


## ARTICLE OPEN

## One-dimensional quantum computing with a ‘segmented chain’ is feasible with today’s gate fidelities

Ying Li <sup>1</sup> and Simon C. Benjamin <sup>1</sup>

Building a quantum computer with a one-dimensional (1D) architecture, instead of the typical two-dimensional (2D) layout, could be significantly less difficult experimentally. However such a restricted topology necessitates a large overhead for shuffling qubits and consequently the fault tolerance threshold is far lower than in 2D architectures. Here we identify a middle ground: a 1D segmented chain which is a linear array of segments, each of which is a well-connected zone with all-to-all connectivity. The architecture is relevant to both ion trap and solid-state systems. We establish that fault tolerance can be achieved either by a surface code alone, or via an additional concatenated four-qubit gauge code. We find that the fault tolerance threshold is 0.12%, a feasible error rate with today’s technology, using 15-qubit segments, while larger segments are superior. For 35 or more qubits per segment one can achieve computation on a meaningful scale with today’s state-of-the-art fidelities without the use of the upper concatenation layer, thus minimising the overall device size.

*npj Quantum Information* (2018)4:25 ; doi:10.1038/s41534-018-0074-2

## INTRODUCTION

Quantum computation can solve certain problems that are intractable for classical computation, e.g., the quantum Shor’s algorithm can solve the integer factorization problem in polynomial time while the best known classical algorithm runs in exponential time.<sup>1</sup> To implement such quantum algorithms in a meaningful scale, we need a quantum computer that contains millions of qubits<sup>2,3</sup> with the noise suppressed to the sub-threshold regime.<sup>4–8</sup> These qubits must be coupled by controllable interactions to form a network. Generally as we consider higher levels of connectivity between qubits, the quantum computer is found to tolerate more errors, i.e. the noise threshold is higher. Schemes where qubits lie in a one-dimensional (1D) array with nearest-neighbouring (NN) interactions have an estimated error-rate threshold ranging from  $10^{-7}$  to  $10^{-5}$  per gate,<sup>4,5</sup> and a recent study<sup>6</sup> indicates this may reach  $10^{-4}$ . However when qubits form a two-dimensional (2D) array with NN interactions, the threshold is about 1% per gate.<sup>7,8</sup> In this paper, we study fault-tolerant quantum computing in a 1D array of qubits which interact locally but with a range beyond NN distance. Specifically we consider a segmented chain, where each segment is a small region within which all qubits can couple directly; we find the error threshold can exceed 0.1% per gate with quite modest segment sizes, therefore fault tolerance in a 1D qubit array is feasible in the sense that the required error rate is realistic.

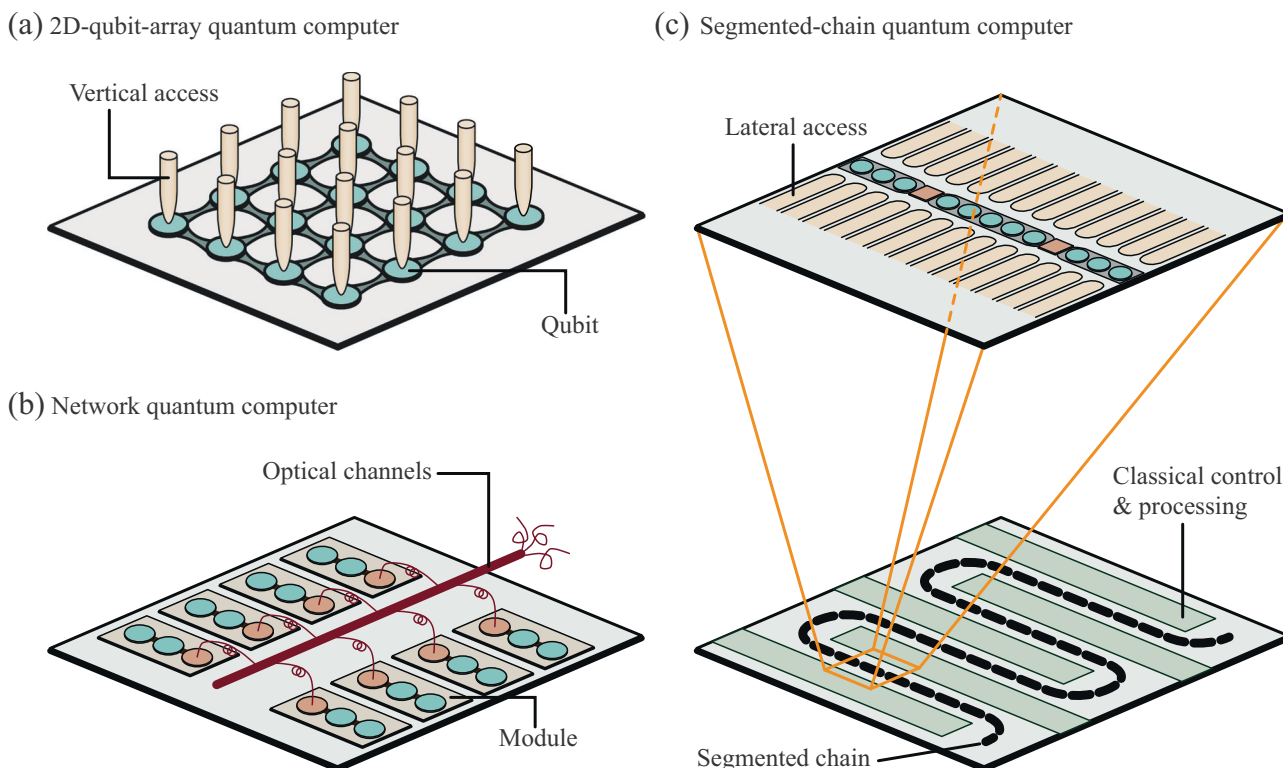
An advantage of using a 1D array of qubits as a quantum computer is that the system could be embedded entirely in a 2D surface: all control systems could also lie in that surface, whereas for a 2D array the qubits themselves and their links would obstruct in-plane access and necessitate access from the z-direction (see Fig. 1a). Gate error rates well below 1% have been demonstrated in 1D qubit arrays using ion traps<sup>9–11</sup> and superconducting qubits.<sup>12</sup> If these same error rates could be achieved in a 2D array, notwithstanding the challenge of routing control systems, then

one would have achieved a sub-threshold fault-tolerant quantum computer.<sup>8</sup> This is therefore the focus of much theoretical and experimental work. For example one proposed computer would be formed from a grid of many abutting small ion traps, with communications achieved by ion transport across gaps.<sup>13</sup> For superconducting qubits, a 2D qubit array can be fabricated on a surface, and solutions for vertical access are being investigated (Fig. 1a).<sup>14</sup> Alternatively, systems with the potential for optical linkage could realise a true network paradigm, e.g. ion traps may be networked by collecting and routing emitted photons (Fig. 1b).<sup>15–17</sup> However, realising any of these 2D (or higher) connectivities on a practical scale remains a tremendous challenge. Here, we will show that a 1D qubit array can also tolerate a high level of noise, approaching the 2D thresholds, while potentially being fraught with fewer engineering challenges (Fig. 1c). Although 2D qubit arrays will still have advantages, e.g., a lower cost of communications within the qubit array, conversely 1D qubit arrays may be easier to optimise versus noise because of their structural simplicity.

Medium-range interactions in a 1D array, as required by our architecture, do exist in many quantum systems. In ion traps, ions in the same trap are coupled to common phonon modes. Mediated by such modes, an entangling two-qubit gate can be directly performed on any pair of qubits, i.e. qubits in the same trap are all-to-all connected (Fig. 2a).<sup>18,19</sup> A long linear trap with isolated sub-regions, each equivalent to a small trap, therefore constitutes one realisation of a segmented chain. Two adjacent regions would be coupled by shuttling an ion between them.<sup>20–22</sup> By a different physical mechanism, our requirement may also be achieved with superconducting qubits: when coupled to the same resonator they can also be all-to-all connected,<sup>23–25</sup> and two NN resonators can be coupled using a qubit interacting with both resonators<sup>24</sup> (Fig. 2a). In both platforms, the quantum computer is a chain of sub-systems, i.e., segments within which qubits are all-

<sup>1</sup>Department of Materials, University of Oxford, Parks Road, Oxford OX1 3PH, UK  
Correspondence: Ying Li (ying.li.phys@gmail.com)

Received: 17 September 2017 Revised: 30 April 2018 Accepted: 1 May 2018  
Published online: 23 May 2018



**Fig. 1** Quantum architectures. **a** A 2D array of coupled qubits fills the plane, necessitating the challenging task of bringing in control lines from above. **b** Alternatively a network of modules can be linked by optical channels, but this requires controlled interfacing of static and flying qubits (photons). **c** A potentially more simple architecture is the 1D qubit array with the segmented chain structure, which allows in-plane lateral access for control lines. Here we depict a snaking path, to illustrate that a 1D array can efficiently use a 2D surface

to-all connected. Two NN segments are coupled by a shared qubit, which is connected to all qubits in both segments. We find that, if there are enough qubits in each segment, a high level of noise is tolerable. We presume that the number of qubits in each segment, i.e. the interaction range, is fixed and does not scale with the overall size of the quantum computer.

## RESULTS

Our protocol for quantum error correction is based on the surface code,<sup>26</sup> with an optional additional level of encoding if the surface code alone proves insufficient. Using the interaction structure in the segmented chain qubit array, the surface code can be efficiently implemented, but its code distance is limited by the number of qubits in each segment.

Note that because qubits in the same segment may need to be operated sequentially, a surface-code error detection cycle in the segmented chain qubit array may be slower (in terms of the circuit depth) than a 2D qubit array by a factor determined by the segment size.

Logical qubits encoded in the surface code form a 1D array with NN interactions, i.e., each surface-code qubit can only directly talk to two NN surface-code qubits. Error rates for logical gates on surface-code qubits are of course determined by error rates of physical qubits and the limited code distance. If error rates of surface-code qubits are low enough, a quantum algorithm can be directly implemented using surface-code qubits; otherwise, we need to combine the surface code with another code above it to further correct errors. We choose the concatenated 1D four-qubit gauge code (see Methods) as the higher-level code.<sup>5,27,28</sup>

The noise threshold of our protocol depends on the segment size. Our numerical results suggest that given a physical error rate  $\sim 0.1\%$  per gate, which has been demonstrated in ion traps,<sup>9–11</sup>

and 35 qubits in each segment, on average  $10^{15}$  CNOT gates can be performed on surface-code qubits before a logical error occurs, in which case the concatenation with the gauge code is not required for implementing many quantum algorithms. When the additional concatenation is indeed used then one can use segments of any size greater than 4 to suppress logical errors arbitrarily, provided that the physical error rate is below a certain threshold. We determine this threshold curve, finding for example the threshold error rate is 0.12% when the segment size is 15.

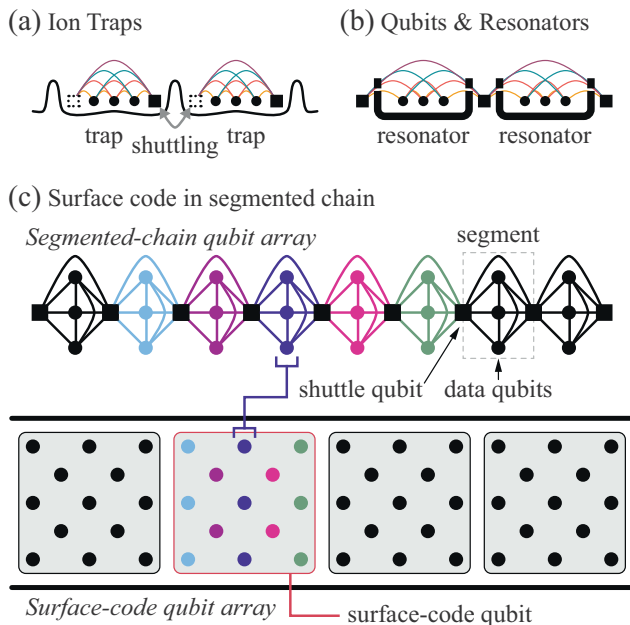
We will begin by discussing the surface code, and then the optional concatenation with the gauge code. We then discuss the performance of the whole error-correction protocol.

### Surface code

A surface-code logical qubit with the code distance  $d$  is encoded in  $d^2 + (d - 1)^2$  physical qubits, as shown in Fig. 2c, in which the code distance is  $d = 3$ . In a 2D array of physical qubits, implementing the surface code only requires interactions between neighbouring qubits.<sup>7,8</sup>

In a 1D array of physical qubits with the segmented chain structure, we allocate one segment to each column of physical qubits in the square array (see Fig. 2c): each qubit in the column of surface-code lattice maps to a qubit in the corresponding segment; intra-column gates are performed using interactions within the segment; and gates between NN columns are realised via shuttle qubits.

The code distance of the surface code depends on the size of segments. We define the size of segments  $s$  as the number of data qubits plus two shuttle qubits. For ion traps, this is the maximum number of ions in one trap; for qubit-resonator systems, this is the total number of qubits coupled to one resonator (Fig. 2a, b). In Fig. 2c, the size of segments is  $s = 5$ . The number of qubits in a column is either  $d$  or  $d - 1$  (long or short column). Limited by the segment



**Fig. 2** **a** A chain of ion traps. In each trap, qubits (ions, represented by circles and squares) are coupled to the same phonon modes, and two-qubit entangling gates (e.g., CNOT gates, represented by colour curves) can be performed on any pair of qubits. Traps are coupled by moving qubits (squares) between nearest neighbouring traps. **b** A chain of resonators. Similar to ion traps, in each resonator, qubits (e.g., superconducting qubits) are coupled to the same phonon mode, and two-qubit entangling gates can be performed on any pair of qubits. Two nearest neighbouring resonators are coupled by sharing a qubit. **c** The surface code in a segmented chain qubit array. The upper panel illustrates the interaction network in the segmented chain qubit array. Each segment contains several data qubits (circles) and two shuttle qubits (squares). Qubits within the same segment are all-to-all connected, i.e., two-qubit entangling gates (e.g., CNOT gates) can be directly performed on any pair of qubits. Two nearest neighbouring segments are coupled by sharing one shuttle qubit. The lower panel illustrates the array of surface-code qubits. Qubits unused in the encoding (e.g., shuttle qubits) are not shown in the lower panel. A surface-code logical qubit is encoded in several segments, and each segment provides qubits in one column of the surface code. CNOT gates can be performed on neighbouring surface-code qubits. These logical qubits form a one-dimensional quantum computer with nearest neighbouring interactions

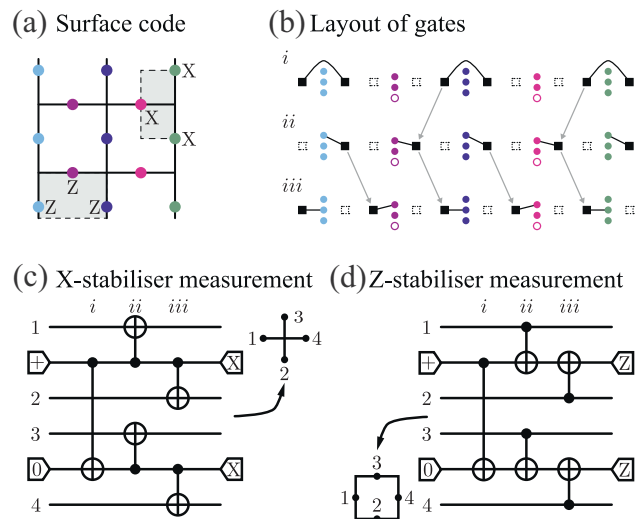
size, the code distance  $d \leq s - 2$ . In order to utilise the full computational power provided by the machine, we always choose  $d = s - 2$ .

In each short column, one data qubit in the corresponding segment is not used in the surface-code encoding. These unused qubits are useful for suppressing logical memory errors and gating surface-code qubits, which will be discussed in the Methods section.

An alternative to the standard surface code (or standard planar lattice) is the rotated lattice,<sup>29–31</sup> in which the surface code lattice is rotated for 45°. The advantage of the rotated lattice is that fewer qubits are demanded to achieve the same code distance. Our protocol could be adapted to the rotated lattice.

In the following, we will show how to implement the surface code in the segmented chain qubit array.

**Stabiliser measurements.** The surface code is a stabiliser code defined on the lattice in Fig. 3a, and errors are detected by repeatedly measuring stabilisers (see Methods for an introduction of the surface code).<sup>26</sup> The protocol for stabiliser measurements in the segmented chain qubit array is shown in Fig. 3b–d. Each



**Fig. 3** Surface-code stabiliser measurements in a segmented chain qubit array. **a** The lattice of the surface code. **b** The layout of two-qubit gates for measuring the first-row of X stabilisers. Circles are data qubits, and empty circles are unused data qubits in short columns. Squares are shuttle qubits, which are moved between nearest neighbouring columns as indicated by gray arrows. Black curves are CNOT gates. **c** Circuit of X-stabiliser measurements. **d** Circuit of Z-stabiliser measurements. Qubits 1, 2, 3 and 4 are data qubits, and other two qubits are shuttle qubits, which are initialised and measured in the circuit. The upper shuttle qubit is shared by segments of qubit-1 and qubit-2, and the lower shuttle qubit is shared by segments of qubit-3 and qubit-4. Qubit-2 and qubit-3 are in the same segment (column), which is a long column in **c** and a short column in **d**. The first CNOT gate prepares a Bell state on two shuttle qubits, which can be viewed as one qubit encoded in two qubits and prepared in the state either  $|0\rangle$  or  $|+\rangle$  depending on the code word. The encoded qubit plays exactly the same role as the only one ancillary qubit in usual stabiliser measurement circuits.<sup>26</sup>

column of physical qubits corresponds to a segment, and each segment has two shuttle qubits (Fig. 2c). We call columns with  $d$  qubits long columns and columns with  $d - 1$  qubits short columns. We use shuttle qubits as ancillaries to measure stabiliser operators as shown in circuits in Fig. 3c, d. Compared with the standard two-dimensional realisation of the surface code, in which one ancillary qubit serves each stabiliser, and stabilisers are measured in parallel, here shuttle qubits are recycled, and stabilisers are measured row by row. In Fig. 3b, we take the top row of X stabilisers (vertices) in Fig. 3a as an example. Firstly, shuttle qubits are initialised according to the circuit in Fig. 3c. In step-i, two shuttle qubits of each long-column segment are entangled using interactions within the segment, corresponding to the step-i CNOT gate in Fig. 3c. In step-ii, shuttle qubits on the right side of long columns stay in long columns, shuttle qubits on the left side of long columns are moved leftward to short columns, and then each shuttle qubit interacts with a data qubit in the corresponding column to perform step-ii CNOT gates in Fig. 3c. In step-iii, all shuttle qubits are moved rightward and interact with data qubits to perform step-iii CNOT gates in Fig. 3c. Finally, shuttle qubits are measured to read stabiliser values according to the circuit in Fig. 3c. We would like to remark that, in ion traps, shuttle qubits are physically moved between NN traps; but in qubit-resonator systems, moving shuttle qubits only means using different resonators. In five steps (including shuttle-qubit initialisation and measurement and three rounds of interactions), one row of X stabilisers are measured. By measuring X stabilisers row by row, we need  $5(d - 1)$  steps to complete the X-stabiliser measurements. Z-stabiliser measurements are similar (see Fig. 3d). To measure Z stabilisers, the roles of long columns and short

columns are exchanged, i.e., two shuttle qubits of each short-column segment are entangled at the beginning, and then shuttle qubits are moved and interact with data qubits in the similar way. By measuring Z stabilisers row by row, we need  $5d$  steps to complete Z-stabiliser measurements. Therefore, a full round of stabiliser measurements needs  $5(2d - 1)$  steps.

Circuits for measuring three-qubit stabilisers can be obtained from circuits in Fig. 3c, d by removing a data qubit and corresponding gates.

We can conclude that stabiliser measurements only require the initialisation and measurement operations on shuttle qubits and interactions involving shuttle qubits. Single-qubit gates are not explicitly shown in the circuits, which may be needed for adjusting the initialisation/measurement basis. Although we focus on platforms with all-to-all interactions in a segment, direct interactions between data qubits are not necessary: we can transfer the state of a data qubit to a shuttle qubit to indirectly implement a two-qubit gate on the data qubit and another data qubit. Similarly, direct initialisation, measurement and single-qubit gate on data qubits are also not necessary and can be indirectly implemented by exchanging states between data qubits and shuttle qubits. Realising data-qubit operations indirectly causes more errors, but because they are not used in stabiliser measurements, these extra errors will not change the performance of the quantum error correction significantly. In this paper, we assume that data qubits can be controlled directly, but the conclusion will be similar for systems without direct control on data qubits. As a remark, we have assumed that only one CNOT gate can be performed in a segment at a time.

**Post-correction error rate.** After the error correction, most of the errors can be corrected. However, there is still a small chance that the protected logical information is affected by errors when the error correction fails. The rate of logical errors depends on the rate of errors in each operation on the physical-qubit level and the code distance. For the surface-code, usually when the physical-qubit error rate is lower than a threshold value, the logical-qubit error rate decreases with the code distance. We numerically studied the performance of the surface code using the Monte Carlo method (see Methods section). The error-rate threshold is determined; the logical-qubit error rate in the shallow sub-threshold regime (the physical error rate is relatively high and the code distance is small) is calculated directly, and the logical-qubit error rate in the deep sub-threshold regime is estimated using extrapolation.

We model the noise in the quantum computer as depolarising errors, and we assume that all these error rates are the same except the single-qubit error rate, which is assumed to be tenth of other error rates, i.e.,  $\epsilon_1 = \epsilon_M = 10\epsilon_2 = \epsilon_3$ . Here,  $\epsilon_1$ ,  $\epsilon_M$ ,  $\epsilon_1$ , and  $\epsilon_2$  are error rates of initialisations, measurements, single-qubit gates and two-qubit gates, respectively. See Methods for more details of the error model. Usually gate fidelities in ion trap systems depend on the temperature of ions. Shuttling an ion between two traps can excite the motional state of the ion, therefore cooling operations are required to mitigate the heat caused by shuttle operations. Fortunately the temperature of ions is stable with the help of such cooling operations,<sup>9–11</sup> and therefore errors associated with shuttle operations can be neglected. In superconducting qubit systems, gate fidelities are usually different when qubits are coupled to different resonators. This difference is hardware specific and is neglected in our model.

The noise in the identity operation (decoherence, memory errors) is modelled in the same fashion as the noise in single-qubit gates (i.e., the noise superoperator is the same as  $\mathcal{N}_1$  in Methods). We use  $\epsilon_0$  (replacing  $\epsilon_1$  in  $\mathcal{N}_1$ ) to denote the rate of memory errors. The memory-error rate depends on the duration of the identity operation. We assume durations of initialisations, measurements and two-qubit gates are the same, and the duration of

single-qubit gates is negligible compared with other operations. Then, we specify that  $\epsilon_0$  denotes the error rate during the time of a two-qubit gate. Implementing the surface code in the segmented chain qubit array, the time to perform one round of stabiliser measurements increases with the code distance. The number of steps required by a full round of stabiliser measurements grows as  $5(2d - 1)$ , whereas in two-dimensional qubits arrays these steps are run in parallel (i.e., the number of steps is independent of  $d$ ). If we assume  $\epsilon_0$  is independent of the code distance, an error-rate threshold for the surface code does not exist. Therefore, we assume that the rate of memory errors during one round of stabiliser measurement is equivalent to the error rate of two-qubit gates, i.e.,  $\epsilon_0$  depends on the code distance and  $5(2d - 1)\epsilon_0 = \epsilon_2$ . This relation sets a requirement on the memory error rate, i.e., a longer coherence time is required when the segment size is larger. We would like to remark that, in a segmented chain quantum computer, the scalability is achieved by using more segments in the chain instead of increasing the size of each segment. Therefore, the code distance  $d$  and the required memory error rate are always finite and do not scale with the computer size. We will show that the required memory error rate is realistic for today's technologies, e.g., in ion traps.<sup>9,10,32</sup>

The rate of errors on a surface-code qubit per round of stabiliser measurements  $p_L$  is plotted in Fig. 4a. The threshold of the CNOT gate error rate is at  $\epsilon_2^{\text{th}} \simeq 0.7\%$ . If the physical error rate is lower than the threshold, the logical error rate decreases with the code distance. The logical error rate is fitted using the formula

$$p_L = \exp[(a \log \epsilon_2 + \beta)(d + \delta) + \gamma], \quad (1)$$

where parameters  $a$ ,  $\beta$ ,  $\gamma$ , and  $\delta$  are obtained from the fitting (see Appendix). As shown in Fig. 4a, this formula provides a good fit to the logical error rate in the shallow sub-threshold regime. In the following, we will use the same formula to estimate the logical error rate in the deep sub-threshold regime. An empirical formula<sup>2</sup> used in the literature is

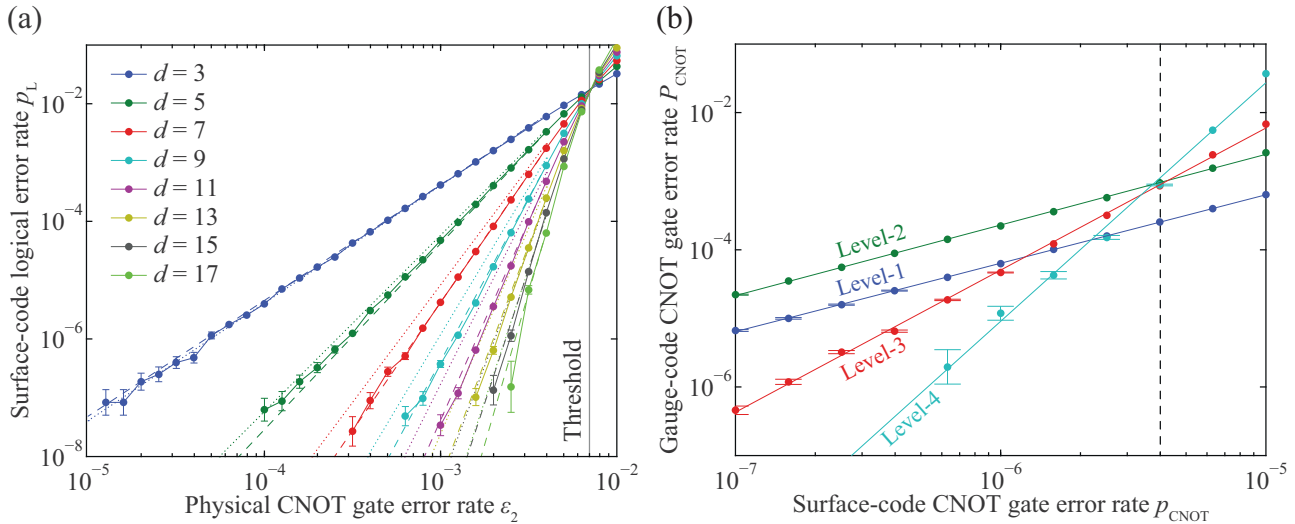
$$p_L = p_{\text{th}} (\epsilon_2 / \epsilon_2^{\text{th}})^{(d+1)/2}, \quad (2)$$

where we take  $p_{\text{th}} = 0.02$ . This empirical formula coincides with Eq. (1) but parameters are different, i.e.,  $a = 0.5$ ,  $\beta = -a \log \epsilon_2^{\text{th}} = 2.4809$ ,  $\gamma = \log p_{\text{th}} = -3.9120$  and  $\delta = 0.5$ . The logical error rate according to the empirical formula is also plotted in Fig. 4a. We can find that the logical error rate estimated using the empirical formula is obviously higher than the value directly calculated using the Monte Carlo method, i.e., Eq. (2) provides a conservative estimation of the logical error rate.

**Surface-code logical gates.** Fault-tolerant operations that can be directly performed on surface-code qubits include initialisations and measurements in the 0/1 basis and the  $+/-$  basis, Pauli gates, Hadamard gates and CNOT gates. A fault-tolerant initialisation (measurement) is realised by initialising (measuring) all data qubits in the corresponding basis, and a fault-tolerant Pauli gate is realised by performing a sequence of Pauli gates. In Appendix, we give protocols of fault-tolerant Hadamard gates and CNOT gates in the 1D qubit array with the segmented chain structure.

Provided these fault-tolerant operations, the universality of quantum computing is completed by introducing magic states.<sup>33</sup> Magic states can be encoded using stabiliser measurements and distilled using fault-tolerant operations. High-fidelity (on the level of fault-tolerant operations) Clifford gates  $S$  and non-Clifford gates  $T$  can be realised using distilled magic states.<sup>34</sup>

Our protocol for fault-tolerant CNOT gates only uses single-qubit operations and stabiliser measurements on a surface-code lattice with the width of one logical qubit [see the surface-code qubit array in Fig. 2c], therefore it can be implemented in the segmented chain qubit array. CNOT gates are transverse gates of the surface code,<sup>26</sup> however, transverse CNOT gates require



**Fig. 4** **a** The rate of errors on a surface-code logical qubit per round of stabiliser measurements  $p_L$  as a function of the physical-qubit CNOT-gate error rate  $\epsilon_2$  and the code distance  $d$ . When the physical error rate is lower than the threshold marked by the vertical gray line, the logical error rate decreases with the code distance. Circles are data calculated numerically using the Monte Carlo method. Dashed lines are obtained by fitting circles (in the sub-threshold regime) using Eq. (1). Dotted lines are calculated using Eq. (2). Error bars show one standard deviation, and error bars smaller than the size of circles have been removed from the figure. **b** The logical CNOT gate error rate of the four-qubit gauge code  $P_{\text{CNOT}}$  as a function of the logical CNOT gate error rate of the surface code  $p_{\text{CNOT}}$ . Gauge-code logical qubits are encoded in surface-code logical qubits, and the gauge code is concatenated. The level of the gauge-code concatenation is marked in the figure. The surface-code CNOT gate error rate is  $p_{\text{CNOT}} = 14dp_L$ . Error bars show one standard deviation, and error bars with invisible gaps have been removed from the figure

interactions over the range of at least one logical qubit (i.e., a qubit needs to talk to a qubit that is  $\sim 2d^2$  qubits away in the 1D qubit array). Other protocols of fault-tolerant CNOT gates include braiding holes on a punched surface<sup>7</sup> and lattice surgery,<sup>30</sup> which only use neighbouring interactions but need the surface-code lattice to be two-dimensional with the minimum length of two logical qubits in both directions.

The overall flow of our protocol for fault-tolerant CNOT gates is shown in Fig. 5, which is based on the lattice surgery.<sup>30</sup> Details of the protocol are given in the Appendix.  $X_c$  and  $Z_c$  ( $X_t$  and  $Z_t$ ) denote Pauli operators of the Control (Target) surface-code qubit. By using ancillary surface-code qubits  $A1$ ,  $A2$ , and  $A3$ , these Pauli operators are deformed in each step, and eventually we realise a transformation from  $Z_c, Z_t, X_c$  and  $X_t$  to  $Z_c, Z_c Z_t, X_c X_t$  and  $X_t$ , respectively, which is a CNOT gate.

The three-dimensional illustration of the protocol is shown in the right column of Fig. 5. Each block has the dimension  $\sim d \times d \times d$  and represents  $\sim d$  rounds of stabiliser measurements on a surface-code qubit. Each fault-tolerant CNOT gate has in total 14 blocks. There are 16 blocks in the figure, but two of them are due to two input surface-code qubits. One can find that the distance between any pair of disconnected red (green) strips is  $\sim d$ . Because red and green strips represent rough and smooth boundaries respectively, the distance between strips corresponds to the code distance, i.e., the minimum number of single-qubit errors that can change the logical state but cannot be detected by stabilisers. Because the strip distance is  $\sim d$ , our protocol is fault-tolerant.

### One-dimensional four-qubit gauge code

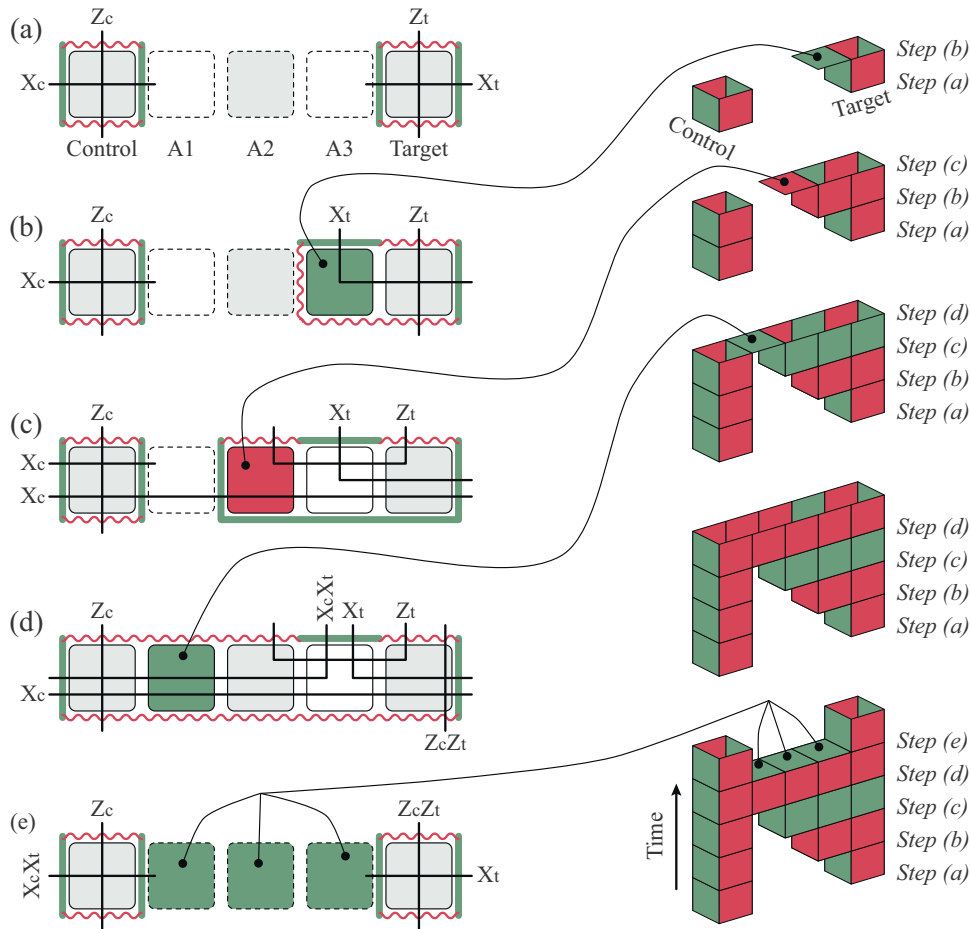
In the segmented chain qubit array, the code distance of the surface code is limited by the size of segments. If the logical error rate provided by the surface code is not low enough for implementing a quantum algorithm, we need another code on top of the surface code to further reduce the logical error rate.

The array of surface-code qubits is 1D and only has NN interactions (Fig. 2c). As shown in Fig. 5, to perform the CNOT gate on a pair of surface-code qubits, we need three surface-code

qubits between them as ancillaries. In the surface-code qubit array, we can choose one surface-code qubit to carry the information in every four of them, and other three surface-code qubits are used as ancillary qubits for performing CNOT gates between information qubits. In this way, we need four surface-code qubits to actually encode one bit of information. We can more efficiently use surface-code qubits by removing some ancillary qubits. The state of a surface-code qubit can be transferred to the NN surface-code qubit (see Appendix). Therefore, in the extreme case that we have only three ancillary surface-code qubits at all, a CNOT gate can be performed by moving these three ancillary qubits to the right place. However, in this case, CNOT gates cannot be performed in parallel. In the following, we assume that only one in four surface-code qubits is the information qubit, so that CNOT gates can be performed in parallel between NN information surface-code qubits.

The quantum error correction in a 1D qubit array with NN interactions has been studied in the literature. The four-qubit gauge code is a successful code for 1D quantum error correction, whose threshold is estimated to be about  $10^{-5}$ .<sup>5</sup> In this paper, we will focus on this four-qubit gauge code.

To study the performance of the four-qubit gauge code implemented using surface-code qubits, we need to know the error rate of surface-code logical operations. Operations required by the four-qubit gauge code includes initialisations and measurements in the 0/1 basis and the  $+/-$  basis, CNOT gates and SWAP gates between NN qubits. We assume that surface-code stabilisers are measured for  $d$  rounds after a surface-code qubit is initialised or before measured. Therefore, we estimate the error rate of initialisations and measurements on surface-code qubits as  $p_{I/M} = dp_L$ . We assume that surface-code stabilisers are also measured for  $h = d$  rounds after each step in the surface-code CNOT gate (Fig. 5). Therefore, we estimate the logical error rate of surface-code CNOT gates as  $p_{\text{CNOT}} = 14dp_L$ , where 14 is number of blocks in the surface-code CNOT gate, and each block corresponds to  $d$  rounds of stabiliser measurements on a surface-code qubit. A SWAP gate is realised by three CNOT gates, and its error rate is  $p_{\text{SWAP}} = 3p_{\text{CNOT}}$ . The rate of logical memory errors depends on the



**Fig. 5** Protocol for CNOT gates in a one-dimensional array of surface-code logical qubits. In the left column, squares denote surface-code qubits, red wavy lines denote rough boundaries, and green bold lines denote smooth boundaries. Any path connecting two rough (smooth) boundaries represents a  $Z$  ( $X$ ) operator of the surface-code qubit. The controlled-NOT gate on two surface-code qubit can be realised following the procedure from **a–e**. In steps **(a–c)**, the target qubit is deformed. In step **(d)** Control and Target qubits are merged, then they are splitted in step **(e)**, as in usual lattice surgery.<sup>30</sup> The right column illustrates the evolution of rough and smooth boundaries with time. In order to separate rough (smooth) boundaries with sufficient distance (so that the logical information is protected), three ancillary surface-code qubits A1, A2, and A3 are required. See Appendix for details

duration of the logical identity operation. For the duration of initialisations and measurements, the rate of memory errors is  $p_0 = dp_L$ ; for CNOT gates, it is  $4p_0$ ; for SWAP gates, it is  $12p_0$ . We remark that these surface-code logical error rates are only for phase-flip errors, and it is similar for bit-flip errors.

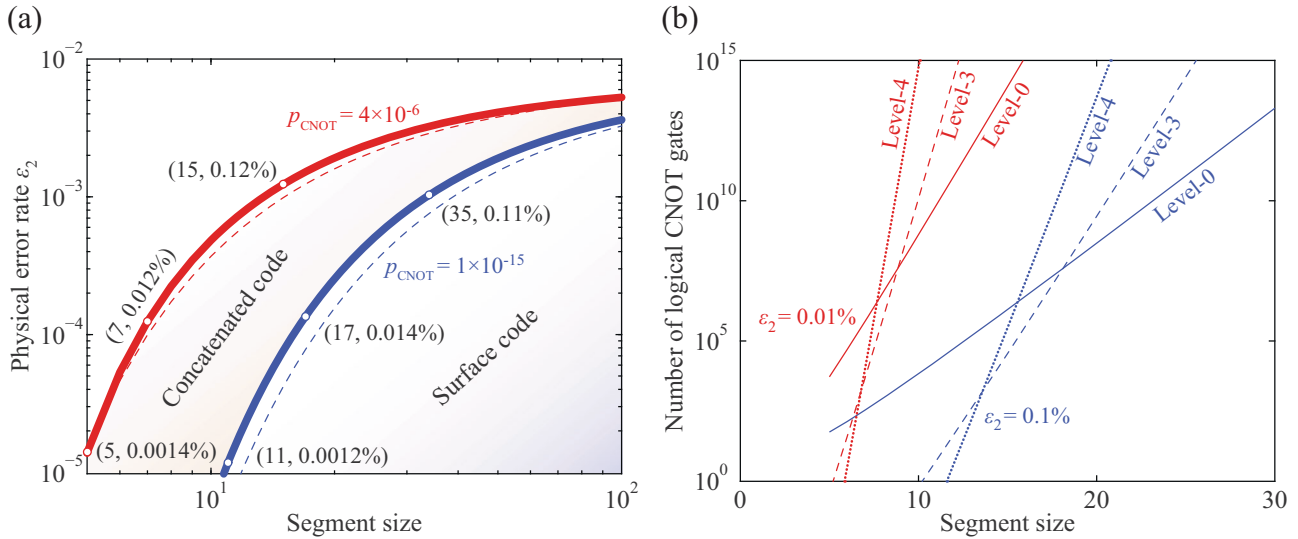
The method of estimating the error rate of logical operations used here, which is calculating the space-time volume of stabiliser measurements,<sup>2</sup> is not strictly accurate. However, a direct calculation of the logical error rate using the Monte Carlo method, e.g., for the CNOT gate, requires a simulation of four logical qubits for  $\sim 5d$  rounds of stabiliser measurement, which would be much harder than the numerical calculation that we have done in this paper (which used about 160,000 CPU hours). We have assumed that errors in logical CNOT gates are depolarised for simplification, which can also cause inaccuracy. All these assumptions in our numerical simulations will only change our result of the segment size slightly, because the logical error rate changes rapidly with the code distance. According to Eq. (2), by increasing the segment size by two qubits, the logical error rate can be reduced by a factor of 7 (70) for the physical error rate 0.1% (0.01%).

Based on our estimation of surface-code logical error rates, the error rate of gauge-code logical qubits is calculated using the Monte Carlo method, and the result is plotted in Fig. 4b. The code distance of the concatenated four-qubit gauge code is  $2^n$ , where  $n$  is the level of concatenation. For the first-level concatenation, the

code can only detect errors, because the code distance is 2. From the second-level concatenation, the code starts to have the ability of correcting errors. In Fig. 4b, for concatenation levels  $n = 2, 3, 4$ , a crossing point at  $p_{\text{CNOT}} = 4 \times 10^{-6}$  is observed, which indicates a threshold. Such a crossing point is evidence of a phase change: if the surface-code logical error rate is lower (higher) than the threshold, the gauge-code logical error rate will be reduced (increased) by increasing the level of concatenation. Consequently, when surface-code logical error rate is any finite distance into the sub-threshold regime, with sufficient levels of concatenation the gauge-code logical error rate can be suppressed to an arbitrarily low level (and thus lower than the error rate before the gauge code is used).

This threshold of the 1D quantum error correction is lower than the threshold  $10^{-5}$  reported in the ref.<sup>5</sup> because a different model of the noise is used. A recent paper proposed a protocol for the 1D quantum error correction using concatenated two-qubit repetition code,<sup>6</sup> in which an error-rate crossing at  $\sim 10^{-4}$  is observed between and error-correction concatenation level and error-detection concatenation levels. This crossing may indicate a threshold higher than the four-qubit gauge code. In our protocol, the code on top of the surface code can be any code that only uses NN interactions in a 1D qubit array.

The error rate  $p_{\text{CNOT}}$  of CNOT gates on gauge-code logical qubits increases with the surface-code error rate  $p_{\text{CNOT}}$ . This



**Fig. 6** **a** The physical error rate  $\epsilon_2$  and the segment size  $s$  required to achieve the surface-code logical error rate  $p_{\text{CNOT}}$ . The red curve corresponds to  $p_{\text{CNOT}} = 4 \times 10^{-6}$ , and the blue curve corresponds to  $p_{\text{CNOT}} = 10^{-15}$ . These two curves are calculated using Eq. (1) with parameters given in Table 1 in the Appendix. Results obtained using Eq. (2) are also plotted in the figure as dashed curves. Errors are firstly corrected using the surface code. When the physical error rate is not low enough or segments are not large enough (e.g., to achieve  $p_{\text{CNOT}} = 10^{-15}$ ), we need the concatenated code on top of the surface code to further correct errors.  $p_{\text{CNOT}} = 4 \times 10^{-6}$  is the threshold of the regime that the concatenated code works, therefore the red curve is the threshold of the overall protocol. **b** The average number of logical CNOT gates that can be performed in a quantum algorithm before getting one logical error. For the level-0 encoding, only the surface code is used to correct errors, and the number of gates is  $1/p_{\text{CNOT}}$ . For the level-3 and level-4 encoding, the number of gates is  $1/p_{\text{CNOT}}$ .  $p_{\text{CNOT}}$  is calculated using Eq. (1), and  $P_{\text{CNOT}}$  is calculated using Eq. (3). Parameters in these two equations are given in Tables 1 and 2 in the Appendix

dependence can be described using the formula

$$P_{\text{CNOT}} = \exp(\kappa_n \log p_{\text{CNOT}} + \eta_n), \quad (3)$$

where parameters  $\kappa_n$  and  $\eta_n$  depend on the concatenation level  $n$ . By fitting data in Fig. 4b, we obtain parameters  $\kappa_n$  and  $\eta_n$  (see Appendix). For first two levels of concatenations,  $\kappa_1, \kappa_2 \sim 1$ , which implies that the error correction does not work ( $P_{\text{CNOT}} \propto p_{\text{CNOT}}^{\kappa_n}$ ). The first-level concatenation does not work because it can only detect errors. The second-level concatenation does not work because of two-qubit errors that cannot be corrected by the code with distance 4. Therefore, the third level is the minimum level of encoding in order to take the advantage of the four-qubit gauge code to reduce errors.

### Fault-tolerant quantum computing

In the segmented chain qubit array, the overall protocol for fault-tolerant quantum computing depends on the logical error rate required by the computing task, the rate of physical errors and the size of segments. If the surface code cannot suppress the logical error rate to the level required by the task, we need to use the concatenated four-qubit gauge code to further reduce the logical error rate. In order to use the concatenated code, the surface code has to firstly suppress the logical error rate to be lower than  $p_{\text{CNOT}} = 4 \times 10^{-6}$ , which leads to a threshold of the physical error rate. In Fig. 6a, this threshold of the physical error rate is plotted as a function of the segment size. If the physical error rate is  $\epsilon_2 = 0.12\%$ , we need segments with more than 15 qubits to build a fault-tolerant quantum computer. If the physical error rate can be reduced to  $\epsilon_2 = 0.012\%$ , the minimum size of segments can be reduced to 7.

When each segment is large enough, the surface code itself is enough for many quantum-computing tasks. See Fig. 6a. Given the physical error rate  $\epsilon_2 = 0.11\%$  and segments with about 35 qubits in each one of them, the surface-code CNOT gate error rate is  $p_{\text{CNOT}} \approx 10^{-15}$ , which is enough for implementing the Shor's algorithm with a thousand qubits.<sup>2,3</sup> Similarly, if the physical error

rate can be reduced to  $\epsilon_2 = 0.014\%$ , the segment size only needs to be 17 to achieve the same logical error rate.

The performance of the overall protocol is plotted in Fig. 6b. The level-0 encoding means that only the surface code is used to correct errors. The performance of the surface-code-only error correction, the third-level and fourth-level concatenated gauge codes are compared. Given the gate error rate  $\epsilon_2 = 0.1\%$  and using the fourth level concatenation,  $10^{15}$  logical CNOT gates can be achieved with 21 qubits in each segment. The concatenation is expensive. Using the four-qubit gauge code, to encode a higher-level qubit, we need six lower-level qubits. If we use only one in every four surface-code qubits as the information qubit, each logical qubit with the level- $n$  concatenation requires  $4 \times 6^n$  surface-code qubits, i.e. the third-level (fourth-level) encoding needs 864 (5184) surface-code qubits per gauge-code logical qubit. We remark that each surface-code qubit needs  $2d - 1$  segments, each segment contains  $s - 1 = d + 1$  physical qubits on the average, and  $d$  is the distance of the surface code. Therefore, the overall number of physical qubits per logical qubit is  $4 \times 6^n \times (s - 1)(2s - 5)$ . As shown in Fig. 6b, using the concatenated code can reduce the required segment size but the effect is modest especially when the physical error rate is as low as  $\epsilon_2 = 0.01\%$ .

Because of the restricted topology of qubit couplings, the computing resource cost in the segmented-chain architecture is higher than the costs in a 2D qubit-array or network architecture. Stabiliser-measurement circuits for the segmented-chain architecture require more gates than the 2D architecture, which usually means that there are more errors caused by stabiliser measurements and a larger code distance is needed in order to correct the errors. However in our numerical results, we find that this effect is minor. The surface-code threshold for the segmented-chain architecture is comparable to the 2D architecture,<sup>2</sup> and according to Eq. (2) the threshold determines how rapidly the logical error rate decreases with the code distance.

In the segmented-chain architecture, we need to use the concatenated code on top of the surface code when the surface

code is not enough. In this case, because the concatenated code in the 1D qubit array is less efficient than the surface code, in the segmented-chain architecture we need significantly more qubits for successful error correction than in other architectures.

Finally, we should also recognise that since the ultimate logical qubits form a 1D nearest-neighbour array, on the algorithmic level the connectivity is lower than other (e.g., 2D or network) architectures. This may result in significant overheads depending on the algorithm.<sup>35</sup> However all the resource costs mentioned above may in practice be offset if lower physical error rates are possible in a 1D system (versus 2D or network) because of the more modest system complexity.

It is worth noting that in the near-term future quantum computers will exist that do not have enough qubits for implementing the full-scale quantum error correction, but which are complex enough so that they cannot be simulated using any classical computers (passing the so-called quantum supremacy limit). Without the quantum error correction, these early-stage quantum computers are not fault-tolerant, therefore only algorithms with shallow circuits can be implemented.<sup>36–43</sup> In the context of such uncorrected, shallow-circuit machines, the all-to-all connectivity within each segment of the architecture we describe here could be exploited on the algorithmic level. Thus the performance of the segmented-chain architecture may be comparable to other architectures in this case.

## DISCUSSION

We have discussed fault-tolerant quantum computing in 1D quantum computers with the segmented chain structure. Given the state-of-the-art error rate 0.1%, the size of each segment must be at least 15 qubits for fault-tolerance to be of benefit using the surface code concatenated with the 1D gauge code, and 35 qubits for large scale algorithms such as Shor's algorithm to be implemented only using the surface code. Each segment is a small quantum processor with all-to-all connections among qubits. Segments with 4 or 5 qubits have been demonstrated with ion traps<sup>18,19</sup> and superconducting qubits,<sup>24</sup> and the qubit number in each segment in these platforms can be extended to tens or even more qubits.<sup>18,44–47</sup> The disadvantage of the segmented chain structure is the computing speed. Because the all-to-all connectivity within each segment is due to the coupling to the same phonon or photon modes, interactions between qubits in the same segment could not be switched on simultaneously. As a result, segmented chain 1D quantum computers need more operation cycles than 2D quantum computers by a factor determined by the segment size. Therefore, a longer coherence time is required. In ion traps, the coherence time of qubits is about 50 s,<sup>9,10</sup> which is 500,000 times longer than  $\sim 100 \mu\text{s}$  the time cost of a two-qubit gate,<sup>9,10,24</sup> i.e. the memory error rate  $\epsilon_0 \sim 2 \times 10^{-6}$ , which allows a segment size as large as about  $50 \sim \epsilon_2/(10\epsilon_0)$  qubits for the gate error rate  $\epsilon_2 = 0.1\%$ . This coherence time of ion qubits can be increased by a factor of 12 by using dynamical decoupling.<sup>32</sup> Such a ratio of coherence time to gate time is still a challenge for superconducting qubits.<sup>48</sup> Some recent works have been focusing on increasing the coherence time of superconducting qubits<sup>49</sup> or coupling them to quantum memories, e.g. nitrogen-vacancy centres in diamond.<sup>50</sup> We remark that the computing speed also depends on the time cost of each operation cycle. As an alternative approach of building a quantum computer, the segmented chain structure avoids the need to expand the qubit array to higher dimensions, which reduces the complexity of the quantum computer and allows us to design the quantum computer based on the well-developed 1D quantum technologies and on-chip integrated circuit manufacturing technologies.

## METHODS

### Surface code

The surface code is a stabiliser code.<sup>1</sup> Conventionally, the stabiliser group of the surface code is illustrated using the lattice in Fig. 3a: each edge  $e$  represents a physical qubit, each vertex  $v$  represents an X stabiliser operator, and each plaquette  $p$  represents a Z stabiliser operator.<sup>26</sup> Using  $v$  to denote the set of edges connected to the vertex and  $p$  to denote the set of edges on the perimeter of the plaquette, stabiliser operators are  $S_v =$

$$\prod_{e \in v} X_e \text{ and } S_p = \prod_{e \in p} Z_e.$$

Here,  $X_e$  and  $Z_e$  are Pauli operators of the corresponding qubit. These operators generate an Abelian group  $\langle \{S_v, S_p\} \rangle$  that defines the logical subspace, i.e. the logical information is encoded in the trivial representation subspace of the group.

Errors are detected by repeatedly measuring  $\{S_v, S_p\}$  to read out their eigenvalues using circuits in Fig. 3c, d. Eigenvalues of the same operator in two sequential measurements are compared. We notice the presence of errors if any change of eigenvalues is observed. The event that the eigenvalue of a stabiliser generator is changed called an error syndrome. By analysing error syndromes, we can work out the correction operation and undo errors. The algorithm working out the correction operation is called decoder. In our numerical simulation, we use the decoder proposed in ref.<sup>8</sup> which is based on the minimum-weight perfect matching algorithm<sup>51</sup> and optimised for errors with correlations. Here, we outline the decoder. Two three-dimensional lattices are introduced to represent potential bit-flip and phase-flip errors, respectively. Two dimensions correspond to the surface-code lattice, and one dimension represents time. Each vertex represents a comparison of eigenvalues, therefore error syndromes form a subset of vertices. Given measurement circuits in Fig. 3c, d, any Pauli error in the model (see next section) can cause at most two error syndromes. Then an edge connecting two corresponding vertices (or connecting one vertex and the boundary) represents the Pauli error. Each edge has a weight determined by the rate of corresponding Pauli error, and the weight is lower if the error rate is higher. The correction operation is to undo Pauli errors on paths that pair error syndromes on the lattice with the minimum total weight, and minimum-weight paths are found using the algorithm in the ref.<sup>51</sup>.

Two different sets of errors may result in the same error syndromes. Therefore, given error syndromes, the correction operation may be different from the actual errors to be corrected, resulting in some remaining errors on the state. These errors after the error correction cannot be detected by further stabiliser measurements, but frequently they are not harmful to the logical state. However, if the logical state is changed by a set of post-correction errors, the error correction has failed.

### Error model

We model the noise in the quantum computer as depolarising errors. Operations used in stabiliser measurements (Fig. 3c, d) include initialisations, measurements, Hadamard gates and CNOT gates (Hadamard gates are used for adjusting the initialisation/measurement basis). When a qubit is supposed to be initialised in the state  $|0\rangle$ , the qubit may be initialised in the incorrect state  $|1\rangle$  with the probability  $\epsilon_1$ . When a qubit is measured in the 0/1 basis, the measurement outcome is incorrect with the probability  $\epsilon_M$ . Initialisations and measurements in the  $+/-$  basis are realised using initialisations and measurements in the 0/1 basis and Hadamard gates. A quantum gate with noise can be expressed as a superoperator  $\mathcal{N}[U]$ , where  $[U]\rho = U\rho U^\dagger$  represents the unitary gate, and  $\mathcal{N}$  is a superoperator represents the noise. For single-qubit gates, the noise superoperator is

$$\mathcal{N}_1 = \left(1 - \frac{4}{3}\epsilon_1\right)[1] + \frac{\epsilon_1}{3} \sum_{a=0}^3 [\sigma^{(a)}]. \quad (4)$$

For two-qubit gates, the noise superoperator is

$$\mathcal{N}_2 = \left(1 - \frac{16}{15}\epsilon_2\right)[1] + \frac{\epsilon_2}{15} \sum_{a=0}^3 \sum_{b=0}^3 [\sigma_1^{(a)} \sigma_2^{(b)}]. \quad (5)$$

Here,  $\epsilon_1$  and  $\epsilon_2$  are rates of errors per gate,  $\sigma_i^{(a)}$  is a Pauli operator of qubit- $i$ , and  $a = 0, 1, 2, 3$  respectively correspond to 1, X, Y and Z. We assume that all these error rates are the same except the single-qubit error rate, which is assumed to be tenth of other error rates, i.e.,  $\epsilon_1 = \epsilon_M = 10\epsilon_2 = \epsilon_2$ .



## Simulation of surface-code error correction

In the model under consideration, errors are stochastic, i.e., a Pauli error either does not occur, or occurs with a specific probability. Therefore, we can simulate the errors using the Monte Carlo method: Pauli errors are randomly generated numerically according to their probabilities. For each run, the error correction is implemented, and we check whether the error correction is successful or not. By repeating the simulation for many runs, we can calculate the probability of failed error correction events, i.e., the logical error rate, which is plotted in Fig. 4a.

We would like to remark that the logical error rate in Fig. 4a is the rate of phase errors. Each X-stabiliser measurement for detecting phase errors needs two more Hadamard gates (for measuring shuttle qubits in the  $+/-$  basis) than a Z-stabiliser measurements (Fig. 3c, d). Therefore, the chance that an X-stabiliser measurement reports a false outcome is higher than a Z-stabiliser measurement, and the rate of logical phase errors is slightly higher than the rate of logical bit errors. In our numerical simulations, we have considered the surface code in the orientation shown in Fig. 3a, in which qubits in a column are in the same segment. As the dimension of the surface code in the vertical (column) direction is restricted, we can use more segments to increase the dimension in the horizontal direction. In this way, we can reduce the rate of logical bit errors exponentially as a function of the horizontal dimension (number of segments) at the price of increasing the rate of logical phase errors linearly. Therefore, it is essential to study logical phase error rather than logical bit errors.

Unused qubits in short-column segments can be used to reduce logical phase errors. Because of the all-to-all connectivity within each segment, exploiting these unused qubits we can change the surface code to a hybrid-boundary-condition code, in which the boundary condition along the horizontal direction is open (like the surface code) but the boundary condition along the vertical direction is closed (like the toric code). Because the toric code has a lower logical error rate due to the boundary condition, we can suppress logical phase errors in this way.<sup>52</sup>

## Four-qubit code

The four-qubit gauge code, which also referred as  $[4,1,2]$  subsystem code or  $2 \times 2$  Bacon-Shor code,<sup>27,28</sup> encodes one logical qubit in four qubits. The stabiliser group is generated by  $X_1X_2X_3X_4$  and  $Z_1Z_2Z_3Z_4$ . Given the stabiliser, up to two qubits can be encoded, and their logical Pauli operators are respectively  $\{X_1X_2, Z_1Z_3\}$  for qubit-1 and  $\{X_1X_3, Z_1Z_2\}$  for qubit-2. We can choose one of two qubits (qubit-1) as the logical qubit storing information, and the other qubit (qubit-2) as the gauge qubit. The gauge qubit is used to assist measuring stabiliser generators.

## Data availability

The data that support the findings of this study are available from the authors upon reasonable request.

## ACKNOWLEDGEMENTS

This work was supported by the EPSRC National Quantum Technology Hub in Networked Quantum Information Technology (EP/M013243/1). The authors would like to acknowledge the use of the University of Oxford Advanced Research Computing (ARC) facility in carrying out this work. <https://doi.org/10.5281/zenodo.22558>.

## AUTHOR CONTRIBUTIONS

Both authors contributed to the research and wrote the paper. Ying Li designed the detailed protocols and performed the numerical simulations.

## ADDITIONAL INFORMATION

**Supplementary Information** accompanies the paper on the *npj Quantum Information* website (<https://doi.org/10.1038/s41534-018-0074-2>).

**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

- Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information*. (Cambridge University Press, Cambridge, 2010).
- 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).
- O'Gorman, J., & Campbell, E.T. Quantum computation with realistic magic state factories. *Phys. Rev. A* **95**, 032338 (2017).
- Szkopek, T. et al. Threshold error penalty for fault-tolerant quantum computation with nearest neighbor communication, *IEEE Trans. IEEE Trans Nano* **5**, 42 (2006).
- Stephens, A. M. & Evans, Z. W. E. Accuracy threshold for concatenated error detection in one dimension. *Phys. Rev. A* **80**, 022313 (2009).
- Jones, C. et al. A logical qubit in a linear array of semiconductor quantum dots, arXiv:1608.06335
- Fowler, A. G., Stephens, A. M. & Groszkowski, P. High-threshold universal quantum computation on the surface code. *Phys. Rev. A* **80**, 052312 (2009).
- Wang, D. S., Fowler, A. G. & Hollenberg, L. C. L. Surface code quantum computing with error rates over 1%. *Phys. Rev. A* **83**, 020302(R) (2011).
- Harty, T. P. et al. High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit. *Phys. Rev. Lett.* **113**, 220501 (2014).
- 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).
- Gaebler, J. P. et al. High-fidelity universal gate set for  $^9\text{Be}^+$  ion qubits. *Phys. Rev. Lett.* **117**, 060505 (2016).
- Barends, R. et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature* **508**, 500 (2014).
- Lekitsch, B. et al. Blueprint for a microwave trapped ion quantum computer. *Sci. Adv.* **3**, e1601540 (2017).
- Béjanin, J. H. et al. Three-dimensional wiring for extensible quantum computing: The quantum socket. *Phys. Rev. Appl.* **6**, 044010 (2016).
- Li, Y. & Benjamin, S. C. High threshold distributed quantum computing with three-qubit nodes. *New J. Phys.* **14**, (093008) (2012).
- Nickerson, N. H., Li, Y. & Benjamin, S. C. Topological quantum computing with a very noisy network and error rates approaching one percent. *Nat. Commun.* **4**, (1756) (2013).
- Monroe, C. et al. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Phys. Rev. A* **89**, 022317 (2014).
- Choi, T. et al. Optimal quantum control of multimode couplings between trapped ion qubits for scalable entanglement. *Phys. Rev. Lett.* **112**, 190502 (2014).
- Debnath, S. et al. Demonstration of a small programmable quantum computer with atomic qubits. *Nature* **536**, 63 (2016).
- Kielinski, D., Monroe, C. & Wineland, D. J. Architecture for a large-scale ion-trap quantum computer. *Nature* **417**, 709 (2002).
- Rowe, M. A. et al. Transport of quantum states and separation of ions in a dual RF ion trap. *Quantum Inf. Comput.* **2**, 257 (2002).
- Kaufmann, P., Gloger, T. F., Kaufmann, D., Johanning, M. & Wunderlich, C. High-fidelity preservation of quantum information during trapped-ion transport. *Phys. Rev. X* **120**, 010501 (2018).
- Paik, H. et al. Experimental demonstration of a resonator-induced phase gate in a multiqubit circuit-QED system. *Phys. Rev. Lett.* **117**, 250502 (2016).
- Takita, M. et al. Demonstration of weight-four parity measurements in the surface code architecture. *Phys. Rev. Lett.* **117**, 210505 (2016).
- Song, C. et al. Continuous-variable geometric phase and its manipulation for quantum computation in a superconducting circuit. *Nat. Commun.* **8**, 1061 (2017).
- Dennis, E., Kitaev, A., Landahl, A. & Preskill, J. Topological quantum memory. *J. Math. Phys.* **43**, 4452 (2002).
- Bacon, D. Operator quantum error-correcting subsystems for self-correcting quantum memories. *Phys. Rev. A* **73**, 012340 (2006).
- Terhal, B. M. Quantum error correction for quantum memories. *Rev. Mod. Phys.* **87**, 307 (2015).
- Bombin, H. & Martin-Delgado, M. A. Optimal resources for topological two-dimensional stabilizer codes: Comparative study. *Phys. Rev. A* **76**, 012305 (2007).
- Horsman, C., Fowler, A. G., Devitt, S. & Van, R. Meter, Surface code quantum computing by lattice surgery. *New J. Phys.* **14**, 123011 (2012).
- Tomita, Y. & Svore, K. M. Low-distance surface codes under realistic quantum noise. *Phys. Rev. A* **90**, 062320 (2014).
- Wang, Y. et al. Single-qubit quantum memory exceeding 10-minute coherence time. *Nature Photonics* **11**, 646–650 (2017).
- Bravyi, S. & Kitaev, A. Universal quantum computation with ideal Clifford gates and noisy ancillas. *Phys. Rev. A* **71**, 022316 (2005).
- Li, Y. A magic state's fidelity can be superior to the operations that created it. *New J. Phys.* **17**, 023037 (2015).
- Brierley, S. Efficient implementation of quantum circuits with limited qubit interactions, arXiv:1507.04263

36. Farhi, E. & Goldstone, J. A quantum approximate optimization algorithm, arXiv:1411.4028
37. Peruzzo, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).
38. Wecker, D., Hastings, M. B. & Troyer, M. Progress towards practical quantum variational algorithms. *Phys. Rev. A* **92**, 042303 (2015).
39. McClean, J. R., Romero, J., Babbush, R., & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **18**, 023023 (2016).
40. Bauer, B., Wecker, D., Millis, A. J., Hastings, M. B. & Troyer, M. Hybrid quantum-classical approach to correlated materials. *Phys. Rev. X* **6**, 031045 (2016).
41. Kreula, J. M., Clark, S. R. & Jaksch, D. Non-linear quantum-classical scheme to simulate non-equilibrium strongly correlated fermionic many-body dynamics. *Sci. Rep.* **6**, 32940 (2016).
42. Kreula, J. M. et al. Few-qubit quantum-classical simulation of strongly correlated lattice fermions, EPJ Quantum. *EPJ Quant Technol* **3**, 11 (2016).
43. Li, Y., & Benjamin, S. C. Efficient variational quantum simulator incorporating active error minimisation. *Phys. Rev. X* **7**, 021050 (2017).
44. Islam, R. et al. Emergence and frustration of magnetism with variable-range interactions in a quantum simulator. *Science* **340**, 583 (2013).
45. Jurcevic, P. et al. Quasiparticle engineering and entanglement propagation in a quantum many-body system. *Nature* **511**, 202 (2014).
46. Kakuyanagi, K. et al. Observation of collective coupling between an engineered ensemble of macroscopic artificial atoms and a superconducting resonator. *Phys. Rev. Lett.* **117**, 210503 (2016).
47. Song, C. et al. 10-qubit entanglement and parallel logic operations with a superconducting circuit. *Phys. Rev. Lett.* **119**, 180511 (2017).
48. Devoret, M. H. & Schoelkopf, R. J. Superconducting circuits for quantum information: an outlook. *Science* **339**, 1169 (2013).
49. Gustavsson, S. et al. Suppressing relaxation in superconducting qubits by quasiparticle pumping. *Science* **354**, 1573 (2016).
50. Saito, S. et al. Towards realizing a quantum memory for a superconducting qubit: Storage and retrieval of quantum states. *Phys. Rev. Lett.* **111**, 107008 (2013).
51. Kolmogorov, V. Blossom V: a new implementation of a minimum cost perfect matching algorithm. *Math. Program. Comput.* **1**, 43 (2009).
52. Fowler, A. G. Accurate simulations of planar topological codes cannot use cyclic boundaries. *Phys. Rev. A* **87**, 062320 (2013).



**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