# Analog Information Decoding of Bosonic Quantum Low-Density Parity-Check Codes

(article starts on next page)

# Analog Information Decoding of Bosonic Quantum Low-Density Parity-Check Codes

Lucas Berent[1,*,‖] Timo Hillmann[2,†,‖] Jens Eisert,[3,4,‡] Robert Wille,[1,5,§] and Joschka Roffe[3,6,¶]

[1]*Technical University of Munich, Germany*

[2]*Department of Microtechnology and Nanoscience (MC2), Chalmers University of Technology, Gothenburg SE-412 96, Sweden*

[3]*Freie Universität Berlin, Germany*

[4]*Helmholtz-Zentrum Berlin für Materialien und Energie, Germany*

[5]*Software Competence Center Hagenberg, Austria*

[6]*Quantum Software Lab, University of Edinburgh, Scotland*

Quantum error correction is crucial for scalable quantum information-processing applications. Traditional discrete-variable quantum codes that use multiple two-level systems to encode logical information can be hardware intensive. An alternative approach is provided by bosonic codes, which use the infinite-dimensional Hilbert space of harmonic oscillators to encode quantum information. Two promising features of bosonic codes are that syndrome measurements are natively analog and that they can be concatenated with discrete-variable codes. In this work, we propose novel decoding methods that explicitly exploit the analog syndrome information obtained from the bosonic qubit readout in a concatenated architecture. Our methods are versatile and can be generally applied to any bosonic code concatenated with a quantum low-density parity-check (QLDPC) code. Furthermore, we introduce the concept of quasi-single shot protocols as a novel approach that significantly reduces the number of repeated syndrome measurements required when decoding under phenomenological noise. To realize the protocol, we present the first implementation of time-domain decoding with the overlapping window method for general QLDPC codes and a novel analog single-shot decoding method. Our results lay the foundation for general decoding algorithms using analog information and demonstrate promising results in the direction of fault-tolerant quantum computation with concatenated bosonic-QLDPC codes.

## I. INTRODUCTION

For quantum computing, an important design consideration is the choice of technology used to physically realize qubits. A standard approach is to engineer *discrete-variable* (DV) qubits, where the basis states are defined by two distinct energy levels of the quantum system.

---

*Corresponding author: lucas.berent@tum.de

†Corresponding author: timo.hillmann@rwth-aachen.de

‡Corresponding author: jense@zedat.fu-berlin.de

§Corresponding author: robert.wille@tum.de

¶Corresponding author: joschka@roffe.eu

‖These authors contributed equally.

Examples include ion-trap qubits [1,2], superconducting qubits [3], and spin qubits [4,5]. However, discrete-variable qubits pose challenges, primarily in the intricate task of effectively isolating the two-level encoding from external influences that introduce errors. There is substantial evidence that, in the absence of error-suppression methods, the coherence of quantum circuits is limited to at most a logarithmic depth [6,7], which represents a challenge for near-term quantum computing. To suppress qubit errors, we can resort to notions of *quantum error correction* (QEC). The goal of QEC is to harness entanglement to redundantly encode quantum information in the logical qubit state of a larger physical Hilbert space, albeit to the extent of substantial overhead. The QEC encoding provides the system with additional degrees of freedom that can be used to detect and correct errors in real time. Beyond the initial encoding, full QEC protocols must incorporate logical gates that allow the encoded quantum information to be manipulated in a fault-tolerant way. This provides the ability to compute fault tolerantly,

whilst keeping the system protected against local noise. Moreover, given a (potentially corrupted) encoded state, a central task is to check whether errors occurred on the encoded information, and if so to decode, i.e., to computationally derive a suitable recovery operation to restore an error-free state. Identifying and realizing feasible and practical schemes for fault-tolerant operation remains one of the core challenges of quantum computing, both in experiment and in theory.

On the highest level, both discrete-variable and *continuous-variable* (CV) codes for quantum error correction have been considered in the literature and are seen as candidates for feasible quantum codes. The latter offer some compelling advantages for a number of platforms, notably for *cat codes* [8] and *Gottesman-Kitaev-Preskill* (GKP) codes [9]. However, there are also some challenges that are unique to the continuous-variable setting. In particular, it is not obvious how to do decoding in light of continuous syndrome information, as it is not clear how to make use of continuous information in this task. For example, the development of good decoders for GKP codes constitutes a well-known technical challenge. The lack of good methods of decoding for quantum error-correcting codes with a continuous component can be seen as a roadblock in the field.

In this work, we report substantial progress on the partial use of continuous syndrome information in the notions of quantum error correction. In this way, we aim to bring the advantages of DV and CV quantum error correction closer together. To this end, we explore the combination of two promising classes of codes, instances of bosonic codes and quantum *low-density parity-check* (LDPC) codes, and investigate suitable decoding protocols for general bosonic-LDPC constructions that make such partial use of analog information. To further motivate this combination and to show how continuous syndrome information comes into play, we give a brief overview in the following.

### A. Bosonic qubits

Bosonic encodings offer an alternative to DV qubits [10,11]. In this approach, the logical qubit state is non-locally embedded within the phase space of an infinite-dimensional quantum harmonic oscillator. The principal advantage of using bosonic qubits is that the infinite-dimensional Hilbert space provides the redundancy needed to correct physical oscillator errors. Consequently, bosonic encodings can be interpreted as intrinsic QEC protocols at the individual qubit level. In principle, this provides an efficient route to fault tolerance, and bosonic codes have been extensively explored using protocols such as GKP codes [9] and cat codes [8].

Similarly to discrete-variable qubits, multiple bosonic qubits can be combined via a QEC code to create a logical state. This strategy is usually referred to as a *concatenated* bosonic code: at the inner level, the individual qubits are protected by their bosonic encoding, and at the outer level, the bosonic qubits are wired together to form a logical state. Concatenated codes therefore combine the benefits of both bosonic and discrete-variable QEC codes.

Another appealing feature of various bosonic codes is that they can be precisely tuned to exhibit noise asymmetry in their qubit-level error model. For example, recent experiments have shown that cat code qubits can be engineered to have phase-flip rates that dominate over bit flips by almost 3 orders of magnitude [12,13]. In a concatenated bosonic code, highly biased inner-level qubits can reduce the overhead required by the outer code. For example, in recent work by Darmawan *et al.* [14], it is proposed that cat code qubits can be concatenated with the *XZZX* code [15], an instance of a surface code modified by Clifford conjugations that has extremely high thresholds in the limit of large bias [16].

### B. Concatenated bosonic codes

To date, most studies of bosonic codes have focused on their concatenation with repetition codes [17–21] or two-dimensional topological codes [14,22–24]. Such codes are favored for near-term experiments because they require only nearest-neighbor interactions. This facilitates their implementation on a two-dimensional array of qubits, making them particularly suitable for architectures such as superconducting qubits. However, from an information theoretical standpoint, two-dimensional topological codes may be seen as being far from optimal. The main drawback lies in their poor rate: the surface code, for instance, encodes only a single logical qubit per patch. This means that increasing the surface code distance comes at the expense of the encoding density. This is in stark contrast to the efficiency of contemporary classical error correction protocols. In particular, numerous classical communication technologies employ LDPC codes. Such codes have the advantage of preserving a constant encoding density even as the code distance is scaled. Moreover, it has been shown that in the asymptotic limit, LDPC codes can approach the Shannon bound, which represents the theoretical limit on the rate of information transfer through a noisy channel.

### C. Low-density parity-check quantum codes

Until recently, it was an open question as to whether *quantum LDPC* (QLDPC) codes with *good* parameter scaling comparable to their classical counterparts exist. This question has recently been answered in the affirmative via a series of theoretical breakthroughs [25–29]. Central to these innovations has been the use of sophisticated product constructions that provide procedures for transforming

classical LDPC codes into quantum codes. The resulting QLDPC codes exhibit constant encoding rates and distances that scale proportionally with the length of the code.

Implementing QLDPC codes poses greater challenges than their planar counterparts. Several no-go theorems indicate that implementing (good) QLDPC codes will not be possible in two-dimensional geometrically local architectures [30–34]: they must necessarily involve geometrically nonlocal connections. Nonetheless, various qubit technologies are being developed that enable long-range interactions needed to implement QLDPC codes [35–39]. In this setting, QLDPC codes promise quantum computation with considerably reduced overhead compared to the surface code [35,40].

In Ref. [24], Raveendran *et al.* have explored the concatenation of GKP bosonic qubits with QLDPC codes based on the lifted product construction [41]. Their results show that there are distinct advantages to using a concatenated bosonic code over directly implementing a QLDPC code with discrete variable qubits. Specifically, it is possible to feed forward analog information from the GKP qubit readout to improve the performance of the outer code's decoder. This leads to improved error suppression beyond what is achievable with discrete variable qubits alone.

In light of this promising line of research on concatenated bosonic codes, we focus on QEC protocols constructed from bosonic-LDPC codes in a general fashion, assuming only that the inner code provides analog syndrome readout and possibly a noise bias and that the outer code is an arbitrary quantum LDPC code. We focus on the decoding of such codes and demonstrate that, by combining their properties, we obtain high-performance QEC protocols.

### D. Overview of contributions

In this work, we develop methods for the decoding and analysis of concatenated bosonic-LDPC codes. To this end, our primary test bed is a protocol in which cat code qubits (inner code) are concatenated with the three-dimensional surface code (outer code). We choose to focus on the three-dimensional surface code as it is perhaps the simplest example of a QLDPC code that extends upon the capabilities of the standard two-dimensional surface code. From an implementation perspective, three-dimensional surface codes can be realized with relatively few long-range connections in two and completely locally in three dimensions [42]. Suitable experimental platforms for implementing three-dimensional codes include architectures with photonic links or neutral atom arrays [37,38]. For comparison, the concatenated lifted product schemes of Ref. [24] require arbitrary qubit connectivity in general. Another distinguishing feature of the three-dimensional surface code is the fact that it has a transversal CCZ gate

[42]. This leaves it fully equipped for universal quantum computation without the need for resource-intensive magic state injection.

We introduce a novel decoding method, *analog Tanner graph decoding* (ATD), which makes use of the analog readout information from the bosonic qubits. Our numerical simulations show that this leads to improved thresholds for decoding with the three-dimensional surface codes. Furthermore, we demonstrate that using ATD, the number of repetitions required by the non-single-shot component of the three-dimensional surface codes can be reduced to a small number. In this setting, we refer to three-dimensional surface codes as being *quasi-single shot*. Finally, this paper is accompanied by open-source software tools that facilitate the reproduction and extension of our analysis of concatenated bosonic-LDPC codes and provide means to automatically conduct respective numerical simulations. Our results are summarized in more detail below.

### E. Analog Tanner graph decoding

Fundamentally, in discrete-variable QEC, the syndromes obtained from stabilizer measurements are discrete, although readout techniques can yield an analog value. This stands in contrast to bosonic qubits, where measurements yield analog outcomes due to the infinite-dimensional Hilbert space. The strength of the analog readout can be used to assign an uncertainty associated with the measurement. Syndromes derived from analog bosonic readout are termed *analog syndromes*. The *analog Tanner graph decoder* we introduce in this work provides a method for mapping analog syndromes to a belief propagation decoder, sketched in Fig. 1(d). Our approach is extremely versatile: ATD can be applied to any stabilizer code with analog syndrome information. Furthermore, it is possible to incorporate ATD as part of both single-shot and time-domain decoding strategies, as illustrated in Fig. 1(e) and Fig. 1(f).

### F. Single-shot decoding with analog information

A problem in QEC is that syndromes must be extracted using auxiliary qubits that are also susceptible to errors. As such, there is uncertainty associated with any syndrome we measure. To counteract this problem, we can adapt the *time-domain* approach in which syndrome measurements are repeated $\propto d$ times, where $d$ is the code distance. Measurement errors can then be accounted for by considering the entire syndrome history in the decoding. An alternative strategy is to use a code that has the so-called *single-shot* property. Single-shot codes have an additional structure that allows measurement errors to be directly corrected, removing the need to decode over time [43]. The three-dimensional surface code is single-shot for phase noise but requires time-domain decoding for bit-flip noise [42,44]. Our numerical results show that under ATD
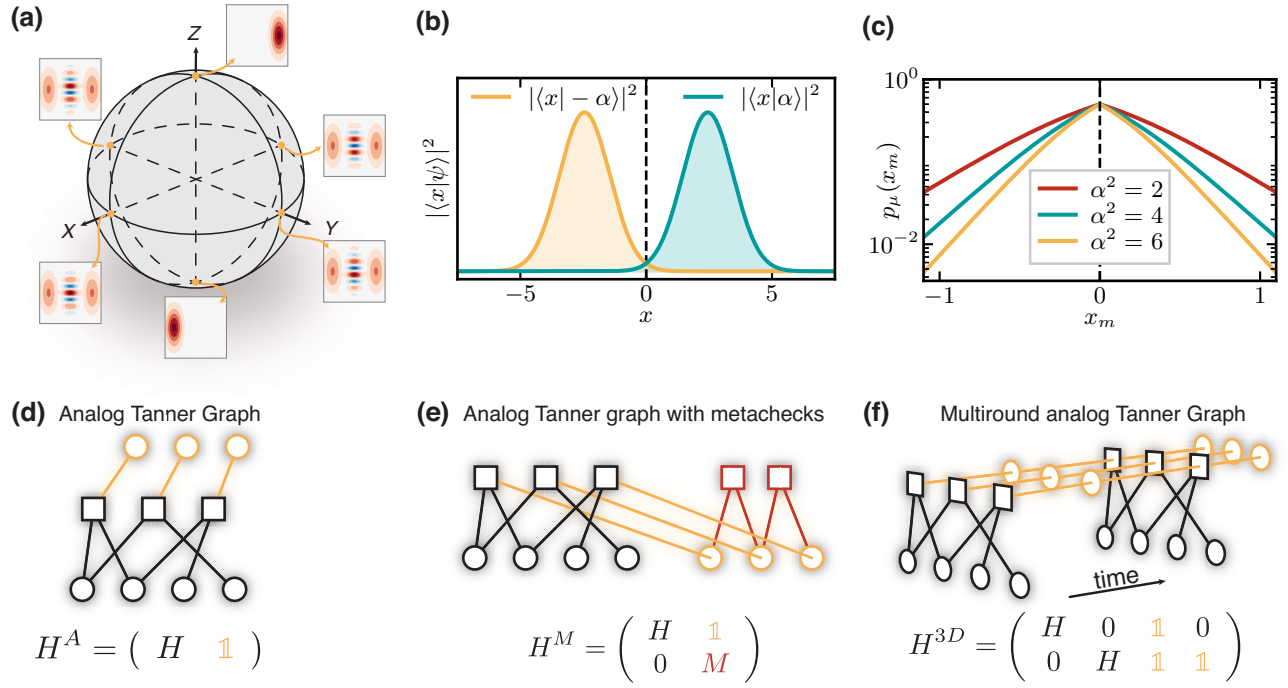
FIG. 1. Overview of our main techniques. (a) We investigate QLDPC codes concatenated with cat qubits encoded in coherent states of a harmonic oscillator. (b) An important property of cat qubits, in addition to their biased noise model, is that the syndrome information obtained from qubit readout is intrinsically analog valued, as the wave function of a coherent state is a Gaussian centered at $\alpha$. (c) Depending on the measured value $x_m$ during (quadrature) readout, we can assign an outcome-dependent error probability $p(x_m)$ that is a function of the size of the cat qubit $\alpha^2$. (d) We incorporate the analog information obtained during the syndrome measurements into a Tanner graph construction that we refer to as an *analog Tanner graph* (ATG). The ATG stores the analog syndrome information directly in the factor graph used for decoding. We show that the ATG construction can be adapted to work with decoding strategies such as (e) single-shot shot decoding and (f) overlapping window time-domain decoding.

decoding, the single-shot component of the concatenated three-dimensional surface code has a sustained threshold of 9.9% under phenomenological noise. This improves over the previous best-observed threshold for the discrete variable three-dimensional surface codes of 7.1% [45].

### G. Quasi-single shot protocol

Time-domain decoding can lead to large overheads for quantum codes, since the number of repeated syndrome measurements required is proportional to the code distance $d$ [46–48]. This increases the length of the decoding cycle and reduces the frequency with which logical operations can be applied [49]. To address this overhead, we propose a novel protocol called (*w*)-*quasi-single shot* decoding. The key idea is to use the analog information obtained during the syndrome readout of the inner bosonic code to enhance the decoding performance and reduce the need to repeat the measurement multiple times in the noisy readout setting. On an intuitive level, this idea is somewhat analogous to the fact that to learn $m$ bits during the quantum phase estimation algorithm, one needs $O(m)$ measurements [50], whereas, using bosonic qubits, one can learn the phase (in principle) using a single measurement [9].

Quasi-single shot decoding leverages our strategies for decoding with analog information, and we obtain a scheme that reduces the required number of repeated syndrome measurement rounds to a small number $w \ll d$. In combination with the tunable noise bias of the inner cat qubit code, the quasi-single shot three-dimensional surface code yields a three-dimensional topological code with significantly higher thresholds and reduced time overhead when compared to two-dimensional topological codes.

### H. Open-source software tools

We have implemented a set of software tools as part of the Munich Quantum Toolkit (MQT) to foster further research in the direction and to provide the community with numerical tools to conduct simulations and reproduce our results.

The remainder of this work is structured as follows. In Sec. II, we review the main concepts of bosonic quantum codes and quantum error correction that are needed throughout this work. In Sec. III, we discuss iterative decoding approaches that are fundamental for state-of-the-art algorithms and the proposed techniques. Then, in Sec. IV our decoding strategies for decoding quantum

codes using analog information and the respective numerical simulation results are presented. In Sec. V, we elaborate on the proposed quasi-single shot protocol and present numerical results. Section VI focuses on more practical aspects around the considered bosonic-LDPC code architectures and reviews important open challenges with respect to potential implementations in superconducting architectures. We conclude and give a brief overview of future work in Sec. VII.

## II. PRELIMINARIES

Here, we introduce basic notions of quantum coding that are needed throughout the rest of this work. We assume the reader is familiar with fundamental notions of quantum information theory and quantum error correction.

### A. Bosonic quantum codes

The use of bosonic codes toward achieving fault-tolerant quantum computing has become a promising alternative to the approach based on so-called *discrete-variable* qubits such as spin qubits, trapped ions, neutral atoms, or superconducting qubits [1–5]. It is often quoted, see, for instance, Refs. [51,52] that the advantage of bosonic codes over discrete-variable codes is the fact that a single bosonic mode, realized, e.g., in a superconducting cavity or the motional states of trapped ions, lives in an infinite-dimensional Hilbert space. Bosonic codes therefore offer a hardware-efficient route toward fault-tolerant quantum computing, since the principle of quantum error correction relies on encoding logical information redundantly in a subspace of a much larger Hilbert space, as opposed to using multiple two-level systems. The main idea of how bosonic codes can be realized in practice is that a particular bosonic code constitutes the first layer in a concatenated quantum error-correction code, consisting of (at least) one continuous-variable and one discrete-variable code, which is called *outer code* in this context. As there are various realizations and families of discrete-variable codes, there are similarly multiple families of bosonic codes. As the choice of the discrete-variable code determines properties, such as the availability of transversal gates, the choice of a particular bosonic code can be tailored to the physical platform on which the code is realized, but also determines the (effective) noise model that is relevant for the outer code as well.

As such, in addition to their large surrounding Hilbert space, bosonic codes offer further advantages over two-level systems relevant to the design of quantum error-correcting codes and decoders. One of these features is that many families of bosonic codes have a biased-noise error model, where some types of errors occur more frequently than others. While discrete-variable qubits can have a biased noise channel, this noise bias cannot be preserved by gate operations needed throughout the QEC protocol,

i.e., there does not exist a bias-preserving CNOT implementation [17]. While a Pauli-$Z$ error occurring before the gate is not converted to other types of Pauli errors, a $Z$ error during the execution of the gate will be converted to other types of Paulis. Thus, intuitively, for the CNOT $= $ CX$(\pi)$ one must require that $Z$ errors commute with the gate at *all times*, i.e., $[Z\text{CX}(\alpha)] = 0, \ \forall \alpha \in [0, \pi]$. However, implementation of the CNOT in such a way is not possible without leaving the code space [53]. Hence, the bias is annihilated during the computation and bias-tailored QEC codes such as the ones proposed in Refs. [54–57] are not beneficial for discrete-variable systems. However, the additional degrees of freedom of continuous-variable states natively allow for bias-preserving operations [17,58].

While the additional degrees of freedom of bosonic codes yield a clear advantage over two-level systems, the continuous support of states in quantum phase space has the consequence that any measurement, e.g., the measurement of a stabilizer check, of such a state is inherently imprecise. Instead of dismissing this characteristic of bosonic quantum codes as a drawback, it can equivalently be seen as a feature that yields additional information during the qubit readout, called *analog information* that can be used for decoding. We note that in practice the readout of a two-level system in finite time also yields a continuous outcome. However, with technological progress in recent years, the readout uncertainty due to the finite measurement time has become almost negligible [59]. For the interested reader, we describe the properties of stabilized cat codes more explicitly in Sec. VI.

### B. Low-density parity-check quantum codes

In the following, all vector spaces are over $\mathbb{F}_2$ unless stated otherwise. We use $[\ell]$ to denote the set $\{1, 2, \ldots, \ell\}$. A (discrete-variable, DV) $[\![n, k, d]\!]$-quantum *stabilizer code* is defined by an Abelian subgroup $S$ of the $n$-qubit Pauli group $\mathcal{P}_n$ that does not contain $-I$. The generators of $S$ are commonly called stabilizer *checks* of the code.

An important class of quantum stabilizer codes are *Calderbank-Shor-Steane* (CSS) codes. The defining feature of CSS codes is that their stabilizer generators can be split into two decoupled sets $S_X$ and $S_Z$, which contain only products of the Pauli-$X$ and Pauli-$Z$ operators, respectively. By using the (isomorphic) mapping between the $n$-qubit Pauli group (modulo global phases) and binary vector spaces [60]

$$\mathcal{P}_n / \{\pm I, \pm iI\} \cong \mathbb{F}_2^{2n}, \tag{1}$$

it is possible to represent the $S_X$ and $S_Z$ stabilizers of a CSS code as two matrices: $H_X \in \mathbb{F}_2^{r_X \times n}$ and $H_Z \in \mathbb{F}_2^{r_Z \times n}$. These matrices can be interpreted as the parity-check matrices of two classical linear codes, the first designed to correct bit flips and the second to correct phase flips. For any CSS

code, the $H_X$ and $H_Z$ matrices must satisfy the following orthogonality condition:

$$H_Z \cdot H_X^T \equiv H_X \cdot H_Z^T, \qquad (2)$$

which ensures that the $X$ and $Z$ stabilizer generators commute, i.e., their supports have even overlap. A CSS code can then be defined as a code over $\mathbb{F}_2^{2n}$ with check matrix

$$H = \begin{pmatrix} 0 & H_Z \\ H_X & 0 \end{pmatrix}. \qquad (3)$$

The CSS *syndrome s* of a qubit error $e = (e_X, e_Z)$ is defined as

$$s = (s_X, s_Z) = (H_Z \cdot e_X, H_X \cdot e_Z). \qquad (4)$$

From the syndrome equations above, it is clear that the decoding of CSS codes amounts to two independent classical decoding problems for phase flips and bit flips, respectively. However, note that decoding the two check sides independently ignores correlations, such as $Y$ errors, and thus is suboptimal in general. In the following, we will drop the subscripts, but it should be assumed that we are referring to a single error type ($X$ or $Z$) unless otherwise stated. In the language of vector spaces, the goal of *decoding* is, given a syndrome, to infer an estimate $\varepsilon$ such that $s = H \cdot \varepsilon$ that can be used to apply a *recovery operation* to restore an error-free logical state. The decoding is successful if the *residual error* $r = e + \varepsilon$ is a stabilizer, and it fails if $r$ is a logical operator.

The *Tanner graph* $\mathcal{T}(H) = (V_D \cup V_C, E)$ of a code defined by the parity-check matrix $H$ is a bipartite graph with data nodes $V_D$ and check nodes $V_C$ whose incidence matrix is $H$. That is, given $H$, $\mathcal{T}(H)$ can be constructed by creating a data node $d \in V_D$ for each column and a check node $c \in V_C$ for each row of $H$, and inserting an edge $e = (v_c, v_d) \in E$ if $H_{(c,d)} = 1$.

*Example 1 (Tanner graph of the Hamming code).* Consider the Hamming code, defined as the kernel of the parity-check matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

The corresponding Tanner graph $\mathcal{T}(H)$ is illustrated in Fig. 2.

In the context of decoding algorithms based on factorization of probability distributions, the Tanner graph is frequently referred to as *factor graph*.

A family of *quantum low-density parity-check* (QLDPC) codes refers to a stabilizer code family with the additional
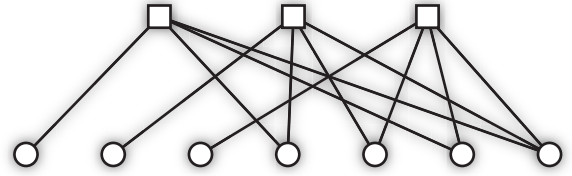


FIG. 2.    Tanner graph of the Hamming code. The square nodes represent the checks $V_C$ (rows of the check matrix $H$) and the circles represent data nodes $V_D$ (columns of $H$).

property that the stabilizer generators are sparse. More specifically, it is required that the degree of the data and check nodes in the Tanner graph is upper bounded by a constant (independent of the code size). Intuitively, each check has a constant weight and each qubit participates in a constant number of checks.

### 1. Three-dimensional surface codes

For the remainder of this work, we focus on *topological quantum codes* [61–63] as the "outer code" of a concatenated code. Such codes are derived from a geometric $D$-dimensional lattice and have local connectivity in $D$-dimensional space. In particular, we focus on the *three-dimensional surface code* (3DSC) [42,48,57,64,65] as a representative QLDPC code. Each constituent is identified with a qubit with Hilbert space $\mathbb{C}^2$. Consider a three-dimensional lattice in Euclidean space captured by a graph $\Lambda = (V, E)$ consisting of vertices $V$, edges $E$, faces $F$, and volumes $W$. For two objects in the lattice $v, f$, we write, $v \sim f$ $(f \sim v)$ to indicate that $v$ is adjacent to $f$.

To define a quantum code on the lattice $\Lambda$, we associate qubits with edges and checks with vertices and faces, i.e., we define *vertex checks $A_v, v \in V$*, acting on edges adjacent to vertices

$$A_v := \prod_{e \in E, e \sim v} Z_e \qquad (5)$$

depicted in Fig. 3(a), and *face checks $B_f, f \in F$* acting on edges adjacent to faces

$$B_f := \prod_{e \in E, e \sim f} X_e \qquad (6)$$

shown in cf. Fig. 3(b). To reason about the logical operators of the code, let us introduce some informal notation. For a more formal discussion in the language of homology, we refer the reader to Sec. A.

Let $\zeta$ be an edge path in $\Lambda$. A *string* operator is defined as a Pauli $X$ or $Z$ operator whose support corresponds to
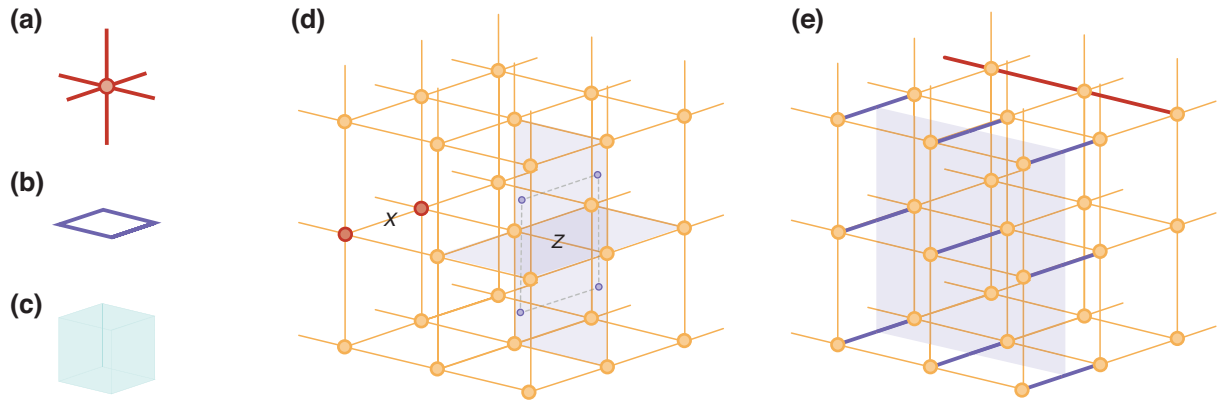
FIG. 3. Three-dimensional surface codes with periodic boundaries indicated by additional edges on the sides of the lattice. (a) A vertex check (of weight six). (b) A face check (of weight four). (c) A volume check (of weight 12). (d) Three-dimensional lattice $\Lambda$ with open boundaries. Rough boundaries are indicated with open edges. A single-qubit $X$ error gives a pairlike syndrome at the endpoints of the error string (indicated by red vertices). A single-qubit $Z$ error produces a looplike syndrome at adjacent faces (indicated by blue faces). The loop can be readily seen in the dual-lattice pictures whose dual edges are indicated with dashed lines. (e) Logical operators of the three-dimensional surface codes. A looplike string corresponds to a logical $X$ operator. A logical $Z$ operator corresponds to a loop of faces in the dual lattice, i.e., forming a *dual sheet* wrapping across the lattice along two axes.

the qubits that are associated to edges in $\zeta$ as

$$S_\zeta^P := \prod_{i \in \zeta} P_i, P \in \{X, Z\}. \tag{7}$$

The weight of a string is the size of its support, i.e., the number of nontrivial Paulis. In addition to the lattice $\Lambda$, consider also the dual lattice $\Lambda^*$, which is obtained from $\Lambda$ by associating dual vertices to volumes, dual edges to faces, dual faces to edges and dual volumes to vertices, and define *co-string operators* as string operators on $\Lambda^*$. Hence, qubits are associated with dual faces, $Z$ checks to dual volumes, and $X$ checks to dual edges.

Errors correspond to $X$ and $Z$ strings on the lattice that cause the anticommuting ($Z$ and $X$) adjacent checks to be "flagged," indicating an error occurred (all other, nonadjacent checks are not violated and thus not flagged). The syndromes caused by violated vertex checks are created in pairs at the vertices that are the endpoints of $X$-error strings, as depicted in Fig. 3(d) for a single-qubit error (i.e., a length-1 error string in $\Lambda$). The syndrome of a $Z$-error string corresponds to the adjacent violated face checks as shown in Fig. 3(d) for a single-qubit error. Syndromes caused by violated face checks are better illustrated in the dual lattice $\Lambda^*$ (recall that edges correspond to dual faces, so a string operator in $\Lambda$ corresponds to the respective collection of faces in $\Lambda^*$). Considering $\Lambda^*$ it can be seen that $Z$ syndromes have a looplike geometry in the lattice. In fact, the looplike syndromes induced by face checks lead to an additional structure that has been shown to imply the single-shot property, see also Sec. II B 2.

Given a three-dimensional lattice, we distinguish two types of boundary conditions. When $\Lambda$ has periodic boundaries (depicted by additional edges in Fig. 3), i.e., a tessellation of a three-dimensional torus (therefore also called the *three-dimensional toric code*, 3DTC), the logical $X$ operators correspond to strings that form loops on the lattice along one axis, as depicted in Fig. 3(e). The logical $Z$ operators correspond to looplike sets of faces, called *sheets* in $\Lambda^*$, i.e., sets of faces that go across the dual lattice along two axes as illustrated in Fig. 3(e). Topologically, logical operators correspond to *nontrivial* loops, i.e., loops (of edges and faces, respectively) that are noncontractible and the contractible loops (those that enclose a region on the lattice) correspond to stabilizers of the code. It is straightforward to see that there are three pairs of logical operators $\bar{X}, \bar{Z}$, corresponding to three (nonequivalent) minimum-weight loops, one along each of the three different axes, and the three corresponding orthogonal sheets. Therefore, the code encodes $k = 3$ logical qubits. The *weight* of a logical operator is the number of qubits in its support, i.e., length of $\zeta$.

If we instead consider a code on a three-dimensional lattice $\Lambda$ with open boundaries, we define two opposite sides of the lattice as $X$-type boundaries, called *smooth boundaries* and the four remaining sides as $Z$-type boundaries, called *rough boundaries*—in analogy to the two-dimensional surface code. A string operator $S_\zeta^X$ (of minimal length) that connects the smooth boundaries is a logical $X$ operator on the code, and a dual string operator $S_\zeta^Z$ connecting the four rough boundaries corresponds to a logical $Z$ operator. Since in the presence of open boundaries, there is only a single pair of such strings such that

they are orthogonal and connect the respective boundaries, the code encodes a single logical qubit.

The $X$-distance $d_X$ of the code is defined as the minimum weight of a logical $X$ operator, and analogously for $d_Z$. That is, $d_X$ is the minimum number of edges across $\Lambda$, and $d_Z$ is the minimum number of edges corresponding to a sheet across the dual lattice in the presence of periodic boundaries. For a lattice with open boundaries, $d_X$ is defined as the minimum number of edges in a string connecting smooth boundaries and $d_Z$ is defined as the minimum number of edges corresponding to a sheet connecting rough boundaries. For instance, the code depicted in Fig. 3 has distances $d_X = 3, d_Z = 9$. In summary, the *three-dimensional surface code* parameters (with open and periodic boundaries) are given by

$$\text{3DSC: } [\![2L(L-1)^2 + L^3, 1, d_X = L, d_Z = L^2]\!],$$

$$\text{3DTC: } [\![3L^3, 3, d_X = L, d_Z = L^2]\!].$$

Note that by associating $X$ checks with vertices and $Z$ checks with faces, we obtain an equivalent code, where the corresponding notions of logicals and distances are simply exchanged.

### 2. Single-shot codes

In general, the *syndrome extraction circuit* used to measure the stabilizer of the code is subject to noise. As such, syndrome errors need to be accounted for in fault-tolerant QEC protocols. Designing fault-tolerant syndrome extraction circuits can add considerable overhead. For instance, the Shor fault-tolerant scheme requires at least $\propto d^3$ check measurements [46,47]. Dennis *et al.* [48] argue that for a two-dimensional surface code with distance $d$, $O(d)$ rounds of noisy syndrome measurements need to be conducted to achieve fault tolerance.

As an alternative to time-domain decoding, Bombin [43] showed that there exist *single-shot codes* for which a single round of noisy syndrome measurements suffices. One of the main advantages of single-shot codes is that the complexity of decoding is reduced since the structure of the decoding problem does not have an additional time dimension—as is the case for non-single-shot codes. Moreover, the time needed to conduct a QEC cycle is reduced since only a single set of stabilizer measurements needs to be done, which results in the fact that more logical operations can be conducted in a time interval compared to time-domain decoding. The explicit construction of single-shot codes has been explored recently [44,66], where the central idea is to use redundancy in the checks to ensure the single-shot property. In Ref. [44], it is shown that single-shot codes with necessary redundancy can be constructed using tensor products of chain complexes (i.e., three-dimensional hypergraph product constructions). Intuitively, in this construction, the extra dimension yields an additional

(classical) code that can be used to detect syndrome errors. In this case, we can define an additional set of classical checks called *metachecks* with check matrix $M$ that defines the corresponding classical linear *metacode* $\mathcal{M}$. The metacode satisfies the condition $M \cdot H_{X/Z} = 0$, which means that every syndrome that can originate from $H_{X/Z}$ is a codeword of the metacode $\mathcal{M}$. The metacode $\mathcal{M}$ is used to determine whether a syndrome is valid, i.e., we can use the metachecks to compute a *metasyndrome* $s_m := M \cdot s_{X/Z}$, which can then be decoded to fix syndrome failures.

Geometrically, in the 3DSC, we can associate metachecks with the volumes of the lattice, as depicted in Fig. 3(c). This gives a $X$ metacode $\mathcal{M}_X$ whose checks act on the syndromes induced by the $X$-face checks. Intuitively, the volume metachecks help to close noisy looplike syndromes.

### C. Noise model

In this section, we describe the noise model that is used for the investigation of the various decoding methods (cf. Sec. III) as well as the quasi-single shot protocol (cf. Sec. V). More details about our simulation procedures can be found in Sec. G.

We consider a biased phenomenological noise model with analog syndrome measurements inspired by stabilized cat qubits. We assume a three-dimensional architecture in which bosonic cat qubits are used as the inner code and a $[\![n = 3L^3, k = 3, d_X = L, d_Z = L^2]\!]$ three-dimensional surface code with periodic boundaries is used as the outer code. i.e., the $n$ physical qubits are cat qubits, so qubits encoded in continuous-variable quantum systems with Hilbert space $\mathcal{L}(\mathbb{R}^2)$, which are used to protect $k$ logical qubits with $X$ ($Z$) distance $d_X = L$ ($d_Z = L^2$). That is to say, this noise model affects the description on two levels: the level of qubits in the outer code, and the level of continuous-variable bosonic modes in the inner code. Note that our phenomenological noise model, while capturing key features of cat qubit noise, does not explicitly correlate effective error rates to physical noise and confinement parameters. Our main motivation for using this model is to retain generality and applicability to related inner qubit noise models while capturing the key features of cat qubit noise.

In each QEC cycle, the stabilizers are measured, yielding an analog syndrome as a real number for bosonic codes. Intuitively, the magnitude of the analog measurement can be interpreted as information on the reliability of the syndrome readout and thus can be used in the overall decoding routine. Note that in this work we use a phenomenological noise model and hence do not consider a specific implementation of the syndrome extraction circuit. For example, the standard Steane scheme [67] could be used.

### 1. Qubit error noise model

Here, we outline specifics concerning the error model on the level of qubits in the outer code. We consider a model defined as a quantum channel that takes the form of a diagonal Pauli channel, reflecting stochastic Pauli noise. Such a noise model can be obtained from a nondiagonal error model via a group twirl. Let $\rho$ denote the density matrix of a single-qubit state that undergoes the quantum noise. Then, the Pauli error-channel $\mathcal{E}$ has the Kraus representation

$$\mathcal{E}(\rho) := (1-p)\rho + p_X X\rho X + p_Y Y\rho Y + p_Z Z\rho Z. \quad (8)$$

The single-qubit physical error rate $p_{\text{err}} \in [0,1]$ is the total probability of $X$-, $Y$-, and $Z$-type errors. The individual Pauli error rates $p_\Omega, \Omega \in \{X,Y,Z\}$ are specified by the physical error rate $p_{\text{err}}$ and a bias vector $\vec{r} = (r_X, r_Y, r_Z)$. This bias $\eta_\Omega$ for a certain error species $\Omega \in \{X,Y,Z\}$ is given by

$$\eta_\Omega := \frac{r_\Omega}{\sum_{\Omega' \neq \Omega} r_{\Omega'}} p_{\text{err}}. \quad (9)$$

We call, for example, a noise channel *Z-biased*, if

$$\eta_Z = \frac{r_Z}{r_X + r_Y} > 0.5. \quad (10)$$

### 2. Syndrome error noise model

Inspired by the physical realization of bosonic cat qubits, we model the syndrome (or check) errors as Gaussian random noise that is added to the continuous syndrome $x \in \mathbb{R}$ obtained in the measurement. This affects the readout of the noiseless stabilizer values $s_i = \pm 1$. Therefore, the noisy stabilizer values $\tilde{s}_i$ have a continuous outcome given by $\tilde{s}_i = s_i + x_i$ where $x_i \sim \mathcal{N}(0, \sigma_i^2)$ is a Gaussian random variable with mean 0 and variance $\sigma^2$. By thresholding the noisy syndrome $\tilde{s}$, i.e., taking signs, $\text{sgn}(\tilde{s}_i)$, we obtain the *hard syndrome*. As detailed in Sec. C 2, the thresholding procedure allows us to relate syndrome error rates $p_{\text{err}}^{\text{synd}}$ and the variance $\sigma^2$ of the Gaussian noise process. i.e., given $p_{\text{err}}^{\text{synd}}$, the associated standard deviation $\sigma$ is given by

$$\sigma = \frac{1}{\sqrt{2}\text{Erfc}^{-1}(2p_{\text{err}}^{\text{synd}})}, \quad (11)$$

where $x \mapsto \text{Erfc}^{-1}(x)$ is the inverse of the complementary error function.

When considering biased-noise error models, we also bias the syndrome error channel in the same way as we bias the qubit error channel discussed in the previous section. From the individual syndrome error rates for $X$ and $Z$ checks, we obtain through Eq. (11) the corresponding variance of the Gaussian random noise model. For example, a

large $X$ bias means that there will be more bit-flip errors, as well as more $X$-syndrome errors compared to phase-flip errors and associated syndrome errors. In other words, we model the error affecting the bosonic modes of the inner code and in consequence the qubits of the outer code in a phenomenological fashion.

When incorporating the analog information into the decoding process, we replace the *log-likelihood ratios* (LLRs) of a discrete error model $\gamma_{\text{synd}} = \log[(1-q)/q]$, where $q$ is the measurement error probability, with

$$\gamma_i = \log\left[\frac{\Pr(\tilde{s}_i | s_i = +1)}{\Pr(\tilde{s}_i | s_i = -1)}\right] = \frac{2\tilde{s}_i}{\sigma^2}, \quad (12)$$

where $\Pr(\tilde{s}_i | s_i = \pm 1)$ is the probability of observing the noisy syndrome $\tilde{s}_i$ under the condition that the noiseless syndrome value is $s_i = \pm 1$.

## III. MIN-SUM BELIEF PROPAGATION ALGORITHMS

In this section, we discuss the decoding of quantum codes using iterative decoding procedures based on minimum-sum belief propagation (BP) decoding. First in Sec. III A, we review standard min-sum BP decoding for hard syndromes (i.e., for discrete-variable codes), which forms the basis for the discussed decoding procedures. In Sec. III B, we review recent work on the use of analog syndrome information to decode quantum codes [68], propose improvements to these techniques, and discuss caveats of the original method. The results of the numerical simulation are presented in the following section, Sec. IV, together with a comparison to the proposed decoding technique, analog Tanner graph decoding.

### A. Hard syndrome MSA decoding

*Belief propagation* (BP) is an iterative algorithm that is known to be efficient in decoding classical LDPC codes and has been adapted to quantum codes successfully in recent years. BP is a message-passing algorithm operating on the Tanner graph of the code (also called the factor graph in this context). The graph is considered a model that describes the factorization of the joint probability distribution of the error. Given the measured syndrome, the goal of BP is to (approximately) compute the marginal probabilities for each bit. For an error $e$ and syndrome $s = H \cdot e$, BP finds an estimate $\varepsilon$ of the error $e$ that yields the same syndrome $s = H \cdot \varepsilon$. The estimate vector $\varepsilon$ is formed as $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ where

$$\varepsilon_i := \text{argmax}_{e_i}[P(e_i|s)]. \quad (13)$$

There are several variations of the standard BP algorithm that differ in the way marginal probabilities are computed. In the following, we focus on min-sum BP, which has been

argued to be easier to implement on hardware-near devices than other variants [68]. For more details on min-sum BP, we refer the reader to Sec. C 3.

When applied to factor graphs that are trees (i.e., that do not contain loops), BP is known to be exact and will converge to a solution within a number of iterations less than the diameter of the graph. However, in general, LDPC codes contain loops. In this setting, BP computes approximate marginals and is not guaranteed to converge to a solution that satisfies the syndrome equation. In cases where BP does not terminate, the decoding can be deferred to a postprocessing routine. Such postprocessing routines are typically more computationally expensive but will ensure the decoder returns a solution satisfying the syndrome equation. A commonly used postprocessing method to improve the overall decoding performance of BP algorithms is *ordered statistics decoding* (OSD). This was first introduced in Ref. [69] and has recently been adapted to QLDPC codes [41,70].

Let us briefly summarize the main aspects of OSD. We consider a single check side in the following syndrome equation:

$$H \cdot e = s. \tag{14}$$

In general, $H$ may not be a square matrix and may not have full rank, and therefore we cannot directly invert $H$ to solve Eq. (14). The strategy applied is to choose a subset of columns that are linearly independent, and hence form a basis. Let $S := \{S_i\}$ be the set of column indices of linearly independent columns we chose and $H_S$ be the matrix of column vectors obtained. Clearly, $H_S$ has full rank and can thus be inverted to give a solution

$$H_S^{-1} \cdot s = e_S. \tag{15}$$

Different choices of $S$ correspond to (unique) solutions $e_S$. The main idea of BP+OSD is to use the marginal probabilities computed by the BP decoding algorithm to select a set $C$ that contains columns corresponding to bits with a high error probability.

## B. Soft-syndrome MSA decoder

Recently, Ravendraan *et al.* [68] introduced a variant of iterative min-sum decoding called *soft-syndrome min-sum algorithm* (SSMSA) to decode QLDPC codes using analog information. We briefly review the main aspects of the algorithm, for which we present the first open-source implementation. We show that the results of our implementation match the original results presented in Ref. [68]. Furthermore, we explore the combination of SSMSA with *ordered statistics decoding* (OSD), which improves the decoding performance compared to the original work.

SSMSA is an iterative message-passing decoding procedure, i.e., a variant of (min-sum) belief propagation
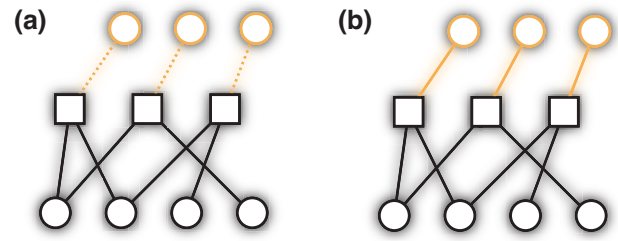


FIG. 4. (a) Illustration of the tanner graph for SSMSA decoding. The analog information taken into account by SSMSA can be illustrated similarly to ATD (although in SSMSA the analog information is *not* directly incorporated in the factor graph), however, because of the cutoff parameter, it may happen that SSMSA discards the analog information, which corresponds to ignoring the virtual nodes, indicated by dashed edges. (b) Sketch of the *analog Tanner graph*. The subgraph colored in black corresponds to the Tanner graph of the code. The subgraph highlighted in yellow corresponds to the virtual nodes that are used to incorporate the analog information. Their union is the ATG.

(BP) that uses soft information, i.e., real syndrome values instead of hard (binary) ones in its update rules. Initially, the corresponding binary syndrome is obtained by "thresholding" the analog syndrome, i.e., determined by the signs of the analog values. The update rules that are used to compute the messages are equivalent to those used in min-sum BP decoding with the addition of a "cutoff" parameter $\Gamma$, which is used to determine if the analog syndrome information should be considered in the update rules or not. If the absolute value of the analog syndrome is below the cutoff, it is taken into account when computing the min-sum updates (in addition to the standard messages). Conversely, if the absolute value of the analog syndrome is above the cutoff, the standard min-sum rules are applied. The SSMSA decoding process is visualized in Fig. 4(a) and the detailed pseudocode is presented in Algorithm 2 in Appendix C 4.

Analogously to BP, it is not guaranteed that SSMSA converges. The algorithm tries to infer a decoding estimate based on marginal probabilities, i.e., it tries to find the most probable error given a syndrome in a given number of maximum steps. Hence, in case the algorithm terminates due to reaching the maximum number of steps, one can in principle use the marginal probabilities the algorithm computed up to termination in a *postprocessing* step to infer a decoding estimate. However, postprocessing techniques were not considered in Ref. [68].

Note that in SSMSA, soft syndromes are not directly included in the parity-check matrix $H$ (i.e., the factor graph used for decoding is not altered), but are only used as an additional parameter to compute the marginal probabilities during the iterative decoding procedure. This means that measurement errors will not be considered for possible fault locations that satisfy the syndrome in OSD postprocessing, i.e., there are situations in which we are

trying to solve for a syndrome that is not in the image of $H$. In that case, it can happen that the "solution" with the highest error probability is not a solution of Eq. (15). This leads to cases where OSD postprocessing does not give a significant improvement over standard SSMSA decoding. In the following section, we propose techniques that amend this problem by ensuring that the factor graph is always invertible, i.e., has full row rank.

## IV. ANALOG TANNER GRAPH DECODING

In this section, we propose a novel decoding technique for quantum stabilizer codes using analog syndrome information that is based on min-sum BP+OSD decoding. We call this technique *analog Tanner graph decoding* as it is based on the construction of a variant of the standard Tanner graph that incorporates analog syndrome information.

In Sec. IV C, we briefly review single-shot decoding of quantum codes. Then, in Sec. IV C, we show how the analog Tanner graph decoding can be adapted to be used to decode single-shot quantum codes. The numerical results in this section are obtained by standard Monte Carlo decoding simulations, which are discussed in more detail in Sec. G. Additionally, in Appendix D we discuss how the proposed techniques can be adjusted to also include analog information of the data qubits and not only of the syndrome readout in similar architectures as the one we focus on in the rest of this paper.

Given the Tanner graph of a code $\mathcal{T} = (V_C \cup V_D, E)$, we can directly incorporate analog information by adding $|V_C|$ additional data nodes, called *virtual (data) nodes*, $V_V$. Each check node $V_C$, is connected to a single virtual data node through a single edge, as visualized in Fig. 4(b). The virtual data node stores the probability that the check is in error. In our protocol, this probability is derived from the magnitude of the analog syndrome readout. We refer to the modified Tanner graph as an *analog Tanner graph*.

In terms of parity-check matrices, building the ATG amounts to appending an $m \times m$ identity matrix $\mathbb{1}_m$ to the original parity-check matrix of the code:

*Definition 1 (Analog check matrix).* Given a parity-check matrix $H \in \mathbb{F}_2^{m \times n}$ corresponding to the incidence matrix of a Tanner graph $\mathcal{T}(H)$, the analog check matrix $H^{\mathcal{A}}$ has the following form:

$$H^{\mathcal{A}} := [H \mid \mathbb{1}_m]. \tag{16}$$

Clearly, $\mathcal{T}^{\mathcal{A}}$ constitutes a valid Tanner graph whose incidence matrix is $H^{\mathcal{A}}$. Moreover, $H^{\mathcal{A}}$ always has full rank. Consequently, there always exists a solution to the syndrome equation, cf. Eq. (14). For decoding, the virtual data nodes are initialized with the analog syndrome LLRs given in Eq. (12). Note that this construction can be seen

as a generalization of *data syndrome codes* for standard, hard (binary) syndromes [71–75], tailored to error models inspired by bosonic codes and their associated analog syndrome information.

To decode using the ATG, we can now apply standard decoding approaches such as BP+OSD, making the approach flexible to different decoding strategies and variations of the algorithm. If not stated otherwise, we refer to decoding using the ATG with BP+OSD as *analog Tanner graph decoding*. Note that the explicit inclusion of virtual nodes that correspond to fault locations for measurement errors eliminates the problems encountered with SSMSA and the OSD postprocessing techniques mentioned at the end of Sec. III B. Moreover, even though SSMSA with a cutoff of $\Gamma = \infty$ is conceptually similar to ATD, the syndrome update rules in SSMSA (as discussed in the previous section) and the fact that the information is not directly incorporated into the factor graph used for decoding, leads to significant performance discrepancies.

To investigate the decoding performance and compare ATD to SSMSA(+OSD), we perform standard Monte Carlo simulations to estimate the logical error rate for increasing syndrome noise $\sigma$ (and fixed data error rate $p = 0.05$) on a family of *lifted product* (LP) codes. The codes are taken from a code family defined in Ref. [24]. For more detail on the construction of codes via the lifted product, we refer the reader to Sec. C 1 in the Appendix. In Fig. 5(a), we compare the performance of SSMSA and ATD in the absence of OSD postprocessing, i.e., only BP is used for ATD (ATD+BP) and the original SSMSA algorithm is used. Following the methodology of Ref. [68], we fix the data error rate of the unbiased depolarizing noise model to be $p = 0.05$ and vary the syndrome noise. The strength of the syndrome noise is characterized by the standard deviation $\sigma$ of the Gaussian noise channel as defined in Sec. II C 2. From Fig. 5(a), it can be seen that ATD+BP always outperforms SSMSA if the logical error rate is limited by the data noise. We examined different cutoff values for the SSMSA implementation and found that a large value of $\Gamma$ yields the best performance for the code and noise parameter settings considered, but that there is no significant performance difference between $\Gamma = 100$ and values $\Gamma > 100$. Additional discussions on different decoder parameterizations are presented in Appendix F. In particular, setting the cutoff for SSMSA to $\Gamma = 0$ (meaning that the decoder does not take the analog syndrome information into account) clearly leads to higher logical subthreshold error rates.

In Fig. 5(b), the results of the same simulation setup as in Fig. 5(a) but using OSD postprocessing for SSMSA and ATD are shown. We observe that OSD always improves the performance of ATD. However, OSD leads to a reduced threshold when combined with SSMSA. Furthermore, we observe higher logical error rates for SSMSA-OSD when $\sigma > 0.35$, as well as the value of
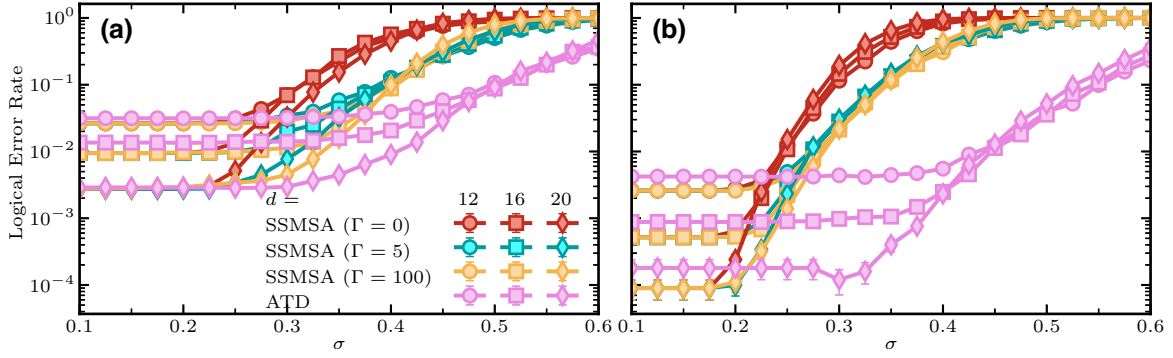
FIG. 5.   Comparison of the SSMSA decoder with ATD for increasing syndrome noise $\sigma$ on a family of lifted product codes with distances $d \in \{12, 16, 20\}$, cf. Sec. C 1. The data error rate of the unbiased depolarizing noise model is fixed at $p = 0.05$. (a) Comparison of the original SSMSA implementation and ATD using BP (without OSD). (b) Comparison of SSMSA+OSD and ATD using BP+OSD. The legend is shared between both panels and $\Gamma$ indicates the cutoff value of the soft information (SI) SSMSA decoder. Note that the case $\Gamma = 0$ corresponds to ignoring the analog information and, therefore, is equivalent to ordinary hard syndrome decoding.

the cutoff $\Gamma$ becoming less relevant to the decoder performance. This illustrates the possible issues that can occur when combining SSMSA and postprocessing without further modifications as discussed in the previous subsection.

### A. Comparison with SSMSA

The ATD method and SSMSA are conceptually similar, but differ in their implementations. First, we focus on the SSMSA cutoff parameter $\Gamma$, an input parameter that the decoding performance explicitly depends on. The two extreme values of $\Gamma$ correspond to $\Gamma = 0$ and $\Gamma = \infty$. For the former case, the syndrome is "trusted" and the analog information is never taken into account. In this scenario, the virtual updates are ignored. In the case where $\Gamma = \infty$, the analog information is always taken into account. The performance of the SSMSA algorithm is therefore related to the choice of $\Gamma$. In contrast, ATD does not have a cutoff parameter: the soft information is incorporated directly into the decoding graph and the amount of "trust" assigned to the syndrome value is determined through the standard MSA update rules.

Secondly, it is nontrivial to adapt SSMSA to more general Tanner graph constructions that go beyond the case of single-shot decoding. For example, SSMSA does not extend directly to the case of time-domain decoding or single-shot decoding with metachecks [44,45]. By comparison, the ATD decoder works out of the box in both of these settings as it considers a different factor graph for decoding.

Finally, when the magnitude of the analog information is below the cutoff, $\Gamma$, the SSMSA update rules can overwrite the original analog syndrome information, [cf. Line 22 in Algorithm 2]. The remaining SSMSA decoding iterations then no longer have access to the initial analog syndrome information. In contrast, this scenario does not occur in

ATD, because the analog syndrome information is stored explicitly in the nodes of the factor graph and the initial values are used throughout the algorithm by definition of the update rules. Hence, the initial analog syndrome information remains accessible in all ATD decoding iterations. As such, we argue that ATD is better positioned to make full use of the anolog syndrome information as the information propagates through the decoding graph explicitly.

### B. Single-shot decoding with metachecks

The three-dimensional surface code is defined by three parity-check matrices: $H_X$, $H_Z$, and $M_X$. The first two are the standard CSS code matrices that define the $X$ and $Z$ stabilizers. The matrix $M_X$ defines a so-called *metacode* $\mathcal{M}_X$, which is designed to provide protection against noise in the $Z$ syndromes (i.e., $X$ checks). More precisely, the metacode is defined so that all valid (nonerrored) $Z$ syndromes are in its code space. In other words, $M_X \cdot s_Z = 0$, when $s_Z = H_X \cdot e_Z$ for all $e_Z \in \mathbb{F}_2^n$.

In Ref. [44], Quintavalle *et al.* defined the *two-stage* single-shot decoder for the three-dimensional surface codes and related homological product codes. The steps of the two-stage single-shot decoding protocol are as follows:

(a) **Stage 1, syndrome repair:** measure the noisy syndrome $s = s_Z + s_e$, where $s_Z$ is the perfect syndrome and $s_e$ is the syndrome error. Solve the metacode decoding problem: given the metasyndrome $s_M = M_X \cdot s$, find an estimate $s'$ for the noiseless syndrome. If the metadecoding is successful, we obtain a corrected syndrome $s_C = s + s'$ satisfying $M_X \cdot s_C = 0$.

(b) **Stage 2, main code decoding:** use $s_C$ to obtain the decoding estimate $e'_Z$ by solving $s_C = H_X \cdot e_Z$.

Quintavalle *et al.* demonstrated that three-dimensional surface codes can be decoded using an implementation

of the two-stage protocol where the first stage uses a *minimum-weight perfect matching* (MWPM) decoder and the second stage a BP+OSD decoder.

A problem with the two-stage single-shot decoder is that the syndrome repair stage is independent of the main decoding stage. As a result, syndrome repair is always prioritized over applying corrections to the data qubits, even in situations where it would be more efficient to apply a combined correction. Furthermore, in certain circumstances, the two-stage decoder is subject to a failure mode whereby the syndrome repair step results in an invalid "corrected" syndrome $s_C$ that is not in the image of the parity-check matrix $H_X$, such that $s_C \notin \mathrm{Im}H_X$. Quintavalle *et al.* proposed a subroutine to handle such failure modes, but this adds to the computational runtime of the protocol [44].

Recently, Higgott and Breuckmann [45] proposed a *single-stage* decoding protocol for single-shot QLDPC codes that solves the problems mentioned earlier with the two-stage decoder. In the following we drop subscripts for simplicity and let $H$ be a check matrix, and $M$ be the corresponding metacheck matrix. The single-stage approach considers the decoding problem of a given noisy syndrome $s = He + s_e$ and metasyndrome $s_M = Ms$, to find a minimum-weight estimate $(e', s'_e)$, such that $H^M \cdot (e', s'_e)^T = (s, s_M)^T$, where

$$H^M := \begin{pmatrix} H & \mathbb{1}_m \\ 0 & M \end{pmatrix}, \tag{17}$$

is called the *single-stage parity-check matrix*. The single-stage parity matrix combines both the decoding of the data errors $e$ and the syndrome errors $s_e$. This ensures that the decoder uses the full information available to it, in contrast to the two-stage decoder that processes the metasyndrome and syndrome separately. Furthermore, note that the first block row of Eq. (17) is full rank. This property guarantees that all solutions are valid, meaning that the single-stage decoder does not suffer from the failure mode that arises in the two-stage decoding approach.

### C. Analog single-shot decoding

We now discuss how ATD can be applied to improve single-stage single-shot decoding and explore connections between single-stage parity-check matrices and the ATG construction. To begin, we first note that the single-stage parity-check matrix of Eq. (17) is conceptually similar to the analog parity-check matrix of Eq. (16): the single-stage parity-check matrix simply introduces a particular set of constraints described by the metacode $M_X$. Note that, however, as these constraints are linear combinations of the top full-rank block $(H \mid \mathbb{1}_m)$ of $H^M$. Hence $H^M$ can be understood as making the implicit meta constraints explicit in the factor graph [76]. An example of a single-stage Tanner graph is depicted in Fig. 6. From this sketch, it is evident
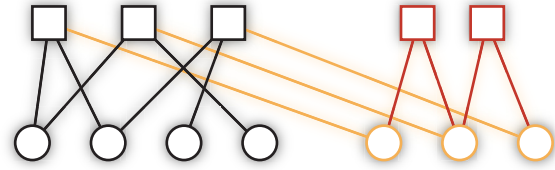


FIG. 6. Sketch of the analog single-stage Tanner graph based on $H^M$. The black checks, bits, and edges correspond to the Tanner graph of the linear code $H$. The orange nodes are due to the identity $\mathbb{1}_m$ in Eq. (17) and are initialized with the analog information of the syndrome readout. The red nodes are due to the metacode, i.e., the metacheck matrix $M$.

that this construction corresponds exactly to the structure of the ATG together with additional (meta)check nodes as defined by the metacode $\mathcal{M}_X$. If we incorporate the metacode directly in the construction of the ATG, (similar to Ref. [45]), we construct the *single-stage ATG*, which can be used to decode single-shot codes using analog information.

Let us briefly review some details of the overall decoding procedure. Consider an analog single-stage data syndrome $s \in \mathbb{R}^{r_z}$. To initialize the ATD, we threshold $s$ to a binary syndrome $s_b \in \{0, 1\}^{r_z}$, which we use to decode. Additionally, we use the analog syndrome values of $s$ to initialize the virtual nodes of the ATG, as in standard ATD. Standard decoding algorithms such as BP+OSD can then be applied to the resulting factor graph to obtain a decoding estimate.

In order to investigate the decoding performance of single-stage single-shot ATD decoding, we conduct sustainable threshold simulations for a concatenated bosonic 3D toric code using a phenomenological noise model. Single-shot error correction will generally leave some residual error after the correction. The goal is to suppress the accumulation of this residual error to the point at which it can be corrected in subsequent rounds. To this end, we define the *sustainable threshold* for a single-shot code as the physical error rate, $p_{\text{sus-th}}$, below which the residual error remains constant and the quantum information can be stored indefinitely by increasing the distance of the code. More precisely, the sustainable threshold is defined as

$$p_{\text{sus-th}} := \lim_{R \to \infty} p_{\text{th}}(R), \tag{18}$$

where $p_{\text{th}}(R)$ is the threshold for $R$ rounds of noisy stabilizer measurements.

To estimate $p_{\text{sus-th}}$ numerically for the 3D toric code, we estimate $p_{\text{th}}(R)$ for increasing values of $R \in \{0, 1, 2, 4, 8, \dots, 128\}$ until the value $p_{\text{th}}(R)$ is constant, i.e., until the threshold does not decrease when increasing the number of rounds $R$. Our simulation results

are shown in Fig. 7(a). We see that under single-stage ATD decoding, the 3D toric code has a sustained threshold of 9.9%. This improves by 2.8% on the sustained threshold of 7.1% obtained by Higgott and Breuckmann for the same family of 3D toric codes, but using DV qubits and hard syndromes [45]. The improved sustainable threshold we observe highlights the benefits of considering analog information in decoding. Figure 7(b) shows the threshold of the 9.9% for concatenated bosonic 3D toric codes of size $L = \{5, 7, 9, 11\}$ after 128 noisy rounds of decoding.

We note that the sustainable thresholds we have found could be further optimized by fine tuning the parameters of the BP+OSD decoder. However, rather than showing optimal improvements, the goal of this experiment is to indicate the improvements that become possible by considering analog readout information. A more complete study would additionally compare below-threshold error rates, which are on this scale more relevant than the improvement of the threshold. We will leave this question open for future work, including a more detailed simulation including circuit-level noise models.

Finally, we note that the check matrix defined in Eq. (17) cannot be used in the SSMSA algorithm without significant modifications. The reason for this is that syndrome errors are already included explicitly in Eq. (17), which can *a priori* not be handled by SSMSA, since for SSMSA the analog information is an input parameter and the algorithm operates on the standard parity-check matrix of the code. Moreover, the metachecks do not correspond to physical measurements, i.e., they do not have analog syndrome information.

## V. QUASI-SINGLE SHOT CODES

Even in the presence of strong noise bias, as in the case of cat-LDPC architectures, both error species, $X$ and $Z$ noise, must be corrected to achieve good overall logical fidelity [19]. When implementing one-sided single-shot codes, such as the three-dimensional surface code, we cannot rely solely on the single-shot side of the code to correct errors.

To decode a quantum code that does not have the single-shot property, multiple rounds of (noisy) syndrome measurements must be performed so that the decoder infers the presence of measurement noise [46–48,77]. Usually, this process is referred to as *repeated measurements* or *time-domain decoding*, since repeating stabilizer measurements adds an additional time dimension when considering the decoding instance. In this section, we investigate the decoding of quantum codes under phenomenological noise with repeated measurements in the presence of analog information. To this end, in Sec. V A, we discuss *overlapping window* decoding [48] that we generalize to decode QLDPC codes over time in the presence of analog information. Moreover, we elaborate on the relation between the overlapping window method and the ATG construction proposed in Sec. IV.

Motivated by the structure of 3D bosonic-LDPC code architectures, we further propose a novel decoding protocol that we call *w-quasi-single shot codes* (*w*-QSS codes). The main idea is to leverage noise bias and analog syndrome information (provided by bosonic-LDPC codes) to demonstrate that only a small number $w \ll d$ of repeated syndrome measurement cycles suffices for the non-single-shot check side to give an overall decoding protocol with high logical fidelity for error rates sufficiently below threshold.

The central result of this section is that we demonstrate numerically that for the cat-3DSC with reasonable code sizes ($L = 11$), the *w*-QSS protocol achieves a threshold of approximately equal to 1.5% for the non-single-shot side under phenomenological noise and that in the sub-threshold regime, $w = 3$ suffices to match the decoding
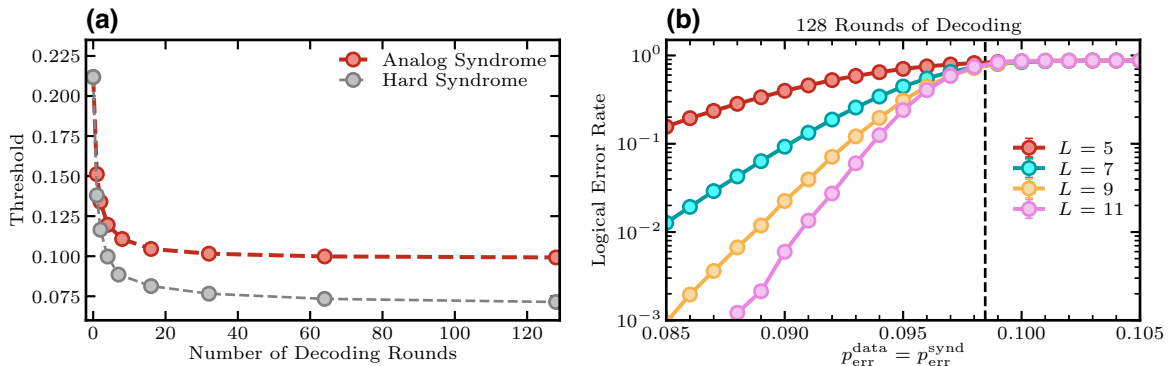


FIG. 7. Performance of single-shot analog Tanner graph decoding for 3D toric codes. (a) Sustainable threshold of the single-shot side of 3D toric codes using ATD. The analog information increases the sustainable threshold of current state-of-the-art methods [45] by almost 3%. The results labeled "hard syndrome" are due to Higgott and Breuckmann [45], while the "analog syndrome" results are obtained with our ATD method. (b) Example of threshold determination after 128 rounds of decoding using toric codes with lattice size $L \in \{5, 7, 9, 11\}$.

performance of time-domain decoding with $\propto d$ repeated measurements.

## A. Analog overlapping window decoding for QLDPC codes

To decode an $[[n, k, d]]$-quantum code under phenomenological noise, i.e., when the syndrome measurements are noisy, we need to repeat the measurements several times (usually at least $d$ times) in order to handle the noisy syndromes [47,48]. Generally, this leads to the dimension of the decoding instance being increased by one, as the time dimension is now also being considered. Hence, we refer to this decoding problem as *time-domain decoding*. For example, the decoding problem of a repetition code under phenomenological noise leads to a two-dimensional decoding problem, analogous to the two-dimensional surface code under bit-flip noise. Similarly, the decoding of the two-dimensional surface code over time leads to a three-dimensional problem, analogous to decoding the three-dimensional surface codes under bit-flip noise. We make the connection between the proposed construction of ATG, the decoding of quantum codes over time, and the tensor product chain complexes explicitly and formally argue that they are equivalent (cf. proposition 1).

*Overlapping window decoding* (OWD) was originally introduced by Dennis *et al.* in Ref. [48] to decode quantum surface codes over (finite) time. In OWD we divide the collected syndrome history into $w$-sized regions, the first two of which are sketched in Fig. 8. At any instance, the decoder computes the correction for two regions, whereas the older one is called "commit" and the newer one "tentative," $R_c, R_\tau$, respectively. A window encompasses a total of $w$ noisy syndrome measurements. For each decoding round, the syndrome data of $2w$ rounds, i.e., two regions, is used to find a recovery operation by applying a decoder. However, only the correction for the first $w$ rounds, i.e., for region $R_c$ is applied (by projecting onto the final time step of $R_c$ and applying the recovery accordingly). After applying the recovery, the region $R_c$ can be discarded, and only $R_\tau$ is kept. Then, in the next decoding round, the same procedure is repeated using the previous $R_\tau$ as the new commit region and the next $w$ rounds as the new temporary region, which is conceptually equivalent to "sliding up" the $2w$-sized decoding window one step.

Dennis *et al.* argue for the two-dimensional surface code that the number of time steps $w$ should be chosen proportionally to the distance of the code, $w \gg d$, to ensure that the probability of introducing a logical operator is kept small. This method is needed to simulate memory experiments over (a finite amount of) time, since the last, perfect round of measurements (corresponding to data qubit readout) may artificially increase the observed threshold leading to the fact that doing fewer repetitions
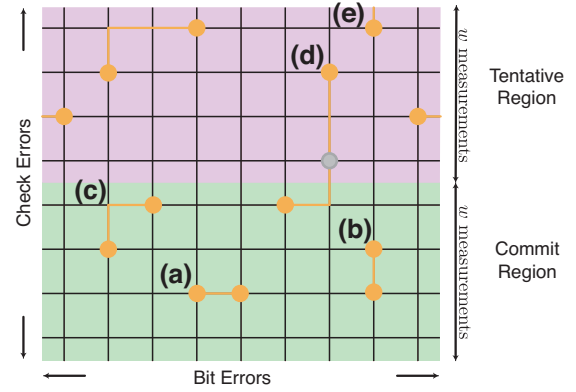


FIG. 8. Sketch of the overlapping window decoding method of a repetition code as proposed in Ref. [48]. (a) A spacelike single data qubit error, (b) a timelike error, (c) a space-time error, (d) a space-time error that extends across the region boundary, (e) a timelike error that extends into the next decoding round. Note that the error (d) will only be partially corrected as it extends over the boundary of the commit region. The inferred correction will imply that all defects in the commit region are removed, but will introduce a new defect in the tentative region shown as a gray dot. Defects created in this way are referred to as *virtual defects* [49] in the literature. An error purely residing in the tentative region will be considered during decoding to infer a decoding estimate matching the syndrome, but will not be corrected in the same round.

always performs better. This aspect has also been observed in Ref. [78]. Additionally, overlapping window decoding also corresponds to how a fault-tolerant quantum computer will likely be decoded in practice [49].

Let us at this point fix some terminology: the window size $w$ is the number of syndrome measurement records in a single window, i.e., the total size of the two regions is given by $|R_c| + |R_\tau| = 2w$. A single round of decoding takes $2w$ noisy syndrome measurements into account, however, due to the sliding nature of the decoding window, $n$ rounds of decoding correspond to $(n + 1)w$ syndrome measurements. We refer to time-domain decoding, where the number of repetitions is proportional to the distance of the code as *standard decoding*. To apply this method to arbitrary QLDPC codes, we first propose the construction of the *3D analog Tanner graph*, i.e., the decoding graph over time for time-domain decoding.

Given the Tanner graph of a QLDPC code $\mathcal{T}(\mathcal{C})$, we first create copies of $\mathcal{T}$ and introduce an additional set of bit nodes between pairs of checks between consecutive copies of $\mathcal{T}$, and an additional set of bit nodes for the last copy of $\mathcal{T}$. These correspond exactly to timelike errors on syndrome nodes. The construction is sketched in Fig. 9. Algebraically, the multiround parity-check matrix $\tilde{H}$, i.e., the incidence matrix of the multiround Tanner graph can be defined as follows.
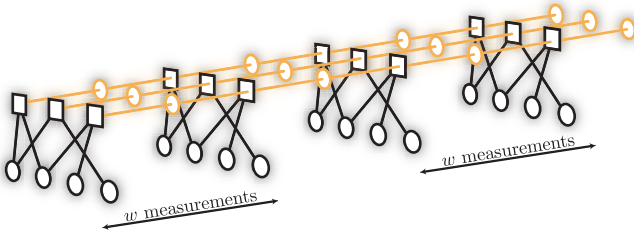
FIG. 9. Overlapping window decoding for QLDPC codes (arbitrary Tanner graphs). The individual Tanner graphs of a code described by the parity-check matrix $H$ are "glued" together by the orange syndrome nodes that correspond to timelike (measurement) errors. A single decoding round with window size $w$ involves $2w$ measurements, but only corrections in the first $w$ are committed, see also Fig. 8.

*Definition 2 (Multiround parity-check matrix and Tanner graph).* Given an $m \times n$ parity-check matrix $H$, the $r$-multi-round parity-check matrix is defined as

$$\tilde{H} := \left( H_{\text{diag}} \mid \mathbb{1}_{\text{sdiag}} \right) \tag{19}$$

$$= \begin{pmatrix} H & 0 & 0 & \ldots & \mathbb{1}_m & & \\ & H & 0 & \ldots & \mathbb{1}_m & \mathbb{1}_m & \\ & & H & 0 & \ldots & \mathbb{1}_m & \mathbb{1}_m \\ & & & \ddots & & \ddots & \\ & & & H & 0 & 0 & \mathbb{1}_m & \mathbb{1}_m \end{pmatrix}, \tag{20}$$

where $H_{\text{diag}}$ is a block-diagonal matrix with $r$ diagonal block entries, and $\mathbb{1}_{\text{sdiag}}$ has a step-diagonal form, consisting of $m \times m$ identity matrices.

Hence, $\tilde{H}$ is an LDPC code whose parity-check matrix consists of copies of $H$ with additional bit nodes between pairs of checks. It is easy to see that this code is LDPC (with respect to the number of repetitions) since the vertex degree for each check is increased by (at most) 2, independent of the number of repetitions. As an example, consider the multiround (analog) Tanner graph and the corresponding multiround check matrix depicted in Fig. 1(f), which corresponds to an instance of a multiround parity-check matrix (and Tanner graph) for a repetition code over two rounds.

We show that the construction of the multiround Tanner graph for $r$ rounds of syndrome measurement can be described as the tensor product chain complex of the chain complex corresponding to the QLDPC code $\mathcal{C}$ and the chain complex of (a slight variant of) the $r$-repetition code $\mathcal{R}$:

*Proposition 1 (Informal statement).* The $r$-multi-round parity-check matrix $\tilde{H}$ of $\mathcal{C}$ is equivalent to the check matrix of the tensor product code $\mathcal{R} \otimes \mathcal{C}$.

The proof is elementary and follows from the chain complex tensor product; we refer the reader to Appendix B for more details. Note that this highlights that the construction is equivalent to "stacking" the ATG constructed from $H$ and connecting the virtual nodes between pairs of checks. This gives a straightforward correspondence between ATGs and phenomenological check matrices and allows us to directly apply ATD to $\tilde{H}$.

The overall procedure for multiround decoding with analog information can be summarized as follows. First, from the parity-check matrix $H$ build the $r$-multi-round check matrix $\tilde{H}$ (corresponding to the multiround analog Tanner graph). Second, use the virtual nodes (corresponding to timelike data nodes in standard models) to incorporate analog syndrome information. And finally, apply the ATD decoder on $\tilde{H}$.

Note that recently and independently, BP+OSD has been used to decode (single-shot) LDPC codes under a circuit-level noise model [35,40]. The proposed overlapping window approach we outline in this work differs in that it also considers analog information in the decoding. Additionally, our methods apply to QLDPC codes in general and do not rely upon codes with a special code structure. Furthermore, recently and independently in Ref. [79], the authors considered decoding of LDPC codes under discrete phenomenological noise using a check-matrix construction equivalent to our definition of the multiround parity-check matrix. However, they do not employ overlapping window decoding.

### B. Quasi-single shot decoding

In the previous section, we discussed how ATD can be used together with multiround (analog) parity-check matrices to decode under phenomenological noise with analog information. In this section, we propose a novel protocol based on these techniques that lowers the overhead induced by repeated measurements.

In the $w$-QSS protocol, we assume a QLDPC code where one side is single-shot and the other side is not—inducing the need for at least $d$ repeated measurements (or a number of repeated measurements proportional to the distance of the code) in the presence of noisy syndromes, where $d$ is the distance of the non-single-shot side. This applies, for instance, to three-dimensional surface codes, which we use as representatives in the following. The single-shot side of the code can be decoded using analog single-stage decoding discussed in Sec. IV C. Complementarily, a slight generalization of the analog *overlapping window decoding* (OWD) as described in Sec. V A is used to decode the non-single-shot side of the code. The overall protocol is straightforward:

(1) Choose a $w \ll d$. Intuitively, $w$ controls the number of noisy syndrome measurements to conduct for the
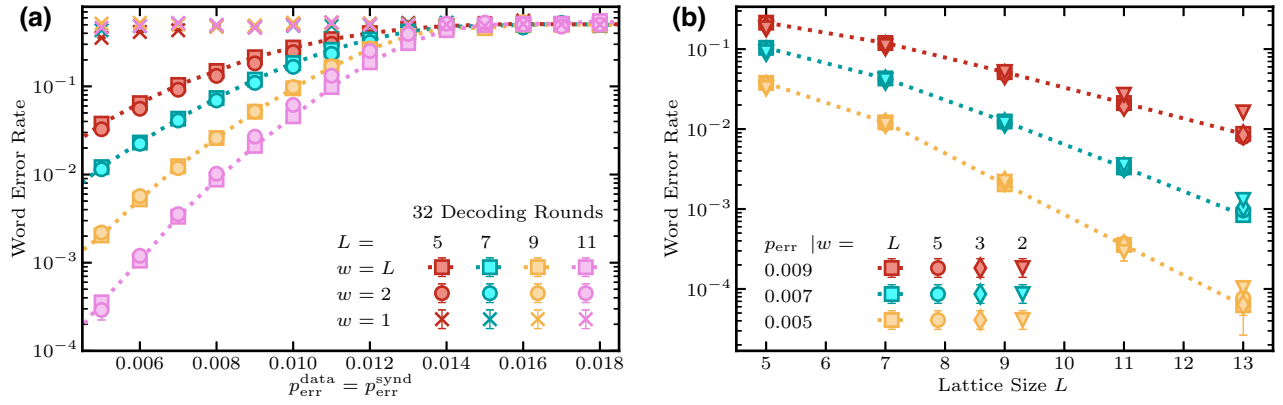
FIG. 10.  Performance of the *w*-quasi-single shot protocol using the three-dimensional surface codes under phenomenological bit-flip noise with analog syndrome readout. The non-single-shot side is decoded using minimum-weight perfect matching. (a) Word error rate after 32 decoding rounds as a function of the phenomenological error rate for various window size values *w*. (b) Below threshold scaling of the word error rate after 32 decoding rounds for various window size values *w*. The results suggest that it is sufficient to repeat the stabilizer measurement a finite number of times *independent* of the code distance *L* when incorporating analog information into the decoder.

non-single-shot side. In other words, *w* is the window size for overlapping window decoding.

(2) On the single-shot side of the code, we apply the usual analog single-shot decoding procedure, where in each time step we do a syndrome measurement and infer a recovery operation.

(3) For the non-single-shot side, we do *w* time steps of repeated measurements and then decode (using analog OWD).

Without analog information, by restricting the number of syndrome measurements to *w* the *effective distance* of the code is lowered to *w*. For instance, consider a code that has $d_X = 4, d_Z = 16$ where the *X* side is single-shot. Then, setting $w = 4$ effectively reduces $d_Z$ to 4 because, in the time dimension, logical errors can be of weight 4 only. This is apparent when considering the multiround Tanner graph used for decoding over time as a tensor product with a repetition code. If we have $d_Z = 16$ and do 16 rounds of noisy syndrome measurements, the repetition code protecting the system from "timelike" errors has distance 16, but when restricting to *w* repetitions, we essentially cut the repetition code in time to a shorter version, thereby lowering the distance along the time dimension.

To investigate the performance of the proposed protocol numerically, we simulate three-dimensional surface codes under phenomenological (cat) noise for lattice sizes $L = 5$ to $L = 11$ and compare different choices of *w* versus the standard approach of taking a number of repeated measurements proportional to the distance. Note that, to conduct numerical simulations for repeated measurements we need to conduct multiple rounds of decoding to avoid overestimation of the threshold [78] (cf. Sec. V A). Since the non-single-shot side of this code can be decoded with matching-based algorithms, we use PyMatching [80,81]

for decoding. The threshold behavior and the subthreshold scaling for 32 decoding rounds are shown in Fig. 10. We discuss the decay of logical fidelity later on in Fig. 11. The main findings are as follows.

(a) For $w = 2$ we observe an increase in logical error rate for $L = 11$ for error rates around the threshold, however, the subthreshold suppression for lower error rates performs as for the standard time-domain decoding as shown in Fig. 10(b).

(b) $w = 3$ is enough to match the results of standard time-domain decoding (for the investigated code sizes and error rates).

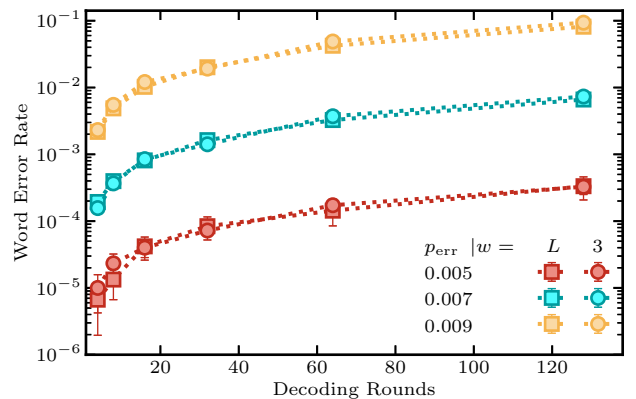(c) For $w = 1$ the protocol does not work.



FIG. 11.  Comparison of the decrease of decoding success of the 3-QSS decoding ($w = 3$) versus the standard time-domain decoding ($w = L$) for an increasing number of decoding rounds and subthreshold physical error rates using the $L = 13$ three-dimensional surface codes under phenomenological noise.

As a byproduct, we obtain a threshold of the non-single-shot side of the 3DSC under phenomenological noise of approximately equal to 1.66%. In Appendix F we also present 3DSC threshold estimates for the case where only the hard syndrome information is available. We find that the threshold is approximately equal to 1.26%. To the best of our knowledge, these are the first numeric threshold estimates for the non-single-shot side of the 3DSC.

Let us make some more detailed remarks on the results. First, the results indicate that $w = 2$ for the $L = 11$ code leads to a worse threshold, indicating a limitation of the protocol in this aspect. However, the subthreshold scaling (which is actually relevant in practice) still shows that the 2-QSS protocol provides sufficient error suppression while lowering the number of syndrome measurement rounds from 11 to 2, inducing a fraction of the time overhead to implement the overall QEC protocol. Secondly, increasing the QSS window size $w$ by 1 already significantly improves the achieved logical error rate. With $w = 5$ we do not find statistically significant differences from $w = L$ for the code sizes considered. Lastly, we note that even for much smaller error rates, we did not find a threshold for $w = 1$.

## C. Discussion

For a QLDPC code that requires time-domain decoding, the effective distance is proportional to the number of repeated syndrome measurements. Thus, taking only a small number $w \ll d$ of syndrome measurements is equivalent to lowering the effective code distance (along the time dimension) to $w$. However, we show numerically that for reasonable code sizes and choices of the QSS window size $w$, the logical error rate of the overall protocol is equivalent to the standard approach of performing (at least) $d$ repeated measurements due to the additional information acquired from the analog syndrome.

Although our numerical results suggest that $w$ can be chosen as a constant independent of the code size for sufficiently small physical error rates, it is reasonable to assume that for larger code sizes (i.e., in the limit $n \to \infty$), the logical error rate for the QSS protocol diverges from the logical error rate obtainable from standard time-domain decoding (i.e., with a number of rounds proportionally to the code size) and will possibly result in an error floor set by timelike errors. However, this gap in error suppression between the QSS protocol and the standard protocol quickly diminishes if the physical error rate is sufficiently below the threshold, as can be seen by inspecting Fig. 10(b). For example, for $p_{err} = 0.009$, $w = 2$ gives a significantly higher word error rate than the larger choices of $w$, but for smaller error rates, e.g., $p_{err} = 0.005$, the discrepancy with standard time-domain decoding is reduced. Thus, overall the main learning is that we observe that for sufficiently small physical error rates, i.e., error rates

sufficiently below threshold, the $w$-QSS protocol can give logical error rates that are equivalent to standard time-domain decoding.

It would be interesting to investigate this aspect analytically for an LDPC code family. For instance, it is reasonable to assume that depending on the LLRs (weights) used for decoding one can argue that if the weights are $w = O(1)$ using hard information decoding and are increased to $\ell w$ by using analog information, one can reduce the number of repetitions by a factor of $\ell$ without affecting the logical error rate significantly, i.e., obtain an $L/\ell$-QSS protocol. We leave further analytic investigation of this manner open for future work. Note that the investigated codes are already well beyond the capabilities of near- to midterm hardware, and thus we argue that our numerical results are valid for "practical sizes."

It is crucial to note that for the QSS protocol, the noise-biased error model is vital. The main reason is that in this setting, the bulk of the decoding is offloaded onto the single-shot component of the 3DSC, while the QSS protocol carries a much lighter load. This is also important due to the asymmetric thresholds of the 3STC for pure bit- and phase-flip noise, of approximately 1.5% and 10%, respectively, which leads to the fact that the overall threshold of the code $p_{err}^{th}$ is limited by bit-flip errors. However, according to Eq. (9), already a small bias of $\eta_Z \approx 10$ will distribute the error-correction load equally, and any bias $\eta_Z \gg 10$ will result in an effective bit-flip error rate $p_X$ that is well below the threshold of the QSS protocol if the overall error rate is $p_{err} < 10\%$.

Although we are currently limited to simulations with codes of size $L \le 11$ and physical error rates $p_{err}$ around the threshold (due to the impracticality of conducting numerical experiments with larger codes), we argue that codes of such size are already reasonable for practical relevance due to good error suppression on the single-shot side of the code [44]. However, for a more quantitative analysis, circuit-level noise model simulations are required. We expect circuit-level simulations will decrease observed thresholds by a factor of 4–8×. This estimate is consistent with the numerical results of Pattison *et al.*, where two-dimensional surface codes were decoded under analog circuit-level noise [78]. Pattison *et al.* also proposed a generalization of the standard circuit-level noise model to include analog measurements to simulate surface code decoding under realistic noise assumptions. Since the generalization to circuit-level noise encompasses several nontrivial questions such as the design of syndrome extraction circuits and the derivation of exact cat qubit noise models, we leave this task open for future work. Nonetheless, we discuss several challenges in more detail in Sec. VI.

To verify that the QSS protocol does not lead to a decrease of the logical error rate (and the threshold) for an increasing number of decoding rounds, we conducted sustained threshold simulations, i.e., threshold simulations

for an increasing number of decoding rounds. However, due to the saturation of the logical error rates in the numerical results, we are only able to obtain a lower bound on the threshold, which decreases with the number of decoding rounds. Therefore, we provide additional results, shown in Fig. 11, which demonstrate that the decay of the decoding success (logical success rate) for a number of decoding rounds scales equivalently for the QSS protocol compared to the standard time-domain decoding, where the number of syndrome extraction rounds is $w = L$ (i.e., proportional to the distance) for physical error rates below the obtained threshold lower bounds. Despite the fact that this result constitutes a weaker statement than a sustained threshold estimate, it indicates that the performance of the QSS protocol is equivalent to standard time-domain decoding even for an increasing number of decoding rounds.

Moreover, we emphasize that while our simulations indicate that the QSS protocol works in a memory experiment, it is an open question whether this is also the case for a setting in which fault-tolerant logical gates are implemented. Another feature that could potentially affect decoding performance is the presence of so-called fragile boundaries, as discussed in Ref. [76].

# VI. TOWARDS THREE-DIMENSIONAL CONCATENATED CAT CODES

In this section, we discuss roads towards building a fault-tolerant quantum computer based upon stabilized cat qubits concatenated with the three-dimensional surface code. To this end, we will elaborate on several open questions and challenges along that road. We begin by recalling some properties of stabilized cat qubits.

## A. Stabilized cat codes

The cat code encodes logical (qubit) information within a two-dimensional subspace of the infinite-dimensional Hilbert space of a harmonic oscillator with Hilbert space $\mathcal{L}(\mathbb{R}^2)$. This qubit subspace is represented by the span $\{|\alpha\rangle, |-\alpha\rangle\}$ of two quasiorthogonal coherent state vectors $|\pm\alpha\rangle$, $\alpha \in \mathbb{C}$ [82] (in the sense that for large values of $|\alpha|$, they approximate an orthogonal pair of state vectors arbitrarily well). The nonorthogonality of this basis does not pose a problem for the definition of the orthogonal qubit space, and one defines the Hadamard-dual basis codewords $|\pm\rangle_{\text{cat}}$ as two-component Schrödinger cat state vectors, i.e.,

$$|\pm\rangle_{\text{cat}} = \mathcal{N}_\pm(|\alpha\rangle \pm |-\alpha\rangle), \qquad (21)$$

which are orthogonal state vectors, and

$$\mathcal{N}_\pm^2 := 1/(2(1 \pm e^{-2|\alpha|^2})) \qquad (22)$$

is the normalization factor [83]. Then, the logical, computational state vectors are obtained as

$$|0\rangle_{\text{cat}} = \frac{1}{\sqrt{2}}(|+\rangle_{\text{cat}} + |-\rangle_{\text{cat}}) = |+\alpha\rangle + O(e^{-2|\alpha|^2}), \tag{23}$$

$$|1\rangle_{\text{cat}} = \frac{1}{\sqrt{2}}(|+\rangle_{\text{cat}} + |-\rangle_{\text{cat}}) = |-\alpha\rangle + O(e^{-2|\alpha|^2}), \tag{24}$$

and the approximations $|0\rangle_{\text{cat}} \approx |\alpha\rangle$ and $|1\rangle_{\text{cat}} \approx |-\alpha\rangle$ become arbitrarily accurate for $|\alpha|^2 \to \infty$.

The cat code space is not stable under noise channels that typically affect the physical realizations of harmonic oscillators—dominated by energy relaxation reflected by losses and dephasing. Thus, any logical information will eventually leak outside of the code space and will be unrecoverable. However, through engineered interactions, it is possible to stabilize the code space through appropriate confinement schemes. While various different confinement schemes, such as Kerr stabilization [84], dissipative stabilization [85], and combined methods [86], exist, they share similar principles. First, to overcome energy relaxation, one actively pumps energy into the system through engineered (two-photon) drives. Then, an actual "confinement" term is added that separates the cat qubit manifold from the rest of the energy spectrum. To ensure a two-fold degenerate ground state of the system, these engineered interactions must be symmetric with respect to the substitution $\hat{a} \mapsto -\hat{a}$, where $\hat{a}$ is the bosonic annihilation operator satisfying the canonical commutation relation $[\hat{a}, \hat{a}^\dagger] = \mathbb{1}$ [87].

Stabilization through some confinement interaction ensures that if leakage occurs, the state will relax back to the code space. As a result, stabilized cat codes allow for arbitrary suppression of bit-flip noise under realistic oscillator noise models [88] [17,58], as we will illustrate in the case of single-photon losses below. However, the confinement does not protect against logical cat qubit $Z$ errors on the code space, which may occur directly through oscillator decoherence, such as phase flips caused by energy relaxation, or indirectly if a noise channel leads to temporary leakage out of the code space, e.g., caused by thermal noise.

The time evolution of a single-mode quantum system undergoing single-photon loss is well described by the Lindblad master equation [89,90],

$$\frac{\partial}{\partial t}\hat{\rho} = \kappa \mathcal{D}[\hat{a}]\hat{\rho} = \frac{\kappa}{2}\left(2\hat{a}\hat{\rho}\hat{a}^\dagger - \hat{a}^\dagger\hat{a}\hat{\rho} - \hat{\rho}\hat{a}^\dagger\hat{a}\right), \tag{25}$$

where $\kappa > 0$ is the single-photon loss rate and $\hat{\rho}$ is the density operator describing the state of the system. Here, the first term leads to quantum jumps, whereas the latter two terms generate a non-Hermitian evolution that

leads to energy relaxation. One can calculate the leading-order estimates for the cat qubit phase- and bit-flip error rates through the Knill-Laflamme conditions [91,92] for the oscillator error $\hat{E}_1 \propto \sqrt{\kappa}\hat{a}$. The task reduces to the following transition matrix elements:

$$\kappa|\langle+\alpha|\hat{a}|-\alpha\rangle|^2 = \kappa|\alpha|^2 e^{-4|\alpha|^2}, \tag{26}$$

$$\kappa|\langle+|\hat{a}|-\rangle_{\text{cat}}|^2 = \kappa|\alpha|^2 \tanh(|\alpha|^2) \approx \kappa|\alpha|^2, \tag{27}$$

which shows that the ratio of bit- to phase-flip errors is exponentially suppressed, i.e., $p_x/p_z \sim e^{-4|\alpha|^2}$, yielding an effective biased-noise error channel for the outer code. We note that exponential suppression (in $|\alpha|^2$) of bit-flip errors comes at the cost of linearly increasing the phase-flip rate. Although this will limit the extent to which one can increase $\alpha$ before $p_z$ exceeds the threshold of the concatenated code, we emphasize that the recently introduced *stabilized squeezed-cat qubit* allows one to suppress bit-flip errors without increasing the phase-flip errors [93].

To benefit from a biased-noise error channel, it is important that the noise bias can be sustained even during gate operations. It has been shown that this is possible for stabilized cat qubits due to the complex-valued displacement amplitude $\alpha$, which contributes additional degrees of freedom and in this way allows the realization of the two-qubit CNOT gate in a bias-preserving way by performing (conditioned) rotations that exchange $|\alpha\rangle$ and $|-\alpha\rangle$. During this rotation, the bias is preserved, and a topological phase is added to the dual-basis codewords, i.e., $|+\rangle_{\text{cat}} \mapsto |+\rangle_{\text{cat}}$ and $|-\rangle_{\text{cat}} \mapsto |-\rangle_{\text{cat}}$ [17,58]. In a circuit-level noise model the ratio of single-photon losses and confinement rate is an important parameter that relates effective encoded cat qubit Pauli error rates to physical noise parameters, see, e.g., Refs. [14,17,19] for more details.

Finally, performing a logical $Z$ measurement can be done, for example, by performing a nondemolition cat quadrature readout [12], which distinguishes the two coherent state vectors $|+\alpha\rangle$ and $|-\alpha\rangle$, see also Fig. 1(b). Due to the finite variance of coherent states, such a measurement will be inherently imprecise because of their continuous distribution in quantum phase space. However, one can incorporate this analog information into the decoding stages of the outer code, for example, by assigning higher error likelihoods to states that have measurement outcome $x_m \approx 0$. Importantly, the *resolvability* of the cat qubit computational state vectors $|0/1\rangle_{\text{cat}}$ is given by the overlap of the two states that scales as approximately $e^{-2|\alpha|^2}$ and thus assignment errors become exponentially suppressed with the size of the stabilized cat qubit.

## B. Open questions

We highlight that an immediate open question, independent from any experimental realization, is the verification

of our decoding protocols in a more realistic noise model, i.e., in the presence of circuit-level noise and the determination of thresholds in these cases. One might expect a reduction in threshold (roughly) proportional to the stabilizer weight, due to additional fault locations that occur in the syndrome extraction circuit, impacting the non-single-shot side of the code more strongly than the single-shot side, which have stabilizers of weights 6 and 4, respectively. However, this will not cause a fundamental issue, as the bias of the stabilized cat qubits can be tuned such that the code performance is effectively limited by the threshold of the single-shot code. Regarding syndrome extraction, very recent work suggests that the ordering of operations in the syndrome extraction circuit does not affect the effective distance of the code, see Ref. [94].

Syndrome extraction based on cat qubits requires bias-preserving CNOT gates, which have not been demonstrated in experiments for stabilized cat qubits so far. Therefore, currently, our estimates for achievable error rates with such gates rely upon theoretical models as proposed, for instance, in Ref. [14] for Kerr-cats and Ref. [19] for dissipative cats. These references also detail the implementation of all other required Clifford operations and Pauli measurements required for the two-dimensional surface code. As there is no fundamental difference in the type of gates required for syndrome extraction in the three-dimensional case, we refer the interested reader to the aforementioned articles.
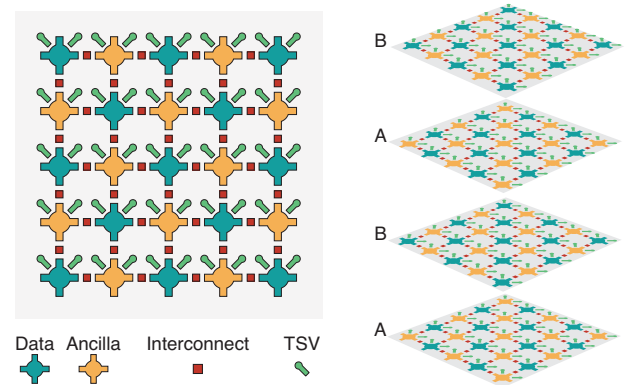


FIG. 12. Sketch of the possible three-dimensional surface code architecture. The grid on the left shows a single two-dimensional layer of data and auxiliary cat qubits arranged with nearest-neighbor connectivity achieved through in-plane interconnects that activate the interaction between the data and auxiliary qubits. *Through-silicon-vias* (TSVs) connect multiple such layers together, one connecting to the layer above, the other to the layer below. The layers are stacked in an ABAB pattern (right), where the difference between A and B is that the placement of data and auxiliary qubits is interchanged.

## C. Conceptional architecture

One could imagine a possible hardware implementation in superconducting circuits as an extension of the proposals in Refs. [14,19], stacking the proposed two-dimensional layouts in a vertical direction in an alternating ABAB pattern as illustrated in Fig. 12. Vertical coupling between chips can be achieved through small form factor superconducting *through-silicon-vias* (TSVs) [95–97]. Although state-of-the-art fabrication techniques currently do not achieve stacking of more than a few layers, the use of TSVs in superconducting circuits is a recent development that will likely mature rapidly in the future [98]. Conceptually, even only a few layers can yield a useful three-dimensional surface code when the noise bias is large enough. The reason is that for the rectangular cubic lattice of spatial extend $L_x, L_y$, and $L_z$, the effective code distances $d_X$ and $d_Z$ are given by [44]

$$d_X = \min\{L_x, L_y, L_z\}, \tag{28}$$

$$d_Z = \min\{L_x L_y, L_y L_z, L_z L_x\}. \tag{29}$$

## VII. CONCLUSION

Recently, there has been progress in the realization of bosonic codes that increase the lifetime of encoded quantum information. Additionally, the discovery of *good* QLDPC codes [25–29] motivates the development of decoding protocols for concatenated bosonic-LDPC codes. These protocols should consider the analog information inherent to the measurement of continuous-variable quantum states.

In this paper, we contribute to this task by presenting methods that feed the analog information obtained during bosonic syndrome measurements into belief propagation and matching decoders. In particular, we show how to decode analog syndromes for single-shot codes that are obtained from higher-dimensional hypergraph product constructions. We also consider codes that are not single shot and thus require repeated stabilizer measurements over time in general. We introduce *analog Tanner graph decoding* as a way of naturally incorporating analog syndrome information directly into the decoding graph.

To support and numerically assess our decoding methods, we consider the three-dimensional surface code as a test case. Our simulations are performed using a phenomenological noise model inspired by bosonic cat code qubits. We find that our analog Tanner graph decoding methods lead to a significantly enhanced sustainable single-shot threshold for the three-dimensional surface code. Furthermore, we show that accounting for analog information from bosonic syndrome measurements can reduce the number of repetitions required for time-domain decoding. We demonstrate this explicitly by incorporating analog Tanner graph methods into an overlapping window decoder for the non-single-shot component of the three-dimensional surface code. For the case of the $L = 13$ three-dimensional surface code, we show that it suffices to decode with a window size of $w = 3$. This is a considerable reduction in the time overhead compared to the case of discrete syndrome decoding where the window size must be equal to the code distance, i.e., $w = 13$. We argue that this renders the three-dimensional surface code *w-quasi-single shot*.

To further boost the development of concatenated bosonic-LDPC codes, we provide open-source software tools for all proposed techniques. With these tools, we hope to emphasize the importance of open-source software and to inspire further research interest into concatenated bosonic codes.

We note that our numerical experiments are performed using a phenomenological noise model. A natural followup to this work will be to further verify the potential of analog Tanner graph decoding under a more realistic circuit-level noise model and to investigate the QSS protocol and the use of analog information in general analytically using appropriate metrics [99]. Moreover, it is an interesting question as to whether other decoders for (3D) QLDPC codes, such as the recently introduced "p-flip decoder" [100] or the three-dimensional tensor network decoder [101], can be modified to incorporate analog information into the decoding process. Finally, it would also be interesting to investigate the performance of analog Tanner graph decoding for other codes, such as three-dimensional subsystem codes [65,102].

This paper has focused on quantum memories. A remaining open problem concerns the questions as to whether analog information can be used to improve decoding performance during the implementation of fault-tolerance logical gates, e.g., during lattice surgery. Such investigations will require a detailed analysis of the physical architecture used to realize the bosonic qubits. For instance, a necessary requirement will be that the qubits support bias-preserving two-qubit gates [58].

## APPENDIX A: $\mathbb{F}_2$ HOMOLOGY

CSS codes are equivalent to three-term chain complexes of binary vector spaces. A chain complex of vector spaces $(C_\bullet, \partial_\bullet)$ is a sequence of vector spaces and linear maps

$$(C_\bullet, \partial_\bullet) = \ldots C_{i+1} \xrightarrow{\partial_{i+1}} C_i \xrightarrow{\partial_i} C_{i-1} \xrightarrow{\partial_{i-1}} \ldots, \quad (A1)$$

with the property

$$\partial_i \partial_{i+1} = 0, \quad \forall i. \quad (A2)$$

The linear maps $\partial_i$ are called *boundary maps* or *boundary operators*. It is standard to define the spaces of *cycles* $Z_i$ and *boundaries* $B_i$ as

$$Z_i := \ker \partial_i \subseteq C_i, \quad (A3)$$

$$B_i := \operatorname{im} \partial_{i+1} \subseteq C_i. \quad (A4)$$

Since Eq. (A2) implies that $Z_i \subseteq B_i$, we can define the quotient

$$\mathcal{H}_i(C_\bullet) := Z_i/B_i, \quad (A5)$$

which is called the *i*th *homology group* of the chain complex.

By inverting the arrows in Eq. (A1), i.e., transposing the corresponding linear maps, we obtain the dual notion called *co-chain complex*

$$C^\bullet := \ldots C_{i+1} \xleftarrow{\partial_i^\top} C_i \xleftarrow{\partial_{i-1}^\top} C_{i-1} \ldots, \quad (A6)$$

and completely analogous definitions for *co-boundaries* $B^i := \operatorname{im} \partial_{i-1}^\top$, *co-cycles* $Z^i := \ker \partial_i^\top$, and the *co-homology group* $\mathcal{H}^i(C^\bullet) = Z^i/B^i$.

By using a three-term subcomplex of a chain complex

$$C_{i+1} \xrightarrow{\partial_{i+1}} C_i \xrightarrow{\partial_i} C_{i-1}, \quad (A7)$$

a CSS code can be obtained by setting

$$H_Z^T = \partial_{i+1}, \quad (A8)$$

$$H_X = \partial_i, \quad (A9)$$

whereby the CSS condition from Eq. (2) is fulfilled by definition. We can now reason about a code in the language of chain complexes and their homology.

The group generated by the $Z$-type stabilizers $S_Z$ correspond to the boundaries $B_i$ and the $Z$-type Pauli operators that commute with all $X$-type stabilizers correspond to the cycles $Z_i$. Analogously, $S_X = B^i$ and the $X$-type Paulis commuting with the $Z$-type stabilizers correspond to the co-cycles $Z^i$. The $Z$-type logical operators correspond to elements of the homology group $\mathcal{H}_i$ and the $X$-type logical operators to the cohomology group $\mathcal{H}^i$.

Note that a linear classical code is a two-term chain complex where the boundary operators map between the space of checks and the code space.

Using the language of homology, codes can be constructed by taking the product of two chain complexes [25–27,103]. The tensor product [104] of two two-term chain complexes $C_1 \xrightarrow{\partial_1^C} C_0$ and $D_1 \xrightarrow{\partial_1^D} C_0$, each corresponding to a classical code, gives a three-term chain complex $C \otimes D$ defined as

$$C_1 \oplus D_1 \xrightarrow{\partial_2} C_1 \oplus D_0 \otimes C_0 \oplus D_1 \xrightarrow{\partial_1} C_0 \otimes D_0, \quad (A10)$$

where the boundary maps are defined as

$$\partial_2 = \begin{pmatrix} \partial_1^C \otimes \mathbb{1} \\ \mathbb{1} \otimes \partial_1^D \end{pmatrix}, \quad (A11)$$

$$\partial_1 = \left( \partial_1^C \otimes \mathbb{1} \mid \mathbb{1} \otimes \partial_1^D \right). \quad (A12)$$

Applying the tensor product to higher-dimensional chain complexes gives a quantum code with higher-dimensional elements.

For example, the repetition (ring) code can be seen as a collection of vertices connected by edges in pairs. The tensor product of two repetition codes then describes a two-dimensional object with faces, edges, and vertices that correspond to the two-dimensional surface code, as illustrated in Fig. 13. Analogously, the three-dimensional surface codes [42] can be obtained as a tensor product of a two-dimensional surface code with a repetition code corresponding to a four-term chain complex (cf. Ref. [39]) as

$$C_3 \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0. \quad (A13)$$
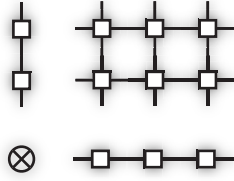
FIG. 13. The two-dimensional surface code obtained from the tensor-product complex of two repetition codes.

*Example 2 (Three-dimensional surface code from repetition codes).* Consider the three-repetition code $R : C_1 \rightarrow C_0$ with check matrix

$$H_{\text{rep}} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}. \tag{A14}$$

A two-dimensional surface code can be obtained by taking $S = R \otimes R$. By taking the tensor product with $R$ again, we obtain a three-dimensional surface code, i.e., a three-dimensional lattice $S_{3D} = S \otimes R$, as sketched in Fig. 3.

Depending on whether we choose period boundary conditions—i.e., a ring code or a repetition code as "seed code"—or not, we obtain the following code parameters of the three-dimensional surface codes. Note that the three-dimensional surface codes with periodic boundaries is also called *three-dimensional toric code* (3DTC),

(a) 3DSC: $[[2L(L-1)^2 + L^3, 1, d_X = L^2, d_Z = L]]$,
(b) 3DTC: $[[3L^3, 3, d_X = L^2, d_Z = L]]$.

Note that, instead of placing $X$ checks on faces and $Z$ checks on vertices, some works consider an assignment with swapped checks.

## APPENDIX B: PROOF OF PROPOSITION 1

Here, we argue that the construction of the multiround parity-check matrix $\tilde{H}$ (cf. Definition 2) for $r$ rounds of syndrome measurement can be described as the tensor product of the chain complex of the code $\mathcal{C}$ and the chain complex of a (slight variant of the) $r$-repetition code $\mathcal{R}$.

The statement is quite straightforward given existing results on product-code constructions and hence the result follows from basic notions from graph theory and homological algebra. However, technically, it is *a priori* not clear that this matches our multiround Tanner graph construction, thus to make these correspondences concrete, we present the result formally in the following.

Let us introduce some additional notation. We consider a two-dimensional *space $X$* as the generalization of a graph (a one-dimensional space) with the $i$-cells, $X_i$, denoting sets of the $i$-dimensional elements, i.e., 0-cells are vertices,

1-cells are edges, 2-cells are faces, and 3-cells are volumes. Analogously to graphs, the *incidence matrices* $\partial_i$ of $X$ are defined as

$$\partial_i^X \in \mathbb{F}_2^{X_i}, (\partial_i^X)_{v,w} = 1 \iff v \sim w, \tag{B1}$$

where $v \sim w$ denotes that $v$ is incident to $w$.

Given a two-dimensional space $X$, the *cellular chain complex* $C_\bullet(X)$ is defined as the chain complex whose vector spaces $C_i$ have the $i$ cells, $X_i$, as basis and boundary maps $\partial_i$ that map an $i$ cell to the formal sum of $(i-1)$ cells at its boundary, e.g., an (edge) to the vertices at its boundary

$$C_\bullet = C_2(X) \rightarrow C_1(X) \rightarrow C_0(X), \tag{B2}$$

where we may identify $C_i(X) = \mathbb{F}_2^{X_i}$, i.e., $C_i(X)$ is the vector space spanned by $i$ cells in $X$, and the boundary operators correspond exactly to the incidence matrices of the $i$ space. Given two graphs (one-dimensional spaces) $X, Y$, their Cartesian product $X \times Y$ is a two-dimensional space $Z$ whose elements are

$$Z_0 = X_0 \times Y_0, \tag{B3}$$

$$Z_1 = X_0 \times Y_1 \coprod X_1 \times Y_0, \tag{B4}$$

$$Z_2 = X_1 \times Z_1, \tag{B5}$$

where the coproduct $\coprod$ is the disjoint sum of sets. The incidence matrices of $Z$ are then given as

$$\partial_1^Z = (\mathbb{1}_{X_0} \otimes \partial_1^Y \mid \partial_1^X \otimes \mathbb{1}_{Y_0}) \in \mathbb{F}_2^{Z_1}, \tag{B6}$$

$$\partial_2^Z = \begin{pmatrix} \partial_1^X \otimes \mathbb{1}_{Y_1} \\ \mathbb{1}_{X_1} \otimes \partial_1^Y \end{pmatrix} \in \mathbb{F}_2^{Z_2}. \tag{B7}$$

Note that this is equivalent to considering the cellular chain complexes $C_\bullet(X)$ and $D_\bullet(Y)$ and constructing the tensor product complex

$$C_\bullet(X) \otimes D_\bullet(Y). \tag{B8}$$

Since $C_\bullet(X), D_\bullet(Y)$ come with bases $X_0, X_1$ and $Y_0, Y_1$, the bases of $C_\bullet(X) \otimes D_\bullet(Y)$ correspond exactly to the spaces obtained by the Cartesian product $X \times Y$ and by ordering the Cartesian products, the matrices of the boundary operators $\partial_i^Z$ are exactly the Kronecker products of the corresponding matrices $\partial_i^X, \partial_i^Y$, hence the boundary map of the tensor product complex are exactly the incidence matrices of the Cartesian product and we can identify $C_\bullet(X) \otimes D_\bullet(Y) = C_\bullet(X \times Y)$. Having the notation in place we can formulate the statement:

*Proposition 2 (Formal version of Proposition 1).* Let $C_\bullet$ be a three-term chain complex corresponding to a CSS

QLDPC code $\mathcal{C}$ and let $R_\bullet = R_1 \xrightarrow{R} R_0$, be a chain complex whose boundary map $R$ corresponds to the $r \times r$ matrix

$$R = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ & & R' & \end{pmatrix},$$

where $R'$ is the $(r-1) \times r$ check matrix of the $r$-repetition code. Then, the $r$-multi-round parity check matrix $\tilde{H}$ of $\mathcal{C}$ is equivalent to the boundary map of the tensor product complex $R_\bullet \otimes C_\bullet$.

*Proof.* Since the code is CSS we focus on a single check side (i.e., the underlying graph of the space) in the following. Let $H \in \mathbb{F}_2^{m \times n}$ be the parity-check matrix of one side of the code $\mathcal{C}$, i.e.,

$$C_1 \xrightarrow{H} C_0.$$

The tensor product chain complex is

$$R_1 \otimes C_1 \xrightarrow{\partial_2} R_0 \otimes C_1 \oplus R_1 \otimes C_0 \xrightarrow{\partial_1} R_0 \otimes C_0, \quad \text{(B9)}$$

where the boundary maps are given by

$$\partial_2 = \begin{pmatrix} \mathbb{1}_R \otimes H \\ R \otimes \mathbb{1}_H \end{pmatrix}, \quad \text{(B10)}$$

$$\partial_1 = (\mathbb{1}_R \otimes H \mid R \otimes \mathbb{1}_H). \quad \text{(B11)}$$

Since qubits are placed on 1 cells, the check matrix given by $\partial_1$ is the one that is relevant.

Viewing $C_\bullet, R_\bullet$ as cellular chain complexes, it is clear that the basis elements of the 1 cells correspond exactly to $Y_0 \times X_1 \coprod Y_0 \times X_1$, where $Y_i, X_i$ are the bases of the $i$-cells of $R$ and $C$, respectively, and $\partial_1$ is exactly the incidence matrix of the underlying 1 complex.

Hence, by the definition of the Kronecker product, the resulting check matrix has the form

$$\partial_1 = \begin{pmatrix} H & & & \mathbb{1}_H & & \\ & H & & \mathbb{1}_H & \mathbb{1}_H & \\ & & \ddots & & & \ddots \\ & & & H & & \mathbb{1}_H & \mathbb{1}_H \end{pmatrix}. \quad \text{(B12)}$$

Thus, $\partial_1 = \tilde{H}(\mathcal{C})$. Since the edge-vertex incidences are given by $\partial_1$ in the corresponding graph whose edges can be identified with the bases $R_0 \times C_1 \coprod R_1 \times C_0$, and whose vertices can be identified with $R_0 \times C_0$, the product graph obtained is equivalent to the multiround Tanner graph $\tilde{\mathcal{T}}$. ∎

Note that to match Definition 2 exactly, we consider a slightly altered check matrix $R$ compared to the standard

repetition code. For example, the check matrix of the four-repetition code is

$$R_{\text{4-rep}} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \text{(B13)}$$

the version we consider is

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}, \quad \text{(B14)}$$

as this accounts for the fact that the first layer of checks is only connected to a single layer of timelike bit nodes. Note that the code spaces of the matrices defined above are not equivalent, since for the repetition code from Eq. (B13) the all-ones vector is the only nontrivial codeword $(1, 1, 1, 1) \in \ker(R_{\text{4-rep}})$, but for the considered variant defined in Eq. (B14) we have $(0, 1, 1, 1) \in \ker(R)$. One could equivalently consider the standard repetition code matrix and then project the final boundary map such that the respective identity block entry is mapped to 0.

## APPENDIX C: IMPLEMENTATION DETAILS

In this section, we present details concerning the code used to conduct the numerical experiments presented in this paper. In Sec. C 1 we review the QLDPC code family used in Sec. IV. Section C 2 reviews the conversion between analog syndrome noise and bitwise syndrome noise channels used in Sec. IV. In Sec. C 3 to Sec. C 5 we review details on belief-propagation decoding and the proposed implementations of ATD and SSMSA.

### 1. Nontopological code constructions

In this section, we give details on the construction of the codes used for numerical evaluations.

### a. Lifted product codes

For the simulations presented in Sec. III, we use a family of *lifted product* (LP) codes [27,41]. The construction of lifted product codes is described below. Algebraically, an $[\![n, k, d]\!]$ LP code can be obtained from the tensor product of a *base* matrix $B$ that corresponds to a classical quasi-cyclic LDPC code [105] with its conjugate transpose $B^*$. The concrete instances of the family used are constructed from the base matrices $B_d$ for distance $d$ from Appendix A

in Ref. [24] and are given as

$$B_{12} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 4 & 7 & 11 \\ 0 & 3 & 10 & 14 & 15 \end{pmatrix}, \quad \text{(C1)}$$

$$B_{16} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 5 & 7 & 17 \\ 0 & 14 & 18 & 12 & 11 \end{pmatrix}, \quad \text{(C2)}$$

$$B_{20} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 14 & 24 & 25 \\ 0 & 16 & 11 & 14 & 13 \end{pmatrix}, \quad \text{(C3)}$$

to obtain the code instances with parameters

   (a) $[[544, 80, d \leq 12]]$,
   (b) $[[714, 100, d \leq 16]]$,
   (c) $[[1020, 136, d \leq 20]]$.

To construct the code instances in software, we use the LDPC library by Roffe [106]. The parity-check matrices are provided in the GitHub repository [107].

## 2. Syndrome noise model conversion

To compare decoding approaches that consider analog or hard syndrome errors, the syndrome noise model under consideration needs to be compatible. This means we need to be able to compare (and convert) the strength of the analog syndrome noise to the hard syndrome noise and vice versa. The analog information decoder considers Gaussian syndrome noise $e_i \sim \mathcal{N}(0, \sigma^2)$. When dealing with syndrome bits $s_i \in \{-1, +1\}$, we want to convert this into an error channel for hard syndrome noise that is equivalent, i.e.,

$$e_i \text{ flips the syndrome} \iff \begin{cases} e_i > 1 & \text{if } s_i = -1, \\ e_i < 1 & \text{if } s_i = +1. \end{cases} \quad \text{(C4)}$$

To satisfy the conditions in Eq. (C4) define the syndrome error rate $p_{\text{syndr}}$ as

$$p_{\text{syndr}} = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{-1} e^{-x^2/2\sigma^2} dx & \text{if } s_i = +1, \\ \frac{1}{\sqrt{2\pi\sigma^2}} \int_{1}^{\infty} e^{-x^2/2\sigma^2} dx & \text{if } s_i = -1. \end{cases} \quad \text{(C5)}$$

By symmetry of the Gaussian distribution, this gives equivalent error probabilities for both cases, which can be derived readily by substituting $x \mapsto -x$,

$$\frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{-1} e^{-x^2/2\sigma^2} dx = \frac{1}{2} \text{Erfc}\left(\frac{1}{\sqrt{2\sigma^2}}\right), \quad \text{(C6)}$$

where $x \mapsto \text{Erfc}(x)$ is the complementary error function. For given $p_{\text{syndr}}$ the solution is

$$\sigma = \frac{\sqrt{2}^{-1}}{\text{Erfc}^{-1}(2p_{\text{sydr}})} = \frac{\sqrt{2}^{-1}}{\text{Erf}^{-1}(1 - 2p_{\text{sydr}})}, \quad \text{(C7)}$$

where $x \mapsto \text{Erfc}^{-1}(x)$ is the inverse of the complementary error function and $\text{Erf}^{-1}(x)$ the inverse of the error function. This allows us to relate the discrete qubit and (analog) cat qubit error models in a one-to-one correspondence.

## 3. Belief propagation

Both the SSMSA decoder proposed in Ref. [68] and our ATD method are based on *belief propagation* (BP), which is a decoding algorithm that has been adapted from classical (LDPC) codes to quantum codes [41,69,108]. In this section, we briefly review the main aspects of BP using minimum-sum update rules relevant to our methods. We refer the reader to the literature in the field for a more in-depth discussion, for instance, Refs. [41,70]. Since we focus on CSS codes that can be seen as a combination of two classical linear codes, we focus on a single check side in the following.

Given a syndrome $s = H \cdot e$, the objective of the decoder is to find the most likely error $e$. In practice, this amounts to finding a minimum (Hamming) weight estimate $\varepsilon$ for the error, i.e.,

$$\varepsilon = \text{argmax}_e \mathbf{Pr}(e|s).$$

In an independent identically distributed (IID) noise model, $\varepsilon$ can be computed bitwise by computing the marginal probabilities

$$\mathbf{Pr}(e_i) = \sum_i^n \mathbf{Pr}(e_1, e_2, \ldots, \hat{e}_i = 1, e_{i+1}, \ldots, e_n|s), \quad \text{(C8)}$$

where the hat $\hat{e}_i$ indicates that the variable is left out, i.e., summation over all variables except $e_i$.

The goal of BP is to compute these probabilities in an iterative way by using the natural factorization given by the Tanner graph of the code (also called the *factor graph* in this context). The marginals $\mathbf{Pr}(e_i)$ are then used to infer an estimate $\varepsilon$ by setting

$$\varepsilon = \begin{cases} 1 & \text{if } \mathbf{Pr}(e_i) \geq 0.5, \\ 0 & \text{otherwise.} \end{cases} \quad \text{(C9)}$$

Belief propagation computes marginals using an iterative message-passing procedure, where in each iteration, a message is sent from each node to its neighbors. The messages constitute sets of "beliefs" on the probabilities to be computed. The value of the messages depends on the syndrome

ALGORITHM 1.    Hard syndrome belief-propagation (MSA) with serial scheduling.

---

1   $s$: Syndrome;
2   $H$: Parity-check matrix;
3   $\mathcal{N}(c)$: Bits in the neighborhood of check $c$;
4   $\mathcal{M}(b)$: Checks in the neighborhood of bit $b$;
5   $\mu_{c,b}$: Check-to-bit update from check $c$ to bit $b$;
6   $\nu_{c,b}$: Bit-to-check update from bit $b$ to check $c$;
7   $\lambda_b = \log((1-p)/p)$: LLR for bit $b$;
8   $p$: Channel probability;
9   bit-count: The number of bits;
10  max-iter: The maximum no. BP iterations to run;
    **Result:** estimate $\varepsilon$
11  **for** *all* $c, b$ *where* $H_{c,b} \neq 0$ **do** // Initialization
12  $\quad$ $\nu_{b,c} := \log((1-p)/p)$
13  **for** *iter to max-iter* **do** // Main iteration loop
14  $\quad$ **for** $b \in \{1, \dots, bit\text{-}count\}$ **do** // Serial bit loop
15  $\quad\quad$ $\lambda_b = \log((1-p)/p)$;   // Initialise LLRs
16  $\quad\quad$ **for** $c \in \mathcal{M}(b)$ **do** // Loop over neighbors
17  $\quad\quad\quad$ $|\mu_{c,b}| = \min_{b' \in \mathcal{N}(c) \setminus \{b\}} |\nu_{b',c}|$;
18  $\quad\quad\quad$ $\text{sgn}(\mu_{c,b}) = \text{sgn}(\lambda_b) \cdot \prod_{b' \in \mathcal{N}(c) \setminus \{b\}} \text{sgn}(\nu_{c,b'})$;
19  $\quad\quad\quad$ $\mu_{c,b} = \alpha \cdot \text{sgn}(\mu_{c,b}) \cdot |\mu_{c,b}|$;   // Check to bit
20  $\quad\quad\quad$ $\lambda_b = \lambda_b + \mu_{c,b}$;          // Update LLRs
21  $\quad\quad$ **for** $c \in \mathcal{M}(b)$ **do**
22  $\quad\quad\quad$ $\nu_{b,c} = \lambda_b - \mu_{c,b}$;        // Bit to check
23  $\quad\quad$ **if** $\lambda_b \leq 0$ **then** // Hard decision on bit $b$
24  $\quad\quad\quad$ $\varepsilon_b = 1$;
25  $\quad\quad$ **else**
26  $\quad\quad\quad$ $\varepsilon_b = 0$;
27  $\quad$ **if** $s = H \cdot \varepsilon$ **then**
28  $\quad\quad$ **return** $\varepsilon$;          // Converged, return estimate.
29  **return** $\varepsilon$;    // Return estimate after maximum number of iterations reached.

---

and the bit-error channel. In this work, we use a serial schedule to compute BP marginals. Our implementation is outlined in Algorithm 1 and is described below.

In the first step of BP decoding, the values of the bit nodes $b_j$ are initialized to the *log-likelihood ratios* (LLRs) $\lambda_i$ of the error channel

$$\lambda_i := \log\left(\frac{1-p}{p}\right). \tag{C10}$$

In every iteration, each check node $i$ sends messages to neighboring bit nodes $j \in \mathcal{N}(i)$, denoted as $\mu_{i,j}$. The value of the check-to-bit message $\mu_{i,j}$ is computed as a function of the syndrome $\gamma_i$ and the incoming bit-to-check messages $\nu_{i,j'}$ as

$$\alpha \cdot \text{sgn}(\gamma_i) \cdot \prod_{j' \in \mathcal{N}(i) \setminus \{j\}} \text{sgn}(\nu_{i,j'}) \cdot \min_{j' \in \mathcal{N}(i) \setminus \{j\}} |\nu_{i,j'}|. \tag{C11}$$

The factor $\alpha \in \mathbb{R}$ is called the *scaling factor* [109]. The bit-to-check node messages $\nu_{j,i}$ are computed as

$$\mu_{j,i} := \lambda_j + \sum_{i' \in \mathcal{N}(j) \setminus i} \mu_{i'j}.$$

It is well known that BP computes only the marginals $\mathbf{Pr}(e_i)$ exactly on factor graphs that are trees (in a single step). In the more general setting, where the graph contains loops, the computed marginals are approximate [110]. To check for termination of the iterative procedure, we use the marginals and Eq. (C9) to infer an estimate $\varepsilon$ from the currently computed marginals and check if $\varepsilon$ is valid for the given syndrome. i.e., if $s = H \cdot \varepsilon$ the solution $\varepsilon$ is valid. If $\varepsilon$ is valid, the BP algorithm terminates and $\varepsilon$ is returned as the decoding estimate.

We use the BP+OSD implementation provided in the LDPC2 package by Roffe *et al.* available on Github [111].

### 4. Soft-syndrome MSA

The SSMSA algorithm is essentially equivalent to BP as sketched in Algorithm 1 with some alterations. Instead of the hard syndrome vector $s$, the input is an analog syndrome vector $\tilde{s}$, whose corresponding LLR vector is denoted $\gamma$ [cf. Eq. (12)]. The initialization and bit-to-check messages are computed equivalently. For computing the check-to-bit messages, the analog syndrome $\gamma$ is taken into account. If the syndrome value is below a predefined cutoff value $\Gamma$ that models the reliability of the syndrome information, the syndrome information is treated as unreliable and the messages $\mu_{c,b}$ are instead computed as

$$\mu_{c,b} := \begin{cases} \min_{b' \in \mathcal{N}(c) \setminus b}(|\nu_{b',c}|) & \text{if } |\gamma_c| > \Gamma, \\ \min_{b' \in \mathcal{N}(c) \setminus b}(|\nu_{b',c}|, |\gamma_c|) & \text{otherwise.} \end{cases} \tag{C12}$$

Note that there is a case in which Algorithm 2 erases the analog syndrome information. This occurs when the absolute value of the analog syndrome is smaller than the value of all incoming messages of the check node $c$, and the sign of the incoming messages matches the sign of the syndrome. From Line 22 of Algorithm 2, we see that in this case the analog syndrome value is overwritten and thus lost.

Our implementation of the SSMSA decoder is made publicly available in the LDPC2 package [111].

### 5. Analog Tanner graph decoder

In *analog Tanner graph decoding*, we use the Tanner graph of the code $\mathcal{T}$ to construct the *analog Tanner graph*. This allows us to directly incorporate the analog syndrome information in *virtual nodes* in the Factor graph.

ALGORITHM 2.   Soft-Syndrome MSA.

---

1  $\gamma_i$: Analog syndrome LLR;
2  $H$: Parity-check matrix;
3  $\mathcal{N}(c)$: Bits in the neighborhood of check $c$;
4  $\mathcal{M}(b)$: Checks in the neighborhood of bit $b$;
5  $\mu_{c,b}$: Check-to-bit update from check $b$ to bit $b$;
6  $\nu_{c,b}$: Bit-to-check update from bit $b$ to check $c$;
7  $\lambda_b$: LLR for bit $b$;
8  $\Gamma$: Cutoff for soft-info decoding;
9  $p$: Channel probability;
10  bit-count: the number of bits;
11  max-iter: The maximum no. BP iterations;
   **Result:** estimate $\varepsilon$
12  **for** *all $c, b$ where $H_{c,b} \neq 0$* **do** // Initialization
13  |    $\nu_{b,c} := \log((1-p)/p)$;
14  **for** *iter to max-iter* **do** // Main iteration loop
15  |    **for** $b \in \{1..bit\text{-}count\}$ **do** // Serial bit loop
16  |    |    $\lambda_b = \log((1-p)/p)$;   // Initialise LLRs
17  |    |    **for** $c \in \mathcal{M}(b)$ **do** // Loop over check bits
18  |    |    |    **if** $|\gamma_c| \leq \Gamma$ **then** // Virtual check update
19  |    |    |    |    $|\mu_{c,b}| = \min\limits_{b' \in \mathcal{N}(c)}(|\nu_{b',c}|, |\gamma_c|)$;
20  |    |    |    |    **if** $|\gamma_c| < \min\limits_{b' \in \mathcal{N}(c)}(|\nu_{b',c}|)$ **then**
21  |    |    |    |    |    **if** $sgn(\gamma_c) == \prod\limits_{b' \in \mathcal{N}(c)} sgn(\nu_{b',c})$ **then**
22  |    |    |    |    |    |    $\gamma_c = sgn(\gamma_c) \cdot \min\limits_{b' \in \mathcal{N}(c)}(|\nu_{b',c}|)$;
23  |    |    |    |    |    **else**
24  |    |    |    |    |    |    $\gamma_c = -1 \times \gamma_c$;
25  |    |    |    **else** // Default to MSA if above cutoff
26  |    |    |    |    $|\mu_{c,b}| = \min\limits_{b' \in \mathcal{N}(c)\backslash\{b\}} |\nu_{b',c}|$;
27  |    |    |    $sgn(\mu_{c,b}) = sgn(\gamma_c) \cdot \prod\limits_{b' \in \mathcal{N}(c)\backslash\{b\}} sgn(\nu_{b',c})$;
28  |    |    |    $\mu_{c,b} = sgn(|\mu_{c,b}|) \cdot |\mu_{c,b}|$;   // Check to bit
29  |    |    |    $\lambda_b = \lambda_b + \mu_{c,b}$;       // Update LLRs
30  |    |    **for** $c \in \mathcal{M}(b)$ **do**
31  |    |    |    $\nu_{b,c} = \lambda_b - \mu_{c,b}$;       // Bit to check
32  |    |    **if** $\lambda_b \leq 0$ **then** // Hard decision on bit $b$
33  |    |    |    $\varepsilon_b = 1$;
34  |    |    **else**
35  |    |    |    $\varepsilon_b = 0$;
36  |    $s_c = sgn(\gamma_c)$;             // Hard syndrome
37  |    **if** $s = H \cdot \varepsilon$ **then**
38  |    |    return $\varepsilon$;          // Converged, return estimate.
39  return $\varepsilon$;    // Return estimate after maximum number of iterations reached.

---

In the initialization phase of the ATD decoder, BP sets the values $\lambda_i, i \in [n]$ of all bit nodes in the factor graph to

$$\lambda_i = \log\left(\frac{1-p}{p}\right), \quad i \in [n]. \tag{C13}$$

To ensure that we initialize the values of the analog nodes $\lambda_{n+j}, j \in [m]$ with the value of the analog syndrome $\gamma_j$, we

derive the error channel probabilities $p_i'$ from Eq. (C13)

$$p_i' = \frac{1}{e^{\gamma_i} + 1}. \tag{C14}$$

Thus we set

$$p_{n+j}' = \frac{1}{e^{\gamma_j} + 1}, \quad j \in [m] \tag{C15}$$

for the analog nodes to ensure that after the initialization phase of BP the bit nodes are initialized with the LLRs, and the virtual nodes with the analog syndrome (i.e., the LLR) values.

## APPENDIX D: OBTAINING ANALOG INFORMATION FOR DATA QUBITS AND CONCATENATED GKP CODES

One might raise the question as to whether it is possible to include more analog information in the decoding graph, e.g., by considering analog values associated with qubits (data nodes in the factor graph used for decoding) as well. The answer to this question is positive under the assumption that one performs active error correction on the bosonic qubit. In the considered phenomenological noise model that is inspired by stabilized cat qubits, this is not the case as the cat qubit is autonomously protected by the engineered stabilization mechanism as discussed in Sec. VI A. We emphasize that in a gate-based scheme, it is not possible to perform active error correction on cat qubits, i.e., measure their stabilizers, as cat codes cannot be described as a stabilizer code in the conventional sense. Fundamentally, this is due to the nonexistence of a phase operator in the harmonic oscillator Hilbert space.

Although in some cases, for example, dissipative stabilized cat qubits, information about certain types of errors can also be obtained by continuously monitoring the buffer mode that is used to implement the dissipation mechanism [112], doing so with sufficiently high reliability seems to be a task of similar complexity as implementing a QEC protocol. Another way of incorporating qubit analog information into the decoding graph requires a changing the computing paradigm from a gate-based scheme to the measurement-based scheme in which qubits are explicitly measured. The measured value will be real valued and its magnitude can be used to assign LLR to that specific qubit.

However, if the bosonic code is actively corrected in some way, one can use this information in the decoder as well. A straightforward example is the single-mode GKP code as it is a stabilizer code with generators given by the displacements in phase space $\mathbb{R}^2$

$$\langle S_X := e^{2i\sqrt{\pi}\hat{p}}, \quad S_Z := e^{-2i\sqrt{\pi}\hat{x}} \rangle, \tag{D1}$$

where $\hat{p}$ and $\hat{x}$ are the momentum and position operators of the harmonic oscillators. Here,

$$e^{2i\sqrt{\pi}\hat{p}} = D(-\sqrt{2}(1,0)^T), \quad e^{-2i\sqrt{\pi}\hat{x}} = D(\sqrt{2}(0,1)^T) \tag{D2}$$

with

$$D(\xi) := \exp\left(-i\sqrt{2\pi}\xi^T(\hat{p}, -\hat{x})^T\right) \tag{D3}$$

being the single-mode shift operator, generating shifts in phase space by $\xi \in \mathbb{R}^2$. One can measure the stabilizer generators using Steane-type error correction, which requires auxiliary GKP qubits [23]. Assuming the availability of noiseless auxiliary qubits, the Steane-type error correction determines the shift error the data qubit has undergone modulo. This means that shift errors up to a magnitude of at most $\sqrt{\pi}/2$ can be corrected, while shift errors that have a larger magnitude typically lead to logical errors in the GKP qubit subspace. Suppose that the measurement yields an outcome $x_m$ for the $x$-quadrature shifts. Then, if $|x_m \mod \sqrt{2\pi}| \approx 0$ it is unlikely that this qubit has undergone a logical GKP error if we assume as above that shift errors follow a Gaussian distribution $x_m \sim \mathcal{N}(0, \sigma^2)$ with mean zero and variance $\sigma^2$. However, if $|x_m \mod \sqrt{2\pi}| \approx \sqrt{\pi}/2$, a logical GKP qubit error is significantly more likely. Thus, it is possible to quantify this likelihood and use it to also initialize the data nodes of the decoding graph [black circles in Figs. 1(d)–1(f)] with analog information and then apply ATD to decode. See also Fig. 14 for a visualization of the likelihood function in analogy to Fig. 1(c).
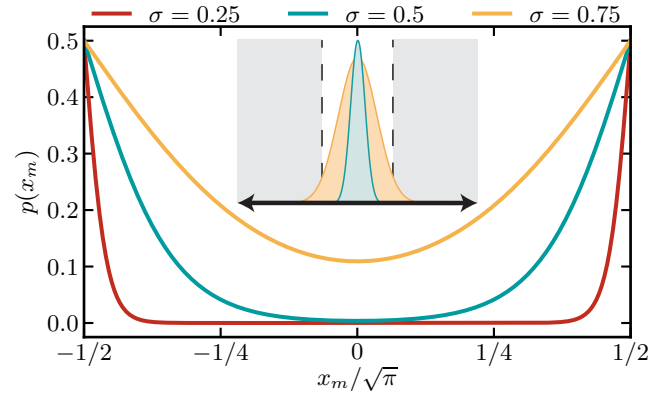


FIG. 14. Obtaining analog information from GKP data qubits. The main panel shows the GKP qubit error probability conditioned on the measurement outcome $x_m$ for different values of noise strength $\sigma$. Detecting a value $x_m$ close to zero corresponds to a small error probability for any value of $\sigma$, while the error probability for measurement outcomes closer to the "decision boundaries" $\pm\sqrt{\pi}/2$ are highly dependent on the assumed noise channel and its variance $\sigma^2$. The inset shows a sketch of the distribution of measurement outcomes $x_m$ for two different values of $\sigma$ and the dashed lines indicate the decision boundaries.

## APPENDIX E: CONCATENATED MULTIMODE GKP AND ROTATION SYMMETRIC BOSONIC CODES

In a measurement-based setting, analog information can be extracted through teleportation-based Knill-type error correction [113]. This works also for *multimode instances of the GKP code* [114–116]. Generally, one can think of multimode encodings beyond single-mode encodings in terms of displacement operators

$$D(\xi) := \exp\left\{-i\sqrt{2\pi}\xi^T J(\hat{p}_1, \ldots, \hat{p}_m, -\hat{x}_1, \ldots, -\hat{x}_m)^T\right\} \tag{E1}$$

in phase space $\mathbb{R}^{2m \times 2m}$ of $m$ bosonic modes, with

$$J := \begin{bmatrix} 0 & \mathbb{1}_m \\ -\mathbb{1}_m & 0 \end{bmatrix}. \tag{E2}$$

Then one can generally define a stabilizer group of a GKP code in terms of displacements

$$\langle D(\xi_1), \ldots, D(\xi_{2m}) \rangle, \tag{E3}$$

where $\xi_1, \ldots, \xi_{2m} \in \mathbb{R}^{2m \times 2m}$ are linearly independent and we have that $\xi_i^T J \xi_j \in \mathbb{Z}$ for all $i, j$. This defines a stabilizer group isomorphic to a lattice [116]. In this way, one obtains multimode information in the syndrome measurements. For this reason, the methods introduced here also contribute to the question of how to decode GKP codes. In a similar way, one can consider large classes of alternative bosonic codes known as *rotation symmetric bosonic codes* [117,118]. Both our decoding techniques and software tools are easily adapted to incorporate analog information for the data qubits, as explained above.

## APPENDIX F: ADDITIONAL RESULTS

In this section, we present additional simulation results for various parameter settings of the considered decoder implementations. We also present results for the phenomenological threshold of the three-dimensional toric code (i.e., the 3DSC with period boundaries).

### 1. Phenomenological noise threshold of the three-dimensional toric code

As a by-product of our QSS simulations, we obtain a threshold of the non-single-shot side of the 3DTC under phenomenological noise of $\approx 1.26\%$ and $\approx 1.66\%$ using *hard syndromes* (HS) and *analog syndromes* (AS), respectively. The corresponding threshold plots are shown in Fig. 15. This generalizes recent code capacity results presented in Ref. [57]. Here, we simulated a pure bit-flip noise
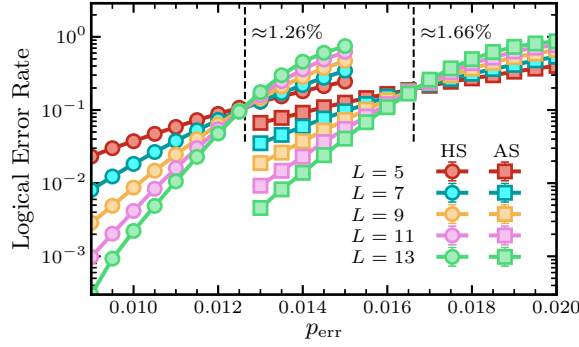
FIG. 15. Phenomenological noise threshold of the three-dimensional toric code. The logical error rates are obtained by simulating a pure bit-flip noise model with $p_{\text{err}}^{\text{data}} = p_{\text{err}}^{\text{synd}}$. The syndrome is collected in $2L$ measurement rounds of which the last one is noiseless. We decode using minimum weight perfect matching that yields a threshold at approximately equal to 1.26% and approximately equal to 1.66% using the *hard syndrome* (HS) and the *analog syndrome* (AS), respectively.

model with $p_{\text{err}}^{\text{data}} = p_{\text{err}}^{\text{synd}}$. Measurement results are collected over $2L$ rounds and then decoded using minimum-weight perfect matching. For the case of hard syndromes, the noise from the syndrome is added as discussed in Sec. II C 2. This ensures the effective error channel for hard syndromes is the same as that for analog syndromes. We use the PyMatching implementation of minimum-weight-perfect matching [80,81] as the decoder for these simulations.

### 2. BP parameter optimization

Since there are various parameters that allow one to fine tune the BP+OSD implementation, we conducted a set of numerical experiments to determine which parameter setting performs the best in the scenario considered. However, note that for this work we focus on techniques and methods rather than low-level optimizations in general. Hence, the presented numerical results are generally implementation dependent and most likely prone to further optimization and fine tuning.

In summary, the conducted numerical experiments demonstrate that there can be quite significant differences in decoding performance (in terms of achieved logical error rate, threshold, number of BP iterations, and elapsed time) depending on the BP+OSD parameters, most notably the chosen BP scaling factor $\alpha$ (cf. Sec. C 3) and the OSD method.

The main findings are that OSD-cs with a scaling factor $\alpha = 0.5$ to $\alpha = 0.6$ perform best. Furthermore, concerning the SSMSA implementation, we find that a cutoff of $\Gamma = 5$ performs the best for the considered lifted product code family. For readers interested in more detailed results,

we refer to the GitHub repository [107], where we present detailed simulation results.

### APPENDIX G: NUMERICAL SIMULATION DETAILS

In this section, we discuss the implementation details of numerical simulations and the main techniques used. Since we focus on CSS codes using depolarizing (biased) noise without correlations, we decode $X$ and $Z$ errors separately (that implies that a Y error on a qubit $q$ is treated as $X$ and $Z$ error on $q$). To determine the *logical error rate* of an $[[n, k, d]]$ single-shot code, decoder combination in the considered phenomenological (cat qubit) noise model, we use the following procedure:

(1) Sample an error vector $e \in \mathbb{F}_2^n$ (bitwise and dependent on the error channel.

(2) Compute the syndrome $s = H \cdot e$.

(3) Sample and apply a syndrome error $e_s \in \mathbb{F}_2^m$ to obtain the noisy syndrome $\hat{s}$.

(4) Apply (analog) single-stage decoding to obtain an estimate $\varepsilon$.

(5) Check if $r = \varepsilon + e$ is a logical.

The sample run is successful if $r$ is not a logical operator (i.e., the correction induced a stabilizer, which does not alter the encoded logical information).

We apply a similar procedure to estimate the logical error rate for a code and decoder combination when applying the (analog) overlapping window method to decode over time with multiple syndrome measurements. To simulate decoding in $R$ rounds of noisy syndrome measurement, we first compute $R$ noisy syndromes and then apply overlapping window decoding as described in the main text to obtain a single sample. We repeat this procedure for a maximum number of samples $N$.

The logical $X$ and $Z$ error rate $p_\ell^{X/Z}$ is the fraction of failed runs $n_{\text{fail}}$ for $N$ total samples

$$p_\ell^{X/Z} := \frac{n_{\text{fail}}}{N}. \tag{G1}$$

The error bars $e_{X/Z}$ are computed by

$$e_{X/Z}^2 = (1 - p_\ell)\frac{p_\ell}{N}. \tag{G2}$$

The simulations are terminated if the maximum number of samples $N$ is reached, or if the errors fall below a certain precision cutoff determined by the ratio of the error bar value and the logical error rate, which we set to $10^{-1}$.

To give a better comparison of logical error rates for codes that encode a different number of logical qubits $k$, we use the *word error rate* (WER) for codes with $k > 1$, which intuitively can be understood as logical error rate

per logical qubit. The WER $p_w$ is computed as

$$p_w = 1 - (1 - p_\ell)^{1/k-1}. \quad \text{(G3)}$$

We can use the methods described above to obtain a *threshold* $p_{th}$ estimation by computing the logical error rates for code instances of a code family with increasing distance and increasing physical error rate and then estimating where the graphs cross. We use a standard approach based on a finite-size scaling regression analysis [57,119,120].

To this end, we perform a quadratic fit on the logical error rate data obtained by numerical experiments. Let $p_{th}$ be the threshold physical error rate we want to determine, $\mu > 0$ a parameter called the *critical exponent*, and define the *rescaled physical error rate*

$$x := (p - p_{th}) \cdot d^{1/\mu}, \quad \text{(G4)}$$

which is $x = 0$ at $p = p_{th}$. We use $x$ to fit the simulation data to the following quadratic ansatz $\Phi(p, d)$

$$\Phi(p, d) := ax^2 + bx + c, \quad \text{(G5)}$$

where $a$, $b$, and $c$ are coefficients of the quadratic ansatz and are parameters to be determined by fitting the data. Note that $\Phi(p, d)$ is only a valid approximation near $p = p_{th}$, therefore only data points close to this region were used to compute the fit. Given this ansatz, we use the logical error rates $p_\ell$ [cf. Eq. (G1)] to obtain the free parameters $(p_{th}, \mu, a, b, c)$ by computing a fit using the minimized mean square error.

Note that, since we focus on CSS codes, where the $X$ and $Z$ decoding is done separately (assuming noncorrelated errors), we need to compute the *combined* logical error rates from the separate experimental data. Given a $X$ and $Z$ logical error rates $p_\ell^X, p_\ell^Z$ from numerical experiments, we compute the combined logical error rate $p_\ell$ as

$$p_\ell := p_\ell^X \cdot (1 - p_\ell^Z) + p_\ell^Z \cdot (1 - p_\ell^X) + p_\ell^X \cdot p_\ell^Z. \quad \text{(G6)}$$

Moreover, the corresponding errors are computed using standard methods for *propagation of uncertainty*. i.e., we use the variance formula [121]

$$s_f^2 := \left(\frac{\partial f}{\partial x}\right)^2 s_x^2 + \left(\frac{\partial f}{\partial y}\right)^2 s_y^2 + \left(\frac{\partial f}{\partial z}\right)^2 s_z^2 + \cdots, \quad \text{(G7)}$$

where $s_f$ is the standard deviation of $f$, and $s_k$ the standard deviation of $k \in \{x, y, z, \ldots\}$. To be concrete, we apply the formula above to the separately computed errors $e_x, e_z$, to obtain the overall error $e$ with

$$e^2 := e_x^2 \cdot (1 - e_z)^2 + e_z^2 \cdot (1 - e_x)^2. \quad \text{(G8)}$$

## APPENDIX H: OPEN-SOURCE SOFTWARE

All our techniques are available in the form of open-source software available on GitHub [107] as part of the *Munich Quantum Toolkit* (MQT) and partly in the LDPC2 package [111]. With the proposed software, we hope to provide a set of useful tools for analog information decoding and to emphasize the need for open-source implementations to foster public review, reproducibility, and extendability.

[1] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, Trapped-ion quantum computing: Progress and challenges, Appl. Phys. Rev **6**, 021314 (2019).

[2] D. J. Wineland, Nobel lecture: Superposition, entanglement, and raising Schrödinger's cat, Rev. Mod. Phys. **85**, 1103 (2013).

[3] M. H. Devoret and R. J. Schoelkopf, Superconducting circuits for quantum information: An outlook, Science **339**, 1169 (2013).

[4] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, Semiconductor spin qubits, Rev. Mod. Phys. **95**, 025003 (2023).

[5] L. Henriet, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, Quantum **4**, 327 (2020).

[6] A. Deshpande, P. Niroula, O. Shtanko, A. V. Gorshkov, B. Fefferman, and M. J. Gullans, Tight bounds on the convergence of noisy random circuits to the uniform distribution, PRX Quantum **3**, 040329 (2022).

[7] D. Stilck França and R. García-Patrón, Limitations of optimization algorithms on noisy quantum devices, Nat. Phys. **17**, 1221 (2021).

[8] P. T. Cochrane, G. J. Milburn, and W. J. Munro, Macroscopically distinct quantum-superposition states as a bosonic code for amplitude damping, Phys. Rev. A **59**, 2631 (1999).

[9] D. Gottesman, A. Kitaev, and J. Preskill, Encoding a qubit in an oscillator, Phys. Rev. A **64**, 012310 (2001).

[10] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. M. Girvin, L. Jiang, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, Extending the lifetime of a quantum bit with error correction in superconducting circuits, Nature **536**, 441 (2016).

[11] V. V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios, S. Ganjam, A. Miano, B. L. Brock, A. Z. Ding, L. Frunzio, S. M. Girvin, R. J. Schoelkopf, and M. H. Devoret, Real-time quantum error correction beyond break-even, Nature **616**, 50 (2023).

[12] A. Grimm, N. E. Frattini, S. Puri, S. O. Mundhada, S. Touzard, M. Mirrahimi, S. M. Girvin, S. Shankar, and M. H. Devoret, Stabilization and operation of a Kerr-cat qubit, Nature **584**, 205 (2020).

[13] R. Lescanne, M. Villiers, T. Peronnin, A. Sarlette, M. Delbecq, B. Huard, T. Kontos, M. Mirrahimi, and Z. Leghtas,

Exponential suppression of bit-flips in a qubit encoded in an oscillator, Nat. Phys. **16**, 509 (2020).

[14] A. S. Darmawan, B. J. Brown, A. L. Grimsmo, D. K. Tuckett, and S. Puri, Practical quantum error correction with the XZZX code and Kerr-cat qubits, PRX Quantum **2**, 030345 (2021).

[15] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, The XZZX surface code, Nat. Commun. **12**, 2172 (2021).

[16] K. Tiurev, P.-J. H. S. Derks, J. Roffe, J. Eisert, and J.-M. Reiner, Correcting non-independent and non-identically distributed errors with surface codes, Quantum **7**, 1123 (2023).

[17] J. Guillaud and M. Mirrahimi, Repetition cat qubits for fault-tolerant quantum computation, Phys. Rev. X **9**, 041053 (2019).

[18] J. Guillaud and M. Mirrahimi, Error rates and resource overheads of repetition cat qubits, Phys. Rev. A **103**, 042413 (2021).

[19] C. Chamberland, K. Noh, P. Arrangoiz-Arriola, E. T. Campbell, C. T. Hann, J. Iverson, H. Putterman, T. C. Bohdanowicz, S. T. Flammia, A. Keller, G. Refael, J. Preskill, L. Jiang, A. H. Safavi-Naeini, O. Painter, and F. G. Brandão, Building a fault-tolerant quantum computer using concatenated cat codes, PRX Quantum **3**, 010329 (2022).

[20] F.-M. L. Régent, C. Berdou, Z. Leghtas, J. Guillaud, and M. Mirrahimi, High-performance repetition cat code using fast noisy operations, Quantum **7**, 1198 (2023).

[21] M. P. Stafford and N. C. Menicucci, Biased Gottesman-Kitaev-Preskill repetition code, Phys. Rev. A **108**, 052428 (2023).

[22] C. Vuillot, H. Asasi, Y. Wang, L. P. Pryadko, and B. M. Terhal, Quantum error correction with the toric Gottesman-Kitaev-Preskill code, Phys. Rev. A **99**, 032344 (2019).

[23] K. Noh and C. Chamberland, Fault-tolerant bosonic quantum error correction with the surface–Gottesman-Kitaev-Preskill code, Phys. Rev. A **101**, 012316 (2020).

[24] N. Raveendran, N. Rengaswamy, F. Rozpędek, A. Raina, L. Jiang, and B. Vasić, Finite rate GLDPC-GKP coding scheme that surpasses the CSS Hamming bound, Quantum **6**, 767 (2022).

[25] M. B. Hastings, J. Haah, and R. O'Donnell, in *Proc. 53rd Annu. ACM SIGACT Symp. Theory Comput.*, STOC 2021 (Association for Computing Machinery, New York, NY, USA, 2021), p. 1276.

[26] N. P. Breuckmann and J. N. Eberhardt, Balanced product quantum codes, IEEE Trans. Inf. Theory **67**, 6653 (2021).

[27] P. Panteleev and G. Kalachev, in *Proc. 54th Annu. ACM SIGACT Symp. Theory Comput.*, STOC 2022 (Association for Computing Machinery, New York, NY, USA, 2022), p. 375.

[28] I. Dinur, M.-H. Hsieh, T.-C. Lin, and T. Vidick, Good quantum LDPC codes with linear time decoders, ArXiv:2206.07750.

[29] A. Leverrier and G. Zémor, Quantum tanner codes, ArXiv:2202.13641.

[30] S. Bravyi and B. Terhal, A no-go theorem for a two-dimensional self-correcting quantum memory based on stabilizer codes, New J. Phys. **11**, 043029 (2009).

[31] N. Baspin and A. Krishna, Connectivity constrains quantum codes, Quantum **6**, 711 (2022).

[32] N. Baspin and A. Krishna, Quantifying nonlocality: How outperforming local quantum codes is expensive, Phys. Rev. Lett. **129**, 050505 (2022).

[33] N. Baspin, V. Guruswami, A. Krishna, and R. Li, Improved rate-distance trade-offs for quantum codes with restricted connectivity, ArXiv:2307.03283.

[34] N. Baspin, O. Fawzi, and A. Shayeghi, A lower bound on the overhead of quantum error correction in low dimensions, ArXiv:2302.04317.

[35] Q. Xu, J. P. B. Ataides, C. A. Pattison, N. Raveendran, D. Bluvstein, J. Wurtz, B. Vasic, M. D. Lukin, L. Jiang, and H. Zhou, Constant-overhead fault-tolerant quantum computation with reconfigurable atom arrays, ArXiv:2308.08648.

[36] J. Ramette, J. Sinclair, Z. Vendeiro, A. Rudelis, M. Cetina, and V. Vuletić, Any-to-any connected cavity-mediated architecture for quantum computing with trapped ions or Rydberg arrays, PRX Quantum **3**, 010344 (2022).

[37] D. Barredo, V. Lienhard, S. de Léséleuc, T. Lahaye, and A. Browaeys, Synthetic three-dimensional atomic structures assembled atom by atom, Nature **561**, 79 (2018).

[38] A. Omran, H. Levine, A. Keesling, G. Semeghini, T. T. Wang, S. Ebadi, H. Bernien, A. S. Zibrov, H. Pichler, S. Choi, J. Cui, M. Rossignolo, P. Rembold, S. Montangero, T. Calarco, M. Endres, M. Greiner, V. Vuletić, and M. D. Lukin, Generation and manipulation of Schrödinger cat states in Rydberg atom arrays, Science **365**, 570 (2019).

[39] A. Strikis and L. Berent, Quantum low-density parity-check codes for modular architectures, PRX Quantum **4**, 020321 (2023).

[40] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, ArXiv:2308.07915.

[41] P. Panteleev and G. Kalachev, Degenerate quantum LDPC codes with good finite length performance, Quantum **5**, 585 (2021).

[42] M. Vasmer and D. E. Browne, Three-dimensional surface codes: Transversal gates and fault-tolerant architectures, Phys. Rev. A **100**, 012312 (2019).

[43] H. Bombín, Single-shot fault-tolerant quantum error correction, Phys. Rev. X **5**, 031043 (2015).

[44] A. O. Quintavalle, M. Vasmer, J. Roffe, and E. T. Campbell, Single-shot error correction of three-dimensional homological product codes, PRX Quantum **2**, 020340 (2021).

[45] O. Higgott and N. P. Breuckmann, Improved single-shot decoding of higher-dimensional hypergraph-product codes, PRX Quantum **4**, 020332 (2023).

[46] P. W. Shor, in *Proc. 37th Conf. Found. Comput. Sci.* (IEEE, Burlington, VT, USA, 1996), p. 56.

[47] N. Delfosse, B. W. Reichardt, and K. M. Svore, Beyond single-shot fault-tolerant quantum error correction, IEEE Trans. Inf. Theory **68**, 287 (2022).

[48] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).

[49] L. Skoric, D. E. Browne, K. M. Barnes, N. I. Gillespie, and E. T. Campbell, Parallel window decoding

enables scalable fault tolerant quantum computation, ArXiv:2209.08552.

[50] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, ArXiv:quant-ph/9511026.

[51] A. Joshi, K. Noh, and Y. Y. Gao, Quantum information processing with bosonic qubits in circuit QED, Quantum Sci. Technol. **6**, 033001 (2021).

[52] W. Cai, Y. Ma, W. Wang, C.-L. Zou, and L. Sun, Bosonic quantum error correction codes in superconducting quantum circuits, Fundam. Res. **1**, 50 (2021).

[53] J. Guillaud, Repetition Cat Qubits, Theses, École Normale Supérieure, 2021, available online at https://hal.archives-ouvertes.fr/tel-03509305.

[54] D. K. Tuckett, S. D. Bartlett, and S. T. Flammia, Ultrahigh error threshold for surface codes with biased noise, Phys. Rev. Lett. **120**, 050505 (2018).

[55] D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, Fault-tolerant thresholds for the surface code in excess of 5% under biased noise, Phys. Rev. Lett. **124**, 130501 (2020).

[56] J. Roffe, L. Z. Cohen, A. O. Quintavalle, D. Chandra, and E. T. Campbell, Bias-tailored quantum LDPC codes, Quantum **7**, 1005 (2023).

[57] E. Huang, A. Pesah, C. T. Chubb, M. Vasmer, and A. Dua, Tailoring three-dimensional topological codes for biased noise, ArXiv:2211.02116.

[58] S. Puri, L. St-Jean, J. A. Gross, A. Grimm, N. E. Frattini, P. S. Iyer, A. Krishna, S. Touzard, L. Jiang, A. Blais, S. T. Flammia, and S. M. Girvin, Bias-preserving gates with stabilized cat qubits, Sci. Adv. **6**, eaay5901 (2020).

[59] F. Swiadek, R. Shillito, P. Magnard, A. Remm, C. Hellings, N. Lacroix, Q. Ficheux, D. C. Zanuz, G. J. Norris, A. Blais, S. Krinner, and A. Wallraff, Enhancing dispersive readout of superconducting qubits through dynamic control of the dispersive shift: Experiment and theory, ArXiv:2307.07765.

[60] D. Gottesman, Stabilizer codes and quantum error correction, ArXiv:quant-ph/9705052.

[61] H. Bombin and M. A. Martin-Delgado, Topological quantum distillation, Phys. Rev. Lett. **97**, 180501 (2006).

[62] S. B. Bravyi and A. Y. Kitaev, Quantum codes on a lattice with boundary, ArXiv:quant-ph/9811052.

[63] A. Yu. Kitaev, Fault-tolerant quantum computation by anyons, Ann. Phys. (N. Y.) **303**, 2 (2003).

[64] A. Hamma, P. Zanardi, and X.-G. Wen, String and membrane condensation on three-dimensional lattices, Phys. Rev. B **72**, 035307 (2005).

[65] A. Kubica and M. Vasmer, Single-shot quantum error correction with the three-dimensional subsystem toric code, Nat. Commun. **13**, 6272 (2022).

[66] E. T. Campbell, A theory of single-shot error correction for adversarial noise, Quantum Sci. Technol. **4**, 025006 (2019).

[67] A. M. Steane, Error correcting codes in quantum theory, Phys. Rev. Lett. **77**, 793 (1996).

[68] N. Raveendran, N. Rengaswamy, A. Pradhan, and B. Vasic, in *2022 IEEE Int. Conf. Quantum Comput. Eng. QCE* (IEEE Computer Society, Los Alamitos, CA, USA, 2022), p. 275.

[69] M. Fossorier and S. Lin, Soft-decision decoding of linear block codes based on ordered statistics, IEEE Trans. Inf. Theory **41**, 1379 (1995).

[70] J. Roffe, D. R. White, S. Burton, and E. Campbell, Decoding across the quantum low-density parity-check code landscape, Phys. Rev. Res. **2**, 043423 (2020).

[71] K.-Y. Kuo, I.-C. Chern, and C.-Y. Lai, in *2021 IEEE Int. Symp. Inf. Theory ISIT* (IEEE, Melbourne, Australia, 2021), p. 1552.

[72] M. Li and T. J. Yoder, in *2020 IEEE Int. Conf. Quantum Comput. Eng. QCE* (IEEE, Denver, CO, USA, 2020), Vol. 109.

[73] A. Grospellier, L. Grouès, A. Krishna, and A. Leverrier, Combining hard and soft decoders for hypergraph product codes, Quantum **5**, 432 (2021).

[74] N. P. Breuckmann and V. Londe, Single-shot decoding of linear rate LDPC quantum codes with high performance, IEEE Trans. Inf. Theory **68**, 272 (2022).

[75] A. Ashikhmin, C.-Y. Lai, and T. A. Brun, in *2014 IEEE Int. Symp. Inf. Theory* (IEEE, Honolulu, HI, USA, 2014), p. 546.

[76] O. Higgott, T. C. Bohdanowicz, A. Kubica, S. T. Flammia, and E. T. Campbell, Improved decoding of circuit noise and fragile boundaries of tailored surface codes, Phys. Rev. X **13**, 031007 (2023).

[77] A. G. Fowler, A. M. Stephens, and P. Groszkowski, High-threshold universal quantum computation on the surface code, Phys. Rev. A **80**, 052312 (2009).

[78] C. A. Pattison, M. E. Beverland, M. P. da Silva, and N. Delfosse, Improved quantum error correction using soft information, ArXiv:2107.13589.

[79] K.-Y. Kuo and C.-Y. Lai, Correcting phenomenological quantum noise via belief propagation, ArXiv:2310.12682.

[80] O. Higgott, Pymatching: A python package for decoding quantum codes with minimum-weight perfect matching, ACM Trans. Quantum Computing **3**, 16 (2022).

[81] O. Higgott and C. Gidney, Sparse blossom: Correcting a million errors per core second with minimum-weight matching, ArXiv:2303.15933.

[82] D. F. Walls and G. J. Milburn, *Quantum Optics* (Springer-Verlag, Berlin Heidelberg, 2008), 2nd ed.

[83] Intuitively, one can recognize that these states are orthogonal by noting that $|+\rangle_{\text{cat}}$ is invariant under the exchange $\alpha \leftrightarrow -\alpha$ while $|-\rangle_{\text{cat}}$ obtains a global phase. This makes them $\pm 1$ eigenstates of the parity operator $\hat{\Pi} = \exp(i\pi \hat{a}^{\dagger}\hat{a})$, respectively, and thus orthogonal.

[84] S. Puri, S. Boutin, and A. Blais, Engineering the quantum states of light in a Kerr-nonlinear resonator by two-photon driving, npj Quantum Inf. **3**, 18 (2017).

[85] M. Mirrahimi, Z. Leghtas, V. V. Albert, S. Touzard, R. J. Schoelkopf, L. Jiang, and M. H. Devoret, Dynamically protected cat-qubits: A new paradigm for universal quantum computation, New J. Phys. **16**, 045014 (2014).

[86] R. Gautier, A. Sarlette, and M. Mirrahimi, Combined dissipative and hamiltonian confinement of cat qubits, PRX Quantum **3**, 020339 (2022).

[87] V. V. Albert, Lindbladians with multiple steady states: Theory and applications, Ph.D. thesis, Yale University, 2018.

[88] It should be emphasized that state-of-the-art experiments with dissipatively stabilized cat qubits cannot arbitrarily suppress bit-flip errors, however. The current understanding is that this is due to additional noise channels caused by auxiliary (few level) qubits, e.g., transmon qubits, used for control and readout [122].

[89] G. Lindblad, On the generators of quantum dynamical semigroups, Commun. Math. Phys. **48**, 119 (1976).

[90] V. Gorini, A. Kossakowski, and E. C. G. Sudarshan, Completely positive dynamical semigroups of n-level systems, J. Math. Phys. **17**, 821 (1976).

[91] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, New York, 2010), 10th ed.

[92] E. Knill and R. Laflamme, Theory of quantum error-correcting codes, Phys. Rev. A **55**, 900 (1997).

[93] T. Hillmann and F. Quijandría, Quantum error correction with dissipatively stabilized squeezed-cat qubits, Phys. Rev. A **107**, 032423 (2023).

[94] A. G. Manes and J. Claes, Distance-preserving stabilizer measurements in hypergraph product codes, ArXiv:2308.15520.

[95] D. R. W. Yost, M. E. Schwartz, J. Mallek, D. Rosenberg, C. Stull, J. L. Yoder, G. Calusine, M. Cook, R. Das, A. L. Day, E. B. Golden, D. K. Kim, A. Melville, B. M. Niedzielski, W. Woods, A. J. Kerman, and W. D. Oliver, Solid-state qubits integrated with superconducting through-silicon vias, npj Quantum Inf. **6**, 1 (2020).

[96] J. L. Mallek, D.-R. W. Yost, D. Rosenberg, J. L. Yoder, G. Calusine, M. Cook, R. Das, A. Day, E. Golden, D. K. Kim, J. Knecht, B. M. Niedzielski, M. Schwartz, A. Sevi, C. Stull, W. Woods, A. J. Kerman, and W. D. Oliver, Fabrication of superconducting through-silicon vias, ArXiv:2103.08536.

[97] K. Grigoras, *et al.*, Qubit-compatible substrates with superconducting through-silicon vias, IEEE Trans. Quantum Eng. **3**, 1 (2022).

[98] T. M. Hazard, W. Woods, D. Rosenberg, R. Das, C. F. Hirjibehedin, D. K. Kim, J. Knecht, J. Mallek, A. Melville, B. M. Niedzielski, K. Serniak, K. M. Sliwa, D. Ruth-Yost, J. L. Yoder, W. D. Oliver, and M. E. Schwartz, Characterization of superconducting through-silicon vias as capacitive elements in quantum circuits, ArXiv:2308.00834.

[99] B. D'Anjou, Generalized figure of merit for qubit readout, Phys. Rev. A **103**, 042404 (2021).

[100] T. R. Scruby and K. Nemoto, Local probabilistic decoding of a quantum code, Quantum **7**, 1093 (2023).

[101] C. Piveteau, C. T. Chubb, and J. M. Renes, Tensor network decoding beyond 2D, ArXiv:2310.10722.

[102] J. C. Bridgeman, A. Kubica, and M. Vasmer, Lifting topological codes: Three-dimensional subsystem codes from two-dimensional anyon models, ArXiv:2305.06365.

[103] J.-P. Tillich and G. Zémor, Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Trans. Inf. Theory **60**, 1193 (2014).

[104] Formally this is denoted as the total complex of the tensor product double complex [26].

[105] M. Fossorier, Quasicyclic low-density parity-check codes from circulant permutation matrices, IEEE Trans. Inf. Theory **50**, 1788 (2004).

[106] J. Roffe, Ldpc: Python tools for low density parity check codes (2022), https://pypi.org/project/ldpc/.

[107] https://github.com/cda-tum/mqt-qecc.

[108] M. Fossorier, Iterative reliability-based decoding of low-density parity check codes, IEEE J. Sel. Areas Commun. **19**, 908 (2001).

[109] A. A. Emran and M. Elsabrouty, in *2014 IEEE 11th Consum. Commun. Netw. Conf. CCNC* (2014), p. 518.

[110] F. Kschischang, B. Frey, and H.-A. Loeliger, Factor graphs and the sum-product algorithm, IEEE Trans. Inform. Theory **47**, 498 (2001).

[111] https://github.com/quantumgizmos/ldpc/tree/ldpc_v2.

[112] R. Gautier, M. Mirrahimi, and A. Sarlette, Designing high-fidelity gates for dissipative cat qubits, ArXiv:2303.00760.

[113] E. Knill, Fault-tolerant postselected quantum computation: Schemes, ArXiv:quant-ph/0402171.

[114] K. Noh, C. Chamberland, and F. G. Brandão, Low-overhead fault-tolerant quantum error correction with the surface-GKP code, PRX Quantum **3**, 010315 (2022).

[115] M. V. Larsen, C. Chamberland, K. Noh, J. S. Neergaard-Nielsen, and U. L. Andersen, Fault-tolerant continuous-variable measurement-based quantum computation architecture, PRX Quantum **2**, 030325 (2021).

[116] J. Conrad, J. Eisert, and F. Arzani, Gottesman-Kitaev-Preskill codes: A lattice perspective, Quantum **6**, 648 (2022).

[117] A. L. Grimsmo, J. Combes, and B. Q. Baragiola, Quantum computing with rotation-symmetric bosonic codes, Phys. Rev. X **10**, 011058 (2020).

[118] T. Hillmann, F. Quijandría, A. L. Grimsmo, and G. Ferrini, Performance of teleportation-based error-correction circuits for bosonic codes with noisy measurements, PRX Quantum **3**, 020334 (2022).

[119] C. Wang, J. Harrington, and J. Preskill, Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory, Ann. Phys. (N. Y.) **303**, 31 (2003).

[120] C. T. Chubb and S. T. Flammia, Statistical mechanical models for quantum codes with correlated noise, Ann. Inst. Henri Poincaré Comb. Phys. Interact. **8**, 269 (2021).

[121] H. H. Ku, Notes on the use of propagation of error formulas, J. Res. Natl. Bur. Stand. (U. S.) **70C**, 263 (1966).

[122] C. Berdou, *et al.*, One hundred second bit-flip time in a two-photon dissipative oscillator, PRX Quantum **4**, 020350 (2023).