

## Network requirements for distributed quantum computation

Hugo Jacinto <sup>1,2,\*</sup>, Élie Gouzien <sup>1</sup> and Nicolas Sangouard <sup>2</sup>

<sup>1</sup>*Alice & Bob, 53 boulevard du Général Martial Valin, 75015 Paris, France*

<sup>2</sup>*Université Paris-Saclay, CNRS, CEA, Institut de Physique Théorique, 91191 Gif-sur-Yvette, France*



(Received 15 April 2025; accepted 9 January 2026; published 23 February 2026)

Physical constraints and engineering challenges, including wafer dimensions, classical control cabling, and refrigeration volumes, impose significant limitations on the scalability of quantum computing units. As a result, a modular quantum computing architecture, comprising small processors interconnected by quantum links, is emerging as a promising approach to fault-tolerant quantum computing. However, the requirements that the network must fulfill to enable distributed quantum computation remain largely unexplored. We consider an architecture tailored for qubits with nearest-neighbor physical connectivity, leveraging the surface code for error correction and enabling fault-tolerant operations through lattice surgery and magic state distillation. We propose measurement teleportation as a tool to extend lattice surgery techniques to qubits located on different computing units interconnected via Bell pairs. Through memory simulations, we build an error model for logical operations and deduce an end-to-end resource estimation of Shor's algorithm over a minimalist distributed architecture. Concretely, for a characteristic physical gate error rate of  $10^{-3}$ , a processor cycle time of 1  $\mu$ s, factoring a 2048-bit RSA integer is shown to be possible with 379 computing processors, each made with 89 781 qubits, with negligible space and time overhead with respect to a monolithic approach without parallelization, if 70 Bell pairs are available per cycle time between each processor with a fidelity exceeding 98.1%.

DOI: [10.1103/v9ln-c4v2](https://doi.org/10.1103/v9ln-c4v2)

### I. INTRODUCTION

Quantum algorithms solving practical problems typically require between  $10^7$  and  $10^{11}$  [1–3] elementary gates to be executed. Available quantum processors, with typical error rates of  $10^{-3}$  per operation [4–6], must incorporate fault-tolerant quantum error correction to implement such sequences of operations reliably [7]. These methods come, however, with a significant qubit overhead. Considering the methods of surface codes, lattice surgery, and magic state distillation [8] and an error rate of  $10^{-3}$ , it is estimated that algorithms requiring ten thousand logical qubits ultimately demand tens of millions of physical qubits [9]. Although alternative quantum error correction protocols have been recently proposed [10–12] and various physical systems are envisioned to build a fault-tolerant quantum computer, scaling in a monolithic approach will be a daunting challenge for all of them.

A solution to this scaling issue consists in using a network of small-scale processors. Achieving true scalability then reduces to the design of a processing unit with a small number of logical qubits and an interface that can be used to interconnect the different units by means of network links. Although algorithms can be split into pieces that can be connected classically by mimicking quantum operations through classical

connections [13], it comes with an exponential cost [14,15], and the generic approach uses quantum links. The basic idea, already involving experimental efforts [16–19], is to establish Bell pairs between the processing units so that qubits can be transferred or gates can be operated between processing units by means of teleportation [20,21]. Because the links are noisy and lossy, entanglement distillation protocols [22–24] were initially thought to be necessary to convert multiple low-quality Bell pairs into a small number of high-fidelity Bell pairs and guarantee high-quality operations between processors. These protocols are either simple but increase the memory error per cycle or more complex but increase the error correction cycle time [25–28]. An alternative is to distribute the computation at the level of the error correction. Recent results showed that logical qubits encoded in unrotated surface code patches in distinct processing units can be fault-tolerantly connected with noise threshold along their shared interface reaching up to 10% [29,30]. Although these results are encouraging, the central questions about how many Bell pairs are required and how faithful they need to be to execute a quantum algorithm in a distributed way with comparable space and time resources with respect to the monolithic approach remain open.

We here consider a distributed architecture where each processing unit consists of a two-dimensional (2D) grid of physical qubits. The layout involves columns of logical data qubits encoded in rotated surface code patches [31] that are separated by physical routing qubits to achieve all-to-all logical connectivity through lattice surgery. Specifically, we show how logical multiqubit gates operating on qubits located on separate units can be implemented fault-tolerantly by combining lattice surgery and a technique that we introduce as

\*Contact author: [hugo.jacinto@alice-bob.com](mailto:hugo.jacinto@alice-bob.com)

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

“measurement teleportation”: using a Bell pair to measure a nonlocal operator without further interaction between remote subsystems. From circuit-level simulations of logical memory experiments, we derive an error model for these multiqubit gates operating remotely. We then consider an execution of Shor’s factorization algorithm on this distributed architecture and provide a detailed estimation of the run-time, qubit, and Bell pair numbers needed to factor large RSA integers. Using plausible physical assumptions for processors based on superconducting qubits, namely, a characteristic physical gate error rate of  $10^{-3}$  and a surface code cycle time of  $1\ \mu\text{s}$ , we show that factoring a 2048-bit RSA integer is feasible with 379 processors, each comprising 89 781 physical qubits. This can be achieved with negligible overhead compared to using the same layout in a monolithic approach, provided that 70 Bell pairs are distributed per cycle time between each processor with a fidelity exceeding 98.1%.

## II. CIRCUIT-LEVEL SIMULATION OF THE ROTATED SURFACE CODE

We first focus on the rotated version of a distance  $d$  surface code [32]; namely, each logical qubit is encoded in a  $d \times d$  square patch, where  $d^2$  physical qubits sit on the vertices.  $d^2 - 1$  stabilizer generators, which are given by the four-qubit and two-qubit tensor products of either  $X$  or  $Z$  Pauli operators, are on the plaquette. The logical Pauli operators are strings of  $X$  and  $Z$  Pauli operators along any vertical and horizontal directions of the patch, respectively. Time is discretized into rounds, with each round corresponding to the measurement of all stabilizers, and the resulting outcomes referred to as a syndrome. In order to deal with faulty measurements, rounds are repeated  $O(d)$  times [33] before being analyzed through classical decoding to identify appropriate correction.

To evaluate accurately the performance of the rotated surface code, we perform memory experiment simulations at the circuit level. Concretely, the results of the  $Z$  stabilizers are obtained by means of CNOT gates, each controlled by the data qubits and targeting an ancilla qubit initially prepared in the  $|0\rangle$  state and finally measured in the  $Z$  basis. The CNOTs are reversed for the  $X$  stabilizer measurements, with the ancilla qubit prepared initially in the  $|+\rangle$  state and finally measured according to the  $X$  basis. The order of the CNOT gates is chosen so that four time steps are enough to extract the results of all stabilizers while avoiding hook errors and preserving the code distance [33–35]. We use the fully depolarizing noise model characterized by the parameter  $p$ , which we refer to as the physical error rate (see Appendix F of the Supplemental Material [36] for details). For the decoding, we choose a minimum-weight perfect-matching decoder [55]. We consider patches encoding either a logical  $|+\rangle$  or  $|0\rangle$  and we compute the logical error rate separately. In both cases, we retrieve a subthreshold behavior where errors are exponentially suppressed [33]. Specifically, the per-round probability to have a  $X$  logical error on  $d \times d$  square patch is extracted from a  $|0\rangle$  memory experiment with

$$P_L^X \approx \alpha \left( \frac{p}{p_{\text{th}}} \right)^{\frac{d+1}{2}}, \quad (1)$$

where  $p_{\text{th}} = (7.43 \pm 0.08) \times 10^{-3}$  and  $\alpha = (5.0 \pm 0.2) \times 10^{-2}$  (see Appendix G of the Supplemental Material [36] for details on the fitting methodology). Similarly,  $P_L^Z$  associated with  $|+\rangle$  is the same, up to uncertainty on the fitted parameters. Conservatively, we attribute a logical error rate per round  $P_L = P_L^X + P_L^Z$  to the storage of an arbitrary state.

## III. REMOTE STABILIZER MEASUREMENT CIRCUIT VIA MEASUREMENT TELEPORTATION

A key challenge when physical qubits are distributed across remote computing units is the efficient extraction of stabilizer measurement results, which is typically achieved through teleportation. This involves either teleporting a quantum state from one unit to another or teleporting a CNOT gate between an ancilla qubit and a data qubit [20]. We propose a strategy utilizing two ancilla qubits, each located in a separate unit and initially prepared in a Bell state,  $1/\sqrt{2}(|00\rangle + |11\rangle)$ . For  $Z$  stabilizer, local CNOT gates are then applied between the data qubit and the ancilla qubit within the same unit, with the ancilla serving as the target. The stabilizer measurement result is extracted by measuring the ancilla qubits in the  $Z$  basis and combining the outcomes with a modulo-2 addition (XOR) (see Appendix A of the Supplemental Material [36]). To extract the  $X$ -stabilizer result, the roles of control and target in the CNOT gates are reversed, followed by  $X$ -basis measurements of the ancilla qubits. In summary, a Bell pair enables a joint measurement between distant qubits without direct interaction. The outcome is known only after the classical communication of measurement results. This process is referred to as remote stabilizer measurement via measurement teleportation.

The next question is how many data qubits need to be connected to each ancilla in the Bell pair? At first glance, one might assume that a single vertical line of ancillary qubits could be replaced with Bell pairs. This would mean connecting two data qubits to each ancilla, requiring only  $d_X$  Bell pairs, where  $d_X$  is the distance with respect to  $X$  errors. However, such a configuration suffers from hook errors; namely, one  $X$  error on a Bell pair used to measure a  $X$  stabilizer would translate into two  $X$  errors on two data qubits located on the same vertical line of the surface code patch. This effectively halves the  $X$  distance, as detailed in Appendix B of the Supplemental Material [36]. The stabilizer measurement circuit presented in Fig. 1 provides a solution. The idea is to connect three qubits to the same ancilla in the right order so that one  $X$  error on the Bell pair would translate in the worst case into two errors on qubits belonging to the same horizontal line of the surface code patch, hence preserving the effective distance [56]. This effectively takes  $2d_X$  Bell pairs. Note that the implementation of the CNOTs can still be organized so that the syndrome can be extracted in four time steps (see Fig. 2).

## IV. CIRCUIT-LEVEL SIMULATION OF THE DISTRIBUTED ROTATED SURFACE CODE

With the syndrome extraction circuits specified, the goal is now to evaluate the performance of the distributed rotated surface code through memory experiment simulations. We consider square patches with various distances and two

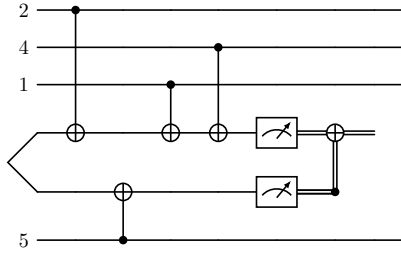


FIG. 1. Teleported Z-stabilizer measurement circuit. The first four qubits are located on the same computing unit, while the last two are situated on a separate unit. The connection between the fourth and fifth qubits indicates that they are prepared in a Bell state. The data qubit numbering follows that of Fig. 2.

columns of Bell pair ancillae aligned with  $X_L$  logical operators (see Fig. 2). The noise model remains the same as before for idle qubits, measurements, single-qubit gates, and CNOT operations. For the noise affecting the ancilla qubits, we consider two-qubit depolarizing noise on the Bell pairs, characterized by an occurrence probability  $p_{\text{Bell}}$ . This corresponds to a noisy Bell pair fidelity given by  $\mathcal{F}(\rho, |\Phi^+\rangle\langle\Phi^+|) = 1 - \frac{4}{5}p_{\text{Bell}}$ . As detailed in Appendix F of the Supplemental Material [36], this preparation noise only propagates to the data qubits located at the “seam,” i.e., between the two lines of teleported stabilizers. Decoding is still carried out using a minimum-weight perfect-matching decoder. As before, logical error rates  $P_L^X$  and  $P_L^Z$  are characterized by storing logical  $|0\rangle$  and  $|+\rangle$ , respectively. For large distances, Bell pair infidelity has a negligible effect on the Z-logical error rate, as errors on the seam only affect one data qubit of any logical Pauli Z operator. Therefore, in this section we restrict our analysis to X-logical errors.

Rough combinatorics to count error chains in the X-error matching graph of a rotated surface code under a phenomeno-

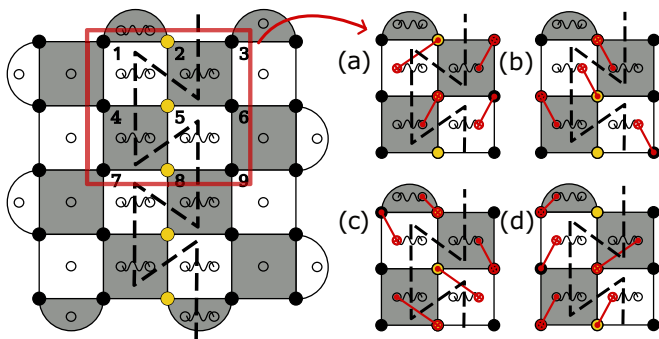


FIG. 2. Distributed rotated surface code definition. Filled dots represent data qubits, and empty dots represent ancillary qubits. Yellow filled dot represent data qubits belonging to the “seam,” the line of data qubits between the two columns of Bell pair ancillae. Empty dots connected by a sinusoidal line correspond to ancillary qubits prepared in a Bell state. The dark gray plaquettes indicate the X stabilizers over the data qubits at their edges, while the white plaquettes represent Z stabilizers. The dashed line represents the physical separation between the two computing units. The data qubit numbering helps defining the order with which the CNOTs of the extraction circuits are applied (see Fig. 1). All stabilizer can be measured in the four time steps shown in panels (a)–(d).

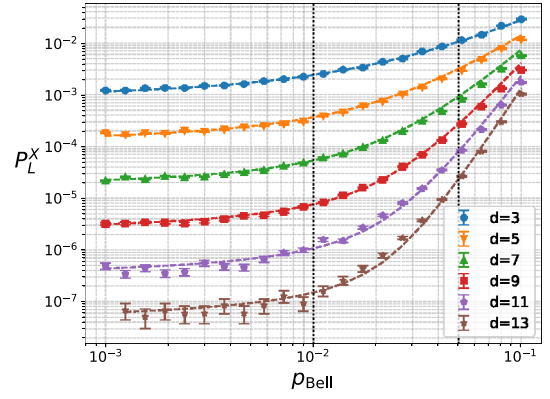


FIG. 3. X-logical error rate per round  $P_L^X$  as a function of the physical error rate on the Bell pairs  $p_{\text{Bell}}$ , under the assumption that every operation is subject to depolarizing noise with a rate  $p = 10^{-3}$ . The results are presented for various code distances, as indicated in the inset. The dashed curves are the logical error model deduced from a collective fit of the ansatz of Eq. (2). The two vertical dotted lines delimit the interval in which we use this model in the resource estimation.

logical noise model motivates the following ansatz for the X-logical error rate (see Appendix C of the Supplemental Material [36]):

$$P_L^X \approx \alpha_1 \left( \frac{p_{\text{Bell}}}{p_{\text{Bell}}^*} \right)^{\frac{d+1}{2}} + \alpha_2 \left( \frac{p}{p^*} \right)^{\frac{d+1}{2}} + \alpha_3 \sum_{1 \leq i \leq d} \left( \frac{p_{\text{Bell}}}{p_{\text{Bell}}^{**}} \right)^{\frac{i}{2}} \left( \frac{p}{p^*} \right)^{\frac{d+1-i}{2}}. \quad (2)$$

The parameter  $p$  represents the error rate for all operations except the Bell pair preparation that is characterized by  $p_{\text{Bell}}$  (see Appendix F of the Supplemental Material [36]). Threshold quantities are indicated with a superscript \*. Specifically, we define a pseudotreshold as  $p_{\text{Bell}}^{**} = p_{\text{Bell}}^* (1 + \frac{\alpha_c}{1 - \sqrt{p/p^*}})^{-2}$ .  $\alpha_1$ ,  $\alpha_2$ ,  $\alpha_3$ , and  $\alpha_c$  are constants that need to be determined through fitting. Despite coming from a simplified noise model, this ansatz appears to capture most of the underlying noise diffusion processes. Transitioning to a circuit-level noise model induces additional diagonal edges in the matching graph due to CNOT noise diffusion. The error chains counting leading to Eq. (2) still stand on this knotty graph but are characterized by different combinatorial constants resulting in modified thresholds [33].

Equation (2) is used to reproduce the memory simulation results. The fit results, covering a set of points in the  $(p_{\text{Bell}}, p, P_L^X)$  space, lead to  $\alpha_1 = 0.98 \pm 0.02$ ,  $\alpha_2 = 0.045 \pm 0.002$ ,  $\alpha_3 = 0.053 \pm 0.002$ ,  $\alpha_c = 0.21 \pm 0.02$ ,  $p^* = (7.18 \pm 0.04) \times 10^{-3}$ , and  $p_{\text{Bell}}^* = 0.298 \pm 0.011$  (see Appendix G of the Supplemental Material [36] for details of the fitting procedure). Figure 3 shows the result of this fit in the region of interest for our study corresponding to the slice with  $p = 10^{-3}$ . Interestingly, the tolerance to Bell pair infidelity is remarkably high, with a threshold  $p_{\text{Bell}}^*$  close to 30%. This can be understood by focusing on errors arising during Bell pair preparation, that is, considering the case where  $p = 0$ . As previously discussed, X errors occurring on the Bell pairs

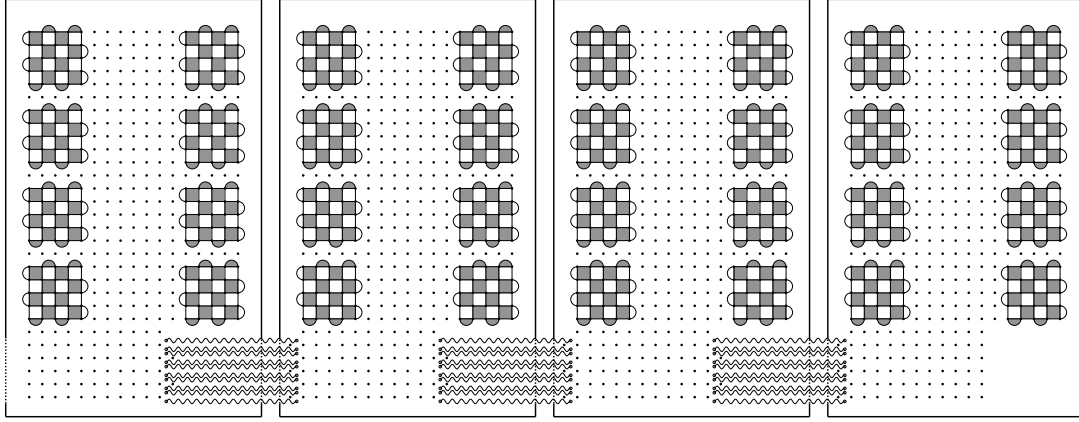


FIG. 4. Chain architecture of processors with the compact layout, here represented with four processors. Dots are data qubits. For readability, ancillae qubits have been omitted except for Bell pairs represented by two circles linked by a sinusoidal line. The different patches exhibit the location of logical qubits in the layout. The other data qubits are routing qubits used to perform lattice surgery between the represented logical qubits.

propagate to the data qubits on the seam when the Bell pair is used to measure an  $X$  stabilizer. Similarly, they induce measurement errors when the Bell pair is used to measure a  $Z$  stabilizer. As a result, error correction in this scenario reduces to the task of a repetition code handling phenomenological noise associated with an error rate  $8p_{\text{Bell}}/15$  (see Appendix F of the Supplemental Material [36] for additional details).

## V. LAYOUT OF A SIMPLE DISTRIBUTED ARCHITECTURE

The layout is designed for a fault-tolerant implementation of gates on the rotated surface code leveraging lattice surgery [57] and magic state injection. Each processor stores logical qubits into patches of distance  $d$ , grouped in two columns and interleaved with routing qubits within each computing unit (see Fig. 4).

The processors are arranged in a one-dimensional chain, successively interconnected by  $2d$  Bell pairs. This arrangement ensures protection against  $\lfloor (d-1)/2 \rfloor$  errors in a patch resulting from a lattice surgery merging of any logical qubits. This layout is particularly simple, but executing multiple lattice surgery operations in parallel is hardly possible since only one routing patch at a time can be shared between the units at the two extremities of the chain (see Appendix I of the Supplemental Material [36] for further details). Additionally, we do not consider a dynamical layout where logical qubits are relocated during the algorithm execution as done in Ref. [9], hence enhancing parallelization while reducing the total number of routing qubits. Nevertheless, our approach provides valuable insights into the space and time overhead of the distributed setting in the presence of Bell pair errors.

## VI. RESOURCE ESTIMATION TO FACTOR 2048-BIT RSA INTEGERS

We here report on a comparative resource estimation to factor a 2048-bit RSA integer using Ekerå and Håstad's [58] version of Shor's algorithm [59] with a monolithic and a distributed architecture. We begin by decomposing Shor's

algorithm into a sequence of subroutines outlined in Appendix K of the Supplemental Material [36]. The number of gates in each subroutine is deduced from their implementations using a Toffoli-based gate set. This provides the total number of logical qubits and gates required for factoring a 2048-bit RSA integer. Using explicit constructions of CNOT,  $CX \cdots X$  (multiple not gates controlled by one qubit) and Toffoli gates at the logical level (see Appendixes I and J of the Supplemental Material [36]), we then estimate their error rates and the time it takes to execute them as a function of the code distance. For each code distance, the algorithm execution time is deduced by dividing the sum of the implementation time of all the operations (gates, measurements, and initializations) by the overall success probability. The space cost is obtained from the number of physical qubits required by the layout to host the required number of logical qubits, magic state factories, and routing areas for a given number of computing units. We choose to take the smallest possible processors while ensuring that a magic states factory would fit inside, which also fixes the number of processors in the architecture. The time and space overheads are deduced by choosing the distance, the subroutine parameters (detailed in Appendix K of the Supplemental Material [36]), and the magic state factory minimizing the volume.

For CNOT gates, we consider an approach with an auxiliary qubit and two joint measurements performed via lattice surgery. Since each joint measurement takes  $\sim d$  rounds, the CNOT gate duration via lattice surgery is *a priori*  $2dt_c$  [8,31], where  $t_c$  is the cycle time, namely, the duration of one syndrome extraction cycle, which we assume to be  $1 \mu\text{s}$ . However, for some specific logical qubit locations, it happens that accessing logical operators for the second joint measurement requires additional  $d$  cycles to move patches. Therefore, we have made the conservative assumption of attributing a time cost of  $4dt_c$  to the CNOT gate (see Appendix I of the Supplemental Material [36]). The probability of an error-free implementation of such a CNOT is evaluated by considering that neither the control, nor the target, nor the other idling data qubits should experience any errors over  $4d$  rounds while the error rate per round is given by Eq. (1). If a CNOT is

TABLE I. Resource estimation table. In the distributed setting, the number of processors is written as # processors storing logical qubits for computation + # processors storing a magic state factory. The number of physical qubits per processor announced in the table is the one of processors storing logical qubits. For more information about the magic state factories, see Table IV of Appendix J of the Supplemental Material [36]. A reading guide of this table is available in Appendix K of the Supplemental Material [36].

|                                 | Monolithic<br>$p = 0.1\%$              | Distributed                            |                                        |                                        |                                        |
|---------------------------------|----------------------------------------|----------------------------------------|----------------------------------------|----------------------------------------|----------------------------------------|
|                                 |                                        | $p = 0.1\%$<br>$p_{\text{Bell}} = 0\%$ | $p = 0.1\%$<br>$p_{\text{Bell}} = 2\%$ | $p = 0.1\%$<br>$p_{\text{Bell}} = 3\%$ | $p = 0.1\%$<br>$p_{\text{Bell}} = 4\%$ |
| Distance $d$                    | 35                                     | 35                                     | 35                                     | 39                                     | 47                                     |
| # Logical qubits per processor  | 8280                                   | 22                                     | 22                                     | 18                                     | 12                                     |
| # Physical qubits per processor | 32 040 090                             | 89 781                                 | 89 781                                 | 91 917                                 | 91 389                                 |
| # Processors                    | 1                                      | 377+2                                  | 377+2                                  | 460+2                                  | 690+2                                  |
| Total # of physical qubits      | 32 040 090                             | 34 018 414                             | 34 018 414                             | 42 455 161                             | 63 218 103                             |
| Estimated duration              | 20 days 9 hours                        | 20 days 13 hours                       | 21 days 12 hours                       | 25 days 20 hours                       | 30 days 13 hours                       |
| Magic state factory             | $p_{\text{out}} = 2.1 \times 10^{-13}$ | $p_{\text{out}} = 2.1 \times 10^{-14}$ | $p_{\text{out}} = 2.1 \times 10^{-14}$ | $p_{\text{out}} = 2.1 \times 10^{-14}$ | $p_{\text{out}} = 9.9 \times 10^{-14}$ |
| Space overhead                  | —                                      | Reference                              | 0%                                     | 25%                                    | 86%                                    |
| Time overhead                   | —                                      | Reference                              | 5%                                     | 26%                                    | 49%                                    |

implemented between qubits belonging to different computing units, an error-free implementation also imposes that no error occurs on the auxiliary-control or auxiliary-target merged patches during  $d$  rounds while the error rate per round, of the form Eq. (2), depends on the number of seams separating the two computing units. Note that both Eqs. (1) and (2) are models relying on memory experiment simulations, so that timelike error effects are not captured. We always consider the worst-case scenario in which the routing patch crosses the entire chain of computing units (see Appendix I of the Supplemental Material [36] for the details on the noise model of CNOT gates and extension to  $CX \cdots X$ ).

For the Toffoli gate, we use a two-level distillation factory combining protocols consuming 15 noisy  $T$  states to produce one cleaner  $T$  state, which is then used as input of an 8  $T$  to 1 CCZ protocol. The space cost of this factory is evaluated for different inner distances (see Ref. [60] and Appendix J1 of the Supplemental Material [36]). We use two factories, so that CCZ states can be both injected and prepared in parallel as the injection time is longer than the duration of the magic state preparation. This choice is further detailed in Appendix J 2 of the Supplemental Material [36], along with the logical error model of the Toffoli gate, mainly based on the CNOT logical error model.

The result of our resource estimation is presented in Table I (see also Appendix K of the Supplemental Material [36]).

For an error rate  $p = 10^{-3}$ , our reference monolithic approach requires over 32 million qubits to factor a 2048-bit RSA integer, with a computation time exceeding 20 days. Resource and run-time are essentially unchanged when the algorithm is executed in a distributed way without error on the Bell pairs. The slight additional space cost is due to the presence of additional routing qubits in the distributed architecture (see Appendix H of the Supplemental Material [36]). The optimal distance remains unchanged for up to  $p_{\text{Bell}} = 2.42\%$  corresponding to  $\mathcal{F}(\rho, |\Phi^+\rangle\langle\Phi^+|) \geq 98.1\%$ . However, since the time cost is defined as the algorithm's expected run-time toward its success, there is a slight overhead due to the increased failure probability of each logical CNOT gate in the circuit. The increase in qubit cost for  $p_{\text{Bell}}$  around 3% is due

to a change in distance, leading to each logical qubit requiring more physical qubits.

## VII. CONCLUSION

Through circuit-level simulations of memory experiments and given a minimalist architecture, we have identified network requirements that enable the execution of Shor's algorithm in a distributed platform using lattice surgery with little or no space-time overhead with respect to the monolithic approach. Concretely, with a chainlike architecture made of 379 processors with 89 781 qubits each, factoring 2048-RSA integers is shown possible in less than 22 days for a physical gate error rate of  $10^{-3}$  if  $2d = 70$  Bell pairs are available per cycle time (1  $\mu\text{s}$ ) between each nearest-neighbor processor, with a fidelity exceeding 98.1%. The simultaneous requirements on fidelity and rate have not yet been achieved experimentally [16–19], but our findings provide valuable guidance on how network performance must improve in order to scale up quantum computing. As a perspective, if inter-cryostat connections do not reach the required fidelity, one can envision employing entanglement distillation to target fidelities of about 99% starting from initial values around 90%. Achieving this would likely require only a small number of physical qubits compared to the size of each processor.

## ACKNOWLEDGMENTS

We thank Jérémie Guillaud, Diego Ruiz, and Christophe Vuillot for insightful discussions. This work was partially supported by the French National program Programme d'investissement d'avenir, IRT Nanoelec, with Reference No. ANR-10-AIRT-05. This research is partially funded by the French State through France 2030.

## DATA AVAILABILITY

The data that support the findings of this article are openly available [61].

- [1] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoefler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vashchillo, Assessing requirements to scale to practical quantum advantage, [arXiv:2211.07629](#).
- [2] A. M. Dalzell, S. McArdle, M. Berta, P. Bienias, C.-F. Chen, A. Gilyén, C. T. Hann, M. J. Kastoryano, E. T. Khabiboulline, A. Kubica, G. Salton, S. Wang, and F. G. S. L. Brandão, *Quantum Algorithms: A Survey of Applications and End-to-end Complexities* (Cambridge University Press, 2025).
- [3] T. L. Scholten, C. J. Williams, D. Moody, M. Mosca, W. Hurley, W. J. Zeng, M. Troyer, and J. M. Gambetta, Assessing the benefits and risks of quantum computers, [arXiv:2401.16317](#).
- [4] S. A. Moses, *et al.*, A race-track trapped-ion quantum processor, *Phys. Rev. X* **13**, 041052 (2023).
- [5] R. Acharya, *et al.*, Suppressing quantum errors by scaling a surface code logical qubit, *Nature (London)* **614**, 676 (2023).
- [6] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, Evidence for the utility of quantum computing before fault tolerance, *Nature (London)* **618**, 500 (2023).
- [7] E. T. Campbell, B. M. Terhal, and C. Vuillot, Roads towards fault-tolerant universal quantum computation, *Nature (London)* **549**, 172 (2017).
- [8] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, *Quantum* **3**, 128 (2019).
- [9] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, *Quantum* **5**, 433 (2021).
- [10] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, *Nature (London)* **627**, 778 (2024).
- [11] S. Gu, Y. Vaknin, A. Retzker, and A. Kubica, Optimizing quantum error correction protocols with erasure qubits, *PRX Quantum* **6**, 040354 (2025).
- [12] C. Gidney, M. Newman, P. Brooks, and C. Jones, Yoked surface codes, *Nat Commun* **16**, 4498 (2025).
- [13] A. C. Vazquez, C. Tornow, D. Ristè, S. Woerner, M. Takita, and D. J. Egger, Combining quantum processors with real-time classical communication, *Nature (London)* **636**, 75 (2024).
- [14] M. Jing, C. Zhu, and X. Wang, Circuit knitting faces exponential sampling overhead scaling bounded by entanglement cost, *Phys. Rev. A* **111**, 012433 (2025).
- [15] C. Piveteau and D. Sutter, Circuit knitting with classical communication, *IEEE Trans. Inf. Theory* **70**, 2734 (2024).
- [16] K. Heya, T. Phung, M. Malekakhlagh, R. Steiner, M. Turchetti, W. Shanks, J. Mamin, W.-S. Lu, Y. P. Kandel, N. Sundaresan, and J. Orcutt, Randomized benchmarking of a high-fidelity remote CNOT gate over a meter-scale microwave interconnect, *Phys. Rev. Lett.* **135**, 200801 (2025).
- [17] S. Storz, J. Schär, A. Kulikov, P. Magnard, P. Kurpiers, J. Lütolf, T. Walter, A. Copetudo, K. Reuer, A. Akin, *et al.*, Loophole-free Bell inequality violation with superconducting circuits, *Nature (London)* **617**, 265 (2023).
- [18] H. Yan, Y. Zhong, H.-S. Chang, A. Bienfait, M.-H. Chou, C. R. Conner, E. Dumur, J. Grebel, R. G. Povey, and A. N. Cleland, Entanglement purification and protection in a superconducting quantum network, *Phys. Rev. Lett.* **128**, 080504 (2022).
- [19] S. Storz, A. Kulikov, J. D. Schär, V. Barizien, X. Valcarce, F. Berterotière, N. Sangouard, J.-D. Bancal, and A. Wallraff, Complete self-testing of a system of remote superconducting qubits, *Phys. Rev. Lett.* **135**, 030801 (2025).
- [20] M. Caleffi, M. Amoretti, D. Ferrari, J. Illiano, A. Manzalini, and A. S. Cacciapuoti, Distributed quantum computing: A survey, *Comput. Networks* **254**, 110672 (2024).
- [21] Q. Xu, A. Seif, H. Yan, N. Mannucci, B. O. Sane, R. Van Meter, A. N. Cleland, and L. Jiang, Distributed quantum error correction for chip-level catastrophic errors, *Phys. Rev. Lett.* **129**, 240502 (2022).
- [22] W. Dür and H.-J. Briegel, Entanglement purification for quantum computation, *Phys. Rev. Lett.* **90**, 067901 (2003).
- [23] E. T. Campbell, Distributed quantum-information processing with minimal local resources, *Phys. Rev. A* **76**, 040302(R) (2007).
- [24] S. Krastanov, V. V. Albert, and L. Jiang, Optimized entanglement purification, *Quantum* **3**, 123 (2019).
- [25] L. Jiang, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, Distributed quantum computation based on small quantum registers, *Phys. Rev. A* **76**, 062323 (2007).
- [26] K. Fujii, T. Yamamoto, M. Koashi, and N. Imoto, A distributed architecture for scalable quantum computation with realistically noisy devices, [arXiv:1202.6588](#).
- [27] N. Nickerson, Y. Li, and S. Benjamin, Topological quantum computing with a very noisy network and local error rates approaching one percent, *Nat. Commun.* **4**, 1756 (2013).
- [28] N. H. Nickerson, J. F. Fitzsimons, and S. C. Benjamin, Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links, *Phys. Rev. X* **4**, 041041 (2014).
- [29] J. Ramette, J. Sinclair, N. P. Breuckmann, and V. Vuletić, Fault-tolerant connection of error-corrected qubits with noisy links, *npj Quantum Inf* **10**, 58 (2024).
- [30] J. Sinclair, J. Ramette, B. Grinkemeyer, D. Bluvstein, M. Lukin, and V. Vuletić, Fault-tolerant optical interconnects for neutral-atom arrays, [arXiv:2408.08955](#).
- [31] A. G. Fowler and C. Gidney, Low overhead quantum computation using lattice surgery, [arXiv:1808.06709](#).
- [32] A. R. O'Rourke and S. Devitt, Compare the pair: Rotated vs. unrotated surface codes at equal logical error rates, *Phys. Rev. Res.* **7**, 033074 (2025).
- [33] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* **43**, 4452 (2002).
- [34] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, *Phys. Rev. A* **86**, 032324 (2012).
- [35] Y. Tomita and K. M. Svore, Low-distance surface codes under realistic quantum noise, *Phys. Rev. A* **90**, 062320 (2014).
- [36] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/v9ln-c4v2> for all the Appendixes mentioned in the main text, which cites additionally [37–54].
- [37] C. Gidney, Stim: a fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).
- [38] A. M. Stephens, Fault-tolerant thresholds for quantum error correction with the surface code, *Phys. Rev. A* **89**, 022321 (2014).
- [39] M. Newville, T. Stensitzki, D. B. Allen, and A. Ingargiola, Lmfit: Non-linear least-square minimization and curve-fitting for python, Astrophysics Source Code Library (2014), doi:10.5281/zenodo.12785036.
- [40] C. Guinn, S. Stein, E. Tureci, G. Avis, C. Liu, S. Krastanov, A. A. Houck, and A. Li, Co-designed superconducting architecture

- for lattice surgery of surface codes with quantum interface routing card, [arXiv:2312.01246](#) [quant-ph].
- [41] C. Gidney and A. G. Fowler, Flexible layout of surface code computations using autoccz states, [arXiv:1905.08916](#) [quant-ph].
  - [42] D. B. Tan, M. Y. Niu, and C. Gidney, A sat scalpel for lattice surgery: Representation and synthesis of subroutines for surface-code fault-tolerant quantum computing, in *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)* (IEEE, New York, 2024), pp. 325–339.
  - [43] C. Chamberland and E. T. Campbell, Universal quantum computing with twist-free and temporally encoded lattice surgery, [PRX Quantum 3, 010331 \(2022\)](#).
  - [44] C. Chamberland and E. T. Campbell, Circuit-level protocol and analysis for twist-based lattice surgery, [Phys. Rev. Res. 4, 023090 \(2022\)](#).
  - [45] D. Litinski and F. von Oppen, Lattice surgery with a twist: Simplifying clifford gates of surface codes, [Quantum 2, 62 \(2018\)](#).
  - [46] C. Gidney, Inplace access to the surface code y basis, [Quantum 8, 1310 \(2024\)](#).
  - [47] E. Gouzien, D. Ruiz, F.-M. Le Régent, J. Guillaud, and N. Sangouard, Performance analysis of a repetition cat code architecture: Computing 256-bit elliptic curve logarithm in 9 hours with 126 133 cat qubits, [Phys. Rev. Lett. 131, 040602 \(2023\)](#).
  - [48] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. Mc- Clean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear complexity, [Phys. Rev. X 8, 041015 \(2018\)](#).
  - [49] B. Domokos, Áron Márton, and J. K. Asbóth, Characterization of errors in a cnot between surface code patches, [arXiv:2405.05337](#) [quant-ph].
  - [50] C. Gidney, Stability experiments: The overlooked dual of memory experiments, [Quantum 6, 786 \(2022\)](#).
  - [51] S. Bravyi and A. Kitaev, Universal quantum computation with ideal clifford gates and noisy ancillas, [Phys. Rev. A 71, 022316 \(2005\)](#).
  - [52] G. P. Gehér, C. McLauchlan, E. T. Campbell, A. E. Moylett, and O. Crawford, Error-corrected hadamard gate simulated at the circuit level, [Quantum 8, 1394 \(2024\)](#).
  - [53] C. Gidney, Windowed quantum arithmetic, [arXiv:1905.07682](#) [quant-ph].
  - [54] C. Gidney, Approximate encoded permutations and piecewise quantum adders, [arXiv:1905.08488](#) [quant-ph].
  - [55] O. Higgott, and C. Gidney, Sparse {B}lossom: correcting a million errors per core second with minimum-weight matching, [Quantum 9, 1600 \(2025\)](#).
  - [56] M. A. Shalby, R. Wang, D. Sedov, and L. P. Pryadko, Optimized noise-resilient surface code teleportation interfaces, [Phys. Rev. A 112, L020403 \(2025\)](#).
  - [57] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, [New J. Phys. 14, 123011 \(2012\)](#).
  - [58] M. Ekerå and J. Håstad, Quantum algorithms for computing short discrete logarithms and factoring RSA integers, in *Post-Quantum Cryptography, Lecture Notes in Computer Science* (Springer International Publishing, Cham, 2017), Vol. 10346, p. 347.
  - [59] P. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (1994), pp. 124–134.
  - [60] D. Litinski, Magic state distillation: Not as costly as you think, [Quantum 3, 205 \(2019\)](#).
  - [61] [https://github.com/HugoJ1/shor\\_multi\\_chip](https://github.com/HugoJ1/shor_multi_chip).