

ARTICLE OPEN

A local and scalable lattice renormalization method for ballistic quantum computation

Daniel Herr^{1,2}, Alexandru Paler³, Simon J. Devitt^{4,5} and Franco Nori^{1,6}

A recent proposal has shown that it is possible to perform linear-optics quantum computation using a ballistic generation of the lattice. Yet, due to the probabilistic generation of its cluster state, it is not possible to use the fault-tolerant Raussendorf lattice, which requires a lower failure rate during the entanglement-generation process. Previous work in this area showed proof-of-principle linear-optics quantum computation, while this paper presents an approach to it which is more practical, satisfying several key constraints. We develop a classical measurement scheme that purifies a large faulty lattice to a smaller lattice with entanglement faults below threshold. A single application of this method can reduce the entanglement error rate to 7% for an input failure rate of 25%. Thus, we can show that it is possible to achieve fault tolerance for ballistic methods.

npj Quantum Information (2018)4:27; doi:10.1038/s41534-018-0076-0

INTRODUCTION

Several physical platforms are aiming at achieving quantum computing.¹ For example, a qubit can be implemented using superconductors,² silicon,^{3,4} trapped ion systems,^{5,6} or using linear optics.^{7,8} Major advances in the fidelity of these qubits have made the application of error-correction codes, such as the surface code,⁹ feasible. The surface code is of particular interest due to its high threshold⁹ and 2D nearest-neighbor layout. While the surface code is suitable for qubit implementations which are relatively easy to control, linear-optics quantum computation¹⁰ is based on a slightly different principle, where photons are entangled in a cluster state which is then consumed during the computation. This quantum one-way computer was proposed by Raussendorf et al.^{11,12} A high-level implementation for such a quantum computer can be divided into three steps:^{13,14}

1. Photon sources: delivers GHZ-triplets.
2. Entangling layer: generates the cluster state.
3. Measurements: measurements in different bases allow for universal computation.

While cluster states have also been studied for solid-state qubits,^{15–17} this scheme is better suited for photonics because it prioritizes measurements over the sequential application of quantum gates, and thus utilizes the ability to generate photons continuously.

The original proposal^{11,12} was not protected against errors but a similar approach can introduce fault tolerance.¹⁸ This approach uses the Raussendorf lattice as an underlying resource which protects the logical state both against noise and photon loss.¹⁹

There are several approaches for creating a cluster state for quantum one-way computation.¹⁹ A drawback of linear optics is that the process of creating entanglement is non-deterministic. Thus, there remains a non-zero probability that each

entanglement operation fails and the resulting lattice misses edges. Some approaches like²⁰ try to remedy the probabilistic nature by adding redundancy to the entangling procedures. However, these approaches require many switches which rely on the outcome of previous entangling operations and add more noise to the system. Another approach is to just use these non-deterministic gates and generate a faulty lattice. This ballistic approach⁸ of linear-optics quantum computation recently gained attention due to improved theoretical entangling operations^{21,22} which fail with 25% probability. In ref.²³, the percolation threshold for the Raussendorf lattice has been calculated to be 37.3%. Thus, a bond-failure rate of 25% is low enough for the lattice to percolate and therefore information can be transported from one end to the other, given a large enough faulty lattice.²⁴

In order to build a large-scale quantum computer, this ballistic approach to linear-optics quantum computing should generate the Raussendorf lattice which then allows for fault-tolerant computation. Classical control software must now be developed in order to cope with 25% of faulty entanglement operations, while still retaining these error-correction capabilities. In this paper, we provide an example of such an algorithm which is based on ref.²⁴ and that acts as a preprocessing step. One should note that the approach in ref.²⁴ has two major limitations: (i) only in 1D; (ii) it is not fault-tolerant. Our approach has neither of these drawbacks when it is combined with the usual Raussendorf lattice error-correction schemes.^{25–27} It should be noted that the preprocessing is not inherently fault-tolerant which results in a trade-off between the rate of missing bonds and an accumulation of errors due to imperfect measurements. The accumulation of these errors, however, only shifts the threshold of the Raussendorf lattice and can be remedied by higher fidelities of the experimental setup.

¹Theoretical Quantum Physics Laboratory, RIKEN Cluster for Pioneering Research, Wako-Shi, Saitama 351-0198, Japan; ²Computational Physics, ETH Zurich, 8093 Zurich, Switzerland; ³Linz Institute of Technology, Johannes Kepler University, 4040 Linz, Austria; ⁴Centre for Quantum Software & Information (QSI), Faculty of Engineering & Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia; ⁵Turing Inc., Berkeley, CA 94705, USA and ⁶Department of Physics, University of Michigan, Ann Arbor, MI 48109-1040, USA

Correspondence: Daniel Herr (daniel.herr@riken.jp)

Received: 10 November 2017 Revised: 5 May 2018 Accepted: 16 May 2018

Published online: 21 June 2018

In the following we will review relevant concepts. We will give the definition of graph states and explain how they can be modified. Then we will review the Raussendorf lattice, its creation, and its error-correction capabilities with a focus on faulty edges.

Graph states

Graph states²⁸ are a generalization of cluster states²⁹ and can be described using a undirected graph $G = (V, E)$ with vertices V and edges E . Each vertex corresponds to a physical qubit initialized in the $|+\rangle$ -state. On each edge, a controlled-phase gate is applied. This results in the final state of:

$$|\psi\rangle_{\text{Graph}} = \prod_{(ij) \in E} CZ_{ij}|+\rangle^{\otimes |V|}. \quad (1)$$

Using measurements, the graph can be modified to another graph. The modification rules have been discussed in ref. ²⁸. Our proposal will only rely on two particular easy measurement operations:

1. Z-measurement on qubit a : Remove $\{a\}$ from the graph and break all connections it was involved in.
2. Y-measurement on qubit a : Invert the neighborhood of a and remove a from the graph. This method can be used to generate long-distance edges.

Here, the neighborhood of a node is defined by $\text{Nbh}(a) = \{x \in V \mid (x, a) \in E\}$ and an inversion over a set of nodes N is defined by the graph $G = (N, \{(a, b) \in N^2 \mid (a, b) \notin E\})$. An example of these two rules on a square lattice is shown in Fig. 1.

Creation of the lattice

A special graph state with error-correction capabilities is given by the Raussendorf lattice.¹⁸ It is a 3D lattice whose unit cell is shown in Fig. 2a.

To create the Raussendorf lattice in a ballistic way, GHZ states are needed as a resource. Three of these GHZ states can be entangled to a five-node micro-cluster, using two probabilistic fusion gates.³⁰ There are two ways to apply these fusion gates, with different additional resources. The fusion gate given in ref. ²¹ requires an additional pair of maximally entangled photons, whereas ref. ²² requires four single photons. The creation of the micro-clusters follows³⁰ and all possible outcomes of the micro-cluster generation are shown in Fig. 3.

The central node of each micro-cluster will correspond to a node in the final lattice, while its surrounding nodes are consumed in additional fusion operations to connect clusters with each other. It can be seen in Fig. 3 that a failure during the

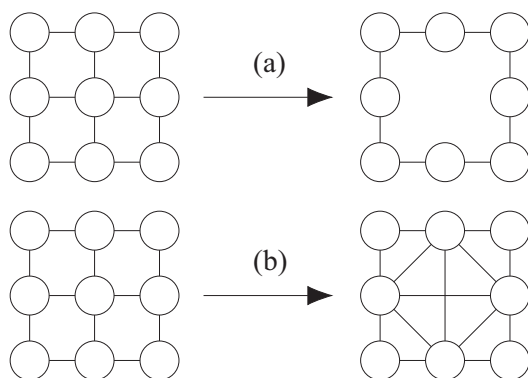


Fig. 1 Graph modification rules under measurement. This figure shows how a regular lattice changes under a Z-basis **a** or a Y-basis **b** measurement. These are the only measurement operations that our purification procedure will need

generation of these micro-clusters results in non-local entanglement. However, it becomes exponentially unlikely for edges with larger distances.

After the creation of the micro-clusters, each of these needs to be entangled to its neighbors on the large lattice. This is where our proposal deviates from,³⁰ since the underlying lattice we try to implement is the Raussendorf lattice and not the diamond lattice from the original proposal.

In Fig. 2b the generation of this lattice is shown. Each micro-cluster will correspond to a single node after all fusion gates have been performed. The fusion of these micro-clusters happens with a probability of 75%. Thus, 25% of the time fusion operations that generate this lattice fail.

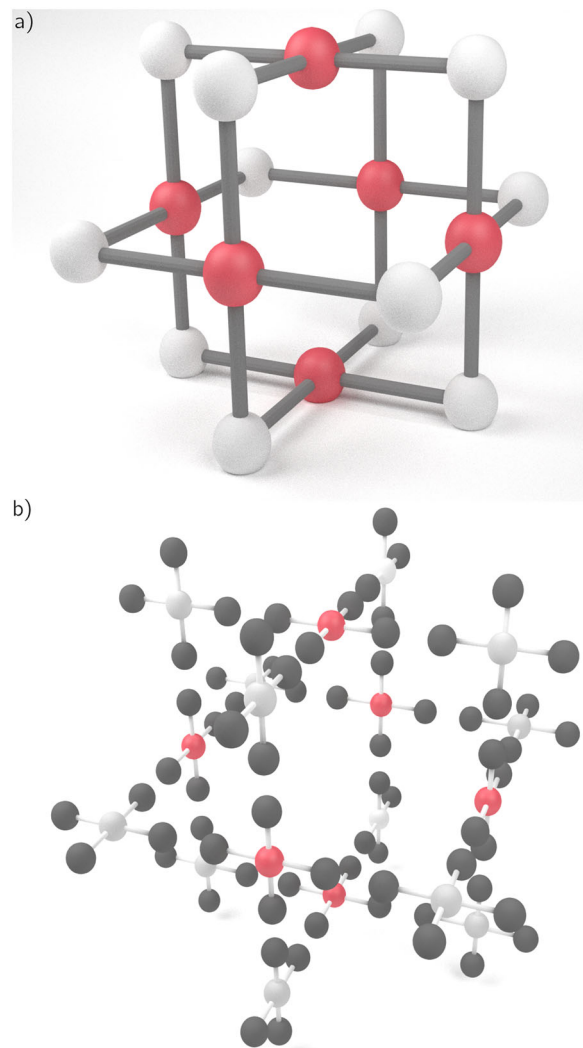


Fig. 2 Raussendorf unit cell and its construction. The Raussendorf unit cell is shown in part **a**. The spheres represent individual photons and the connections between them represent entanglement given by the definition of a graph state. The photons which are colored in red contribute to a single X-parity check. Whereas the white spheres correspond to the faces of the dual lattice, which make up Z-parity check operations. Part **b** shows how the unit cell is composed of micro-clusters. All neighboring pairs of dark nodes are consumed during the application of the fusion operations. The remaining white and red nodes correspond to the same color as in part **a**. For simplicity, only successful micro-clusters are shown here. See Fig. 3 to see all possible micro-cluster shapes

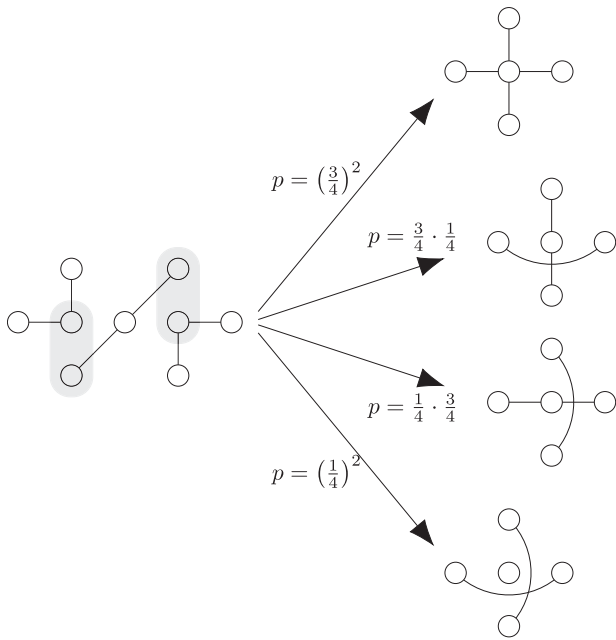


Fig. 3 The creation of micro-clusters as described in ref. ³⁰ The areas shaded in gray indicate which two qubits are used for the fusion gates. Depending on the measurement outcome of these fusion gates, the structure will take one of the shapes on the right

Error correction

Error detection and correction can be done using parity checks on particular nodes on the purified Raussendorf lattice. As an example, the total parity in the X -basis of the qubits colored in red (Fig. 2a) is conserved unless there has been an error. A self-similar lattice which is shifted by half a unit-cell uses the photons shown in white to perform Z -parity checks.

These parity checks together enable the error-correction capabilities of the Raussendorf lattice. Furthermore, with the method of ref. ²⁶ the Raussendorf lattice is also protected against photon losses. The main idea in this approach is to use the linearity of the parity checks to form super cells and perform parity checks on these. This resulted in a trade-off between the error rate due to perturbations and the rate of photon loss. The best photon loss rates that could still be corrected were around 25%.²⁶

A lattice with faulty edges can be translated to a lattice with missing nodes, by deliberately losing one of the photons at the end of a faulty edge. This is done by performing a measurement in Z -basis on one of these photons. A recent paper²⁷ described this as an adaptive correction scheme where the measurement basis needs to be changed depending on the error. This adaptive scheme can tolerate a loss rate of 14.5% of all edges. Another approach is to keep measuring as usual and then in the classical tracking software treat both qubits that are involved in the faulty connection as lost photons. There, still correctable loss rates lie at around 6.5%. Unfortunately, neither approach can correct for bond failure rates of 25%, and thus preprocessing in some form has to be performed.

RESULTS

The general idea of our graph purification proposal is to develop a measurement scheme that translates a large Raussendorf lattice with many faults into a smaller Raussendorf lattice with fewer faults. Our procedure is based on ref. ²⁴ which investigated how path-finding procedures can help for quantum computation on a faulty lattice. It is not a quantum error correcting code, such that errors will accumulate during this step. Nevertheless, after this

preprocessing, the original lattice has been translated to a lattice with fewer faults such that a general error-correction procedure can be used.

The main requirements for such an algorithm are

1. The algorithm should be local, i.e., the algorithm's corrections should only rely on faults in the vicinity of the lattice. This is important since the lattice is generated continuously and only a part of it is physically available at any time.
2. The algorithm should give the corrections fast. This is important because photons are fast and delays in computation translate to large sizes of the quantum computer with long optical fibers.
3. The algorithm should require as little overhead as possible in terms of photons.
4. Scalability: adding more photons should be possible (e.g., the algorithm should be parallelizable).

Here, we will briefly summarize the main idea behind the algorithm. A more detailed description is given in the Methods section.

This proposed algorithm is based on the idea that while a 25% fusion failure rate is very high the lattice still percolates, and information can be transported from one end to the other. For a larger lattice, the probability to find paths from one node to another increases. This means that a subset of nodes from the faulty lattice can be chosen, such that there exist paths between the nodes which will generate a purified Raussendorf lattice. All photons that are not part of these paths or not part of the subset of chosen nodes are measured in the Z -basis. Therefore, all these qubits are removed from the lattice. All photons that belong to the found paths have to be measured in the Y -basis. This will also remove these qubits from the lattice, but edges between the subset of nodes still exist. With a proper choice of nodes and paths, a purified Raussendorf lattice will then be obtained.

To find individual nodes that make up the purified lattice, we first divide the faulty lattice into finite-size boxes. These boxes will contain at most one node of the purified lattice, and paths have to be found to nodes in neighboring boxes. This somewhat artificial division into boxes is only done for the performance of our algorithm and the underlying lattice is not changed during this step. Thus, non-local interactions are not broken up and have no influence on the algorithm with respect to its finite-sized boxes. The partition into boxes gives a defining variable for our algorithm: the edge size of such boxes. Due to percolation, larger box sizes translate to higher success rates to find nodes and paths between these nodes. This also implies that more nodes are needed from the faulty lattice to obtain a single node in the purified lattice.

In order to see how well our code performs we ran this algorithm on lattices created with different success probabilities for the fusion gates, using different box sizes, and compared the rate of faults on the purified lattice to the initial failure rate. The behavior of the output bond error rate with increasing size is plotted in Fig. 4a. One can see that for an initial failure rate of 25% it is possible to reach an output error rate of about 7% for the purified lattice. This is below the threshold rate of 14.5%²⁷ of the Raussendorf lattice. Thus, it should be possible to use this code as a preprocessor for fault-tolerant ballistic quantum computation.

In Fig. 4b the relation between fusion failure rate and output error rate is shown. Every data point below the black curve shows an improvement over the input error rate. Thus, it makes sense to use this algorithm for fusion failure rates below 32%.

Figure 5 shows a histogram of the length distribution of the paths. The average path length is larger than the box size because the shortest possible path is not always possible due to missing edges on the graph. It is possible to obtain shorter paths due to non-local interactions and differences in structure positions. The average path length for a box size of 20 is given by 28.42 ± 0.05 .

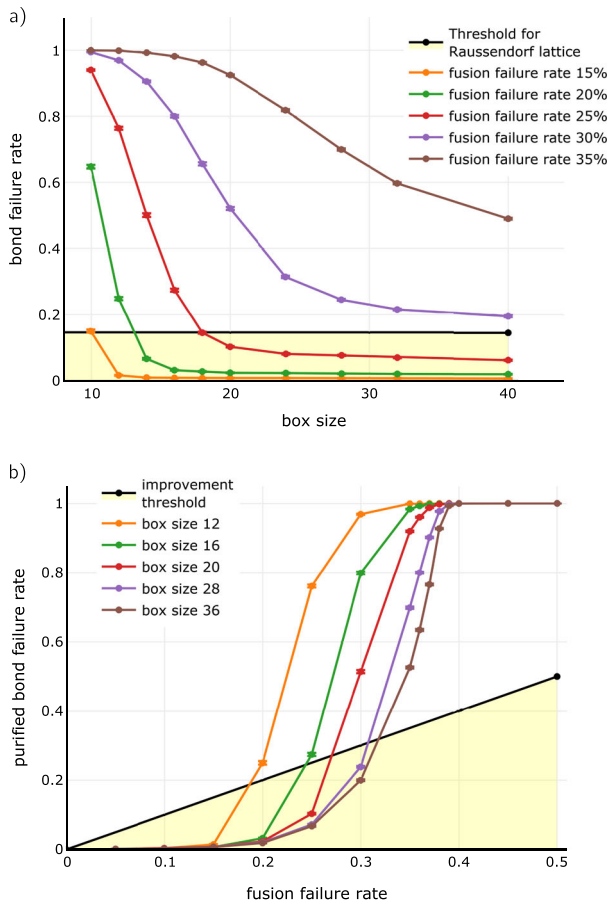


Fig. 4 Bond failure rate of the purified lattice. The bond failure rate as a function of the box size is plotted in part **a**. The black horizontal line indicates the threshold where the Raussendorf lattice can correct for missing bonds. For a fusion failure rate of 25% the purified lattice for box sizes above 18 are below the threshold of the Raussendorf lattice. Part **b** shows the bond failure rate of the purified lattice as a function of the fusion failure rate. The black line shows the border below which this purification algorithm can decrease the failure rate. Thus, it only makes sense to use this algorithm below an input fusion failure rate of around 32% if box sizes up to 36 are used

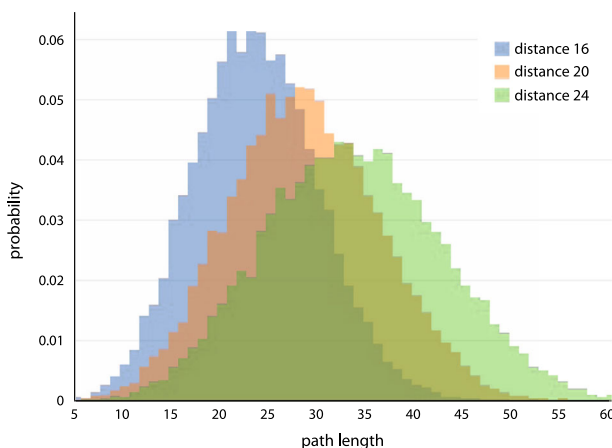


Fig. 5 Length distribution for all found paths. The mean lies at 23.48 ± 0.04 for box size 16, at 28.42 ± 0.05 for box size 20, and at 33.54 ± 0.06 for box size 24. All probabilities were calculated with at least 18,000 different paths and using a constant fusion failure rate of 25%

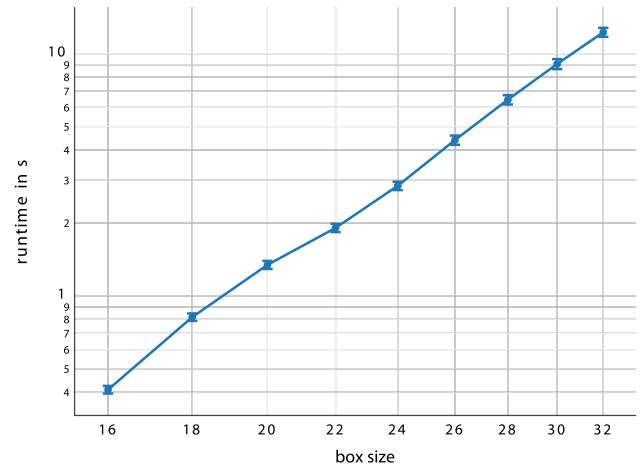


Fig. 6 Execution time. This figure shows the absolute time that the algorithm needs to find structures and paths (the generation of the lattice was not measured). The plot has been performed for a constant number of $5 \times 5 \times 3$ boxes and at a constant 25% error rate for the fusion gates. From the slope of this log-log plot one can see that the algorithm has polynomial scaling with a leading power of 4.83 ± 0.06

We will use these values in the following analysis to estimate the effects of errors.

We performed a simple timing analysis by running the algorithm on a single core of an i7-4558U (2.8 GHz) CPU, to give a rough estimate on the speed of the algorithm. The results are plotted in Fig. 6. For a box size of 20 and a $5 \times 5 \times 3$ lattice of boxes, the algorithm needs on average 1.34 ± 0.05 s. However, it should be noted that not much effort was put into optimization and better performance can be expected from optimized implementations. The scaling of this algorithm is polynomially both in box size and number of boxes.

DISCUSSION

The purification process is not inherently fault-tolerant, so errors can accumulate along the paths between nodes of the purified lattice. Fault tolerance is introduced afterwards, on the purified Raussendorf lattice. In the following, we want to estimate the effects of induced Pauli errors due to measurement errors. This will lead to an adjusted error rate for the purified Raussendorf lattice. The main purpose of this discussion is to show that fault tolerance can be maintained, although with stricter requirements on the hardware, even if the purification procedure results in additional noise that accumulates with the length of the paths.

To analyze the sources of error it makes sense to discuss the measurement procedure. A qubit state is encoded as a spatial mode, which can be measured using a pair of photon detectors. There are two ways of how a Hadamard operation can be implemented to enable a change in the measurement basis (Fig. 7). Because our proposal requires to change the measurement basis depending on the lattice that has been created, the ability to add a Hadamard operator in reasonably short time has to be guaranteed. A simple Hadamard operation can be implemented by bringing together the two wave guides for a length of $\pi/2$. In order to add a choice of measurement basis, one can use switches, as shown in the first approach of Fig. 7. The other approach divides the Hadamard operation into two by bringing together both wave guides for a length of $\pi/4$ both times and adds a gate that creates a phase-difference between the two wave guides. If this phase difference is zero, the Hadamard operation is performed, but if the phase-difference is π one will obtain the identity operation.

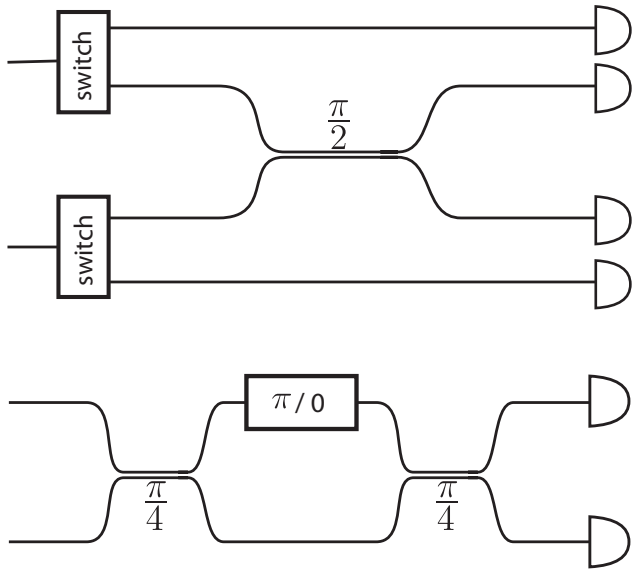


Fig. 7 Measurement procedures. Two possible procedures for the measurements³¹

There are several sources of errors that can occur in such setups. One source of errors comes from imperfections due to switches which will lead to photon loss. Another source of errors are imperfect rotations and Hadamard operations. For linear-optics applications the implementation of Hadamard operations is precise^{31,32} and using the second approach of Fig. 7, switches are not needed during measurements. For a proper threshold calculation, these are still important sources of errors, but in our discussion we will neglect both photon loss and imperfect rotations.

We will only focus on a third source of errors which happens at the detectors. These errors result in false positives (the detector detects a non-existent photon) as well as false negatives (a photon is not registered at the detector). Henceforth, we will refer to the rate of successful detection events as the fidelity of the photon detectors.

All first-order errors at the detectors result in a non-sensical measurement: either both photon detectors are triggered at the same time or neither of the photon detectors is triggered. This corresponds to either a false positive or a false negative at one of the detectors. If such a case happens, it is clear that an error occurred, but it is impossible to know the nature of the error. A second-order error will result in the opposite measurement outcome. The wrong measurement will result in a Pauli- X error for a Z -basis measurement or a Pauli- Z error for a X -basis measurement.

One possible correction scheme for first-order detector errors is to choose a measurement outcome randomly. With 50% this choice is incorrect. A single measurement acts as a teleportation operation. If a wrong measurement outcome is observed, a wrong by-product Pauli operation is specified due to the rules of measurement-based quantum computing. In the end, additional Pauli errors appear on the purified lattice and the Raussendorf lattice has to locate and identify them.

The total probability for at least one Pauli-error on an average path can be calculated using:

$$P_{\text{err}} = 1 - f^{2\bar{L}}. \quad (2)$$

For a box size 20, the mean length of a path is $\bar{L} = 29$ and given a detector fidelity $f = 0.9999$, the resulting error rate is $P_{\text{err}} = 0.57\%$ for each bond. The factor 2 in the equation comes from the fact that each measurement involves two photon detectors.

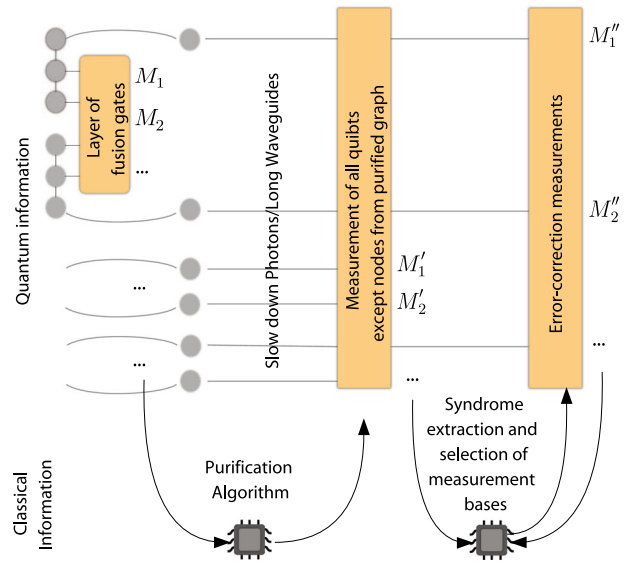


Fig. 8 Flow of information. This figure shows the quantum and classical flow of information. Starting from GHZ-states, some qubits will be measured in a layer of fusion gates and the measurement information M_i is fed to our proposed algorithm. Meanwhile, the remaining photons are led along a long wave guide to give enough time to the classical computation. These qubits are then fed into a layer of measurements, where all photons except the ones from the purified lattice are measured out. The measurements M'_i are used to determine if any errors happened. This information is then sent to the Raussendorf error-correction processor that takes care of the syndrome extractions and measurements in proper bases

To obtain the probability for an induced Pauli error per purified node, we assume that if a detector error occurs we attribute it to the node in the same box. For a single node there are on average $4(\bar{L}/2)$ qubits for all four paths. Using this in the exponent the resulting probability of Pauli errors on each node is $P_{\text{NodeErr}} = 1.15\%$. However, this error rate can be halved when randomly applying one of two correctional gates for first-order errors. The effective probability for induced Pauli errors per node is $P_{\text{NodeErr}} \approx 0.58\%$.

For a box size 20, the bond-loss probability is 10%. In ref. ²⁷ it has been shown that, to still remain below threshold, the remaining measurement errors need to be below 0.6%. Thus, the above fidelity for the detectors is low enough to have errors below threshold for the Raussendorf lattice. If this threshold ($p_{\text{th}} \approx 0.01\%$) is compared to the threshold error rate of $p_{\text{th}} \approx 2.9\%$ on the Raussendorf lattice without bond failures,³³ one can see that our method constitutes a trade-off between the correction for bond errors and induced Pauli errors. It should be noted that our comparison to the measurement error rate of $p_{\text{th}} \approx 2.9\%$ is justified because Pauli errors are induced after the creation of the lattice and thus do not spread.

Larger box sizes provide better stability under bond failures, at the cost of a decreased threshold. This trade-off is currently associated with a large overhead, such that our required fidelity for photon detectors poses a strong requirement on experiments. However, we expect improvements with advances in our preprocessing algorithm which will relax the requirements for the experimental setup.

Workflow

In the introduction we mentioned a high-level design of a linear-optics quantum computer. Here, we want to refine on it, with the inclusion of our purification step. To this end, we show in Fig. 8 a possible quantum and classical flow of information and the

actions that are taken due to that information. To the left of the figure three-qubit GHZ-states are created and used by fusion gates. It should be noted that additional GHZ states and photons are needed in that step but for simplicity they are not shown here. Using the measurement results of the fusion gates, our proposed algorithm will find a smaller graph, using only a subset of nodes. These nodes are routed through to the final layer of measurements, which requires a switch each. The last measurements are the actual measurements needed for the purified Raussendorf lattice, where syndrome extraction and the actual fault-tolerant quantum computation happen.

CONCLUSION

The algorithm's performance seems to depend heavily on the details of the implementation. We already used several heuristics that improve the probabilities of success for the first steps of the path-finding algorithm but more advanced heuristics might improve the error rate of the purified lattice even further. Furthermore, a clustering algorithm should help in choosing good node positions at the cost of an increased runtime. The effects of this should be included in future analysis. Changes to the distance heuristic for the A*-search might also affect the performance of this proposal, but it was not investigated here.

In conclusion, we have presented a way to purify the 3D lattice obtained from the ballistic procedure proposed in ref. ³⁰ using ideas from ref. ²⁴. This purification process can suppress entanglement errors due to probabilistic fusion gates and bring the error rate from 25% down below the Raussendorf lattice threshold. This procedure, however, has the cost that errors along the generated paths can accumulate and requires higher precision in their measurement operations. Nevertheless, this approach shows that fault-tolerant quantum computation using ballistic lattice generation is possible. Looking back at the requirements we posed, we can see that our proposal fulfills several of them:

1. The algorithm is local. Due to the exponential decay in large distance edges, all connected nodes are located in the same box or neighboring boxes.
2. The algorithm scales polynomially in lattice size, but our implementation should still be improved in terms of absolute speed.
3. The overhead in terms of qubits could be better: each box consists of about 20^3 nodes, which are all consumed to generate one node in the purified lattice. Errors also accumulate, with larger sizes.
4. The algorithm is easily scalable, with only little communication required by different processes (see Methods).

While our code works, many improvements can be made to this preprocessing step, such as using different measurement schemes to create entanglement with X-basis measurements. Thus, it is likely that the output bond failure rate and therefore resource requirements are further reduced. Furthermore, it should be investigated how a more realistic error model that includes both photon loss and imperfect rotations affect the overall performance of the error correction scheme. Afterwards, fair comparisons between different ways to generate the lattice²⁰ and the ballistic approach with preprocessing should be made in terms of overhead for the Raussendorf lattice.

METHODS

In this section we describe in more detail how our implementation to purify the Raussendorf lattice works. A C++ implementation to this description is open-source and hosted on Github (https://github.com/herr-d/photonic_lattice). In the Supplementary we give a short technical overview of the code.

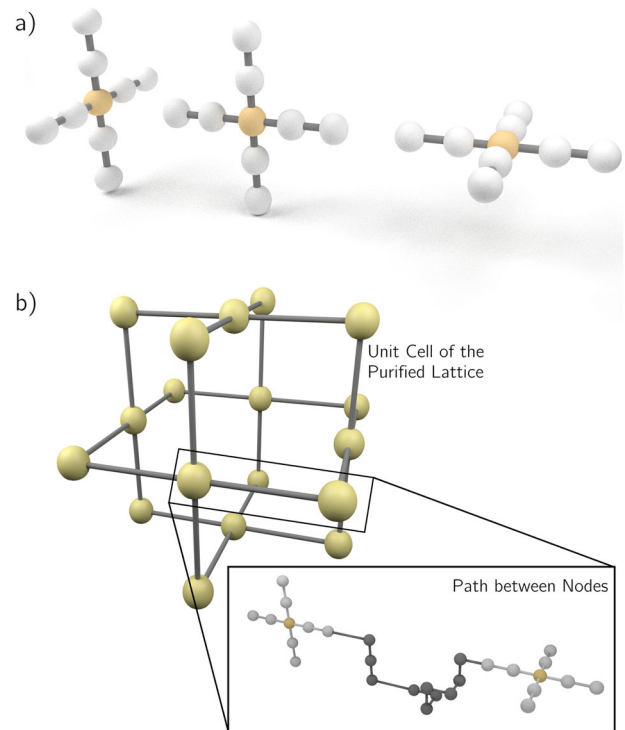


Fig. 9 Creation of a purified unit cell. In part **a** all possible orientations for the structure are shown. The node in the center will be used as the node for the purified lattice. Part **b** shows a unit cell in a $3 \times 3 \times 3$ square lattice. For illustrative purposes we show in the inset how one of the edges were obtained by finding a path from one structure to its neighbor. All qubits not shown need to be measured in Z-basis to remove them from the graph. It should be noted that this is a constructed example and a lattice with realistic failure probabilities requires larger sizes

The implementation divides the original Raussendorf lattice into boxes. Each box has the same number of qubits along its edges. This partition results in a square lattice, but the unit cell of the Raussendorf lattice can be easily embedded in a $3 \times 3 \times 3$ square lattice by removing half of the boxes. In each remaining box, exactly one of the structures from Fig. 9a is placed. Which orientation is needed depends on which location the box is in the purified lattice. In Fig. 9b one can see an example of the proper orientations deduced from the purified lattice. The structure position is then chosen by looping over all nodes inside the structure and seeing if the connections exist for this structure. If several positions are found, the position with the most neighbors is chosen. This helps to increase the success rate of the path-finding algorithm by avoiding immediate dead ends.

Each structure has four nodes that are farthest from the center. We call these the handles of the structure. They are the start or end points for the path-finding algorithm. For our implementation, the handles in the right, up, and back directions will be the starting points for individual path-finding runs. Whereas the handles in the left, down and front directions of neighboring structures will be the end points. A single path-finding instance, thus, only needs to have knowledge about two neighboring boxes: the one with the starting handle and the one with the goal node. Therefore, the algorithm is local.

The paths are used to connect different structures with each other and the nodes that are part of a path are intended to be measured in the Y-basis. In order to create a single edge between two structures, a further requirement for the paths arises. At no time must two of them cross or even be neighbors (there must be at least one node between paths that does not belong to a path). This means that after all other qubits are measured in the Z-basis, the qubits of a path only have two neighbors, both part of the same path or part of a structure.

Besides the previous requirements, our proposed algorithm does not require a specific implementation for finding such paths. For example Dijkstra's algorithm³⁴ can be employed, or an A*-implementation³⁵ can be

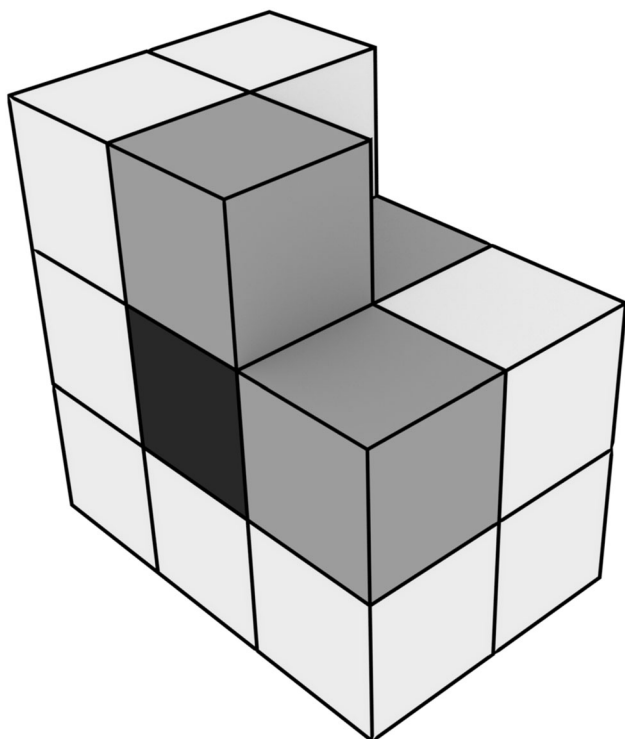


Fig. 10 Parallelization of the algorithm. The process responsible for the calculation of the black box only requires information about the three nearest-neighbor gray boxes. One of these boxes is in temporal direction and, thus, only information between two neighboring processes needs to be exchanged

used. One does not need to find shortest paths, but it helps with respect to the accumulation of errors.

Our implementation uses the A*-algorithm³⁵ with the Manhattan-metric as its heuristic function. This heuristic should give a decent estimate on the remaining distance but this estimate is not strictly smaller than the actual distance because of non-local entanglement due to the fusion gates. Therefore, this A*-implementation is not guaranteed to find the shortest path.

The result is shown in Fig. 9b, where the purified lattice is visualized. The inset shows one of the paths between two neighboring structures. All qubits that are not part of either a structure or a path are measured in the Z-basis and thus completely removed from the graph. All remaining qubits, except the ones marked in yellow, need to be measured in the Y-basis, and thus contribute to edges in the purified lattice.

While our current implementation is not yet parallelized, it should be possible to parallelize it with only a few changes. Each processor could have its own set of boxes. Only information for boxes on the surface need to be exchanged with other processors. For the most parallel implementation, each box has its own processor. This is shown in Fig. 10, where the process responsible for the black boxes needs information only about two of the boxes colored in gray.

Every process needs to find a structure position in each of its boxes. Each process needs to send the position of its qubit structures which lie on the boundary surface to the process on the left, and down (opposite direction). The box in the back is treated by the same process so no communication is required. After every box receives the information from its two neighbors, it can continue to find two paths in two of the other directions: right, up, and back. The overhead of communication scales with the surface and not the volume and each process only needs to know a small part of the whole lattice, such that memory problems can be avoided.

Thus, a parallelized version has the same scaling in terms of box size as the sequential implementation, because each processor has a constant number of boxes. When more processors are added for a bigger faulty lattice, the total execution time will remain constant if the number of boxes per processor stays constant.

Data availability

All relevant data are accessible on Github (https://github.com/herr-d/photonic_lattice).

ACKNOWLEDGEMENTS

D.H. is supported by the RIKEN IPA program and A.P. is supported by the Linz Institute of Technology project CHARON. F.N. is partially supported by the MURI Center for Dynamic Magneto-Optics via the AFOSR Award No. FA9550-14-1-0040, the Army Research Office (ARO) under grant number 73315PH, the AOARD grant No. FA2386-18-1-4045, the CREST grant no. JPMJCR1676, the IMPACT program of JST, the RIKEN-AIST Challenge Research Fund, the JSPS-RFBR grant No. 17-52-50023, and the Sir John Templeton Foundation.

AUTHOR CONTRIBUTIONS

All authors (D.H., A.P., S.D., and F.N.) contributed to the inception of the algorithm. D. H. and A.P. wrote the code and generated the data. S.D. checked the results, and all authors helped writing the manuscript.

ADDITIONAL INFORMATION

Supplementary Information accompanies the paper on the *npj Quantum Information* website (<https://doi.org/10.1038/s41534-018-0076-0>).

Competing interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

REFERENCES

- Buluta, I., Ashhab, S. & Nori, F. Natural and artificial atoms for quantum computation. *Rep. Progress. Phys.* **74**, 104401 (2011).
- Barends, R. et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **508**, 500–503 (2014).
- Veldhorst, M. et al. An addressable quantum dot qubit with fault-tolerant control fidelity. *Nat. Nanotechnol.* **9**, 981 (2014).
- Takeda, K. et al. A fault-tolerant addressable spin qubit in a natural silicon quantum dot. *Sci. Adv.* **2**, e1600694 (2016).
- Ballance, C. J., Harty, T. P., Linke, N. M., Sepiol, M. A. & Lucas, D. M. High-fidelity quantum logic gates using trapped-ion hyperfine qubits. *Phys. Rev. Lett.* **117**, 060504 (2016).
- Linke, N. M. et al. Fault-tolerant quantum error detection. *Science Advances* Vol. 3, no. 10, e1701074 (20 Oct 2017) (2016).
- Knill, E., Laflamme, R. & Milburn, G. J. A scheme for efficient quantum computation with linear optics. *Nature* **409**, 46–52 (2001).
- Kieling, K., Rudolph, T. & Eisert, J. Percolation, renormalization, and quantum computing with nondeterministic gates. *Phys. Rev. Lett.* **99**, 130501 (2007).
- Fowler, A. G., Mariantoni, M., Martinis, J. M. & Cleland, A. N. Surface codes: towards practical large-scale quantum computation. *Phys. Rev. A* **86**, 032324 (2012).
- Nielsen, M. A. Optical quantum computation using cluster states. *Phys. Rev. Lett.* **93**, 040503 (2004).
- Raussendorf, R. & Briegel, H. J. A one-way quantum computer. *Phys. Rev. Lett.* **86**, 5188–5191 (2001).
- Raussendorf, R., Browne, D. E. & Briegel, H. Measurement-based quantum computation with cluster states. *Phys. Rev. A* **68**, 022312 (2003).
- Devitt, S. J. et al. Architectural design for a topological cluster state quantum computer. *New J. Phys.* **11**, 083032 (2009).
- Devitt, S. J., Fowler, A. G., Tilma, T., Munro, W. J. & Nemoto, K. Classical processing requirements for a topological quantum computing system. *Int. J. Quantum Inf.* **08**, 121–147 (2010).
- Tanamoto, T., Liu, Y., Fujita, S., Hu, X. & Nori, F. Producing cluster states in charge qubits and flux qubits. *Phys. Rev. Lett.* **97**, 230501 (2006).
- You, J. Q., Wang, X., Tanamoto, T. & Nori, F. Efficient one-step generation of large cluster states with solid-state circuits. *Phys. Rev. A* **75**, 052319 (2007).
- Tanamoto, T., Liu, Y., Hu, X. & Nori, F. Efficient quantum circuits for one-way quantum computing. *Phys. Rev. Lett.* **102**, 100501 (2009).
- Raussendorf, R., Harrington, J. & Goyal, K. A fault-tolerant one-way quantum computer. *Ann. Phys.* **321**, 2242–2270 (2006).
- Briegel, H. J., Browne, D. E., Dür, W., Raussendorf, R. & Van den Nest, M. Measurement-based quantum computation. *Nat. Phys.* **5**, 19–26 (2009).
- Li, Y., Humphreys, P. C., Mendoza, G. J. & Benjamin, S. C. Resource costs for fault-tolerant linear optical quantum computing. *Phys. Rev. X* **5**, 041007 (2015).

21. Grice, W. P. Arbitrarily complete Bell-state measurement using only linear optical elements. *Phys. Rev. A* **84**, 042331 (2011).
22. Ewert, F. & van Loock, P. 3/4-efficient Bell measurement with passive linear optics and unentangled ancillae. *Phys. Rev. Lett.* **113**, 140403 (2014).
23. Pant, M., Towsley, D., Englund, D. & Guha, S. Percolation thresholds for photonic quantum computing (2017).
24. Morley-Short, S. et al. Physical-depth architectural requirements for generating universal photonic cluster states. *Quantum Sci. Technol.* **3** 015005 (2018).
25. Raussendorf, R. & Harrington, J. Fault-tolerant quantum computation with high threshold in two dimensions. *Phys. Rev. Lett.* **98**, 190504 (2007).
26. Barrett, S. D. & Stace, T. M. Fault tolerant quantum computation with very high threshold for loss errors. *Phys. Rev. Lett.* **105**, 200502 (2010).
27. Auger, J. M., Anwar, H., Gimeno-Segovia, M., Stace, T. M. & Browne, D. E. Fault-tolerant quantum computation with non-deterministic entangling gates. *Phys. Rev. A* **97**, 030301 (2018).
28. Hein, M., Eisert, J. & Briegel, H. J. Multiparty entanglement in graph states. *Phys. Rev. A* **69**, 062311 (2004).
29. Briegel, H. J. & Raussendorf, R. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.* **86**, 910–913 (2001).
30. Gimeno-Segovia, M., Shadbolt, P., Browne, D. E. & Rudolph, T. From three-photon Greenberger-Horne-Zeilinger states to ballistic universal quantum computation. *Phys. Rev. Lett.* **115**, 020502 (2015).
31. O'Brien, J. L., Furusawa, A. & Vučković, J. Photonic quantum technologies. *Nat. Photon* **3**, 687–695 (2009).
32. Politi, A., Cryan, M. J., Rarity, J. G., Yu, S. & O'Brien, J. L. Silica-on-silicon waveguide quantum circuits. *Science* **320**, 646–649 (2008).
33. Wang, C., Harrington, J. & J. Preskill, J. G. Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. *Ann. Phys.* **303**, 31–58 (2003).
34. Dijkstra, E. A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271 (1959).
35. Hart, P. E., Nilsson, N. J. & Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **4**, 100–107 (1968).



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2018