeping track of phases is important, as $\psi_{\text{FLO}}(t)$ and $\phi_{\text{FLO}}(t)$ are different FLO states, and O is some FLO operator/linear combinations of FLO operators. This can be done via the method developed by Bravyi and Gosset [83], which generalizes Wick's theorem to compute matrix elements of exactly this type of Gaussian state overlaps. Although the theory exists, sampling from the magic state was not implemented again due to the exponential cost arising from the number of triplets in the initial state. Alternatively, a gate-by-gate sampling method proposed by Bravyi *et al.* [84] or the sampling scheme proposed by Reardon-Smith *et al.* [78] are also viable but again hindered by the exponential cost of the triplet initial state. For these reasons, we omitted the implementation and analysis of magic state sampling.

Appendix F: Majorana Propagation Simulations

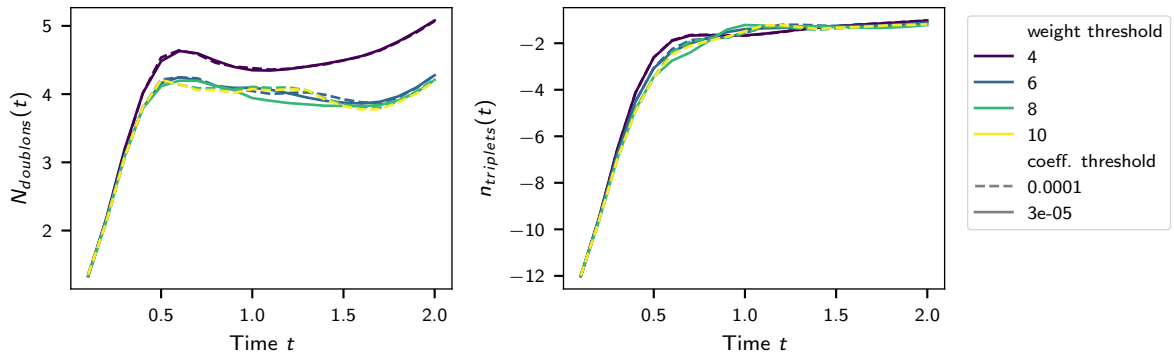


FIG. 32. Majorana propagation simulations time series for different lengths and coefficient truncation thresholds at $U = 4$. The left panel shows the number of doublons shown in Figure 2 of the main text, while the right panel shows the connected spin correlation function also shown in Figure 2 of the main text.

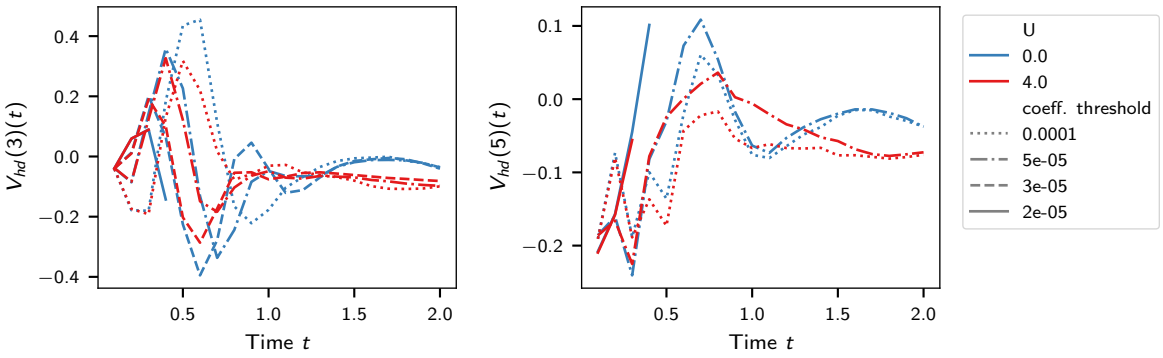
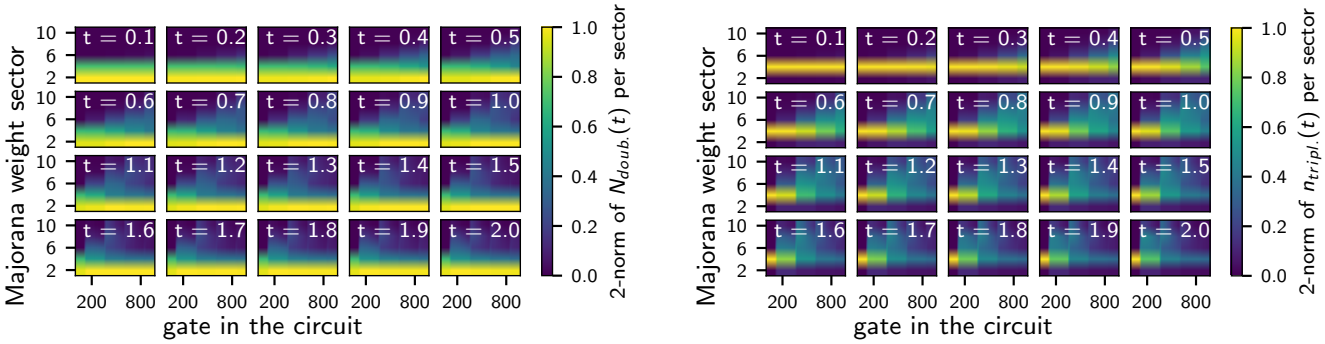


FIG. 33. Majorana propagation simulations time series of the open Wilson lines Equation (7) of length 3 and 5. The $U = 0$ simulation was run at weight threshold 12 (the maximum weight of that observable) and the $U = 4$ simulation at weight threshold 16. Even when lowering the coefficient threshold to 2×10^{-5} the results quickly diverge from the physically more sensible results reported in Figure 6.



(a) (b)

FIG. 34. Spreading of the observables over the Majorana weight sectors as they Heisenberg evolve through the circuit for number of doublons (left) and connected spin correlation function (right) at all simulated times for $U = 4$. Note that for the MP simulations we wrote the circuit entirely in the fermionic language, i.e. as a sequence of hopping and onsite evolutions instead of a hardware native sequence of single qubit and R_{zz} gates. This means the gate count for MP simulations is much lower than for the hardware circuits, even though they describe equivalent unitaries.

Observable	Total runtime
n_{triplets}	152 hours
N_{doublons}	22 hours
$n_{1,0,\downarrow}$	9 hours
$S^z_{0,2}S^z_{1,2}$	53 hours
$V_{hd}(3)$	243 hours
$V_{hd}(5)$	29 hours

TABLE IV. Total simulation cost for the Majorana propagation simulations reported. Note that the 29 hours reported for $V_{hd}(5)$ were stopped early when it became evident that these simulations would not converge to quantitatively accurate results.

Besides the tensor network simulations discussed in Section G, we also implemented the Majorana Propagation (MP) algorithm first proposed in [34]. It computes expectation values of observables by writing them as polynomials of Majorana operators, Heisenberg evolving these Majorana polynomials through the circuit and then computing the overlap of the Heisenberg evolved observable with the initial state. To keep the simulation feasible, the Majorana monomial is regularly truncated throughout the simulation by discarding terms whose coefficient is below a given coefficient threshold or whose Majorana weight is above a given weight threshold. For the latter truncation, it is important to note that the only gates in our circuits that increase the Majorana weight are evolution with the onsite terms, but that the majority of gates (FSWAP and hopping) leave the Majorana weight of each term unchanged. Furthermore, many observables of physical interest (in particular the triplet density Equation (5) and the number of doublons Equation (4)) have only low Majorana weight. This makes the Majorana Propagation algorithm a natural candidate to simulate the time series shown in Figure 2. Our implementation is largely inspired by the implementation of the related Pauli Propagation algorithm [85, 86] in Ref. [87] and is built on top of the `Yao.jl` quantum simulation framework [88].

One part of the Majorana propagation simulation that is not entirely straightforward is handling the triplets in the initial state. Naively, we could simply absorb the state preparation circuit Equation (C30) into the Heisenberg evolution and then compute the overlap with the computational zero state. But this would exponentially increase the number of terms in the Majorana polynomial (relative to the number of triplets) and would have prohibitive memory requirements. However, because we know that state preparation circuits are the last (in the Heisenberg picture) gates acting on the four qubits involved, we can discard all terms that are orthogonal to the zero-state on these four qubits. This implies that the presence of the triplets in the initial state does not increase the hardness of computing expectation values using MP.

To assess the accuracy of our Majorana propagation simulations we ran each simulation at various coefficient and weight thresholds up to a length threshold of 10 and a weight threshold of 3×10^{-5}. The results of these various simulations are compared in Figures 32 and 33. For N_{doublons} and n_{triplets}, we observe that decreasing the coefficient threshold has relatively little effect. Additionally, once we include the Majorana weight 6 sector, the time series changes little. This suggests that the time series of N_{doublons} and n_{triplets} presented in the main text are quantitatively accurate. On the other hand, for the two open Wilson lines $V_{hd}(3)$ and $V_{hd}(5)$, the time series vary wildly as the truncation parameters are changed, even for $U = 0$. This suggests that the MP simulations of $V_{hd}(3)$ and $V_{hd}(5)$ are not reflective of the exact values in the absence of truncation.

In Figure 34 we also visualise how the Majorana polynomials spread over the different Majorana weight sectors as they Heisenberg evolve through the circuit. The connected spin correlation function begins fully supported in the weight 4 sector and then spreads to the higher weight sectors, while the number of doublons is consistently supported in the weight 2 sector and spreads much less to the higher weight sectors. This indicates that the number of doublons is easier to simulate for MP than the connected spin correlation function. Note that for the observables in question, the simulations for $U = 0$ are essentially exact because the Majorana weight of each term in the Majorana polynomial is never increased, and the weight 2 and 4 sectors can be simulated without any need for truncation.

Appendix G: Tensor Network Simulations

We used a variety of tensor network methods to attempt to simulate the experiments we performed. These included the time-dependent variational principle (TDVP), direct circuit contraction, and fermionic matrix product states.

1. Time Dependent Variational Principle

A common algorithm for the simulation of the time evolution of quantum systems is the time-dependent variational principle (TDVP) [32]. We found TDVP to be the most effective tensor network method we used, so the figures throughout this article, used for comparison against results from hardware, report TDVP outputs. We use the implementation in Tenpy [89], namely the `TwoSiteTDVPEngine`. We vary bond dimension as seen in analysis throughout, but fix hyperparameters in all simulations: we iteratively evolve each system for $\Delta t = 0.01$, and set the threshold for truncation in the Singular Value Decomposition (SVD) subroutine at 10^{-10}.

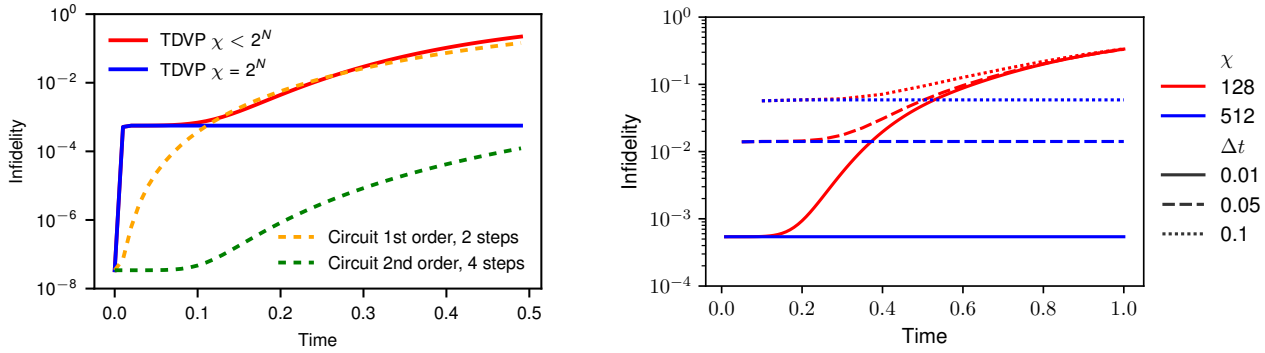


FIG. 35. Infidelity of TDVP for a 3×3 FH model with $U = 0$. **(a)** Comparison with simulation via quantum circuits. The solid (dashed) lines show simulation via TDVP (circuit emulation). The blue line shows $\chi = 2^N = 512$, while red line shows $\chi = 2^{N-3} = 64$. The green (orange) lines simulate the same configuration through an 18-qubit quantum circuit, following the same circuit structure as our experiment, i.e., four 2^{nd} order Trotter steps (two 1^{st} order steps). **(b)** Infidelity of TDVP for varying χ and Δt.

To validate the correctness of our TDVP and experimental implementation, in Figure 35 we compare the state infidelity achieved by TDVP against that of our circuits, for a 3×3 FH model. In all cases, statevector fidelity is with respect to exact diagonalisation through Quspin [90]. We prepare a state of similar structure to that used in our experiment, i.e., a double-periodic lattice with two holons, one doublon, and triplet pairs between remaining sites. We set $U = 0, \phi = 0.5$ in Equation (1). Our system thus has 9 sites, with 8 occupied modes. Tensor network representations of quantum systems are characterised by their number of sites, N, and physical dimension, d, i.e., each site corresponds to a d-dimensional Hilbert space. An MPS simulation can completely capture the state of a target system given a bond dimension $\chi = d^{\frac{N}{2}}$, as stated in Ref. [91, Theorem 1]. For fermionic systems such as the FH model under study, there are 4 available occupation levels for each site ($|\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle$), so $d = 4$ and hence we require $\chi = 2^N$ to fully represent these systems, while an MPS with $\chi < 2^N$ provides an approximate (inexact) model of the system.

We see the difference in accuracy in Figure 35(a): the blue (red) line represents an MPS of $\chi = 2^N = 512$ ($\chi = 2^{N-3} = 64$). The constant infidelity of the $\chi = 2^N$ line indicates the error due to the choice of Δt; decreasing Δt improves the fidelity at the cost of a higher run time. The circuit infidelities indicate a decrease in Trotter error as the circuit is given a higher Trotter order and more Trotter steps. In cases where the bond dimension is insufficient (for TDVP) or the number of Trotter steps is insufficient (for the quantum circuits), infidelity increases with time.

In Figure 35(b) we examine the effect of bond dimension, χ, and time step, Δt, on the evolution through TDVP as a function of time, by computing the state infidelity with respect to evolution of the same model with Quspin [90], which can exactly compute time evolution for this system size. We study a 3×3 FH model of the same structure as in the main text, i.e. $J = 1, U = 4, \phi = \pi$ in Equation (1) and with periodic boundary conditions on a chain of triplet pairs, with one holon and one doublon. We see that, where the bond dimension is sufficient to capture the entire state, i.e. $\chi = 2^N = 512$, and after a few time steps, at which point the bond dimension is reached, the error is determined solely by the time step. In the regime $\chi < 2^N$, however, the error is not fixed and grows with time. The choice of time step directly determines the run time of the TDVP instance: we run $\frac{T_{\max}}{\Delta t}$ iterations of the algorithm, which take roughly the same time each (provided the MPS has reached the full bond dimension, which occurs at early times for large systems). For the largest systems under study, we know that we cannot provide a large

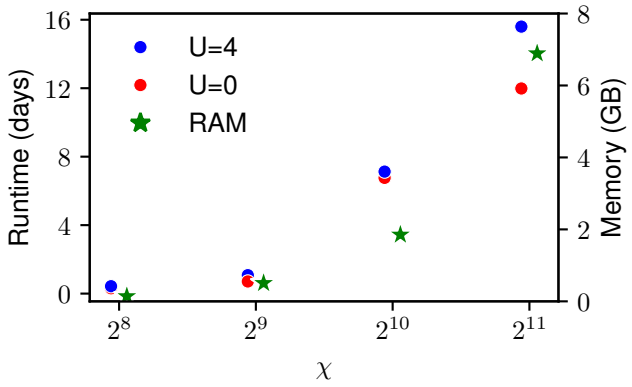


FIG. 36. Time and memory requirements for the TDVP algorithm to simulate the 4×7 FH model, as a function of bond dimension χ. These requirements refer to running on a 2.6 GHz Intel Cascade Lake processor with 8 CPUs. The time reported on the left-hand y-axis is to simulate the complete timeseries, i.e., up to $T = 2$ in time steps $\Delta t = 0.01$, with an SVD truncation threshold of 10^{-10}. The right-hand y-axis shows the RAM requirements for the same instances, which correspond to the green stars.

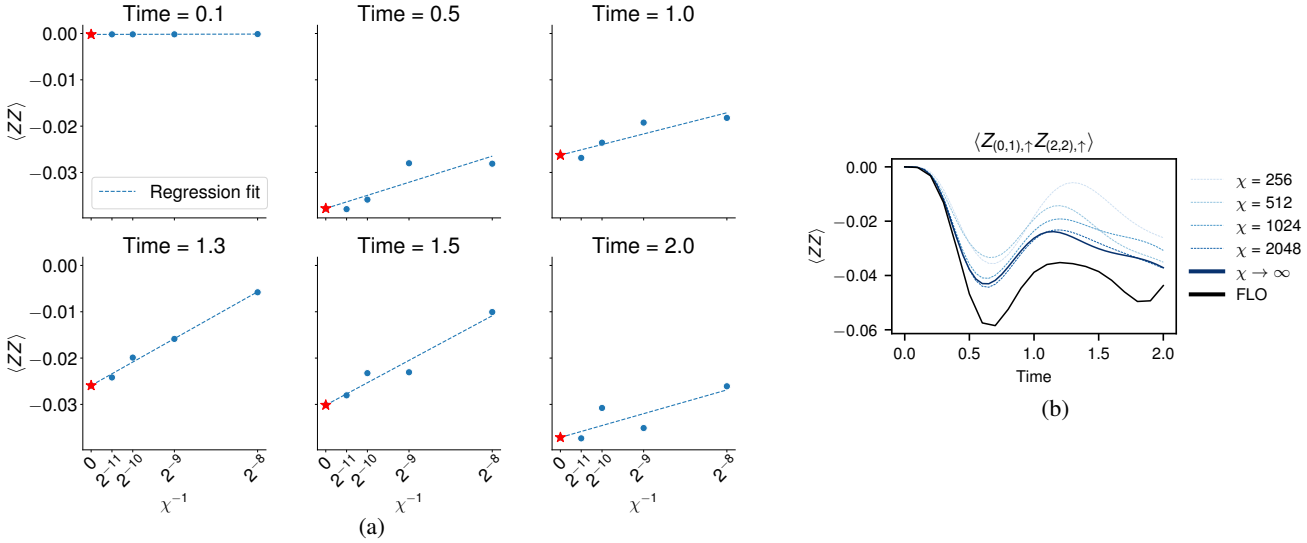


FIG. 37. **(a)** Extrapolation of observables to $\chi = \infty$. Blue dots are computed through TDVP with χ, and a linear regression is fit through χ^{-1}. The intercepts of the regression (red stars) are taken as estimates of the observable at $\chi^{-1} = 0$, i.e., to estimate the observable at $\chi \to \infty$. **(b)** Dynamics of $\langle ZZ \rangle$ on sites $(0, 1), (1, 2)$ for $|\uparrow\uparrow\rangle$. The true dynamics, computed through FLO, are shown in black.

enough bond dimension to capture the state fully; therefore, we opt for a small Δt to give the TDVP routine a fair opportunity to match or outperform the experiment. Throughout this work, we have fixed $\Delta t = 0.01$ to achieve a balance between accuracy and practical runtime. We also fix the truncation threshold at 10^{-10} within the singular value decomposition subroutine.

A practical consideration for the implementation of TDVP are the runtime requirements, as shown in Figure 36. We performed our simulations on the Google Cloud Platform with virtual machines of type `n2-highmem-32`,i.e. on Intel Lake 2.6 GHz processors with 8 vCPUs and 12 GB RAM per TDVP instance [92]. Evidently, the gains in accuracy from higher bond dimension incur significant computational time and cost. We deem $\sim$ 2 weeks for a single TDVP run to be feasible, but any meaningful boost to χ will yield calculations potentially lasting several months and tens of gigabytes of random access memory (RAM), without guaranteeing convergence in observables. In comparison, we saw in Section C 7 that the total run time on the quantum device was 4.5 hours, i.e. 2.25 hours per timeseries, albeit with additional overhead for compilation and queueing for access to the device.

Following the procedure described in [27], we extrapolate our observables in an attempt to find more accurate approximations. Given access to observables for $\langle O \rangle \in \{\langle Z \rangle, \langle ZZ \rangle\}$ at all evolution times for $\chi \in \{2^8, 2^9, 2^{10}, 2^{11}\}$, we we wish to estimate the observable as computed by a tensor network of higher bond dimension. We perform linear regression on the set $(\chi^{-1}, \langle O_\chi \rangle)$, and take the intercept of the regression as the estimate of the observable at $\chi^{-1} = 0$, i.e. $\chi \to \infty$. Figure 37 illustrates this

regression for a single observable, along with the resulting dynamics. We repeat this procedure on all $\langle O\rangle \in \{\langle Z\rangle, \langle ZZ\rangle\}$, for each combination of (U, t) and report these results as $\chi \to \infty$ throughout.

2. Direct circuit contraction with `quimb`

In this section, we quantify the space and time complexity for the exact contraction of the tensor network corresponding to the experimental quantum circuits. To do this, we use the rehearse feature of the `quimb` package [93], which allows us to estimate the complexity of contraction without the need to contract the tensor network explicitly. When contracting a tensor network, one has to consider both the contraction cost C, which is an estimate of the time complexity of the contraction, and the contraction width W, which represents the size of the largest intermediate tensor occurring in the contraction and which is directly connected to the memory required to store it.

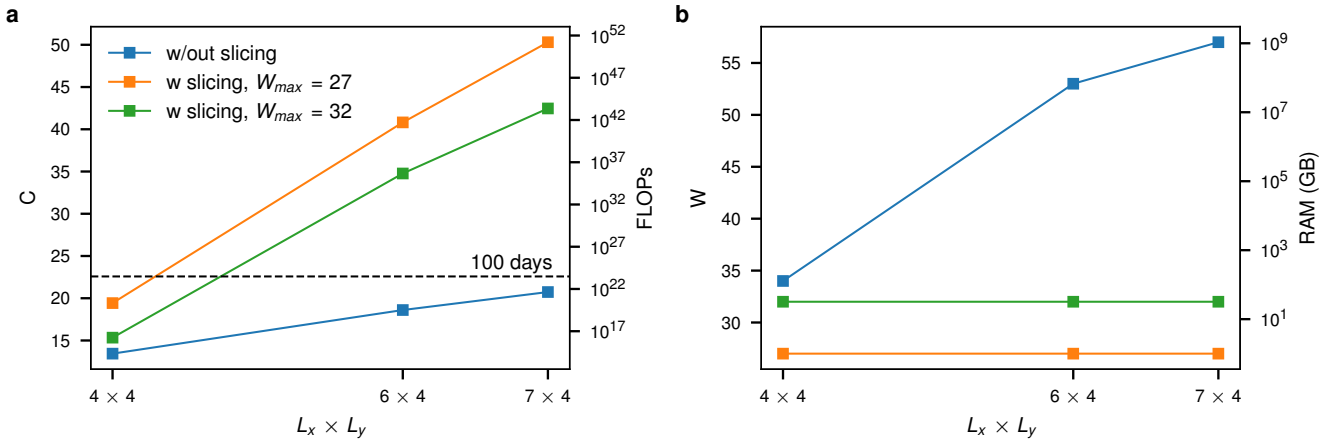


FIG. 38. Time and space complexity of the contraction of the quantum circuit with $U = 4$ for different system sizes, L_x and L_y. a) Contraction cost C and b) contraction width W of the largest intermediate tensor to compute a single amplitude via exact contraction of the tensor network with `quimb`. For a complex tensor with single precision, the contraction cost can be converted into FLOPs as $\mathrm{FLOPs} = 8^C$, while the memory required to store a tensor with width W is RAM (GB) $= 2^W \times 8/20^{30}$. In a), the horizontal dashed line represents approximately the number of FLOPs corresponding to 100 days of computation on the supercomputer Summit [94]. All the hyper-optimisers have run on four threads with the same time cutoff (six hours).

In Figure 38 we show both the contraction cost C and contraction width W associated with the computation of a single amplitude [95] for different system sizes, L_x and L_y. We used three high-quality `cotengra` [94] hyper-optimisers with subtree reorganisation to find the optimal contraction path, namely, an unconstrained one targeting both cost and width, and two hyper-optimisers with dynamic index slicing with target sizes $W_{\max} = 27$ and $W_{\max} = 32$, respectively [94, 96, 97]. Recalling that storing a complex, single-precision tensor with width W requires $2^W \times 8/20^{30}$ GB of memory, the two target widths $W_{\max}$ correspond to a memory of 1GB and 32GB, respectively. The latter values have been chosen such that the contraction can be performed on a commercial GPU ($W_{\max} = 28$) or on an HPC GPU cluster ($W_{\max} = 32$). Indeed, note that this is the memory required to store the largest intermediate tensor, so for a whole computation, one usually has to take into account an additional buffer factor of ≈ 4 [94]. As can be seen in Figure 38, dynamic slicing makes it possible to trade memory for time complexity. The latter is measured by the contraction cost C, which for a complex tensor with single precision translates to 8×10^C real FLOPs. In panel c), we draw a horizontal line that approximately corresponds to a computational time of 100 days on the Summit supercomputer (281 petaFLOPs [94, 98]). From Figure 38, it is therefore clear that the exact contraction of the tensor network corresponding to a 6×4 instance is already unrealistic (note that the memory required by unconstrained contraction is larger than 10^7 GB).

3. Fermionic Matrix Product States

We implement further algorithms for time dynamics simulation of the FH model through the MPS-FQE package [33]. We solve the time-dependent Schrödinger equation with two methods: time-dependent version of the density matrix renormalisation group (TD-DMRG) within the MPS/MPO framework and an explicit fourth-order Runge–Kutta (RK4) integrator applied to the MPS representation with compression at each step. Both of these techniques use the Runge-Kutta formula. TD-DMRG updates the MPS tensors in time while controlling bond dimensions via truncation of the reduced density matrices to retain the most relevant Schmidt coefficients in the singular value decomposition. The RK4 method corresponds to a global RK4 propagation via repeated MPO applications.

For both of these methods, as well as TDVP, we assessed the accuracy, stability and algorithmic cost with respect to exact evolution, computed through Quspin [90]. The truncation error accumulates for long-time dynamics, as expected for the genuinely two-dimensional system under study, since the entanglement growth shows volume-law behavior. The TD-DMRG method suffers when targeting the basis, due to the level of truncation. On the other hand, the RK4 scheme outperforms TD-DMRG when small time steps are used, but loses fidelity due to the compression steps, which make the evolution non-unitary. The RDM used for optimising the basis targets multiple times within each step of time evolution to preserve the evolving subspace, thus avoiding a Trotter factorisation. We ran TD-DMRG++ ([99]), which improves the accuracy at fixed bond dimension by using independent MPS to represent states at different times.

In contrast, the TDVP family of approaches offers variationally consistent flows and preserves unitarity better than TD-DMRG and RK4, which prove crucial when the matrix product operator contains long-range coupling from the 2D-to-1D mapping. This is achieved by enforcing a variationally consistent, norm-conserving evolution within the MPS manifold, handling the MPO long-range terms without Trotterization. The TDVP method, therefore, is more suitable for the 2D problem of the main text, as for higher times, the aggressive truncation needed in the TD-DMRG or RK4 methods is penalised by effective long-range couplings from the Jordan-Wigner mapping.

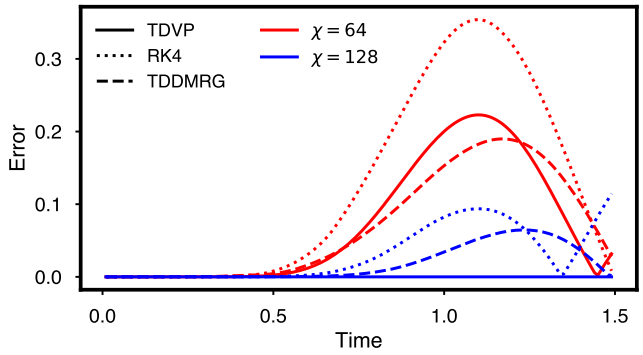


FIG. 39. Error in expectation value of density-density correlation $n_0 n_8$, i.e. the correlation between the furthest sites of a 3×3 square Fermi-Hubbard model with open boundary conditions. TDVP (solid), TK4 (dotted) and TD-DMRG (dashed) are shown for two values of bond dimension, $\chi = 64$ (red) and $\chi = 128$ (blue).

We investigated non-local correlated observables for the 3×3 FH lattice with open boundary conditions. In Figure 39, we see that, in the non-interacting limit ($U = 0$), the long-time dynamics computed with restricted bond dimensions exhibit significant deviations from the exact solution, when we considered the density-density correlations between the first and last sites of the lattice. Among the methods considered, TDVP provides the best accuracy compared to both TD-DMRG and the RK4 scheme. We hence conclude that TD-DMRG and RK4 techniques are less suited to the problem under study in our main text, and therefore include TDVP to represent tensor network simulations throughout.

4. Compressed MPS

The final technique considered for the simulation of time dynamics is to group several lattice sites together within the MPS. That is, instead of a lattice of size $L_x \times L_y$ sites, each with $d = 4$, we group the sites along L_x into a single *super site*, such that we build a lattice of size 1×7 with $d = 4^{L_x}$.

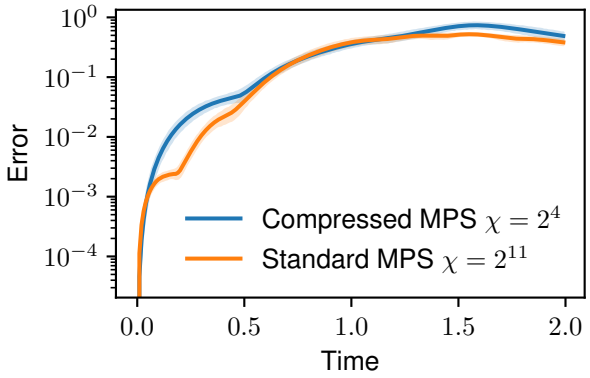


FIG. 40. Errors in density observable computed through TDVP using standard MPS structure compared with compressed MPS structure.

We wish to determine whether this technique can outperform TDVP as described in Section G 1. For a fair comparison, we keep resources constant, i.e. we limit RAM to $\sim$ 10GiB, which allows $\chi = 2^4 = 16$ for the compressed method and $\chi = 2^{11} = 2048$ for the standard MPS. Figure 40 shows how closely each method can recover the exact dynamics of a 4×7 FH model with $U = 0$, for which the ground truth can be computed through FLO. We see that, while the compressed MPS technique can faithfully match the dynamics at low times, the standard MPS performs more reliably up to $T \approx 0.5$, and both methods struggle similarly at higher times. Ultimately, although the compressed technique is similarly viable for TDVP of this system, it does not clearly outperform the standard MPS, and we hence opt to use the more widely accepted standard MPS throughout.

5. Density Matrix Renormalization Group

The ground state of the Hubbard model is estimated using the density matrix renormalisation group (DMRG) procedure [100]. We use the DMRG algorithm implemented in the ITensor Julia package [101, 102]. We start the DMRG algorithm with an initial state in the correct number sector and set the option to preserve quantum numbers during the DMRG procedure. Doing this reduces the amount of memory required and can speed up the code. For all runs of the algorithm, the cutoff is fixed at 10^{-7} and 50 sweeps are done, but in Figure 41 we present results where the maximum bond dimension is varied. This figure demonstrates the convergence of DMRG as the bond dimension is increased and justifies our choice of bond dimension 4096 in Figure 11 as a reasonable approximation to the actual ground state.

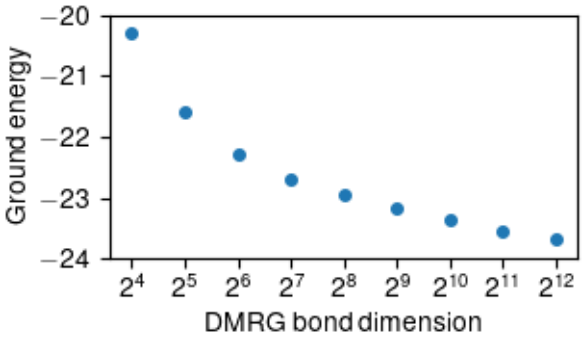


FIG. 41. DMRG ground energies for varying maximum bond dimension, for a 4×7 Fermi-Hubbard model at half-filling with periodic boundary conditions and a flux of π.

Appendix H: Cross-Entropy Benchmarking with FLO

The method of cross-entropy benchmarking (XEB) [103–107] is by now well-established in the context of random quantum circuits, as well as limited classes of circuits that allow efficient polynomial time computation of output amplitudes [108, 109] and also quantum time evolution of physical systems [106, 110]. In this work, we combine the latter two attributes into one protocol, leveraging the fact that amplitudes of the ideal $U = 0$ Near FLO time evolution can be exactly computed by practically efficient determinant methods described in Section E. The cross-entropy estimators we compute are

$$\mathrm{XE}(\text{generator}||\text{FLO}) = \langle \log_2(1/p_{\text{FLO}}) \rangle_{\text{generator}}$$

for the various choices of sample generators: experiment (exp), MPS, uniform (baseline) (see Figure 42. Here we are using p_{FLO} as the ideal model and testing the "surprisal" of applying this model to the different sample sets generated by the experiment, the highest bond dimension MPS ($\chi = 2048$), and a uniform sample generator. Low scores are better in that they correspond to less surprise at the samples that are seen, but note that just like XEB for random circuits this is not meant to be a strict test of fidelity (it can be "spoofed" by low Shannon entropy distributions that output strings with a large value of p_{FLO}), rather it is meant to compare "honest" alternatives to the quantum circuit such as the large bond dimension MPS approximation we consider.

Since the various sample generators occasionally produce bit strings z with extremely small probability $p_{\text{FLO}}(z)$, a technical caveat is that we eliminate divergences in the $\log$ function by including some uniform noise in the ideal model

$$p_\epsilon(z) = (1 - \epsilon)p_{\text{FLO}}(z) + \epsilon/D \tag{H1}$$

where $D = \binom{28}{14}^2$ is the dimension of the system and $\epsilon = 0.1$ is chosen in Figure 42.

1. Experimental advantage in sampling.

When the difference in cross entropy $\mathrm{XE}(\text{generator}||\text{FLO})$ between two sample generators is $r = \mathrm{XE}(g_1||\text{FLO}) - \mathrm{XE}(g_2||\text{FLO})$ bits, then the set of M samples from g_2 is 2^{rM} times less surprising to see when one is expecting the samples to come from p_{FLO}. This means that a decrease in the LogXEB value increases the likelihood of the model exponentially in the number of samples (according to this particular test). At evolution time $t = 2$, the experimental estimator improves on the MPS by 1.5 bits in expectation, a difference which is larger than the standard error (the MPS and uniform results are computed with 1000 samples per time point, and so the standard error is small on this scale).

The results indicate a decisive advantage for the experiment over the $\chi = 2048$ bond dimension MPS as a sample generator. The experiment achieves this advantage despite the fact that the ideal FLO model and the MPS are untrotterized models of the continuous time evolution. Suppose one considers constraining to "honest" sample generators that reproduce local expectations as well as the MPS in Section G. In that case, the results support the conclusion that the experiment is the best sample generator we know of (since it is practical to compute Near FLO amplitudes, but not to sample from the output distribution).

2. Modeling circuit fidelity with LogXEB

The term "benchmarking" in XEB refers to an estimate of circuit fidelity. Usually, this is done by computing the linear XEB estimators $\langle p_{\text{FLO}} \rangle_{\text{generator}}$ for the various sample generators, and then scaling and shifting them to relate to an estimate of the circuit fidelity F under the global depolarizing model. Instead, we devise an alternate approach to estimate the global circuit fidelity under the global depolarising model, by minimizing the LogXEB score $\mathrm{XE}(\text{exp}||\text{FLO}, \epsilon) = \langle \log(1/p_{\text{FLO},\epsilon}) \rangle_{\text{exp}}$ for the distribution (H1) as a function of ϵ. This noise estimation is a proper scoring rule with a fixed sample generator and a parameterized class of models: at each time point, we can apply it to extract the most likely ϵ among this set of models. The value of the optimised circuit infidelity ϵ^* is shown as a function of time in Figure 42, and the resulting cross entropy estimators in Figure 43 indicate an even larger advantage for quantum hardware over the MPS at late times.

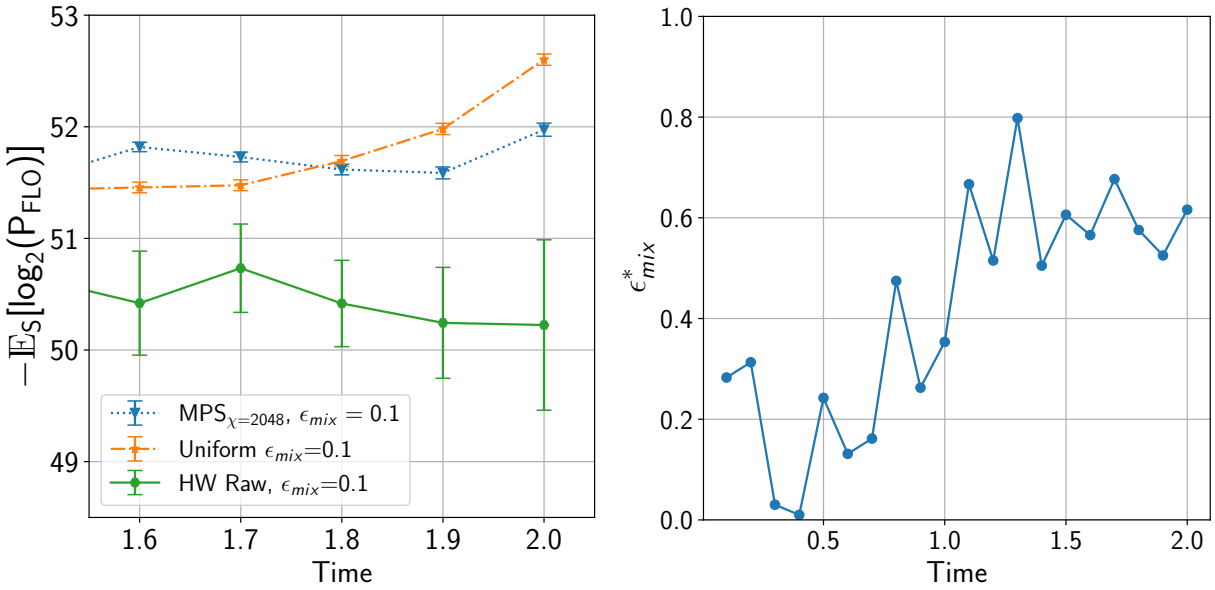


FIG. 42. Left: A comparison of the ideal model p_{FLO} against different sample generators: experiment, MPS with $\chi = 2048$, and uniform (baseline) as a baseline. A lower score means that the samples produced by the generator are less surprising according to the FLO likelihoods of those samples. Right: Estimates of the circuit infidelity ϵ under the global depolarising noise model obtained by minimizing XE(exp||FLO, ϵ) at each point in time.

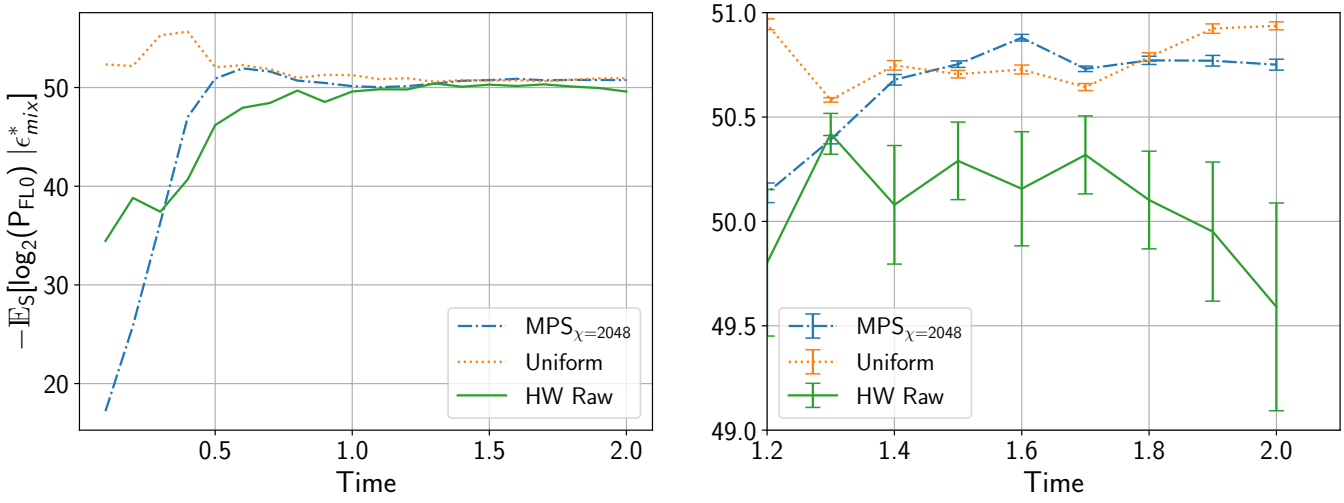


FIG. 43. Cross entropy estimators for the optimised choice of $\epsilon^*(t)$ for the FLO model mixed with depolarising noise in (H1). The right-hand plot zooms in on the late-time behaviour in the left-hand plot.

[1] Dagotto, E. Correlated electrons in high-temperature superconductors. *Rev. Mod. Phys.* **66**, 763–840 (1994). URL https://link.aps.org/doi/10.1103/RevModPhys.66.763.

[2] Anderson, P. W. & Khveshchenko, D. Charge-spin separation in 2D Fermi systems: Singular interactions as modified commutators, and solution of the 2D Hubbard model in the bosonized approximation. *Phys. Rev. B* **52**, 16415–16419 (1995). URL https://link.aps.org/doi/10.1103/PhysRevB.52.16415.

[3] Arute, F. *et al.* Observation of separated dynamics of charge and spin in the Fermi-Hubbard model (2020). URL https://arxiv.org/abs/2010.07965. arXiv:2010.07965.

[4] Vijayan, J. *et al.* Time-resolved observation of spin-charge deconfinement in fermionic Hubbard chains. *Science* **367**, 186–189 (2020). URL http://dx.doi.org/10.1126/science.aay2354.

[5] Akiyama, S., Kuramashi, Y. & Yamashita, T. Metal–insulator transition in the (2+1)-dimensional Hubbard model with the tensor renormalization group. *Prog. Theor. Exp. Phys.* **2022**, 023I01 (2022). URL https://doi.org/10.1093/ptep/ptac014.

[6] Hart, R. A. *et al.* Observation of antiferromagnetic correlations in the Hubbard model with ultracold atoms. *Nature* **519**, 211–214 (2015). URL http://dx.doi.org/10.1038/nature14223.

[7] Parsons, M. F. *et al.* Site-resolved measurement of the spin-correlation function in the Fermi-Hubbard model. *Science* **353**, 1253–1256 (2016). URL http://dx.doi.org/10.1126/science.aag1430.

[8] Mazurenko, A. *et al.* A cold-atom Fermi–Hubbard antiferromagnet. *Nature* **545**, 462–466 (2017). URL http://dx.doi.org/10.1038/nature22362.

[9] Yamada, S., Imamura, T. & Machida, M. 16.447 TFlops and 159-billion-dimensional exact-diagonalization for trapped Fermion-Hubbard model on the Earth simulator. In *SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, 44–44 (2005). URL https://doi.org/10.1109/SC.2005.1.

[10] LeBlanc, J. P. F. *et al.* Solutions of the two-dimensional Hubbard model: Benchmarks and results from a wide range of numerical algorithms. *Phys. Rev. X* **5**, 041041 (2015). URL https://link.aps.org/doi/10.1103/PhysRevX.5.041041.

[11] Bakr, W. S., Ba, Z. & Prichard, M. L. Microscopy of ultracold fermions in optical lattices (2025). URL https://arxiv.org/abs/2507.04042. arXiv:2507.04042.

[12] Xu, M. *et al.* A neutral-atom Hubbard quantum simulator in the cryogenic regime. *Nature* **642**, 909–915 (2025). URL https://doi.org/10.1038/s41586-025-09112-w.

[13] Innerberger, M., Worm, P., Prauhart, P. & Kauch, A. Electron-light interaction in nonequilibrium: exact diagonalization for time-dependent Hubbard hamiltonians. *The European Physical Journal Plus* **135** (2020). URL http://dx.doi.org/10.1140/epjp/s13360-020-00919-2.

[14] Thompson, A. P., Soeteman, A., Cade, C. & Niesen, I. Non-zero noise extrapolation: accurately simulating noisy quantum circuits with tensor networks (2025). URL https://arxiv.org/abs/2501.13237. arXiv:2501.13237.

[15] Vilchez-Estevez, L., Santos, R. A., Wang, S. Y. & Gambetta, F. M. Extracting the spin excitation spectrum of a fermionic system using a quantum processor. *Phys. Rev. B* **112** (2025). URL http://dx.doi.org/10.1103/ydfw-k83n.

[16] Chowdhury, T. A., Korepin, V., Pascuzzi, V. R. & Yu, K. Quantum utility in simulating the real-time dynamics of the Fermi-Hubbard model using superconducting quantum computers (2025). URL https://arxiv.org/abs/2509.14196. arXiv:2509.14196.

[17] Evered, S. J. *et al.* Probing the Kitaev honeycomb model on a neutral-atom quantum computer. *Nature* **645**, 341–347 (2025). URL http://dx.doi.org/10.1038/s41586-025-09475-0.

[18] Guardado-Sanchez, E. *et al.* Subdiffusion and heat transport in a tilted two-dimensional Fermi-Hubbard system. *Phys. Rev. X* **10**, 011042 (2020). URL https://link.aps.org/doi/10.1103/PhysRevX.10.011042.

[19] Brown, P. T. *et al.* Bad metallic transport in a cold atom Fermi-Hubbard system. *Science* **363**, 379–382 (2019). URL https://www.science.org/doi/abs/10.1126/science.aat4134.

[20] Nichols, M. A. *et al.* Spin transport in a Mott insulator of ultracold fermions. *Science* **363**, 383–387 (2019). URL https://www.science.org/doi/abs/10.1126/science.aat4387.

[21] Hebenstreit, M., Jozsa, R., Kraus, B., Strelchuk, S. & Yoganathan, M. All pure fermionic non-Gaussian states are magic states for matchgate computations. *Phys. Rev. Lett.* **123**, 080503 (2019). URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.123.080503.

[22] Oszmaniec, M., Dangniam, N., Morales, M. E. & Zimborás, Z. Fermion sampling: A robust quantum computational advantage scheme using fermionic linear optics and magic input states. *PRX Quantum* **3**, 020328 (2022). URL https://link.aps.org/doi/10.1103/PRXQuantum.3.020328.

[23] Kivlichan, I. D. *et al.* Quantum simulation of electronic structure with linear depth and connectivity. *Phys. Rev. Lett.* **120**, 110501 (2018). URL https://link.aps.org/doi/10.1103/PhysRevLett.120.110501.

[24] Cade, C., Mineh, L., Montanaro, A. & Stanisic, S. Strategies for solving the Fermi-Hubbard model on near-term quantum computers. *Phys. Rev. B* **102**, 235122 (2020). URL https://link.aps.org/doi/10.1103/PhysRevB.102.235122.

[25] Stanisic, S. *et al.* Observing ground-state properties of the Fermi-Hubbard model using a scalable algorithm on a quantum computer. *Nat. Commun.* **13**, 5743 (2022). URL http://dx.doi.org/10.1038/s41467-022-33335-4.

[26] P. Santos, J., Bar, B. & Uzdin, R. Pseudo twirling mitigation of coherent errors in non-Clifford gates. *npj Quantum Inf.* **10** (2024). URL http://dx.doi.org/10.1038/s41534-024-00889-8.

[27] Haghshenas, R. *et al.* Digital quantum magnetism at the frontier of classical simulations (2025). URL https://arxiv.org/abs/2503.20870. arXiv:2503.20870.

[28] Montanaro, A. & Stanisic, S. Error mitigation by training with fermionic linear optics (2021). URL https://arxiv.org/abs/2102.02120. arXiv:2102.02120.

[29] Terhal, B. M. & DiVincenzo, D. P. Classical simulation of noninteracting-fermion quantum circuits. *Phys. Rev. A* **65**, 032325 (2002). URL https://link.aps.org/doi/10.1103/PhysRevA.65.032325.

[30] Rasmussen, C. E. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning* (Springer Berlin Heidelberg, 2004). URL https://doi.org/10.1007%2F978-3-540-28650-9_4.

[31] Bishop, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).

[32] Haegeman, J. *et al.* Time-dependent variational principle for quantum lattices. *Phys. Rev. Lett.* **107**, 070601 (2011). URL http://dx.doi.org/10.1103/PhysRevLett.107.070601.

[33] Provazza, J. *et al.* Fast Emulation of Fermionic Circuits with Matrix Product States. *J. Chem. Theory Comput.* **20**, 3719–3728 (2024). URL https://pubs.acs.org/doi/10.1021/acs.jctc.4c00200.

[34] Miller, A. *et al.* Simulation of fermionic circuits using Majorana propagation (2025). URL https://arxiv.org/abs/2503.18939. arXiv:2503.18939.

[35] Chao, K. A., Spałek, J. & Oleś, A. M. Canonical perturbation expansion of the Hubbard model. *Phys. Rev. B* **18**, 3453–3464 (1978). URL https://link.aps.org/doi/10.1103/PhysRevB.18.3453.

[36] Lee, P. A., Nagaosa, N. & Wen, X.-G. Doping a Mott insulator: Physics of high-temperature superconductivity. *Rev. Mod. Phys.* **78**, 17–85 (2006). URL https://link.aps.org/doi/10.1103/RevModPhys.78.17.

[37] Lee, P. A. From high temperature superconductivity to quantum spin liquid: progress in strong correlation physics. *Reports on Progress in Physics* **71**, 012501 (2007). URL https://dx.doi.org/10.1088/0034-4885/71/1/012501.

[38] Kotliar, G. & Ruckenstein, A. E. New functional integral approach to strongly correlated fermi systems: The Gutzwiller approximation as a saddle point. *Phys. Rev. Lett.* **57**, 1362–1365 (1986). URL https://link.aps.org/doi/10.1103/PhysRevLett.57.1362.

[39] Polyakov, A. Thermal properties of gauge fields and quark liberation. *Phys. Lett. B* **72**, 477–480 (1978). URL https://www.sciencedirect.com/science/article/pii/0370269378907372.

[40] Borgs, C. Area law for spatial Wilson loops in high-temperature lattice gauge theories. *Nucl. Phys. B* **261**, 455–460 (1985). URL https://www.sciencedirect.com/science/article/pii/0550321385905826.

[41] Bosse, J. L. *et al.* Efficient and practical Hamiltonian simulation from time-dependent product formulas. *Nat. Commun.* **16**, 2673 (2025). URL http://dx.doi.org/10.1038/s41467-025-57580-5.

[42] Fava, S. *et al.* Magnetic field expulsion in optically driven YBa2Cu3O6.48. *Nature* **632**, 75–80 (2024). URL https://doi.org/10.1038/s41586-024-07635-2.

[43] White, S. R., Noack, R. M. & Scalapino, D. J. Resonating valence bond theory of coupled Heisenberg chains. *Phys. Rev. Lett.* **73**, 886–889 (1994). URL https://link.aps.org/doi/10.1103/PhysRevLett.73.886.

[44] Sierra, G. The nonlinear sigma model and spin ladders. *J. Phys. A* **29**, 3299 (1996). URL https://dx.doi.org/10.1088/0305-4470/29/12/032.

[45] Dagotto, E. & Rice, T. M. Surprises on the way from one- to two-dimensional quantum magnets: The ladder materials. *Science* **271**, 618–623 (1996). URL https://www.science.org/doi/abs/10.1126/science.271.5249.618.

[46] Cabra, D. C., Honecker, A. & Pujol, P. Magnetization plateaux in n-leg spin ladders. *Phys. Rev. B* **58**, 6241–6257 (1998). URL https://link.aps.org/doi/10.1103/PhysRevB.58.6241.

[47] Haldane, F. D. M. Nonlinear field theory of large-spin Heisenberg antiferromagnets: Semiclassically quantized solitons of the one-dimensional easy-axis Néel state. *Phys. Rev. Lett.* **50**, 1153–1156 (1983). URL https://link.aps.org/doi/10.1103/PhysRevLett.50.1153.

[48] Haldane, F. Continuum dynamics of the 1-D Heisenberg antiferromagnet: Identification with the $o(3)$ nonlinear sigma model. *Phys. Lett. A* **93**, 464–468 (1983). URL https://www.sciencedirect.com/science/article/pii/037596018390631X.

[49] Otsuka, Y. & Hatsugai, Y. Mott transition in the two-dimensional flux phase. *Phys. Rev. B* **65**, 073101 (2002). URL https://link.aps.org/doi/10.1103/PhysRevB.65.073101.

[50] Otsuka, Y., Yunoki, S. & Sorella, S. Universal quantum criticality in the metal-insulator transition of two-dimensional interacting Dirac electrons. *Phys. Rev. X* **6**, 011029 (2016). URL https://link.aps.org/doi/10.1103/PhysRevX.6.011029.

[51] Mermin, N. D. & Wagner, H. Absence of ferromagnetism or antiferromagnetism in one- or two-dimensional isotropic Heisenberg models. *Phys. Rev. Lett.* **17**, 1133–1136 (1966). URL https://link.aps.org/doi/10.1103/PhysRevLett.17.1133.

[52] Anderson, P. W. An approximate quantum theory of the antiferromagnetic ground state. *Phys. Rev.* **86**, 694–701 (1952). URL https://link.aps.org/doi/10.1103/PhysRev.86.694.

[53] Takahashi, M. Modified spin-wave theory of a square-lattice antiferromagnet. *Phys. Rev. B* **40**, 2494–2501 (1989). URL https://link.aps.org/doi/10.1103/PhysRevB.40.2494.

[54] Huse, D. A. Ground-state staggered magnetization of two-dimensional quantum Heisenberg antiferromagnets. *Phys. Rev. B* **37**, 2380–2382 (1988). URL https://link.aps.org/doi/10.1103/PhysRevB.37.2380.

[55] Okabe, Y. & Kikuchi, M. Quantum monte carlo simulation of the spin 1/2 XXZ model on the square lattice. *J. Phys. Soc. Jpn.* **57**, 4351–4358 (1988). URL https://doi.org/10.1143/JPSJ.57.4351.

[56] Reger, J. D. & Young, A. P. Monte carlo simulations of the spin-1/2 Heisenberg antiferromagnet on a square lattice. *Phys. Rev. B* **37**, 5978–5981 (1988). URL https://link.aps.org/doi/10.1103/PhysRevB.37.5978.

[57] Lee, C. H., Yamada, K., Hiraka, H., Venkateswara Rao, C. R. & Endoh, Y. Spin pseudogap in $la_{2-x}sr_xcuo_4$ studied by neutron scattering. *Phys. Rev. B* **67**, 134521 (2003). URL https://link.aps.org/doi/10.1103/PhysRevB.67.134521.

[58] Hashimoto, M., Vishik, I. M., He, R.-H., Devereaux, T. P. & Shen, Z.-X. Energy gaps in high-transition-temperature cuprate superconductors. *Nat. Phys.* **10**, 483–495 (2014). URL http://dx.doi.org/10.1038/nphys3009.

[59] Li, Y. *et al.* Low-energy antiferromagnetic spin fluctuations limit the coherent superconducting gap in cuprates. *Phys. Rev. B* **98** (2018). URL http://dx.doi.org/10.1103/PhysRevB.98.224508.

[60] LeBlanc, J. P. F. & Gull, E. Equation of state of the fermionic two-dimensional Hubbard model. *Phys. Rev. B* **88** (2013). URL http://dx.doi.org/10.1103/PhysRevB.88.155108.

[61] Ogata, M. & Shiba, H. Bethe-ansatz wave function, momentum distribution, and spin correlation in the one-dimensional strongly correlated Hubbard model. *Phys. Rev. B* **41**, 2326–2338 (1990). URL https://link.aps.org/doi/10.1103/PhysRevB.41.2326.

[62] Faddeev, L. D. How algebraic Bethe ansatz works for integrable model (1996). URL https://arxiv.org/abs/hep-th/9605187. arXiv:hep-th/9605187.

[63] Childs, A. M., Su, Y., Tran, M. C., Wiebe, N. & Zhu, S. Theory of Trotter error with commutator scaling. *Phys. Rev. X* **11** (2021). URL http://dx.doi.org/10.1103/PhysRevX.11.011020.

[64] Sivarajah, S. *et al.* t|ket⟩: a retargetable compiler for NISQ devices. *Quantum Sci. Technol.* **6**, 014003 (2020). URL http://dx.doi.org/10.1088/2058-9565/ab8e92.

[65] pytket-quantinuum API documentation — docs.quantinuum.com. https://docs.quantinuum.com/tket/extensions/pytket-quantinuum/. [Accessed 24-09-2025].

[66] Fagan, A. & Duncan, R. Optimising Clifford circuits with Quantomatic. *Electronic Proceedings in Theoretical Computer Science* **287**, 85–105 (2019). URL http://dx.doi.org/10.4204/EPTCS.287.5.

[67] Quantinuum. Quantinuum hardware specifications. https://github.com/CQCL/quantinuum-hardware-specifications (2025). Accessed: 2025-09-24.

[68] Bennett, C. H., DiVincenzo, D. P., Smolin, J. A. & Wootters, W. K. Mixed-state entanglement and quantum error correction. *Phys. Rev. A* **54**, 3824–3851 (1996). URL https://link.aps.org/doi/10.1103/PhysRevA.54.3824.

[69] Cai, Z. *et al.* Quantum error mitigation. *Rev. Mod. Phys.* **95**, 045005 (2023). URL https://link.aps.org/doi/10.1103/RevModPhys.95.045005.

[70] Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011). URL http://jmlr.org/papers/v12/pedregosa11a.html.

[71] Lebigot, E. O. `uncertainties` – a Python package for calculations with uncertainties. *http://pythonhosted.org/uncertainties/* (2025). URL http://pythonhosted.org/uncertainties/.

[72] Wasserman, L. *All of Nonparametric Statistics*. Springer Texts in Statistics (Springer, New York, NY, 2007), 1 edn.

[73] Gordon, N., Salmond, D. & Smith, A. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Proc. Inst. Electr. Eng.* **140**, 107 (1993). URL https://doi.org/10.1049/ip-f-2.1993.0015.

[74] Doucet, A. & Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. In *Handbook of Nonlinear Filtering*, 656–704 (Oxford University Press, 2011).

[75] Liu, Z.-W. & Winter, A. Many-body quantum magic. *PRX Quantum* **3**, 020333 (2022). URL https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.3.020333.

[76] Howard, M. & Campbell, E. Application of a resource theory for magic states to fault-tolerant quantum computing. *Phys. Rev. Lett.* **118**, 090501 (2017). URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.118.090501.

[77] Cudby, J. & Strelchuk, S. Gaussian decomposition of magic states for matchgate computations (2025). URL https://arxiv.org/abs/2307.12654. arXiv:2307.12654.

[78] Reardon-Smith, O., Oszmaniec, M. & Korzekwa, K. Improved simulation of quantum circuits dominated by free fermionic operations. *Quantum* **8**, 1549 (2024). URL https://quantum-journal.org/papers/q-2024-12-04-1549/.

[79] Dias, B. & Koenig, R. Classical simulation of non-Gaussian fermionic circuits. *Quantum* **8**, 1350 (2024). URL https://quantum-journal.org/papers/q-2024-05-21-1350/.

[80] Hakkaku, S., Tashima, Y., Mitarai, K., Mizukami, W. & Fujii, K. Quantifying fermionic nonlinearity of quantum circuits. *Phys. Rev. Res.* **4**, 043100 (2022). URL https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.4.043100.

[81] Bravyi, S. & Koenig, R. Classical simulation of dissipative fermionic linear optics (2011). URL https://arxiv.org/abs/1112.2184. arXiv:1112.2184.

[82] Bravyi, S. & König, R. Disorder-assisted error correction in Majorana chains. *Commun. Math. Phys.* **316**, 641–692 (2012).

[83] Bravyi, S. & Gosset, D. Complexity of quantum impurity problems. *Commun. Math. Phys.* **356**, 451–500 (2017). URL http://dx.doi.org/10.1007/s00220-017-2976-9.

[84] Bravyi, S., Gosset, D. & Liu, Y. How to simulate quantum measurement without computing marginals. *Phys. Rev. Lett.* **128**, 220503 (2022). URL http://dx.doi.org/10.1103/PhysRevLett.128.220503.

[85] Angrisani, A. *et al.* Classically estimating observables of noiseless quantum circuits. *Phys. Rev. Lett.* – (2025). URL https://link.aps.org/doi/10.1103/lh6x-7rc3.

[86] Fontana, E., Rudolph, M. S., Duncan, R., Rungger, I. & Cîrstoiu, C. Classical simulations of noisy variational quantum circuits. *npj Quantum Inf.* **11** (2025). URL http://dx.doi.org/10.1038/s41534-024-00955-1.

[87] Rudolph, M. S., Jones, T., Teng, Y., Angrisani, A. & Holmes, Z. Pauli propagation: A computational framework for simulating quantum systems (2025). URL https://arxiv.org/abs/2505.21606. arXiv:2505.21606.

[88] Luo, X.-Z., Liu, J.-G., Zhang, P. & Wang, L. Yao.jl: Extensible, efficient framework for quantum algorithm design. *Quantum* **4**, 341 (2020). URL http://dx.doi.org/10.22331/q-2020-10-11-341.

[89] Hauschild, J. *et al.* Tensor network Python (TeNPy) version 1. *SciPost Phys. Codebases* 41 (2024). URL https://scipost.org/SciPostPhysCodeb.41.

[90] Weinberg, P. & Bukov, M. QuSpin: a Python package for dynamics and exact diagonalisation of quantum many body systems part i: spin chains. *SciPost Phys.* **2**, 003 (2017). URL https://scipost.org/10.21468/SciPostPhys.2.1.003.

[91] Perez-Garcia, D., Verstraete, F., Wolf, M. M. & Cirac, J. I. Matrix product state representations (2007). URL https://arxiv.org/abs/quant-ph/0608197. arXiv:quant-ph/0608197.

[92] Google Cloud. *Compute Engine machine types*. Google LLC (2025). URL https://cloud.google.com/compute/docs/general-purpose-machines. Accessed: 2025-10-15.

[93] Gray, J. quimb: a python library for quantum information and many-body calculations. *J. Open Source Softw.* **3**, 819 (2018). URL https://doi.org/10.21105/joss.00819.

[94] Gray, J. & Kourtis, S. Hyper-optimized tensor network contraction. *Quantum* **5**, 410 (2021). URL https://doi.org/10.22331/q-2021-03-15-410.

[95] The contraction cost and width associated with the computation of other quantities are similar and not shown here.

[96] Chen, J., Zhang, F., Huang, C., Newman, M. & Shi, Y. Classical simulation of intermediate-size quantum circuits (2018). URL https://arxiv.org/abs/1805.01450. arXiv:1805.01450.

[97] Villalonga, B. *et al.* A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. *npj Quantum Inf.* **5** (2019). URL http://dx.doi.org/10.1038/s41534-019-0196-1.

[98] Villalonga, B. *et al.* Establishing the quantum supremacy frontier with a 281 Pflop/s simulation. *Quantum Sci. Technol.* **5**, 034003 (2020). URL http://dx.doi.org/10.1088/2058-9565/ab7eeb.

[99] Ronca, E., Li, Z., Jimenez-Hoyos, C. A. & Chan, G. K.-L. Time-Step Targeting Time-Dependent and Dynamical Density Matrix Renormalization Group Algorithms with ab Initio Hamiltonians. *J. Chem. Theory Comput.* **13**, 5560–5571 (2017). URL https://pubs.acs.org/doi/10.1021/acs.jctc.7b00682.

[100] White, S. R. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.* **69**, 2863–2866 (1992). URL https://link.aps.org/doi/10.1103/PhysRevLett.69.2863.

[101] Fishman, M., White, S. R. & Stoudenmire, E. M. The ITensor Software Library for Tensor Network Calculations. *SciPost Phys. Codebases* 4 (2022). URL https://scipost.org/10.21468/SciPostPhysCodeb.4.

[102] Fishman, M., White, S. R. & Stoudenmire, E. M. Codebase release 0.3 for ITensor. *SciPost Phys. Codebases* 4–r0.3 (2022). URL https://scipost.org/10.21468/SciPostPhysCodeb.4-r0.3.

[103] Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–510 (2019). URL http://dx.doi.org/10.1038/s41586-019-1666-5.

[104] Morvan, A. *et al.* Phase transitions in random circuit sampling. *Nature* **634**, 328–333 (2024).

[105] Zlokapa, A., Villalonga, B., Boixo, S. & Lidar, D. A. Boundaries of quantum supremacy via random circuit sampling. *npj Quantum Inf.* **9**, 36 (2023). URL http://dx.doi.org/10.1038/s41534-023-00703-x.

[106] Cheng, B., Meng, F., Zhang, Z.-J. & Yung, M.-H. Generalized cross-entropy benchmarking for random circuits with ergodicity. *Chip* 100127 (2025). URL http://dx.doi.org/10.1016/j.chip.2025.100127.

[107] DeCross, M. *et al.* Computational power of random quantum circuits in arbitrary geometries. *Phys. Rev. X* **15**, 021052 (2025). URL http://dx.doi.org/10.1103/PhysRevX.15.021052.

[108] Chen, J., Ding, D., Huang, C. & Kong, L. Linear cross-entropy benchmarking with Clifford circuits. *Phys. Rev. A* **108**, 052613 (2023). URL http://dx.doi.org/10.1103/PhysRevA.108.052613.

[109] Kaneda, T., Fujii, K. & Ueda, H. Large-scale quantum device benchmarking via LXEB with particle-number-conserving random quantum circuits (2025). URL https://arxiv.org/abs/2505.10820. arXiv:2505.10820.

[110] Mark, D. K., Choi, J., Shaw, A. L., Endres, M. & Choi, S. Benchmarking quantum simulators using ergodic quantum dynamics. *Phys. Rev. Lett.* **131**, 110601 (2023). URL http://dx.doi.org/10.1103/PhysRevLett.131.110601.