Entanglement boosting: Low-volume logical Bell pair preparation for distributed fault-tolerant quantum computation

Shinichi Sunami,[1,2,*] Yutaka Hirano,[1,†] Toshihide Hinokuma,[1] and Hayata Yamasaki[1,3,‡]

[1]*Nanofiber Quantum Technologies, Inc. (NanoQT), 1-22-3 Nishiwaseda, Shinjuku-ku, Tokyo 169-0051, Japan.*
[2]*Clarendon Laboratory, University of Oxford, Oxford OX1 3PU, United Kingdom*
[3]*Department of Computer Science, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*

Distributed architecture is a promising route to scaling fault-tolerant quantum computing (FTQC) beyond the inherent limitations of single processors, for which high-fidelity logical Bell pairs need to be prepared from many noisy physical Bell pairs with high efficiency. For practical implementation of distributed FTQC, logical Bell pair preparation must be designed not only for efficient Bell pair consumption but also for the spacetime volume of the protocol; however, entanglement distillation protocols have primarily focused on minimizing the consumption of Bell pairs, often resulting in protocols that require a substantial number of local operations. A key challenge is to find an appropriate balance between these two contrasting features. To resolve this issue, we introduce a metric for characterizing the practical cost of preparing high-fidelity logical Bell pairs, *link-limited volume* (LLV), which is a circuit-volume metric incorporating, in a single quantity, both the cost of physical Bell pair consumption and the volume associated with local operations. Guided by this metric, we propose the *entanglement boosting* protocol that achieves efficient preparation of logical Bell pairs encoded in rotated surface code, with LLV reduced by orders of magnitude compared to prior state-of-the-art methods. In this protocol, paralleling recent advances in magic state cultivation, we employ soft-information decoders and postselection to suppress the logical error rates of Bell pairs to practical levels in the order of 10^{-10} from fewer than 100 noisy physical Bell pairs, while all local operations are implementable within a spatial region of a single surface code patch with two-dimensional connectivity. To further augment the entanglement boosting, we also present a pipelined implementation of entanglement distillation using high-rate quantum error-correcting codes, enabling arbitrarily low logical error rates while also maintaining physically efficient implementations. These results pave the way for the practical implementation of distributed FTQC, reinforcing the benefits of fast interconnect technologies and serving as a guiding principle for the efficient design of protocols and devices.

I. INTRODUCTION

High-fidelity, maximally entangled qubit pairs (Bell pairs) prepared over a network are a fundamental resource for distributed quantum technologies, including quantum communication [1, 2], blind quantum computing [3], quantum sensing [4], and distributed fault-tolerant quantum computing (FTQC) [5, 6]. In particular, distributed FTQC requires stringent fidelity for the Bell pairs, which must be encoded in quantum error-correcting codes to ensure faithful logical operations. Since the physical Bell pairs generated over the network are generally noisy, it is necessary to execute protocols that turn noisy Bell pairs into high-fidelity ones.

In the standard setting for information theory and entanglement theory, *entanglement distillation* refers to the process of converting many noisy Bell pairs into a smaller number of nearly maximally entangled states using noiseless local operations and classical communication (LOCC) [7]. In this framework, LOCC are free, so the task is to characterize which noisy states are distillable, and at what rate, under this idealized model [8]. Canonical examples include recurrence protocols [7, 9] and one-way hashing protocols [10], which established the foundations of entanglement distillation and highlighted the close relationship with quantum error-correcting codes [10, 11].

In contrast, in the context of distributed FTQC, the LOCC assumptions break down: local operations are noisy, and therefore, error correction must be performed explicitly to implement protocols for obtaining high-fidelity remote Bell pairs. In practice, these costs can be comparable to, or even exceed, the costs associated with network usage, especially in hardware platforms enabling high-speed remote Bell pair generation methods, such as neutral atoms and trapped ions [6, 12–15]. A crucial metric for local operation cost is the spacetime volume, i.e., the number of physical qubits involved in a protocol multiplied by the duration of the computation, with a typical time unit being the number of syndrome extraction (SE) cycles, where a cycle corresponds to measuring all syndrome checks of a quantum error-correcting code once [16–18]. This volume, in units of *qubit-cycles*, comprises operations needed for local error correction and logical-level circuit execution on encoded qubits, typically involving hundreds to thousands of physical qubits and tens to hundreds of SE cycles, even for small-scale logical circuits for entanglement distillation. Therefore, protocols must be judged not only by how few Bell pairs they consume, but also by the spacetime volume for local operations.

The reduction of spacetime volume has been the focus of the development of magic state distillation protocols in recent years [16, 19], leading to rapid improvements in protocols for preparing logical magic states, such as better layout design and multi-stage strategies [20–24]. Critical differences in the operational constraints between logical magic state preparation

* shinichi.sunami@nano-qt.com; equal contribution
† Equal contribution
‡ hayata.yamasaki@nano-qt.com

and logical Bell pair preparation necessitate distinct metrics and strategies. The physical magic states are nearly free: these can be obtained by local single-qubit gates with no latency at any location. In contrast, the generation of remote physical Bell pairs is inherently different from local operations, achievable through only sequential generation at a finite throughput, which limits their usage. While it is possible to buffer the required Bell pairs before starting a protocol, the associated space and time costs for the buffering cannot be ignored in practice. This favors protocols with a careful balance between the Bell pair consumption, a standard metric for the LOCC framework, and the spacetime volume for the protocol, a metric used for magic state preparation; however, finding such a balance is challenging without a guiding principle.

There are currently two common approaches to preparing high-fidelity logical Bell pairs, each of which primarily focuses on only one of the two contrasting desired properties discussed above. First, the physical-to-logical approach utilizes the physical Bell pairs directly in the protocol, such as for syndrome extraction across the network in lattice-surgery-based protocols [25, 26], as well as through direct projections of many physical Bell pairs onto a code space via syndrome extraction [27–29]. These protocols require hundreds to thousands of physical Bell pairs to achieve the high fidelities required for large-scale FTQC. Second, the injection-and-distillation approach begins by first injecting physical Bell pairs onto logical qubits and executing the distillation protocols with logical gates [6]. This includes LOCC protocols implementable by logical gates, such as recurrence protocols [7, 9] and concatenated stabilizer-code distillation [30]. While the physical-to-logical protocols are implementable with relatively small local circuit volume, the required number of Bell pairs is generally large, resulting in a substantial requirement for network performance. On the other hand, while the injection-and-distillation protocols are efficient in Bell pair usage, the local circuit volume is significant due to the inherent overhead of fault-tolerant gates, thus potentially diminishing the benefits of modular scaling.

In this work, we develop an efficient protocol to turn noisy physical Bell pairs into high-fidelity logical Bell pairs encoded in a rotated surface code [31]. As a guiding metric for logical Bell pair generation, we propose a *link-limited volume* (LLV) to quantify the overall cost of high-fidelity logical Bell pair preparation, in the presence of noisy local operations and the finite speed of physical remote Bell pair generation. This is defined in qubit-cycles [16] and accounts for both the spacetime volume of local operations and the buffering cost imposed by the finite throughput of physical Bell pair generation. By expressing the requirements for networks and local operations within a single quantity, LLV provides a common metric that enables efficient optimization for the realistic implementation of logical Bell pair preparation for distributed FTQC.

Guided by this metric, we propose *entanglement boosting*, a physical-to-logical Bell pair preparation protocol for the rotated surface code. This protocol combines (i) a code projection step that projects the physical Bell pairs onto the logical code space of a surface code with a small code distance, (ii) code expansion to the target surface code distance, and (iii) postselection based on soft-information decoding. The techniques are partly inspired by the magic state cultivation approach for efficient logical magic state preparation [21] while tailored for logical Bell pair preparation. This allows for both efficient use of physical Bell pairs and a small circuit volume while maintaining scalable error suppression, thus achieving a large reduction in the LLV. We also design an efficient pipelined approach for implementing entanglement distillation circuits using logical gates and parallel logical-qubit reconfiguration, complementing the boosting stage. The combined approach improves the yield with additional local operations, achieving a lower LLV in regimes with limited network throughput and enabling arbitrary logical error suppression.

These results are key to the scalable realization of FTQC, where modular architecture is expected to play a central role. By formulating a unified metric that quantifies both network throughput and local circuit volume, our framework provides a principled basis for optimizing protocols for logical Bell pair preparation under realistic hardware constraints. The protocols proposed in this work demonstrate low-volume implementations with a flexible design adaptable to a wide range of interconnect speeds and fidelity. Beyond quantum computation, our theoretical framework is broadly applicable to distributed information processing settings based on remote entanglement generation with local fault-tolerant operations, such as device-independent quantum key distribution [32] and the communication-based demonstration of energy-consumption advantage of quantum computation [33], offering a systematic recipe for scalable and efficient distributed quantum technologies.

This article is organized as follows. In Sec. II, we cover preliminaries, including stabilizer codes and stabilizer entanglement distillation. In Sec. III, we describe the setup for the distributed FTQC assumed in this work and introduce LLV. We present the entanglement boosting protocol in Sec. IV, together with circuit-level numerical simulation results. We then discuss the implementation of pipelined entanglement distillation with parallel logical-qubit reconfiguration in Sec. V. In Sec. VI, we conclude our results and provide an outlook.

II. PRELIMINARIES

Here, we first summarize the basic notations for quantum error correction relevant to this work in Sec. II A, stabilizer entanglement distillation in Sec. II B, and our assumptions on the noise model in Sec. II C.

A. Stabilizer codes

The single-qubit Hilbert space $\mathbb{C}^2$ is spanned by the computational basis $\{|0\rangle, |1\rangle\}$, where $|\pm\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. On this space, the Pauli operators are defined as $X := |0\rangle\langle 1| + |1\rangle\langle 0|$, $Y := \mathrm{i}|1\rangle\langle 0| - \mathrm{i}|0\rangle\langle 1|$, $Z := |0\rangle\langle 0| - |1\rangle\langle 1|$, $I := |0\rangle\langle 0| + |1\rangle\langle 1|$, where $\mathrm{i} = \sqrt{-1}$. The n-qubit Pauli group $\mathcal{P}_n$ consists of tensor

products of single-qubit Pauli operators up to a global phase $\alpha \in \{\pm 1, \pm i\}$.

A *stabilizer* $\mathcal{S}$ is an Abelian subgroup of $\mathcal{P}_n$ that does not include $-I^{\otimes n}$. The corresponding *stabilizer code* is the joint $+1$ eigenspace of $\mathcal{S}$:

$$C = \{|\psi\rangle \in (\mathbb{C}^2)^{\otimes n} \mid s|\psi\rangle = |\psi\rangle,\ \forall s \in \mathcal{S}\}.$$

If $\mathcal{S}$ is generated by $n-k$ independent stabilizer generators, then C encodes k logical qubits, i.e., $\dim C = 2^k$. Let $\mathcal{N}(\mathcal{S}) \coloneqq \{P \in \mathcal{P}_n \mid P\mathcal{S}P^{-1} = \mathcal{S}\}$ be the normalizer of $\mathcal{S}$, where $P\mathcal{S}P^{-1} \coloneqq \{PsP^{-1} \mid s \in \mathcal{S}\}$. Let $\mathcal{Z} \coloneqq \{\alpha I^{\otimes n} \mid \alpha \in \{\pm 1, \pm i\}\}$ be the center of $\mathcal{P}_n$, the set of elements that commute with all elements in $\mathcal{P}_n$. The logical Pauli group is the quotient group $\mathcal{L} \coloneqq \mathcal{N}(\mathcal{S})/(\mathcal{S} \cdot \mathcal{Z})$, where $\mathcal{S} \cdot \mathcal{Z} = \{sz \mid s \in \mathcal{S}, z \in \mathcal{Z}\}$. An operator $\mathcal{L} \in \mathcal{N}(\mathcal{S})$ is called a logical operator if we view it modulo $\mathcal{S} \cdot \mathcal{Z}$, i.e., via its coset $[L] \in \mathcal{L}$. It is called a *nontrivial logical operator* if and only if its coset $[L] \in \mathcal{L}$ is nontrivial, i.e., $[L] \neq [I]$. The *weight* $|P|$ of $P \in \mathcal{P}_n$ is the number of qubits on which P acts nontrivially, i.e. $|P|$ counts the tensor factors of $P = P_1 \otimes \cdots \otimes P_n$ that are X, Y, or Z, rather than the identity. The *distance* d of a stabilizer code C is the minimum weight of all the nontrivial logical operators. The logical Pauli operators $\overline{X}_i$ and $\overline{Z}_i$ $(i = 1, \ldots, k)$ are representatives of cosets in $L = \mathcal{N}(\mathcal{S})/(\mathcal{S} \cdot \mathcal{Z})$ that act nontrivially on the i-th logical qubit. They commute with all $s \in S$ and satisfy $[\overline{X}_i, \overline{Z}_j] = 0$ $(i \neq j)$ and $\{\overline{X}_i, \overline{Z}_i\} = 0$, where $[A, B] \coloneqq AB - BA$, and $\{A, B\} \coloneqq AB + BA$.

A stabilizer code encoding k logical qubits into n physical qubits with distance d is called an $[[n, k, d]]$ code. A particularly important family is the Calderbank-Shor-Steane (CSS) codes [34, 35], where stabilizer generators are tensor products of only I and X (X-type) or only I and Z (Z-type). Quantum error correction involves two processes: *syndrome extraction* and *decoding*. Syndrome extraction measures stabilizer generators, referred to as *stabilizer checks*, to obtain error information, which is called the *error syndrome*. Decoding is a classical computation process that, given an error syndrome, returns a recovery operator consistent with the error syndrome.

B. Entanglement distillation with stabilizer codes

An $[[n, k, d]]$ stabilizer code C can be used to construct entanglement distillation protocols [27, 36, 37]. Here, we outline two approaches used in this work, illustrated in Fig. 1.

The first approach (Fig. 1a) starts with n Bell pairs and prepares k logical Bell pairs encoded in stabilizer code C by measuring stabilizer checks of the code in nodes A and B [27]. Specifically, stabilizer measurements are performed by $n-k$ auxiliary qubits in each node, and the measurement results are sent between A and B, where B combines the results by taking the parities of the corresponding stabilizer values. The measurement results can be used to either postselect the output states via two-way communication (error detection), or to perform correcting operations identified by classical decoding (error correction) via one-way communication. The stabilizer measurements and error detection (correction) on the joint

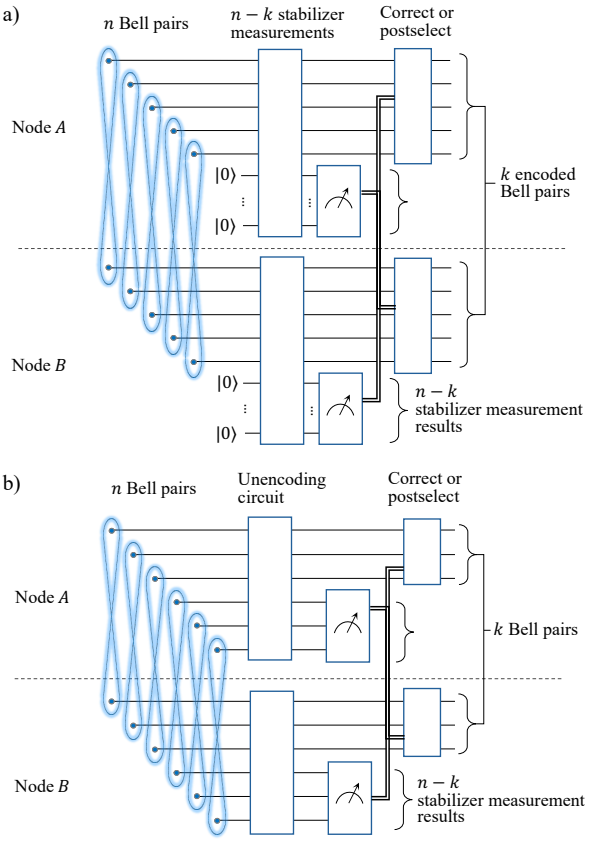

FIG. 1. Two entanglement distillation protocols based on $[[n, k, d]]$ stabilizer code C used in this work. a) Auxiliary-qubit-assisted, projection-based preparation of logical Bell pairs from physical Bell pairs. Here, stabilizer checks of n Bell pairs are measured by $n-k$ auxiliary qubits, allowing either postselection or error correction to achieve k high-fidelity logical Bell pairs encoded in code C. b) A decoding-based approach, which combines the protocol in a) and decoding of the encoded state, to result in k output Bell pairs. The stabilizer measurements are performed by the measurement of $n-k$ qubits that do not constitute the output.

checks leave the state stabilized by $\mathcal{S}' = \{s \otimes s, s \in \mathcal{S}\}$. The effect of the above procedure on the logical qubits can be identified by the fact that the protocol begins with n physical Bell pairs. The i-th pair (for $i = 1, ..., n$) is stabilized by $X_{i,A} \otimes X_{i,B}$ and $Z_{i,A} \otimes Z_{i,B}$, where $X_{i,A(B)}$ and $Z_{i,A(B)}$ are the corresponding Pauli operators on the i-th physical qubit in node A(B). For X and Z logical operators for k logical qubits, $\mathcal{L}_X = \{\overline{X}_1, ...\overline{X}_k\}$, $\mathcal{L}_Z = \{\overline{Z}_1, ...\overline{Z}_k\}$, the resulting logical qubits are stabilized by $\overline{X}_{i,A} \otimes \overline{X}_{i,B}$ and $\overline{Z}_{i,A} \otimes \overline{Z}_{i,B}$, for $i = 1$ to k; therefore, the resulting states are logical Bell pairs [29].

The second approach does not leave the resulting state encoded in the chosen code. Let U_{enc} denote the unitary encoding map that prepares a codeword $|\bar{\psi}\rangle = U_{\text{enc}}(|\psi\rangle \otimes |0\rangle^{\otimes(n-k)})$ for k-qubit state $|\psi\rangle$. The inverse operation U_{enc}^{-1}, referred to as the *unencoding unitary*, acts as $U_{\text{enc}}^{-1}|\bar{\psi}\rangle = |\psi\rangle \otimes |\text{syndrome}\rangle$, where the last $n-k$ qubit states, denoted as syndrome qubits, can be measured in the computational basis to obtain the stabilizer values. By appending an unencoding unitary to the

protocol described above (Fig. 1a), we obtain k unencoded Bell pairs as a result of successful execution. This can be simplified to only an application of U_{enc}^{-1} to n Bell pairs [11], which results in k output qubits along with $n-k$ qubits that can be measured to provide the stabilizer checks needed for error correction or detection (Fig. 1b).

C. Noise model

In this work, we perform circuit-level simulations to numerically evaluate the logical Bell pair preparation protocols. We assume that physical operations are associated with the following noise model with noise strength $p = 10^{-3}$: $|0\rangle(|+\rangle)$-state qubit preparation is flipped to $|1\rangle(|-\rangle)$ with a probability of p, qubit measurement results are flipped with a probability of p, and single-qubit gates are followed by X, Y, or Z by probabilities of $p/3$ each. Two-qubit gates are followed by one of the two-qubit Pauli operators $I\otimes X$, $I\otimes Y$, $I\otimes Z$, ..., $Z\otimes Z$, except for the identity, with a probability of $p/15$ each. We consider no errors for qubit idling, as is appropriate for neutral atoms and trapped ions with coherence times many orders of magnitude longer than gate times [38, 39]. Remote physical Bell pair generation is associated with an error rate p_{Bell}. Bell pairs $|\Phi^+\rangle_{AB} = \frac{1}{\sqrt{2}}(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B)$ are shared between nodes A and B, followed by qubit B of the pair experiencing X, Y or Z error with probability $p_{\text{Bell}}/3$ each.

III. DISTRIBUTED FTQC AND LINK-LIMITED VOLUME

We consider two computing nodes A and B linked by an interconnect. The interconnect generates physical Bell pairs at a fixed speed (throughput), R qubits in each syndrome extraction (SE) cycle of the rotated surface code (Fig. 2). In distributed computing, computing nodes are typically placed nearby, resulting in classical communications with high bandwidth and negligible latency compared to the required time for SE. As such, we treat classical communication as free, be it one-way or two-way, throughout this article.

The *spacetime volume* is an important metric for evaluating how costly a certain protocol is. In the context of FTQC, this is typically computed in units of qubit-cycles [16], where the space cost is counted by the number of physical qubits actively involved in the protocol, and the time cost is evaluated by the number of SE cycles. For example, a transversal CNOT gate between two distance-d_s rotated surface code patches requires $2d_s^2-1$ qubits per patch and is followed by d_s cycles of SEs, with the leading-order term of the volume being $4d_s^3$, while the corresponding term for lattice-surgery CNOT is $12d_s^3$ [21, 40, 41].

Our primary interest is the spacetime volume required to prepare logical Bell pairs, which we call LLV (Fig. 2),

$$\mathcal{V} = \mathcal{V}_b + \mathcal{V}_f, \tag{1}$$

with $\mathcal{V}_b$ and $\mathcal{V}_f$ specified in the following. In (1), the first term $\mathcal{V}_b$ is the volume associated with buffering N physical

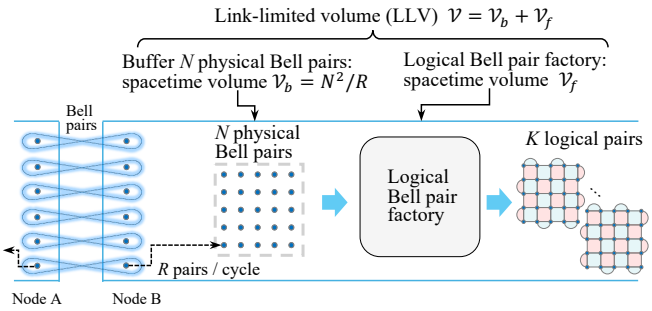

FIG. 2. The link-limited volume (LLV). LLV consists of network-related volume $\mathcal{V}_b = N^2/R$ and Bell pair factory volume $\mathcal{V}_f$ in each node (Eq. (1)), where N is the number of physical Bell pairs needed for the factory and R is the throughput of the physical Bell pairs, defined by the number of Bell pairs generated in the duration a single syndrome extraction cycle.

Bell pairs needed to perform an instance of the logical Bell pair preparation. For physical Bell-pair generation throughput of R pairs per cycle, the volume is N^2/R, since a buffer space of N must be kept for N/R cycles of SEs. The second term $\mathcal{V}_f$ in (1) is the spacetime volume for local operations needed to prepare a high-fidelity logical Bell pair using the buffered physical Bell pairs. This can be computed by counting the number of physical qubits involved in the logical Bell pair preparation protocol at each SE cycle and summing this value throughout the protocol. LLV must be modified from the above for protocols that consume physical Bell pairs sequentially or in a pipelined manner. For example, the remote lattice-surgery-based logical Bell pair preparation protocols [25, 26] consume $O(d_s)$ Bell pairs in each SE cycle over d_s cycles, instead of requiring all Bell pairs at the start of the protocol. In such a case, if the physical Bell pair throughput R is larger than the consumption speed, $\mathcal{V}_b$ becomes 0 since no buffering is required; if R is smaller than the Bell pair consumption speed, then some of the Bell pairs must be buffered initially to ensure that SE cycles are performed without additional latency, thus finite $\mathcal{V}_b$ is required. For protocols involving postselection with an acceptance rate of q, the LLV must be computed by the protocol volume multiplied by the expected attempt count $1/q$. We further let $\mathcal{Y}$ denote the yield of the Bell pair factory operation as the number of output logical Bell pairs per input physical Bell pair, considering retries in the case of probabilistic protocols. For example, a factory protocol requiring N physical Bell pair inputs and K logical Bell pair outputs, with a success probability of q, has a yield of $\mathcal{Y} = qK/N$. Inverse yield $1/\mathcal{Y} = N/qK$ thus denotes the number of physical Bell pairs required to output one logical Bell pair.

LLV quantifies how the optimal balance between the physical Bell pair requirement $\mathcal{V}_b$ and the volume of local operations $\mathcal{V}_f$ changes with the Bell pair generation throughput R. With a slow network with small R, where the shared entanglement is costly, protocols with small physical Bell pair consumption are favored; in contrast, with a fast interconnect with large R, LLV is affected more strongly by the volume of local operations, precisely reflecting the situation with less cost for creating Bell

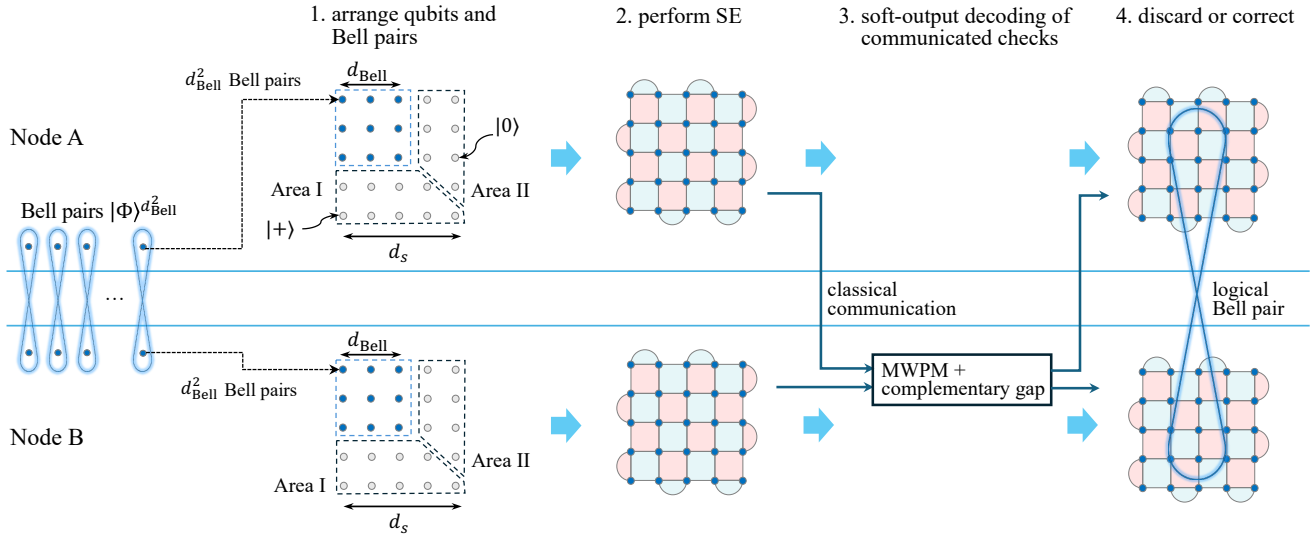

FIG. 3. Entanglement boosting. Entanglement boosting begins with the preparation of d^2_{Bell} physical Bell pairs, which are to be arranged in a square grid, together with qubits in $|0\rangle$ and $|+\rangle$ around them to form a $d_s \times d_s$ square grid. This is followed by d_s cycles of syndrome extraction (SE) and MWPM decoding. We additionally perform decodings for complementary logical outcomes to compute the complementary gap [42], which allows efficient postselection (see Appendix A 2 for the details of complementary gap calculation).

pairs.

The concrete values of the throughput R vary by orders of magnitude depending on the remote entanglement generation protocols, qubit types, and the implementation details of the SE cycles. For example, the state-of-the-art remote entanglement generation speed for trapped ions is on the order of $100\,\text{s}^{-1}$ [12, 43], and an estimation of the duration of a surface-code SE cycle is on the order of 100 µs to 1 ms [44, 45], giving R that ranges from 10^{-2} to 10^{-1}. In general, however, R can differ by many orders of magnitude across platforms. The characteristic gate times already vary by over three orders of magnitude [45] among leading qubit technologies, and the achievable rate of remote entanglement generation is expected to span an even wider range: while optical losses and inefficient qubit-photon coupling may significantly reduce the speed, fast interconnects such as microwave channels [46] and optical cavities [6, 14, 47] may enhance the speed by orders of magnitude. As a representative range, in this work, we consider R between 10^{-3} and 10^{2}.

IV. ENTANGLEMENT BOOSTING

Entanglement boosting is a logical Bell pair generation protocol operated within a single surface code patch, combining code projection onto a small surface code patch, code expansion, and postselection based on soft-output decoding [21, 48]. Conceptually, entanglement boosting consists of two steps. The first is preparing a logical Bell pair encoded in a $[[d^2_{\text{Bell}}, 1, d_{\text{Bell}}]]$ rotated surface code of distance d_{Bell} using the logical Bell pair preparation procedure of Fig 1a, and the second is its expansion to a larger distance d_s required to preserve the postselected states and for further operations, such as entanglement distillation, implemented by logical gates. The two steps can, in fact, be implemented simultaneously, achieving both reduced local error effects and spacetime volume, as described below.

In the first step, we prepare d^2_{Bell} physical Bell pairs between two parties A and B. The ith pair is stabilized by $X_i^{(A)} \otimes X_i^{(B)}$ and $Z_i^{(A)} \otimes Z_i^{(B)}$ where the superscript $A(B)$ denotes the operators acting on qubits in nodes $A(B)$. Both parties arrange the respective endpoints of the Bell pairs into a square lattice and then locally perform syndrome extraction of the rotated surface code. For the rotated surface code on a $d_{\text{Bell}} \times d_{\text{Bell}}$ square lattice, stabilizer checks are arranged in a checkerboard pattern [31], with plaquettes representing $X(Z)$-type stabilizer checks $g_{X(Z)}$, shown as red and blue plaquettes in Fig. 3. Each stabilizer check acts as the tensor product of Pauli $X(Z)$ operators on the four (or two at the boundary) qubits at the plaquette corners. Syndrome extraction requires $d^2_{\text{Bell}}-1$ auxiliary qubits in each node. Following the syndrome extraction, the parities of the corresponding measurement outcomes between the two parties are computed via classical communication, obtaining the values for joint stabilizer checks such as $g_X^{(j,A)} \otimes g_X^{(j,B)}$. In the noiseless case, these values must all be even, whereas noise may flip them. Upon successful execution of this protocol, the state is stabilized by $\overline{X}^{(A)} \otimes \overline{X}^{(B)}$ and $\overline{Z}^{(A)} \otimes \overline{Z}^{(B)}$, hence the output state is a $k = 1$ logical Bell pair encoded in the rotated surface code [36, 37].

In the next step, the code is expanded to a larger code distance d_s, as in the protocol in Ref. [49]. In this procedure, as shown in Fig. 3, additional physical qubits are prepared in $|+\rangle$ ($|0\rangle$) in areas I (II) around the initial surface code patch of distance d_{Bell} separated by the diagonal line, and syndrome extraction for the expanded code is performed for d_s cycles.

In entanglement boosting, the above two steps are performed simultaneously. While the code expansion deforms the two-

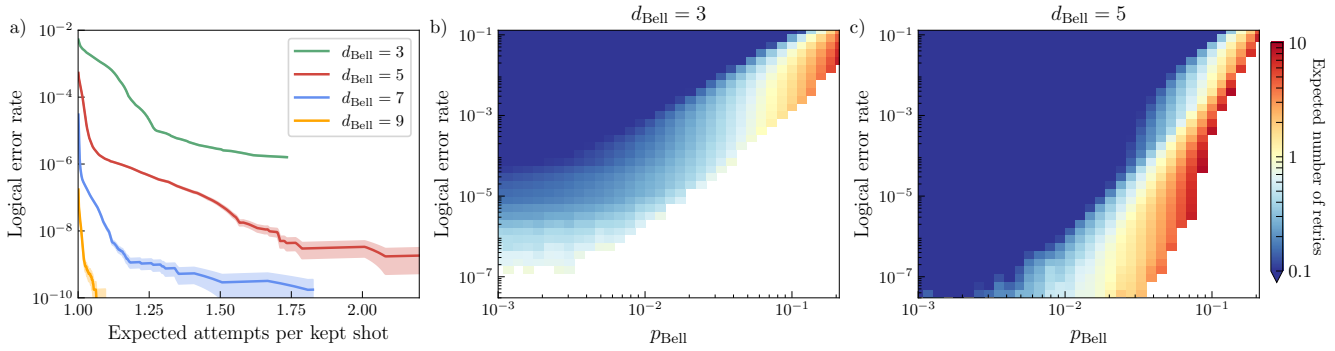

FIG. 4. Numerical simulation results for the entanglement boosting protocol. a) Numerical simulation results of the logical error rates of the logical Bell pairs generated by the entanglement boosting protocol, for varying postselection criteria, for $d_{\text{Bell}} = 3, 5, 7$ and 9, with d_s fixed at 19 and $p_{\text{Bell}} = 0.01$. Varying threshold values for the complementary gap results in different acceptance rate q_0; here, the logical error rate is shown for varying expected attempt count per kept shots, $1/q_0$. The shaded region is the standard error of estimated probability from 10^{10} sampling results. b-c) Numerical simulation results for the expected number of retries (colors), $1/q_0 - 1$, to achieve target error rates (vertical axis) for varying input Bell pair error rates p_{Bell} (horizontal axis) over two orders of magnitude, for $d_{\text{Bell}} = 3$ and 5.

weight stabilizers along the bottom and right edges of the initial distance-d_{Bell} surface code to four-weight stabilizers, the additional qubits are arranged such that the new stabilizers yield the same outcomes as the original two-weight stabilizers; for example, as shown in Fig. 3, each of the additional physical qubits below the initial patch is prepared in $|+\rangle$, and thus the two-weight X stabilizer check in the original patch and the corresponding four-weight X stabilizer check in the expanded patch yield the same outcome. This allows the simultaneous operation of the two steps to reduce the effects of local errors and the overall volume.

Operationally, the combined procedure proceeds as follows. Initially, both parties arrange d_{Bell}^2 physical Bell pairs and $d_s^2 - d_{\text{Bell}}^2$ physical qubits prepared in $|+\rangle$ and $|0\rangle$ (Fig. 3). Next, d_s cycles of syndrome extraction are performed for the distance-d_s surface code, where the stabilizer checks are denoted by red and blue plaquettes in Fig. 3. A sends the outcomes of the first cycle of syndrome extraction to B via classical communication, and B obtains the error syndrome by computing the parities of the corresponding syndromes.

The error syndrome is then decoded using the minimum-weight perfect matching (MWPM) decoder. The complementary gap [21, 42, 48] is also computed, which is the absolute difference between the minimum weights identified in the MWPM decoder conditioned on the original and complementary logical outcomes (see Appendix A 2 for more details and illustrations of the complementary gap). If this value is below a chosen threshold, the boosting protocol is aborted. Otherwise, the resulting logical qubit pair is kept, and error correction is performed. The complementary gap captures error information from both the encoding and expansion steps, allowing postselection on the decoded error information and thereby achieving a substantial reduction in output error rates. In our numerical simulation, we sampled the above protocol and obtained the complementary gap value and the existence of logical errors for each shot by evaluating the $\overline{X}^{(A)} \otimes \overline{X}^{(B)}$ and $\overline{Z}^{(A)} \otimes \overline{Z}^{(B)}$ logical operators. For a given threshold complementary gap value, we first obtain the acceptance rate q_0 by dividing the number of accepted shots (the complementary gap value below the threshold) by the total number of samples; then, we determine the logical error rate by counting the number of shots that have a gap value below the threshold and also contain a logical error, and dividing this count by the number of accepted shots (see Appendix A 1 for the details of the numerical simulation).

There is a conceptual connection between entanglement boosting and recent work on magic state cultivation [21, 23]. Both protocols make use of soft-output decoding, such as the complementary gap for code expansion of postselected logical qubits; the initial state is prepared by the logical double-checking (H_{XY} measurement) for magic state cultivation, while the projection of Bell pairs to a logical Bell pair is used for entanglement boosting. This difference arises from the fact that Bell pairs are stabilizer states that admit simpler methods for error detection via stabilizer codes [27], allowing the entanglement boosting to have a much simpler implementation, as described above.

However, from an operational viewpoint, fundamental differences in the physical implementation of the physical T gates and remote Bell pair generation lead to different design considerations. In magic state cultivation, physical T gates are operated locally with negligible latency at any location and time, providing flexibility in protocol design. By contrast, in distributed FTQC, the remote Bell pairs are generated only by the photonic interconnect, which has a limited generation speed; therefore, the protocol must balance Bell pair usage against the local spacetime volume. For this reason, entanglement boosting only partially adapts ideas from magic state cultivation and tailors them to the fundamentally different theoretical and operational characteristics of remote entanglement. By optimizing the protocol based on the LLV metric to meet the distinct requirements for logical Bell pair preparation, entanglement boosting fully leverages advances in magic state preparation to achieve efficient logical Bell pair preparation.

In Fig. 4, we show the circuit-level simulation results for the

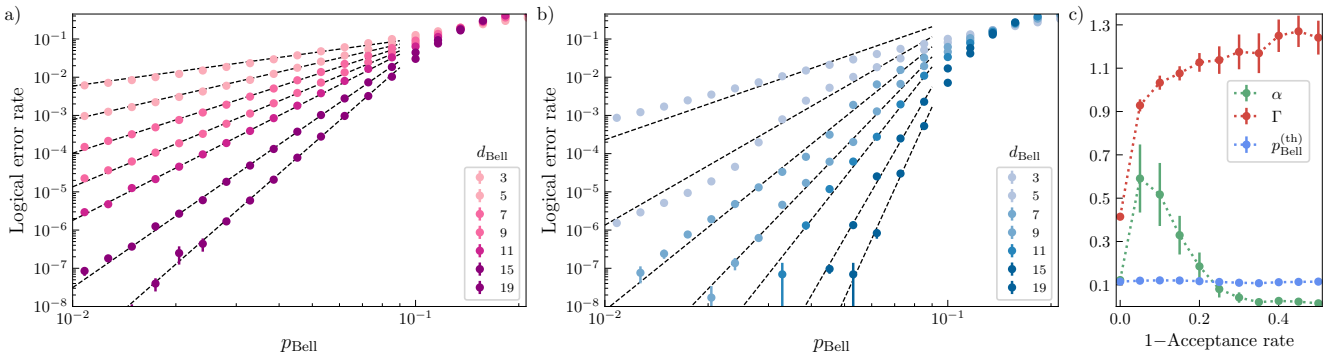

FIG. 5. Scaling of the logical error rate of the Bell pairs produced with the entanglement boosting protocol. a-b) logical error rate as a function of the physical Bell pair error rate p_{Bell} for different d_{Bell}, with acceptance rates q_0 of 100% (a) and 90% (b) based on the complementary gap. Each point is obtained from circuit-level simulations of the entanglement boosting protocol, and the dashed line represents the fitting with Eq. (2). Error bars represent the standard error of estimated probability from the 10^8 sampling results. b) the dependence of the fitted values of parameters α, Γ and $p_{\mathrm{Bell}}^{(\mathrm{th})}$ on the discard rates $1-q_0$ of entanglement boosting, where the error bars represent 95% confidence intervals.

boosting protocol, with the final distance of the rotated surface code being $d_s = 19$ and the input Bell pair error rate being $p_{\mathrm{Bell}} = 1\%$. We then vary d_{Bell} and the postselection criteria, which are set by the threshold values for the complementary gap to discard the trial. From the results of the numerical simulation, we identify the acceptance rate q_0 for a given threshold complementary gap value, and the logical error rate of kept shots is plotted against the *expected attempts per kept shot*, $1/q_0$.

These plots highlight the tradeoff between the output logical error rate and the acceptance rate of the boosting protocol. In Fig. 4a, tightening the postselection criterion (moving to the right along the horizontal axis) initially rapidly suppresses the logical error rate of the kept shots, at the cost of requiring more attempts per successful output. Figures 4b and 4c further show the expected number of attempts required to reach a given logical error rate for varying physical Bell pair error rates p_{Bell}. The error suppression is observed for up to $p_{\mathrm{Bell}} \approx 10\%$, demonstrating a wide operating regime of this protocol. Overall, these results demonstrate that entanglement boosting provides a tunable mechanism to trade physical Bell pair consumption and postselection criteria for output error rates, offering a high-performance and highly tunable method to produce logical Bell pairs while maintaining all operations within a single surface code patch. In Appendix A 3, we analyze the effect of idling errors by performing additional numerical simulations of entanglement boosting protocols, with $p_{\mathrm{Bell}} = 1\%$ and $d_{\mathrm{Bell}} = 3$ and 5, with the same assumptions as Fig. 4a except for the presence of idling errors with noise strength $p = 0.1\%$. For these parameters, the logical error rates reached by the entanglement boosting protocol remain similar in the presence of idling errors, while the expected attempts per kept shot increase by up to 40%.

To further analyze the scaling of the logical error rates as a function of d_{Bell} and acceptance rates q_0, we present the logical Bell pair error rate of the entanglement boosting protocol in Fig. 5a-b as a function of the physical Bell pair error rates p_{Bell} and d_{Bell}, for acceptance rates of 100% (Fig. 5a; error correction) and 90% (Fig. 5b). Points with different colors correspond to different d_{Bell} values, illustrating how the logical error probability decreases with increasing d_{Bell} and a reduced Bell pair error rate; noise strengths for the local physical operations are maintained at 0.1% throughout the simulations. The dashed lines indicate the fitting with an approximate scaling function for the logical error rate p_L as a function of p_{Bell}, d_{Bell}, and $p_{\mathrm{Bell}}^{(\mathrm{th})}$,

$$p_L = \alpha \left(\frac{p_{\mathrm{Bell}}}{p_{\mathrm{Bell}}^{(\mathrm{th})}} \right)^{\Gamma d_{\mathrm{Bell}}}, \tag{2}$$

where the fit is performed once for each panel, considering all d_{Bell} shown in each panel. From the fits, we obtain α, Γ and $p_{\mathrm{Bell}}^{(\mathrm{th})}$ that depend on the discard rate of the protocol. Figure 5c shows how the fitted parameters depend on the discard rates. As the discard rates increase, Γ increases from 0.4 to 1.2, while the parameter $p_{\mathrm{Bell}}^{(\mathrm{th})}$ remains constant, quantifying the enhanced error suppression thanks to the postselection based on soft-output decoding [50]. We remark that, due to the physical operations with noise strengths at $p = 0.1\%$ error probabilities, the observed scaling has a limited range of applicability, as discussed in Appendix A 4, resulting in the saturation of error suppression for p_{Bell} below 1%. Further, it is expected that the logical error rate of the boosting protocol will saturate at the logical error rates of the rotated surface codes with d_s, such as 10^{-12} for $d_s = 19$. We also show the additional results, including the results for larger discard rates, in Appendix A 4.

Finally, to assess the performance of entanglement boosting using a practically relevant circuit-volume metric (Sec. III), in Fig. 6, we plot the LLV for preparing logical Bell pairs at a logical error rate of 10^{-10} encoded in the $d_s = 19$ rotated surface code, starting from $p_{\mathrm{Bell}} = 1\%$, 3%, and 5%, for varying physical Bell pair throughput R. For this evaluation, we used the scaling of the logical error rates for varying postselection fractions, with discard rates $1 - q_0$ of up to 50% considered. We further compare the LLV with remote lattice surgery protocol for logical Bell pair generation with rotated

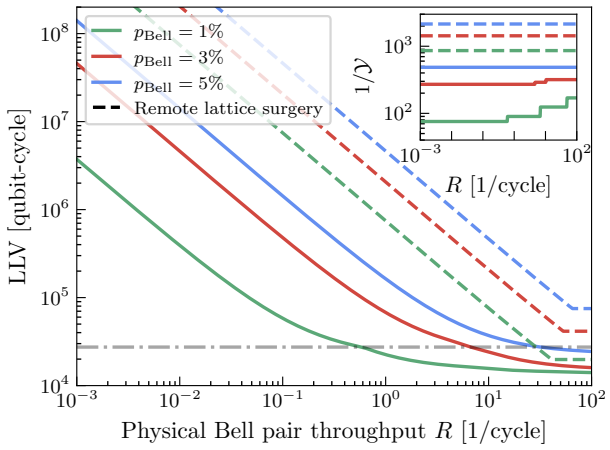

FIG. 6. The link-limited volume (LLV) to prepare a logical Bell pair encoded in the rotated surface code at logical error rates of 10^{-10}. We set the distance of the expanded surface code to be $d_s = 19$. The dashed line is the corresponding LLV for the remote lattice surgery protocol for the rotated surface code (see Appendix A 5 and Refs . [25, 26]), with corresponding colors for each p_{Bell}. Horizontal dash-dotted line is the circuit volume for local transversal CNOT gate followed by d_s cycles of SE, at $d_s = 19$. (inset) The number of physical Bell pairs consumed per output logical Bell pair (inverse yield, $1/\mathcal{Y}$). The jumps in the plots are due to the changes in optimal d_{Bell} that minimizes the LLV for varying R.

surface code [25, 26], which requires $O(d_s^2)$ Bell pairs over d_s cycles of SE. We perform circuit-level simulation of the remote lattice surgery protocol with the same noise model as the one used for the simulation of the entanglement boosting protocol, as detailed in Appendix A 5, to evaluate the required d_s to reach the target logical error rate of 10^{-10}. Appendix C 1 describes the LLV model for remote lattice surgery. For the boosting protocol, for each p_{Bell} and R, we choose the d_{Bell} and postselection criteria that minimize the LLV while keeping the final code distance at $d_s = 19$. Thus, we observe jumps in the inverse yield for varying R in the inset of Fig. 6; for $p_{\mathrm{Bell}} = 1\%$, the chosen d_{Bell} ranged between 9 and 13, while for $p_{\mathrm{Bell}} = 3\%$, the range is between 13 and 17, with larger numbers favored for higher R; this is because the reduced volume for Bell pair buffering favors reduced retries, resulting in smaller volumes required for local operations. For $p_{\mathrm{Bell}} = 5\%$, $d_{\mathrm{Bell}} = 17$ was favored throughout the range of R considered. For remote lattice surgery, the required surface-code distance to achieve the target Bell pair logical error rate is $d_s = 21, 27$ and 33, for physical Bell pair error rates of $p_{\mathrm{Bell}} = 1\%$, 3%, and 5%, resulting in significant Bell pair consumption compared to the boosting protocol (see Appendix A 5). Entanglement boosting, therefore, achieves orders of magnitude improvements in the LLV compared to the remote lattice surgery protocol for a wide range of bell pair throughput, while both protocols operate within a single surface-code patch in each node and maintain 2D local connectivity.

V. PIPELINED ENTANGLEMENT DISTILLATION

The entanglement boosting stage can be complemented by logical-level entanglement distillation to further suppress logical error rates. In particular, we consider the entanglement distillation protocol based on $[[n, k, d]]$ stabilizer code C defined by a set of independent $n - k$ stabilizer generators $S = \{s_i\}_{i=1,\ldots,n-k}$. By choosing codes with a high encoding rate k/n, it is possible to improve the logical error rates of the Bell pairs with only a moderate reduction in yield $\mathcal{Y}$, at the cost of an increased circuit volume for local operations. This is in contrast to the error suppression by the entanglement boosting only, where a quadratic increase in the number of Bell pairs is required to achieve stronger error suppression with increased d_{Bell}. Therefore, the combined approach is expected to provide improved LLV in the regime of small R or very low target logical Bell pair error rates, complementing the entanglement boosting protocol for further scalability.

Here, we follow the protocol illustrated in Fig. 1b, with the n Bell pair inputs encoded in rotated surface code, such as those from the boosting stage; distillation is therefore implemented by logical gates on the rotated surface code. Below, we describe how to construct a distillation circuit that can be implemented efficiently with reconfigurable qubit platforms by the parallel use of local one-way qubit shuttling.

In the stabilizer formalism, each stabilizer generator of C can be represented by a binary vector of length $2n$. For the first n entries, the presence of 1 at index i indicates the X operator on qubit i, and the remaining n entries indicate the Z operators. Collecting these binary vectors for all stabilizer generators gives an $(n - k) \times 2n$ matrix,

$$H_q = [\, H_X \mid H_Z \,], \tag{3}$$

The two $(n - k) \times n$ submatrices H_X and H_Z describe, respectively, the X and Z components of the set of stabilizer generators. The stabilizer generators are not unique, and H_q can be transformed by Gaussian elimination into a standard form following the procedure of Ref. [51]. Here, with r denoting the rank of H_X, the resulting matrix is

$$H_s = \left[\begin{array}{ccc|ccc} I_1 & A_1 & A_2 & B & C_1 & C_2 \\ 0 & 0 & 0 & D & I_2 & E \end{array}\right], \tag{4}$$

where I_1 and I_2 are identity matrices of size $r \times r$ and $(n-k-r)\times(n-k-r)$, B is $r\times r$, A_1 and C_1 are $r\times(n-k-r)$, A_2 and C_2 are $r\times k$, D is an $(n-k-r)\times r$, I_2 is $(n-k-r)\times(n-k-r)$, and E is $(n-k-r)\times k$. The logical X operator can be expressed in block-matrix form as

$$X_s = [\, 0\, E^{\mathsf{T}}\, I_3 \mid V\, 0\, 0 \,], \tag{5}$$

where I_3 is a $k\times k$ identity matrix and $V = E^{\mathsf{T}}C_1^{\mathsf{T}} + C_2^{\mathsf{T}}$ modulo 2 [51], and T denotes matrix transpose. We show the H_s for $[[6, 4, 2]]$ quantum parity code [52] and $[[7, 1, 3]]$ Steane code [53] in Fig. 7a-b, along with binary representations of X logical operators.

An encoding circuit can be synthesized from the standard form following Ref. [51], and its reverse (unencoding circuit)

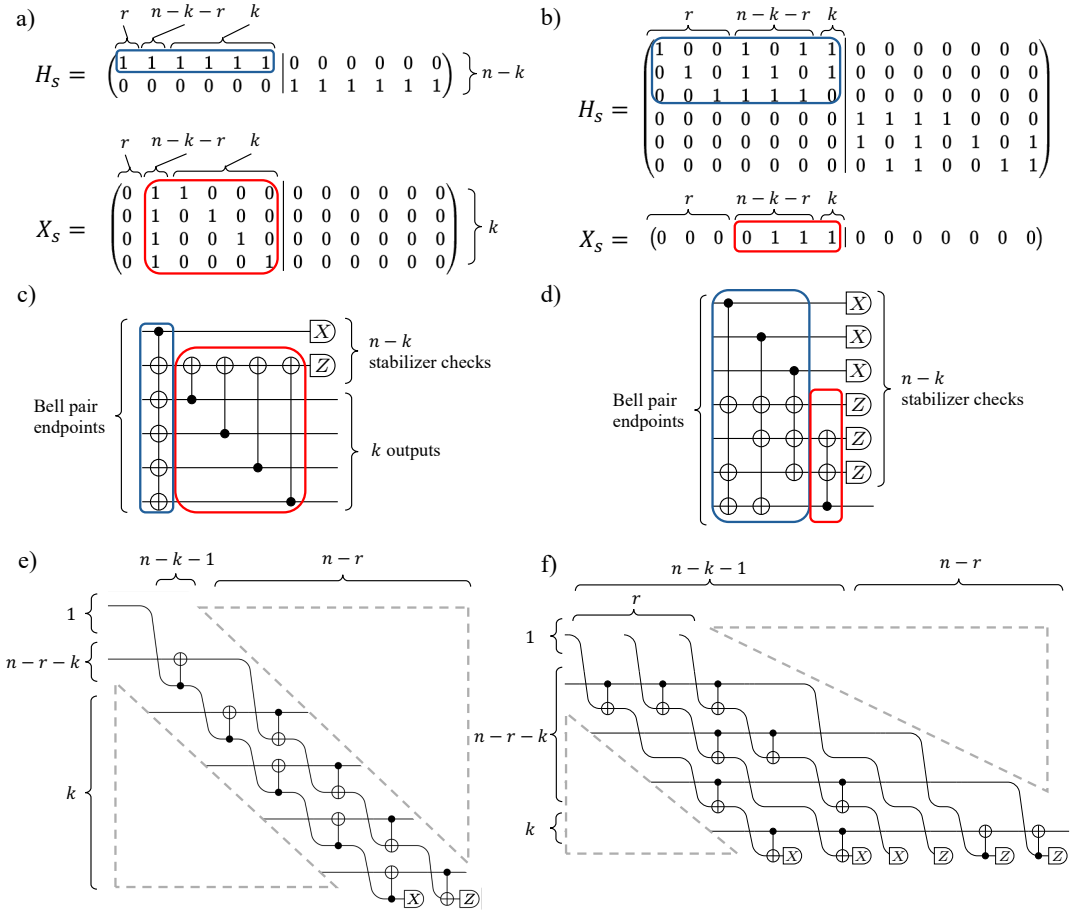

FIG. 7. Pipelined entanglement distillation based on CSS codes. (a,b) standard form of $(n-k)\times 2n$ binary matrices H_s representing the stabilizer generators of $[[6,4,2]]$ quantum parity code and $[[7,1,3]]$ Steane code, along with binary matrix representation of logical X gates. (c,d) entanglement distillation circuit (only for one of the nodes shown), based on the encoding circuit synthesis of Ref. [51]. (e,f) same circuits as (c,d), implemented with qubit reconfigurations (wire bending); see Fig. 8.

can be used for entanglement distillation, which we describe below for CSS codes. First, n Bell pairs are prepared in nodes A and B, and both parties first apply gates following the first r rows of H_s: for the ith row, if the jth column entry is 1, place a CNOT between qubits i and j, with qubit i being the control. For CSS codes, entries $n+1,...,2n$ of the ith row are 0; hence, this completes the first step (blue rectangles in Fig. 7a-d). Second, following Eq. (5), CNOT gates are placed following the matrix E^{T}: if jth entry of ith row is 1, then place CNOT between qubits $n-k+i$ and j, where qubit j is the target (red rectangles in Fig. 7a-d). In general, for non-CSS codes, not only CNOT gates but CZ and controlled-Y gates appear in the circuit [51]. Following the application of CNOT gates, the first r qubits are measured in the X basis, while $n-r-k$ qubits are measured in the Z basis. These measurement results correspond to the stabilizer checks of the code C, allowing the postselection or error correction based on the measurement outcomes. We note that the same circuit is executed in the node A and B for entanglement distillation (Fig. 1b). For the case of error detection, if any of the stabilizer generators are measured to be in an odd parity between nodes A and B, the output state is discarded; if the error rate of the Bell pair is p_{Bell}, and the code distance of C is d, then the error rate of the post-selected output state is $O\left(p_{\text{Bell}}^{d}\right)$.

For reconfigurable qubits with efficient parallel qubit shuttling capability, such as neutral atoms and trapped ions, we propose to implement the distillation circuits illustrated in Fig. 7c-d using an equivalent *pipelined* implementation illustrated in Fig. 7e-f. Here, instead of preparing n qubits at the start and applying the gates following the above procedure, $n-r$ qubits are first prepared, and r remaining qubits are sequentially moved across the $n-r$ qubits in a pipelined manner. This is illustrated in the modified circuits of Fig. 7e-f as the bending of the wires, corresponding to qubit reconfigurations. Since reconfigurations occur in only one direction at each step, this is compatible with fully parallel qubit shuttling. Reconfiguration and transversal CNOT gates for the first r qubits implement the gates in the blue rectangle in Figs. 7a-d, and $n-r-k$ qubits are then shuttled in the same direction, realizing the gates in the red rectangle, completing the required gate network. This construction enables the concentration of the intrinsic idle volume of the circuit, as illustrated by the dashed triangles in Figs. 7e and f, which can be utilized efficiently for

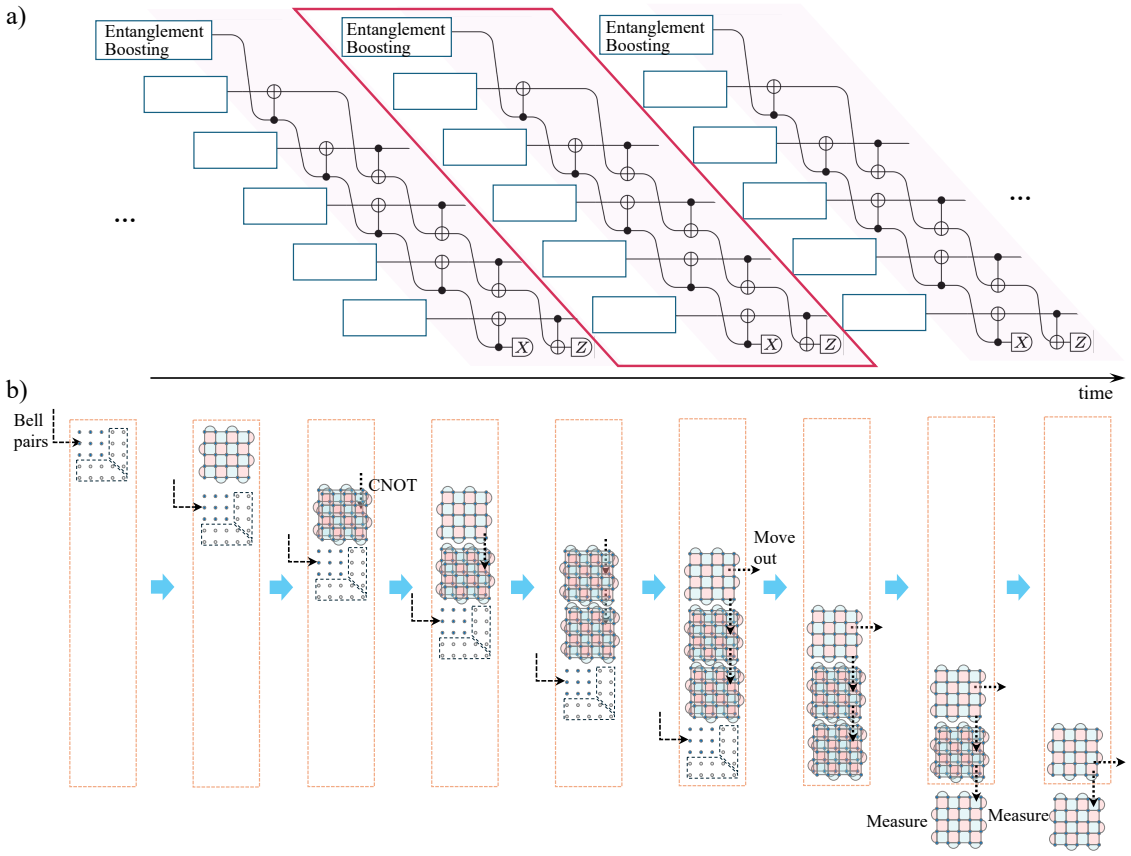

FIG. 8. Reconfigurable-qubit implementation of pipelined entanglement distillation with $[[2m, 2m-2, 2]]$ quantum parity code. a) an example circuit for $[[6, 4, 2]]$ code, distilling 4 logical Bell pairs out of 6 logical Bell pairs. The curved wires in the circuit represent the logical qubit reconfigurations, which move the location of the logical qubits for transversal CNOT gates. For example, we assume the input to the circuit originates from the entanglement boosting protocol. b) More detailed illustration of qubit reconfiguration for one instance of the distillation in a) (marked by red rhombus). The first two input qubits are moved along the vertical direction inside the Bell pair factory (orange rectangle), interacting with other qubits. Once the interactions are completed, they are moved out from the factory for logical X- and Z-basis measurements. Bell pair input and output can be directed along the horizontal axis, with sequential input of n boosted Bell pairs during the execution of the distillation.

another instance of the same or a different distillation circuit.

In Fig. 8, we further clarify the implementation of the pipelined distillation using $[[2m, 2m-2, 2]]$ quantum parity code [52], which has stabilizer generators $X_1X_2...X_{2m}$ and $Z_1Z_2...Z_{2m}$. We use $m = 3$ to illustrate a small-scale example in Fig. 8, while larger m results in a better encoding rate $k/n = (2m-2)/2m$. Figure 8a shows the time-multiplexed operation of multiple instances of the distillation circuits, where the idle volumes indicated in Figs. 7e-f are used for another instance of the distillation circuit. Figure 8b is a more concrete qubit reconfiguration procedure, taking into account the input and output of the logical Bell pair factory (orange dotted rectangle). Qubit shuttling along the vertical direction and transversal gates implement the logical circuit for entanglement distillation, while the Bell pair inputs (e.g., entanglement boosting) and outputs can be moved along the horizontal direction, allowing fully pipelined operations based on qubit reconfiguration. In general, the circuit volume of this protocol is approximated by

$$\begin{aligned} \mathcal{V} &\approx k\mathcal{V}_k + r\mathcal{V}_r + (n-r-k)\mathcal{V}_{(n-r-k)}, \\ \mathcal{V}_k &= (n-k)(2d_s^3 - d_s), \\ \mathcal{V}_r &= (n-r)(2d_s^3 - d_s) \\ \mathcal{V}_{(n-r-k)} &= (n-1)(2d_s^3 - d_s), \end{aligned} \tag{6}$$

for the distance-d_s rotated surface code (see Appendix C 3 for a more detailed description of the above model). The fully parallelized reconfiguration of the logical patches in each circuit depth, as illustrated in Fig. 8b, maintains a small reconfiguration time cost between circuit layer executions.

Figure 9 quantitatively compares the LLV of the entanglement boosting and the combined approach of entanglement boosting and the pipelined distillation with the $[[10, 8, 2]]$ quantum parity code. Here, we utilized a larger code than shown in Fig. 8 to achieve a better encoding rate, which improves the yield at the cost of slightly weaker error suppression in entanglement distillation (see Appendix B). Figure 9a shows

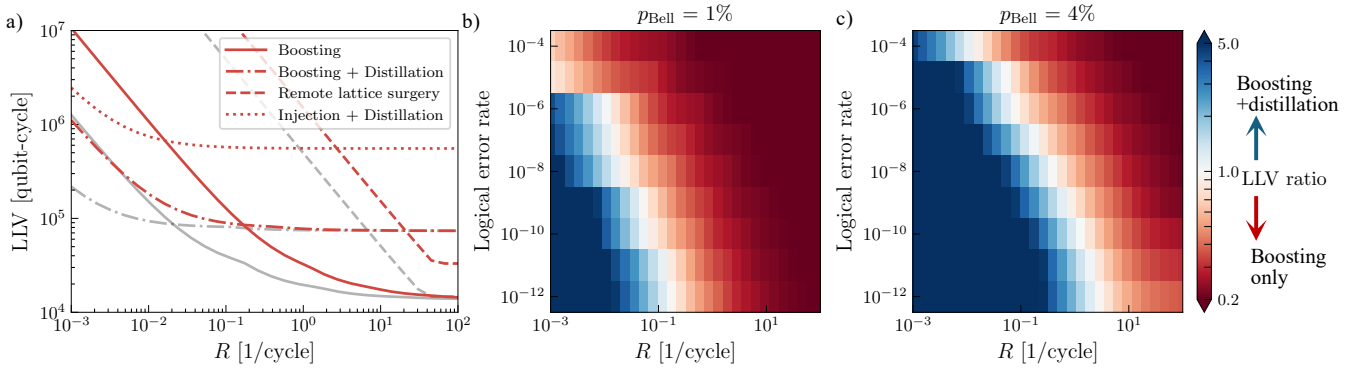

FIG. 9. Comparison between the entanglement boosting and the combined boosting + pipelined distillation scheme, with an example of [[10, 8, 2]] quantum parity code used for the pipelined distillation. a) Link-limited volume (LLV) as a function of the Bell pair generation throughput R, for target logical error rates of 10^{-12} (red) and 10^{-8} (gray), with $p_{\mathrm{Bell}} = 1\%$. Solid lines represent the entanglement boosting, the dash-dotted lines denote the boosting-and-pipelined distillation protocol, dashed lines indicate remote lattice surgery and dotted lines is the injection-distillation approach of Ref. [30] (see Appendix C 4). With a higher R, boosting yields the smallest LLV, whereas for a lower R, combining with distillation becomes advantageous due to improved yield at the cost of larger local circuit volume. b-c) Ratio of LLVs for the boosting and combined boosting-distillation protocols, as functions of R for $p_{\mathrm{Bell}} = 1\%$ and 4%, respectively, identifying the crossover regime where the combined protocol outperforms boosting alone. For $p_{\mathrm{Bell}} = 1\%$, the crossover lies between $R \approx 10^{-4}$ and 10^{-1} per cycle; for $p_{\mathrm{Bell}} = 4\%$, it extends up to $R \approx 1$ per cycle. The distance-2 $[[2m, 2m-2, 2]]$ code used here serves as a representative toy model; higher-distance, high-rate codes, such as quantum Hamming or QLDPC codes, further enhance error suppression.

the LLV as a function of the Bell pair generation throughput R, for output logical error rates of 10^{-12} (red) and 10^{-8} (gray), where the logical error rates of the entanglement boosting are obtained from the scaling reported in Fig. 4. We also show the LLV for the remote lattice surgery protocol [25, 26] and a state-of-the-art injection-distillation protocol based on concatenated distillation [30] (see Appendix C for the details of LLV evaluation for these protocols), both of which have an order-of-magnitude larger LLV than the boosting or boosting+distillation protocols for a wide range of R. With the pipelined distillation operation using the high-rate code, the inverse yield $1/\mathcal{Y}$ is smaller than that of the boosting-only protocol, while the local operation volume is larger. These distinct features lead to the crossover of the LLV as a function of the Bell pair generation throughput R: for high R, the boosting-only approach (solid line) is efficient, while for lower R, it is cheaper to increase the local circuit volume and increase the yield by using a combination of boosting and pipelined distillation (dash-dotted lines). The crossover point is also dependent on the output logical error rate, with a lower target error rate resulting in larger crossover throughput R, since the required number of physical Bell pairs is larger. The remote lattice surgery protocol has a significantly larger LLV due to its large physical Bell pair consumption (dashed lines).

In Fig. 9b-c, we show the ratio of LLVs for the boosting-only protocol and the combined boosting-and-distillation protocol with $[[10, 8, 2]]$ code used for the pipelined distillation in order to identify the crossover point as a function of varying output logical error rates. For $p_{\mathrm{Bell}} = 1\%$ and a target logical error rate of 10^{-12}, the crossover is at $R \approx 10^{-1}$ per cycle, while for $p_{\mathrm{Bell}} = 4\%$, the crossover increases to $R \approx 1$ per cycle, suggesting that the combined protocol is more efficient for a wider range of parameters where the physical Bell pair error rates are higher. With a fast quantum interconnect of $R \gtrsim 1$ per cycle, the boosting protocol is favored for the range of target error rates shown.

It should be emphasized that the distance-2 code used here serves only as a simple example to demonstrate the pipelined distillation and its general behavior. In realistic architectures, higher-distance, high-rate codes, such as the quantum Hamming codes [54, 55], high-rate quantum low-density parity-check (QLDPC) codes [56, 57], and quantum BCH codes [54, 58], would potentially provide even more scalable implementations with stronger error suppression. The concatenation of pipelined distillation protocols also supports a scalable approach, with automated code sequence optimization recently demonstrated for reducing inverse yield and memory footprint [30]. Optimizing the combination of entanglement boosting with pipelined entanglement distillation, across a range of code choices and protocol variants, is left for future investigation.

VI. CONCLUSION

In this work, we have proposed entanglement boosting to efficiently transform noisy physical Bell pairs into logical Bell pairs encoded in rotated surface codes. With all operations kept within a rotated surface code patch, this approach achieves a substantial reduction of the logical error rates by using variable d^2_{Bell} physical Bell pair inputs, thus achieving low spacetime volume to prepare logical Bell pairs. This protocol can be complemented by a pipelined implementation of stabilizer entanglement distillation executed with logical gates of the surface code, designed for a reconfigurable qubit platform with parallel qubit shuttling. The existing protocols for

logical Bell pair preparation are constrained by two opposing characteristics: achieving small local circuit volume demands substantial physical Bell-pair consumption, while injection-and-distillation protocols incur large circuit volume. By contrast, the combined approach presented here resolves these limitations by providing a flexible interpolation between the two regimes.

As an outlook, further performance improvement is expected through the use of soft information regarding physical Bell pair errors, readily accessible in experiments via the photon detection times [14, 59] and other techniques such as erasure conversion [60, 61]. Correlated decoding methods for the transversal-gate FTQC protocols [62, 63], with $O(1)$ syndrome extraction cycles between logical gates, further enhance the efficiency of pipelined entanglement distillation.

ACKNOWLEDGEMENTS

We acknowledge discussions with Nicholas Fazio, and we thank Akihisa Goban and Shiro Tamiya for discussions and their detailed feedback on the manuscript.

CODE AVAILABILITY

Code used for the numerical simulation in this work is available at https://github.com/nano-qt/entanglement-boosting.

DECLARATION OF COMPETING INTERESTS

S. Sunami and H. Yamasaki are employees, and Y. Hirano and T. Hinokuma are interns of Nanofiber Quantum Technologies, Inc.

APPENDICES

Appendices are organized as follows. In Appendix A, we summarize our notation and provide a detailed account of the numerical simulations presented in the main text. In Appendix B, we present the details of the performance scaling of the entanglement distillation with quantum parity codes. Appendix C describes the detailed models for the LLV we used for the results in the main text.

Appendix A: Numerical simulations

Our terminology is summarized in Table A1.

1. Circuit-level simulation of entanglement boosting

We perform a Monte Carlo sampling simulation to evaluate the performance of the entanglement boosting protocol in Sec. IV using Stim [64] and PyMatching [65]. We adopt the following noise model for the simulation: for the input Bell pairs, we first prepare ideal Bell pairs and subject one of the arms of the pairs to a single-qubit depolarizing channel with parameter p_{Bell}. Other local operations experience noise as described in Table A2 with a noise strength of 0.1%, while we do not consider idle errors; this is justified by the fact that idling error rates are small in atomic qubits such as neutral atoms [38].

Figure A1 depicts the implementation of the protocol. We have two surface code patches of distance d_s, one for A and one for B. Each party prepares its surface code patch using the endpoints of d_{Bell}^2 physical Bell pairs, and the remaining qubits are separable, prepared following the layout in Fig. 8.

Subsequently, A and B each perform d_s cycles of syndrome extraction using local physical operations. While A performs syndrome extraction independently of B, in our simulation, B uses A's first-cycle error syndrome as the baseline error syndrome of its code. This corresponds to computing error syndrome parities, as described in Sec. IV, which is equivalent to communicating the readout results after the syndrome extraction.

After d_s cycles of syndrome extraction for each party, we perform one cycle of *noise-free* syndrome extraction, followed by noise-free logical $\overline{XX}$ and $\overline{ZZ}$ measurements. The syndrome values are used to compute the complementary gap [42],

Description	symbol
Bell pair generation throughput	R
Bell pair error rate	p_{Bell}
noise strength for physical operations (see Table A2)	p
acceptance rate	q_0
code distance for Bell pair projection	d_{Bell}
code distance of full surface code after expansion	d_s
spacetime volume of single trial	$\mathcal{V}_0$

TABLE A1. Notation for the entanglement boosting protocol used in this paper.

physical operations	noise
reset (X)	Z error with probability p
reset (Z)	X error with probability p
measurement (X)	Z error with probability p
measurement (Z)	X error with probability p
single-qubit gates	X, Y or Z error with with probability $p/3$ each
two-qubit gates	two-qubit Pauli errors except $I \otimes I$, with probability $p/15$ each

TABLE A2. Noise model of local operations for circuit-level simulation of the entanglement boosting stage. Noise model for the physical Bell pairs is given in Sec. II C.

described in more detail in Appendix A 2, while the final measurements provide the reference logical error to be compared with the decoding result.

Following the decoding, a sample is labeled as discarded if its complementary gap is below a certain threshold. A sample is "valid" if it is not discarded and the outcomes of the $\overline{XX}$ and $\overline{ZZ}$ measurements match the expected values. A sample is "wrong" if it is not discarded and the outcomes of the $\overline{XX}$ and $\overline{ZZ}$ measurements deviate from the expected values. The logical error rate is $\frac{\#\text{wrong}}{\#\text{valid}+\#\text{wrong}}$, and the acceptance rate is $\frac{\#\text{valid}+\#\text{wrong}}{\#\text{discarded}+\#\text{valid}+\#\text{wrong}}$, which are reported in the main text.

2. Complementary gap

The complementary gap is a value that represents the decoder's confidence in its decoding results. As a simple example, we consider the distance-5 rotated surface code (Fig. A2). For simplicity, in this subsection, we focus on X stabilizer checks that detect Z errors. We additionally assume that syndrome extraction is noise-free.

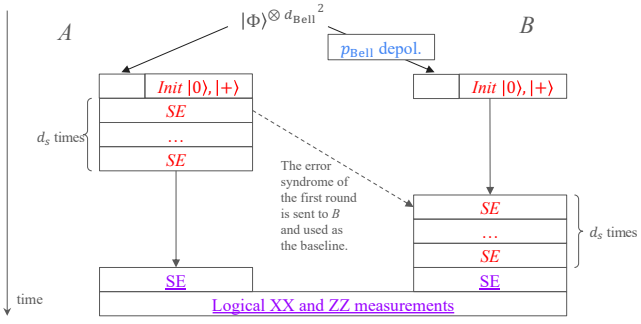

FIG. A1. Numerical simulation of the boosting stage described in Sec. IV. Steps in red are subject to the noise model shown in Table A2, while the steps in purple (with underlines) are noiseless. We initially prepare d_{Bell}^2 Bell pairs and subject one of the arms of the pairs to a single-qubit depolarizing channel. The remaining $d^2 - d_{\text{Bell}}^2$ physical qubits are prepared with noise strength 0.1% in each party. We simulate the SE steps of two parties in series for implementation purposes, which is equivalent to simultaneous SE operations followed by classical communication to exchange the syndrome information in an actual operation.

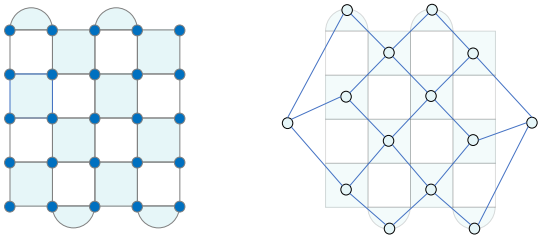

FIG. A2. Left: X stabilizer checks on the distance-5 rotated surface code. Each filled circle represents d_s^2 qubits comprising the code, and each light-blue square plaquette represents an X stabilizer check. Right: the corresponding error graph. Each vertex (open circles), except for the leftmost and rightmost, corresponds to an X stabilizer check, and each edge corresponds to a data qubit shared by two stabilizer checks. The leftmost (rightmost) vertex is a virtual vertex referred to as the left (right) boundary node.

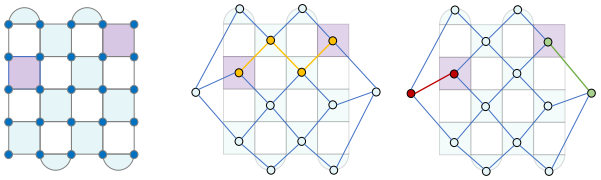

FIG. A3. Left: X stabilizer checks on the distance-5 rotated surface code. Z errors are detected by the purple checks. Middle: a solution of the MWPM problem conditioned with the left boundary check off. Right: a solution of the MWPM problem conditioned with the left boundary check on. Matching paths are shown in different colors.

Figure A2 (left) depicts the X stabilizer checks on the distance-5 rotated surface code. The minimum-weight perfect matching (MWPM) decoder [17] performs decoding by solving a matching problem on a graph known as the *error graph*. The right panel of Fig. A2 shows the corresponding error graph. Each vertex corresponds to an X stabilizer check, and each edge connecting two vertices represents the data qubit shared by those stabilizer checks. There are two virtual vertices, i.e, the left and right boundary nodes, to handle data qubits checked by only one stabilizer check. Each edge is associated with a weight, $w_i = -\log p_i$, where p_i denotes the Z-error probability of qubit i. The MWPM decoder finds the minimum-weight matching on this error graph.

The complementary gap, defined as the absolute difference between the minimum weights conditioned on the complementary logical outcomes, is illustrated in Fig. A3. The complementary gap is computed by running the MWPM decoder with the left boundary node forced on and off, and taking the absolute difference between the two resulting weights. If the complementary gap is small, the decoder's confidence in its decision is low. In the left panel, purple checks detect Z errors. Two matchings conditioned on the complementary logical outcomes are illustrated in the middle and right panels. Both of these are valid interpretations of the error syndrome, differing by a logical Z chain. In Fig. 3, we use this value for postse-

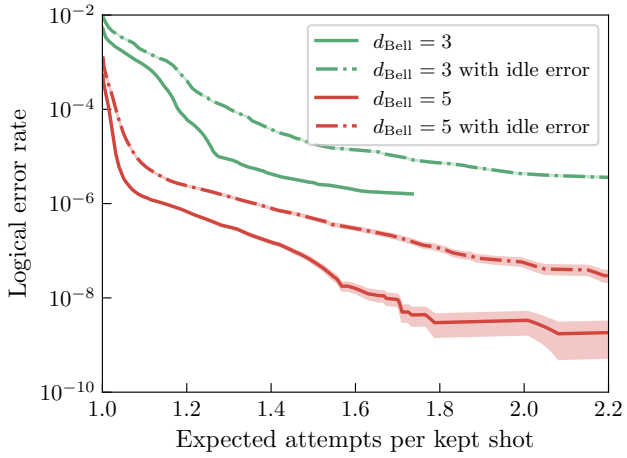

FIG. A4. Effect of idling errors on the logical error rates of logical Bell pairs generated by entanglement boosting. Here, we set $p_{\mathrm{Bell}} = 1\%$ and $d_s = 19$, and physical operation errors shown in Table A2.

lection; that is, if the complementary gap is smaller than the chosen threshold, the distillation attempt is rejected.

3. Effect of idling errors

We have neglected the effect of idling errors throughout this work, which is justifiable for reconfigurable qubits such as neutral atoms and trapped ions with coherence times of orders of magnitude longer than typical gate times (for example, for neutral atoms, the coherence times are on the order of seconds while the Rydberg gate times are on the order of 100 ns). In Fig. A4, we plot the logical error rates of Bell pairs from the entanglement boosting protocol with and without idling error at p =0.1%. Here, we use $p_{\mathrm{Bell}} = 1\%$ and $d_s = 19$, the same configuration as Fig. 4a. In the presence of idle errors, the required number of attempts increases by up to 40% to reach the same logical error rates, for the range of parameters shown in this plot.

4. Additional circuit-level simulation data and the scaling of logical error rate

Here, we show additional data from the circuit-level simulation of the entanglement boosting protocol. In Fig. A5, we show the logical error rates for varying d_{Bell}, p_{Bell}, and discard rates 0, 10, 35, 50, and 70%. With small d_{Bell} and p_{Bell}, we observe no gap values that can be used to discard large fractions, such as $d_{\mathrm{Bell}} = 3$ with discard rates of 50% and 70%, based on the 10^8 sampling that we performed. Therefore, we have not shown data with $d_{\mathrm{Bell}} = 3$ in Fig. A5d-e. A notable feature of Fig. A5a-c is the saturation of the error rates for $p_{\mathrm{Bell}} < 1\%$ (vertical dashed line); as such, we perform the fits with Eq. (2) only for $1\% < p_{\mathrm{Bell}} < 8\%$.

5. Remote lattice surgery protocol for rotated surface code

We perform numerical simulations of the remote lattice surgery by adopting the protocol of Ref. [25] for a rotated surface code, following the syndrome extraction schedule proposed in Refs. [26] and using the same error model as that used for the simulation of entanglement boosting, summarized in Table A2. This allows a fair LLV comparison between the two protocols, as shown in Fig. 6.

More concretely, we simulate logical Bell pair generation via the splitting of a merged patch [40], where the inter-patch remote physical CNOT gates are implemented by gate teleportation, and the resource state is the input physical Bell pairs with error rates p_{Bell} [25, 26]. Local operations follow the error model of Table A2, and the boundary condition follows the 'zig-zag interface' of Ref. [26], illustrated in Fig. A6, which avoids the hook errors arising in other configurations [26]. For a simulation with a distance-d_s rotated surface code, we prepare a merged code patch with an X boundary of distance d_s and a Z boundary of distance $2d_s + 1$, prepared in $|0\rangle^{\otimes(2n+d_s)}$. Next, we perform d_s cycles of syndrome extraction on the merged patch, followed by the split operation, which measures the linking region in the Z basis. Finally, we perform a noiseless syndrome extraction cycle on split patches, followed by noiseless $\overline{ZZ}$ and $\overline{XX}$ measurements, obtaining the logical error rate. The simulation is implemented with stim [64], and decoding is performed by PyMatching [65]. The resulting error scaling is plotted in Fig. A7, showing the remote Bell pair error threshold of 15.3% for the case of $p = 0.1\%$ noise strength for local physical operations. To analyze the required distance to achieve a certain logical error rate, we obtain the thresholds for both local and remote error rates, following the analysis in Ref. [25], and use the scaling,

$$p_{\mathrm{out}} = \kappa(d_s+1)^{\eta}\left(\left(\frac{p_{\mathrm{Bell}}}{p_{\mathrm{th}}^{\mathrm{Bell}}}\right)^{(d_s+1)/2} + \left(\frac{p}{p_{\mathrm{th}}^{\mathrm{local}}}\right)^{(d_s+1)/2} + \sum_{\gamma_S=1}^{d}\left[\frac{p_{\mathrm{Bell}}}{p_{\mathrm{th}}^{\mathrm{Bell}}}\left(1+\alpha_c p_{\mathrm{local}}\frac{p_{\mathrm{th}}^{\mathrm{Bell}}}{1-\sqrt{p/p_{\mathrm{th}}^{\mathrm{local}}}}\right)^2\right]^{\gamma_S/2}\left[\frac{p}{p_{\mathrm{th}}^{\mathrm{local}}}\right]^{\frac{d_s+1-\gamma_S}{2}}\right) \quad \text{(A1)}$$

where $\kappa(d_s+1)^{\eta}$ is the approximation of poly(L) shown in Ref. [25], and for odd d_s, the threshold values are $p_{\mathrm{th}}^{\mathrm{local}} =$

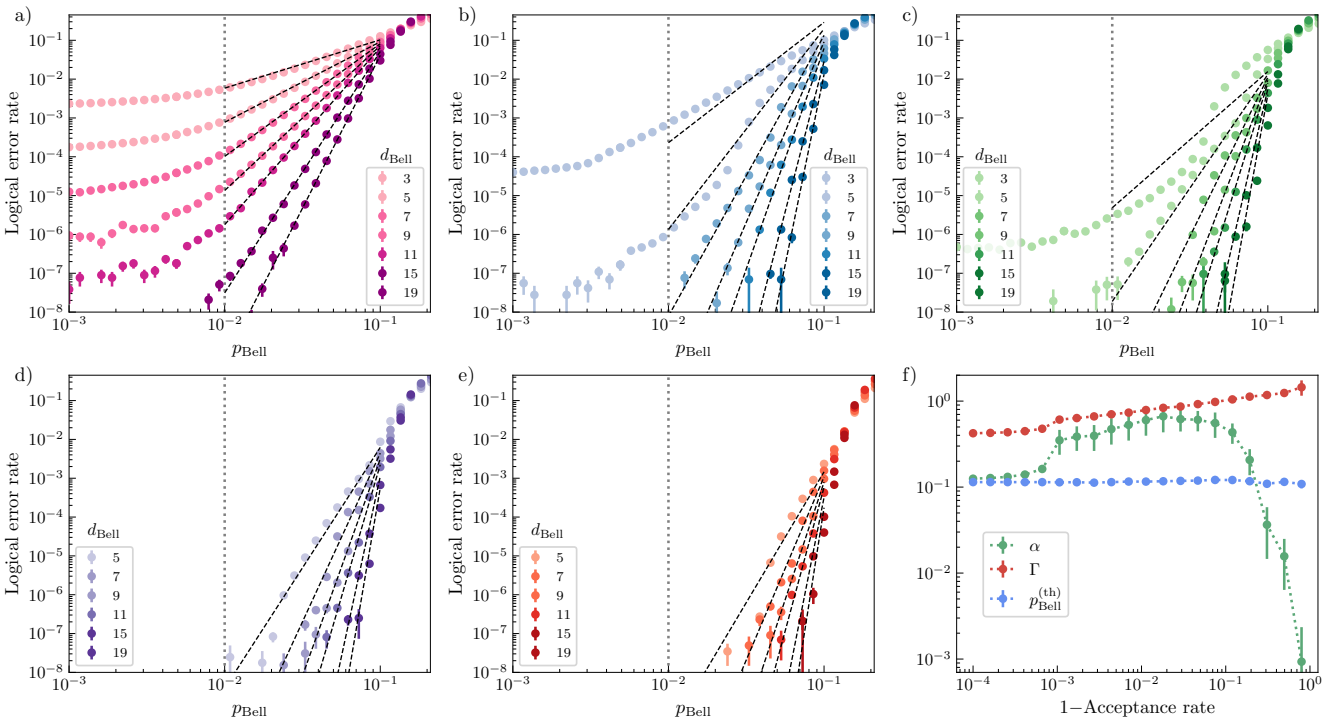

FIG. A5. Additional plots showing the scaling of the logical error rate for logical Bell pairs produced with the entanglement boosting protocol. a-e) logical error rate as a function of the physical Bell-pair error rate p_{Bell} for different d_{Bell}, with discard probabilities of 0% (a), 10% (b), 35% (c), 50% (d) and 70 % (e), based on the complementary gap. Each point is obtained from circuit-level simulations of the entanglement boosting protocol with soft-output decoding and postselection based on varying complementary-gap thresholds that result in different discard probabilities. The dashed line represents the fitting with Eq. (2) in the range $10^{-2} < p_{\mathrm{Bell}} < 9 \times 10^{-2}$. In panels d and e, the statistics of the $d_{\mathrm{Bell}} = 3$ simulation results are insufficient to obtain cases with complementary-gap-based discard rates reaching 50% or above; thus, no points are shown for $d_{\mathrm{Bell}} = 3$. f) the dependence of the fitted values of parameters α, Γ and $p_{\mathrm{Bell}}^{(\mathrm{th})}$ on the discard probabilities of entanglement boosting.

0.0102 and $p_{\mathrm{th}}^{\mathrm{Bell}} = 0.153$ and $p_{\mathrm{th}}^{\mathrm{Bell}} = 0.198$, obtained from separate simulations. The fitted values are $\kappa = 5.44 \times 10^{-2}$, $\eta = 5.34 \times 10^{-1}$ and $\alpha_c = 3.15 \times 10^{2}$ obtained from fitting shown in Fig. A7.

Appendix B: Entanglement distillation with quantum parity codes

Here, we perform simplified evaluations of the scaling of entanglement distillation with quantum parity codes C_m, which we use for the combined boosting+distillation protocol in Fig. 9. For distillation based on the $[[2m, 2m-2, 2]]$ quantum parity code, we first prepare $2m$ ideal Bell pairs and subject one of the arms of the pairs individually to single-qubit Pauli errors with probability p_{in}, where we apply Pauli X, Y or Z with probability $p_{\mathrm{in}}/3$ each, and assign the $2m$ endpoints to A and the $2m$ remaining endpoints of the pairs to B. Both A and B run their respective input states through an ideal distillation circuit locally and perform measurements in the X and Z bases. Conditioned on the measurement patterns being the same for A and B, we obtain $2m-2$ output pairs, and the probability of such measurement patterns provides the success probability of the distillation protocol. To obtain the output error rates, we track the propagation of Bell pair Pauli errors through the distillation circuit using the tableau simulator of stim [64] and identify the leading-order error probabilities of the postselected output Bell pairs. The error rates of the postselected output states depend on the code size $2m$, and we fit the output error rate as a function of m and p_{in} with a polynomial model, which yields

$$p_{\mathrm{out}} = 0.69 m^{1.36} p_{\mathrm{in}}^2. \tag{B1}$$

This is used for the evaluation of the combined boosting+distillation protocol, shown in Fig. 9.

Appendix C: Volume calculations

In this section, we describe the details of LLV used for Fig. 6, and Eq. (6) used for Fig. 9.

1. Remote lattice surgery

For the lattice-surgery-based protocol described in Appendix A 5, $2d_s - 1$ Bell pairs are consumed in each SE cycle,

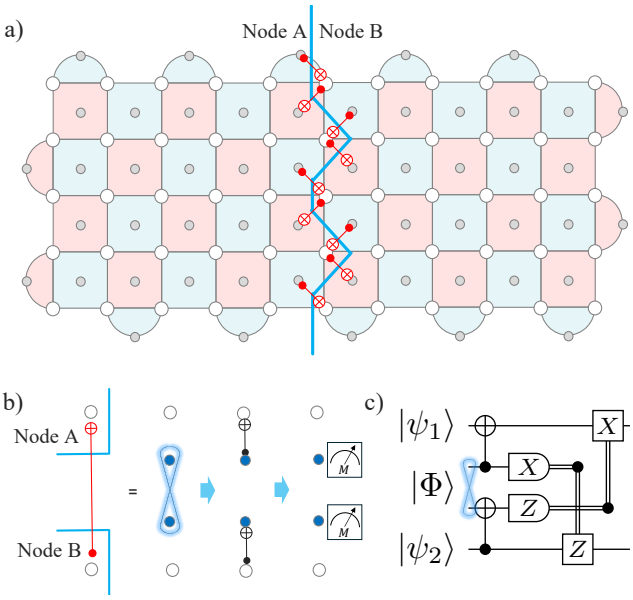

FIG. A6. Remote lattice surgery with rotated surface code. a) The zig-zag interface for remote lattice surgery of a merged patch [26], where the blue line represents the interface between the two nodes and red CNOTs represent the teleported CNOT gates implemented by Bell pairs. b-c) Gate teleportation by physical Bell pairs. Remote Bell pair (blue circles) interacts with qubits in each node (white circles) via CNOT gates, before the Bell pair is measured for feedforward Pauli gates, as shown in c).

repeated for d_s cycles, with a total of $d_s(2d_s-1)$ physical Bell pairs consumed. If the physical Bell pair generation throughput cannot keep up with the consumption speed, i.e., $R < (2d_s-1)$, then it is necessary to accumulate Bell pairs before initiating this protocol; for this, we only need to prepare $N = \max[d_s(2d_s-1) - Rd_s, 0]$, accounting for the number of Bell pairs generated during the SE cycles. Therefore, total

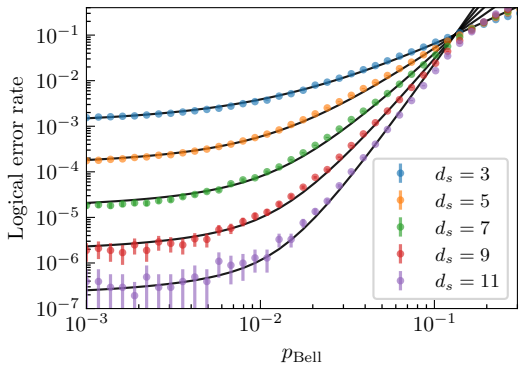

FIG. A7. Circuit-level simulation results of the logical error rate of logical Bell pairs generated by remote lattice surgery protocol, as a function of the physical Bell pair error rates p_{Bell}, with noise model of Table A2. Solid curves are the fits to the data below the threshold with Eq. (A1). Error bars represent the standard error of estimated probability from the 10^7 sampling results.

LLV is

$$\mathcal{V}_{\mathrm{surgery}} = \frac{N^2}{R} + d_s \times \left[(2d_s^2-1) + \frac{2d_s-1}{2}\right], \tag{C1}$$

where the first term represents the spacetime volume for Bell pair accumulation, the second represents the consumption of the accumulated pairs over d_s cycles, and the third is the volume for local operations; the second term in the square brackets is the additional boundary qubits for lattice surgery, with a factor of 1/2 representing the boundary qubit cost $2d_s-1$ split over the two parties involved.

2. Entanglement boosting

As discussed in Sec. IV, the boosting stage proceeds by first preparing d_{Bell}^2 physical Bell pairs and $d_s^2 - d_{\mathrm{Bell}}^2$ qubits, followed by d_s cycles of SEs. To initiate this protocol, we first wait for d_{Bell}^2/R cycles, during which d_{Bell}^2 physical Bell pairs are accumulated (Fig. 2). The single-trial LLV is hence

$$\mathcal{V}_0 = \frac{d_{\mathrm{Bell}}^4}{R} + d_s \times (2d_s^2-1). \tag{C2}$$

For the acceptance rate of q_0, the resulting LLV is $\mathcal{V}_{\mathrm{boosting}} = \mathcal{V}_0/q_0$.

3. Pipelined entanglement distillation

The general expression in Eq. (6) for the pipelined entanglement distillation with qubit reconfigurations is derived as follows. First, $\mathcal{V}_k$ corresponds to the k output patches that remain stationary throughout the circuit in Fig. 7e–f. Each of these patches interacts with up to $(n-k)$ incoming qubits through transversal CNOT layers before being moved out from the factory, and the associated spacetime volume is obtained by multiplying this layer count by the volume of d_s cycles of syndrome extraction (SE) for a single patch. Next, $\mathcal{V}_r$ accounts for the r patches that are sequentially reconfigured across the other $(n-r)$ patches, as illustrated in Fig. 7e–f. Each moving patch performs up to $(n-r)$ transversal CNOT layers during its traversal, again followed by d_s SE cycles per layer, yielding the contribution $\mathcal{V}_r$. Finally, the remaining $(n-r-k)$ patches interact with both the traversing r patches and the k output patches. During the first stage, they undergo up to r transversal CNOT gates with the moving patches, followed by additional layers with the rest of $(n-r-k)$ patches and output patches, up to $(n-r-k-1)$ and k CNOT layers, respectively, as depicted in Fig. 7c–f. Summing these contributions gives the total spacetime volume in Eq. (6).

4. Concatenated entanglement distillation

In Fig. 9a, we show the LLV for concatenated entanglement distillation in Ref. [30] to achieve the output logical error rates of 10^{-12}, with buffer space (the space allocated for logical

entanglement distillation) of 10 rotated surface code patches, which is the same space required for the boosting+distillation with the $[[10, 8, 2]]$ code shown in Fig. 9. For this evaluation, we assume the use of $d_s = 19$ rotated surface code, which is the same as the boosting and boosting+distillation protocols evaluated in Fig. 9a. For an input Bell pair error rate of 1%, the optimized code sequence identified in Ref. [30] is $[3, 1, 3]_X$, $[2, 1, 2]_Y$, $[2, 1, 2]_X$, and$[[6, 4, 2]]$, where $[n, 1, n]$ denotes the classical repetition code used for entanglement purification [7] along a specific basis (subscripts). To evaluate the LLV, we first note that the number of logical CNOTs required to perform entanglement purification with the $[2, 1, 2]$ code is 1, and 2 for $[3, 1, 3]$. For the highest level of concatenation with $[[6, 4, 2]]$ code, we use the circuit volume model for the pipelined implementation (Eq. (6)). We obtain the success rates of state injection and each distillation step from the open-source code accompanying Ref. [30], and calculate the required number of distillation trials at each concatenation level to achieve a single successful output. Adding the volumes described above, along with an additional term for the buffering cost of Bell pairs ($\mathcal{V}_b$ in Fig. 2), we obtain the LLV of the injection-distillation approach shown in Fig. 6. We remark that, for simplicity, the above evaluation for the concatenated protocol in Ref. [30] neglects the additional spacetime volume associated with inter-level buffering required while accumulating the outputs of lower level distillations for use in the next concatenation level, which may be significant for operations within a limited space; such costs are ignored by assuming that all required distillation instances operate in parallel.

[1] C. Panayi, M. Razavi, X. Ma, and N. Lütkenhaus, Memory-assisted measurement-device-independent quantum key distribution, New Journal of Physics **16**, 043005 (2014).

[2] K. Azuma, S. E. Economou, D. Elkouss, P. Hilaire, L. Jiang, H.-K. Lo, and I. Tzitrin, Quantum repeaters: From quantum networks to the quantum internet, Rev. Mod. Phys. **95**, 045006 (2023).

[3] J. F. Fitzsimons, Private quantum computation: an introduction to blind quantum computing and related protocols, npj Quantum Information **3**, 23 (2017).

[4] D. Gottesman, T. Jennewein, and S. Croke, Longer-baseline telescopes using quantum repeaters, Phys. Rev. Lett. **109**, 070503 (2012).

[5] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects, Phys. Rev. A **89**, 022317 (2014).

[6] S. Sunami, S. Tamiya, R. Inoue, H. Yamasaki, and A. Goban, Scalable networking of neutral-atom qubits: Nanofiber-based approach for multiprocessor fault-tolerant quantum computers, PRX Quantum **6**, 010101 (2025).

[7] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Purification of noisy entanglement and faithful teleportation via noisy channels, Phys. Rev. Lett. **76**, 722 (1996).

[8] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, Quantum entanglement, Rev. Mod. Phys. **81**, 865 (2009).

[9] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, Quantum privacy amplification and the security of quantum cryptography over noisy channels, Phys. Rev. Lett. **77**, 2818 (1996).

[10] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Mixed-state entanglement and quantum error correction, Phys. Rev. A **54**, 3824 (1996).

[11] W. Dür and H. J. Briegel, Entanglement purification and quantum error correction, Reports on Progress in Physics **70**, 1381 (2007).

[12] L. J. Stephenson, D. P. Nadlinger, B. C. Nichol, S. An, P. Drmota, T. G. Ballance, K. Thirumalai, J. F. Goodwin, D. M. Lucas, and C. J. Ballance, High-rate, high-fidelity entanglement of qubits across an elementary quantum network, Phys. Rev. Lett. **124**, 110501 (2020).

[13] D. Main, P. Drmota, D. P. Nadlinger, E. M. Ainley, A. Agrawal, B. C. Nichol, R. Srinivas, G. Araneda, and D. M. Lucas, Distributed quantum computing across an optical network link, Nature **638**, 383 (2025).

[14] Y. Li and J. D. Thompson, High-rate and high-fidelity modular interconnects between neutral atom quantum processors, PRX Quantum **5**, 020363 (2024).

[15] J. Sinclair, J. Ramette, B. Grinkemeyer, D. Bluvstein, M. D. Lukin, and V. Vuletić, Fault-tolerant optical interconnects for neutral-atom arrays, Phys. Rev. Res. **7**, 013313 (2025).

[16] D. Litinski, A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery, Quantum **3**, 128 (2019).

[17] 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).

[18] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, Quantum **5**, 433 (2021).

[19] D. Litinski, Magic State Distillation: Not as Costly as You Think, Quantum **3**, 205 (2019).

[20] T. Itogawa, Y. Takada, Y. Hirano, and K. Fujii, Efficient magic state distillation by zero-level distillation, PRX Quantum **6**, 020356 (2025).

[21] C. Gidney, N. Shutty, and C. Jones, Magic state cultivation: growing t states as cheap as cnot gates (2024), arXiv:2409.17595 [quant-ph].

[22] Y. Hirano, T. Itogawa, and K. Fujii, Leveraging zero-level distillation to generate high-fidelity magic states, in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 01 (2024) pp. 843–853.

[23] Z.-H. Chen, M.-C. Chen, C.-Y. Lu, and J.-W. Pan, Efficient magic state cultivation on $\mathbb{RP}^2$ (2025), arXiv:2503.18657 [quant-ph].

[24] C. Gidney, How to factor 2048 bit rsa integers with less than a million noisy qubits (2025), arXiv:2505.15917 [quant-ph].

[25] J. Ramette, J. Sinclair, N. P. Breuckmann, and V. Vuletíc, Fault-tolerant connection of error-corrected qubits with noisy links, npj Quantum Information **10**, 58 (2024).

[26] M. A. Shalby, R. Wang, D. Sedov, and L. P. Pryadko, Optimized noise-resilient surface code teleportation interfaces, Phys. Rev. A **112**, L020403 (2025).

[27] S. Glancy, E. Knill, and H. M. Vasconcelos, Entanglement purification of any stabilizer state, Phys. Rev. A **74**, 032319 (2006).

[28] Y. Maeda, Y. Suzuki, T. Kobayashi, T. Yamamoto, Y. Tokunaga, and K. Fujii, Logical entanglement distribution between distant

2d array qubits (2025), arXiv:2503.14894 [quant-ph].
[29] J. P. Bonilla Ataides, H. Zhou, Q. Xu, G. Baranes, B. Li, M. D. Lukin, and L. Jiang, Constant-overhead fault-tolerant bell-pair distillation using high-rate codes, Phys. Rev. Lett. **135**, 130804 (2025).
[30] C. Pattison, G. Baranes, J. P. Bonilla Ataides, M. D. Lukin, and H. Zhou, Constant-rate entanglement distillation for fast quantum interconnects, in *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, ISCA '25 (Association for Computing Machinery, New York, NY, USA, 2025) p. 257–270.
[31] H. Bombin and M. A. Martin-Delgado, Optimal resources for topological two-dimensional stabilizer codes: Comparative study, Phys. Rev. A **76**, 012305 (2007).
[32] V. Zapatero, T. van Leent, R. Arnon-Friedman, W.-Z. Liu, Q. Zhang, H. Weinfurter, and M. Curty, Advances in device-independent quantum key distribution, npj quantum information **9**, 10 (2023).
[33] F. Meier and H. Yamasaki, Energy-consumption advantage of quantum computation, PRX Energy **4**, 023008 (2025).
[34] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A **54**, 1098 (1996).
[35] A. Steane, Multiple-particle interference and quantum error correction, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452**, 2551 (1996).
[36] R. Matsumoto, Conversion of a general quantum stabilizer code to an entanglement distillation protocol, Journal of Physics A: Mathematical and General **36**, 8113 (2003).
[37] E. Hostens, J. Dehaene, and B. D. Moor, The equivalence of two approaches to the design of entanglement distillation protocols (2004), arXiv:quant-ph/0406017 [quant-ph].
[38] D. Bluvstein, H. Levine, G. Semeghini, T. T. Wang, S. Ebadi, M. Kalinowski, A. Keesling, N. Maskara, H. Pichler, M. Greiner, V. Vuletić, and M. D. Lukin, A quantum processor based on coherent transport of entangled atom arrays, Nature **604**, 451 (2022).
[39] L.-M. Duan and C. Monroe, Colloquium: Quantum networks with trapped ions, Rev. Mod. Phys. **82**, 1209 (2010).
[40] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New Journal of Physics **14**, 123011 (2012).
[41] K. Sahay, Y. Lin, S. Huang, K. R. Brown, and S. Puri, Error correction of transversal cnot gates for scalable surface-code computation, PRX Quantum **6**, 020326 (2025).
[42] C. Gidney, M. Newman, P. Brooks, and C. Jones, Yoked surface codes, Nature Communications **16**, 4498 (2025).
[43] J. O'Reilly, G. Toh, I. Goetting, S. Saha, M. Shalaev, A. L. Carter, A. Risinger, A. Kalakuntla, T. Li, A. Verma, and C. Monroe, Fast photon-mediated entanglement of continuously cooled trapped ions for quantum networking, Phys. Rev. Lett. **133**, 090802 (2024).
[44] B. Lekitsch, S. Weidt, A. G. Fowler, K. Mølmer, S. J. Devitt, C. Wunderlich, and W. K. Hensinger, Blueprint for a microwave trapped ion quantum computer, Science Advances **3**, e1601540 (2017).
[45] M. E. Beverland, P. Murali, M. Troyer, K. M. Svore, T. Hoefler, V. Kliuchnikov, G. H. Low, M. Soeken, A. Sundaram, and A. Vaschillo, Assessing requirements to scale to practical quantum advantage (2022), arXiv:2211.07629 [quant-ph].
[46] P. Magnard, S. Storz, P. Kurpiers, J. Schär, F. Marxer, J. Lütolf, T. Walter, J.-C. Besse, M. Gabureac, K. Reuer, A. Akin, B. Royer, A. Blais, and A. Wallraff, Microwave quantum link between superconducting circuits housed in spatially separated cryogenic systems, Phys. Rev. Lett. **125**, 260502 (2020).
[47] L. Hartung, M. Seubert, S. Welte, E. Distante, and G. Rempe, A quantum-network register assembled with optical tweezers in an optical cavity, Science **385**, 179 (2024).
[48] H. Bombín, M. Pant, S. Roberts, and K. I. Seetharam, Fault-tolerant postselection for low-overhead magic state preparation, PRX Quantum **5**, 010302 (2024).
[49] Y. Li, A magic state's fidelity can be superior to the operations that created it, New Journal of Physics **17**, 023037 (2015).
[50] H. Chen, D. Xu, G. M. Sommers, D. A. Huse, J. D. Thompson, and S. Gopalakrishnan, Scalable accuracy gains from postselection in quantum error correcting codes (2025), arXiv:2510.05222 [cond-mat.stat-mech].
[51] D. Gottesman, Stabilizer codes and quantum error correction (1997), arXiv:quant-ph/9705052 [quant-ph].
[52] D. Gottesman, Theory of fault-tolerant quantum computation, Phys. Rev. A **57**, 127 (1998).
[53] A. M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett. **77**, 793 (1996).
[54] A. M. Steane, Simple quantum error-correcting codes, Phys. Rev. A **54**, 4741 (1996).
[55] H. Yamasaki and M. Koashi, Time-efficient constant-space-overhead fault-tolerant quantum computation, Nature Physics **20**, 247 (2024).
[56] J.-P. Tillich and G. Zémor, Quantum ldpc codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Transactions on Information Theory **60**, 1193 (2014).
[57] A. Leverrier, J.-P. Tillich, and G. Zémor, Quantum expander codes, in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science* (2015) pp. 810–824.
[58] M. Grassl, T. Beth, and T. Pellizzari, Codes for the quantum erasure channel, Phys. Rev. A **56**, 33 (1997).
[59] S. Kikura, R. Inoue, H. Yamasaki, A. Goban, and S. Sunami, Taming recoil effect in cavity-assisted quantum interconnects (2025), arXiv:2502.14859 [physics.atom-ph].
[60] Y. Wu, S. Kolkowitz, S. Puri, and J. D. Thompson, Erasure conversion for fault-tolerant quantum computing in alkaline earth Rydberg atom arrays, Nature Communications **13**, 4657 (2022).
[61] S. Kikura, K. Tanji, A. Goban, and S. Sunami, Passive quantum interconnects: High-fidelity quantum networking at higher rates with less overhead (2025), arXiv:2507.01229 [quant-ph].
[62] H. Zhou, C. Zhao, M. Cain, D. Bluvstein, N. Maskara, C. Duckering, H.-Y. Hu, S.-T. Wang, A. Kubica, and M. D. Lukin, Low-overhead transversal fault tolerance for universal quantum computation, Nature **646**, 303 (2025).
[63] S. Sunami, A. Goban, and H. Yamasaki, Transversal surface-code game powered by neutral atoms (2025), arXiv:2506.18979 [quant-ph].
[64] C. Gidney, Stim: a fast stabilizer circuit simulator, Quantum **5**, 497 (2021).
[65] O. Higgott and C. Gidney, Sparse Blossom: correcting a million errors per core second with minimum-weight matching, Quantum **9**, 1600 (2025).