

Fault-tolerant non-Clifford Gottesman-Kitaev-Preskill gates using polynomial phase gates and on-demand noise biasing

Minh T. P. Nguyen^{1,2,*} and Mackenzie H. Shaw^{1,3}

¹*QuTech, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands*

²*Kavli Institute of Nanoscience, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands*

³*Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands*



(Received 7 December 2025; revised 30 March 2026; accepted 3 June 2026; published 27 July 2026)

The Gottesman-Kitaev-Preskill (GKP) error correcting code uses a bosonic mode to encode a logical qubit and has the attractive property that its logical Clifford gates can be implemented using Gaussian unitary gates. In contrast, a direct unitary implementation of the T gate using the cubic phase gate has been shown to have a logical error floor unless the GKP codestate has a biased-noise profile. In this work, we propose a method for on-demand noise biasing based on a standard GKP error-correction circuit. This on-demand biasing circuit can be used to bias the GKP codestate before a T gate and return it to a nonbiased state afterward. With the on-demand biasing circuit, we prove that the logical error rate of the T gate can be made arbitrarily small as the quality of the GKP codestates increases. We complement our proof with a numerical investigation of the cubic phase gate subject to a phenomenological noise model, showing that the T gate can achieve average gate fidelities above 99%, with 12 dB of GKP squeezing without the use of postselection. Moreover, we develop a formalism for finding optimal unitary representations of logical diagonal gates in higher levels of the Clifford hierarchy that is based on a framework of “polynomial phase stabilizers” whose exponents are polynomial functions of one of the quadrature operators. This formalism naturally extends to multiqubit logical gates and even to number-phase bosonic codes, providing a powerful algebraic tool for analyzing non-Clifford gates in bosonic quantum codes.

DOI: [10.1103/2pg9-nsn8](https://doi.org/10.1103/2pg9-nsn8)

I. INTRODUCTION

Quantum error correction (QEC) is widely believed to be necessary for large-scale, fault-tolerant quantum computing. In discrete-variable (DV) QEC, logical qubits are encoded into many physical qubits [1,2]. However, the required resource overhead remains challenging for near-term hardware. On the other hand, continuous-variable (CV) QEC offers a more hardware-efficient alternative by encoding logical information into a single infinite-dimensional bosonic Hilbert space rather than multiple two-level systems [3,4]. The logical information stored in such a code may already be low enough to perform an algorithm, or, more likely, the error rate would be low enough to concatenate with a code with lower overhead than would be possible using DV qubits.

Among the CV codes, the Gottesman-Kitaev-Preskill (GKP) code [5] emerges as one of the leading candidates since all Clifford operations can be implemented unitarily using Gaussian operations [5]. Moreover, when concatenated with a DV code, the decoded information given by the inner GKP code can be used to improve the decoder of the outer DV code

[6–9], reducing the required hardware overhead. Approximate GKP states have been realized experimentally in multiple platforms, including superconducting circuits [10–12], trapped ions [13,14], and integrated photonics [15].

Nevertheless, achieving universal quantum computation requires the inclusion of a non-Clifford gate, typically the $T = \text{diag}(1, e^{i\pi/4})$ gate. In the original GKP proposal [5], it was shown that the logical T gate can be implemented unitarily using a *cubic phase gate*, which is the exponential of a cubic polynomial of the CV position operator. However, Hartrup *et al.* [16] recently showed that—in the absence of noise biasing—the fidelity of the cubic phase gate is upper bounded when applied to approximate GKP states, even in the limit of infinite squeezing and even when the states are decoded using ideal QEC. To circumvent this bound, one needs to perform *on-demand* noise biasing, since Clifford gates (such as the Hadamard gate) can alter the noise bias during the algorithm.

In this work, we reexamine the possibility of implementing the logical T gate in the GKP code unitarily. We show that the unitary T gate is both fault tolerant and outperforms the vacuum-state injection method in a practical regime of interest. Key to our work is the observation that on-demand noise biasing can be performed using a standard GKP syndrome extraction circuit. This on-demand biasing circuit allows us to convert between a nonbiased GKP code for Clifford gates and a biased GKP code for T gates with the same overhead as is required for standard GKP QEC, as illustrated in Fig. 1. With this noise biasing, we prove that the T gate implemented is fault tolerant, meaning that the logical gate fidelity goes to 1 as

*Contact author: m.t.phamnguyen@tudelft.nl

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

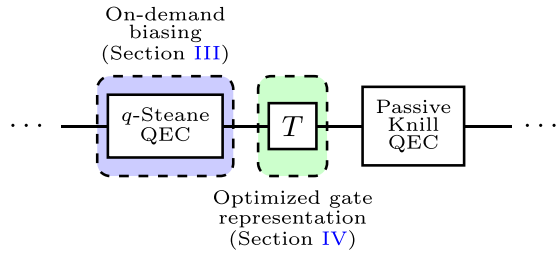


FIG. 1. Fault-tolerant circuit for implementing the logical T gate using the cubic phase gate. The q -Steane QEC circuit [Eq. (13a)] is applied before the T gate to perform on-demand noise biasing of the GKP qubit. After the gate, the passive Knill QEC circuit [Eq. (14)] resets the noise profile for subsequent computation. The implemented cubic phase gate can be optimized using our polynomial phase stabilizer framework to obtain a representation that minimizes error propagation.

the quality of the GKP codestate state improves. We compare the unitary T gate implementations with the vacuum-state injection method numerically under a phenomenological noise model and find that for the vacuum-state method to achieve the same performance as the T gate a postselection probability below 20% is necessary for GKP states with $\Delta < 0.25$ (which corresponds to 12 dB of squeezing).

Alongside our analysis of the cubic phase gate, we also develop an analytical framework of *polynomial* phase gates that can implement gates in higher levels of the Clifford hierarchy. Central to this framework is the introduction of the *polynomial phase stabilizer*, that is, any operator that acts trivially on the GKP codespace but whose exponent is a polynomial function of one of the quadrature operators. Using this, we develop a method to minimize the coefficients of the polynomial required to implement a given logical gate, enabling the systematic identification of optimal gate representations. For example, we derive a representation of the logical T gate that first appeared in Ref. [17], which we show to be optimal and to significantly outperform the original construction in Ref. [5]. We also numerically investigated optimized polynomial phase gates for the $T^{1/2}$, $T^{1/4}$, and $T^{1/8}$ gates, and found that these higher-order gates can achieve higher average gate fidelities than the T gate in some regimes. Moreover, the framework applies to any gate whose exponent is a polynomial of one or more commuting position, momentum, or number operators, and therefore naturally generalizes to multiqubit GKP gates as well as other bosonic codes, such as the multileg rotationally symmetric code [18] and the number-phase code [19].

The article is organized as follows. In Sec. II, we briefly review the square and rectangular GKP codes and introduce the noise models considered in this work. In Sec. III, we explain why the fidelity of the cubic phase gate is fundamentally limited when applied to nonbiased approximate GKP states [16]. We then present the on-demand noise biasing circuit based on the Steane QEC circuit [5], which is applied immediately before the T gate, as shown in Fig. 1. In Sec. IV, we introduce the concept of polynomial phase stabilizers and use it to classify all representations of logical diagonal non-Clifford gates in the GKP code and identify the optimal representations. In Sec. V, we analyze the propagation of errors when polynomial

phase gates are applied to approximate GKP states and prove our fault-tolerance theorem. Finally, in Sec. VI, we numerically compute the average gate fidelity of the polynomial phase gates using on-demand noise biasing and compare their performance against the vacuum-state method [20], before providing concluding remarks in Sec. VII. While there remain significant challenges in performing our fault-tolerant unitary T gate in experimental platforms such as the implementation of the unitary polynomial phase gates in photonics or the suppression of DV qubit errors in superconducting platforms, our work provides a proof of principle that using polynomial phase gates can reduce the required overhead of implementing non-Clifford logical gates in the GKP code.

A. Relationship to existing literature

There are a variety of current approaches to realize fault-tolerant T gates in the GKP code that we now briefly summarize and compare. The current leading approach to perform a T gate is magic-state preparation by measuring the GKP stabilizers of a vacuum state [20], followed by injection and a nonlinear feedforward operation [17]. The main advantage of this scheme is simplicity as it requires only Gaussian operations alongside standard GKP error correction. However, achieving high-fidelity magic states with the vacuum-state method requires postselection, significantly increasing the time overhead. In comparison, our approach instead assumes the additional capability to implement a unitary cubic phase gate. We show that the resulting magic-state fidelity exceeds that of the vacuum-state method with moderate postselection rates, potentially reducing the subsequent cost of magic-state distillation.

An alternative approach is to implement a fault-tolerant logical T through dissipative engineering. For example, Ref. [21] introduces a pieceable-circuit technique to realize logical phase gates by utilizing DV ancillas as bath, building on results of Refs. [22,23]. However, this approach is primarily tailored to superconducting platforms and typically requires multiple rounds—along with biased-noise DV ancillas—to achieve high-fidelity performance, which can lead to significant time overhead.

In our work, we limit ourselves to approaches performing the T gate that do not use DV qubits for two reasons. First, some hardware platforms, such as in optics, do not have easy access to DV qubits. And second, we want to be able to prove the fault tolerance of our approach in the limit of increasing GKP codestate quality. DV qubits complicate this picture because one has to also consider the effects of noise arising from the DV qubits themselves.

Finally, depending on the physical implementation of the GKP qubits, the dissipative bath may be realized directly. This idea has recently been explored in superconducting platforms [24,25], where a dissipative element (e.g., a resistor) is coupled to the GKP mode through a controllable switch that can be used to implement a logical T gate. We note that the optimal gate representations that we identified in Sec. IV—which realize logical T and $\sqrt{T}$ gates using quartic and quadratic potentials—are naturally implementable within the same scheme, as also pointed out by the authors of Ref. [25]. The primary challenge here lies in realizing a

picosecond-scale switch that has not yet been demonstrated experimentally.

II. PRELIMINARIES

In this section, we introduce the notation and review the states, gates, and QEC circuits for the single-mode square/rectangular GKP code.

For a single CV mode, we use both the ladder operators $a, a^\dagger$ and the position q and momentum p operators that obey the canonical commutation relations $[a, a^\dagger] = 1$ and $[q, p] = i$. We denote the position x eigenstates as $|x\rangle_q$. We define the single-mode displacement operator $W(\mathbf{v})$, where $\mathbf{v} = (v_q, v_p) \in \mathbb{R}^2$, as

$$W(\mathbf{v}) = W(v_q, v_p) = \exp[i\sqrt{2\pi}(v_p q - v_q p)]. \quad (1)$$

Note that our definition differs from the conventional one (e.g., in Ref. [26]) by a factor of $\sqrt{\pi}$. The displacement operators form a group under multiplication, with the commutation relation

$$W(\mathbf{u})W(\mathbf{v}) = e^{-i2\pi(u_q v_p - u_p v_q)} W(\mathbf{v})W(\mathbf{u}) \quad (2)$$

and the composition rule

$$W(\mathbf{u})W(\mathbf{v}) = e^{-i\pi(u_q v_p - u_p v_q)} W(\mathbf{u} + \mathbf{v}). \quad (3)$$

Displacement operators act on the quadrature operators via

$$W(\mathbf{u})^\dagger q W(\mathbf{u}) = q + \sqrt{2\pi} v_q, \quad (4a)$$

$$W(\mathbf{u})^\dagger p W(\mathbf{u}) = p + \sqrt{2\pi} v_p. \quad (4b)$$

In this work, we will make use of both square and rectangular single-mode GKP codes. Similar to DV stabilizer codes, the GKP code can be defined in terms of its stabilizers [5]

$$S_p^{(\lambda)} = W(\sqrt{2\lambda}, 0), \quad S_q^{(\lambda)} = W\left(0, \sqrt{\frac{2}{\lambda}}\right), \quad (5)$$

where $\lambda > 0$. When $\lambda = 1$, this corresponds to the standard square GKP code, in which case we drop the λ label from the stabilizers. In contrast to the square GKP code, the rectangular GKP code protects one quadrature at the expense of the other. As a result, it can offer improved performance under biased-noise models, where errors are more likely to occur in one quadrature than the other. We will make use of this property later, where we will show that the logical T gate causes an effectively asymmetric noise model to arise on the GKP code.

The logical Pauli operators are the displacement operators that commute with the stabilizers, given by

$$X^{(\lambda)} = \sqrt{S_p^{(\lambda)}} = W(\sqrt{\lambda/2}, 0), \quad (6a)$$

$$Z^{(\lambda)} = \sqrt{S_q^{(\lambda)}} = W(0, 1/\sqrt{2\lambda}). \quad (6b)$$

The logical computational states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ are supported on even and odd integer multiples of $\sqrt{\lambda\pi}$ in the position basis:

$$|\bar{0}\rangle^\lambda = \sum_{n \in \mathbb{Z}} |2n\sqrt{\lambda\pi}\rangle_q, \quad |\bar{1}\rangle^\lambda = \sum_{n \in \mathbb{Z}} |(2n+1)\sqrt{\lambda\pi}\rangle_q. \quad (7)$$

Logical Clifford gates in the GKP code are given by Gaussian unitary operators (i.e., operators with an exponent that is a quadratic function of q and p) and can be implemented by [5]

$$H^{(\lambda)} = \exp\left(i\frac{\pi}{4}(q^2/\lambda + \lambda p^2)\right), \quad (8a)$$

$$S^{(\lambda)} = \exp\left(\frac{i}{2\lambda}q^2\right), \quad (8b)$$

$$CX^{(\lambda)} = \exp(-iq_1 p_2), \quad (8c)$$

where we use subscripts to label the two modes participating in the $CX^{(\lambda)}$ gate. Throughout this work, we will also make frequent use of two other Gaussian unitary operators that do not represent logical Clifford gates: the squeezing and beam-splitter operators defined by

$$\text{Sq}(\alpha) = \exp\left(i\frac{\ln(\alpha)}{2}(qp + pq)\right), \quad (9a)$$

$$\text{BS}(\theta) = \exp(i\theta(p_1 q_2 - q_1 p_2)). \quad (9b)$$

The squeezing operator is named because it “squeezes” the quadrature operators

$$\text{Sq}(\alpha)q\text{Sq}(\alpha)^\dagger = \alpha q, \quad \text{Sq}(\alpha)p\text{Sq}(\alpha)^\dagger = p/\alpha. \quad (10)$$

Moreover, squeezing operators can be used to convert between square and rectangular GKP codes:

$$|\bar{\psi}\rangle^\lambda = \text{Sq}(\sqrt{\lambda})^\dagger |\bar{\psi}\rangle, \quad O^{(\lambda)} = \text{Sq}(\sqrt{\lambda})^\dagger O \text{Sq}(\sqrt{\lambda}), \quad (11)$$

for any codestate $|\bar{\psi}\rangle$ and logical operator or stabilizer O .

To perform error correction on the GKP code, we need to measure the two stabilizers S_q, S_p , which is equivalent to measuring the position and momentum operators modulo $\sqrt{\lambda\pi}$ and $\sqrt{\pi/\lambda}$ (respectively). Then, given a measured modular position q_m and momentum p_m , the state can then be returned to the GKP codespace by applying the corrective displacement $W(q_m, p_m)^\dagger$. A displacement error $W(\mathbf{v})$ acting on an ideal GKP state can therefore be detected and corrected if the displacement satisfies [5,27]

$$-\frac{\sqrt{\lambda}}{2\sqrt{2}} \leq v_q < \frac{\sqrt{\lambda}}{2\sqrt{2}}, \quad -\frac{1}{2\sqrt{2\lambda}} \leq v_p < \frac{1}{2\sqrt{2\lambda}}. \quad (12)$$

A simple circuit that can be used to measure the stabilizers is the Steane QEC circuit. This consists of two subcircuits (which can be implemented either sequentially or in parallel) that measure the S_q and S_p stabilizers, respectively. These are given explicitly by

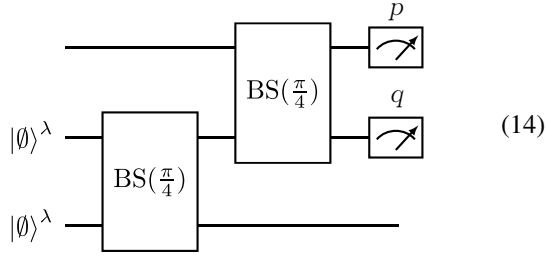
$$\begin{array}{c} |\bar{\psi}\rangle^\lambda \\ |\bar{+}\rangle^\lambda \end{array} \rightarrow \begin{array}{c} \boxed{e^{-iq_1 p_2}} \\ \text{---} \end{array} \rightarrow \begin{array}{c} \text{---} \\ \boxed{q} \end{array}, \quad (13a)$$

$$\begin{array}{c} |\bar{\psi}\rangle^\lambda \\ |\bar{0}\rangle^\lambda \end{array} \rightarrow \begin{array}{c} \boxed{e^{ip_1 q_2}} \\ \text{---} \end{array} \rightarrow \begin{array}{c} \text{---} \\ \boxed{p} \end{array}, \quad (13b)$$

where the measurements represent homodyne measurements of the q or p operators. We refer to Eq. (13a) as the q -Steane

QEC circuit since it measures S_q , and to Eq. (13b) as the p -Steane QEC circuit. We will show in Sec. III that the q -Steane QEC circuit allows us to perform the on-demand biasing that is required to perform. By tracking how a displacement operator $W(\mathbf{v})$ propagates through the CX gates, we find that an initial position displacement error on the data qubit propagates through the q -Steane QEC circuit to the ancilla and is detected by the position homodyne measurement (and likewise for momentum displacement errors in the p -Steane QEC circuit).

In our analysis, we also use another type of syndrome extraction circuit that we call the passive Knill QEC circuit, given by [28]



Here, the state $|\emptyset\rangle^\lambda$ is defined as

$$|\emptyset\rangle^\lambda = \sum_{n \in \mathbb{Z}} |n\sqrt{2\lambda\pi}\rangle_q. \quad (15)$$

The passive Knill QEC circuit is convenient for a number of reasons: it uses only passive (i.e., energy preserving) Gaussian unitary operators, it is more robust against noise on the ancilla modes than the Steane QEC circuit [28], and the output state is guaranteed to be in the GKP codespace with only a logical correction needing to be applied after the circuit.

So far, we have only considered ideal GKP codewords [Eq. (7)]. In practice, however, only approximate GKP states can be generated, since the ideal GKP codewords in Eq. (7) require infinite energy to realize. We define the approximate codestates as

$$|\bar{\psi}\rangle_\Delta^\lambda = e^{-\Delta^2(q^2+p^2)/2} |\bar{\psi}\rangle^\lambda, \quad (16)$$

where we refer to $e^{-\Delta^2(q^2+p^2)/2} = e^{-\Delta^2 a^\dagger a} = \text{Env}_\Delta$ as the (nonbiased) envelope operator.

In this work, we also need to define *biased* approximate GKP codestates, which are obtained by applying a biased envelope operator given by

$$\text{Env}_{\Delta_q, \Delta_p} = e^{-(\Delta_q^2 q^2 + \Delta_p^2 p^2)/2} \quad (17a)$$

$$= \text{Sq}(\sqrt{\lambda}) e^{-\Delta^2(q^2+p^2)/2} \text{Sq}(\sqrt{\lambda})^\dagger, \quad (17b)$$

where $\Delta_q = \Delta/\sqrt{\lambda}$ and $\Delta_p = \sqrt{\lambda}\Delta$. The biased approximate codestate $|\bar{\psi}\rangle_{\Delta_q, \Delta_p}^\lambda = \text{Env}_{\Delta_q, \Delta_p} |\bar{\psi}\rangle^\lambda$ is the same as the ideal codestate but where in the position basis, each Dirac-delta function in Eq. (7) has been replaced by a Gaussian function of width $\sim \Delta_q$ and multiplied by a Gaussian envelope of width $\sim 1/\Delta_p$ (and vice versa in the momentum basis). This means that the approximate codestates are normalizable, but also that approximate codestates have inherent errors on them due to the width of the Gaussian functions.

Importantly, we can always map a *biased* square GKP qubit to a *nonbiased* rectangular GKP qubit using a unitary

squeezing operator

$$|\bar{\psi}\rangle_{\Delta_q, \Delta_p}^\lambda = \text{Sq}(\sqrt{\lambda}) |\bar{\psi}\rangle_\Delta^\lambda, \quad (18)$$

where $\lambda = \Delta_p/\Delta_q$ and $\Delta = \sqrt{\Delta_q \Delta_p}$. Therefore, in the absence of other noise, the biased square GKP qubit $|\bar{\psi}\rangle_{\Delta_q, \Delta_p}^\lambda$ is equivalent to the nonbiased rectangular GKP qubit $|\bar{\psi}\rangle_\Delta^\lambda$ in terms of the logical information and error rate.

Since the GKP code is designed to protect against displacement errors that are small enough as in Eq. (12), a common way to analyze an error operator or channel is to decompose it into displacement operators. Because the displacement operators form a complete orthogonal basis for the operator space, any operator O can be decomposed in terms of displacement operators as

$$O = \int d\mathbf{v} \chi_O(\mathbf{v}) W(\mathbf{v}), \quad (19)$$

where the characteristic function χ_O of operator O is given by

$$\chi_O(\mathbf{v}) = \text{Tr}[O W^\dagger(\mathbf{v})]. \quad (20)$$

Analogously, any quantum channel $\mathcal{E}$ can be decomposed in terms of displacement operators as

$$\mathcal{E}(\rho) = \int d\mathbf{v} d\mathbf{u} \chi_{\mathcal{E}}(\mathbf{v}, \mathbf{u}) W(\mathbf{v}) \rho W(\mathbf{u})^\dagger, \quad (21)$$

where the χ -function $\chi_{\mathcal{E}}(\mathbf{u}, \mathbf{v})$ is a CV generalization of the DV χ -matrix and corresponds to the characteristic function of the superoperator representation of $\mathcal{E}$ [27]. In our analysis, we frequently employ the twirling approximation, which retains only the diagonal components of Eq. (21):

$$\text{Twirl}[\mathcal{E}](\rho) = \int d\mathbf{v} p_{\mathcal{E}}(\mathbf{v}) W(\mathbf{v}) \rho W(\mathbf{v})^\dagger. \quad (22)$$

for a more detailed discussion, see Ref. [29]. The twirling approximation is useful for us because Eq. (22) can be viewed as applying a displacement $W(\mathbf{v})$ randomly according to the probability density $p_{\mathcal{E}}(\mathbf{v}) = \chi_{\mathcal{E}}(\mathbf{v}, \mathbf{v})$. This is convenient because we can quickly determine the probability that the random displacement is correctable by integrating $p_{\mathcal{E}}(\mathbf{v})$ over the set of correctable displacements [Eq. (12)].

When $p_{\mathcal{E}}(\mathbf{v})$ is a Gaussian distribution, we call the twirled channel a Gaussian random displacement channel

$$\mathcal{G}_\Sigma(\rho) = \int d\mathbf{v} \mathcal{N}(\Sigma, \mathbf{v}) W(\mathbf{v}) \rho W(\mathbf{v})^\dagger, \quad (23)$$

where we use a slightly nonstandard definition of the two-dimensional (2D) Gaussian distribution $\mathcal{N}(\Sigma, \mathbf{v})$ given by

$$\mathcal{N}(\Sigma, \mathbf{v}) = \frac{\exp[-\pi \mathbf{v}^T \Sigma^{-1} \mathbf{v}]}{\sqrt{|\det \Sigma|}}. \quad (24)$$

Here, Σ is the covariance matrix of the distribution (rescaled by a factor of $\sqrt{2\pi}$).

It was shown in Ref. [27] that the biased envelope operator $\text{Env}_{\Delta_q, \Delta_p}$ has a characteristic function given by a Gaussian function

$$\chi_{\text{Env}}(\mathbf{v}) \propto \mathcal{N}\left[2 \tanh\left(\frac{\Delta_q \Delta_p}{2}\right) \text{Diag}\left(\frac{\Delta_q}{\Delta_p}, \frac{\Delta_p}{\Delta_q}\right), \mathbf{v}\right] \quad (25a)$$

$$\approx \mathcal{N}(\text{Diag}(\Delta_q^2, \Delta_p^2), \mathbf{v}), \quad (25b)$$

where Eq. (25b) is obtained by taking a first-order approximation of the tanh function. Under the twirling approximation, it therefore becomes a Gaussian random displacement channel $\mathcal{G}_\Sigma$ with $\Sigma \approx \text{Diag}(\Delta_q^2, \Delta_p^2)/2$.

We remark that in physical devices, photon loss and dephasing channels are also significant sources of error. However, we do not include these noise channels in our analysis. This choice is made to clearly illustrate our core ideas using a realistic yet mathematically tractable model. Importantly, under the twirling approximation, these noise channels can also be converted into random displacement channels. For example, applying the twirling approximation to loss gives us directly the Gaussian random displacement channel [30]. One could therefore imagine extending the analysis presented in this work to account for photon loss or dephasing noise with some appropriate modifications.

III. ON-DEMAND NOISE BIASING CIRCUITS

In this section, we discuss why on-demand noise biasing is essential for achieving a high-fidelity implementation of the cubic phase T gate and outline how on-demand biasing can be realized. In Sec. III A, we provide an intuitive explanation of how the cubic phase gate amplifies noise from the envelope operator. Building on this intuition, we show how on-demand noise biasing can mitigate the resulting amplified noise. Then, in Sec. III B, we demonstrate that a fixed amount of on-demand noise biasing can be implemented using the standard q -Steane QEC circuit, and thus requires no additional resources beyond those of a conventional QEC circuit. General amounts of noise biasing can then be achieved by a simple generalization to the q -Steane QEC circuit that makes use either of a biased square GKP ancilla codestate or a nonbiased rectangular codestate that can be prepared in advance.

A. Error propagation of the cubic phase gate

To understand the existing problems with the cubic phase gate, we consider a simple circuit to generate the ideal square GKP state $|\bar{T}\rangle = T|\bar{+}\rangle$:

$$|\bar{+}\rangle \xrightarrow{T_{\text{GKP}}} \quad (26)$$

Here, T_{GKP} is the specific realization of the cubic phase gate proposed in Ref. [5] for implementing the logical T gate, defined as

$$T_{\text{GKP}} = \exp \left\{ i2\pi \left[\frac{1}{4} \left(\frac{q}{\sqrt{\pi}} \right)^3 + \frac{1}{8} \left(\frac{q}{\sqrt{\pi}} \right)^2 - \frac{1}{4} \left(\frac{q}{\sqrt{\pi}} \right) \right] \right\}. \quad (27)$$

If the input state to Eq. (26) is ideal, then the output state is an ideal square GKP $|\bar{T}\rangle$ state. To see this, note that the ideal $|\bar{0}\rangle$ (and, respectively, $|\bar{1}\rangle$) codestates are superpositions of position eigenstates with positions at even (odd) multiples of $\sqrt{\pi}$ [Eq. (7)]. Meanwhile, note that the cubic polynomial

$$P(x) = \frac{1}{4}x^3 + \frac{1}{8}x^2 - \frac{1}{4}x \quad (28)$$

has the property that if x is an even integer then $P(x)$ is an integer, and if x is an odd integer then $P(x)$ is $n+1/8$

(for some integer n). Therefore, the cubic phase gate $T_{\text{GKP}} = \exp(i2\pi P(q/\sqrt{\pi}))$ acts as

$$T_{\text{GKP}}|2n\sqrt{\pi}\rangle_q = |2n\sqrt{\pi}\rangle_q, \quad (29a)$$

$$T_{\text{GKP}}|(2n+1)\sqrt{\pi}\rangle_q = e^{i\pi/4}|(2n+1)\sqrt{\pi}\rangle_q, \quad (29b)$$

for all integers n , and implements a logical T gate.

However, as discussed in Sec. II, only approximate GKP states can be prepared in practice. Approximate codestates always introduce some logical errors due to the envelope operator, and the probability of a logical error goes to 0 as $\Delta \rightarrow 0$. However, it was shown in Ref. [16] that the T_{GKP} gate amplifies the noise due to the envelope operator in such a way that the probability of a logical error no longer goes to 0 as $\Delta \rightarrow 0$; we will also rederive this result using the twirling approximation in Sec. V.

One can understand this intuitively by considering a small perturbation to the position eigenstate $|(n+\epsilon)\sqrt{\pi}\rangle_q$ that the cubic phase gate acts upon. In particular, the polynomial $P(n+\epsilon)$ is evaluated at a position that is a small distance ϵ away from being an integer n . Roughly speaking, the error due to the T_{GKP} gate comes from the difference in phase that is applied due to the perturbation $P(n+\epsilon) - P(n)$. To estimate how this error scales with Δ , note that in the position basis, the approximate codestate is a superposition of Gaussian functions with width $\sim \Delta_q$ multiplied by an envelope of width $\sim 1/\Delta_p$. For an unbiased approximate codestate with $\Delta = \Delta_q = \Delta_p$, we therefore set $\epsilon = O(\Delta)$ and $n = O(1/\Delta)$ because larger values of ϵ and n are exponentially suppressed by the Gaussian distribution. In this case, the error in the phase that is applied due to the perturbation $P(n) - P(n+\epsilon)$ contains terms that do not decrease as $\Delta \rightarrow 0$:

$$P(n+\epsilon) - P(n) = \frac{3}{4}n^2\epsilon + O(1) = O(\Delta^{-1}). \quad (30)$$

We call this leading term the “shear error” and it is the leading source of error in the T_{GKP} gate. In particular, due to the shear error, the error in the phase of the resulting state does not decrease as Δ tends to zero—limiting the logical error rate that can be achieved [31]. Note, however, that this intuitive argument is not watertight and does not produce quantitatively correct predictions of the behavior of the T_{GKP} gate (see Ref. [32]); for this reason, we will more carefully derive the error induced by the T gate in Sec. V.

The solution to this is to instead *bias* the noise profile by using a biased envelope operator [Eq. (17)], which improves gate fidelity by two ways. First, decreasing Δ_q reduces the typical displacement in the position quadrature ϵ ; and second, increasing Δ_p tightens the envelope of the codeword position-space wavefunction, effectively reducing n . Of course, increasing Δ_p also increases the error rate because it *increases* the width of the envelope in the *momentum* quadrature, introducing a trade-off between large and small noise biases.

An alternative, entirely equivalent solution is to use nonbiased *rectangular* codestates. This is equivalent because of the unitary relationship between biased square and nonbiased rectangular codestates given by Eq. (18); note that in this case one also has to use the rectangular $T_{\text{GKP}}^{(\lambda)} = \text{Sq}(\sqrt{\lambda})^\dagger T_{\text{GKP}} \text{Sq}(\sqrt{\lambda})$ logical gate.

Regardless of whether one uses a biased square or nonbiased rectangular codestate to perform a T gate, it is crucial that the bias is prepared *on demand*. This is primarily because Clifford gates (for example, the Hadamard gate [Eq. (8a)]) are not bias preserving: in the square GKP code, performing a logical Hadamard gate causes the bias to be inverted $\Delta_q \leftrightarrow \Delta_p$. Moreover, a biased square GKP qubit is less protected while idling compared to a nonbiased square GKP qubit [see, for example, Fig. 6(a)]. Therefore, in the following section we will derive an on-demand biasing circuit that allows one to bias the codestate just before the T gate and return to a nonbiased codestate afterward.

B. On-demand biasing circuit

The on-demand biasing circuit is based on the q -Steane QEC circuit [Eq. (13a)]. We will first begin by explaining how the q -Steane QEC circuit introduces a fixed amount of noise bias, and then we will generalize the q -Steane QEC circuit to be able to introduce an arbitrary amount of bias λ .

We begin with an identity from Ref. [33], which rewrites the q -Steane measurement circuit in terms of beam splitters and single-mode squeezers [Eq. (9)]:

(31a)

$$= \begin{array}{c} |\bar{\psi}\rangle_{\Delta} \\ |\bar{\tau}\rangle_{\Delta} \end{array} \rightarrow \begin{array}{c} \text{BS}(\frac{\pi}{4}) \\ \text{BS}(\frac{\pi}{4}) \end{array} \rightarrow \begin{array}{c} \text{Sq}^{\dagger}(\sqrt{2}) \\ \text{Sq}^{\dagger}(\frac{1}{\sqrt{2}}) \end{array} \rightarrow \begin{array}{c} \text{---} \\ \text{---} \end{array} \quad (31b)$$

Note that Eq. (31a) is the same as the q -Steane circuit depicted in Eq. (13a) except for the presence of the $\exp(ip_1q_2/2)$ gate. However, this gate can be compensated for by applying a corrective displacement $\exp(-ip_1q_m/2)$ based on the outcome q_m of the measurement; this correction can be performed at the same time as the usual displacement correction that is done to return the GKP qubit to the codespace. Equation (31b) is simply a linear algebra rearrangement of the unitary operators, but it allows us to reason about the spread of noise as we now explain.

The beam splitter $\text{BS}(\theta)$ preserves the nonbiased-noise profile of both data and ancilla qubits, as shown by the commutation identity [27,28]:

$$\begin{array}{c} \text{--- Env}_\Delta \\ \text{--- Env}_\Delta \end{array} \text{--- BS}(\theta) = \text{BS}(\theta) \begin{array}{c} \text{--- Env}_\Delta \\ \text{--- Env}_\Delta \end{array} \quad (32)$$

However, the single-mode squeezing operator on the data qubit after does modify the bias according to

$$\Delta_q = \Delta \rightarrow \frac{\Delta}{\sqrt{2}}, \quad \Delta_p = \Delta \rightarrow \sqrt{2}\Delta. \quad (33)$$

Therefore, after the displacement correction, the output codestate of the q -Steane QEC circuit [implemented either with Eq. (13a) or (31b)] will be a square GKP codestate $|\bar{\psi}\rangle_{\Delta/\sqrt{2}, \sqrt{2}\Delta}$ with a bias of 2. This shows that by measuring just one of the two GKP stabilizers, we have effectively turned a nonbiased GKP codestate into a biased GKP codestate: in other words, we have performed on-demand noise biasing.

Intuitively, the reduction of Δ_q , a factor of $\sqrt{2}$, is the direct consequence of measuring and correcting the position quadrature, which reduces its error variance. Conversely, the momentum quadrature becomes noisier because the momentum displacement errors arising from the envelope operator on the ancilla propagate to the data qubit through the *CX* gate, adding to the noise in the momentum quadrature of the data qubit.

However, as Δ becomes smaller, we need increasing levels of noise bias to be able to perform the GKP gate with decreasing noise levels. Indeed, we will later see in Sec. VI that for $\Delta < 0.25$ (GKP squeezing above 12 dB) the optimal level of asymmetry is already larger than what can be achieved with the q -Steane QEC circuit alone. There are a number of ways to generate higher levels of bias using modifications of the q -Steane QEC circuit, which we now outline one at a time.

The most conceptually simple way to increase the bias is to perform multiple q -Steane QEC circuits in a row, which increases the bias of the square GKP data qubit each time. With r applications of the q -Steane QEC circuit, one can prepare a biased square GKP codestate $|\psi\rangle_{\Delta/\sqrt{r+1}, \sqrt{r+1}\Delta}$.

However, a more time-efficient alternative is to use just a single q -Steane QEC circuit with a *biased* square GKP ancilla

(34)

In this circuit, due to the reduced variance $\Delta/\sqrt{\lambda}$ of the position displacement error, the syndrome measurement becomes more sensitive to position displacement errors coming from the data qubit, reducing the noise in the position quadrature. Conversely, momentum displacement errors on the ancilla are also amplified by the factor $\sqrt{\lambda}$ and propagate back to the data qubit through the CX gate. A straightforward calculation shows that the effective squeezing parameters of the data qubit are transformed as

$$\Delta_q = \Delta \rightarrow \frac{\Delta}{\sqrt{1+\lambda}}, \quad \Delta_p = \Delta \rightarrow \sqrt{1+\lambda}\Delta. \quad (35)$$

Note that even though Eq. (34) itself already uses a biased-noise ancilla, this state can be prepared “offline” instead of on demand.

Finally, an entirely equivalent alternative to Eq. (34) is to use a nonbiased rectangular ancilla:

$$\begin{array}{c}
|\bar{\psi}\rangle_{\Delta} \\
|\bar{\chi}\rangle_{\Delta}^{\lambda}
\end{array}
\rightarrow
\begin{array}{|c|}
\hline
e^{-i\sqrt{\lambda}q_1p_2} \\
\hline
\end{array}
\rightarrow
\begin{array}{c}
\text{---} \\
\text{---} \rightarrow \boxed{\text{q}}
\end{array}, \quad (36)$$

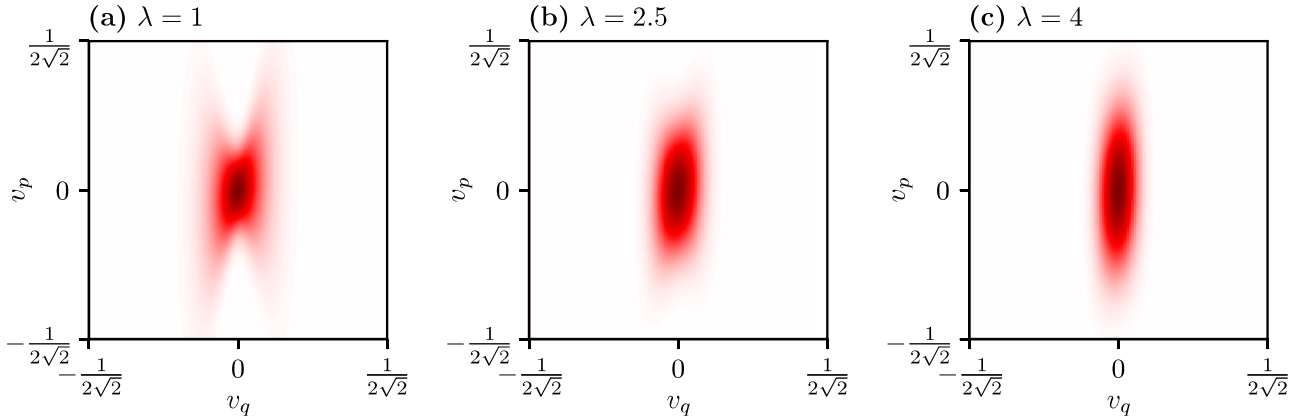


FIG. 2. A visual representation of twirled channel $\text{Twirl}(\mathcal{E})$ for the optimized T_3 gate [Eq. (41)] and for $\Delta = \sqrt{\Delta_q \Delta_p} = 0.2$. Specifically, we plot the probability distribution $p_{\mathcal{E}}(\mathbf{v})$ of the random displacement function that is obtained by twirling the error operator E as in Eq. (22). While $\mathbf{v} = (v_q, v_p)$ can take any value in $\mathbb{R}^2$, we show here only the correctable displacements. Moving from panel (a) to panel (c), we increase the noise bias $\lambda = \Delta_p / \Delta_q$, where panel (b) is the optimal noise biasing for this value of Δ , as found by numerics in Fig. 3(a). When not enough noise biasing is applied as in panel (a), the noise from the T gate dominates and increases the noise, while when too much noise biasing is applied, as in panel (c), the noise from the biasing itself dominates. At intermediate values, these two effects balance out so that the overall noise rate is minimal.

which also generates the same biasing as in Eq. (35). This can be derived from Eq. (34) by inserting squeezing operator identities $\text{Sq}(\sqrt{\lambda})\text{Sq}(\sqrt{\lambda})^\dagger$ into the circuit.

After biasing the noise with either Eq. (34) or (36) to the optimal amount of biasing for the given Δ and applying the corrective displacement, we proceed with the application of the cubic phase gate T_{GKP} . Following the gate, the noise profile becomes highly asymmetric and distorted (see Fig. 2). If left uncorrected, subsequent gates can further amplify these distortions, leading to rapid degradation of the logical state fidelity. Therefore, using the same idea as Ref. [34], we return to the unbiased square GKP code using the passive Knill QEC circuit [28], which is guaranteed to output a fresh unbiased square GKP codestate. The passive Knill QEC circuit can be used exactly as shown in Eq. (14) because the input state is a square GKP codestate (even though it is biased).

In summary, to apply the cubic phase gate, we first perform noise biasing using Eq. (34) before the gate and then perform full round of error correction afterward using the passive Knill QEC circuit:

$$\dots \rightarrow \boxed{\text{On-demand biasing}} \rightarrow \boxed{q\text{-Steane QEC}} \rightarrow \boxed{T} \rightarrow \boxed{\text{Passive Knill QEC}} \rightarrow \dots \quad (37)$$

In contrast, to apply any Clifford gate we simply apply the gate between two rounds of QEC:

$$\dots \rightarrow \boxed{\text{Passive Knill QEC}} \rightarrow \boxed{C} \rightarrow \boxed{\text{Passive Knill QEC}} \rightarrow \dots \quad (38)$$

An important practical question is how to generate the biased-noise square GKP ancilla $|\bar{\psi}\rangle_{\Delta/\sqrt{\lambda}, \Delta\sqrt{\lambda}}$ required in Eq. (34) [or the nonbiased rectangular ancilla in Eq. (36)]. Existing preparation protocols [35–39] typically yield nonbiased square GKP states. While these protocols likely could be

adapted to generate rectangular states, we do not detail such modifications here. Instead, as a proof of principle, we show in Appendix B how to use a slight modification of the q -Steane QEC circuits to convert nonbiased square GKP states $|\bar{\psi}\rangle_{\Delta}$ into nonbiased rectangular GKP states $|\bar{\psi}\rangle_{\Delta}^{\lambda}$ using only nonbalanced beam splitters and displacement operations. The nonbiased rectangular GKP state $|\bar{\psi}\rangle_{\Delta}^{\lambda}$ can be converted to the biased-noise square GKP $|\bar{\psi}\rangle_{\Delta/\sqrt{\lambda}, \Delta\sqrt{\lambda}}$ by a single-mode squeezer. The advantage of our procedure is that it is independent of the specific method used to generate the initial nonbiased square GKP qubit.

We briefly remark that the above method applies the T gate on a biased square GKP codestate. However, one can also apply the T gate on an unbiased rectangular GKP codestate. To do this, one needs an on-demand *morphing* circuit that “morphs” from the square GKP code to the rectangular GKP code while preserving a nonbiased-noise profile. We derive such an on-demand morphing circuit in Appendix A and explain how it can be used to morph to the rectangular code for the T gate and back to the square code for Clifford gates. This approach may be more attractive experimentally because it only uses a *passive* Gaussian operation (a beam splitter) instead of a CX gate, and because a nonbiased rectangular codestate has a lower average photon number than the corresponding biased square codestate, reducing its susceptibility to loss and dephasing.

One may wonder whether the on-demand biasing circuits that we propose can be improved upon to provide output states with better squeezing parameters. In particular, given an input state $|\bar{\psi}\rangle_{\Delta}$, what output state $|\bar{\psi}\rangle_{\Delta_q, \Delta_p}$ is achievable when using Gaussian operations? In Appendix C, we prove that, when limited to circuits using any number of approximate GKP ancillas $|\bar{\psi}\rangle_{\Delta}$ and any deterministic Gaussian operations, the output GKP qubit will always have a bias that satisfies

$$\Delta_q \Delta_p \geq \Delta^2. \quad (39)$$

In particular, the on-demand biasing circuits [Eqs. (34) and (36)] already saturate this bound, demonstrating that these circuits are indeed optimal among deterministic Gaussian circuits.

IV. OPTIMIZED LOGICAL GATE REPRESENTATIONS

In the previous section, we introduced the on-demand noise-biasing protocol, which allows us to generate bias that mitigates the effect of propagated errors induced by the cubic phase T gate. In this section, we move beyond the cubic phase gate and develop a general formalism for characterizing polynomial phase representations of logical gates, enabling the identification of representations that minimize error propagation when acting on approximate GKP states. Central to this framework is the introduction of *polynomial phase stabilizers* that preserve the GKP codestates but are not displacement operators. We show that these generalized stabilizers impose strong algebraic constraints on the allowed polynomial phase gates of logical gates. Finally, we emphasize that the formalism is broadly applicable: it extends naturally to other bosonic codes and can be generalized to multiqubit GKP logical gates, as discussed in Appendix E.

A. Polynomial phase stabilizers

In this section, we introduce the framework of polynomial phase stabilizers, which will lead to a systematic method to construct alternative representations of logical gates in the GKP code that minimize error propagation.

As a motivating example, consider the following operator:

$$O = \exp \left\{ i2\pi \left[-\frac{1}{6} \left(\frac{q}{\sqrt{\pi}} \right)^3 + \frac{1}{6} \left(\frac{q}{\sqrt{\pi}} \right) \right] \right\}. \quad (40)$$

How does this operator act on square GKP codestates? As before, we can write $O = \exp(i2\pi P(q/\sqrt{\pi}))$, where $P(x) = -x^3/6 + x/6$ is a cubic polynomial. Moreover, note that $P(n)$ is an integer for every input integer n . This means that not only does O preserve the GKP codespace, but it also acts as the logical identity on that codespace—it is a cubic phase stabilizer.

Because O acts trivially on the GKP codespace, it can be multiplied by the T_{GKP} gate to obtain an alternative cubic phase gate T_3 that still implements a logical T gate:

$$T_3 = T_{\text{GKP}} O \quad (41a)$$

$$= \exp \left\{ i2\pi \left[\frac{1}{12} \left(\frac{q}{\sqrt{\pi}} \right)^3 + \frac{1}{8} \left(\frac{q}{\sqrt{\pi}} \right)^2 - \frac{1}{12} \left(\frac{q}{\sqrt{\pi}} \right) \right] \right\}. \quad (41b)$$

This same representation was previously found in Ref. [17] through numerical optimization and shown to achieve significantly higher gate fidelity than the original T_{GKP} gate. This improvement can be attributed to the reduced amplitude of the cubic (non-Gaussian) term, which reduces the magnitude of the induced shear errors discussed in Sec. III A. Our approach here, however, suggests a more systematic general framework to obtain minimal representations of polynomial phase gates by characterizing the corresponding polynomial phase stabilizers, as we now explain.

Specifically, polynomial phase stabilizers are operators that act trivially on the GKP codespace and can be written as the exponential of a polynomial

$$S_i = \exp[i2\pi P(x)]. \quad (42)$$

Here, $P(x)$ is a real polynomial that generates the stabilizer S_i , and x is any operator such that the linear stabilizer $\exp(i2\pi x)$ acts trivially on the ideal GKP states [40]. Importantly, the linear stabilizer guarantees that the GKP codestates are superpositions of eigenstates of x at integer eigenvalues. For the present discussion, we fix $x = q/\sqrt{\pi}$, corresponding to a scaled position operator, such that $\exp(i2\pi x) = \exp(i2\sqrt{\pi}q) = S_q$ is one of the usual GKP stabilizers. However, we will later show that other choices for x , such as $a^\dagger a/2$, also yield valid stabilizer structures and reveal additional structure in the codespace.

Since the ideal GKP states have support only on eigenstates $|m\sqrt{\pi}\rangle_q$, where $m \in \mathbb{Z}$, it follows that the polynomial $P(x)$ must return integer values when evaluated at integer arguments x . In other words, we require

$$P(\mathbb{Z}) \subseteq \mathbb{Z}. \quad (43)$$

This class of polynomials is well known in mathematics as integer-valued polynomials. Importantly, Pólya [41] found that there is a canonical basis for the set of integer-valued polynomials, namely,

$$L_n(x) = \begin{cases} \frac{1}{n!} \prod_{i=1}^n (x + i - \frac{n}{2}), & \text{for } n \text{ even,} \\ \frac{1}{n!} \prod_{i=1}^n (x + i - \frac{n+1}{2}), & \text{for } n \text{ odd.} \end{cases} \quad (44)$$

(See also Ref. [42] for a more pedagogical introduction.) All integer-valued polynomials $P(x)$ can be written as an integer linear combination of the basis polynomials $\{L_n(x)\}$; that is,

$$P(x) = \sum_{n=1}^m c_n L_n(x), \quad (45)$$

for some integer coefficients c_n .

Since the construction of polynomial phase stabilizers relies solely on the fact that the square GKP code has support on a discrete position-space grid, the same approach can be readily extended to other sets of grid states. In particular, we can generalize the operator x by replacing it with any of the scaled operators $\{q, p, q - p\}/\sqrt{\pi}$ that correspond, respectively, to the logical Z , X , and Y bases. Furthermore, we observe that the square GKP code has support only on even-number Fock states and is therefore stabilized by $\exp(i\pi a^\dagger a)$, and so we can also set x to $a^\dagger a/2$. More generally, this observation implies that the concept of polynomial phase stabilizers can be naturally extended to the broader class of bosonic codes [3] such as the multileg rotationally symmetric code [18] or the number-phase code [19]. For instance, we can choose the operator x to be $a^\dagger a/\alpha$ in the multileg rotationally symmetric code where α is the number of legs.

B. Minimal logical gate representations

In this section, we consider the family of polynomial phase gates defined by

$$\Lambda_m[x] = \exp(i2\pi P_{\Lambda_m}(x)), \quad (46)$$

in which its actions on the eigenstates $|x\rangle$ ($x \in \mathbb{Z}$) of the operator x are given by

$$\Lambda_m[x]|x\rangle = \begin{cases} |x\rangle, & \text{for } x \equiv 0 \pmod{2}, \\ \exp(i2\pi/2^m)|x\rangle, & \text{for } x \equiv 1 \pmod{2}. \end{cases} \quad (47)$$

When the operator x is chosen to be $q/\sqrt{\pi}$, the gate family $\Lambda_m[x]$ encompasses the logical diagonal gates $\{Z, S, T, \sqrt{T}, \dots\}$ of the GKP code. More generally, by selecting alternative choices for x , the family $\Lambda_m[x]$ also captures other important gate families. For example, by setting $x = a^\dagger a/2$, we yield the Hadamard gate family $\{H, \sqrt{H}, \sqrt[4]{H}, \dots\}$ [40].

In what follows, we will detail an algorithm that “minimizes” the representation of a particular polynomial phase gate. In particular, the minimization is with respect to a “lexicographic” ordering on the polynomials (technically this is a partial order). In the lexicographic ordering, given two real polynomials $P(x) = \sum_n c_n x^n$ and $Q(x) = \sum_n d_n x^n$, we compare the coefficients of the polynomials $P(x)$ and $Q(x)$ starting from the leading coefficient to lowest-order coefficient. We define the polynomial $P(x)$ to be less than $Q(x)$ if

$$P(x) < Q(x) \Leftrightarrow \text{there exists } i \text{ such that} \\ |c_k| = |d_k| \text{ for all } k > i, \text{ and } |c_i| < |d_i|. \quad (48)$$

The advantage of the lexicographic ordering is that it minimizes the coefficients of the higher-degree terms in the polynomial before moving to the lower-degree terms. Intuitively, this is important because the higher-degree terms in the polynomial contribute to higher-degree errors that affect GKP codestates more severely. For this reason, as $\Delta \rightarrow 0$, we expect a minimal polynomial phase gate to perform optimally among the polynomial phase gates that implement the same logical operator (although see Ref. [43]).

The algorithm that we describe will output a polynomial phase gate whose corresponding polynomial is not larger under the lexicographic ordering than any other polynomial that implements the same gate. The algorithm consists of two steps: First, we write down any “starting” representation of the $\Lambda_m[x]$ gate, and then we use the polynomial phase stabilizers to minimize the coefficients of the gate under the lexicographic order (we call this second step the “coefficient reduction” procedure). We show in Appendix D that every $\Lambda_m[x]$ gate admits a “starting” representation with polynomial

$$P_{\text{start}}(x) = \frac{x^{2^{m-1}}}{2^m}. \quad (49)$$

Alternatively, if a good representation $F(x)$ of the $\Lambda_{m-1}[x]$ gate is already known, one can choose a better starting point

$$P_{\text{start}}(x) = 2^{m-2} F(x)^2. \quad (50)$$

In either case, the candidate polynomial for $P_{\text{start}}(x)$ can be expressed as

$$P_{\text{start}}(x) = \sum_{i=1}^N a_i x^i. \quad (51)$$

The goal of the coefficient reduction procedure is to systematically adjust the coefficients of $P_{\text{start}}(x)$ by adding or subtracting suitable integer multiple of the basis polynomials

$\{L_n(x)\}$, starting from the highest-degree term and proceeding downward. Importantly, the leading coefficient of $L_n(x)$ is $1/n!$, which means that we can reduce each coefficient to be between $-1/(2n!)$ and $1/(2n!)$. Specifically, let j denote a pointer, initialized at $j = N$ and decreased sequentially until $j = 1$, and initialize $P(x) = P_{\text{start}}(x)$. At each step, we decompose the coefficient a_j of the polynomial [Eq. (51)] as

$$a_j = \frac{n_j}{j!} + r_j, \quad (52)$$

where $n_j \in \mathbb{Z}$ and the remainder satisfies $|r_j| \leq \frac{1}{2j!}$. We then update the polynomial $P(x)$ via

$$P(x) \leftarrow P(x) - n_j L_j(x), \quad (53)$$

where $L_j(x)$ is the degree- j integer-valued polynomial defined in Eq. (44). This step removes the part of each coefficient that is an integer multiple of $1/j!$. Note that n_j and r_j are unique here *unless* $r_j = \pm \frac{1}{2j!}$, in which case there are two choices of n_j . In this case, we proceed with both choices of n_j until one of the choices is lexicographically smaller than the other, at which point we can discard the other choice of n_j . Iterating this procedure from $j = N$ down to $j = 1$ produces a reduced polynomial representation of the form

$$P(x) = \sum_{k=0}^m a_k x^k, \quad (54)$$

with coefficients satisfying bound

$$|a_k| \leq \frac{1}{2k!}, \quad \forall k. \quad (55)$$

In Appendix D, we present multiple results about the coefficient reduction procedure. In Lemma D2, we formally prove that the above algorithm indeed returns a minimal representation under the lexicographic ordering. Moreover, we prove in Theorem D1 that the degree of the polynomial that is returned is exactly m , and therefore that the lowest degree polynomial that can implement the $\Lambda_m[x]$ gate is degree m . While this result is well known for diagonal GKP gates specifically, our theorem applies to all polynomial phase gates of the form described here (including the Hadamard gate hierarchy).

As an example, let us construct the minimal representation of the $\sqrt{T}$ gate. Our starting point is obtained by substituting the minimal T gate into Eq. (50), giving

$$P_{\Lambda_4}(x) \leftarrow P_{\text{start}}(x) = 4P_{\Lambda_3}(x)^2 \quad (56a)$$

$$= \frac{x^6}{36} + \frac{x^5}{12} + \frac{x^4}{144} - \frac{x^3}{12} + \frac{x^2}{36}. \quad (56b)$$

Now, the leading coefficient $1/36$ is equal to 20 times $1/6!$, and therefore we update

$$P_{\Lambda_4}(x) \leftarrow P_{\Lambda_4}(x) - 20L_6(x) \quad (57a)$$

$$= \frac{7x^4}{48} + \frac{x^3}{3} - \frac{x^2}{12} - \frac{x}{3}. \quad (57b)$$

Notice that this update has coincidentally reduced the fifth-order coefficient to zero. Moving on to the fourth-order coefficient, there are actually two choices for the coefficient reduction of $7x^4/48$ since it can be written in two ways:

$7/48 = 3 \times 1/4! + 1/48 = 4 \times 1/4! - 1/48$. After running the full algorithm, it turns out that the second choice returns a smaller polynomial under the lexicographic ordering, so we update

$$P_{\Lambda_4}(x) \leftarrow P_{\Lambda_4}(x) - 4L_4(x) \quad (58a)$$

$$= -\frac{x^4}{48} + \frac{x^2}{12}. \quad (58b)$$

This polynomial can no longer be reduced by any of $L_3(x)$, $L_2(x)$, or $L_1(x)$, and therefore we are done.

Equation (58) gives rise not only to a minimal representation of the $\sqrt{T}$ gate,

$$\sqrt{T} = \exp \left\{ i2\pi \left[-\frac{1}{48} \left(\frac{q}{\sqrt{\pi}} \right)^4 + \frac{1}{12} \left(\frac{q}{\sqrt{\pi}} \right)^2 \right] \right\}, \quad (59)$$

but also to a minimal representation of the $\sqrt[8]{H}$ gate,

$$\sqrt[8]{H} = \exp \left\{ i2\pi \left[-\frac{1}{48} \left(\frac{a^\dagger a}{2} \right)^4 + \frac{1}{12} \left(\frac{a^\dagger a}{2} \right)^2 \right] \right\}. \quad (60)$$

It is important to note, however, that fault tolerance is only guaranteed for polynomial phase gates of a quadrature operator such as Eq. (59) and not for Eq. (60) since it is unclear how one would bias the noise in the latter case.

Moreover, taking the square of Eq. (59) gives a quartic representation of the T gate that we label T_4 :

$$T_4 = \exp \left\{ i2\pi \left[-\frac{1}{24} \left(\frac{q}{\sqrt{\pi}} \right)^4 + \frac{1}{6} \left(\frac{q}{\sqrt{\pi}} \right)^2 \right] \right\}. \quad (61)$$

Although Eq. (61) is not *minimal*, it has the advantage over the minimal T_3 gate [Eq. (41)] that the polynomial that implements it is even, making it more convenient in some implementations. Indeed, in Ref. [44] it was shown that logical phase gates of the GKP code can be implemented protectively in $0-\pi$ qubits, provided the underlying Hamiltonians involve only quadratic and quartic potentials. This example demonstrates that we can use the polynomial phase stabilizers to find logical gates tailored to hardware constraints, while preserving logical equivalence.

Although we have so far focused on single-qubit gates in this section, the same polynomial-stabilizer framework naturally extends to multiqubit gates between square GKP qubits. In Appendix E, we demonstrate how the coefficient reduction procedure can be generalized to construct minimal representations of such multiqubit gates. For example, we find the following minimal representations of the CS and CCZ gates for the square GKP qubits:

$$CS = \exp \left\{ -\frac{i\pi}{2} \left[\left(\frac{q_1}{\sqrt{\pi}} \right)^2 \left(\frac{q_2}{\sqrt{\pi}} \right) + \left(\frac{q_1}{\sqrt{\pi}} \right) \left(\frac{q_2}{\sqrt{\pi}} \right)^2 + \left(\frac{q_1}{\sqrt{\pi}} \right) \left(\frac{q_2}{\sqrt{\pi}} \right) \right] \right\}, \quad (62a)$$

$$CCZ = \exp(iq_1 q_2 q_3 / \sqrt{\pi}), \quad (62b)$$

acting on two and three modes, respectively.

V. ASYMPTOTIC ANALYSIS

In the previous sections, we showed how to construct minimal representations of logical gates, and we showed how to perform on-demand noise biasing. We now turn to analyzing the error rates of these gates when acting on approximate biased square GKP states $|\bar{\psi}\rangle_{\Delta_q, \Delta_p}$. Throughout this analysis, we will assume that $x = q/\sqrt{\pi}$. In this section, we present our analytic results concerning the behavior as $\Delta_q, \Delta_p \rightarrow 0$, with our numerical results explained later in Sec. VI. In particular, we take two separate analytical approaches to analyze the fault tolerance of polynomial phase gates. The first is to use the twirling approximation to obtain an asymptotic estimate of the error scaling in the limit of small Δ . This estimate, however, uses a number of approximations as opposed to strict lower bounds, and as such we also rigorously prove a fault-tolerance theorem for the minimal T_3 gate [Eq. (41)].

We begin, however, by explaining the error that we wish to characterize analytically. Given a polynomial phase gate Λ , we define the error operator as

$$E = \Lambda \text{Env}_{\Delta_q, \Delta_p} \Lambda^\dagger. \quad (63)$$

To understand E , consider applying each of its terms to an *ideal* GKP codestate $|\bar{\psi}\rangle$. From right to left, the first term $\Lambda^\dagger$ acts ideally because it is acting on an ideal GKP codestate. Then, the envelope operator prepares an approximate GKP codestate with logical information given by $\Lambda^\dagger|\bar{\psi}\rangle$. Finally, the last term Λ acts nonideally because it is applied to an approximate GKP codestate. Importantly, because we acted initially with $\Lambda^\dagger$, the ideal *logical* action of E is the identity, which can be recovered by setting $\Delta_q = \Delta_p = 0$. Therefore, E captures the effect of spreading the envelope operator through the gate Λ while avoiding the logical action of Λ .

In our asymptotic analysis, we will consider the channel $\mathcal{E}(\cdot) = E \cdot E^\dagger$ (see Ref. [45]) and calculate the characteristic function $p_{\mathcal{E}}(\mathbf{v})$ of the twirled channel $\text{Twirl}(\mathcal{E})$. Recall that this twirling approximation approximates E as a random displacement channel, where the displacements are sampled from the (non-Gaussian) probability distribution $p_{\mathcal{E}}(\mathbf{v})$. This makes it easy to analyze in the context of GKP codes: If the displacement sampled is correctable as defined by Eq. (12), then no logical error occurs. Therefore, to obtain an asymptotic estimate of the error rate, we calculate the mean and covariance matrix of the probability distribution $p_{\mathcal{E}}(\mathbf{v})$, and how these parameters scale with Δ_q and Δ_p . In particular, if the mean and covariance matrix are small, there is a large chance that the sampled displacement will satisfy Eq. (12) and be correctable.

Specifically, let the logical operator have the form

$$\Lambda = \exp[i2\pi P(q/\sqrt{\pi})], \quad P(x) = \sum_{j=1}^n a_j x^j, \quad (64)$$

where $n > 1$ is the degree of the polynomial $P(x)$. Then, using the characteristic function expansion of the envelope operator $\chi_{\text{Env}}(\mathbf{v})$ from Eq. (25), one can explicitly write the

error operator as

$$E = \Lambda \text{Env}_{\Delta_q, \Delta_p} \Lambda^\dagger \quad (65a)$$

$$= \int d\mathbf{v} \chi_{\text{Env}}(\mathbf{v}) e^{i2\pi P(q/\sqrt{\pi})} W(\mathbf{v}) e^{-i2\pi P(q/\sqrt{\pi})} \quad (65b)$$

$$= \int d\mathbf{v} \chi_{\text{Env}}(\mathbf{v}) W(\mathbf{v}) e^{i2\pi [P(q/\sqrt{\pi} + \sqrt{2}v_q) - P(q/\sqrt{\pi})]} \quad (65c)$$

$$\approx \int d\mathbf{v} \chi_{\text{Env}}(\mathbf{v}) W(\mathbf{v}) e^{i2\sqrt{2}\pi v_q P'(q/\sqrt{\pi})}, \quad (65d)$$

where the first-order derivative approximation in Eq. (65d) is valid for small v_q because $\chi_{\text{Env}}(\mathbf{v})$ decays exponentially in v_q and is made to simplify the calculations.

Next, we apply the twirling approximation and calculate the characteristic function of the twirled channel $\text{Twirl}(\mathcal{E})$; see Fig. 2 for a visual representation of the twirled channel. In Appendix F, we exactly calculate the mean and covariance matrix of the resulting distribution $p_{\mathcal{E}}(\mathbf{v})$ as a function of Δ_q and Δ_p . In particular, the mean $\boldsymbol{\mu}$ of the distribution is zero:

$$\boldsymbol{\mu} = \begin{bmatrix} \mathbb{E}(v_q) \\ \mathbb{E}(v_p) \end{bmatrix} = \mathbf{0}, \quad (66)$$

where $\mathbb{E}()$ denotes the expectation value of a variable. Moreover, the terms in the covariance matrix are given by

$$\Sigma = \begin{bmatrix} \mathbb{E}(v_q^2) & \mathbb{E}(v_q v_p) \\ \mathbb{E}(v_q v_p) & \mathbb{E}(v_p^2) \end{bmatrix}, \quad (67a)$$

$$\mathbb{E}(v_q^2) = O(\Delta_q^2), \quad (67b)$$

$$\mathbb{E}(v_p^2) = O(\Delta_p^2) + O(\Delta_q^2/\Delta_p^{2n-4}), \quad (67c)$$

$$\mathbb{E}(v_q v_p) = O(\Delta_q^2/\Delta_p^{n-2}), \quad (67d)$$

where the $O()$ notation is valid in the limit of $\Delta_q, \Delta_p \rightarrow 0$. For comparison, the bare envelope operator has $\mathbb{E}(v_q^2) = O(\Delta_q^2)$, $\mathbb{E}(v_p^2) = O(\Delta_p^2)$, and $\mathbb{E}(v_q v_p) = 0$. Therefore, the logical gate Λ has the effect of adding the additional n -dependent terms in Eq. (67). In the case of no biasing $\Delta_q = \Delta_p = \Delta$, the variance $\mathbb{E}(v_p^2)$ does not go to 0 in the limit of small Δ for $n \geq 3$, necessitating the use of noise biasing to achieve arbitrarily low error rates.

At this point, we now wish to choose a noise biasing $\lambda = \Delta_p/\Delta_q$ that minimizes the components of the covariance matrix Σ . For our purposes, we will simply choose λ to minimize the most problematic component of the covariance matrix, $\mathbb{E}(v_p^2)$. Indeed, choosing the correct λ is important because Eq. (67c) contains both a $O(\Delta_p^2)$ term from the envelope that will increase with increased biasing and a $O(\Delta_q^2/\Delta_p^{2n-4})$ term from the Λ gate that will decrease with increased biasing. We illustrate this trade-off visually in Fig. 2, which shows how the probability distribution $p_{\mathcal{E}}(\mathbf{v})$ varies with λ . Letting $\Delta = \sqrt{\Delta_q \Delta_p}$, it is straightforward to show that the optimal λ scales as

$$\lambda_{\text{opt}} = O(\Delta^{4/n-2}), \quad (68)$$

which scales *inversely* with Δ for $n \geq 3$. With this choice, the remaining terms in the covariance matrix become

$$\mathbb{E}(v_q^2) = O(\Delta^{4-4/n}), \quad (69a)$$

$$\mathbb{E}(v_p^2) = O(\Delta^{4/n}), \quad (69b)$$

$$\mathbb{E}(v_q v_p) = O(\Delta^2). \quad (69c)$$

Now, with this optimized choice λ_{opt} , we see that all the components of the covariance matrix go to zero in the limit of small Δ , as desired (see also Ref. [46]). Note that the *quantitative* estimate of the optimal biasing λ_{opt} is not the same as the numerical estimates we obtain later in Sec. VI, but should reflect the correct scaling with Δ .

Equation (69) also justifies our choice of the lexicographic ordering in Sec. IV. In particular, the performance of the polynomial phase gate Λ in the limit of small Δ is governed by the degree n of the polynomial. Moreover, in Appendix F we show that the elements of the covariance matrix also depend on the leading coefficient $|a_n|$ of the polynomial. To give a concrete example, the variance $\mathbb{E}(v_p^2)$ is 9 times larger for the T_{GKP} logical gate [Eq. (28)] than for the optimized T_3 gate [Eq. (41)]. Therefore, to improve the asymptotic performance, the most important thing to reduce is the degree of the polynomial n and the magnitude of the leading coefficient $|a_n|$. We therefore expect that for two polynomials $P(x)$ and $Q(x)$ that implement the same logical gate, if one polynomial $P(x)$ is less than $Q(x)$ under the lexicographic ordering, then for sufficiently small Δ , the polynomial phase gate of $P(x)$ will outperform that of $Q(x)$. Reducing the magnitude of the nonleading coefficients $\{a_j\}_{j=1}^{n-1}$ of $P(x)$ can have an effect for fixed larger values of Δ , but this effect will become small for sufficiently small Δ .

While Eq. (69) provides strong evidence of the fault tolerance of polynomial phase gates with an appropriate bias, it falls short of a rigorous proof. Not only did we use the twirling approximation to perform the calculations, but we also only calculated the covariance matrix of the corresponding random displacement channel rather than explicitly calculating the error rate of the gate that also depends on higher statistical moments.

Therefore, in Appendix G we also rigorously prove that for the optimized T_3 gate [Eq. (41)], the “logical” average gate fidelity of the error operator (precisely defined in Appendix G) goes to 1 as $\Delta \rightarrow 0$. This is mathematically equivalent to saying that, for any target fidelity F_{targ} , there exists a Δ_{targ} such that for any $\Delta < \Delta_{\text{targ}}$, there exists a choice of noise biasing $\lambda_{\text{opt}}(\Delta)$ such that the fidelity F of the error operator E is greater than F_{targ} . We expect that similar results could also be proven for higher-order polynomial phase gates, but due to the somewhat tedious nature of the proof we leave this to future work.

VI. NUMERICAL ANALYSIS

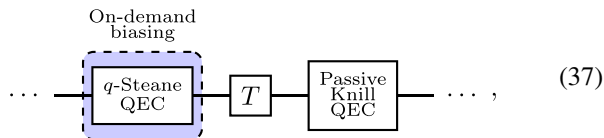
To complement our asymptotic analysis of the performance of the logical gates in Sec. V, here we numerically assess the performance of the logical gates in a more realistic parameter regime. The aim of this section is to provide a proof of principle that implementing non-Clifford gates unitarily in the GKP code can have good performance even for realistic values of Δ . To provide such a proof of principle, we will use the assumption that the only source of noise is the approximate GKP codestates themselves; we do not attempt to consider all experimentally relevant sources of noise in our simulations and leave such investigations to future work.

Throughout this section, we express the quality of the approximate codestates $\Delta = \sqrt{\Delta_q \Delta_p}$ in two different ways: first in terms of the average photon number $\bar{n} \approx 1/(2\Delta^2) - 1/2$ [47] and second in terms of the GKP squeezing $\Delta_{dB} = -10 \log_{10}(\Delta^2)$; note that both $\bar{n}$ and Δ_{dB} increase as Δ decreases. Also note that the average photon number $\bar{n}$ that we use corresponds to a *nonbiased* square (or rectangular) GKP state; a biased square codestate would have a higher average photon number than listed here. For comparison, recent experiments in superconducting qubits used nonbiased square codestates with $\Delta \approx 0.32$ ($\bar{n} \approx 4.4$, $\Delta_{dB} \approx 9.9$) [11], while experiments in optics [15] only achieved $\Delta \approx 0.93$ ($\bar{n} \approx 0.074$, $\Delta_{dB} \approx 0.6$). The achieved squeezing in superconducting devices of $\Delta = 0.32$ is exactly the same as the theoretically predicted threshold of the surface-GKP code when only considering noise from approximate GKP codestates [9]; however, the threshold is expected to be lower in the presence of additional experimentally relevant sources of noise.

In Sec. VIA, we benchmark a variety of polynomial phase gates including those that implement logical gates up to the sixth level of the Clifford hierarchy. Taking $\Delta = 0.25$ ($\bar{n} = 7.5$, $\Delta_{dB} = 12$) as an example data point, all of the optimized polynomial phase gates achieve infidelities of less than 1% with moderate asymmetries of $\lambda \leq 2$. Moreover, in Sec. VIB we find that polynomial phase gates can outperform the vacuum-state method [20] for preparing $|T\rangle$ magic states, requiring a low postselection rate of approximately 20% for the vacuum-state method to match the performance of the minimal cubic phase gate at the same $\Delta = 0.25$ data point. This comparison also underestimates the noise in the vacuum-state method: it ignores the noise that would arise from Clifford gates in the $|T\rangle$ state injection circuit itself, which could be avoided in the polynomial phase gate method by directly applying a T gate on the desired GKP qubit.

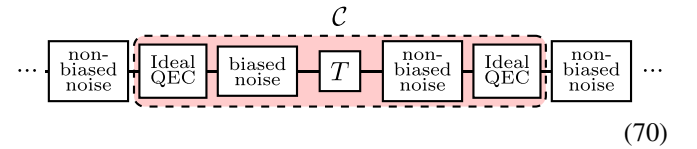
A. On-demand polynomial phase gates

We are interested in benchmarking the performance of polynomial phase gates in two contexts, the first of which is “on demand” at an arbitrary location in an algorithm. Specifically, we want to model polynomial phase gates (for simplicity here represented by a T gate) in the square GKP code that are preceded by a round of on-demand biasing using the q -Steane QEC circuit



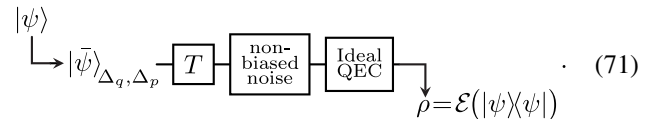
as explained in Sec. III. We assume that noise arises in this circuit only due to the envelope operator, which appears in Eq. (37) in the q -Steane and passive Knill QEC circuits [Eqs. (13a) and (14)] but not in the polynomial phase gate itself. This noise model therefore captures how the noise from the envelope operator is propagated through logical gates, but does not explicitly include some other experimentally relevant noise sources such as loss, dephasing, and imperfect gate implementation.

Intuitively, when performing a round of QEC with approximate GKP ancilla codestates, the approximate GKP codestates introduce noise in two ways. First, it causes noise on the stabilizer measurement outcomes (“measurement” noise), and second, it adds envelope noise to the output state (“incoming” noise). In fact, this intuition can be justified theoretically: If one applies the twirling approximation to the envelope operator to obtain a Gaussian random displacement channel, then the noise from approximate codestates in the passive Knill circuit [Eq. (14)] exactly corresponds to an ideal QEC cycle preceded *and* followed by two independent Gaussian random displacement channels representing the measurement error and envelope noise discussed above [30,48]. A similar result holds for the q -Steane QEC circuit but with biased noise following the gate. Our noise model is therefore analogous to the “phenomenological” noise model commonly used in the DV QEC literature. With this, Eq. (37) becomes



See Appendix A for a discussion of how this could be modified if one wished to simulate the T gate using on-demand morphing instead of on-demand biasing.

Now, to capture the performance of the polynomial phase gate we will isolate only the part of the circuit that is “in between” the two ideal QEC circuits, labeled $\mathcal{C}$ in Eq. (70). This is because the noise outside of this part of the circuit can be considered as the noise due to the other logical gates that come before and after the T gate. Next, we define the *effective logical channel* $\mathcal{E}$ that maps a *qubit* input state to a *qubit* output state and captures the performance of the circuit $\mathcal{C}$:



In words, the effective logical channel $\mathcal{E}$ has the following steps:

- (1) First, we take as input an arbitrary qubit state $|\psi\rangle$ and encode it into the ideal square GKP code $|\tilde{\psi}\rangle$ [this can be viewed as the output of the initial ideal QEC round in Eq. (70)];
- (2) we apply the biased envelope operator that prepares the state $|\tilde{\psi}\rangle_{\Delta_q, \Delta_p}$ (the “incoming” noise);
- (3) we apply the logical gate T ;
- (4) we apply a nonbiased Gaussian random displacement channel to model the measurement noise in the passive Knill QEC circuit; and finally,
- (5) we apply a round of ideal QEC, which returns us to the ideal GKP codespace, and read out the resulting logical qubit density matrix ρ .

We can then use the average gate fidelity to quantify the quality of the logical operation, given by [49]

$$F(\mathcal{E}, U) = \int d\psi \langle \psi | U^\dagger \mathcal{E}(|\psi\rangle\langle\psi|) U | \psi \rangle. \quad (72)$$

TABLE I. Polynomials used to construct the polynomial phase gates. In particular, each CV gate is implemented using the unitary $\exp(2\pi i P(q/\sqrt{\pi}))$. Note that all of the gates labeled T implement a logical T gate using a different polynomial; each of these are equivalent up to polynomial stabilizers.

CV gate	Polynomial $P(x)$
I	0
T_3	$x^3/12 + x^2/8 - x/12$
T_{GKP}	$x^3/4 + x^2/8 - x/4$
T_4	$-x^4/24 + x^2/6$
$\sqrt{T}$	$-x^4/48 + x^2/12$
$T^{1/4}$	$x^5/240 - x^4/96 - x^3/48 + x^2/24 + x/60$
$T^{1/8}$	$x^6/1440 - 5x^4/576 + 17x^2/720$

We will also refer to $1 - F(\mathcal{E}, U)$ as the average gate infidelity. Some subtleties (such as the orthonormalization of the GKP codewords) have been omitted in the above description for the sake of brevity (see Appendix H for more details as well as a description of our numerical methods, which are a truncated Fock space simulation of the channel $\mathcal{E}$).

The gates and polynomials that we simulated are listed in Table I. For each CV gate, we calculated the average gate fidelity of the corresponding effective logical channel $\mathcal{E}$ as a function of the asymmetry λ and the strength Δ of the envelope operator. We then took the optimal asymmetry λ_{opt} for each value of Δ and plotted the average gate fidelity with the optimized asymmetry.

The results are shown in Figs. 3 and 4. The plots labeled with I represent idling, where the only noise comes from the incoming envelope operator and the noisy syndrome extraction, and serves as a comparison for the other nontrivial logical gates. In Fig. 3, we plot three different implementations of the T gate: T_{GKP} from the original GKP proposal [5], T_3 that optimizes the cubic polynomial required to implement a T gate, and T_4 whose polynomial is even but quartic. Meanwhile, in Fig. 4 we plot optimized implementations of the Λ_m gates for $m = 3, 4, 5, 6$, corresponding to logical Bloch sphere rotations around the Z axis by angles of $\pi/4, \pi/8, \pi/16$, and $\pi/32$, respectively. For all of the curves, achieving the precise optimal asymmetry is not necessary to achieve nearly optimal gate fidelities, as we show in Figs. 6 and 7.

We can see that T_3 outperforms both the original T_{GKP} and the T_4 gate across all values of Δ , as expected. Surprisingly, however, when compared with phase gates in higher levels of the Clifford hierarchy, T_3 does not have the best performance for lower values of Δ , with the $\sqrt{T}$ and $T^{1/4}$ gates outperforming T_3 for $\Delta > 0.2$ ($\bar{n} < 12$, $\Delta_{\text{dB}} < 14$) and $T^{1/8}$ outperforming T_3 for $\Delta > 0.16$ ($\bar{n} < 19$, $\Delta_{\text{dB}} < 16$).

This performance can be explained by two factors: First, because the coefficients in the polynomials are smaller for these higher-degree polynomials, at lower $\bar{n}$ (higher Δ) these higher-degree polynomial phase gates may spread errors less than the T_3 gate. Despite this, it is worth emphasizing that the T_3 gate is guaranteed to perform best asymptotically because it only contains up to cubic terms, while the remaining gates have higher-degree polynomials, as we showed in Sec. V.

Second, however, it is important to note that the gates representing smaller angle rotations in the Bloch sphere also tend

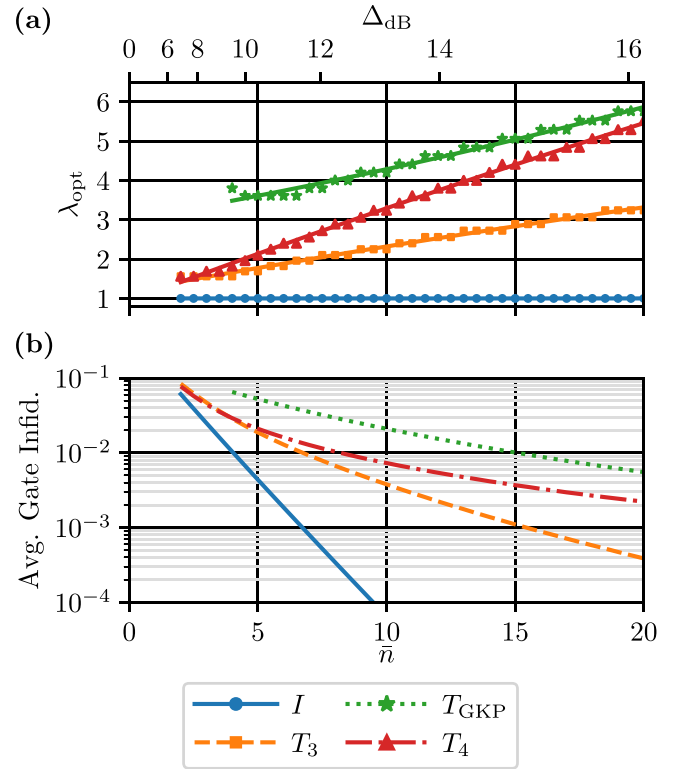


FIG. 3. Performance of T gates generated by different polynomials. (a) The optimal asymmetry for each gate. We sampled asymmetry values in finite increments, resulting in the optimal asymmetry data points that are fitted with a polynomial fit. Note that for the T_{GKP} gate, for $\bar{n} < 4$, no optimal value was found within the sampled range of λ (up to 6.5). (b) The average gate infidelity of each gate when using states prepared with the optimal asymmetry.

to inherently perform better under the average gate fidelity metric. As a back-of-the-envelope calculation, consider the average gate infidelity between the identity gate and a θ Z -axis rotation in the Bloch sphere, which is given explicitly by

$$1 - F(I, e^{i\theta Z/2}) = (1 - \cos \theta)/3. \quad (73)$$

For the T gate, $\theta = \pi/4$ and $1 - F(I, e^{i\theta Z/2}) \approx 10\%$, indicating that if the infidelity of the *logical* T gate is greater than 10%, it would be better to simply not perform the T gate at all. In Figs. 3 and 4, we see that the T gates have lower than 10% infidelity for all the sampled values of Δ and therefore are achieving nontrivial performance. However, for the $T^{1/8}$ gate we have $\theta = \pi/32$ and $1 - F(I, e^{i\theta Z/2}) \approx 1.6 \times 10^{-3}$, which means that only at $\Delta < 0.24$ ($\bar{n} > 8$, $\Delta_{\text{dB}} > 12.4$) is the $T^{1/8}$ gate achieving nontrivial performance (see Appendix H for more analysis of this). Nevertheless, we expect that directly implementing a $T^{1/8}$ would outperform a compilation of the $T^{1/8}$ gate into logical T and H gates in this regime.

B. Polynomial phase gates for magic-state preparation

The second context that we are interested in is using the polynomial phase gates as a method for preparing magic states. This is useful in part because it allows us to directly compare to existing work to prepare encoded $|T\rangle = T|+\rangle$ states [20]. This latter method—which we refer to as the

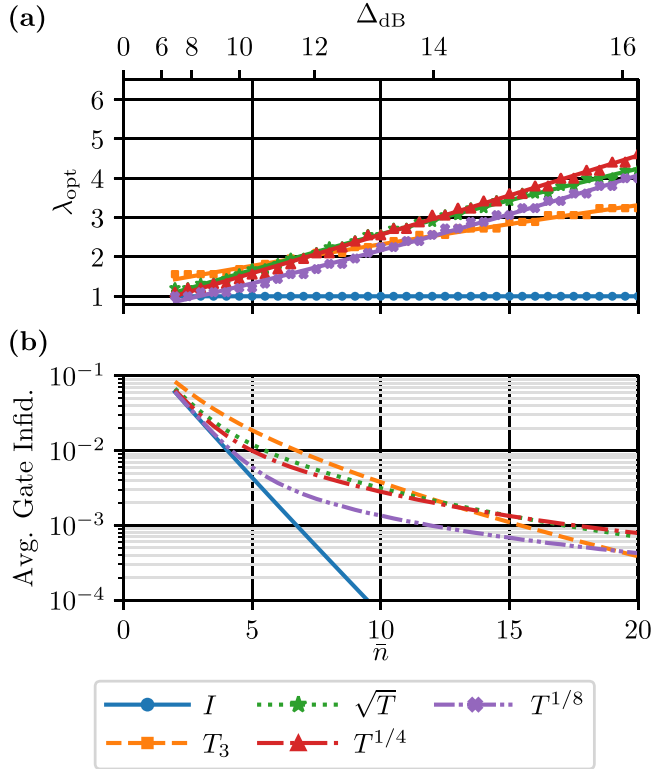


FIG. 4. Performance of Λ_m gates. Panels (a) and (b) represent the same quantities as in Fig. 3.

vacuum-state method—consists of preparing a vacuum state $|\text{vac}\rangle$ and performing a single round of GKP QEC. Remarkably, this procedure has a high chance of projecting into a logical GKP codestate that is close to a Hadamard eigenstate or a magic state that is Clifford equivalent to a Hadamard eigenstate. Therefore, after a Clifford correction dependent on the stabilizer outcome, one can prepare a noisy encoded logical $|T\rangle$ state (which is itself Clifford equivalent to a Hadamard eigenstate).

We use the state fidelity of the produced state to quantify the quality of the states prepared by each method. For the polynomial phase gate method, we consider the three polynomial phase gates from Table I that implement a T gate, using the effective logical channel $\mathcal{E}$ from Sec. VIA applied to the $|+\rangle$ state and taking the state fidelity between the output state and $|T\rangle$. Meanwhile, for the vacuum-state method we follow the numerical analysis of Ref. [20] modified to add measurement noise to the round of QEC. We assume that the Clifford correction is noiseless since it could be considered a part of the T -injection gadget. We also allow for postselection based on the stabilizer measurement outcomes in the vacuum-state method.

In Fig. 5(a), we show the state infidelity achieved using each of the three polynomial phase gates and using the vacuum-state method. For the vacuum-state method, we plot three different postselection fractions: no postselection ($p=100\%$), postselecting on the best 25% of stabilizer outcomes, and postselecting on obtaining +1 stabilizer measurement outcomes for both measurements. The latter curve serves as a tight lower bound on the state infidelity that can be

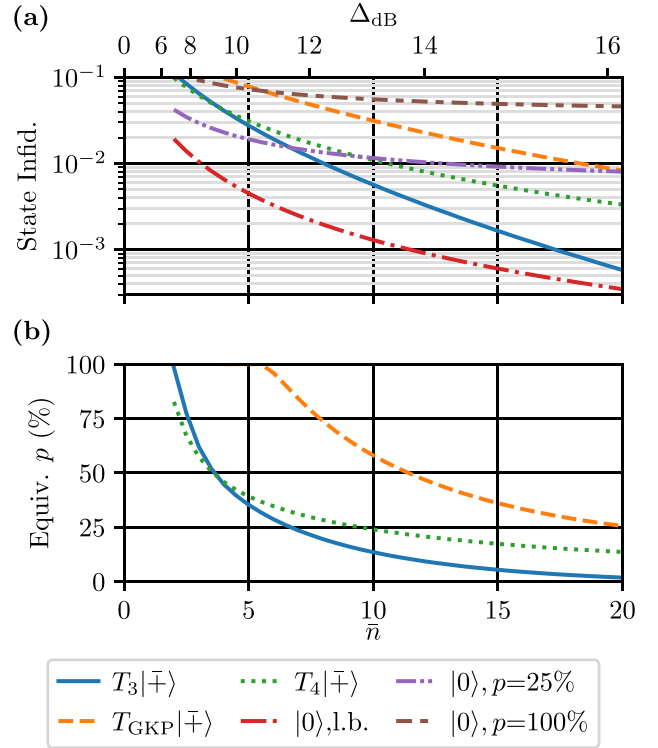


FIG. 5. Performance of T gates in preparing magic states. (a) The state infidelity of $|T\rangle$ states prepared by applying a logical phase gate to an encoded $|\bar{+}\rangle$ state, compared to those obtained via vacuum-state injection using various postselection probabilities p [20]. “l.b.” here stands for “lower bound” and corresponds to projecting the vacuum state $|\text{vac}\rangle$ onto the ideal GKP codespace, which occurs with probability zero, and therefore lower bounds the infidelity of any postselection scheme obtained by vacuum-state injection. (b) The postselection probability p that is required for the vacuum-state injection scheme to match each unitary T gate scheme.

achieved by the vacuum-state method, since it corresponds to the limit as the postselection probability goes to 0. Meanwhile, in Fig. 5(b), we take the state infidelity achieved by each of the polynomial phase gates and calculate the postselection fraction required in the vacuum-state method to achieve the same infidelity. From both these plots, we see that the T_3 method performs well compared to the vacuum-state method, requiring a postselection fraction of less than 20% for all values of $\Delta < 0.25$ ($\bar{n} > 7.5$, $\Delta_{\text{dB}} > 12$). It is also possible that at higher $\bar{n}$ the T_3 method achieves a lower infidelity than even the vacuum-state method lower bound, although this is not clear from the collected data.

It is worth noting that the comparison shown in Fig. 5, if anything, underestimates the performance of the polynomial phase gate method when compared to the vacuum-state method. This is because the polynomial phase gate can be implemented directly whenever a T gate appears in a circuit, rather than needing to be used in a magic-state injection step. With this in mind, the vacuum-state method would also suffer from noise occurring in the injection step, for example, from the CX gate and the conditional Clifford correction. On the other hand, if one really wants to use a magic-state injection step even with access to a polynomial phase gate, we would gain the ability to postselect on stabilizer

measurement outcomes, which would improve the fidelity further. It is therefore reasonable to expect that with these factors taken into account, the polynomial phase gate method outperforms the vacuum-state method by a larger degree than shown in Fig. 5.

VII. DISCUSSION AND OUTLOOK

In this article, we have presented a scheme for implementing fault-tolerant non-Clifford gates in the GKP code using polynomial phase gates. The key insight is that on-demand noise biasing can be achieved within the standard Steane QEC circuit by measuring only one of the two stabilizers (Sec. III B). The cost of performing on-demand biasing is, therefore, the same as the cost of performing a round of QEC. The scheme relies on noise-biased ancilla states, which we show can be prepared directly from nonbiased approximate square GKP states. Furthermore, we prove that the cubic phase gate T_3 , when implemented with the on-demand biasing circuit, is fault tolerant (Sec. V).

Beyond the cubic phase gate, we develop an analytical framework for studying and optimizing general polynomial phase gates, including those at higher levels of the Clifford hierarchy (Sec. IV). Central to this framework is the concept of polynomial phase stabilizers, which enable the identification of minimal polynomial phase gate representations through a coefficient reduction procedure. Using this approach, we identify T_3 as the minimal representation of the logical T gate, and we also find minimal representations for more exotic gates such as the $T^{1/2}$ or $H^{1/2}$ gates.

We numerically test our results using Fock-state simulations. We find the T_3 gate significantly outperforms the originally proposed T_{GKP} gate [5] and compares favorably to the vacuum-state method [20] even with low postselection rates. We also investigate minimal polynomial phase representations for logical gates such as the $T^{1/2}$, $T^{1/4}$, and $T^{1/8}$ gates. Surprisingly, these optimized polynomial phase gates can achieve logical gate fidelity comparable to or even exceeding that of the cubic phase T_3 gate in some regimes.

There still remain multiple challenges to exactly implement our fault-tolerant unitary T gate in experimental platforms. In photonics, implementing the polynomial phase gates remains a challenge, although we note that our scheme has multiple attractive features to photonic platforms because our unitary T gate does not require any DV qubits and because the on-demand morphing (Appendix A) can be implemented using passive Gaussian operations and GKP input states. In superconducting platforms, the polynomial phase gate is more realistic, but these platforms are still limited by DV qubit errors used in error correction. As a final remark, it is of course possible to use autonomous error-correction techniques that have already been demonstrated in superconducting platforms [10] for most of the error correction, and only use active error correction to implement the on-demand biasing for the non-Clifford gates.

There are several interesting questions that we have not addressed in this article. A natural next step is to incorporate more realistic noise models into our numerical analysis, including experimentally relevant sources such as photon loss, dephasing, and gate imperfections. We note, however, that

photon loss can be converted into a Gaussian random displacement channel (and, more generally, any noise channel can be converted into a not-necessarily-Gaussian random displacement channel) via the CV twirling approximation, and that performing this approximation only has a small effect on the logical gate fidelity [30]. We therefore expect that many of our results that we obtained for Gaussian random displacement channels would still hold in a more general setting. It would also be interesting to compare polynomial phase gates and vacuum-state methods for implementing T gates in the context of a full T gate injection. Another interesting direction is to numerically investigate the performance of multiqubit gates such as the CS and CCZ gates. While we present their minimal polynomial representations in Appendix E, and we expect on-demand biasing would improve multiqubit gate fidelities like it does in the single-qubit T gate, we leave it to future work to confirm this. Part of the challenge here is that our numerical simulations used Fock space simulations at high truncation dimensions (roughly 1000), which is infeasible for multiple modes.

Finally, an open question is whether a multimode GKP code exists in which implementing the logical T gate is easier and induces less errors. A trivial solution to this is to concatenate the GKP code with a DV code that has a transversal T gate (which is equivalent to defining a multimode GKP code with the same stabilizers); however, moving beyond such a concatenated framework would be interesting. If such a code exists, one could envision performing code switching to this code for the T gate and switching back to the square GKP code for standard Clifford operations.

One potentially interesting avenue of future work is to find other applications of our on-demand noise-biasing technique beyond enabling fault-tolerant non-Clifford logical gates. As a possible example, perhaps on-demand morphing that we discussed in Appendix A could be used to improve the error rate of operations involving DV qubits by shortening the conditional displacement gates between CV and DV qubits. However, we leave such considerations to future work.

ACKNOWLEDGMENTS

This work was supported by QuTech NWO funding 2020-2028—Part I “Fundamental Research,” Project No. 601.QT.001-1, financed by the Dutch Research Council (NWO). We acknowledge the use of the DelftBlue supercomputer for running the numerics. We thank Barbara M. Terhal for her support throughout the project and for providing feedback on a draft article. We thank Stefano Bosco for discussions on efficient numerical generation of approximate GKP codewords and for providing feedback on the draft article. We thank Xanda Kolesnikow for thoughtful discussions during the project.

DATA AVAILABILITY

There are no publicly available research data or software supporting this manuscript. Requests for further information or data should be sent to the authors.

APPENDIX A: ON-DEMAND MORPHING CIRCUITS

In the main text, we showed how to perform on-demand biasing by using a q -Steane QEC circuit in two ways in Eqs. (34) and (36). In this Appendix, we show how to perform on-demand *code morphing* from a nonbiased square GKP codestate to a nonbiased rectangular GKP code. This code morphing method has the advantage of using only passive Gaussian unitaries, while still enabling improved performance under polynomial phase gates due to the equivalence between nonbiased rectangular and biased square GKP codestates in Eq. (18). We follow a similar structure here as in the main text, beginning by how to construct an on-demand morphing circuit that introduces a *fixed* amount of asymmetry, before generalizing to a circuit that can generate arbitrary amounts of asymmetry.

We start with the q -Steane circuit rewritten in terms of a beam splitter and squeezing operators:

$$\begin{array}{c} |\bar{\psi}\rangle_{\Delta} \\ |\bar{\pm}\rangle_{\Delta}^{\lambda} \end{array} \rightarrow \text{BS}(\frac{\pi}{4}) \rightarrow \begin{array}{c} \text{Sq}^{\dagger}(\sqrt{2}) \\ \text{Sq}^{\dagger}(\frac{1}{\sqrt{2}}) \end{array} \rightarrow \text{meter } q, \quad (31b)$$

where the output (after the correction) is a biased square codestate with a fixed amount of bias $|\bar{\psi}\rangle_{\Delta/\sqrt{2}, \sqrt{2}\Delta}$. Now, if we remove the squeezing operators, the output state will be

$$\text{Sq}(\sqrt{2})|\bar{\psi}\rangle_{\Delta/\sqrt{2}, \sqrt{2}\Delta} = |\bar{\psi}\rangle_{\Delta}^{\lambda=2} \quad (A1)$$

by Eq. (18). Therefore, the circuit

$$\begin{array}{c} |\bar{\psi}\rangle_{\Delta} \\ |\bar{\pm}\rangle_{\Delta}^{\lambda} \end{array} \rightarrow \text{BS}(\frac{\pi}{4}) \rightarrow \text{meter } q \quad (A2)$$

is an on-demand morphing circuit from the square to the rectangular GKP code with $\lambda = 2$. Note that, compared to Eq. (31b), we will need to rescale the measurement outcome to account for the squeezing operator that we removed. Moreover, from Eq. (33), the envelope operator is preserved through Eq. (A2), and therefore both the input and output states will be nonbiased.

So far, we have only been able to morph from the square to the $\lambda = 2$ rectangular GKP code; however, we wish to be able to morph to rectangular GKP codes with arbitrary λ . To derive such a circuit, we begin with the (arbitrary- λ) on-demand biasing circuit from the main text:

$$\begin{array}{c} |\bar{\psi}\rangle_{\Delta} \\ |\bar{\pm}\rangle_{\Delta}^{\lambda} \end{array} \rightarrow e^{-i\sqrt{\lambda}q_1p_2} \rightarrow \text{meter } q, \quad (36)$$

which is written in the form such that it uses a nonbiased rectangular ancilla codestate.

Now, we wish to perform the same rearrangement as before by rewriting this circuit in terms of a beam splitter and

squeezing operations. In particular, we wish to find ξ, θ, α_1 , and α_2 such that

$$\begin{array}{c} |\bar{\psi}\rangle_{\Delta} \\ |\bar{\pm}\rangle_{\Delta}^{\lambda} \end{array} \rightarrow e^{-i\sqrt{\lambda}q_1p_2} \rightarrow e^{i\xi p_1q_2} \rightarrow \text{meter } q \\ = \begin{array}{c} |\bar{\psi}\rangle_{\Delta} \\ |\bar{\pm}\rangle_{\Delta}^{\lambda} \end{array} \rightarrow \text{BS}(\theta) \rightarrow \begin{array}{c} \text{Sq}(\alpha_1) \\ \text{Sq}(\alpha_2) \end{array} \rightarrow \text{meter } q \quad (A3)$$

As before, we are allowed to freely add the $e^{i\xi p_1q_2}$ term to Eq. (36) because it can be compensated for by a displacement correction $e^{-i\xi p_1q_m}$ based on the measurement outcome q_m .

One can solve this equation using the formalism of Gaussian operators. Given a two-mode Gaussian unitary operator U , we can always define a 4×4 matrix S such that

$$S \begin{bmatrix} q_1 \\ q_2 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} U^{\dagger} q_1 U \\ U^{\dagger} q_2 U \\ U^{\dagger} p_1 U \\ U^{\dagger} p_2 U \end{bmatrix}. \quad (A4)$$

Moreover, given an arbitrary 4×4 matrix S that satisfies the relation

$$S^T \Omega S = \Omega, \quad \Omega = \begin{bmatrix} 0 & \text{Id} \\ -\text{Id} & 0 \end{bmatrix} \quad (A5)$$

(that is, S is a *symplectic* matrix), one can uniquely define a corresponding Gaussian unitary operator [50]. With this, the corresponding S matrices for the left- and right-hand sides of Eq. (A3) are equal if and only if the unitaries themselves are equal. The corresponding matrices for the left- and right-hand sides, respectively, are given by

$$S_{\text{LHS}} = \begin{bmatrix} 1 - \xi\sqrt{\lambda} & -\xi & 0 & 0 \\ \sqrt{\lambda} & 1 & 0 & 0 \\ 0 & 0 & 1 & -\sqrt{\lambda} \\ 0 & 0 & \xi & 1 - \xi\sqrt{\lambda} \end{bmatrix}, \quad (A6a)$$

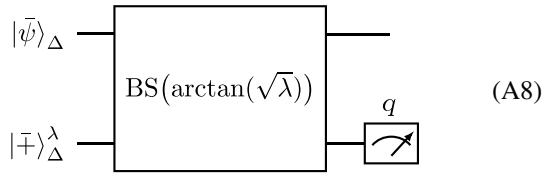
$$S_{\text{RHS}} = \begin{bmatrix} \cos(\theta)/\alpha_1 & -\sin(\theta)/\alpha_1 & 0 & 0 \\ \sin(\theta)/\alpha_2 & \cos(\theta)/\alpha_2 & 0 & 0 \\ 0 & 0 & \alpha_1 \cos(\theta) & -\alpha_1 \sin(\theta) \\ 0 & 0 & \alpha_2 \sin(\theta) & \alpha_2 \cos(\theta) \end{bmatrix}. \quad (A6b)$$

From here, it is straightforward to see that the solution is given by

$$\xi = \frac{\sqrt{\lambda}}{\lambda + 1}, \quad \theta = \arctan(\sqrt{\lambda}), \quad (A7a)$$

$$\alpha_1 = \sqrt{\lambda + 1}, \quad \alpha_2 = \frac{1}{\sqrt{\lambda + 1}}. \quad (A7b)$$

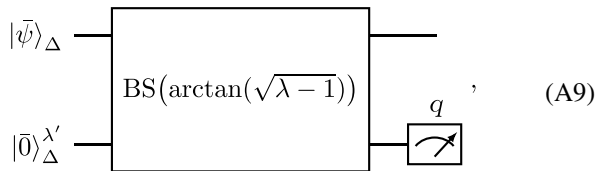
Therefore, the circuit



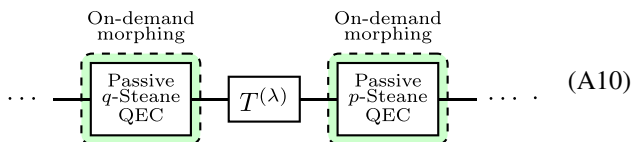
is an on-demand morphing circuit from the square to the $\lambda + 1$ rectangular GKP code, with output state (after the correction) given explicitly by $|\bar{\psi}\rangle_{\Delta}^{\lambda+1}$.

Having explained how the on-demand morphing circuit works, we now briefly summarize the difference between the on-demand biasing and on-demand morphing methods for performing a polynomial phase gate associated with the polynomial $P(x)$. In the on-demand biasing method, we follow the steps outlined in Eq. (37), first performing the on-demand biasing circuit [Eq. (34) or (36)], followed by the square GKP polynomial phase gate $\exp(i2\pi P(q/\sqrt{\pi}))$, and finally the passive Knill error-correction circuit [Eq. (14)] to return to the nonbiased codespace. In the on-demand morphing method, we first perform the on-demand morphing circuit [Eq. (A8)] and then perform the *rectangular* GKP polynomial phase gate $\exp(i2\pi P(q/\sqrt{\lambda\pi}))$. At this point, however, we are in a rectangular GKP code and therefore cannot simply perform the passive Knill error-correction circuit because this would keep us in the rectangular codespace instead of morphing us back to the square one.

Therefore, we need to perform a morphing circuit based on the p -Steane QEC circuit that returns us to the square GKP code. The derivation of this circuit is much the same as for Eq. (A8), but one does need to take care that the p -Steane QEC circuit that one begins with is for the rectangular GKP code instead of the square GKP code. With this, the morphing circuit is given explicitly by

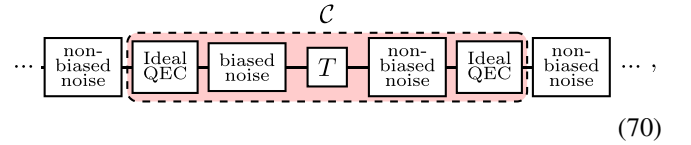


where $\lambda' = (1 - 1/\lambda)^{-1}$. In the full circuit, this looks like

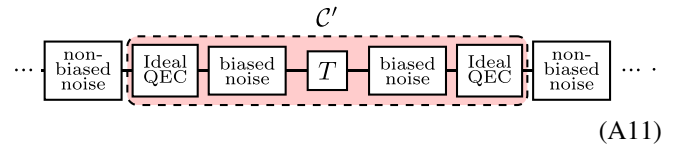


On-demand morphing has two potential advantages over on-demand biasing. First, Eqs. (A8) and (A9) use only passive Gaussian unitary operators (beam splitters) instead of logical CX gates that are not energy preserving, which may simplify experimental implementations. And second, because the resulting rectangular states are nonbiased, their average photon number is lower than the corresponding biased square codestates. This means that loss and dephasing may affect the nonbiased rectangular states less severely than the biased square codestates.

In Sec. VI, our simulations were based on the on-demand biasing method [Eq. (37)] instead of the on-demand morphing method [Eq. (A10)]. In particular, we simulated the circuit $\mathcal{C}$ given by



where the specific choices of biased and nonbiased noise are motivated by the QEC circuits used in the on-demand biasing method [Eq. (37)]. To adapt this circuit for on-demand morphing [Eq. (A10)], we would do two things. First, replace the biased and nonbiased boxes to match the noise that arises from the QEC circuits used in Eq. (A10), and second, use Eq. (18) (change the noise bias as appropriate) to find an equivalent circuit that acts in the square GKP code, giving



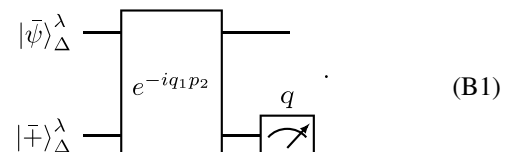
Simulating $\mathcal{C}'$ would therefore give quantitatively different results to $\mathcal{C}$, but should give *qualitatively* similar results because the noise coming from the final QEC round does not get spread by the T gate.

APPENDIX B: PASSIVE BREEDING CIRCUIT FOR NONBIASED-NOISE RECTANGULAR GKP QUBIT STATES

In all of the on-demand biasing and morphing circuits with $\lambda > 2$ in Eqs. (34), (36), and (A8), we have made use of either a biased square or nonbiased rectangular GKP ancilla state. However, we have not explained how one can produce such a state in the first place. While it may be possible to directly produce such an ancilla state by modifying existing state generation schemes [35–37,39,44], here we instead show how to obtain a nonbiased rectangular codestate iteratively from only nonbiased square GKP codestates. If desired, the nonbiased rectangular codestate could then be converted to a biased square codestate using a squeezing unitary [Eq. (18)].

The circuit to breed such a state is conceptually quite similar to the on-demand morphing circuit [Eq. (A8)], but here we have slightly different requirements that need to be satisfied. In particular, we require that each input ancilla state is a nonbiased square GKP codestate $|\bar{\psi}\rangle_{\Delta}$, but each input *data* state is a nonbiased rectangular GKP codestate $|\bar{\psi}\rangle_{\Delta}^{\lambda}$. We then want to output a nonbiased rectangular GKP codestate with an increased λ ; in particular, this will end up being $|\bar{\psi}\rangle_{\Delta}^{\lambda+1}$.

To construct this circuit, we begin with the q -Steane QEC circuit for the rectangular GKP code:



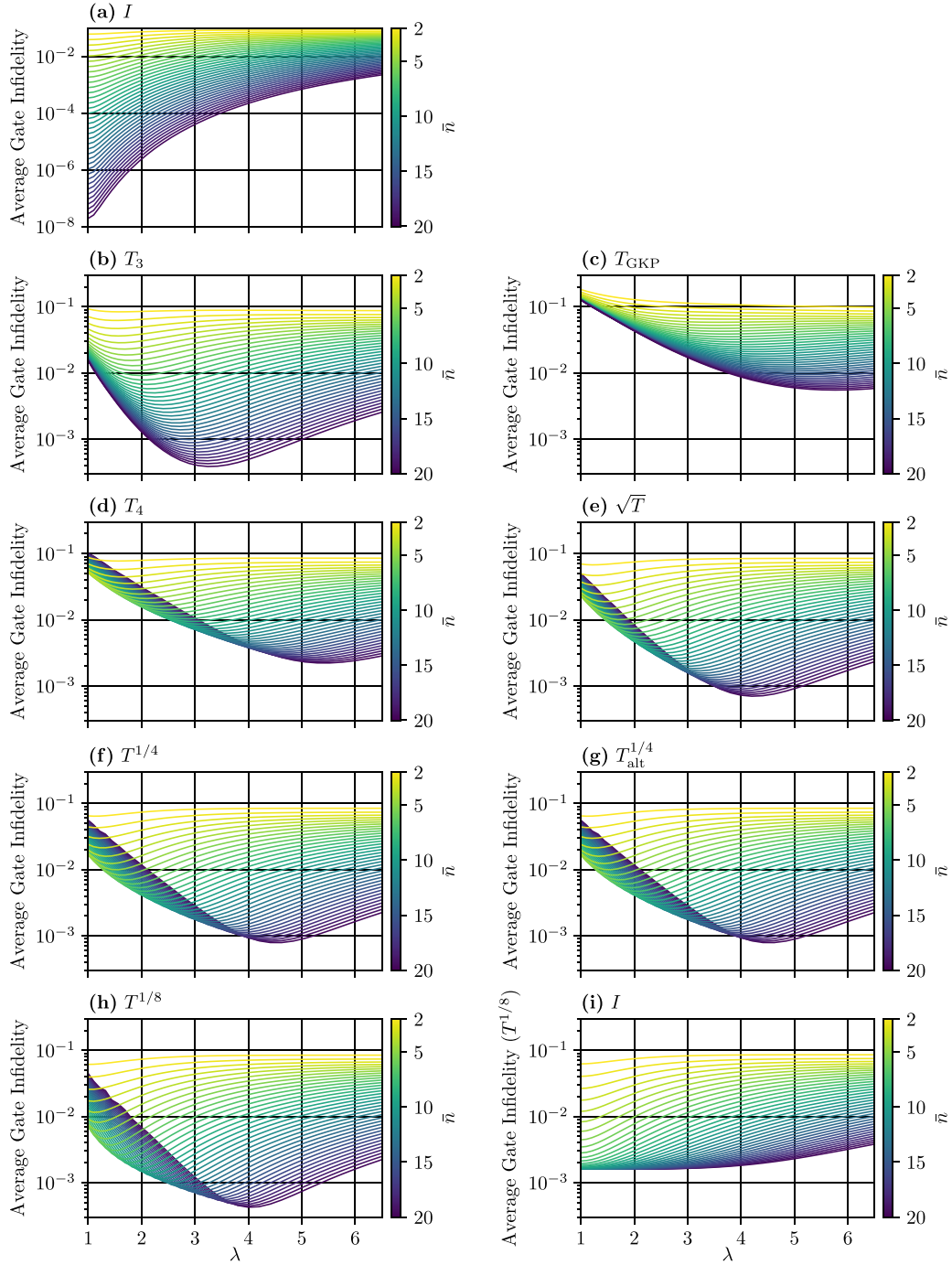


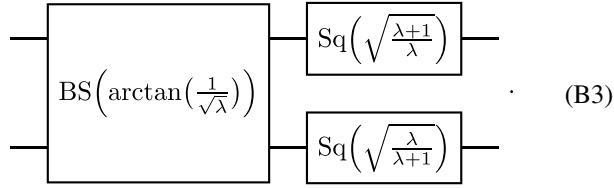
FIG. 6. Average gate infidelity plots for the polynomial phase gates in Table I and Eq. (H19), for each value of asymmetry and Δ as described in the text. Note that in panel (i), we implement the identity gate on the CV mode but calculate the average gate infidelity with respect to the $T^{1/8}$ gate.

We can turn the input rectangular ancilla codestate into a square codestate using Eq. (18):

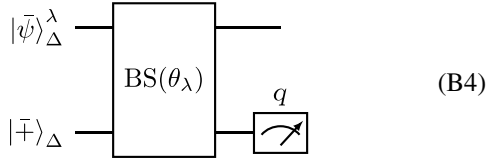
$$\begin{array}{c} |\bar{\psi}\rangle_{\Delta}^{\lambda} \\ |+\rangle_{\Delta} \end{array} \rightarrow \left[e^{-i \frac{q_1 p_2}{\sqrt{\lambda}}} \right] \rightarrow q \quad (B2)$$

It is straightforward following the same method as in Appendix A that the q -Steane QEC circuit in Eq. (B2) can be rewritten in terms of beam splitters and squeezers

$$\left[e^{-i \frac{q_1 p_2}{\sqrt{\lambda}}} \right] \rightarrow \left[e^{i \frac{\sqrt{\lambda}}{\lambda+1} p_1 q_2} \right] =$$



By absorbing the single-mode squeezer on the ancilla into the homodyne measurement and omitting the squeezer on the data qubit, we can map a rectangular GKP state with nonbiased noise and asymmetry λ to one with asymmetry $\lambda+1$:



Therefore, by repeating Eq. (B4) a total of $(\lambda - 1)$ times, we can generate a nonbiased rectangular GKP state $|+\rangle_\Delta^\lambda$ from λ nonbiased square GKP states $|+\rangle_\Delta$. We refer to this procedure as the passive rectangular GKP state breeding protocol. Importantly, the output of this protocol will always have $\lambda \in \mathbb{Z}$; however, this is not a problem because λ does not need to be fine-tuned in order to achieve close-to-optimal error rates, as we show in Figs. 6 and 7.

APPENDIX C: PROOF OF NO-GO THEOREM

In the main text, we showed how to perform on-demand biasing by using a q -Steane QEC circuit in two ways in Eqs. (34) and (36). We found that the noise profile after the

on-demand biasing circuit satisfies the trade-off relation

$$\Delta_q \Delta_p \geq \Delta^2. \quad (39)$$

This naturally raises the question of whether a syndrome measurement circuit could surpass the trade-off relation in Eq. (39). In other words, is it possible to suppress noise in one quadrature without inducing a corresponding increase in the other?

In this Appendix, we show in Theorem C1 that no such circuit can be constructed using deterministic Gaussian operations and approximate GKP ancillas. As a consequence, the q -Steane QEC circuits that we present achieve the optimal output Δ_q and Δ_p parameters for deterministic Gaussian circuits.

Before presenting the formal statement of Theorem C1, we briefly review the generalization of the Gaussian random displacement channel to multimode system and the definition of symplectic matrices. In an N -mode system, the N -mode displacement operator $W(\mathbf{v})$ is defined as

$$W(\mathbf{v}) = \exp(i\sqrt{2\pi}\boldsymbol{\zeta}\Omega\mathbf{v}), \quad (C1)$$

where $\boldsymbol{\zeta}$ is the quadrature column vector

$$\boldsymbol{\zeta} = [q_1 \quad \cdots \quad q_N \quad p_1 \quad \cdots \quad p_N]^T, \quad (C2)$$

and the matrix Ω is the $2N \times 2N$ symplectic form

$$\Omega = \begin{bmatrix} \mathbf{0}_{N \times N} & \text{Id}_{N \times N} \\ \text{Id}_{N \times N} & \mathbf{0}_{N \times N} \end{bmatrix}. \quad (C3)$$

The group of $2N \times 2N$ symplectic matrices S is the group of real matrices that preserve the symplectic form

$$S^T \Omega S = \Omega. \quad (C4)$$

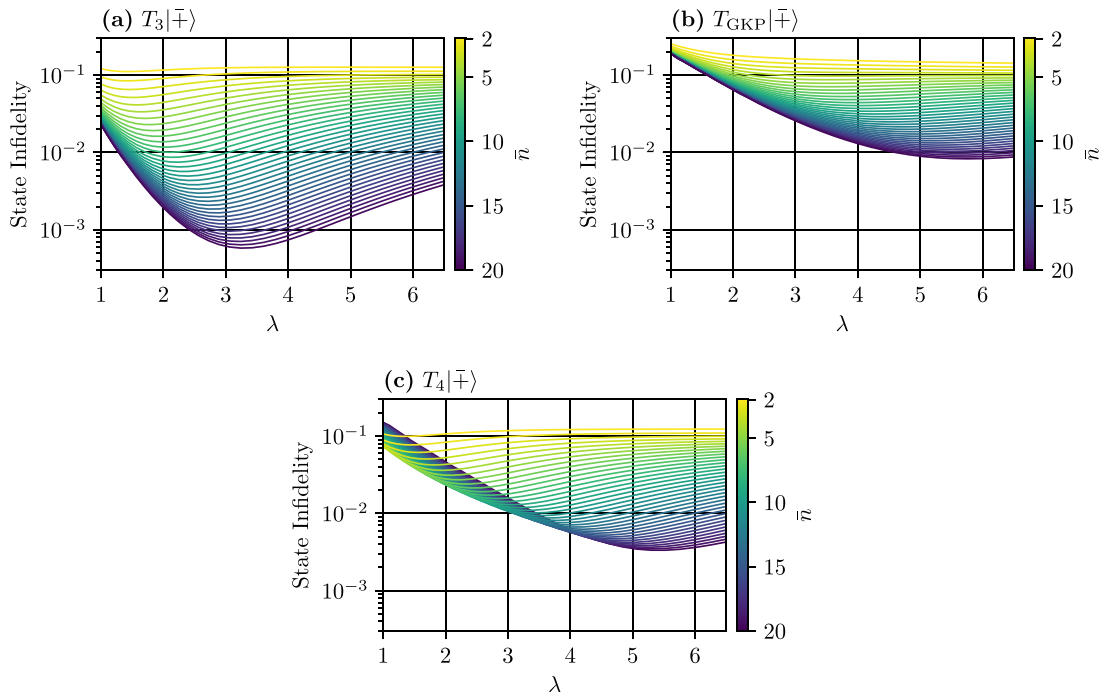


FIG. 7. State infidelity plots for the polynomial phase gates in Table I that implement a T gate, for each value of asymmetry and Δ as described in the text.

Gaussian unitary operators are, by definition, unitary operators that map quadrature operators to other quadrature operators; or, equivalently, they are operators that can be written as a product of exponents that are quadratic in the quadrature operators. Gaussian unitary operators that contain *only* quadratic terms are in one-to-one correspondence with symplectic matrices, given by the relation

$$U_S^\dagger \xi U_S = S\xi, \quad (\text{C5})$$

where U_S acts componentwise on the elements of ξ .

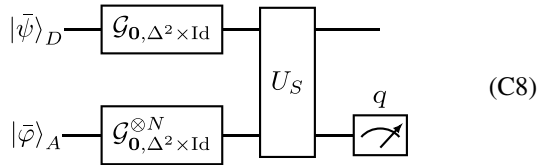
The multimode Gaussian random displacement channel $\mathcal{G}_{\mu, \Sigma}$ is defined similar to the two-mode channel in Eq. (23):

$$\mathcal{G}_{\mu, \Sigma}(\rho) = \int d\mathbf{v} \frac{\exp[-\pi(\mathbf{v} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{v} - \boldsymbol{\mu})]}{\sqrt{|\det \Sigma|}} \quad (\text{C6})$$

$$\times W(\mathbf{v})\rho W(\mathbf{v})^\dagger, \quad (\text{C7})$$

but here we have a $2N$ -dimension normal distribution; note here that unlike Eq. (23) we also consider channels with nonzero mean $\boldsymbol{\mu}$.

We consider the following model. We assume that the data qubit and the ancillary qubits can be encoded using different GKP codes. All physical modes are subjected to independent Gaussian displacement noise channels $\mathcal{G}_{0, \Delta^2 \times \text{Id}}$, which is the noise model one obtains when applying the twirling approximation to Env_Δ . The syndrome extraction circuit consists of Gaussian unitary operators, ideal homodyne measurements on the ancillas, and arbitrary classical decoders. Under these assumptions, the circuit model under consideration is given by



where $|\bar{\psi}\rangle_D$ is a single-mode ideal GKP “data” codestate, $|\bar{\phi}\rangle_A$ is an N -mode ideal GKP “ancilla” codestate, U_S is a Gaussian unitary with the associated symplectic matrix S , and Δ^2 is the variance of the noise applied to each GKP codestate. Note that homodyne measurements in other bases can be incorporated into this model by writing them as a Gaussian unitary preceding a position measurement, and then incorporating the Gaussian unitary into U_S .

Theorem C1. The resulting state of the data qubit after the circuit [Eq. (C8)] and the displacement correction step is

$$\int d\mathbf{v} \mathcal{X}(\mathbf{v}) W(\mathbf{v}) |\bar{\psi}\rangle_D \langle \bar{\psi}|_D W(\mathbf{v})^\dagger, \quad (\text{C9})$$

where $\mathcal{X}(\mathbf{v})$ is a mixture of Gaussian distributions

$$\mathcal{X}(\mathbf{v}) = \sum_i p_i \mathcal{N}(\boldsymbol{\mu}_i, \Delta^2 \Sigma', \mathbf{v}) \quad (\text{C10})$$

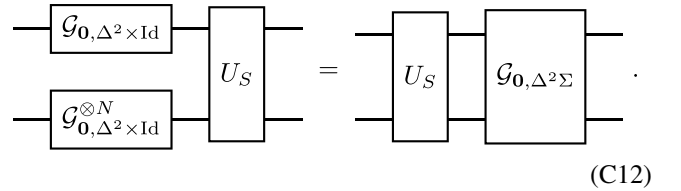
for some positive weights p_i satisfying $\sum_i p_i = 1$, for some mean vectors $\boldsymbol{\mu}_i$, and for some covariance matrix Σ' satisfying $\det(\Sigma') = 1$. The random displacement vector $\mathbf{v}$ drawn from the distribution $\mathcal{X}(\mathbf{v})$ satisfies the inequality

$$\det(E[(\mathbf{v} - E[\mathbf{v}])(\mathbf{v} - E[\mathbf{v}])^T]) \geq \Delta^4. \quad (\text{C11})$$

The circuit in Eq. (C8) can be interpreted as a noise-biasing circuit with noisy ancillas, while Eq. (C11) provides a lower bound on the product of the variances of position and momentum errors. The determinant appearing in Eq. (C11) is the generalization of the informal trade-off relation in Eq. (39), extending it to include cross-correlations between position and momentum displacement errors.

As a final comment, we emphasize that although Theorem C1 holds regardless of the choice of decoder, it does not imply that all decoders perform equally well. The role of a good decoder is to approach the lower bound as close as possible. Theorem C1 also shows that the biased q -Steane QEC circuit presented in Eq. (34) achieves this optimal bound.

Proof. Since U_S is a Gaussian unitary operator, we can commute the random Gaussian displacement channels $\mathcal{G}_{0, \Delta^2 \times \text{Id}}$ through U_S , resulting in a new channel $\mathcal{G}_{0, \Delta^2 \Sigma}$.



The normalized covariance matrix $\Sigma = S S^T$ is both symplectic and symmetric, and because it is symplectic it has unit determinant $\det(\Sigma) = 1$.

We decompose Σ into a 4×4 block matrix

$$\Sigma = \begin{pmatrix} [\Sigma_{qq}]_{DD} & [\Sigma_{qq}]_{DA} & [\Sigma_{qp}]_{DD} & [\Sigma_{qp}]_{DA} \\ [\Sigma_{qq}]_{AD} & [\Sigma_{qq}]_{AA} & [\Sigma_{qp}]_{AD} & [\Sigma_{qp}]_{AA} \\ [\Sigma_{pq}]_{DD} & [\Sigma_{pq}]_{DA} & [\Sigma_{pp}]_{DD} & [\Sigma_{pp}]_{DA} \\ [\Sigma_{pq}]_{AD} & [\Sigma_{pq}]_{AA} & [\Sigma_{pp}]_{AD} & [\Sigma_{pp}]_{AA} \end{pmatrix}. \quad (\text{C13})$$

Here, the subscripts q and p refer to the position and momentum quadratures, respectively. The labels D and A indicate whether the corresponding block involves the data qubit (D) or ancillary modes (A). For example, $[\Sigma_{qp}]_{DA}$ denotes the cross-covariance between the position quadrature of the data qubit and the momentum quadratures of the ancilla qubits. The size of each block is

$$\begin{aligned} [\Sigma_{qq}]_{DD}, [\Sigma_{pp}]_{DD}, [\Sigma_{pq}]_{DD}, [\Sigma_{qp}]_{DD} &\in \mathbb{R}^{1 \times 1}, \\ [\Sigma_{qq}]_{AD}, [\Sigma_{qp}]_{AD}, [\Sigma_{pq}]_{AD}, [\Sigma_{pp}]_{AD} &\in \mathbb{R}^{N \times 1}, \\ [\Sigma_{qq}]_{DA}, [\Sigma_{qp}]_{DA}, [\Sigma_{pq}]_{DA}, [\Sigma_{pp}]_{DA} &\in \mathbb{R}^{1 \times N}, \\ [\Sigma_{qq}]_{AA}, [\Sigma_{pp}]_{AA}, [\Sigma_{pq}]_{AA}, [\Sigma_{qp}]_{AA} &\in \mathbb{R}^{N \times N}. \end{aligned} \quad (\text{C14})$$

Since Σ is symplectic, its inverse Σ^{-1} is also symplectic and can be expressed in terms of the entries of Σ , i.e., $\Sigma^{-1} = \Omega^T \Sigma^T \Omega$. The explicit 4×4 block matrix of Σ^{-1} is given by

$$\Sigma^{-1} = \begin{pmatrix} [\Sigma_{pp}]_{DD} & [\Sigma_{pp}]_{DA} & -[\Sigma_{qp}]_{DD} & -[\Sigma_{qp}]_{DA} \\ [\Sigma_{pp}]_{AD} & [\Sigma_{pp}]_{AA} & -[\Sigma_{qp}]_{AD} & -[\Sigma_{qp}]_{AA} \\ -[\Sigma_{pq}]_{DD} & -[\Sigma_{pq}]_{DA} & [\Sigma_{qq}]_{DD} & [\Sigma_{qq}]_{DA} \\ -[\Sigma_{pq}]_{AD} & -[\Sigma_{pq}]_{AA} & [\Sigma_{qq}]_{AD} & [\Sigma_{qq}]_{AA} \end{pmatrix}. \quad (\text{C15})$$

Because we measure only the position quadrature of the ancillary modes, we gain no information about displacement

errors occurring in the momentum quadrature. Therefore, the joint distribution of displacement errors on the position and momentum quadratures of the data qubit, as well as the position quadratures of the ancillas, is described by the marginal of the full error distribution. Specifically, this corresponds to marginalizing the original Gaussian distribution with covariance matrix $\Delta^2 \Sigma$. The marginal of a multivariate Gaussian distribution is itself a Gaussian, with covariance matrix $\Delta^2 \tilde{\Sigma}$, where $\tilde{\Sigma}$ is obtained by removing the rows and columns of Σ corresponding to the unmeasured variables—in this case, the momentum quadratures of the ancilla qubits. For our block ordering, this corresponds to removing the fourth row and fourth column of Σ . For convenience, we permute the second and third block row and block column of $\Delta^2 \tilde{\Sigma}$ to obtain

$$\Delta^2 \tilde{\Sigma} = \Delta^2 \begin{pmatrix} [\Sigma_{qq}]_{DD} & [\Sigma_{qp}]_{DD} & [\Sigma_{qq}]_{DA} \\ [\Sigma_{pq}]_{DD} & [\Sigma_{pp}]_{DD} & [\Sigma_{pq}]_{DA} \\ [\Sigma_{qq}]_{AD} & [\Sigma_{qp}]_{AD} & [\Sigma_{qq}]_{AA} \end{pmatrix}, \quad (\text{C16a})$$

$$= \Delta^2 \begin{pmatrix} [\tilde{\Sigma}]_{DD} & [\tilde{\Sigma}]_{DA} \\ [\tilde{\Sigma}]_{AD} & [\tilde{\Sigma}]_{AA} \end{pmatrix}. \quad (\text{C16b})$$

After performing position homodyne measurements on the ancillary qubits, we obtain partial information about the displacement error introduced by the Gaussian random displacement channel: Since the ancillary qubits are encoded in GKP codestates, the measurement only reveals the displacement modulo the GKP grid. This means that we can only determine the error up to a logical operation. For example, in the standard q -Steane circuit where both the data and ancillary qubits are square-lattice GKP qubits, the measurement outcome allows us to infer the displacement error on the ancillary qubit only up to a logical X operator.

Consequently, the measurement outcome is consistent with a discrete set of possible position displacements $\{\mathbf{u}_{q,i}\}$, each occurring with probability p_i such that $\sum_i p_i = 1$ and $p_i \geq 0$. These tuples $\{p_i, \mathbf{u}_{q,i}\}$ depend on the GKP encodings of the data and ancillary modes and are typically intractable to compute analytically. Fortunately, explicit knowledge of these values is not required for the analysis that follows.

Now, suppose that a specific displacement error $\mathbf{u}_{q,i}$ occurred on the ancillary modes. We are then interested in the resulting displacement error on the data qubit, conditioned on this event. In other words, we wish to determine the conditional distribution of the random displacement on the data qubit given that $\mathbf{u}_{q,i}$ occurred on the ancillary qubit.

As discussed earlier, the joint distribution of displacement errors is a multivariate Gaussian with zero mean and covariance matrix $\Delta^2 \tilde{\Sigma}$. Given this, the conditional distribution of the data qubit's displacement—conditioned on a displacement error $\{\mathbf{u}_{q,i}\}$ occurring on the ancillary qubit—is also Gaussian. Its mean and covariance are given by

$$\begin{aligned} \mu_i &= [\tilde{\Sigma}]_{DA} \frac{1}{[\tilde{\Sigma}]_{AA}} \mathbf{u}_{q,i}, \\ \Sigma' &= [\tilde{\Sigma}]_{DD} - [\tilde{\Sigma}]_{DA} \frac{1}{[\tilde{\Sigma}]_{AA}} [\tilde{\Sigma}]_{AD} = \frac{1}{[\tilde{\Sigma}^{-1}]_{DD}}. \end{aligned} \quad (\text{C17})$$

Therefore, the state of the data qubit after the homodyne measurement step is given by

$$\sum_i p_i \int d\mathbf{v} \mathcal{N}(\mu_i, \Delta^2 \Sigma', \mathbf{v}) W(\mathbf{v}) |\bar{\psi}\rangle_D \langle \bar{\psi}|_D W(\mathbf{v})^\dagger. \quad (\text{C18})$$

We note that the conditional covariance matrices are identical and independent of the measured syndrome. This fact is crucial since it shows that the role of a decoder is simply to determine the correction displacement that we will apply. Therefore, the decoder only shifts the error profile, and cannot change the covariance matrix Σ' .

We now prove that $\det(\Delta^2 \Sigma') = \Delta^4$ [or equivalently $\det(\Sigma') = 1$] by utilizing the following fact, which holds for any general invertible 2×2 block matrix:

$$\det(A) = \frac{\det([A]_{11})}{\det([A^{-1}]_{22})}. \quad (\text{C19})$$

If we identify $A = \Delta^2 \Sigma$, $[A]_{11} = \Delta^2 \tilde{\Sigma}$, and $[A^{-1}]_{22} = \frac{[\Sigma_{pp}^{-1}]_{AA}}{\Delta^2} = \frac{[\Sigma_{qq}]_{AA}}{\Delta^2}$, the determinant of the marginalized covariance matrix $\tilde{\Sigma}$ is given by

$$\det(\Delta^2 \tilde{\Sigma}) = \det\left(\frac{[\Sigma_{pp}^{-1}]_{AA}}{\Delta^2}\right) \det(\Delta^2 \Sigma) \quad (\text{C20a})$$

$$= \frac{\det([\Sigma_{pp}]_{AA})}{(\Delta^2)^N} (\sigma^2)^{2N+2} \quad (\text{C20b})$$

$$= (\Delta^2)^{N+2} \det([\Sigma_{qq}]_{AA}). \quad (\text{C20c})$$

Applying the same formula for $A = \Delta^2 \tilde{\Sigma}$ in Eq. (C16a) and using the Bayesian rules from Eq. (C17), we also have

$$\det(\Delta^2 \tilde{\Sigma}) = \frac{\det(\Delta^2 [\tilde{\Sigma}]_{AA})}{\det\left(\frac{[\tilde{\Sigma}^{-1}]_{DD}}{\Delta^2}\right)} \quad (\text{C21a})$$

$$= (\Delta^2)^N \det(\Delta^2 \Sigma') \det([\Sigma_{qq}]_{AA}). \quad (\text{C21b})$$

Comparing Eqs. (C20) and (C21), we conclude that

$$\det(\Delta^2 \Sigma') = \Delta^4. \quad (\text{C22})$$

The decoder then decides which correction $\tilde{\mu}$ to apply based on some strategy. Therefore, after the correction steps, the error profiles on the data qubit state is

$$\sum_i p_i \mathcal{G}_{\tilde{\mu}_i, \Delta^2 \Sigma'}(|\psi_D\rangle \langle \psi_D|), \quad \tilde{\mu}_i = \mu_i - \tilde{\mu}. \quad (\text{C23})$$

Since Eq. (C23) is a mixture of Gaussian displacement channels with identical covariance matrices $\Delta^2 \Sigma'$, we intuitively expect the covariance matrix of the random variable $\mathbf{v}$ with distribution function $\sum_i p_i \mathcal{N}(\tilde{\mu}_i, \Delta^2 \Sigma', \mathbf{v})$ to have larger generalized variance than $\Delta^2 \Sigma'$, namely,

$$\det(\mathbb{E}[(\mathbf{v} - \mathbb{E}[\mathbf{v}])(\mathbf{v} - \mathbb{E}[\mathbf{v}])^T]) \geq \det(\Delta^2 \Sigma') = \Delta^4. \quad (\text{C24})$$

This fact can be proven rigorously using the generalized Jensen inequality and the Minkowski determinant inequality. By direct computation, we can show that

$$\begin{aligned} &\mathbb{E}[(\mathbf{v} - \mathbb{E}[\mathbf{v}])(\mathbf{v} - \mathbb{E}[\mathbf{v}])^T] \\ &= \Delta^2 \Sigma' + \sum_i p_i (\tilde{\mu}_i \tilde{\mu}_i^T) - \left(\sum_i p_i \tilde{\mu}_i \right) \left(\sum_j p_j \tilde{\mu}_j^T \right) \end{aligned} \quad (\text{C25a})$$

$$= \Delta^2 \Sigma' + \Sigma_r. \quad (\text{C25b})$$

The matrix Σ_r is symmetric and also positive semidefinite. To establish the positive semidefiniteness of Σ_r , we will

demonstrate that

$$\mathbf{x}^T \Sigma_r \mathbf{x} \geq 0, \quad \forall \mathbf{x} \in \mathbb{R}^2. \quad (\text{C26})$$

To this end, we define the function $\varphi_{\mathbf{x}} : \mathbb{R}^2 \rightarrow \mathbb{R}$ for all $\mathbf{x} \in \mathbb{R}^2$:

$$\varphi_{\mathbf{x}}(\tilde{\mu}_i) = (\mathbf{x}^T \tilde{\mu}_i)^2, \quad (\text{C27})$$

which is a convex function. Therefore, using the generalized Jensen inequality, we can prove that

$$\begin{aligned} \sum_i p_i \varphi_{\mathbf{x}}(\tilde{\mu}_i) - \varphi_{\mathbf{x}}\left(\sum_i p_i \tilde{\mu}_i\right) &\geq 0, \\ \forall p_i &\geq 0 \quad \text{and} \quad \sum_i p_i = 1. \end{aligned} \quad (\text{C28})$$

Since the left-hand side (LHS) of the above inequality is simply $\mathbf{x}^T \Sigma_r \mathbf{x}$ and the inequality holds for any $\mathbf{x}$, the covariance matrix Σ_r is positive semidefinite.

As the matrices Σ' and Σ_r are positive semidefinite matrices, the Minkowski determinant inequality implies

$$\begin{aligned} \det(E[(\mathbf{v} - E[\mathbf{v}])(\mathbf{v} - E[\mathbf{v}])^T]) \\ \geq \det(\Delta^2 \Sigma') + \det(\Sigma_r) \end{aligned} \quad (\text{C29a})$$

$$\geq \det(\Delta^2 \Sigma') = \Delta^4 \quad (\text{C29b})$$

ending the proof. $\blacksquare$

We remark that the main result presented in Eq. (C11) continues to hold even when displacement operators are allowed in combination with Gaussian unitaries. Two modifications arise in the proof:

(1) In Eq. (C12), after commuting through the gates, the mean displacement of the resulting Gaussian displacement channel may no longer remain zero.

(2) In Eq. (C17), the mean displacement of the Gaussian displacement channel acting on the data qubit after measuring the ancilla will be modified.

Aside from these shifts in the mean, the structure of the proof is unchanged. The key reasoning relies solely on the covariance matrix of the noise profile after measurement, as given in Eq. (C17), and this covariance matrix cannot be altered by displacement operators.

APPENDIX D: MINIMAL REPRESENTATIONS OF POLYNOMIAL PHASE GATES

In Sec. IV, we introduced the formalism of polynomial phase stabilizers and the coefficient reduction procedure to generate minimal polynomial phase gate representations of logical gates. In this Appendix, we prove a number of results relating to the coefficient reduction procedure. These results can be summarized in the following theorem.

Theorem D1. The minimal polynomial $P_{\min,m}(x)$, under the lexicographical order, that generates the logical gate Λ_m has degree

$$\deg P_{\min,m}(x) = m \quad (\text{D1})$$

and k th-order coefficients a_k that satisfy the bound

$$|a_k| \leq \frac{1}{2(k!)} \quad (\text{D2})$$

To prove this, we begin by proving that the coefficient reduction procedure always returns a minimal polynomial under the lexicographic partial ordering introduced in Sec. IV, thereby establishing the second part of Theorem D1. Appendix D2 demonstrates that the minimal polynomials $P_{\min,m}(x)$ form a hierarchy satisfying $\deg(P_{\min,m+1}) > \deg(P_{\min,m})$, which serves as a stepping stone for the following section. Finally, in Appendix D3, we use mathematical induction to show that if $\deg(P_{\min,m}) = m$, then $\deg(P_{\min,m+1}) = m+1$. Together with the base case $\deg(P_{\min,1}) = 1$, this completes the proof of the first part of Theorem D1.

A recent result [51] establishes that non-Clifford gates in the GKP code cannot be implemented *deterministically* using Gaussian operations alone. Our theorem complements this by identifying the minimal non-Gaussian resources required to realize a given logical non-Clifford gate with a single polynomial phase gate.

We remark that if x denotes either q or p , corresponding to Z -type or X -type phase gates in the GKP code, Theorem D1 can be readily proven using the Clifford hierarchy and induction. However, the proof we present below is more general, as it also applies to other families of gates, such as the Hadamard-type gates that lie outside the Clifford hierarchy, and is applicable to other bosonic codes.

Throughout this Appendix, we write

$$W(x) \equiv Q(x) \quad (\text{D3})$$

to denote that the polynomials $W(x)$ and $Q(x)$ are equivalent up to the addition of an integer-valued polynomial $\sum_i c_i L_i(x)$ for $c_i \in \mathbb{Z}$. One simple fact that we will repeatedly use is as follows.

Lemma D1. For any integer-valued polynomial $P(x) \equiv 0$ of degree m , its leading coefficient is an integer multiple of $1/m!$.

The proof is fairly simple but is important to show that linear combinations of higher-degree generating polynomials cannot be combined to create a lower-degree polynomial with a smaller leading coefficient.

Proof. This is a direct consequence of the generating set of integer-valued polynomials $\{L_i(x)\}$ defined in Eq. (44). Because $\{L_i(x)\}$ is a generating set, we can always write

$$P(x) = \sum_{j=1}^J c_j L_j(x), \quad (\text{D4})$$

for some $J \in \mathbb{Z}$ and $c_j \in \mathbb{Z}$, where we assume without loss of generality that $c_J \neq 0$. Now, note that the degree of $P(x)$ is J , and therefore we have $J = m$. Because the leading coefficient of $L_m(x)$ is $1/m!$, the lemma follows. $\blacksquare$

1. Optimality of the coefficient reduction procedure

We first prove that the coefficient reduction procedure is guaranteed to return a minimal polynomial under the lexicographical order. Consequently, starting from any polynomial $P_{\text{start}}(x)$ of the Λ_m gate, we can always obtain a minimal polynomial $P_{\min,m}(x)$ by applying the coefficient reduction procedure. Since the resulting polynomial from this procedure always has coefficients a_k that satisfy Eq. (D2), this establishes the second part of Theorem D1.

We begin by present how to generate a starting polynomial representation $P_{\text{start}}(x)$ for any Λ_m gate. We note that for the Λ_1 gate, there is a minimal representation

$$P_{\text{min},1}(x) = \frac{x}{2}, \quad (\text{D5})$$

which, in the usual case where $x = q/\sqrt{\pi}$, corresponds to the logical $Z = e^{i\sqrt{\pi}q}$ gate via Eq. (46). In the rest of this Appendix, we assume that the polynomial only takes integer inputs; of course, when it is turned into a polynomial phase gate the polynomial will take real values again. Let us assume that we have found a representation P_m for the Λ_m gate. Then, a representative $P_{\text{start}}(x)$ for the Λ_{m+1} gate can be constructed as

$$P_{\text{start}}(x) = 2^{m-1}P_m^2(x), \quad (\text{D6})$$

which is a generalization of a result from Ref. [40].

The recursive formula presented in Eq. (D6) can be explained as follows. Since P_m is a polynomial whose polynomial phase gate implements Λ_m , we can write $P_m(x)$ as

$$P_m(x) = n(x) + \frac{a(x)}{2^m}, \quad (\text{D7})$$

where $n(x)$ is some integer depending on x and $a(x) = 0$ for even x and $a(x) = 1$ for odd x . Substituting Eq. (D7) into Eq. (D6), we obtain

$$P_{\text{start}}(x) = 2^{m-1}n^2(x) + n(x)a(x) + \frac{a(x)^2}{2^{m+1}} \quad (\text{D8a})$$

$$\Rightarrow P_{\text{start}}(x) \bmod 1 = \frac{a(x)^2}{2^{m+1}}, \quad (\text{D8b})$$

proving that $P_{\text{start}}(x)$ constructed from Eq. (D6) is indeed a polynomial representation of Λ_{m+1} .

From the definition of $P_{\text{min},1}(x)$ in Eq. (D5), the explicit formula for $P_m(x)$ is

$$P_m(x) = \frac{x^{2^{m-1}}}{2^m}. \quad (\text{D9})$$

Consequently, there exists a starting polynomial representation of the logical gate Λ_m for all $m \geq 1$. These trivial polynomial representations can be used as the starting polynomial $P_{\text{start}}(x)$ for the coefficient reduction procedure.

Now, we prove the following lemma relating to the optimality of the coefficient reduction procedure itself.

Lemma D2. Let $P_{\text{start}}(x)$ be any polynomial phase gate representation of the logical gate O :

$$P_{\text{start}}(x) = \sum_{i=0}^N a_i x^i. \quad (\text{D10})$$

Applying the coefficient reduction procedure to $P_{\text{start}}(x)$ always yields a minimal polynomial phase gate representation of the logical gate O under the lexicographic order, independent of the starting polynomial $P_{\text{start}}(x)$.

Proof. From the definition of the lexicographic order, if we can prove that at a given order j the coefficient reduction procedure produces the minimal coefficient in j , then we know that the overall procedure will output a minimal polynomial.

Due to the way we define the coefficient reduction procedure, at each order j the coefficient reduction procedure

outputs a coefficient r_j satisfying $|r_j| \leq 1/(2j!)$. From Lemma D1, the output coefficient is unique and cannot be reduced by any other integer-valued polynomial unless the remainder $r_j = \pm \frac{1}{2j!}$. In the case that the remainder $r_j = \pm \frac{1}{2j!}$, there are two possible branches in the procedure. There is no guarantee which branch will return the minimal polynomial under the lexicographic order. Therefore, we need to keep track of both branches and continue the procedure. Subsequent iterations might generate additional branches that we also need to keep track of. At the end of the procedure, choose the branch with the minimal polynomial. This guarantees that the coefficient reduction procedure always returns the minimal polynomial. ■

2. Hierarchy of minimal polynomials $P_{\text{min},m}$

From Lemma D2, we now have a constructive method to generate a minimal polynomial representation $P_{\text{min},m}(x)$ corresponding to the polynomial phase gate Λ_m . We now prove some properties of the resulting minimal polynomial representations, beginning in this section by establishing that the minimal polynomials $P_{\text{min},m}(x)$ form a hierarchy where the degree of $P_{\text{min},m+1}(x)$ is strictly larger than the degree of $P_{\text{min},m}(x)$. It will serve as a first step to the proof of Eq. (D1) that we will finish in the following section. In the rest of this Appendix, we denote the coefficients of the polynomial $P_{\text{min},m}(x)$ as

$$P_{\text{min},m}(x) = \sum_{j=1} p_{m,j} x^j. \quad (\text{D11})$$

We simply begin with a corollary to Lemma D1.

Corollary D1. Let $P(x)$ and $Q(x)$ be polynomials of degree n and m , respectively, where $n > m$. If $P(x) \equiv Q(x)$, then leading coefficient of $P(x)$ must be an integer multiple of $1/n!$.

Proof. This result follows directly from Lemma D1. Since $P(x) - Q(x) \equiv 0$, Lemma D1 implies that the leading coefficient of $P(x) - Q(x)$ must be an integer multiple of $1/n!$. Moreover, the leading coefficient of $P(x) - Q(x)$ coincides with the leading coefficient of $P(x)$ since $\deg P(x) > \deg Q(x)$, proving the result. ■

Given Corollary D1, we are now ready to present and prove the main result of this section.

Lemma D3. The minimal polynomial $P_{\text{min},m}(x)$ satisfies a hierarchy

$$\deg(P_{\text{min},m+1}) > \deg(P_{\text{min},m}). \quad (\text{D12})$$

Proof. Note that Eq. (D12) is already true for $m = 2$ since we know the minimal polynomials $P_{\text{min},1}(x) = x/2$ and $P_{\text{min},2}(x) = x^2/4$ by using the coefficient reduction procedure.

Therefore, working toward a proof by induction, we assume that Eq. (D12) holds for $P_{\text{min},m}(x)$, i.e.,

$$n = \deg(P_{\text{min},m}) > \deg(P_{\text{min},k}), \quad \forall k < m. \quad (\text{D13})$$

In the following paragraph, we determine the leading coefficient of $P_{\text{min},m}(x)$. Since the logical gates satisfy the recurrence relation $\Lambda_{m-1} = \Lambda_m^2$, their polynomial representations obey the corresponding recurrence equation

$$2P_{\text{min},m}(x) \equiv P_{\text{min},m-1}(x). \quad (\text{D14})$$

Given our assumption that $P_{\min,m}(x)$ is a minimal polynomial representation, its leading coefficient $p_{m,n}$ must satisfy the upper bound

$$|p_{m,n}| \leq \frac{1}{2(n!)} \quad (\text{D15})$$

However, from the recurrence relation in Eq. (D14) we know that $2P_{\min,m}(x) \equiv P_{\min,m-1}(x)$. The inductive assumption [Eq. (D13)] therefore allows us to use Corollary D1, showing that the leading coefficient of $2P_{\min,m}(x)$ must be

$$2|p_{m,n}| = \frac{1}{n!} \quad (\text{D16})$$

Given the leading coefficient of $P_{\min,m}(x)$ satisfies $|p_{m,n}| = 1/(2n!)$, we now show that it leads to

$$\deg P_{\min,m+1} > \deg P_{\min,m} \quad (\text{D17})$$

Assume, for the sake of contradiction, that

$$\deg P_{\min,m+1} < \deg P_{\min,m} \quad (\text{D18})$$

From the recurrence relation in Eq. (D14), we have

$$2P_{\min,m+1}(x) \equiv P_{\min,m}(x) \quad (\text{D19})$$

By the assumption [Eq. (D18)], the left-hand side is a polynomial of degree strictly less than n , while the right-hand side (RHS) has degree n . Corollary D1 implies that the leading coefficient $|p_{m,n}|$ of $P_{\min,m}(x)$ is $1/n!$, which is a contradiction to Eq. (D16). Therefore, we conclude that

$$\deg P_{\min,m+1} \geq \deg P_{\min,m} \quad (\text{D20})$$

We now eliminate the scenario in which the degree of $P_{\min,m+1}(x)$ equals the degree of $P_{\min,m}(x)$. Again, for the sake of contradiction, we assume that

$$\deg P_{\min,m+1} = \deg P_{\min,m} \quad (\text{D21})$$

Under this assumption, the recurrence relation [Eq. (D14)] implies that

$$2p_{m+1,n} + j \frac{1}{n!} = p_{m,n} = \pm \frac{1}{2(n!)}, \quad (\text{D22})$$

where j is some integer and $1/n!$ is simply the leading coefficient of the stabilizer $L_n(x)$. Together with the requirement that $|p_{m+1,n}| \leq 1/(2n!)$ from Eq. (D15), we can deduce that the leading coefficient of $P_{\min,m+1}(x)$ must necessarily be

$$|p_{m+1,n}| = \frac{1}{4(n!)} \quad (\text{D23})$$

In the following argument, we will show that Eq. (D23) cannot be true. We will make use of the identity

$$P_{\min,m+1}(x+2) - P_{\min,m+1}(x) \equiv 0, \quad (\text{D24})$$

which holds for all polynomial representations of the phase gate Λ_{m+1} . This identity follows from the observation that x and $x+2$ share the same parity; both are either even or odd. Consequently, the polynomials $P_{\min,m+1}(x+2)$ and $P_{\min,m+1}(x)$ represent the same logical gate.

However, the left-hand side of Eq. (D24) cannot be an integer-valued polynomial. To see this, we explicitly expand

the LHS:

$$P_{\min,m+1}(x+2) - P_{\min,m+1}(x) = \sum_{j=1}^n p_{m+1,j} [(x+2)^j - x^j] \quad (\text{D25a})$$

$$= 2np_{m+1,n}x^{n-1} + O(x^{n-2}) \quad (\text{D25b})$$

$$= \pm \frac{1}{2(n-1)!} x^{n-1} + O(x^{n-2}), \quad (\text{D25c})$$

where we have used that $(x+2)^j - x^j$ is a polynomial of degree $j-1$, and in the last line we used Eq. (D23). Since the coefficient of the leading term x^{n-1} is $\pm \frac{1}{2(n-1)!}$, Corollary D1 tells us that the LHS of Eq. (D24) cannot be an integer-valued polynomial. This contradicts the assumption in Eq. (D21). Therefore, we conclude that

$$\deg P_{\min,m+1} > \deg P_{\min,m}(x). \quad (\text{D26})$$

■

3. Degree of the minimal polynomial $P_{\min,m}(x)$

In the previous section, we have proven that the degree of the minimal polynomials $P_{\min,m}(x)$ follows a hierarchy Lemma D3. In this section, we will prove that the difference in the degrees of the polynomials $P_{\min,m}(x)$ and $P_{\min,m+1}(x)$ is 1, which leads to the theorem.

Lemma D4. The degree of any minimal polynomial phase gate representation of the Λ_m gate satisfies

$$\deg P_{\min,m} = m. \quad (\text{D27})$$

Proof. We will use induction to prove this lemma; the base case is $\deg(P_{\min,1}) = 1$. Before continuing, we will present two facts about the basis of integer-valued polynomials $\{L_n(x)\}$ that we will use. First, if n is odd, then $L_n(x)$ contains only odd powers of x . This can be shown by noting that for odd n the integer-valued polynomial $L_n(x)$ from Eq. (43) can be written as

$$L_n(x) = \frac{x}{n!} \prod_{i=1}^{(n-1)/2} (x^2 - i^2), \quad (\text{D28})$$

which contains only odd powers of x . Second, when n is even, the second-leading coefficient of $L_n(x)$ is nonzero and given by

$$\ell_{n,n-1} = \frac{1}{2(n-1)!}, \quad (\text{D29})$$

where we have written $L_n(x) = \sum_i \ell_{n,i} x^i$. This can be seen by noting that for even n , the stabilizer polynomial $L_n(x)$ can be expressed as

$$L_n(x) = \frac{(x+n/2)x \prod_{i=1}^{n/2-1} (x^2 - i^2)}{n(n-1)!} \quad (\text{D30a})$$

$$= \left(\frac{x}{n} + \frac{1}{2} \right) L_{n-1}(x). \quad (\text{D30b})$$

Expanding this product shows that the two leading coefficients of $L_n(x)$ are

$$\ell_{n,n} = \frac{1}{n!}, \quad \ell_{n,n-1} = \frac{1}{2(n-1)!}. \quad (\text{D31})$$

As our inductive assumption, we assume that the degree of $P_{\min,m}(x)$ is m . For convenience, we denote the degree of $P_{\min,m+1}(x)$ as

$$\deg(P_{\min,m+1}) = m + c, \quad (\text{D32})$$

where $c \geq 1$ due to Lemma D3. Our goal is to prove that $c = 1$. To that end, we will repeatedly use the following two identities:

$$P_{\min,m+1}(x) + P_{\min,m+1}(-x) \equiv P_{\min,m}(x), \quad (\text{D33a})$$

$$P_{\min,m+1}(x+2) - P_{\min,m+1}(x) \equiv 0. \quad (\text{D33b})$$

These identities follow directly from the recurrence relation for the phase gates in Eq. (D14), along with the fact that the action of Λ_m on a y eigenstate of the x operator $|y\rangle_x$ is determined only by whether y is even or odd and not on its specific value.

Our proof strategy is to show that if $c > 1$, then either Eq. (D33a) or/and Eq. (D33b) must be violated. This contradiction will imply that $c = 1$ is the only consistent possibility. To carry out the argument, we consider two separate cases depending on the parity of m : when m is even and when m is odd.

Case 1: m is an even number: Subcase A: $c \notin \{2, 4, \dots\}$. As a first step, we show that c cannot belong to the set of even integers greater than or equal to 2, i.e., $c \notin \{2, 4, \dots\}$. Suppose, for the sake of contradiction, that c belongs to this set. Then, Eq. (D33a) implies that

$$P_{\min,m}(x) \equiv P_{\min,m+1}(x) + P_{\min,m+1}(-x) \quad (\text{D34a})$$

$$= \pm \frac{x^{m+c}}{(m+c)!} + 2 \sum_{i=2,4,\dots}^{m+c-2} p_{m+1,i} x^i \quad (\text{D34b})$$

$$\equiv 2 \sum_{i=2,4,\dots}^{m+c-2} p_{m+1,i} x^i \mp \sum_{i=1}^{m+c-1} \ell_{m+c,i} x^i, \quad (\text{D34c})$$

where we have used Eq. (D16) in the second line, and we have added or subtracted the polynomial $L_{m+c}(x)$ in the third line to remove the leading term on the second line.

The leading coefficient in Eq. (D34c) is $\ell_{m+c,m+c-1}$ and its value is given by Eq. (D29), which is *not* an integer multiple of $1/(m+c-1)!$. On the other hand, the degree of Eq. (D34c) is $m+c-1$, which is greater than the degree of $P_{\min,m}(x)$. Therefore, Corollary D1 implies that the leading coefficient on the third line of Eq. (D34) must be integer multiple of $1/(m+c-1)!$. This leads to a contradiction and we conclude that c cannot belong to the set of positive even integers.

Subcase B: $c \notin \{3, 5, \dots\}$. We now prove that c cannot be in the set $\{3, 5, \dots\}$. Let us assume by contradiction that $c \in \{3, 5, \dots\}$. Then, Eq. (D33a) implies that

$$P_{\min,m}(x) \equiv P_{\min,m+1}(x) + P_{\min,m+1}(-x) \quad (\text{D35a})$$

$$= 2 \sum_{i=2,4,\dots}^{m+c-1} p_{m+1,i} x^i. \quad (\text{D35b})$$

Due to $P_{\min,m+1}(x)$ being a minimal polynomial, by definition, there are three possibilities for $p_{m+1,m+c-1}$. They are

$$0 < 2|p_{m+1,m+c-1}| < \frac{1}{(m+c-1)!}, \quad (\text{D36a})$$

$$2|p_{m+1,m+c-1}| = \frac{1}{(m+c-1)!}, \quad (\text{D36b})$$

$$p_{m+1,m+c-1} = 0. \quad (\text{D36c})$$

For the first possibility [Eq. (D36a)], the leading coefficient on the right-hand side of Eq. (D35a) is not an integer multiple of $1/(m+c-1)!$ and thus we can again use Corollary D1 to find a contradiction.

For the second possibility [Eq. (D36b)], the right-hand side of Eq. (D35a) can be reduced using the integer-valued polynomial $L_{m+c-1}(x)$, which yields

$$P_{\min,m}(x) \equiv 2 \sum_{i=2,4,\dots}^{m+c-3} p_{m+1,i} x^i - \sum_{i=1}^{m+c-2} \ell_{m+c-1,i} x^i. \quad (\text{D37})$$

However, the leading coefficient $\ell_{m+c-1,m+c-2}$ on the right-hand side is not an integer multiple of $1/(m+c-2)!$ due to Eq. (D29). Therefore, we can again use Corollary D1 to find a contradiction.

For the third possibility [Eq. (D36c)], we apply Eq. (D16) together with Eq. (D33b). This yields

$$0 \equiv P_{\min,m+1}(x+2) - P_{\min,m+1}(x) \quad (\text{D38a})$$

$$= \pm \frac{x^{m+c-1}}{(m+c-1)!} \pm \frac{x^{m+c-2}}{(m+c-2)!} + O(x^{m+c-3}) \quad (\text{D38b})$$

$$\equiv \mp \ell_{m+c-1,m+c-2} x^{m+c-2} + O(x^{m+c-3}), \quad (\text{D38c})$$

where on the second line we have used that $|p_{m+1,m+c}| = 1/(2(m+c)!)$ due to Eq. (D16), and to get to the third line we have subtracted the polynomials $L_{m+c-1}(x)$ and $L_{m+c-2}(x)$, respectively, to eliminate the leading terms. The resulting leading coefficient on the final line is not an integer multiple of $1/(m+c-2)!$ due to Eq. (D29). Hence, we can use Lemma D1 to find a contradiction.

In summary, c cannot be in the set $\{3, 5, \dots\}$; the only remaining possibility is $c = 1$. In this case, the right-hand side of Eq. (D35a) matches the degree of the left-hand side, and thus we cannot use Corollary D1. Since we know from Appendix D1 that $P_{\min,m+1}(x)$ exists, we conclude that if m is even, then $\deg(P_{\min,m+1}(x)) = m+1$.

Case 2: m is an odd number. The proof for the case where m is odd follows a similar structure to that of the even case. However, the order of the arguments is reversed: We first show that $c \notin \{3, 5, \dots\}$, and then show that $c \notin \{2, 4, \dots\}$. The logic of each step remains analogous, with appropriate adjustments to account for the parity of m . ■

Finally, Lemmas D4 and D2 trivially combine to give the main theorem (Theorem D1), as promised.

APPENDIX E: REPRESENTATION OF MULTIQUBIT POLYNOMIAL PHASE GATES

In this Appendix, we present a method for constructing minimal representations of multiqubit gates in the square GKP code using multimode polynomial phase stabilizers. These

results extend the single-qubit polynomial-stabilizer formalism to the multimode setting, highlighting its generality. We will begin by outlining how to construct a multimode polynomial phase gate that can implement a given logical diagonal multiqubit gate. In particular, we focus on the multicontrolled phase gate $C^{N-1}\Lambda_m$ acting on N qubits, which applies a phase $\exp(i2\pi/2^m)$ to the logical state $|\overline{1 \cdots 1}\rangle$ while leaving all other logical computational basis states unchanged. Any diagonal logical multimode gate can then be generated by a product of $C^{N-1}\Lambda_m$ gates. After this, we will provide a characterization of the multimode polynomial phase stabilizers (which correspond to integer-valued polynomials on multiple variables) and describe the multimode coefficient reduction procedure that returns a minimal representation of a given multimode polynomial phase gate.

We begin by constructing a multimode polynomial phase gate that implements the logical $C^{N-1}\Lambda_m$ gate. As in Appendix D, we start with the base case $C^{N-1}\Lambda_1$ and then recursively build higher-order gates $C^{N-1}\Lambda_m$ from it.

We first claim that the $C^{N-1}\Lambda_1$ gate can be implemented by the polynomial

$$P[C^{N-1}\Lambda_1](\mathbf{x}) = \frac{x_1 x_2 \cdots x_N}{2}, \quad (\text{E1})$$

where $\mathbf{x} = (x_1, \dots, x_n)$. To see this, we decompose each variable x_i as

$$x_i = 2n_i + r_i, \quad (\text{E2})$$

with $n_i \in \mathbb{Z}$ and $r_i \in \{0, 1\}$, indicating whether x_i is even ($r_i = 0$) or odd ($r_i = 1$). Substituting into $P[C^{N-1}\Lambda_1](\mathbf{x})$ gives

$$P[C^{N-1}\Lambda_1](\mathbf{x}) \bmod 1 = \frac{r_1 r_2 \cdots r_N}{2}, \quad (\text{E3})$$

which implements the Λ_1 gate precisely when all x_i are odd, i.e., when all encoded qubits are in the logical state $|\overline{1}\rangle$.

Next, we construct a (nonoptimized) representation of the logical $C^{N-1}\Lambda_{m+1}$ gate from $P[C^{N-1}\Lambda_m]$. The key is to reuse the recursion formula from Eq. (D6), suitably generalized to multiqubit gates. In this setting, we require that each representation $P[C^{N-1}\Lambda_m](\mathbf{x})$ gate must be able to be decomposed as

$$P[C^{N-1}\Lambda_m](\mathbf{x}) = n(\mathbf{x}) + \frac{a(\mathbf{x})}{2^m}, \quad (\text{E4})$$

where $n(\mathbf{x})$ is again some integer depending on $\mathbf{x}$ but now the parity term satisfies

$$a(\mathbf{x}) = \begin{cases} 1, & \text{if } x_1 = \cdots = x_N = 1, \\ 0, & \text{otherwise.} \end{cases} \quad (\text{E5})$$

Consequently, we obtain the general recursive formula for the $C^{N-1}\Lambda_m$ gate:

$$P[C^{N-1}\Lambda_{m+1}](\mathbf{x}) = \frac{(x_1 \cdots x_N)^{2^m}}{2^{m+1}}. \quad (\text{E6})$$

The proof of Eq. (E6) follows the same steps as in Appendix D 1.

Next, we need to identify a complete basis for the integer-valued polynomials on N variables. Such a basis is given by the set

$$\{L_{i_1}(x_1)L_{i_2}(x_2)\cdots L_{i_n}(x_n)\}. \quad (\text{E7})$$

To see why this is indeed a complete basis, note that we already know this is true for $N = 1$, so preparing for an inductive argument let us assume it is true for $N = n$. Now suppose we have an integer-valued polynomial $P(\mathbf{x})$ on $N = n+1$ variables; that is, it satisfies $P(\mathbb{Z}^{n+1}) \subseteq \mathbb{Z}$. Viewing $P(\mathbf{x})$ as a univariate integer-valued polynomial of the variable x_{n+1} , we can always write it as

$$P(\mathbf{x}) = \sum_{i_{n+1}} c_{i_{n+1}}(x_1, \dots, x_n) L_{i_{n+1}}(x_{n+1}^{n+1}), \quad (\text{E8})$$

with $c_{i_{n+1}}(x_1, \dots, x_n)$ being an integer-valued polynomial using the completeness of the univariate integer-valued polynomial basis [Eq. (43)].

Note, by our inductive assumption, the integer-valued polynomial $c_{i_{n+1}}(x_1, \dots, x_n)$ can be decomposed into

$$c_{i_{n+1}}(x_1, \dots, x_n) = \sum_{i_1, \dots, i_n} c_{i_1, \dots, i_n} L_{i_1}(x_1) \cdots L_{i_n}(x_n), \quad (\text{E9})$$

with the coefficient $c_{i_1, \dots, i_n}$ being integers. Therefore, by substituting Eq. (E9) back to Eq. (E8), we find that $P(\mathbf{x})$ can always be written as

$$P(\mathbf{x}) = \sum_{i_1, \dots, i_{n+1}} c_i (L_{i_1}(x_1) \cdots L_{i_{n+1}}(x_{n+1})), \quad (\text{E10})$$

for some integer coefficients $c_i \in \mathbb{Z}$. This shows that Eq. (E7) is indeed a complete basis of the multivariate integer-valued polynomials.

Finally, given a polynomial phase gate $P[C^{N-1}\Lambda_m](\mathbf{x})$, we minimize the representation using a multivariate coefficient reduction procedure as follows. In particular, the coefficient of each monomial $x_1^{d_1} \cdots x_N^{d_N}$ in $P[C^{N-1}\Lambda_m](\mathbf{x})$ will be reduced by adding or subtracting a suitable integer multiple of the integer-valued polynomial $L_{d_1}(x_1) \cdots L_{d_N}(x_N)$. However, defining the order in which these coefficients are reduced is crucial for the coefficient reduction procedure to work correctly.

In particular, for any monomial $x_1^{d_1} \cdots x_N^{d_N}$ that appears in the polynomial $P[C^{N-1}\Lambda_m]$, we define its degree as

$$\deg(x_1^{d_1} \cdots x_N^{d_N}) = d_1 + d_2 + \cdots + d_N. \quad (\text{E11})$$

The degree of the polynomial $P[C^{N-1}\Lambda_m](\mathbf{x})$ is simply the maximal degree of its terms. The key difference in this multivariate coefficient reduction procedure is that, for any given degree D of the polynomial $F_m[C^{N-1}\Lambda_m](\mathbf{x})$, there can be multiple monomials of the form

$$x_1^{d_1} \cdots x_N^{d_N} \quad (\text{E12})$$

such that

$$d_1 + \cdots + d_N = D. \quad (\text{E13})$$

The crucial observation is that the coefficients of any two monomials with the same degree can be independently reduced, without the reduction of one of the monomials affecting the reduction of the other. This follows from the fact that the polynomial $L_{d_1}(x_1) \cdots L_{d_N}(x_N)$ only contains monomials $x_1^{m_1} \cdots x_N^{m_N}$, where every m_n satisfies $m_n \leq d_n$. This greatly simplifies the coefficient reduction procedure: we simply begin by reducing the monomials in $P[C^{N-1}\Lambda_m]$ that have the highest degree (in any order), before reducing the monomials

with the next highest degree (in any order), and so on until we have minimized all the coefficients in the polynomial. The output $P_{\min}[C^{N-1}\Lambda_m](\mathbf{x})$ will once again be minimal under a suitably generalized version of the lexicographic order defined in Sec. IV.

As an example, consider the controlled- S gate, $CS \equiv C^1\Lambda_2$. From Eq. (E6), the recursive formula gives

$$P[CS](x_1, x_2) = \frac{x_1^2 x_2^2}{4}. \quad (\text{E14})$$

This representation, however, is reducible by the stabilizer $L_2(x_1)L_2(x_2)$. Subtracting this term yields the optimized form

$$P_{\min}[CS](x_1, x_2) = -\frac{x_1^2 x_2^2 + x_1 x_2^2 + x_1 x_2}{4}. \quad (\text{E15})$$

This irreducible representation achieves the minimal possible coefficients and interaction order, hence, constitutes a minimal representation for the CS gate. The representation for the $CCZ \equiv C^2\Lambda_1$ gate given in Eq. (62b) can be derived in the same way.

APPENDIX F: STATISTICAL MOMENT OF PROPAGATED ERRORS

In this Appendix, we present a quantitative analysis of the propagated error operator $\mathcal{E}(\cdot) = E \cdot E^\dagger$ induced by general polynomial phase gates $\exp[i2\pi P(q/\sqrt{\pi})]$, filling in the gaps in our derivation in Sec. V. As a reminder, we define

$$\Lambda = \exp[i2\pi P(q/\sqrt{\pi})], \quad P(x) = \sum_{j=1}^n a_j x^j, \quad (\text{64})$$

$$E = \Lambda \text{Env}_{\Delta_q, \Delta_p} \Lambda^\dagger, \quad \mathcal{E}(\cdot) = E \cdot E^\dagger, \quad (\text{65})$$

and we will derive Eqs. (66) and (67), which are asymptotic expressions for the covariance and mean of the probability density function $p_{\mathcal{E}(\mathbf{v})}$ corresponding to the characteristic function of the twirled channel $\text{Twirl}[\mathcal{E}]$ [Eq. (22)].

In Appendix F1, we perform the twirling approximation and write down an explicit expression for $p_{\mathcal{E}(\mathbf{v})}$. To characterize the noise induced by the $\text{Twirl}[\mathcal{E}]$, we compute the first and second statistical moments—the mean displacement and covariance matrix—of $p_{\mathcal{E}(\mathbf{v})}$, as detailed in Appendix F2. These quantities serve as proxies for the gate infidelity, since the exact error rate is generally analytically intractable. In Appendix F3, we derive our main result: the asymptotic scaling $\Delta \rightarrow 0$ of these statistical moments in terms of both the polynomial degree defining the logical gate and its coefficients. This analysis confirms our intuition that the highest-order interaction term sets the fundamental limit on gate fidelity and provides a quantitative justification for the lexicographic ordering introduced in the main text. Finally, in Appendix F4, we analyze the scaling of the first and second moments of $p_{\mathcal{E}(\mathbf{v})}$ at finite Δ and show that minimal polynomial phase gates $\Lambda_n[q/\sqrt{\pi}]$ with high degree n can have lower gate infidelity compared to the minimal cubic phase gate, providing a theoretical explanation for the results in Fig. 4.

1. Twirling approximation of E

Throughout this Appendix, we employ three approximations to simplify the analytical calculation. First, we approximate the characteristic function of the biased envelope operator using the Taylor approximation $\tanh(x) \approx x$ given in Eq. (25b). Second, we assume that the GKP state is strongly squeezed in the position quadrature so that $v_q \propto O(\Delta_q)$ and make the approximation

$$P_n(x + \sqrt{2}v_q) - P_n(x) \approx \sqrt{2}v_q \frac{dP_n(x)}{dx} \equiv \sqrt{2}v_q P'_n(x). \quad (\text{F1})$$

The third approximation is used in Appendix F3, where we estimate the optimal asymmetry λ_{opt} by minimizing the propagated shear error, following the same approach as in Sec. V.

While we could proceed without resorting to this approximation, it significantly simplifies the algebra and still provides us the dominant contribution in the error operator. Under these two approximations, the characteristic function of the propagated error E is given by

$$\chi_E(\mathbf{v}) = \frac{\mathcal{N}(0, \Delta_q^2, v_q)}{\sqrt{2\pi}} \int dx e^{-i\sqrt{2\pi}v_p x} e^{-\frac{\Delta_p^2 x^2}{2}} e^{i2\sqrt{2\pi}v_q P'_n(\frac{x}{\sqrt{\pi}})}, \quad (\text{F2})$$

which we derive more explicitly in Eq. (G20).

Performing the twirling approximation, the error channel $\mathcal{E}$ becomes a random displacement channel

$$\text{Twirl}[\mathcal{E}](\rho) = \int d\mathbf{v} \frac{(\mathcal{N}[0, \Delta_q^2, v_q])^2 f(v_p|v_q)}{2\pi} W(\mathbf{v}) \rho W(\mathbf{v})^\dagger, \quad (\text{F3})$$

where we have denoted

$$f(v_p|v_q) = \left| \int dx e^{-i\sqrt{2\pi}v_p x} e^{-\frac{\Delta_p^2 x^2}{2}} e^{i2\sqrt{2\pi}v_q P'_n(\frac{x}{\sqrt{\pi}})} \right|^2. \quad (\text{F4})$$

As explained in the main text, we can interpret the integrand

$$\frac{(\mathcal{N}[0, \Delta_q^2, v_q])^2 f(v_p|v_q)}{2\pi} \quad (\text{F5})$$

as an *unnormalized* probability density function $p_E(\mathbf{v})$ of the displacement error $\mathbf{v}$ on the data qubit. In the following, we will find the normalization constant of this twirled channel $\text{Twirl}[\mathcal{E}]$, which is defined as

$$\int d\mathbf{v} \frac{(\mathcal{N}[0, \Delta_q^2, v_q])^2 f(v_p|v_q)}{2\pi}. \quad (\text{F6})$$

Our first step is to rewrite $f(v_p|v_q)$ in a more illuminating form

$$f(v_p|v_q) = \left| \int dx e^{-i\sqrt{2\pi}v_p x} e^{-\frac{\Delta_p^2 x^2}{2}} e^{i2\sqrt{2\pi}v_q P'_n(\frac{x}{\sqrt{\pi}})} \right|^2 \quad (\text{F7a})$$

$$= \int dx_1 e^{-i\sqrt{2\pi}v_p x_1} e^{-\frac{\Delta_p^2 x_1^2}{2}} e^{i2\sqrt{2\pi}v_q P'_n(\frac{x_1}{\sqrt{\pi}})} \times \int dx_2 e^{i\sqrt{2\pi}v_p x_2} e^{-\frac{\Delta_p^2 x_2^2}{2}} e^{-i2\sqrt{2\pi}v_q P'_n(\frac{x_2}{\sqrt{\pi}})} \quad (\text{F7b})$$

$$= \iint dx_+ dx_- e^{i2\sqrt{2\pi}v_q (P'_n(\frac{x_+ + x_-}{\sqrt{2}}) - P'_n(\frac{x_+ - x_-}{\sqrt{2}}))} \times e^{-\frac{\Delta_p^2}{2}(x_+^2 + x_-^2)} e^{-i2\sqrt{\pi}v_p x_-}, \quad (\text{F7c})$$

where on the third equality, we have performed a transformation of variable with unit Jacobian

$$x_+ = \frac{x_1 + x_2}{\sqrt{2}}, \quad x_- = \frac{x_1 - x_2}{\sqrt{2}}. \quad (\text{F8})$$

For clarity, we denote

$$[\delta P'_n](x, y) = P'_n\left(\frac{x+y}{\sqrt{2}}\right) - P'_n\left(\frac{x-y}{\sqrt{2}}\right). \quad (\text{F9})$$

The integral over x_+ and x_- cannot be evaluated analytically. However, the integral in Eq. (F6) can be computed exactly by changing the order of integration. Specifically, we first integrate over v_p , which yields a Dirac-delta function $\delta(2\sqrt{\pi}x_-)$. We then successively integrate over x_- , x_+ , and finally v_q . After some straightforward but tedious algebra, the normalization constant is found to be

$$\int d\mathbf{v} \frac{(\mathcal{N}[0, \Delta_q^2, v_q])^2 f(v_p|v_q)}{2\pi} = \frac{1}{2\Delta_q \Delta_p}. \quad (\text{F10})$$

Dividing Eq. (F3) by this normalization factor yields the normalized error channel $\text{Twirl}[\tilde{\mathcal{E}}]$:

$$\begin{aligned} \text{Twirl}[\tilde{\mathcal{E}}](\rho) &= (2\Delta_q \Delta_p) \int d\mathbf{v} \frac{(\mathcal{N}[0, \Delta_q^2, v_q])^2 f(v_p|v_q)}{2\pi} \\ &\quad \times W(\mathbf{v}) \rho W(\mathbf{v})^\dagger \end{aligned} \quad (\text{F11a})$$

$$\begin{aligned} &= \int d\mathbf{v} [\sqrt{2}\Delta_q (\mathcal{N}[0, \Delta_q^2, v_q])^2] \left(\Delta_p \frac{f(v_p|v_q)}{\sqrt{2\pi}} \right) \\ &\quad \times W(\mathbf{v}) \rho W(\mathbf{v})^\dagger \end{aligned} \quad (\text{F11b})$$

$$\begin{aligned} &= \int d\mathbf{v} \mathcal{N}\left[0, \frac{\Delta_q^2}{2}, v_q\right] \left[\frac{1}{\pi} \frac{\Delta_p}{\sqrt{2}} f(v_p|v_q) \right] \\ &\quad \times W(\mathbf{v}) \rho W(\mathbf{v})^\dagger. \end{aligned} \quad (\text{F11c})$$

In the transformation from the second to the third equality, we have absorbed the factor $\sqrt{2}\Delta_q$ into the squared normal distribution in v_q , resulting in a single Gaussian with variance $\Delta_q^2/2$.

2. Conditional statistical moments of the twirled error operator

In the previous section, we renormalized the twirled channel $\text{Twirl}[\mathcal{E}]$ such that $p(\mathbf{v})$ is a normalized probability distribution function. In this Appendix, we describe how to compute the statistical moments of this probability distribution function. Of particular interest are the first and second statistical moments, which characterize the average and the spread of the displacement error in the twirled channel $\text{Twirl}[\tilde{\mathcal{E}}]$, respectively. Since $f(v_q|v_p)$ cannot be computed analytically, we cannot determine the exact error rate of the random displacement channel. Instead, we use these two moments as proxies for the error rate. If the mean is far away from the origin or the covariance matrix has large eigenvalues, we expect that the average gate fidelity will be low because the probability of sampling an uncorrectable displacement error [Eq. (12)] is high.

As we have explained previously, we can view $\frac{\Delta_p}{\sqrt{2}} f(v_p|v_q)$ as a conditional distribution of v_p conditioned on the displace-

ment error in position v_q . Therefore, we will first compute the m th conditional statistical moment $\mathbb{E}(v_p^m|v_q)$ defined as

$$\mathbb{E}(v_p^m|v_q) = \int dv_p \frac{v_p^m}{\pi} \frac{\Delta_p}{\sqrt{2}} f(v_p|v_q) \quad (\text{F12a})$$

$$\begin{aligned} &= \int dx_+ \mathcal{N}\left(0, \frac{2}{\Delta_p^2}, x_+\right) \int dx_- \left(\frac{i}{2\pi}\right)^m \delta^{(m)}(x_-) \\ &\quad \times e^{i2\sqrt{2}\pi v_q \delta P'_n(x_+, x_-)} e^{-\pi \frac{\Delta_p^2}{2} x_-^2}. \end{aligned} \quad (\text{F12b})$$

From the first line to the second line of Eq. (F12), we have simply performed the integral over v_p and transformed the dummy variables everything in term of the variables

$$x_{+, \text{new}} = \frac{x_{+, \text{old}}}{\sqrt{\pi}}, \quad x_{-, \text{new}} = \frac{x_{-, \text{old}}}{\sqrt{\pi}}. \quad (\text{F13})$$

Here, $\delta^{(m)}(x)$ is the m th-order derivative of the Dirac-delta function, which is defined to have the property

$$\int dx \delta^{(m)}(x) g(x) = (-1)^m g^{(m)}(0). \quad (\text{F14})$$

For brevity, we denote $\Phi(x_+, x_-, v_q)$ to be the exponent that appears inside the integral over x_- in Eq. (F12b):

$$\Phi(x_+, x_-, v_q) = i2\sqrt{2}\pi v_q [\delta P'_n](x_+, x_-) - \frac{\pi \Delta_p^2}{2} x_-^2, \quad (\text{F15})$$

such that $\mathbb{E}(v_p^m|v_q)$ is written as

$$\mathbb{E}(v_p^m|v_q) = \int dx_+ \mathcal{N}\left(0, \frac{2}{\Delta_p^2}, x_+\right) \quad (\text{F16a})$$

$$\times \int dx_- \left(\frac{i}{2\pi}\right)^m \delta^{(m)}(x_-) e^{i\Phi(x_+, x_-, v_q)}. \quad (\text{F16b})$$

Due to the definition of the polynomial $\delta P'_n(x)$, we find that

$$\Phi(x_+, 0, v_q) = 0, \quad \forall x_+, v_q \in \mathbb{R}. \quad (\text{F17})$$

Using this property, the integral over x_- in Eq. (F16) can be evaluated using Faà di Bruno's formula [52]

$$\begin{aligned} &\int dx_- \delta^{(m)}(x_-) e^{\Phi(x_+, x_-, v_q)} \\ &= (-1)^m B_m[\partial_{x_-}^{(1)} \Phi(x_+, 0, v_q), \dots, \partial_{x_-}^{(m)} \Phi(x_+, 0, v_q)], \end{aligned} \quad (\text{F18})$$

where $B_m(x_1, \dots, x_n)$ is the m th complete exponential Bell polynomial

$$B_m(x_1, \dots, x_n) = (m!) \sum_{1j_1 + \dots + mj_m = m} \prod_{k=1}^m \frac{x_k^{j_k}}{(k!)^{j_k} (j_k)!}. \quad (\text{F19})$$

Consequently, the analytical formula for the conditional statistical moments $\mathbb{E}(v_p^m|v_q)$ is given by

$$\begin{aligned} \mathbb{E}(v_p^m|v_q) &= \int dx_+ \mathcal{N}\left(0, \frac{2}{\Delta_p^2}, x_+\right) \left(\frac{i}{2\pi}\right)^m (-1)^m \\ &\quad \times B_m[\partial_{x_-}^{(1)} \Phi(x_+, 0, v_q), \dots, \partial_{x_-}^{(m)} \Phi(x_+, 0, v_q)]. \end{aligned} \quad (\text{F20})$$

From the general formula [Eq. (F20)] for $\mathbb{E}(v_p^m|v_q)$, it is then a straightforward calculation to show that the first conditional moment $\mathbb{E}(v_p|v_q)$ for a general polynomial $P_n(x)$ with coefficients $\{a_i\}$ is given by

$$\begin{aligned}\mathbb{E}(v_p|v_q) &= \sqrt{2}v_q \sum_{k=2}^n a_k \beta_k \int dx_+ \mathcal{N}\left(0, \frac{2}{\Delta_p^2}, x_+\right) x_+^{k-2} \\ &= \sqrt{2}v_q \sum_{k=2,4,\dots}^n a_k \beta_k \mathbb{E}(x_+^{k-2}),\end{aligned}\quad (\text{F21})$$

where the coefficients β_k are given by

$$\beta_k = \frac{2k(k-1)}{\sqrt{2}^{k-1}}, \quad (\text{F22})$$

and we define the expectation value $\mathbb{E}[g(x_+)]$ of a function $g(x_+)$ as

$$\mathbb{E}(g(x_+)) = \int dx_+ \mathcal{N}\left(0, \frac{2}{\Delta_p^2}, x_+\right) g(x_+). \quad (\text{F23})$$

Note that only even values of k contribute to Eq. (F21) since $\mathbb{E}(x_+^k)$ vanishes for odd k due to the symmetry of the Gaussian distribution. We have seen this previously in Eq. (G21) where the third-order coefficient a_3 gives correction to $\mathbb{E}(v_p|v_q)$ to the second order $O(v_q^2)$. Similarly, we find the second conditional moment $\mathbb{E}(v_p^2|v_q)$ to be

$$\mathbb{E}(v_p^2|v_q) = \frac{\Delta_p^2}{4\pi} + 2v_q^2 \mathbb{E}\left[\left(\sum_{k=2}^n a_k \beta_k x_+^{k-2}\right)^2\right]. \quad (\text{F24})$$

Given the formula for the conditional moments, we can obtain the formulas for the unconditional first and second moments, which are presented in the next section.

3. Asymptotic scaling of the first and second unconditional moments

In the previous section, we derived an expression for the conditional moment $\mathbb{E}(v_p^m|v_q)$. In this section, we use this result to determine the asymptotic scaling behavior of the mean vector

$$\begin{pmatrix} \mathbb{E}(v_q) \\ \mathbb{E}(v_p) \end{pmatrix} \quad (\text{F25})$$

and covariance matrix

$$\begin{pmatrix} \mathbb{E}(v_q^2) & \mathbb{E}(v_q v_p) \\ \mathbb{E}(v_q v_p) & \mathbb{E}(v_p^2) \end{pmatrix}. \quad (\text{F26})$$

As we already discussed in the main text, the covariance matrix can be used as a proxy for the average gate fidelity. Using this, we can obtain an asymptotic estimate of the optimal asymmetry λ_{opt} by minimizing the elements of covariance matrix.

By definition, the unconditional moment $\mathbb{E}(v_p^m)$ is obtained from the conditional moment $\mathbb{E}(v_p^m|v_q)$ by averaging over the distribution of position displacement errors v_q , i.e.,

$$\mathbb{E}(v_p) = \int dv_q \mathcal{N}\left[0, \frac{\Delta_q^2}{2}, v_q\right] \mathbb{E}(v_p|v_q) = 0, \quad (\text{F27a})$$

$$\mathbb{E}(v_p^2) = \int dv_q \mathcal{N}\left[0, \frac{\Delta_q^2}{2}, v_q\right] \mathbb{E}(v_p^2|v_q) \quad (\text{F27b})$$

$$= \frac{\Delta_p^2}{4\pi} + \frac{\Delta_q^2}{2\pi} \mathbb{E}\left[\left(\sum_{k=2}^n a_k \beta_k x_+^{k-2}\right)^2\right]. \quad (\text{F27c})$$

To find the first and second moments $\mathbb{E}(v_q)$ and $\mathbb{E}(v_q^2)$, we note that the probability that a displacement error v_q happens is independent of v_p . Therefore, from Eq. (F11), we find that

$$\mathbb{E}(v_q) = 0, \quad \mathbb{E}(v_q^2) = \frac{\Delta_q^2}{4\pi}. \quad (\text{F28})$$

Finally, the cross-correlation term can be computed by combining Eqs. (F21) and (F28):

$$\mathbb{E}(v_q v_p) = \int dv_q \mathcal{N}\left(0, \frac{\Delta_q^2}{2}, v_q\right) v_q \mathbb{E}(v_p|v_q) \quad (\text{F29a})$$

$$= \sqrt{2} \mathbb{E}(v_q^2) \sum_{k=2,4,\dots}^n a_k \beta_k \mathbb{E}(x_+^{k-2}). \quad (\text{F29b})$$

$$= \frac{\Delta_q^2}{2\sqrt{2}\pi} \sum_{k=2,4,\dots}^n a_k \beta_k \mathbb{E}(x_+^{k-2}). \quad (\text{F29c})$$

We can obtain an explicit formula for Eqs. (F27c) and (F29c) by using the known formula for the statistical moments of a Gaussian distribution

$$\mathbb{E}(x_+^k) = \int dx_+ \mathcal{N}\left(0, \frac{2}{\Delta_p^2}, x_+\right) x_+^k \quad (\text{F30a})$$

$$= 2^{-1+\frac{k}{2}} [1+(-1)^k] \pi^{-\frac{(1+k)}{2}} \Gamma\left[\frac{1+k}{2}\right] \Delta_p^{-k}. \quad (\text{F30b})$$

In the limit of infinitely squeezed GKP states, the leading contribution to $\mathbb{E}(v_p^2)$ comes from the leading coefficient a_n of the polynomial $P_n(x)$. As a result, the second moment $\mathbb{E}(v_p^2)$ acquires a simple form in this limit:

$$\mathbb{E}(v_p^2) \approx \frac{\Delta_p^2}{4\pi} + a_n^2 \beta_n^2 \frac{2^{n-3} \pi^{1/2-n} \Gamma\left[n - \frac{3}{2}\right] \frac{\Delta_q^2}{\Delta_p^2}}{\Delta_p^{2n-6} \frac{\Delta_q^2}{\Delta_p^2}}, \quad (\text{F31a})$$

$$= \frac{\Delta_p^2}{4\pi} + O\left(\frac{\Delta_q^2}{\Delta_p^{2n-4}}\right). \quad (\text{F31b})$$

In a similar fashion, we retain the leading contribution in $\mathbb{E}(v_q v_p)$. However, depending on whether the degree n of P_n is even or odd, the leading contribution can either come from the leading coefficient or the second-leading coefficient. If we assume that the degree of $P_n(x)$ is even, the cross-correlation is given by

$$\mathbb{E}(v_q v_p) = \frac{\Delta_q^2}{2\sqrt{2}\pi} a_n \beta_n 2^{-2+\frac{n}{2}} \pi^{-\frac{(n-1)}{2}} \Gamma\left[\frac{n-1}{2}\right] \Delta_p^{-(n-2)} \quad (\text{F32})$$

$$= O\left(\frac{\Delta_q^2}{\Delta_p^{n-2}}\right). \quad (\text{F33})$$

Again, assuming that the degree $P_n(x)$ is odd, we find that

$$\mathbb{E}(v_q v_p) = O\left(\frac{\Delta_q^2}{\Delta_p^{n-3}} a_{n-1}\right) = O\left(\frac{\Delta_q^2}{\Delta_p^{n-2}}\right). \quad (\text{F34})$$

We assume that the GKP has asymmetry λ such that squeezing parameters are given by

$$\Delta_q = \frac{\Delta}{\sqrt{\lambda}}, \quad \Delta_p = \sqrt{\lambda} \Delta. \quad (\text{F35})$$

Since $\mathbb{E}(v_p^2)$ is the dominant error in the limit of infinitely squeezed state, our goal is to minimize it. The global minimum of $\mathbb{E}(v_p^2)$ is obtained at the optimal asymmetry value

$$\lambda_{\text{opt}} = \sqrt[n]{4\pi(n-1) \frac{a_n^2 \beta_n^2 2^{n-3} \pi^{1/2-n} \Gamma[n - \frac{3}{2}]}{\Delta^{2n-4}}} \quad (\text{F36a})$$

$$= O(\Delta^{-(2-4/n)}). \quad (\text{F36b})$$

Consequently, we can easily show that the second moments $\mathbb{E}(v_p^2)$ and $\mathbb{E}(v_q^2)$ evaluated at λ_{opt} scale as

$$\mathbb{E}(v_q^2) \propto \Delta^{4-4/n}, \quad \mathbb{E}(v_p^2) \propto \Delta^{4/n}, \quad (\text{F37a})$$

$$\mathbb{E}(v_q v_p) \propto \begin{cases} \Delta^2, & \text{for even } n, \\ \Delta^{2+\frac{2}{n}}, & \text{for odd } n, \end{cases} \quad (\text{F37b})$$

which vanish in the limit of infinitely squeezed GKP states. This result implies that, even for high-order polynomial phase gates, the propagated error induced by the non-Gaussian interaction can be mitigated by on-demand noise biasing.

4. Scaling of covariance matrices at finite squeezing

We have analyzed the asymptotic scaling of polynomial phase gates and shown that, in the limit of infinite squeezing, gates with lower-degree polynomials asymptotically achieve higher fidelities than gates with higher-degree polynomials. However, this asymptotic behavior does not imply that higher-degree polynomial phase gates perform worse for all finite values of Δ . To investigate this, we now analyze how the leading coefficient a_n of a polynomial phase gate affects its error scaling when applied to GKP states with finite Δ . In particular, we are interested in how the error in lexicographically minimal polynomial phase gates $\Lambda_n[q/\sqrt{\pi}]$ scales with increasing n .

We first note that the prefactor β_n scales as $\beta_n \propto O(2^{-n/2})$, which exactly cancels the exponential dependence on $2^{n/2}$ appearing in Eqs. (F31) and (F32). In other words,

$$\beta_n 2^{n/2} \propto O(1). \quad (\text{F38})$$

With this simplification, the second moments $\mathbb{E}(v_p^2)$ and $\mathbb{E}(v_q v_p)$ for finite Δ scale with n as

$$\mathbb{E}(v_p^2) \propto \frac{\Delta_p^2}{4\pi} + O\left(a_n^2 \frac{\Gamma[n - \frac{3}{2}]}{\pi^n} \frac{\Delta_q^2}{\Delta_p^{2n-4}}\right), \quad (\text{F39a})$$

$$\mathbb{E}(v_q v_p) \propto O\left(a_n \frac{\Gamma[\frac{n-1}{2}]}{\pi^{n/2}} \frac{\Delta_q^2}{\Delta_p^{n-2}}\right). \quad (\text{F39b})$$

The appearance of the Gamma functions in the numerators implies a factorial growth with n , suggesting that, for fixed Δ , nonoptimized higher-order polynomial phase gates could lead

to larger propagated errors. However, for minimal polynomial phase gates, the leading coefficients satisfy $a_n \propto O(1/n!)$ due to Theorem D1. Substituting this scaling, we obtain

$$a_n^2 \Gamma\left[n - \frac{3}{2}\right] \propto O\left(\frac{1}{n!}\right), \quad a_n \Gamma\left[\frac{n-1}{2}\right] \propto O\left(\frac{1}{n!}\right). \quad (\text{F40})$$

As a result, for lexicographically minimal polynomial phase gates, the second moments $\mathbb{E}(v_p^2)$ and $\mathbb{E}(v_q v_p)$ would scale with n as

$$\mathbb{E}(v_p^2) \propto \frac{\Delta_p^2}{4\pi} + O\left(\frac{1}{\pi^n n!} \frac{\Delta_q^2}{\Delta_p^{2n-4}}\right), \quad (\text{F41a})$$

$$\mathbb{E}(v_q v_p) \propto O\left(\frac{1}{\pi^{n/2} n!} \frac{\Delta_q^2}{\Delta_p^{n-2}}\right). \quad (\text{F41b})$$

The factorial suppression in n indicates that, for finite but fixed squeezing Δ , higher-order minimal polynomial phase gates can in principle outperform lower-order ones such as the cubic phase gate.

This analytical scaling is consistent with our numerical simulations shown in Fig. 4, where the $T^{1/8}$ gates have higher fidelity than the T gates in the simulated parameter range. Nevertheless, we emphasize that this analysis neglects experimental challenges associated with realizing high-order polynomial phase gates, such as their nonlinearity requirements and potential control errors. Thus, our results should be viewed as a proof of principle that establishes the theoretical potential for improved performance, rather than a practical prescription for near-term implementations.

APPENDIX G: PROOF OF FAULT-TOLERANCE THEOREM

In this Appendix, we prove that the minimized T_3 gate with on-demand noise biasing is fault tolerant. Our strategy is to show that, for any given squeezing parameter Δ , there exists an asymmetry value $\lambda(\Delta)$ such that the corresponding error operator $E(\Delta, \lambda(\Delta))$, defined the same as in the main text by

$$E(\Delta, \lambda(\Delta)) = \Lambda \text{Env}_{\Delta/\lambda(\Delta), \Delta\lambda(\Delta)} \Lambda^\dagger, \quad (\text{G3})$$

has logical error rate going to zero in the limit $\Delta \rightarrow 0$.

Specifically, we consider the following circuit:

$$|\bar{\psi}_{\text{in}}\rangle \longrightarrow \boxed{E(\Delta, \lambda(\Delta))} \longrightarrow \boxed{\text{Ideal QEC}} \longrightarrow \bar{\rho}_{\text{out}} \quad (\text{G1})$$

Because the input and output of the circuit are contained within the ideal GKP codespace, we can define a logical channel $\mathcal{E}$ that acts from qubits to qubits defined such that $\rho_{\text{out}} = \mathcal{E}(|\psi_{\text{in}}\rangle\langle\psi_{\text{in}}|)$, which implicitly depends on the squeezing parameter Δ and the asymmetry λ . Here, $\mathcal{E}$ differs slightly from the logical channel defined in Eq. (71), because we assume that the QEC round following the T gate is ideal. This assumption allows us to isolate the contribution of errors arising solely from the T_3 gate, independent of those originating from noisy QEC round. The proof can be straightforwardly extended to include noisy QEC rounds if desired because the noise arising from the noisy QEC round also goes to zero as $\Delta \rightarrow 0$.

We define the average logical fidelity of the channel $\mathcal{E}$ gate, which also defines the “logical” average gate fidelity of the T_3

gate, as

$$\bar{F}(\Delta, \lambda) = \int d\psi \langle \psi | \mathcal{E}(|\psi\rangle\langle\psi|) | \psi \rangle. \quad (\text{G2})$$

To understand why $\bar{F}(\Delta, \lambda)$ characterizes the gate fidelity of the T_3 gate, recall that the operator E represents the effect of propagating the envelope operator through the T_3 gate. The ideal logical action of E is therefore the identity. Consequently, the average gate fidelity of E with respect to the identity directly corresponds to the logical gate fidelity of the T_3 operation.

We now prove our fault-tolerance theorem.

Theorem G1. Let $\varepsilon_{\text{target}}$ be a desired logical average gate infidelity for the T_3 gate. Then, there exists a threshold parameter Δ_{th} such that

$$\inf_{\lambda > 0} [1 - \bar{F}(\Delta, \lambda)] \leq \varepsilon_{\text{target}} \quad (\text{G3})$$

for all $\Delta \leq \Delta_{\text{th}}$, where $\bar{F}(\Delta, \lambda)$ is defined in Eq. (G2).

Proof. The proof proceeds in three main steps. First, we use the characteristic function $\chi_E(\mathbf{v})$ of the propagated error operator E [defined in Eq. (63)] to write down an exact expression for the average fidelity $\bar{F}(\Delta, \lambda)$. Second, we derive a lower bound $\bar{F}(\Delta, \lambda)_{\text{l.b.}}$ for this average fidelity $\bar{F}(\Delta, \lambda)$ that simplifies the expression found in the first step. Third, we show that for each Δ there exists a biasing $\lambda(\Delta)$ such that $\bar{F}(\Delta, \lambda(\Delta))_{\text{l.b.}} \rightarrow 1$ as $\Delta \rightarrow 0$. This proves that $\inf_{\lambda > 0} \bar{F}(\Delta, \lambda) \rightarrow 1$ as $\Delta \rightarrow 0$, which is equivalent to the statement in Theorem G1 via the ϵ - δ definition of a limit.

Step 1: An exact expression for $\bar{F}(\Delta, \lambda)$. As a starting point, the propagated error operator E of T_3 is given exactly by Eq. (65c), which we write here as

$$E(\Delta, \lambda) = \int d\mathbf{v} \mathcal{N} \left(2 \tanh \left(\frac{\Delta^2}{2} \right) \text{Diag}(\lambda^{-1}, \lambda), \mathbf{v} \right) \times W(\mathbf{v}) e^{i2\pi [P_{\min,3}(q/\sqrt{\pi} + \sqrt{2}v_q) - P_{\min,3}(q/\sqrt{\pi})]}, \quad (\text{G4})$$

where $P_{\min,3}(x)$ is the minimal polynomial that implements the T gate defined in Eq. (41).

Now, our next step will be to evaluate the so-called χ -matrix of the logical channel $\mathcal{E}$, which is defined such that

$$\mathcal{E}(\rho) = \sum_{\mu, \mu'=0}^3 \chi_{\mathcal{E}}^{\sigma_{\mu}, \sigma_{\mu'}} \sigma_{\mu} \rho \sigma_{\mu'}, \quad (\text{G5})$$

where as a reminder the logical channel $\mathcal{E}$ is a qubit-to-qubit channel, and where $\sigma_0, \sigma_1, \sigma_2, \sigma_3 = \text{Id}, X, Y, Z$. This representation is convenient because the average gate fidelity of $\mathcal{E}$ is straightforward to obtain from the χ -matrix via [49]

$$\bar{F}(\Delta, \lambda) = \frac{2}{3} \frac{\chi_{\mathcal{E}}^{\text{Id}, \text{Id}}}{\sum_{\mu} \chi_{\mathcal{E}}^{\sigma_{\mu}, \sigma_{\mu}}} + \frac{1}{3}. \quad (\text{G6})$$

The χ -matrix elements of $\mathcal{E}$ can be obtained from the characteristic function of $E(\Delta, \lambda)$. As a stepping stone to this, we define the logical characteristic function $\xi_E^{\sigma_{\mu}}(\mathbf{v})$ of $E(\Delta, \lambda)$ as

$$\xi_E^{\sigma_{\mu}}(\mathbf{v}) = \text{Tr}[\Pi_{\text{GKP}, \mu} W^{\dagger}(\mathbf{v}) E], \quad (\text{G7a})$$

$$\Pi_{\text{GKP}, \mu} = \Pi_{\text{GKP}, \text{Id}} \bar{\sigma}_{\mu} \Pi_{\text{GKP}, \text{Id}} = \sum_{j, k \in \{0, 1\}} [\sigma_{\mu}]_{j, k} |\bar{j}\rangle \langle \bar{k}|, \quad (\text{G7b})$$

which is based on the so-called “outcome-dependent” GKP projectors defined in Ref. [20]. With these logical characteristic functions, we can directly write

$$\chi_E^{\sigma_{\mu}, \sigma_{\mu'}} = \int_{\square} d\mathbf{v} \xi_E^{\sigma_{\mu}}(\mathbf{v}) \xi_E^{\sigma_{\mu'}}(\mathbf{v})^*, \quad (\text{G8})$$

where the integral is taken over the correctable error patch $\square$ defined as

$$\square = (-1/\sqrt{8}, 1/\sqrt{8})^{\times 2} \subset \mathbb{R}^2, \quad (\text{G9})$$

which is simply a rescaling of Eq. (12).

Reference [27] presents an alternative way of calculating the logical characteristic function. First, we write the error operator $E(\Delta, \lambda)$ as

$$E = \sum_{\mathbf{n} \in \mathbb{Z}^2} \sum_{\mu=0}^3 \int_{\square} d\mathbf{v} f_E(\mathbf{v}, \sigma_{\mu}, \mathbf{n}) W(\mathbf{v}) \bar{\sigma}_{\mu} W(\sqrt{2}\mathbf{n}). \quad (\text{G10})$$

Since $W(\mathbf{v}) \bar{\sigma}_{\mu} W(\sqrt{2}\mathbf{n})$ is a displacement operator [up to a geometric phase from Eq. (3)], the function f can be written as

$$f_E(\mathbf{v}, \sigma_{\mu}, \mathbf{n}) = e^{i\theta(\mathbf{v}, \sigma_{\mu}, \mathbf{n})} \chi_E(\mathbf{v} + \sqrt{2}\mathbf{n} + \boldsymbol{\ell}_{\mu}), \quad (\text{G11})$$

for some function θ , where $\boldsymbol{\ell}_{\mu} = (0, 0), (1/\sqrt{2}, 0), (1/\sqrt{2}, 1/\sqrt{2}), (0, 1/\sqrt{2})$, respectively, for $\mu = 0, 1, 2, 3$. After calculating f from the characteristic function χ_E , the logical characteristic function is simply

$$\xi_E^{\sigma_{\mu}}(\mathbf{v}) = \sum_{\mathbf{n} \in \mathbb{Z}^2} f_E(\mathbf{v}, \sigma_{\mu}, \mathbf{n}). \quad (\text{G12})$$

From Eqs. (G6) and (G8), the average gate fidelity is simply given by

$$\bar{F}(\Delta, \lambda) = \frac{2}{3} \frac{\int_{\square} d\mathbf{v} |\xi_E^{\text{Id}}(\mathbf{v})|^2}{\sum_{\mu} \int_{\square} d\mathbf{v} |\xi_E^{\sigma_{\mu}}(\mathbf{v})|^2} + \frac{1}{3}, \quad (\text{G13})$$

which comes from Eq. (B17) of Ref. [30].

Our first task is to evaluate the denominator in the first term of Eq. (G13), which we call the “normalization constant” of the logical channel $\mathcal{E}$. Computing each individual contribution of the normalization constant is a nontrivial task. However, we can use the following identity to simplify the computation:

$$\sum_{\sigma_{\mu}} \int_{\square} d\mathbf{v} |\xi_E^{\sigma_{\mu}}(\mathbf{v})|^2 = \xi_{E^{\dagger}E}^{\text{Id}}(\mathbf{0}). \quad (\text{G14})$$

To prove Eq. (G14), we start from the right-hand side and make use of the decomposition given in Eq. (G10). Recall that $\xi_{E^{\dagger}E}^{\text{Id}}(\mathbf{0})$ consists of contributions from displacement errors $W(\mathbf{v})$ that act trivially on the GKP codespace. Therefore, when we express the error operators $E^{\dagger}$ and E in terms of their decompositions from Eq. (G10) and combine them, only the terms in $\xi_{E^{\dagger}E}^{\text{Id}}(\mathbf{0})$ with opposite correctable displacements and identical logical Pauli operators contribute to $\xi_{E^{\dagger}E}^{\text{Id}}(\mathbf{0})$. Consequently, we find that

$$\xi_{E^{\dagger}E}^{\text{Id}}(\mathbf{0}) = \int_{\square} d\mathbf{v} \sum_{\mathbf{n}', \mathbf{n} \in \mathbb{Z}^2} \sum_{\mu=0}^3 f(\mathbf{v}, \sigma_{\mu}, \mathbf{n}')^* f(\mathbf{v}, \sigma_{\mu}, \mathbf{n}), \quad (\text{G15})$$

which is just the LHS of Eq. (G14).

We will not compute $\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})$ directly, but use the following trick:

$$\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0}) = \text{Tr}[\Pi_{\text{GKP,Id}} E^\dagger E] \quad (\text{G16a})$$

$$= \text{Tr}[T_3^\dagger E^\dagger E \Pi_{\text{GKP,Id}} T_3] \quad (\text{G16b})$$

$$= \text{Tr}[T_3^\dagger E^\dagger E T_3 \Pi_{\text{GKP,Id}}] \quad (\text{G16c})$$

$$= \text{Tr}[\text{Env}_{\Delta/\lambda, \Delta\lambda}^\dagger \text{Env}_{\Delta/\lambda, \Delta\lambda} \Pi_{\text{GKP,Id}}] \quad (\text{G16d})$$

$$= \text{Tr}[\text{Env}_{\sqrt{2}\Delta/\lambda, \sqrt{2}\Delta\lambda} \Pi_{\text{GKP,Id}}], \quad (\text{G16e})$$

where we have used the fact that T_3 commutes with the projector onto the GKP code $\Pi_{\text{GKP,Id}}$ since it corresponds to a GKP logical gate. Therefore, the normalization constant is entirely dependent on the biased envelope operator. Using the characteristic function of the biased envelope operator given in Eq. (25b), we find that the normalization constant is simply

$$\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0}) = \frac{\coth(\Delta^2)}{2} \vartheta_3(0, e^{-\frac{\pi \coth(\Delta^2)}{\lambda}}) \vartheta_3(0, e^{-\pi \lambda \coth(\Delta^2)}), \quad (\text{G17})$$

where ϑ_3 is the THIRD Jacobi-theta function, defined as

$$\vartheta_3(z, q) = 1 + 2 \sum_{n=1}^{\infty} q^{n^2} \cos(2nz). \quad (\text{G18})$$

Our next step is to evaluate the numerator of the first term in Eq. (G13). We start by evaluating $\xi_E^{\text{Id}}(\mathbf{v})$, which involves calculating the characteristic function $\chi_E(\mathbf{v})$. The first step is decomposing the shear error in Eq. (G4) into displacement errors

$$\begin{aligned} & e^{i2\pi[P_{\min,3}(\frac{q+\sqrt{2\pi}vq}{\sqrt{\pi}}) - P_{\min,3}(\frac{q}{\sqrt{\pi}})]} \\ &= \frac{1}{\sqrt{2\pi}} \iint du_p dx (e^{i2\pi[P_{\min,3}(\frac{x+\sqrt{2\pi}vq}{\sqrt{\pi}}) - P_{\min,3}(\frac{x}{\sqrt{\pi}})]} \\ &\quad \times e^{-i\sqrt{2\pi}u_p x} W(0, u_p)). \end{aligned} \quad (\text{G19})$$

By combining Eq. (G19) with Eq. (G4), through some tedious algebra, we find that χ_E is

$$\begin{aligned} \chi_E(\mathbf{v}) &= \frac{\mathcal{N}(\frac{2 \tanh(\Delta^2/2)}{\lambda}, v_q)}{\sqrt{2\pi}} \int dx (e^{-iv_p \sqrt{2\pi} x} \\ &\quad \times e^{-\lambda \tanh(\frac{\Delta^2}{2}) x^2} e^{i2\pi[P_3(\frac{x}{\sqrt{\pi}}) - P_3(\frac{x-\sqrt{2\pi}vq}{\sqrt{\pi}})]}). \end{aligned} \quad (\text{G20})$$

The integral over x can be performed exactly. Then, by taking the modulus squared and dividing by the normalization constant $\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})$, the result can be written in terms of one-dimensional normal distribution functions

$$\frac{|\chi_E(\mathbf{v})|^2}{\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})} = \frac{\mathcal{N}(\tanh(\frac{\Delta^2}{2})/\lambda, v_q) \mathcal{N}(\frac{v_q^2}{2\lambda \tanh(\frac{\Delta^2}{2})} + \lambda \tanh(\frac{\Delta^2}{2}), v_p - \frac{v_q}{2} + \frac{v_q^2}{\sqrt{2}})}{C(\Delta, \lambda)}, \quad (\text{G21})$$

where

$$C(\Delta, \lambda) = 2 \coth(\Delta^2) \tanh\left(\frac{\Delta^2}{2}\right) \vartheta_3(0, e^{-\frac{\pi \coth(\Delta^2)}{\lambda}}) \vartheta_3(0, e^{-\pi \lambda \coth(\Delta^2)}). \quad (\text{G22})$$

This is not quite equal to the fraction in Eq. (G13), but in the next step we will use Eq. (G21) to lower bound $\bar{F}(\Delta, \lambda)$.

Step 2: Lower bound for the average fidelity $\bar{F}(\Delta, \lambda)$. In this stage of the proof, our goal is to obtain a lower bound for the numerator in Eq. (G13). From Eqs. (G11) and (G12), we can write

$$|\xi_E^{\text{Id}}(\mathbf{v})|^2 = \left| |\chi_E(\mathbf{v})| + \sum_{\substack{\mathbf{n} \in \mathbb{Z}^2 \\ \mathbf{n} \neq \mathbf{0}}} |\chi_E(\mathbf{v} + \sqrt{2}\mathbf{n})| e^{i\theta(\mathbf{v}, \text{Id}, \mathbf{n})} \right|^2 \quad (\text{G23a})$$

$$\equiv \left| |\chi_E(\mathbf{v})| + \chi_E^{\text{Id,rest}}(\mathbf{v}) \right|^2. \quad (\text{G23b})$$

We have also defined $\chi_E^{\text{Id,rest}}$ as the sum of all the contributions to ξ_E^{Id} that come from outside the correction patch located at the origin. Substituting $\arg(\chi_E^{\text{Id,rest}}(\mathbf{v})) = \pi$ for all $\mathbf{v}$ gives a lower bound

$$\int_{\square} d\mathbf{v} |\xi_E^{\text{Id}}(\mathbf{v})|^2 = \int_{\square} d\mathbf{v} \left| |\chi_E(\mathbf{v})| + \chi_E^{\text{Id,rest}}(\mathbf{v}) \right|^2 \quad (\text{G24a})$$

$$\geq \int_{\square} d\mathbf{v} (|\chi_E(\mathbf{v})|^2 - 2|\chi_E(\mathbf{v})||\chi_E^{\text{Id,rest}}(\mathbf{v})| + |\chi_E^{\text{Id,rest}}(\mathbf{v})|^2). \quad (\text{G24b})$$

At this step, we apply the Cauchy-Schwarz inequality successively:

$$\int_{\square} d\mathbf{v} |\chi_E(\mathbf{v})| |\chi_E^{\text{Id,rest}}(\mathbf{v})| \quad (\text{G25a})$$

$$\leq \int dv_q \sqrt{\int dv_p |\chi_E(\mathbf{v})|^2} \sqrt{\int dv_p |\chi_E^{\text{Id,rest}}(\mathbf{v})|^2} \quad (\text{G25b})$$

$$\leq \sqrt{\iint d v_q d v_p |\chi_E(\mathbf{v})|^2} \sqrt{\iint d v_q d v_p |\chi_E^{\text{Id,rest}}(\mathbf{v})|^2} \quad (\text{G25c})$$

$$= \sqrt{\int_{\square} d \mathbf{v} |\chi_E(\mathbf{v})|^2} \sqrt{\int_{\square} d \mathbf{v} |\chi_E^{\text{Id,rest}}(\mathbf{v})|^2}. \quad (\text{G25d})$$

The main idea behind applying the Cauchy-Schwarz inequality is that it allows us to complete the square and thereby obtain a new lower bound

$$\int_{\square} d \mathbf{v} |\xi_E^{\text{Id}}(\mathbf{v})|^2 \quad (\text{G26a})$$

$$\geq \left(\sqrt{\int_{\square} d \mathbf{v} |\chi_E(\mathbf{v})|^2} - \sqrt{\int_{\square} d \mathbf{v} |\chi_E^{\text{Id,rest}}(\mathbf{v})|^2} \right)^2. \quad (\text{G26b})$$

In the following, we find an upper bound for the second term in the parentheses, which will give us the final lower bound to the fidelity. We note that $|\chi_E^{\text{Id,rest}}(\mathbf{v})|$ is upper bounded by

$$|\chi_E^{\text{Id,rest}}(\mathbf{v})| \leq \sum_{\substack{\mathbf{n} \in \mathbb{Z}^2 \\ \mathbf{n} \neq \mathbf{0}}} |\chi_E(\mathbf{v} + \sqrt{2}\mathbf{n})|. \quad (\text{G27})$$

Given this upper bound, we find that

$$\begin{aligned} \int_{\square} d \mathbf{v} |\chi_E^{\text{Id,rest}}(\mathbf{v})|^2 &\leq \int_{\square} d \mathbf{v} \left(\sum_{\substack{\mathbf{n} \in \mathbb{Z}^2 \\ \mathbf{n} \neq \mathbf{0}}} |\chi_E(\mathbf{v} + \sqrt{2}\mathbf{n})| \right)^2 \\ &\leq \left(\sum_{\substack{\mathbf{n} \in \mathbb{Z}^2 \\ \mathbf{n} \neq \mathbf{0}}} \sqrt{\int_{\square} d \mathbf{v} |\chi_E(\mathbf{v} + \sqrt{2}\mathbf{n})|^2} \right)^2, \end{aligned} \quad (\text{G28a})$$

$$(\text{G28b})$$

where in the final inequality, we follow the same steps as in Eqs. (G24) and (G25). In summary, our final lower bound is

$$\begin{aligned} \frac{\int_{\square} d \mathbf{v} |\xi_E^{\text{Id}}(\mathbf{v})|^2}{\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})} &\geq \left(\sqrt{\int_{\square} \frac{d \mathbf{v} |\chi_E(\mathbf{v})|^2}{\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})}} - \sum_{\substack{\mathbf{n} \in \mathbb{Z}^2 \\ \mathbf{n} \neq \mathbf{0}}} \sqrt{\int_{\square} \frac{d \mathbf{v} |\chi_E(\mathbf{v} + \sqrt{2}\mathbf{n})|^2}{\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})}} \right)^2. \end{aligned} \quad (\text{G29})$$

Let us denote

$$\frac{p_E(\mathbf{n})}{C(\Delta, \lambda)} = \int_{\square} d \mathbf{v} \frac{|\chi_E(\mathbf{v} + \mathbf{n})|^2}{\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})}, \quad (\text{G30})$$

so that we rewrite Eq. (G29) as

$$\frac{\int_{\square} d \mathbf{v} |\xi_E^{\text{Id}}(\mathbf{v})|^2}{\xi_{E^\dagger E}^{\text{Id}}(\mathbf{0})} \geq \left(\frac{p_E(\mathbf{0})}{C(\Delta, \lambda)} - \sum_{\substack{\mathbf{n} \in \mathbb{Z}^2 \\ \mathbf{n} \neq \mathbf{0}}} \frac{p_E(\sqrt{2}\mathbf{n})}{C(\Delta, \lambda)} \right)^2. \quad (\text{G31})$$

Based on Eq. (G21), we can interpret $p_E(\mathbf{n})$ as the probability that a displacement error $\mathbf{v}$ lies inside a square patch $\square$

centered at $\mathbf{n}$. Hence, using Eq. (G21), we find that

$$\sum_{\mathbf{n}} p_E\left(\frac{\mathbf{n}}{\sqrt{2}}\right) = 1. \quad (\text{G32})$$

In the next step, we will make use of the following property of Gaussian integrals to obtain a lower bound for $p_E(\mathbf{0})$:

$$\int_{-D}^D dx \mathcal{N}(\Delta^2, x - \mu_1) \geq \int_{-D}^D dx \mathcal{N}(\Delta^2, x - \mu_2), \quad (\text{G33})$$

for all D and Δ^2 , provided that $|\mu_1| \leq |\mu_2|$. Intuitively, Eq. (G33) expresses the fact that the integral over the interval $(-D, D)$ represents the probability that the random variable x lies within that range. This probability is maximized when the mean of the Gaussian μ is at the center of the interval, i.e., at the origin $\mu = 0$. As the mean μ moves away from the origin, the Gaussian distribution shifts, and because its tails decay exponentially, the probability of finding x within $(-D, D)$ decreases accordingly. For $\mathbf{v} \in \square$, we find that

$$\max_{\mathbf{v} \in \square} \left| \frac{v_q}{2} - \frac{v_q^2}{\sqrt{2}} \right| = \frac{3}{4\sqrt{8}}, \quad (\text{G34})$$

obtained at $v_q = -1/\sqrt{8}$. Consequently, by using Eq. (G33), we can lower bound $p_E(\mathbf{0})$ by

$$\begin{aligned} p_E(\mathbf{0}) &\geq \int_{\square} d \mathbf{v} \mathcal{N}\left(\tanh\left(\frac{\Delta^2}{2}\right)/\lambda, v_q\right) \\ &\quad \times \mathcal{N}\left(\frac{v_q^2}{2\lambda \tanh\left(\frac{\Delta^2}{2}\right)} + \lambda \tanh\left(\frac{\Delta^2}{2}\right), v_p - \frac{3}{4\sqrt{8}}\right) \end{aligned} \quad (\text{G35a})$$

$$\begin{aligned} &\geq \int_{-1/\sqrt{8}}^{1/\sqrt{8}} d v_q \mathcal{N}\left(\tanh\left(\frac{\Delta^2}{2}\right)/\lambda, v_q\right) \int_{\frac{1}{2\sqrt{8}}}^{\frac{1}{\sqrt{8}}} d v_p \\ &\quad \times \mathcal{N}\left(\frac{v_q^2}{2\lambda \tanh\left(\frac{\Delta^2}{2}\right)} + \lambda \tanh\left(\frac{\Delta^2}{2}\right), v_p - \frac{3}{4\sqrt{8}}\right) \end{aligned} \quad (\text{G35b})$$

$$\begin{aligned} &= \int_{-1/\sqrt{8}}^{1/\sqrt{8}} d v_q \mathcal{N}\left(\tanh\left(\frac{\Delta^2}{2}\right)/\lambda, v_q\right) \\ &\quad \times \text{Erf}\left(\frac{\sqrt{\pi}}{4\sqrt{8} \sqrt{\frac{v_q^2}{2\lambda \tanh\left(\frac{\Delta^2}{2}\right)} + \lambda \tanh\left(\frac{\Delta^2}{2}\right)}}\right) \end{aligned} \quad (\text{G35c})$$

In the next step, we limit the integral over v_q to the interval

$$v_q \in \left(-\sqrt[4]{\tanh\left(\frac{\Delta^2}{2}\right)/\lambda}, \sqrt[4]{\tanh\left(\frac{\Delta^2}{2}\right)/\lambda}\right), \quad (\text{G36})$$

under the assumption that

$$\sqrt[4]{\tanh\left(\frac{\Delta^2}{2}\right)/\lambda} \leq \frac{1}{\sqrt{8}}. \quad (\text{G37})$$

We will later derive a condition that specifies for which values of Δ and λ this assumption holds. By limiting the interval of the integral, we can use $v_q^2 \leq \sqrt{\tanh(\Delta^2/2)/\lambda}$ to obtain our final lower bound to $p_E(\mathbf{0})$ as

$$p_E(\mathbf{0}) \geq \text{Erf}\left(\frac{\sqrt{\pi}}{(\tanh(\Delta^2/2)/\lambda)^{1/4}}\right) \times \text{Erf}\left(\frac{\sqrt{\pi}}{(4\sqrt{8})\sqrt{\frac{\sqrt{\tanh(\Delta^2/2)/\lambda}}{2\lambda \tanh(\frac{\Delta^2}{2})} + \lambda \tanh(\frac{\Delta^2}{2})}}\right). \quad (\text{G38})$$

Step 3: Choosing the asymmetry value $\lambda(\Delta)$. Based on the argument presented in the main text, we know that the shear error dominates in the small- Δ limit. The second Erf term can be identified as the contribution associated with the shear error. Therefore, we choose $\lambda(\Delta)$ to be the value of λ that maximizes the second Erf function in Eq. (G38), which amounts to minimizing

$$\min_{\lambda} \left(\frac{\sqrt{\tanh(\Delta^2/2)/\lambda}}{2\lambda \tanh(\frac{\Delta^2}{2})} + \lambda \tanh\left(\frac{\Delta^2}{2}\right) \right). \quad (\text{G39})$$

Through some simple algebra, we find that

$$\lambda(\Delta) = \frac{3^{2/5}}{2^{4/5}} (\tanh(\Delta^2/2))^{-3/5} \quad (\text{G40a})$$

$$\propto (\tanh(\Delta^2/2))^{-3/5}. \quad (\text{G40b})$$

Substituting back, we find that

$$\tanh(\Delta^2/2)/\lambda(\Delta) \propto \tanh(\Delta^2/2)^{8/5}, \quad (\text{G41a})$$

$$\frac{\sqrt{\tanh(\Delta^2/2)/\lambda(\Delta)}}{2\lambda(\Delta) \tanh(\frac{\Delta^2}{2})} + \lambda(\Delta) \tanh\left(\frac{\Delta^2}{2}\right) \propto \tanh(\Delta^2/2)^{2/5}. \quad (\text{G41b})$$

Given our choice of $\lambda(\Delta)$, we can also derive the condition on Δ under which Eq. (G38) is valid, namely,

$$\Delta \leq \sqrt{2} \sqrt{\text{arctanh}\left(\frac{1}{16} \sqrt{\frac{3}{2}}\right)} \approx 0.372. \quad (\text{G42})$$

Again using our choice of $\lambda(\Delta)$ given in Eq. (G40), by substituting Eq. (G41) back into Eq. (G38), we find that the RHS of Eq. (G38) approaches 1 as $\Delta \rightarrow 0$. Due to Eq. (G32), we can conclude that

$$\lim_{\Delta \rightarrow 0} p_E(\mathbf{0}) = 1 \quad (\text{G43})$$

and

$$\lim_{\Delta \rightarrow 0} p_E(\sqrt{2}\mathbf{n}) = 0, \quad \forall \mathbf{n} \in \mathbb{Z}^2 \setminus \{\mathbf{0}\}. \quad (\text{G44})$$

One final observation is that the function $C(\Delta, \lambda(\Delta))$ from Eq. (G22) is a strictly decreasing function as Δ decreases and

$$\lim_{\Delta \rightarrow 0} C(\Delta, \lambda(\Delta)) = 1. \quad (\text{G45})$$

By combining Eqs. (G43) and (G45), we find that the RHS of Eq. (G31) approaches 1 in the limit $\Delta \rightarrow 0$. Consequently, the gate fidelity $\bar{F}$ also approaches 1 in this limit. Hence, it implies that for any given target gate infidelity ϵ_{target} , there exists a threshold Δ_{th} such that

$$\Delta \leq \Delta_{\text{th}} \Rightarrow 1 - F(\Delta, \lambda(\Delta)) \leq \epsilon_{\text{target}}, \quad (\text{G46})$$

which proves Theorem G1.

APPENDIX H: NUMERICAL METHODS

In this Appendix, we detail the numerical methods we used to produce the plots in Sec. VI and provide some additional plots that we did not include in the main text.

1. Polynomial phase gates

We will begin by detailing the numerical methods we use to model the effective logical channel $\mathcal{E}$ that we introduced in Sec. VI. To recap, here we define the effective logical channel as

$$|\psi\rangle \xrightarrow{\quad} |\bar{\psi}\rangle_{\Delta_q, \Delta_p} \xrightarrow{T} \boxed{\text{non-biased noise}} \xrightarrow{\quad} \boxed{\text{Ideal QEC}} \xrightarrow{\quad} \rho = \mathcal{E}(|\psi\rangle\langle\psi|) \quad (71)$$

where $\Delta_q = \Delta/\lambda$ and $\Delta_p = \Delta\lambda$ as usual. Our definition of $\mathcal{E}$ here differs from the definition in Appendix G by including a nonbiased-noise channel to account for noise in the QEC cycle that follows.

Our numerics are state vector simulations in the Fock basis with a variable truncation dimension that increases over the course of the simulation. In particular, the key subroutine of our simulation takes the following inputs: an input qubit state $|\psi\rangle$, a polynomial phase gate with polynomial $P(x)$, a GKP codestate approximation level Δ , an asymmetry λ , and a measured Pauli operator P . The subroutine then outputs the expectation value $\text{tr}(P\mathcal{E}(|\psi\rangle\langle\psi|))$, where $\mathcal{E}$ is the effective logical channel of the gate. We will explain later how this subroutine can be used to determine the average gate fidelity and state fidelity.

Before proceeding, note that Eq. (71) can be rewritten using Eq. (18) as

$$|\psi\rangle \xrightarrow{\quad} |\bar{\psi}\rangle_{\Delta}^{\lambda} \xrightarrow{T(\lambda)} \boxed{\text{biased noise}} \xrightarrow{\quad} \boxed{\text{Ideal QEC}} \xrightarrow{\quad} \rho = \mathcal{E}(|\psi\rangle\langle\psi|) \quad (\text{H1})$$

Note that this is *not* the same as the on-demand morphing scheme described in Appendix A because the noise preceding the ideal QEC box is biased. Equation (H1) is preferable numerically because the nonbiased envelope of the input state $|\bar{\psi}\rangle_{\Delta}^{\lambda}$ lowers the average photon number of the state being simulated, reducing the truncation error.

With the rewrite in Eq. (H1), we can describe the simulation in three steps:

(1) state preparation of a logical rectangular GKP codestate with a nonbiased envelope $|\bar{\psi}\rangle_\Delta^\lambda$, as defined in Eq. (7);

(2) the polynomial phase gate acting on the rectangular GKP code, given by $\exp(2\pi i P(q/(\sqrt{\lambda}\pi)))$; and

(3) a biased Gaussian random displacement channel $\mathcal{G}_\Sigma$ with covariance matrix $\Sigma = \tanh(\Delta^2/2)\text{diag}(\lambda, \lambda^{-1})$, followed by ideal QEC and a trivial unencoding into a logical qubit density matrix. The noise, QEC, and unencoding will all be simulated together.

We now proceed in deriving the equations that we use in our simulation, starting with the preparation of the $|\bar{\psi}\rangle_\Delta^\lambda$ state. For notational clarity, we denote the vacuum state as $|\text{vac}\rangle$, single-mode displacement operators as $W(\mathbf{v}) = W(v_1, v_2)$, and coherent states as displaced vacuum states $W(\mathbf{v})|\text{vac}\rangle$. Working toward an expression in the Fock basis $\{|n\rangle_{\text{Fock}}\}$ that we can numerically evaluate, we begin with a decomposition of ideal GKP codestates into coherent states following Ref. [5]. In particular, the $|\bar{0}\rangle^\lambda$ state is the unique simultaneous $+1$ eigenstate of the rectangular GKP stabilizer $S_q = W(\sqrt{2\lambda}, 0)$ and the logical operator $Z = W(0, 1/\sqrt{2\lambda})$. Therefore, we can write the projector onto the $|\bar{0}\rangle^\lambda$ state as [5]

$$|\bar{0}\rangle^\lambda \langle \bar{0}|^\lambda = \sum_{m,n \in \mathbb{Z}} S_q^m Z^n, \quad (\text{H2})$$

since any other simultaneous eigenstate of S_q and Z has at least one of these eigenvalues not equal to 1, and will therefore be a 0-eigenstate of Eq. (H2). Applying this projector to the vacuum state gives

$$|\bar{0}\rangle^\lambda = \sum_{m,n \in \mathbb{Z}} (-1)^{mn} W(m\sqrt{2\lambda}, n/\sqrt{2\lambda})|\text{vac}\rangle, \quad (\text{H3})$$

and applying $X = W(\sqrt{\lambda/2}, 0)$ to Eq. (H3) gives

$$|\bar{1}\rangle^\lambda = \sum_{m,n \in \mathbb{Z}} i^{2mn-n} W((2m+1)\sqrt{\lambda/2}, n/\sqrt{2\lambda})|\text{vac}\rangle. \quad (\text{H4})$$

Next, we need an expression for the approximate codestates $e^{-\Delta^2 a^\dagger a}$. This is simple to derive from the identity

$$\begin{aligned} e^{-\Delta^2 a^\dagger a} W(v_q, v_p) e^{\Delta^2 a^\dagger a} \\ = e^{-\pi(v_q^2 + v_p^2)(1 - e^{-2\Delta^2})/2} W(e^{-\Delta^2} v_q, e^{-\Delta^2} v_p) \\ \times e^{2\sqrt{\pi}(-v_1 + iv_2) \sinh(\Delta^2) a}, \end{aligned} \quad (\text{H5})$$

which itself can be derived from the Baker-Campbell-Hausdorff formula. Since $e^{-\Delta^2 a^\dagger a}|\text{vac}\rangle = e^{\alpha a}|\text{vac}\rangle = |\text{vac}\rangle$, this gives

$$|\bar{0}\rangle_\Delta^\lambda = \sum_{m,n \in \mathbb{Z}} c_{2m,n} (-1)^{mn} W(m\sqrt{2\lambda}, n/\sqrt{2\lambda})|\text{vac}\rangle, \quad (\text{H6a})$$

$$|\bar{1}\rangle_\Delta^\lambda = \sum_{m,n \in \mathbb{Z}} (c_{2m+1,n} i^{2mn-n} W((2m+1)\sqrt{\lambda/2}, n/\sqrt{2\lambda})|\text{vac}\rangle), \quad (\text{H6b})$$

where we have defined $c_{m,n} = \exp(-\pi(m^2\lambda + n^2/\lambda)(1 - e^{-2\Delta^2})/4)$ for convenience. Meanwhile, coherent states them-

selves can be represented in the Fock basis using the standard expression, which in our notation is

$$W(v_q, v_p)|\text{vac}\rangle = e^{-\pi(v_q^2 + v_p^2)/2} \sum_{n=0}^{\infty} \frac{\sqrt{\pi}^n (v_q + iv_p)^n}{\sqrt{n!}} |n\rangle_{\text{Fock}}. \quad (\text{H7})$$

Using Eqs. (H6) and (H7) allows us to generate GKP codestates in the Fock basis numerically, truncated to a desired Fock truncation dimension $d_{\text{trunc,init}}$; in this work, we generate the codestates at $d_{\text{trunc,init}} = 400$.

Some care needs to be taken to ensure the state generation method is stable at high Fock dimensions. In particular, calculating $(v_q + iv_p)^n / \sqrt{n!}$ for large n can be numerically unstable and can return large or infinite results. This is undesirable because coherent states with large $|v_q + iv_p|$ are exponentially suppressed in the GKP codestates themselves due to the $c_{m,n}$ coefficient. To combat this, we simply check whether the coherent state has returned finite values; otherwise, we set it to zero to avoid corrupting the remaining GKP codestate. The exclusion of some of these coherent states will cause some numerical error in addition to the usual truncation error, although in our tests we found this to be lower than 10^{-6} across the full range of Δ that we probed.

One important subtlety that must be taken into account is that the codestates $|\bar{0}\rangle_\Delta^\lambda$ and $|\bar{1}\rangle_\Delta^\lambda$ are not orthogonal. This is a problem because it would mean that the effective logical “channel” $\mathcal{E}$ would not be a channel because it is not trace preserving. To rectify this, we follow the same procedure as in Ref. [27] and orthonormalize the codewords in Eq. (H6). First, the output states from Eq. (H6) are not normalized, so we begin by normalizing them. Then, to orthogonalize two arbitrary, normalized vectors $|\psi\rangle$ and $|\phi\rangle$, we first calculate the angle $\theta = \arg(\langle\psi|\phi\rangle)$. Then, the superpositions $|+\rangle = N_+ (|\psi\rangle + e^{-i\theta}|\phi\rangle)$ and $|-\rangle = N_- (|\psi\rangle - e^{-i\theta}|\phi\rangle)$ are guaranteed to be orthogonal. Moreover, we choose the constants $N_\pm$ such that the states $|\pm\rangle$ are normalized. We can then construct orthogonalized states $|\psi\rangle_{\text{orth}} = (|+\rangle + |-\rangle)/\sqrt{2}$ and $|\phi\rangle_{\text{orth}} = e^{i\theta}(|+\rangle - |-\rangle)/\sqrt{2}$. This procedure coincides with the Löwdin orthogonalization [53] in the special case of having two 2D vectors. The additional error introduced by this procedure is negligible for $\bar{n} > 2$ ($\Delta < 0.45$); see, for example, Fig. 11 of Ref. [27]. With this, we can obtain arbitrary logical codestates by taking superpositions of the orthonormal codestates.

With the GKP codestate $|\bar{\psi}\rangle_\Delta^\lambda$ now prepared in the Fock basis, we can apply the polynomial phase gate. The truncated Fock basis representation of the polynomial phase gate $\exp(2\pi i P(q/(\lambda\sqrt{\pi})))$ is straightforward to obtain simply by calculating $q = (a + a^\dagger)/\sqrt{2}$ in the truncated Fock basis and taking the appropriate matrix exponential. However, there are two problems with this simple approach. The first is that performing the matrix exponential over the truncated Fock basis does not coincide with truncating the exponent of the infinite-dimensional operator, leading to additional numerical errors. This can be solved by performing the matrix exponential over a temporary, higher truncation dimension, say $d_{\text{trunc,temp}} = 3d_{\text{trunc,init}}$, and then truncating to $d_{\text{trunc,init}}$ afterward. The second problem is that even if the input state has no support on Fock basis states above $d_{\text{trunc,init}}$, the gate may

increase its photon number such that the output state after the gate does have support on Fock basis states above $d_{\text{trunc,init}}$, leading to additional truncation error. We solve this second problem by only truncating the *input* dimension of the matrix, so that the gate is represented by a $d_{\text{trunc,out}} \times d_{\text{trunc,init}} = 3d_{\text{trunc,init}} \times d_{\text{trunc,init}} = 1200 \times 400$ matrix. This way, the output state has a higher truncation dimension than the input state, avoiding any significant truncation error due to the gate.

Finally, we explain how to model the biased-noise channel $\mathcal{G}_\Sigma$, the final ideal QEC round, and the unencoding into the logical qubit subspace. We start by explaining how to model the ideal error correction and unencoding without the preceding biased-noise channel; we will explain how to incorporate $\mathcal{G}_\Sigma$ afterward. To begin, note that applying ideal QEC to any CV state ρ maps it to an ideal GKP codestate $\bar{\tau}$ whose logical state is given by [30]

$$\tau = \frac{1}{2} (I\text{tr}(\rho) + X\text{tr}(\rho X_m^\lambda) + Y\text{tr}(\rho Y_m^\lambda) + Z\text{tr}(\rho Z_m^\lambda)). \quad (\text{H8})$$

Here, the operators X_m^λ , Y_m^λ , and Z_m^λ operators are the *ideal Pauli measurement operators* introduced in Ref. [27]. In particular, Z_m^λ is defined such that it acts as a logical Z on any ideal GKP codestate that has been displaced by a *correctable* displacement error [Eq. (12)]; that is, it acts as

$$Z_m^\lambda W(v_q, v_p) |\bar{0}\rangle^\lambda = +W(v_q, v_p) |\bar{0}\rangle^\lambda, \quad (\text{H9a})$$

$$Z_m^\lambda W(v_q, v_p) |\bar{1}\rangle^\lambda = -W(v_q, v_p) |\bar{1}\rangle^\lambda, \quad (\text{H9b})$$

for all $-\sqrt{\lambda/8} < v_q \leq \sqrt{\lambda/8}$ and $-1/\sqrt{8\lambda} < v_p \leq 1/\sqrt{8\lambda}$. Similar expressions also exist for the other Pauli operators X_m and Y_m acting on their respective correctable displaced GKP logical eigenstates.

Our task is, therefore, to obtain the Fock basis representation of the ideal Pauli measurement operators. In Appendix D of Ref. [30], it was shown that these ideal Pauli measurement operators can be written as an infinite sum of displacement operators given by

$$X_m^\lambda = \frac{1}{\pi} \sum_{n \in \mathbb{Z}} \frac{(-1)^n}{n + \frac{1}{2}} W((2n+1)\sqrt{\lambda/2}, 0), \quad (\text{H10a})$$

$$Z_m^\lambda = \frac{1}{\pi} \sum_{n \in \mathbb{Z}} \frac{(-1)^n}{n + \frac{1}{2}} W(0, (2n+1)/\sqrt{2\lambda}). \quad (\text{H10b})$$

The displacement operators themselves can be obtained by taking a matrix exponential; again, we use a higher temporary Fock truncation dimension to perform the truncation to avoid numerical errors. At this stage in the simulation, our Fock state vector has dimension $d_{\text{trunc,out}} = 3d_{\text{trunc,init}} = 1200$; therefore, the matrix exponential is performed at a dimension of $d_{\text{trunc,temp}} = 3d_{\text{trunc,out}} = 3600$ before truncating back to a $d_{\text{trunc,out}} \times d_{\text{trunc,out}} = 1200 \times 1200$ matrix. With this, we can obtain the X_m^λ and Z_m^λ operators by truncating the sum over the displacement operators [Eq. (H10)] such that $(2n+1)$ runs over all odd numbers from -59 to $+59$, which we found to be sufficient to avoid significant errors from this truncation. Finally, the Y_m^λ operator is calculated by

$$Y_m^\lambda = \frac{1}{2} (iX_m^\lambda Z_m^\lambda - iZ_m^\lambda X_m^\lambda), \quad (\text{H11})$$

with the symmetric form of this expression chosen to improve numerical stability.

Now, we return to the issue of the Gaussian random displacement channel (arising from the noisy stabilizer measurements) that must be applied before taking the expectation value with the ideal Pauli measurement operators. In particular, Gaussian random displacement channels obey the relation

$$\text{tr}(O\mathcal{G}_\Sigma(\rho)) = \text{tr}(\mathcal{G}_\Sigma(O)\rho) \quad (\text{H12})$$

for any operator O . Therefore, we can rewrite the ideal QEC and decoding steps in Eq. (H8) to incorporate $\mathcal{G}_\Sigma$ as

$$\tau = \frac{1}{2} (I\text{tr}(\mathcal{G}_\Sigma(\rho)) + X\text{tr}(\mathcal{G}_\Sigma(\rho)X_m^\lambda) + Y\text{tr}(\mathcal{G}_\Sigma(\rho)Y_m^\lambda) + Z\text{tr}(\mathcal{G}_\Sigma(\rho)Z_m^\lambda)) \quad (\text{H13a})$$

$$= \frac{1}{2} (I\text{tr}(\rho) + X\text{tr}(\rho X_{m,\Sigma}^\lambda) + Y\text{tr}(\rho Y_{m,\Sigma}^\lambda) + Z\text{tr}(\rho Z_{m,\Sigma}^\lambda)), \quad (\text{H13b})$$

where we have introduced the *approximate* Pauli measurement operators defined as $P_{m,\Sigma}^\lambda = \mathcal{G}_\Sigma(P_m^\lambda)$. The expressions for the approximate Pauli measurement operators can be derived from their ideal counterparts [Eq. (H10)] via the equation

$$\mathcal{G}_\Sigma(W(v_q, v_p)) = \exp(-\pi \mathbf{v}^T \Omega^T \Sigma \Omega \mathbf{v}) W(v_q, v_p), \quad (\text{H14})$$

where $\Omega = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ is the single-mode version of Eq. (C3) and $\mathbf{v} = \begin{bmatrix} v_q \\ v_p \end{bmatrix}$. This amounts simply to a rescaling of the coefficients in Eq. (H10). Therefore, applying Eq. (H14) to Eq. (H10) gives the approximate Pauli measurement operators in the Fock basis, allowing us to model the approximate QEC and unencoding steps at the same time in our numerics. Moreover, this has the advantage that these approximate Pauli measurement operators can be applied to pure states, avoiding the need to compute and store a CV density matrix.

To briefly summarize, we can calculate an expectation value $\text{tr}(P\mathcal{E}(|\psi\rangle\langle\psi|))$ by first obtaining the $|\bar{\psi}\rangle_\Delta^\lambda$ state using superpositions of the orthonormalized versions of the states Eq. (H6) truncated to $d_{\text{trunc,init}} = 400$, applying the polynomial phase gate truncated to a $d_{\text{trunc,out}} \times d_{\text{trunc,init}} = 1200 \times 400$ matrix, and then taking the expectation value of this output state with the approximate Pauli measurement operator $P_{m,\Sigma}^\lambda$ (truncated to $d_{\text{trunc,out}} \times d_{\text{trunc,out}} = 1200 \times 1200$) with $\Sigma = \tanh(\Delta^2/2)\text{diag}(\lambda, \lambda^{-1})$, using Eqs. (H14) and (H10).

With these ingredients, we can model the effective logical channel applied to any pure input qubit state. To calculate the average gate fidelity efficiently, we utilize the formula from Ref. [49], giving

$$F(\mathcal{E}, U) = \frac{1}{3} + \frac{1}{12} \sum_{j=0}^3 \text{tr}(U\sigma_j U^\dagger \mathcal{E}(\sigma_j)), \quad (\text{H15})$$

where the σ_j operators are the I, X, Y , and Z Pauli matrices, respectively, for $j = 0, 1, 2, 3$. After some rearrangements, we obtain

$$F(\mathcal{E}, U) = \frac{1}{3} + \frac{1}{12} \sum_{k,\ell=0}^3 \text{tr}(U V_k U^\dagger \sigma_\ell) \text{tr}(\sigma_\ell \mathcal{E}(|\psi_k\rangle\langle\psi_k|)), \quad (\text{H16})$$

where $\{|\psi_k\rangle\}$ is a set of (qubit) pure states such that $|\psi_k\rangle\langle\psi_k|$ is a basis for the space of qubit operators and $V_k = \sum_j \alpha_{jk} \sigma_j / 2$, where α_{jk} are coefficients such that $\sigma_j = \sum_k \alpha_{jk} |\psi_k\rangle\langle\psi_k|$. In this work, we choose $|\psi_k\rangle \in \{|1\rangle, |+\rangle, |i\rangle, |0\rangle\}$ where $|i\rangle = (|0\rangle + i|1\rangle)/\sqrt{2}$, although we expect that different choices would give identical numerical performance. Equation (H16) is convenient because the latter trace can be evaluated simply by evaluating the relevant expectation value of the ideal Pauli operator as described above, instead of explicitly evaluating $\mathcal{E}$.

A similar method can also be used to extract the *state* fidelity in Fig. 5. In particular, we wish to calculate

$$F = \langle T | \mathcal{E}(|+\rangle\langle+|) | T \rangle. \quad (\text{H17})$$

To do this, note that $|T\rangle\langle T| = I/2 + (X+Y)/(2\sqrt{2})$, and therefore we can write

$$F = \frac{1}{2} + \frac{1}{2\sqrt{2}} \sum_{j=1}^2 \text{tr}(\sigma_j \mathcal{E}(|+\rangle\langle+|)), \quad (\text{H18})$$

which again can be easily evaluated using the Pauli measurement operators $X_{m,\Sigma}^\lambda$ and $Y_{m,\Sigma}^\lambda$.

For each polynomial phase gate, we calculated the average gate fidelity for input values of $\Delta = 1/\sqrt{2\bar{n}+1}$, where $\bar{n}$ ranges from 2 to 20 in increments of 0.5, and for 32 input values of the asymmetry λ ranging from 1 to 6.5. We also collected the same data for the state fidelity of the produced $|T\rangle$ state for each polynomial phase gate that we tested that implements a T gate. The slowest stage of the simulation is the generation of the ideal Pauli measurement operators, so we precomputed these for each value of λ and Δ and reused it for each gate.

The full results are presented in Figs. 6 and 7, from which the plots in the main text are produced. Note that there are no error bars here since the simulations are matrix calculations instead of Monte Carlo simulations. One can see here that achieving the precisely optimal asymmetry is not crucial to achieve good performance. Moreover, note that for the identity gate, the optimal asymmetry is always 1, indicating that under idling noise the square GKP code outperforms all rectangular GKP codes as expected. Compared to the main text, we also tested two additional gates. The first is an alternative polynomial phase gate implementation of the $T^{1/4}$ gate given by

$$P(x) = -x^5/240 - x^4/96 + x^3/48 + x^2/24 - x/60, \quad (\text{H19})$$

which is related to the polynomial for the $T^{1/4}$ gate given in Table I by the transformation $x \leftrightarrow -x$. Both polynomials are minimal under the lexicographic ordering because they differ only in the absolute value of each of the coefficients; the two implementations perform identically up to numerical error. The second is an “implementation” of the $T^{1/8}$ qubit gate that at the CV level implements an identity gate I [that is, the polynomial is $P(x) = 0$]. In other words, we implement the identity gate I at the CV level, but then evaluate the average gate infidelity with respect to the $T^{1/8}$ gate. The purpose of this is to evaluate a trivial benchmark in terms of average gate infidelity that the nontrivial implementation of the $T^{1/8}$ gate must beat.

In Fig. 8, we compare the trivial and nontrivial implementations of the $T^{1/8}$ gates. Note that the trivial and

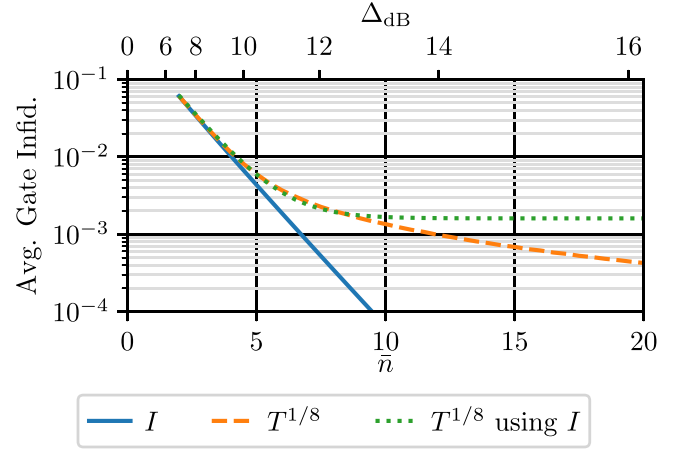


FIG. 8. Comparison of the trivial (green dotted) and nontrivial (orange dashed) implementations of the $T^{1/8}$ gate. For reference, we also plot the identity gate I evaluated with the average gate infidelity with respect to the identity gate (blue solid).

nontrivial implementations have almost identical performance until the trivial implementation reaches an infidelity floor at $\sim 1.6 \times 10^{-3}$, which corresponds to the qubit-level average gate infidelity between $T^{1/8}$ and I , as explained in the main text. To achieve this, one requires GKP codestates of higher quality than $\Delta = 0.25$ ($\bar{n} = 8$, $\Delta_{\text{dB}} = 12$).

2. Vacuum-state method

Finally, we explain how we simulate the vacuum-state method, following the same method as in Ref. [20].

The scheme that we wish to simulate consists of the following steps:

- (1) Begin with a vacuum state $|\text{vac}\rangle$;
- (2) measure the stabilizers with a round of square GKP QEC (with noisy measurements); and
- (3) based on the stabilizer measurement outcomes, perform a logical Clifford correction to make the state as close to the $|T\rangle$ state as possible.

After this procedure, we average over the possible measurement outcomes that could have occurred (weighted appropriately by the probability of obtaining each outcome) giving a CV density matrix ρ_{CV} . The round of QEC is modeled with noisy measurements but projects the state into the ideal GKP subspace, which follows the same noise model as the QEC round 2 in the circuit $\mathcal{C}$ as explained in the main text. This means that ρ_{CV} is in the ideal GKP subspace and can be trivially converted into a qubit density matrix ρ , which is the output of this procedure. We can then quantify the quality of the output state by taking the state fidelity between ρ and $|T\rangle = T|+\rangle$.

In our simulations, the only noise arises from the noisy QEC measurements, which we model as a nonbiased Gaussian random displacement function $\mathcal{G}_{\tanh(\Delta^2/2)I}$ acting on the vacuum state prior to an ideal GKP QEC round. Conveniently, a nonbiased Gaussian random displacement function applied to the vacuum state corresponds exactly to a *thermal* state $\rho_{\text{th}}(\bar{n})$ with average photon number in our case given by $\bar{n} = \tanh(\Delta^2/2)$. This allows us to directly use the equations given

in Ref. [20], which give us the output qubit state conditioned on a given pair of stabilizer measurement outcomes ($e^{i\phi_1}$, $e^{i\phi_2}$) along with the probability of these outcomes occurring. To perform our simulations, we sample the continuous set of possible stabilizer measurement outcomes from a 500×500 grid. For each output state, we calculate the state fidelity between it and each of the 12 states that are Clifford equivalent to a $|T\rangle$ state and return the maximum of these 12 fidelities. Finally, we sort the outcomes from highest fidelity to lowest.

In the case of no postselection ($p = 100\%$), we take the average of all these fidelities weighted according to the probability of obtaining the corresponding measurement outcome. With postselection ($0\% < p < 100\%$), we take the (weighted) average of the best fidelities up until the postselection fraction has been reached. To obtain the infidelity lower bound ($p = 0\%$), we simply return the maximum fidelity (minimum infidelity) that was obtained in the grid, which in each case corresponded to obtaining $+1$ outcomes for both stabilizers. This is sufficient to produce Fig. 5.

- [1] B. M. Terhal, Quantum error correction for quantum memories, *Rev. Mod. Phys.* **87**, 307 (2015).
- [2] N. P. Breuckmann and J. N. Eberhardt, Quantum low-density parity-check codes, *PRX Quantum* **2**, 040101 (2021).
- [3] V. V. Albert, Bosonic coding: Introduction and use cases, in *Quantum Fluids of Light and Matter, International School of Physics “Enrico Fermi”*, edited by A. Bramati, I. Carusotto, and C. Ciuti (IOS Press, NL, 2025), Vol. 209, pp. 79–107.
- [4] B. M. Terhal, J. Conrad, and C. Vuillot, Towards scalable bosonic quantum error correction, *Quantum Sci. Technol.* **5**, 043001 (2020).
- [5] D. Gottesman, A. Kitaev, and J. Preskill, Encoding a qubit in an oscillator, *Phys. Rev. A* **64**, 012310 (2001).
- [6] Y. Xu, Y. Wang, E.-J. Kuo, and V. V. Albert, Qubit-oscillator concatenated codes: Decoding formalism and code comparison, *PRX Quantum* **4**, 020342 (2023).
- [7] J. Zhang, Y.-C. Wu, and G.-P. Guo, Concatenation of the Gottesman-Kitaev-Preskill code with the XZZX surface code, *Phys. Rev. A* **107**, 062408 (2023).
- [8] 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).
- [9] 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).
- [10] 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 (London)* **616**, 50 (2023).
- [11] B. L. Brock, S. Singh, A. Eickbusch, V. V. Sivak, A. Z. Ding, L. Frunzio, S. M. Girvin, and M. H. Devoret, Quantum error correction of qudits beyond break-even, *Nature (London)* **641**, 612 (2025).
- [12] L. B. Nguyen, H. Kim, D. T. Le, T. Ersevı, S. P. Chitta, T. Chistolini, C. Jünger, W. C. Smith, T. M. Stace, J. Koch, D. I. Santiago, and I. Siddiqi, The superconducting grid-states qubit, [arXiv:2509.14656](https://arxiv.org/abs/2509.14656).
- [13] B. de Neeve, T.-L. Nguyen, T. Behrle, and J. P. Home, Error correction of a logical grid state qubit by dissipative pumping, *Nat. Phys.* **18**, 296 (2022).
- [14] V. G. Matsos, C. H. Valahu, T. Navickas, A. D. Rao, M. J. Millican, X. C. Kolesnikov, M. J. Biercuk, and T. R. Tan, Robust and deterministic preparation of bosonic logical states in a trapped ion, *Phys. Rev. Lett.* **133**, 050602 (2024).
- [15] M. V. Larsen, J. E. Bourassa, S. Kocsis, J. F. Tasker, R. S. Chadwick, C. González-Arciniegas, J. Hastrup, C. E. Lopetegui-González, F. M. Miatto, A. Motamedi, *et al.*, Integrated photonic source of Gottesman-Kitaev-Preskill qubits, *Nature (London)* **642**, 587 (2025).
- [16] J. Hastrup, M. V. Larsen, J. S. Neergaard-Nielsen, N. C. Menicucci, and U. L. Andersen, Unsuitability of cubic phase gates for non-Clifford operations on Gottesman-Kitaev-Preskill states, *Phys. Rev. A* **103**, 032409 (2021).
- [17] S. Konno, W. Asavanant, K. Fukui, A. Sakaguchi, F. Hanamura, P. Marek, R. Filip, J.-I. Yoshikawa, and A. Furusawa, Non-Clifford gate on optical qubits by nonlinear feedforward, *Phys. Rev. Res.* **3**, 043026 (2021).
- [18] A. L. Grimsmo, J. Combes, and B. Q. Baragiola, Quantum computing with rotation-symmetric bosonic codes, *Phys. Rev. X* **10**, 011058 (2020).
- [19] D.-L. Hu, W. Cai, C.-L. Zou, and Z.-L. Xiang, Generalized number-phase lattice encoding of a bosonic mode for quantum error correction, *Nat. Commun.* **16**, 1 (2025).
- [20] B. Q. Baragiola, G. Pantaleoni, R. N. Alexander, A. Karanjai, and N. C. Menicucci, All-Gaussian universality and fault tolerance with the Gottesman-Kitaev-Preskill code, *Phys. Rev. Lett.* **123**, 200502 (2019).
- [21] S. Singh, B. Royer, and S. M. Girvin, Towards non-Abelian quantum signal processing: Efficient control of hybrid continuous- and discrete-variable architectures, [arXiv:2504.19992](https://arxiv.org/abs/2504.19992).
- [22] P. Campagne-Ibarcq, A. Eickbusch, S. Touzard, E. Zalys-Geller, N. E. Frattini, V. V. Sivak, P. Reinhold, S. Puri, S. Shankar, R. J. Schoelkopf, *et al.*, Quantum error correction of a qubit encoded in grid states of an oscillator, *Nature (London)* **584**, 368 (2020).
- [23] B. Royer, S. Singh, and S. M. Girvin, Stabilization of finite-energy Gottesman-Kitaev-Preskill states, *Phys. Rev. Lett.* **125**, 260509 (2020).
- [24] F. Nathan, L. O’Brien, K. Noh, M. H. Matheny, A. L. Grimsmo, L. Jiang, and G. Refael, Self-correcting Gottesman-Kitaev-Preskill qubit and gates in a driven-dissipative circuit, *PRX Quantum* **6**, 030352 (2025).
- [25] L. O’Brien, G. Refael, and F. Nathan, Exponentially robust non-Clifford gate in a driven-dissipative circuit, [arXiv:2507.19713](https://arxiv.org/abs/2507.19713).
- [26] J. J. Sakurai, *Modern Quantum Mechanics*, revised ed. (Addison-Wesley, Reading, MA, 1994).
- [27] M. H. Shaw, A. C. Doherty, and A. L. Grimsmo, Stabilizer subsystem decompositions for single- and multimode Gottesman-Kitaev-Preskill codes, *PRX Quantum* **5**, 010331 (2024).
- [28] B. W. Walshe, B. Q. Baragiola, R. N. Alexander, and N. C. Menicucci, Continuous-variable gate teleportation and bosonic-code error correction, *Phys. Rev. A* **102**, 062411 (2020).

- [29] J. Conrad, Twirling and Hamiltonian engineering via dynamical decoupling for Gottesman-Kitaev-Preskill quantum computing, *Phys. Rev. A* **103**, 022404 (2021).
- [30] M. H. Shaw, A. C. Doherty, and A. L. Grimsmo, Logical gates and read-out of superconducting Gottesman-Kitaev-Preskill qubits, [arXiv:2403.02396](#).
- [31] I. Tzitrin, J. E. Bourassa, N. C. Menicucci, and K. K. Sabapathy, Progress towards practical qubit computation using approximate Gottesman-Kitaev-Preskill codes, *Phys. Rev. A* **101**, 032315 (2020).
- [32] Indeed, this argument incorrectly predicts the level of noise bias that would be optimal compared to our more detailed analysis in Appendix F 3 and Sec. VI. Moreover, repeating the argument for the S gate suggests that it suffers from the same problems as the T gate; however, this is not the case, and the error spread due to the S gate can be corrected for exactly if one has access to ideal QEC [30].
- [33] S. Glancy and E. Knill, Error analysis for encoding a qubit in an oscillator, *Phys. Rev. A* **73**, 012325 (2006).
- [34] T. Matsuura, N. C. Menicucci, and H. Yamasaki, Continuous-variable fault-tolerant quantum computation under general noise, *Nat. Commun.* **17**, 1709 (2026).
- [35] D. T. Le, A. Grimsmo, C. Müller, and T. M. Stace, Doubly nonlinear superconducting qubit, *Phys. Rev. A* **100**, 062321 (2019).
- [36] M. Rymarz, S. Bosco, A. Ciani, and D. P. DiVincenzo, Hardware-encoding grid states in a nonreciprocal superconducting circuit, *Phys. Rev. X* **11**, 011032 (2021).
- [37] J. Hastrup, K. Park, J. B. Brask, R. Filip, and U. L. Andersen, Measurement-free preparation of grid states, *npj Quantum Inf.* **7**, 17 (2021).
- [38] X. C. Kolesnikow, R. W. Bomantara, A. C. Doherty, and A. L. Grimsmo, Gottesman-Kitaev-Preskill state preparation using periodic driving, *Phys. Rev. Lett.* **132**, 130605 (2024).
- [39] B. M. Terhal and D. Weigand, Encoding a qubit into a cavity mode in circuit QED using phase estimation, *Phys. Rev. A* **93**, 012315 (2016).
- [40] B. Royer, S. Singh, and S. Girvin, Encoding qubits in multi-mode grid states, *PRX Quantum* **3**, 010335 (2022).
- [41] G. Pólya, Ueber ganzwertige ganze Funktionen, *Rend. Circ. Matem. Palermo* **40**, 1 (1915).
- [42] D. Singmaster, On polynomial functions (mod m), *J. Number Theory* **6**, 345 (1974).
- [43] It is worth noting that in principle the lexicographic ordering does not necessarily guarantee optimal performance of the minimal polynomial phase gate across *all* values of Δ . This could happen, for example, if a polynomial $P(x)$ has a smaller leading coefficient than $Q(x)$ and thus $P(x) < Q(x)$, but $P(x)$ has larger coefficients than $Q(x)$ in other positions. We did not, however, find any specific examples of this for two polynomials that implement *the same* logical gate, but we do find it for some polynomials that implement different logical gates (see, for example, T_3 and $\sqrt{T}$ in Sec. VI).
- [44] X. C. Kolesnikow, T. B. Smith, F. Thomsen, A. Alase, and A. C. Doherty, Protected phase gate for the 0 - π qubit using its internal modes, *PRX Quantum* **7**, 010306 (2026).
- [45] We use the term “channel” loosely here because $\mathcal{E}$ is not trace preserving.
- [46] Technically, what we need to ensure is that the *eigenvalues* of the covariance matrix go to zero as Δ goes to zero, because it is the maximum eigenvalue that determines the maximum “spread” of the probability distribution. This is guaranteed, however, because both the trace and determinant of Σ are positive and go to zero as $\Delta \rightarrow 0$.
- [47] V. V. Albert, K. Noh, K. Duivenvoorden, D. J. Young, R. T. Brierley, P. Reinhold, C. Vuillot, L. Li, C. Shen, S. M. Girvin, B. M. Terhal, and L. Jiang, Performance and structure of single-mode bosonic codes, *Phys. Rev. A* **97**, 032346 (2018).
- [48] F. Rozpędek, K. P. Seshadreesan, P. Polakos, L. Jiang, and S. Guha, All-photonics Gottesman-Kitaev-Preskill-qubit repeater using analog-information-assisted multiplexed entanglement ranking, *Phys. Rev. Res.* **5**, 043056 (2023).
- [49] M. A. Nielsen, A simple formula for the average gate fidelity of a quantum dynamical operation, *Phys. Lett. A* **303**, 249 (2002).
- [50] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, Gaussian quantum information, *Rev. Mod. Phys.* **84**, 621 (2012).
- [51] O. Hahn, G. Ferrini, and R. Takagi, Bridging magic and non-Gaussian resources via Gottesman-Kitaev-Preskill encoding, *PRX Quantum* **6**, 010330 (2025).
- [52] S. Roman, The formula of Faa di Bruno, *Am. Math. Mon.* **87**, 805 (1980).
- [53] P.-O. Löwdin, On the non-orthogonality problem connected with the use of atomic wave functions in the theory of molecules and crystals, *J. Chem. Phys.* **18**, 365 (1950).