Cellular-automaton decoders for topological quantum memories

M. Herold,1 E. T. Campbell,1,2 J. Eisert,1 and M. J. Kastoryano1

1Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, 14195 Berlin, Germany
2Department of Physics and Astronomy, University of Sheffield, Sheffield S3 7RH, UK

Prolonging the lifetime of quantum information stored in a quantum device is a monumental challenge. Yet, it is the necessary first step in the effort to scale up quantum computing and quantum communication to a commercially viable level. As quantum coherence is intrinsically fragile, it is clear that increased robustness of the encoded information needs to rely heavily on quantum error correction [1, 2].

We introduce a new framework for constructing topological quantum memories, by recasting error recovery as a dynamical process on a cellular automaton. We envisage quantum systems controlled by a classical hardware composed of small local memories, communicating with neighbors, and repeatedly performing identical simple update rules. This approach does not require any global operations or complex decoding algorithms. Our cellular automata draw inspiration from classical field theories, with a Coulomb-like potential naturally emerging from the local dynamics. For a 3D automaton coupled to a 2D toric code, we present evidence of an error correction threshold above 6.1% for uncorrelated noise. A 2D automaton equipped with a more complex update rule yields a threshold above 8.2%. Our framework provides decisive new tools in the quest for realizing a passive dissipative quantum memory.

When performing active error correction, it is essential for the decoding process to be much quicker than the decoherence time. Prior proposals have focused on the development of efficient decoding algorithms [7–9] or decoders with high error thresholds [10, 11]. Some of them offer the possibility of parallelization enabling a runtime logarithmic in the system size [8, 12]. This improves prospects, but still requires a computer with rapid communication between many spatially separated cores. Hence, key obstacles remain.

In this work, we propose an entirely new approach towards designing topological quantum memories that naturally incorporates parallelization. Our design can be implemented via classical hardware composed of small local memories and a small set of local operations only depending on neighboring memories and neighboring physical qubits, without constituting universal local processors [13, 14]. That is, we develop simple cellular automata which efficiently perform active error correction on the toric code. The operational principles of our automata are based on the mediation of attractive long-range interactions between excitations. That way excitations will tend to collapse together rather than to extend out and create logical errors in the code. Inspired by classical electrostatics and gravitational fields, we identify local update rules that induce such error correcting long range fields.

We provide extensive numerical analysis of three specific decoders, two of which use a 2D automaton, and the third using a 3D automaton. Two of our decoders perform competitively when compared against conventional decoders that require more sophisticated hardware. We also provide detailed analysis of the equilibration and propagation characteristics of the fields generated by the automata, which justifies the choice of tunable parameters in our decoders. Finally, we study a class of long
range fields and their decoding capabilities, and observe that certain fields are too long range to reliably identify excitation pairs.

Our schemes share some features with other proposals which add an auxiliary system in order to enforce suppression of error creation [15–21]. Prior proposals typically use quantum auxiliary systems instead of classical auxiliary systems for studies of self-correcting memories, where full numerical investigations are prohibitively difficult. Our class of models in some sense combines the benefits of active and of passive memories. Indeed, provided that the cellular automaton update rules do not change in time, the decoding process can be interpolated down to continuous time. In this setting error correction acts simultaneously with error creation, effectively preventing the creating of long error strings. This would amount to a dissipative self-correcting memory [22, 23]. In this work, we focus on decoding devices but it should be clear that our design incentives promise a generalization to new schemes of dissipative self-correction.

RESULTS

Our model. The main purpose of this work is to present a fundamentally new class of decoders for the 2D toric code. The toric code has the important feature that it can be considered as a physical model with purely local interactions. In this model, the physical spins reside on the edges of a periodic \( L \times L \) lattice which we henceforth denote \( \mathbb{V} \), and the stabilizer operators act on the four neighboring spins in the star or plaquette formation (see the Methods section for a brief review of the toric code). Errors in the toric code appear as strings, where only the endpoints of the strings are revealed in the error syndrome measurements. We will refer to these endpoints as anyons, and they play the role of excitations in a classical statistical model [3, 24]. An error configuration is successfully decoded if all anyons are paired, such that all anyon paths are contractible loops on the torus. Throughout, we assume independent single site Pauli-spin \( X \) and \( Z \) errors at a rate \( p \). In this way both error sectors decouple, and we can treat them independently. Here we consider only \( X \) errors, the \( Z \) errors can be treated identically.

The decoders we propose were conceived with the following goal in mind: To arrive at purely local update rules that are as simple as possible. By this we mean that the decoding can be implemented with simple local units of computation that depend only on nearest neighbors. The most natural model for this type of computation is a cellular automaton, which is indeed the nature of our class of decoders. In particular, we will consider an auxiliary cellular automaton, whose purpose is to communicate long range information between anyons, that enables local decisions to correct errors in the toric code (see Fig. 1).

The automaton extracts anyon information from the physical qubits via local stabilizer measurements at each cell of the toric code lattice \( \mathbb{V} \), the resulting outcome is stored in some local registry which we label \( q_E(y) \). The automaton also uses an auxiliary classical system on another lattice \( \Lambda \), and at each cell \( x \in \Lambda \) and at each time \( t \) it stores a real number \( \phi_t(x) \). We will refer to the auxiliary system as the field generated by the \( \phi \)-automaton. The simplest auxiliary lattice has cells coinciding with the toric code cell (so \( \Lambda = \mathbb{V} \)), but we also consider larger dimensional auxiliary systems provided they include the toric code as a sub-lattice \( \mathbb{V} \subset \Lambda \). Every auxiliary lattice considered herein is either a 2D torus, so \( \Lambda \cong \mathbb{V} \), or a 3D torus with the \( x_3 = 0 \) plane coinciding with the toric code lattice. To simplify later expressions, we extend the domain of \( q_E \) to the whole auxiliary lattice with the understanding that anyons never leave the toric plane, so \( q_E(x) = 0 \) whenever \( x_3 \neq 0 \).

The system’s configuration at any given time is the triple \( \{E,q_E,\phi\} \), where \( E \) is the actual error configuration. Assuming ideal measurements, \( q_E \) is redundant as it is determined by the error configuration. The dynamics of the system are divided into update sequences, labeled by \( \tau \), in which \( \{E,q_E,\phi\} \) evolve. Each update sequence is subdivided into \( c+1 \) elementary steps, first there are \( c \) repetitions of a \( \phi \)-update (in which only \( \phi \) changes), followed by a single anyon update (a partial error correction where \( E \) and hence also \( q_E \) change). Since lattice cells only interact with their nearest neighbors, these elementary update rules must be local. In a single \( \phi \)-update from time \( t \) to \( t + 1 \), the field \( \phi_{t+1}(x) \) can only depend on \( \phi_t(x), q_E(x) \) and values \( \phi_t(y) \) where \( y \) is a neighbor of \( x \). For each excited cell (where \( q_E(x) = 1 \)) the anyon update rule decides whether the anyon hops to an adjacent site, and being similarly local, this rule depends only on \( \phi \) values at adjacent sites. With good design of update rules, the \( \phi \) values will mediate information between anyons about the location of their closest potential partners. Later, we derive suitable updates rules, which can be roughly characterized as: take the average of your neighbors and add the charge. This corresponds to a local discretization of Gauss’ law.

There are innumerable conceivable rules for anyon update, but we found the following simple rule functioned well: find the adjacent cell with (unique) largest field value, then move there with probability \( 1/2 \). Physically, an anyon hopping from \( x \) to \( y \) is implemented by a Pauli \( X \) applied to the qubit on the lattice edge separating cells \( x \) and \( y \). All anyon hops occur in parallel resulting in partial correction of the error string \( E \), to a new string \( E' \) and the anyon distribution similarly updates to

| Table 1. Summary of results for \( \phi \)-automaton decoders. |
|---------------------------------------------------------|
| Lattice (\( \Lambda \)) | 2D | 2D* | 3D |
| Field velocity (\( c \)) | \( \tau \) indep. | \( 1 + 0.2 \cdot \tau \) | \( 10 \cdot \log^2(L) \) |
| Threshold (\( p_\mathrm{th} \)) | N/A | 8.2% | 6.1% |
| Required sequences (\( \tau_{\mathrm{RT}} \)) | N/A | \( o(\log^{2.5}(L)) \) | \( o(\log(L)) \) |

\( \tau \) refers to the sequence index, where each sequence contains \( c + 1 \) elementary updates per cell. The decoder terminates after \( \tau_{\mathrm{RT}} \) sequences on average. For the 2D* decoder the rules for the field updates can be thought of as time dependent since \( c \) increases with \( \tau \). The smoothing parameter as defined in Eq. (4) is \( \eta = 1/2 \).
Our results are summarized in Table 1. We present three decoders called the 2D, 2D* and 3D automata. The 2D* and 3D cases yield asymptotically working decoders, whereas the simple 2D decoder only suppresses errors up to a finite size lattice (e.g. \( L \sim 60 \), see Fig. 2c). While the 2D decoder is expected to be most suitable for modest storage needs, no strict asymptotic threshold exists. However, if we let \( c \) increase with each sequence \( \tau \), then the resulting 2D* \( \phi \)-automaton has a threshold of \( \sim 8.2\% \) (see Fig. 2a). A 2D toric code embedded in a 3D \( \phi \)-automaton exhibits a noise threshold around \( \sim 6.1\% \) (see Fig. 2b) for a fixed \( c \). That is, \( c \) is constant across sequences, but scales logarithmically with \( L \). These thresholds are only a few percent below the best thresholds using a centralized computing architecture.

Ideally, we would like all update rules to be completely independent of the system size \( L \), and not to change throughout sequences. Yet, we find that this is not possible within the framework of \( \phi \) automaton decoders. Indeed, simulations show that in order for the decoder to converge in a time which is sub exponential in \( L \), the field has to have propagated a distance of order \( \log(L) \), which corresponds to a lower bound on the field velocity \( c \geq c_{\text{min}} \sim \log^2(L) \). The dependence on \( \log(L) \) can be understood by the fact that maximal clusters of errors below the error threshold are of order \( \log(L) \) (see Refs. [26, 27] and Methods), and the decoder must collapse these maximal clusters in a time proportional to their size. If \( c \) were to be taken independent of \( L \), then beyond a critical cluster size, the field contribution due to an anyon’s self-interaction would dominate the contribution from the anyons at the other end of the cluster, hence causing persistent flickering of the anyons and preventing the decoder from converging in a reasonable time. Thus, in any field based model, the field velocity must scale with the system size. Later in the paper, we give an explicit lower bound on \( c \) which is derived from the equilibration properties of the \( \phi \)-automaton.

Given that we want our class of decoders to be adaptable to a setting where measurement data is regularly refreshed, it is desirable to have update rules that are invariant throughout sequences. However, this appears not to be possible when the fields are restricted to two spatial dimensions (see Fig. 2c). The \( \phi \)-automaton decoder works asymptotically in three or higher dimensions because of the steady state profile of the field. Indeed, the steady state poisson field of a single charge in \( L \) dimensions is \( \log(L) \) in two dimensions and \( \log^2(L) \) in higher dimensions, where \( r \) is the distance from the source charge. From numerical simulations, we are lead to conclude that the 2D equilibrium field is too long range, and tends to break the cluster structure of the anyons by extending error strings rather than shrinking them. In higher dimensions the field profile decays steeply enough so that this is not the case. We also later address whether there is a proper threshold between the \( \log(r) \) and the \( 1/r \) decaying fields.

Should we not be concerned with the field velocity changing across sequences, the 2D* \( \phi \)-automaton decoder is preferable as it exhibits a higher threshold and can more easily be implemented in an integrated circuit type architecture [28, 29]. With \( c \) increasing linearly at each update sequence, this allows elements within the clusters to pair up, while simultaneously preventing the field from extending across clusters. The main mechanism responsible for the success of the 2D* decoder is analogous to the “Expanding Diamonds” decoder from Refs. [30, 31].
Finally, in Fig. 2d we summarize the runtime estimates of our two working $\phi$-automaton decoders. The 3D decoder completes an error correction cycle, pairing all anyons, in the order of $\log(L)$ update sequences, while the 2D$^*$ decoder terminates after $\sim \log^{2.5}(L)$ sequences. This can be seen as very strong evidence that the typical maximal error cluster is indeed of size $\sim \log(L)$, and that our decoders collapse these clusters in optimal time. The 2D$^*$ decoder has a runtime $\sim \log^{2.5}(L)$ sequences as the value of $c$ needs to exceed $\sim \log^2(L)$ before the $\phi$-automaton can transmit information across a maximal cluster to collapse it. It should be noted that our estimate of the exponent of 2.5 is based on fitting a poly-logarithmic function using 11 data points, and could be slightly off. Finally, each sequence is composed of $c$ field updates and one anyon update, and so we can also quantify runtime in units of updates. For the 3D decoder, $\log(L)$ sequences with $c = 10 \cdot \log^2(L)$ gives a runtime of order $\log^3(L)$ updates. For the 2D decoder $\log^{2.5}(L)$ sequences with $c = 1 + \tau/5$, we sum over $\tau = 1, \ldots, \log^2(L)$ to get a runtime of order $\log^5(L)$ updates.

The $\phi$-automaton. In this section we explore in detail the local update rules of the $\phi$-automaton from physically motivated considerations. The key idea is to emulate a long-range field, such as from Newton’s law of gravitation or Coulomb’s law, which mediates an attractive force between the excitations of the toric code. In nature such long range forces emerge from simple local dynamics in the field. Here we achieve the same effect on a discrete lattice governed by a cellular automaton. The notion of mutually attracting particles with equal charge rather corresponds to gravitational fields, but we will stick to a language that is mainly inspired by electrodynamics.

To elaborate on this analogy, we detour into electrostatics, where the electric field is the gradient of a scalar potential $\Phi$. Gauss’s law simplifies to Poisson’s equation

$$\nabla^2 \Phi(x) = \sum_{j=1}^{D} \frac{d^2 \Phi(x)}{dx_j^2} = q(x), \quad (1)$$

where $D$ is the spatial dimension and $q$ is the charge distribution. The only isotropic solution of Poisson’s equation with a single charge at the origin is

$$\Phi(x) = \begin{cases} -\log r & \text{for } D = 2, \\ r^{-D} & \text{otherwise}, \end{cases} \quad (2)$$

where $r = \text{dist}(x, 0)$ denotes the distance of some lattice site $x$ from the source charge at the origin. The minus in front of the logarithm is chosen to ensure that $\Phi$ is convex for all $D$. The variety of long range behavior motivates the use of $\Phi$ as an information mediator. Our goal is now to approximate $\Phi$ via a cellular automaton.

A convenient discretization of the derivative (in 1D) is $d: \phi(x) \mapsto \phi(x + 1/2) - \phi(x - 1/2)$. Applying this prescription twice yields the double derivative $d^2: \phi(x) \mapsto \phi(x + 1) + \phi(x - 1) - 2\phi(x)$. On a $D$-dimensional square lattice $\Lambda$, this generalizes to the discrete Laplacian operator

$$\nabla^2 \phi(x) = -2D \phi(x) + \sum_{y \neq x} \phi(y), \quad (3)$$

where the sum $\langle y, x \rangle$ is over all cells $y$ neighboring cell $x$; so all $y$ for which $\text{dist}(x, y) = 1$.

We proceed by specifying a set of dynamical equations, or $\phi$-automata update rules, whose stationary distribution satisfy the discrete Poisson equation $\nabla^2 \phi(x) = C q(x)$. Here the role of the charge is replaced by anyonic excitations, and we identify $q$ with $q_E$. Note that only gradients of $\phi$ will be considered meaningful for anyon movement, and so fields differing by an additive constant are deemed equivalent. Invoking the Jacobi method [32], we consider the following $\phi$-automaton update rule

$$\phi_{t+1}(x) = (1 - \eta) \phi_t(x) + \frac{\eta}{2D} \sum_{y \neq x} \phi_t(y) + q_E(x), \quad (4)$$

where $0 < \eta < 1/2$ is a smoothing parameter we can freely choose and for convenience the unit charge is set to $C = -2D/(1 - \eta)$. These update rules are manifestly local. We can cast Eq. (4) as a matrix equation $\phi_{t+1} = G \phi_t + q_E$ which should be read as

$$\phi_{t+1}(x) = \sum_y G_{x,y} \phi_t(y) + q_E(x), \quad (5)$$

where $G$ is a doubly stochastic block circulant matrix encoding the $\phi$-automaton update steps. This reformulation allows us to leverage the machinery of matrix analysis [33].

Stationary states and equilibration. Assume for now that the anyon configuration is fixed. By recursively iterating Eq. (5), we get

$$\phi_t = G^t \phi_0 + \sum_{m=0}^{t-1} G^{t-m} q_E. \quad (6)$$

It is easy to see that in general the $\phi$-automaton satisfying Eq. (4) does not converge, since the total field values build up steadily due to the charges. Indeed, when $Q = \sum_{x} q_E(x)$ denotes the total charge present in the lattice, the total field increases by $Q$ with each anyon update. However, it is important that the gradient of the field equilibrates towards a fixed value. With this in mind, we define $\phi_t$ to be the rescaled version of $\phi_t$ such that $\tilde{\phi}_t(x) = \phi_t(x) - K$, where the constant $K$ is chosen such that $\sum_x \tilde{\phi}_t(x) = 0$.

The matrix $G$ can be diagonalized by Fourier transform (see the Supp. Mat.), allowing us to obtain a unique solution such that $\tilde{\phi}_t \to \tilde{\phi}_\infty$ as $t \to \infty$. We find that the stationary field produced by a single anyon at the origin is

$$\tilde{\phi}_\infty(x) = \varphi(x) := L^{-D} \sum_{k \in \Lambda, k \neq 0} (1 - \lambda_k)^{-1} e^{i k \cdot x}, \quad (7)$$

where the sum is over all Fourier components except the zero vector and $\lambda_k$ are the eigenvalues of the matrix $G$,

$$\lambda_k = 1 - \eta + \frac{\eta}{D} \sum_{j=1}^{D} \cos(k_j). \quad (8)$$

The $k = 0$ term is excluded from the sum in Eq. (7) which would provide an additive constant to the field. By omitting
Anyon update

Figure 3. Self-interaction A setup with two isolated anyons $A$ and $B$, and their fields in a self-interaction cycle. Suppose that the fields from each anyon starts in its stationary configuration (a). After one anyon update move, anyon $A$ moves in the direction of largest field gradient – towards anyon $B$. (b) Immediately after anyon $A$ moves, the field which it leaves behind is much larger that the field from anyon $B$. (c) After a certain number of field updates, the residual self field from anyon $A$ decreases, but might still tell the anyon to move in the wrong direction (d).

The distance between $\tilde{\phi}_1$ and $\tilde{\phi}_\infty$ (as measured by Euclidean distance) can be shown by using matrix inequalities to decrease exponentially fast, so that (details in Supp. Mat.)

$$\|\tilde{\phi}_1 - \tilde{\phi}_\infty\|_2 \leq e^{-(\eta^2/2) t/L^2} \|\phi_0 - \phi_\infty\|_2,$$

where the distance is measured in the vector 2-norm $\|u\|_2 = u \cdot u$. This would suggest a relaxation time of the $\phi$-automaton of the order of $L^2$, which can be understood as diffusive spreading of information. In the following section we will argue that the relevant contribution of the field converges in a time much faster for our decoders, because we are only interested in information propagating on distance scales of the order $\log(L)$, the maximal error cluster size.

Self interaction. The $\phi$-automaton carries information about distant anyons through local update rules. As has already been hinted at in the previous section, information takes time to propagate through the field, and the most relevant manifestation is in the pernicious phenomenon of self interaction. Consider two close-by anyons $A$ and $B$, as in Fig. 3. Our story begins with anyon $A$ moving in the direction of largest local field gradient, towards anyons $B$. After anyon $A$ has moved, it leaves behind its own field, which will typically be a local maximum of the $\phi$-automaton. In order for anyon $A$ at the end of the following sequence to again move in the direction of anyon $B$, the field gradient generated by anyon $B$ has to be larger than the residual self field gradient around the cell of anyon $A$. We call this phenomenon self interaction, and next derive conditions on the field velocity to prevent this from occurring.

Suppose anyon $A$ starts at the origin and anyon $B$ is at cell $y$. Assuming the initial configuration is at equilibrium, $\phi_0(x) = \varphi(x) + \varphi(x-y)$. After anyon $A$ moves, the field begins to relax towards $\phi_\infty(x) = \varphi(x-e) + \varphi(x-y)$, where $e$ is a unit vector in the direction of anyon $B$. We want to estimate the time required before the field gradient generated by anyon $B$ is larger than the self-interaction of anyon $A$. In other words, assuming $\phi_0(x) = \varphi(x) + \varphi(x-e)$, after what time do we have

$$|\phi_t(x) - \phi_\infty(x)| \leq g_t,$$

where $g_t$ is the gradient at the origin due to a charge at a distance $l = \text{dist}(0, y)$. It is not difficult to see that the left hand side of Eq. (10) does not depend on the field from anyon $B$.

A straightforward, but lengthy, calculation (in the Supp. Mat.) shows that for all $\epsilon > 0$

$$|\phi_t(x) - \phi_\infty(x)| \leq \epsilon$$

whenever

$$t \geq \chi(x)e^{-2/(D-1)},$$

where $\chi$ is a function of position, but is independent of $t$ (see Supp. Mat. for the exact form). Considering the second anyon at point $y = e$, it creates a field gradient near the first anyon (at the origin)

$$\nabla \varphi(x-y)|_{x=0} = \sum_j (\varphi(-y) - \varphi(e_j - y)) \cdot e_j,$$

where $\{e_j\}$ is an orthonormal basis of unit vectors. Therefore, we must require that

$$\chi(0) \epsilon \leq |\nabla \varphi(x-y)|_{x=0} := g_1.$$  

By percolation arguments, the maximum cluster size of anyons can be argued to typically be order $l \sim \log(L)$. Given that the gradient is of the order $l^{1-D}$, require $\epsilon \sim \log(L)^{1-D}$,

$$c \geq c_{\text{crit}} \sim \log^2(L),$$

where the dependence on $D$ has now cancelled.

The field profile. We already discussed in detail that when the field velocity $c$ is kept invariant throughout sequences, the 2D and 3D decoders exhibit fundamentally different behavior. Here we supplement these results by investigating another decoder model, not a cellular automata, to investigate the large $c$ regime when fields are always close to stationary solutions. In order to address this question, we consider a class of power law potentials

$$\Phi(r) = r^{-\alpha}$$

for $\alpha > 0$. 

We simulate anyon movements in these perfect fields, by inserting the sum of all anyon fields by hand, instead of simulating it locally by a cellular automaton. We alternate between these instantaneous field updates and the anyon update rule as defined before. For $\alpha = 1$ we expect a behavior similar to the decoder with 3D $\phi$-automaton. The benchmark presented in Fig. 4b indeed shows a threshold at $p_{th} \sim 6.3\%$, which is very close to the threshold of $\sim 6.1\%$ seen for the 3D $\phi$-automaton. This independently confirms that our choice of finite $c$ is sufficiently large for equilibration between distinct anyon moves.

Seeking to understand the transition between the 2D and the 3D stationary field behavior, we consider various values of $\alpha$ and lattice sizes for fixed $p = 5\%$ in Fig. 4a. In general the decoder fail rate reduces with increasing $\alpha$. This supports our claim that fields with a shorter range are more suited for efficient shrinking of errors. However, our main observation is that for $\alpha < 0.5$ the decoders fail rate increases drastically with the system size $L$. This means that the decoder does not exhibit a threshold above $p = 5\%$ for $\alpha < 0.5$. Although it is difficult to certify numerically, for $0.5 \leq \alpha \leq 0.7$ we did not either find an indication of a threshold at any $p$ up to lattice size $L = 400$.

From these observations we infer that there has to exist a critical $\alpha_T$ such that $\Phi$ provides a asymptotically-working decoder. For values of $\alpha$ below the transition, the increased contribution from far away anyons leads to the misidentification of error strings. Recall that anyons move according to the maximal field gradient of $r^{-\alpha}$, which is proportional to $r^{-3(\alpha + 1)}$. The total field at a point is the sum of the contributions from all of the anyons, hence for an infinite lattice the gradient is only guaranteed to be finite for $1/r^\beta$, when $\beta > 2$. Only for $\alpha > 1$ does the gradient have a well defined value asymptotically.

Hence we expect $\alpha_T = 1$ for the 2D toric code. This intuition is further supported by a simple simulation of the 1D toric code (i.e., the repetition code), where we observe a clear transition at $\alpha = 0$, which corresponds to a gradient of $1/r$. Here, we have allowed for values of $\alpha \leq 0$ by setting $\Phi(r) = -r^{-\alpha}$ for $\alpha < 0$ and $\Phi = -\log(r)$ for $\alpha = 0$.

We highlight that very high values of $\alpha$ are not favorable in general, since they require increased precision in the field resolution. Meanwhile, we expect that the 3D $\phi$-automaton always gives rise to a field with slightly shorter range than $1/r$ (i.e., $\alpha > 1$). Therefore, the 3D $\phi$-automaton seems to be sitting exactly at the sweet spot, providing a functioning long range $\phi$-field decoder with a maximally robust field.

DISCUSSION

In this work, we have introduced a new class of decoders that pair anyonic excitations by mediating long range information through an auxiliary field. The field, and the anyon movement, is generated by a cellular automaton, in such a way that the decoder has an intrinsically parallelized architecture. We have identified two particular decoders within this class that exhibit an error threshold, one requiring a 3D auxiliary system with homogeneous update rules, with an error threshold above $6.1\%$, and another with a 2D auxiliary system and time dependent update rules with an error threshold above $8.2\%$. These threshold values, although smaller than the best decoders using the minimum weight perfect matching (MWPM) algorithm [34] or Monte Carlo techniques [10, 11], are similar to recent popular proposals based on renormalisation group techniques [8, 9] or so-called expanding diamonds [30, 31]. An approximate MWPM decoder can be parallelized down to a local communication length of $O(\log L)$ [12]. Whereas, our decoders have a strict locality neighborhood, and are hence potentially very attractive for implementations of topological quantum memories in integrated circuit type architectures, as were recently proposed in Refs. [28, 29].

The use of an auxiliary system to mediate long-range information between the anyons has recently attracted broader attention in the field of quantum memories [15, 18, 35–37]. Most of these proposals are based on coupling of the toric code to another quantum system (e.g. cavity modes [36] or a boson crystal [37]). Those proposals have the drawback that the full dynamics of the auxiliary system is difficult to certify by simulation, and so a detailed time scale or threshold analysis is hard to obtain. Rather, those approaches assume rapid equilibration of the auxiliary system, and then derive effective Hamiltonians on the toric code qubits. Such quantum memories are then so-called passive memories; i.e., protection against noise is not provided by measurement and feedback, but rather by a special subspace of the effective Hamiltonian that is naturally resistant to certain types of noise. An exception to the above class of protocols is Ref. [18], which is more similar to our proposal as it uses a classical auxiliary system. However, it has the severe drawback that it requires linearly growing local energies, and numerical evidence is limited to small systems (up-to $L = 20$) and extensions to larger systems rest on scant analytic foundations. In contrast, our schemes have been extensively explored numerically, and two of them show very clear evidence of a

![Figure 4](image-url)
threshold. Furthermore, we have very good control on the time scales in our system, which clearly show that provided the field velocity \( c \) is of order \( \log^2(L) \), the system completes an error correction cycle in sub-polynomial time.

Our 3D decoder can be made to work in continuous time by considering an asynchronous application of the \( \phi \)-automaton, and updating the field and the anyon movement in continuous time, at rates which incorporate the field velocity \( c \). This amounts to a dissipative self-correcting memory, as considered in Refs. [18, 22, 23]. Preliminary numerical investigations have shown that the continuous asynchronous application of the \( \phi \)-automaton memory exhibits the same qualitative behavior as the decoder. A full presentation and analysis of this extension will be provided elsewhere.

Our schemes have two main conceptual imperfections in their present formulation. First, that the field velocity \( c \) has to scale logarithmically with the system size \( L \), and second, that the resolution of the field, when represented digitally, also scales logarithmically with \( L \). In analogy to the seminal work of Gacs [13, 38] on the resolution of the positive rates conjecture, one could expect that there exists a local cellular automaton decoder with update rules that are strictly system size independent. Harrington sketched such a program for toric code decoders [14], but only gave explicit details for a scheme with logarithmic overhead. Unfortunately, in Gacs’ original work, the noise threshold was prohibitively small, rendering the proposal practically infeasible. Furthermore, Gacs’ or Harrington’s update rules are very complicated, requiring more sophisticated computing hardware than our scheme. Our proposals have high thresholds with only logarithmic system size dependencies for local parameters, and are therefore fully adequate for realistic implementations.

We have focused on some specific instances of cellular automata decoders working against a particular noise model, but the research project opens up many new possibilities within the same paradigm. The space of potential cellular automata is vast, small variations such as to anyon movement rules or lattice geometry could have substantial consequences on performance. Gauss’s law has proved an invaluable tool to our intuition, but it was just a guide, and departures from an electrostatic mindset could prove rewarding. We expect our models to be naturally robust to certain types of more invasive errors, because the field encodes global information in a smooth and local manner. Nevertheless, a more detailed study of correlated noise and errors in the measurements and in the field would be a valuable extension of the present study. The latter two points are especially interesting stepping stones towards extending our proposal to a passive dissipative quantum memory. Exploring this wealth of decoder models will surely reveal many technological opportunities and a rich world of varied automata behaviors.

### METHODS

**Toric codes and noise models** [3, 5, 39]. Any lattice that is locally two-dimensional defines a surface code. A lattice is composed of faces, edges and vertices. At each edge resides a qubit. Each face \( f \) is surrounded by a set of edges, which we denote \( \mu(f) \). Similarly, each vertex, \( v \), has a set of edges incident to the vertex, denoted \( \mu'(v) \). Faces and vertices have associated operators

\[
S_f = \bigotimes_{e \in \mu(f)} Z_e, \quad S_v = \bigotimes_{e \in \mu'(v)} X_e. \tag{17}
\]

We consider square lattices on the torus, which are equivalent to an \( L \times L \) square lattice with periodic boundary conditions. Consequently, we can label vertices using a vector of two integers modulo \( L \), denoted \( v \in \mathbb{Z}^2 \). A quantum state is in the code space, or ground state, when it is in the “+1” eigenspace for all face and vertex operators. If any face (vertex) operator is measured and yields “−1”, then we say an excitation, or anyon, is present at that face (vertex).

A Pauli operator acting on the code space, such as from noise, will typically produce anyons. Any Pauli can be written as \( P \propto X[E]Z[E'] \), where \( E \) denotes a set of edges and \( X[E] \) and similarly for \( Z[E'] \). The \( X[E] \) component creates face anyons, whereas \( Z[E'] \) creates vertex anyons. The two problems are identical, or dual, and so herein we discuss only \( X[E] \) operators. For a \( X[E] \) error, we use \( q_E \) as the indicator function for excitations at faces. We see \( X[E] \) creates an anyon at face \( f \) (and so \( q_E(f) = 1 \)) iff the set \( E \) has an odd number of edges which are facets of \( f \) (and otherwise \( q_E(f) = 0 \)). For \( E \) that form a path of edges, anyons appear at the endpoints of the path. A loop, also called a cycle, has no endpoints, and so no anyons (\( q_E = 0 \)). Furthermore, a non-contractible loop around the torus yields a non-trivial unitary within the code space with no accompanying anyons.

Here we design decoders that take the anyon information from a random error \( X[E] \) and determine a recovery operator \( X[R] \), with the goal that \( X[R]X[R] \) acts as the identity on the code space, e.g. when \( R = E \). We fail to decode when \( X[E]X[R] = X[W] \), with \( W \) being a non-contractible error. Essentially, decoding consists of pairing up anyons with the closest partner. Since we observe anyons at endpoints of error chains, this pairing process requires long range information telling us the distance between anyons. The contribution of this work is to provide a simple local method of communicating this information between anyons.

**Error model.** Throughout we assume random noise that is uncorrelated between \( X \) and \( Z \) errors, such that \( X[E]Z[E'] \) occurs with probability \( \Pr(E) \Pr(E’) \) where \( \Pr(E) = \mu(E)(1 - p)^{L^2 - |E|} \). We compare the probability of a decoder failing, with the initial error probability \( p \). A suitable decoder will have a threshold \( p_{th} \), such that for all \( p < p_{th} \), the failure probability can be arbitrarily reduced by increasing \( L \). Equivalently, the expected lifetime of quantum information in the code space is preserved for arbitrarily long time, by increasing the lattice size. The merit of the uncorrelated noise model lies in its simplicity, with \( X \) and \( Z \) errors as statistically independent we only need to simulate half the problem. Furthermore, decoders with the desired threshold behavior for uncorrelated noise will also have a threshold for depolarizing, thermal and other noise models.

The spatial structure of error configurations is essential for the working principle of our decoder. In Ref. [26] it was shown that error strings with a size larger than \( O(\log L) \) are exponentially unlikely in all quantum LDPC codes. The toric code is one of the simplest and most useful instance of a quantum LDPC code [26]. However, there is a constant probability for a decoder to miss-identify strings, which for our decoder would correspond to creating larger strings first before contracting them. In Theorem 3, they show that the size of objects that have to be misidentified – so called violating \( (a, m) \) sets – is also \( O(\log L) \) in the same sense as before. In Ref. [27] it was outlined that this proof only applies to minimum weight perfect matching decoders. However, the observed logarithmic scaling of the recovery sequences in our decoder (see Fig. 2d) has to be ascribed to similar properties of the spatial error structure.

**Numerical details.** The pseudo code for our \( \phi \)-automaton decoders is sketched in Table 2. All numerical results provided in this work, were produced using a C program. While the algorithm is highly optimized for speed on a local parallelized architecture, especially the simulation of the 3D \( \phi \)-automaton on a traditional architecture can be quite expensive. To optimize the benchmark performance, every decoding process is aborted and considered a fail if the number of sequences exceeds \( \tau \geq 10 \cdot L \) (2D, 2D*) or \( \tau \geq L \) (3D). This is justified by the fact that we are looking for a decoder which has an efficient runtime. It is easy to see that the results on required sequences presented in Fig. 2d are not influenced by the exit condition. Creating the threshold plot
presented in Fig. 2b still took approximately one month on 500 CPUs. An open Python implementation of our decoders is available [40].

Table 2. Simplified pseudo code.

| Repeat \( c \) times: |
|--------------------|
| **Parallel for all** \( \xi \in \Lambda \): |
| \( \phi(\xi) = \frac{1}{|\Lambda|} \sum_{\xi'} \phi(\xi') + q(\xi) \) |
| **Parallel for all** \( x \in V \): |
| \( q(x) \neq 0 \): |
| With probability 0.5: |
| Move anyon from \( x \) to \( y \) of max \( \phi(y) \) \( (y,x) \) |
| **If still anyons present:** |
| \( \tau = \tau + 1 \) |
| Go to beginning |

Note that for the 2D* decoder the field velocity \( c \) depends on the sequence \( \tau \).

ACKNOWLEDGEMENTS

We thank J. Preskill and F. Pastawski for fruitful discussions, and S. Bravyi, K. Michnicki, and A. Hutter for helpful comments on the manuscript. MJK especially thanks K. Michnicki for very valuable insight and motivation in the buildup to the present work. We also thank D. Poulin for – upon completion of the present work – making us aware of the existence of unpublished results by D. Poulin and G. Duclos-Cianci, who have previously considered cellular automata based decoders in the same spirit as us [41]. A set of slides from 2011 alluding to their work are available online [42]. We acknowledge support from the EU (SIQS, RAQUEL, COST), the Alexander-von-Humboldt Foundation, the FQXi, the BMBF, and the ERC (TAQ).

1. D. Aharonov and M. Ben-Or, SIAM J. Comput. 38, 1207 (2008)
2. D. Gottesman, Phys. Rev. A 57, 127 (1998)
3. E. Dennis, A. Y. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. 43, 4452 (2002)
4. A. Kitaev, Ann. Phys. 321, 2 (2006)
5. A. A. Kovalev and L. P. Pryadko, Phys. Rev. A 86, 20304 (2013)
6. B. Yoshida, Annals of Physics 326, 2566 (2011)
7. D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Phys. Rev. A 83, 020302 (2001)
8. G. Duclos-Cianci and D. Poulin, Phys. Rev. Lett. 104, 050504 (2010)
9. S. Bravyi and J. Haah, Phys. Rev. Lett. 111, 200501 (2013)
10. J. R. Wootton and D. Loss, Phys. Rev. Lett. 109, 160503 (2012)
11. A. Hutter, J. R. Wootton, and D. Loss, Phys. Rev. A 89, 022326 (2014)
12. A. G. Fowler, arXiv:1307.1740 (2013)
13. F. Gács, J. Stat. Phys. 103, 45 (2001)
14. J. W. Harrington, “Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes,” (2004), http://thesis.library.caltech.edu/1747/1/jimh_thesis.pdf
15. A. Hamma, C. Castelnovo, and C. Chamon, Phys. Rev. B 79, 245122 (2009)
16. F. L. Pedrocchi, S. Chesi, and D. Loss, Phys. Rev. B 83, 115415 (2011)
17. F. L. Pedrocchi, A. Hutter, J. R. Wootton, and D. Loss, Phys. Rev. A 88, 062313 (2013)
18. K. Fuji, M. Negoro, N. Imoto, and M. Kitagawa, “Measurement-free topological protection using dissipative feedback,” E-print arXiv:1401.6350
19. A. Hutter, J. R. Wootton, B. Röhlüsberger, and D. Loss, Phys. Rev. A 86, 052340 (2012)
20. F. L. Pedrocchi, A. Hutter, J. R. Wootton, and D. Loss, “Local 3D spin Hamiltonian as a thermally stable surface code,” (2009), e-print arXiv:1209.5289
21. D. Becker, T. Tanamoto, A. Hutter, F. L. Pedrocchi, and D. Loss, Phys. Rev. A 87, 042340 (2013)
22. F. Pastawski, L. Clemente, and J. I. Cirac, Phys. Rev. A 83, 012304 (2011)
23. M. J. Kastoryano, M. M. Wolf, and J. Eisert, Phys. Rev. Lett. 110, 110501 (2013)
24. H. Bombin, R. S. Andrist, M. Ohzeki, H. G. Katzgraber, and M. A. Martin-Delgado, Phys. Rev. X 2, 021004 (2012)
25. F. H. E. Watson and S. D. Barrett, “Estimating overheads for topological quantum codes,” (2013), e-print arXiv:1312.5213
26. A. A. Kovalev and L. P. Pryadko, Phys. Rev. A 87, 020304 (2013)
27. D. Gottesman, “What is the overhead required for fault-tolerant quantum computation?” (2013), e-print arXiv:1310.2984
28. A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A 86, 032324 (2012)
29. R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. White, J. Mutus, A. Fowler, B. Campbell, et al., Nature 508, 500 (2014)
30. E. Dennis, Ph.D. thesis, UCSB (September 2003)
31. J. R. Wootton, V. Lahtinen, B. Doucot, and J. K. Pachos, Ann. Phys. 326, 2307 (2011)
32. Y. Saad. *Iterative methods for sparse linear systems* (Siam, 2003)
33. R. Bhatia, *Matrix analysis* (Springer, Heidelberg, 1997)
34. A. G. Fowler, A. M. Stephens, and P. Groszkowski, Phys. Rev. A 80, 052312 (2009)
35. S. Chesi, B. Röhlüsberger, and D. Loss 82, 022305 (2010)
36. F. L. Pedrocchi, S. Chesi, and D. Loss, Phys. Rev. B 83, 115415 (2011)
37. F. L. Pedrocchi, A. Hutter, J. R. Wootton, and D. Loss, Phys. Rev. A 88, 062313 (2013)
38. L. F. Gray, J. Stat. Phys. 103, 1 (2001)
39. R. Raussendorf and J. Harrington, Phys. Rev. Lett. 98, 190504 (2007)
40. http://users.physik.fu-berlin.de/~herold/phi-automaton-decoder/
41. D. Poulin, private communication
42. http://www.physique.usherbrooke.ca/poulin/utilisateur/files/seminaires/2011Simon.pdf
Supplementary Material

The matrix formalisation

In Eq. (5), we have introduced the matrix formulation of Eq. (3) and the associated $G$ matrix. By comparing these equations, we see

$$G_{x,y} = \begin{cases} 
1 - \eta & \text{for } \text{dist}(x,y) = 0, \\
\eta/(2D) & \text{for } \text{dist}(x,y) = 1, \\
0 & \text{for } \text{dist}(x,y) > 1.
\end{cases} \tag{18}$$

The matrix is manifestly invariant under translations, so that $G_{k+x,k+y} = G_{x,y}$ for all $k$, modulo the system size, and such matrices are said to be block circular. Furthermore, it has the property that $\sum_x G_{x,y} = 1$ and $\sum_y G_{x,y} = 1$, and so is doubly stochastic. A consequence of doubly stochasticity is that is preserves the total field potential

$$\sum_{x,y} G_{x,y} \phi_t(y) = \sum_y \phi_t(y). \tag{19}$$

It is instructive to consider the total field potential at a time $t$, which is $T_t = \sum_y \phi_t(y)$. If no anyons are present, so $\phi_{t+1} = G \phi_t$, then $T_{t+1} = T_t$. However, in the presence of anyons $\phi_{t+1} = G \phi_t + q$ and so $T_{t+1} = T_t + Q$, where $Q = \sum_x q(x)$. In this setting, the absolute field never equilibrates, though rescaling the fields resolves this problem. Note that for practical implementations it is possible to avoid the infinity growing field by local generation of the gradient field $\nabla \phi$. For simplicity we will restrict ourselves to the analysis of the generation of the potential $\phi$.

Diagonalizing the update matrix

Since the update rules are translationally invariant, the associated update matrix $G$ is block circulant. This entails that the Fourier transform can be employed to diagonalize this matrix and so provide solutions to many dynamical questions. The kernel of the multi-dimensional Fourier transform $F$ has entries

$$F_{j,k} = \frac{1}{L^{D/2}} e^{-2\pi i j k / L}. \tag{20}$$

$G$ can be diagonalized as

$$(FGF^\dagger)_{j,k} = \hat{\delta}_{j,k} \sum_{l \in \Lambda} G_l e^{-2\pi i j l / L}. \tag{21}$$

What is more,

$$G_{j,k} = G_{j-k} = \frac{1}{L^D} \sum_{l \in \Lambda} \lambda_l e^{2\pi i (j-k) / L}, \tag{22}$$

where the second expression reflects the block circulant property of $G$. The eigenvalues of $G$ are given by

$$\lambda_k = \sum_{l \in \Lambda} G_l \cos(2\pi l \cdot k / L). \tag{23}$$

Using Eq. (18) one immediately finds

$$\lambda_k = 1 - \eta + \frac{\eta}{D} \sum_{j=1}^D \cos(2\pi k_j / L) \tag{24}$$

for $k = (k_1, \ldots, k_D)^T$. Furthermore, the eigenvectors are

$$v_k(x) = e^{2\pi i k \cdot x / L} / L^{D/2}, \tag{25}$$

which are simply Fourier waves.

Stationary field configurations

In this sub-section, we identify field configurations that are stationary under field updates assuming a fixed anyon configuration. As defined in the main text, we use $\tilde{\phi}_t$ to denote a rescaled field $\phi_t$ so that $\sum_x \tilde{\phi}_t(x) = 0$. The renormalized fields equilibrate to a stationary value, and we now set out to estimate the equilibration time. The dynamical equation for the field is

$$\dot{\phi}_t = G^t \phi_0 + \sum_{m=0}^{t-1} G^m q. \tag{26}$$

Using $\tilde{\phi}_0 = \phi_0 - T_0$ and $\tilde{\phi}_t = \phi_t - (T_0 + Qt)$, we have

$$\dot{\tilde{\phi}}_t = G^t \tilde{\phi}_0 + \sum_{m=1}^{t-1} G^m q E - Qt. \tag{27}$$

Let us consider the large $t$ limit. Since $G$ is doubly stochastic, the absolute value of the eigenvalues are no greater than unity, and indeed Eq. (24) shows that only the $k = 0$ subspace has a unit eigenvalue. Consequently, $G^t$ tends to the projector onto the $v_0$ vector. However, all rescaled fields have no $v_0$ component and so the first term vanishes. Considering the second term, and denoting by $P_k = v_k v_k^\dagger$ a projector onto the eigenvector $v_k$, we have

$$\sum_{m=1}^{t-1} G^m = \sum_k \sum_{m=0}^{t-1} \lambda_k^m P_k \to \sum_k (1 - \lambda_k)^{-1} P_k, \tag{28}$$

where we have used the well-known limit of geometric series. So far we have

$$\tilde{\phi}_t \to \sum_k (1 - \lambda)^{-1} P_k q E - Qt. \tag{29}$$

Again using the normalization condition, we know the $k = 0$ and constant $-Qt$ terms must cancel exactly, so that

$$\tilde{\phi}_t \to \tilde{\phi}_\infty = \left( \sum_{k \neq 0} (1 - \lambda_k)^{-1} P_k \right) q E. \tag{30}$$

Using $P_k = v_k v_k^\dagger$, we find

$$\tilde{\phi}_\infty(x) = \sum_{k \in \Lambda \setminus \{0\}} a_k e^{2\pi i k \cdot x / L}, \tag{31}$$

where $a_k$ is a constant depending on $q$ and $\eta$.
where
\[ a_k = \frac{\sum_x e^{-2\pi i \cdot k \cdot x/L} q_E(x)}{L^D (1 - \lambda_k)}. \] (32)

For a single anyon at the origin, the stationary solution is defined as \( \varphi(x) \) and the Fourier coefficients simplify to \( a_k = (L^D (1 - \lambda_k))^{-1} \), so that
\[ \varphi(x) = \sum_{k \neq 0} e^{2\pi i \cdot k \cdot x/L} \frac{a_k}{L^D (1 - \lambda_k)}. \] (33)

More generally, a single anyon at cell \( y \) has a stationary field with \( a_k = e^{-2\pi i \cdot k \cdot y/L} (L^D/2 (1 - \lambda_k))^{-1} \), so that
\[ \varphi_y(x) = \sum_{k \neq 0} e^{-2\pi i \cdot k \cdot y/L} \frac{e^{2\pi i \cdot k \cdot x/L}}{L^D (1 - \lambda_k)}. \] (34)

One can verify that this solution agrees with translation invariance since \( \varphi_y(x) = \varphi(x - y) \). For more than a single anyon the stationary state is a linear combination of the contribution from each anyon.

**Global equilibration**

We now proceed to estimate the number of field updates that are needed for a non-equilibrium field, say \( \phi_0 \), to approach a stationary field \( \phi_{\infty} \). We first observe that we can time evolve \( \phi_{\infty} \) by \( t \) time steps, and by virtue of being a stationary state, we have
\[ \phi_{\infty} = G^t \phi_{\infty} + \sum_{m=1}^{t-1} G^m q - Qt. \] (35)

Taking the difference of \( \phi_{\infty} \) and \( \phi_t \), only the first terms remain
\[ \phi_t - \phi_{\infty} = G^t (\phi_0 - \phi_{\infty}). \] (36)

By the Hölder inequality, for a single charge at the origin,
\[ \| \phi_t - \phi_{\infty} \|_2 \leq \lambda_{\text{max}} \| \phi_0 - \phi_{\infty} \|_2, \] (37)

where fields are measured in the vector 2-norm \( \| u \|_2^2 = u \cdot u \), and \( \lambda_{\text{max}} = \max_{k \neq 0} |\lambda_k| \). Again, the \( k = 0 \) eigenvalue can be neglected since the fields are normalized to have no overlap with this eigenvector. Provided \( \eta \leq 1/2 \), we have that \( |\lambda_k| = \lambda_k \), and we find
\[ \lambda_{\text{max}} = (1 - \eta) + \frac{\eta}{D} \left( \cos(2\pi/L) + (D - 1) \right), \] (38)

and using \( \cos(x) \leq 1 - x^2/4 \) for \( x \in [0, 2] \), we find
\[ \lambda_{\text{max}} \leq 1 - \frac{\eta \pi^2}{D L^2}. \] (39)

for \( L \geq 4 \). Furthermore,
\[ 1 - x \leq e^{-x}, \] (40)

for all \( 1 \geq x \geq 0 \), so \( \lambda_{\text{max}} \leq e^{-\eta \pi^2/(D L^2)} \). Combined with Eq. (37) gives the result
\[ \| \phi_t - \phi_{\infty} \|_2 \leq e^{-(\eta \pi^2/D) t} L^2 \| \phi_0 - \phi_{\infty} \|_2. \] (41)

While the equilibration is exponentially fast in \( t \) for constant lattice size, this bound tells us little in the large lattice limit as \( \lambda_{\text{max}} \to 1 \). This is intuitive since after an event of anyon movement the field can only change within a light cone of the event. Consequently, for any period of time \( (t < L^2) \) there is always a distance at which the field has not equilibrated. However, we are interested in the local equilibration of a cluster of anyons in a small region, and contained within a larger lattice. This finer estimate is presented next.

**Convergence after anyon steps**

In this subsection, we consider the point-wise convergence of the field, following an anyonic hopping from \( x = 0 \) to \( x = -e \). The initial distribution is determined by \( \phi_{t=0}(x) = \varphi(x) \) for \( x \in \Lambda \), then at sufficiently late times \( \phi_{\infty}(x) = \varphi(x + e) \). Recall that
\[ \tilde{\phi}_{t=0}(x) = \sum_{k \neq 0} e^{-2\pi i \cdot k \cdot x/L} \frac{e^{-2\pi i \cdot k \cdot e/L}}{L^D (1 - \lambda_k)}. \] (42)

The final distribution after a shift by one position (see Eq. (34)) is
\[ \tilde{\phi}_{\infty}(x) = \sum_{k \neq 0} e^{-2\pi i \cdot (x+e) \cdot k/L} \frac{e^{-2\pi i \cdot k \cdot e/L}}{L^D (1 - \lambda_k)}. \] (43)

Therefore,
\[ \tilde{\phi}(0) - \tilde{\phi}_{\infty}(x) = \sum_{k \neq 0} e^{-2\pi i \cdot k \cdot x/L} \frac{e^{-2\pi i \cdot k \cdot e/L}}{L^D (1 - \lambda_k)}. \] (44)

Using Eq. (36), we find that at time \( t \)
\[ \Delta_t(x) = \sum_{k \neq 0} |\lambda_k|^t \frac{e^{-2\pi i \cdot k \cdot x/L} (1 - e^{-2\pi i \cdot k \cdot e/L})}{L^D (1 - \lambda_k)}, \] (45)

where \( \Delta_t(x) := \tilde{\phi}(x) - \tilde{\phi}_{\infty}(x) \). We know the left hand side is real and so may take the real part of the right, and using the triangle inequality
\[ \Delta_t(x) \leq \sum_{k \neq 0} |\lambda_k|^t \frac{A_k}{L^D B_k}, \] (46)

where we have collected many details into \( A_k \) and \( B_k \),
\[ A_k = |\text{Re}[e^{-2\pi i \cdot k \cdot x/L} (1 - e^{-2\pi i \cdot k \cdot e/L})]|, \] (47)
\[ B_k = 1 - \lambda_k. \]

The next few steps of the derivation just concern these quantities, and culminate in the bound
\[ \frac{A_k}{B_k} \leq \frac{D \| 2x + e \|_1}{2\eta}, \] (48)
where \( \|x\|_1 = \sum_j |x_j| \).

First, we recast the numerator \( A_k \). 

\[
A_k = |\cos(2\pi x \cdot k/L) - \cos(2\pi (e + x) \cdot k/L)|, \\
= 2 \sin \left( \frac{\pi}{L} k \cdot (2x + e) \right) \sin \left( \frac{\pi}{L} k \cdot e \right). 
\]  

(49)

Before proceeding we demonstrate the following trigonometric inequality: \( |\sin(un)| \leq |\sin(u)|/n \) if \( n \) is a positive integer. When \( n = 0 \) both sides are zero so it clearly holds. For positive integers, we write \( un = u(n-1) + u \) to get

\[
|\sin(un)| = |\sin(u(n-1)) \cos(u(n-1)) + \sin(u) \cos(u)|. 
\]

Using the triangle inequality and \( |\cos(x)| \leq 1 \), we get

\[
|\sin(un)| \leq |\sin(u(n-1))| + |\sin(u)|. 
\]

This recursive relation and the \( n = 0 \) case shows \( |\sin(un)| \leq |\sin(u)|n \) for positive integer \( n \). We use this result on the first factor of Eq. (49), setting

\[
u = \frac{\pi |k \cdot (2x + e)|}{L \|2x + e\|_1}, \\
n = \|2x + e\|_1. 
\]

(50)

This yields

\[
|\sin \left( \frac{\pi}{L} k \cdot (2x + e) \right)| \leq \sin \left( \frac{\pi |k \cdot (2x + e)|}{L \|2x + e\|_1} \right) \|2x + e\|_1 
\]

By applying the Hölder inequality, one finds \( |k \cdot (2x + e)| \leq \|k\|_\infty \|2x + e\|_1 \) where \( \|x\|_\infty = \max_j |x_j| \). Therefore, 

\[
u \leq \pi \|k\|_\infty / L. 
\]

(51)

Since \( k \in \Lambda, 0 \leq k_j \leq L \) and so one might infer \( \|k\|_\infty \leq L \).

However, due to the periodicity of the lattice we can switch to a co-ordinate system where \(-L/2 \leq k_j \leq L/2\), and so \( \|k\|_\infty \leq L/2 \). This entails \( 0 < v \leq \pi/2 \), which is important because \( \sin(u) \) is monotonic on this interval. Monotonicity allows us to upper bound \( \sin(u) \) by \( \sin(v) \) whenever \( u \leq v \leq \pi/2 \), and using \( v \) as the RHS of Eq. (51) we deduce

\[
|\sin \left( \frac{\pi}{L} k \cdot (2x + e) \right)| \leq \sin \left( \frac{\pi \|k\|_\infty / L}{\|2x + e\|_1} \right) \|2x + e\|_1. 
\]

(52)

Similarly the second \( \sin \) factor of Eq. (49) satisfies \( |\sin(\pi k \cdot e/L)| \leq |\sin(\pi \|k\|_\infty \|e\|_1/L)| \), and since \( e \) is a unit vector, \( |\sin(k \cdot e/L)| \leq |\sin(\pi \|k\|_\infty)| \). Combining these upper bounds with Eq. (49), we find

\[
A_k \leq \sin^2(\pi \|k\|_\infty / L) \|2x + e\|_1, 
\]

which completes our simplification of the numerator \( A_k \). Next we turn to the denominator \( B_k \), which is simply

\[
B_k = 1 - \lambda_k = \eta - \frac{\eta}{D} \sum_{j=1}^{D} \cos(2\pi k / L). 
\]

(53)

Clearly, \( 1 - 2 \sin^2(x) = \cos(2x) \) for all \( x \), so that

\[
B_k = \frac{2\eta}{D} \sum_{j=1}^{D} \sin^2(\pi k / L). 
\]

(54)

Bringing numerator and denominator together and rearranging we finally have

\[
\frac{A_k}{B_k} \leq \left( \frac{\sin^2(\pi \|k\|_\infty / L)}{\sum_j \sin^2(\pi |k_j| / L)} \right) \left( \frac{D \|2x + e\|_1}{2\eta} \right). 
\]

(55)

Lastly we note the first bracket is upper bounded by unity since the numerator is simply one of the positive terms in the denominator, and this proves Eq. (48). Putting this into Eq. (46) we arrive at

\[
|\Delta_l(x)| \leq \frac{D \|2x + e\|_1}{2\eta L D} \sum_{k \neq 0} |\lambda_k|^l. 
\]

(56)

We proceed by dividing the eigenvalues into shells of equal \( \|k\|_\infty \). We denote with \( O_{L,l} \) the number of elements in \( k \)-space shell for which \( \|k\|_\infty = l \) for \( l = 1, \ldots, (L-1)/2 \). Here, we have assumed for simplicity that \( L \) is odd, with even \( L \) admitting a similar treatment. Let us assume \( D \geq 2 \). Clearly, we have

\[
O_{L,l} = ((l+2)^D - (l+1)^D) \leq 2^D l^{D-1}. 
\]

(57)

We can now upper bound

\[
V_l := \sup \{|\lambda_k| : k \in O_{L,l}\} \\
= \lambda_{(l,0,\ldots,0)} \\
= (1 - \eta) + \frac{\eta}{D} \left( \cos(2\pi l / L) + D - 1 \right), 
\]

where we have again assumed \( \eta \leq 1/2 \) to ensure that \( |\lambda_k| = \lambda_k \). Making use of \( \cos(x) \leq 1 - x^2/4 \) for \( x \in [-\pi/2,\pi/2] \), one obtains

\[
V_l \leq 1 - \frac{\pi^2 \eta}{2D^2} \frac{l^2}{L^2}. 
\]

(59)

This gives

\[
\sum_{k \neq 0} |\lambda_k|^l \leq \sum_{l=1}^{(L-1)/2} O_{L,l} V_{l,l} \\
\leq 2^D \sum_{l=1}^{(L-1)/2} l^{D-1} \left( 1 - \frac{\pi^2 \eta}{2D^2} \frac{l^2}{L^2} \right)^t. 
\]

(60)

Plugging this into Eq. (56) and collecting constants, gives

\[
\Delta_l(x) = \frac{\alpha}{L} \sum_{x = \frac{1}{L}, \frac{1}{L^2}, \ldots, \frac{1}{L^D}} x^{D-1} (1 - \beta x^2)^l 
\]

(61)

where

\[
\alpha = \frac{D \|2x + e\|_1}{2\eta}, \\
\beta = \frac{\pi^2 \eta}{2D^2}, \\
x = 1/L. 
\]

(62)

(63)
if
\[ x^{D-1} (1 - \beta x^2)^t \leq \epsilon / \alpha, \tag{64} \]
for all \( x \in [0, 1/2] \). This is satisfied provided
\[ t \geq t_{\text{min}} := \max_{x \in [0,1/2]} \frac{\log (\epsilon x^{1-D} / \alpha)}{\log (1 - \beta x^2)}. \tag{65} \]
We can now use that
\[ -\log(1 - \beta x^2)^{-1} \leq \frac{1}{\beta x^2}, \tag{66} \]
for \( \beta x^2 \in [0, 1/2] \) (which can be inferred from \( \eta \leq 1/2, d \geq 2, \) and \( x \leq 1/2 \)), to bound
\[ t_{\text{min}} \leq \beta \max_{x \in [0,1]} \frac{\log (\alpha x^{D-1} / \epsilon)}{x^2}. \tag{67} \]
The extremum of
\[ f(y) = \log(\alpha y^{D-1}) / y^2 \tag{68} \]
can be found by differentiation to be at \( y_{\text{max}} = e^{1/2} a^{-1/(D-1)} \) taking the functional value \( f(y_{\text{max}}) = (D - 1)a^{2/(D-1)}/2\epsilon \).

In Eq. (67) we have this form with \( a = \alpha / \epsilon \) and so
\[ t_{\text{min}} \leq \frac{\beta(D - 1)}{2\epsilon} \left( \frac{\alpha}{\epsilon} \right)^{\frac{2}{D-1}}. \tag{69} \]
Inserting the values of \( \alpha \) and \( \beta \) we get
\[ t_{\text{min}} \leq \chi(x) \epsilon^{\frac{2}{D-1}} \tag{70} \]
where
\[ \chi(x) = \frac{\pi^2 \eta(D - 1)}{2\epsilon D^2} \left( \frac{D \|2x + e\|_1^2}{2\eta} \right)^{\frac{2}{D-1}}. \tag{71} \]
This provides Eq. (12) with an explicit value of \( \chi(x) \). Setting \( x = 0 \) and \( \eta = 1/2 \), we find \( \chi(0) \leq 15 \) and \( \chi(0) \leq 5 \) for dimensions \( D = 2, 3 \), respectively. For general \( x \), we can obtain a neater upper bound on \( \chi(x) \), so that
\[ \chi'(x) \leq \begin{cases} 8(2\|x\|_1 + 1)^2 / \eta & \text{for } D = 2, \\ 5(2\|x\|_1 + 1) & \text{for } D = 3. \end{cases} \tag{72} \]
This be achieved using the triangle inequality, so \( 2\|x + e\|_1 \leq \|2x + e\|_1 + \|e\|_1 = 2\|x\|_1 + 1 \), and either \( \pi^2 / (8\epsilon) < 1/2 \) (used for \( D = 2 \)) or \( 4\pi^2 / (3\epsilon) < 5 \) (used for \( D = 3 \)).