

End-to-end switchless architecture for fault-tolerant photonic quantum computing

Paul Renault, Patrick Yard, Raphael C. Pooser, Miller Eaton, and Hussain Asim Zaidi

QC82 Inc., 7757 Baltimore Ave, College Park, MD 20740

Photonics presents one of the most promising approaches to large-scale quantum computation with millions of qubits and billions of gates, owing to the potential for room-temperature operation, high clock speeds, miniaturization of photonic circuits, and repeatable fabrication processes in commercial photonic foundries. We present an end-to-end architecture for fault-tolerant continuous-variable (CV) quantum computation using only passive on-chip components that can produce photonic qubits above the fault tolerance threshold with probabilities above 90%, and encodes logical qubits using physical qubits sampled from a distribution around the fault tolerance threshold. By requiring only low photon number resolution, the architecture enables the use of high-bandwidth photodetectors in CV quantum computing. Simulations of our qubit generation and logical encoding processes show a Gaussian cluster squeezing threshold of 12 dB to 13 dB. Additionally, we present a novel magic state generation protocol which requires only 13 dB of cluster squeezing to produce magic states with an order of magnitude higher probability than existing approaches, opening up the path to universal fault-tolerant quantum computation at less than 13 dB of cluster squeezing.

Paul Renault: These authors contributed equally to this work

Patrick Yard: These authors contributed equally to this work

Raphael C. Pooser: These authors contributed equally to this work

Hussain Asim Zaidi: hussain@qc82.tech

1 Introduction

Quantum computation promises the ability to exponentially speed up certain problems intractable to classical computers including simulations of large quantum systems, an observation first formalized by Richard Feynman [1]. Noisy intermediate scale quantum computers [2] with ≈ 100 physical qubits and ≈ 100 to 1000 quantum gates have been demonstrated on a variety of physical platforms [3, 4, 5, 6], but it is widely expected that the computers must scale to millions of physical qubits and billions of operations to be useful for commercially valuable quantum simulations [7, 8, 9]. Matter based approaches face hurdles to get to such a computational volume, owing to the hardware requirements of low temperatures, slow gate rates, or lossy conversions between matter and photons [10, 4, 6, 5, 11]. Photonics provides a promising path towards large-scale quantum computing with scalable generation and manipulation of qubits at room temperature, high-speed measurements, miniaturization using photonic integrated circuits, and repeatable low-loss mass manufacturing at commercial foundries.

Photonic quantum information can be encoded into either discrete quantum degrees of freedom of individual single photons, known as the discrete variable (DV) approach [12, 13], or in continuous-valued orthogonal quadratures of the electric field, known as the continuous variable (CV) approach [14, 15]. CV quantum photonics brings an advantage over DV approaches, namely the deterministic generation of entanglement over large photonic quantum states [16, 17]. Millions of entangled optical modes have been experimentally generated in CV systems [18]. This makes CV photonics ideal for measurement-based QC (MBQC) implementations [19], in which quan-

tum computation is performed via single-qubit measurements on a large entangled quantum state. In photonics, single modes or collections of modes in the entangled state act as the analogues of qubits in matter-based approaches.

The key concept to enable quantum computation with large-scale entangled states, or in general with a large number of qubits, is that of fault tolerance. *Fault-tolerant quantum computing* (FTQC) is an approach to QC in which a computational (logical) qubit is encoded in many physical qubits on which quantum error correction (QEC) protocols are employed to correct for errors such as decoherence and photon loss [20, 21, 22]; If the probability of physical errors is below a certain value determined by the QEC protocol, called the *fault tolerance threshold*, the probability of logical error can be exponentially suppressed. There exist no direct fault-tolerant protocols for quadrature encoded continuous-variable quantum computing (CVQC) [23]. An alternative approach is to discretise the CV space into qubits [23, 17].

The most promising qubits for CV FTQC are the *Gottesman-Kitaev-Preskill* (GKP) qubit [20] and the *magic states* [24]. Native logical gates are implemented with Gaussian operations on GKP qubits, while universal quantum computation is possible with the ability to create magic states. In fact, using only CV Gaussian resources combined with GKP states, one has all the necessary ingredients to achieve universal CVQC [25]. Therefore, much effort in CVQC has gone into devising scalable methods of producing high quality GKP qubits (that is, with high effective squeezing; see Sec. 2), but producing these states with high success rates in photonics has remained a challenge (see Sec. 5). After obtaining high quality GKP and magic qubits, CV architectures achieve fault tolerance by concatenating a second qubit-level error correcting code with the GKP states as the physical qubits. To implement these codes with optical fields, photonic switches were proposed either to increase the low success probabilities of GKP state generation [26, 27] or for quantum error correction [28] (see Sec. 8). Great progress has been made in lowering the noise and losses in photonic switches, but the loss levels are two orders-of-magnitude higher than passive photonics [29], making re-

moval of switches from the architecture a desirable goal.

With this context, we present four contributions in synthesizing an end-to-end CV FTQC architecture. First, we provide a protocol that produces fault-tolerant CV photonic qubits – GKP qubits – at state-of-the-art probabilities of more than 90%. Notably, the architecture makes use of all physical qubits, both above and below the FTQC threshold, to construct logical qubits. Second, we provide methods to generate GKP qubits on an integrated switchless photonic chip, opening up the possibility of low-loss manufacturability of the photonic chips with passive components. Our methods for state generation require only low photon number resolution (PNR) for which either superconducting microstrip single photon detectors (SMSPDs) [30] or room temperature photon number resolving detectors [31] would suffice. Third, we present protocols for magic state generation with state-of-the-art success probability, an order of magnitude better than existing methods, for universal FTQC with feasible resource requirements. Fourth, we present the results of our quantum error correction simulations including GKP resource generation costs. We arrive at the *gaussian squeezing* threshold, the quantum noise metric in CVQC [17, 32], of 12 dB to 13 dB for fault tolerance, and distillable magic state success probability of 4.8% also at 13 dB. To our knowledge, the presented simulations are the first end-to-end CVQC simulations incorporating a realistic process producing GKP states with a range of effective squeezing values for fault-tolerant operation.

In the following sections, we start by providing an overview of photonic CVQC. We then describe our architecture before proceeding to discussing the details and physics of each architectural component, and conclude by providing the results of the fault tolerance simulations.

2 CVQC Toolkit and Non-Gaussian States

In photonic CV MBQC, the set of entangling operations, non-destructive gates, and destructive measurements come from the CV photonics toolkit [33], the most prominent components

of which are the squeezing operations, linear optics, quadrature displacements, homodyne measurements, and photon number resolving measurements. In the following, we define the key terms, including entangled states, measurement operations, and non-Gaussian modes, which form the basic building blocks of our architecture.

The entangled state often used in photonic CV MBQC is the *cluster state*, canonically defined as Gaussian-squeezed modes entangled using controlled-phase gates (C_Z) [34]. While CVQC was initially investigated with Gaussian cluster states requiring inline squeezing due to the use of controlled-phase gates [35, 36, 19], an alternative formalism exists where inline squeezing is replaced by offline squeezing and passive interferometers [37, 38, 39], resulting in a *macronode cluster state*, where collections of physical optical modes can be treated as a single mode on an underlying lattice [40]. An example of macronode cluster is the multimode state called a dual rail quantum wire, generated using the two-mode-squeezing operation implemented in a non-linear crystal followed by passive beamsplitters [41].

The name squeezing stems from the shape of the Wigner quasiprobability distribution of the conjugate photonic quadratures (denoted p and q), in which the distribution for one quadrature has a narrower width compared to a coherent state, while the other quadrature has a broader distribution. The term also refers to the quantum noise on a measurement along one or a combination of the quadrature axes in phase space. In CV MBQC, because all gates are carried out via measurement operations, the noise on these gates is parametrized by the squeezing observed in each measurement. Thus, physical gate noise is directly tied to the squeezing level associated with each mode being measured. Ideal Gaussian squeezed states (infinitely squeezed) are denoted by $|m\rangle_p$, where m is the eigenvalue of the quadrature p . For the results presented in this paper, all simulations are run with finitely squeezed states (i.e., resulting from applying the finite squeezing (single-mode or two-mode) operator to vacuum). FT thresholds in terms of squeezing are important for CV FTQC as single and two-mode squeezing forms the fundamental resource for CVQC [42, 43].

The operations used to project a quantum

state onto quadratures eigenstates are called *homodyne measurements* [44]: a homodyne measurement with outcome m on quadrature p projects the measured mode onto the state $|m\rangle_p$. An important property of homodyne measurements chosen with appropriate bases is that they can be used to either delete a node from a cluster state or teleport the quantum information from one node to another [42], allowing one to shape cluster states during quantum computation.

While deterministic and scalable Gaussian entanglement generation gives us a substrate quantum state to use in CVQC, and homodyne measurements can be used to shape the cluster state and implement quantum gates, universal CVQC requires additional resources to enable quantum computations that are not efficiently classically simulable [45]. One prominent operation to enable universal CVQC is photon subtraction [46], which can be used to implement gates consisting of arbitrary powers of the quadrature operators on squeezed states. Experimentally, in quantum optics, the most common method for performing photon subtraction involves using a highly reflective beamsplitter, with a photon detector—capable or not of resolving the number of detected photons—placed in the transmission arm [46, 47]. Throughout this article, whenever the photon subtraction process is mentioned, we refer to this experimental setup. One of the key applications of these higher-order gates is the production of *non-Gaussian states* in an entangled cluster, which along with homodyne detections and displacements, give us the ability to apply quantum error correction and operate the universal quantum computer above the FT threshold [17].

The fundamental non-Gaussian resource states in our architecture are the squeezed cat states $|\pm\alpha, r\rangle$, which are superpositions of displaced squeezed states, i.e., $|\pm\alpha, r\rangle \sim (D(\alpha) + D(-\alpha))\mathcal{S}(r)|0\rangle$, where D is the displacement operator in the phase space, $D(\alpha) = \exp(\alpha a^\dagger - \alpha^* a)$, \mathcal{S} is the squeezing operator, $\mathcal{S}(r) = \exp(\frac{1}{2}(r^* a^2 - r a^{\dagger 2}))$, and a^\dagger and a are the creation and annihilation operators, respectively. Real α results in q quadrature displacements (D_q), while imaginary values result in p displacements (D_p). Cat states are parametrized by squeezing (r) and displacement (α), which we

take to be real-valued in the following results. The Wigner function for a squeezed cat state is shown in Fig. 1. While FTQC codes exist with cat states as the logical qubits, these require the experimentally-challenging parity check operator [48], limiting the role of cat states in photonic FTQC to that of resource states which we consume to produce qubits.

GKP states, and more generally grid states, were first proposed in [Ref. [20]], with a considerable amount of effort spent studying their use in fault-tolerant architectures in recent years [49, 50, 51]; a typical Wigner function is shown in 1. The sensor state, denoted $|S\rangle$, is a symmetric grid state defined as the +1 eigenstate of the displacements $D_q(\sqrt{2\pi})$ and $D_p(\sqrt{2\pi})$, while a GKP qubit is the +1 eigenstate of the displacements $D_q(2\sqrt{\pi})$ and $D_p(\sqrt{\pi})$, where we have set $[a, a^\dagger] = 1$ [20, 51] (see Appendix Sec. 11.1 for more details). Interference between sensor states using passive linear optics can be used to produce entangled GKP Bell pairs [52].

Ideal GKP states have wavefunctions that are a series of delta functions in the q -basis (or p -basis) of the continuous-variable (q, p) phase space of a bosonic mode, and are therefore unphysical (see Appendix 11.1 for ideal GKP wavefunctions). Physically-relevant approximations to the ideal GKP states can be obtained by applying the damping operator to the ideal state [17, 53]. The width of the peaks provides an *effective squeezing* parameter, which describes the closeness of a given GKP state to the ideal state. The probability of physical gate errors in CVQC depends on the effective squeezing of the underlying GKP states, which denotes the noise level of measurements associated with the qubit compared to the standard noise of a coherent state [54]; minimizing this noise corresponds to reduced physical error rates. For any given density matrix ρ and the desired displacement operator with displacement u , we estimate the effective squeezing of the output state as $\Delta_D = \frac{1}{u} \sqrt{\ln |\text{Tr} D(u)\rho|}$ where Tr is the trace operation [54]. The quality of a realistic GKP state for FTQC is quantified most readily by the effective squeezing parameter since the probability of error in the logical X or Z operations is bounded by $P < \frac{2\Delta}{\pi} \exp -\pi/(4\Delta^2)$, where Δ is the effective squeezing in the re-

spective quadrature [54]. Note that the GKP state can have different effective squeezing in the two quadratures owing to different parameters at play in the generation protocols. A final important non-Gaussian state in our architecture is the magic state. These are essential for universal quantum computation by providing an implementation of a complete gate set. In CVQC, magic states are also grid states; their Wigner functions contain periodic structure (see Fig. 1; also Appendix Sec. 11.1), and they can be derived from CV GKP states [25]. An important metric is the distillation threshold of magic states, which quantifies the fidelity above which many copies of noisy magic states can be used to distill a high fidelity magic state useful for applying fault-tolerant non-Clifford gates, e.g., the T gate [24].

3 Architecture and Design Overview

Our photonic QC architecture contains three key components for physical qubit generation (see Fig 2): cat state generation via the Photon-counting-Assisted Node-Teleportation Method (PhANTM) [55] (see also section 4), an adaptive breeding protocol (detailed in section 5), and a magic state generation protocol (described in section 6). The logical layer consists of two major components: logical qubit generation, which consists of distributing physical qubits as the nodes within the Raussendorf-Harrington-Goyal (RHG) lattice [56]; and the quantum error correction (QEC) encoding/decoding system, which is also responsible for running logical instructions. All of the physical qubit generation and RHG lattice generation components are possible on a set of integrated photonic chips without active photonic switches or variable beamsplitters on the quantum modes to minimize loss and noise. While fast electro-optic modulation is needed on local oscillators during homodyne measurements, the local oscillator power can be corrected to account for any losses.

We use cluster engineering with time multiplexing and photon-number-resolving detection to prepare cat states, followed by adaptive breeding to create GKP states using homodyne measurements. Magic states are produced by applying CV QEC on cat states, using bred GKP states

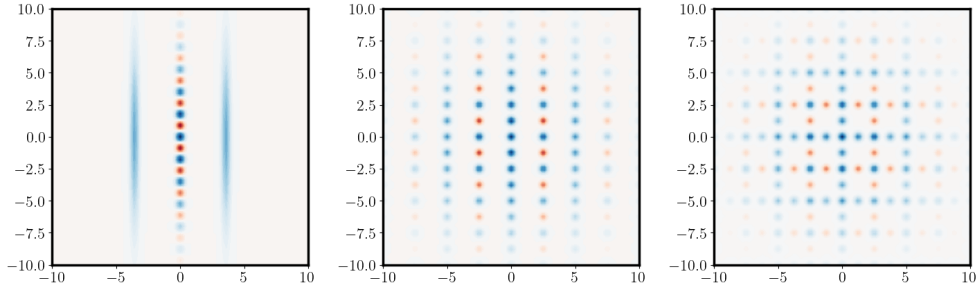


Figure 1: Wigner functions of squeezed cat state, GKP zero qubit $|0\rangle_G$, and magic H state $|H\rangle$. The horizontal and vertical axes are the q and p quadrature axes, respectively. Each of the states has squeezing of approximately 13dB in the q direction (cat state) or both the q and p directions (GKP and magic states). Red peaks correspond to negative values of the Wigner distribution, while blue peaks are positive. The color scale is otherwise arbitrary and is meant only to notionally represent the location of probability density.

as resource states. We generate the RHG lattice using static linear optics and perform syndrome extraction and logical operations by single qubit homodyne measurements on logical qubits within the surface code. Classical compute resources are used to implement the user interface, code compilation, and transpilation to local oscillator phase controls, which implement all gate controls in the architecture. Homodyne detection measurement results are sent to the logical instruction and QEC system where syndrome decoding and error tracking are performed. Any necessary modifications to the measurement results as a consequence of error correction are applied before using measurement outcomes to compute the algorithm results. When online changes to future measurements are required, the result of error detection is sent to the phase modulators which control the local oscillator phases that set the measurement bases.

The substrate to create GKP and magic states is a cluster state formed from two-mode squeezed light and passive beamsplitters, where each node is a photonic mode characterized by the quadratures q and p , and features nearest-neighbor entanglement between the nodes. The time-multiplexed cluster state in our architecture is indexed by spatial and optical frequencies along with temporal modes (see Fig. 4 and later figures on cluster state engineering). To produce cat states our architecture uses PNR detection at specific optical frequencies and timesteps. Specifically, Schrödinger cat states are produced using the Photon-counting-Assisted Node-Teleportation Method (PhANTM) [55]. PhANTM is well-suited to our time-frequency-space multiplexed architecture, as individual nodes of a cluster state

can be made non-Gaussian with photon subtraction, and the non-Gaussianity can be teleported through a cluster state with homodyne measurements. This allows us to repeat the PhANTM process over multiple time steps to increase the amplitude of the generated cats and make them suitable for high-probability GKP generation, in contrast to single-shot GKP generation methods without time or frequency multiplexing [26, 57].

The cat states, which are embedded in the cluster state, are squeezed using teleportation-based squeezing gates [58] and then “bred” into either sensor states for qubit production or GKP states for magic state production [25]. Switching between the two output states is achieved by changing the required amplitude of the input cat states. Given that our protocols produce cat states with probabilistic squeezing and amplitude located at the nodes of a cluster state, we use an *adaptive breeding* protocol to produce GKP qubits or sensor states. In our adaptive protocol, cat states of various multiples of the desired grid state spacing are used including a single mode squeezed vacuum that can be substituted for lower quality cat states when desired (see Algorithm 1).

These fundamental components can all be implemented in passive photonic integrated circuits, with squeezed light, sensor states, magic states, and the RHG lattice all produced on integrated chips. The chips integrate linear optics components, such as beamsplitters and phase shifters, and nonlinear components such as $\chi^{(3)}$ media, as shown in Fig. 3. A squeezing resonator is bidirectionally pumped by pumps centered at ν_0 with $\nu_{p1} = \nu_0 + \Delta\nu/2$ and $\nu_{p2} = \nu_0 - \Delta\nu/2$,

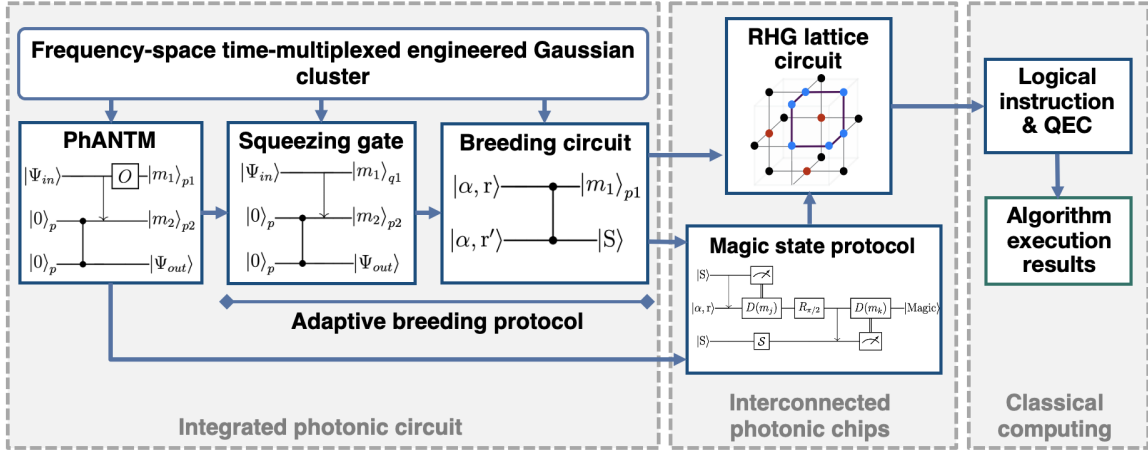


Figure 2: Conceptual overview of the creation of GKP states, magic states, MBQC-based lattices, and QEC to enable FTQC. Repeating photon subtraction and teleportation along a cluster state generates cat states with a high success probability and low photon number resolution. The cats are then squeezed to produce cats with deterministic amplitudes which are bred together to create GKP states without the need for post-selected measurements. All steps happen on the cluster state. The GKP states and cat states are used together to create magic states. The architecture produces H -type magic states in this configuration. Notations used in circuits are defined in main text. Additionally, S and R stand respectively for squeezing and rotation operators. Note that for brevity we use the notation for infinitely squeezed Gaussian states in the circuits, but in practice they are replaced by finitely squeezed states in the simulations. Downward arrows in circuits are beamsplitters while vertical lines with circular caps are C_Z gates.

where $\Delta\nu$ is the free spectral range (FSR) of the resonator. This creates two squeezed combs each centered about the respective pump. Interfering the two squeezed combs at a balanced beamsplitter results in the formation of a 1D dual-rail quantum wire [59]. Subsequently, homodyne measurement are performed locally within this cluster to reshape it in preparation to apply PhANTM or adaptive breeding at different time steps. If the local oscillator phase is aligned with the q quadrature then the node is removed whereas if the phase is that of the p quadrature, the quantum information is teleported to a neighboring node, with an additional $\pi/2$ rotation applied [19]. Each frequency mode needs to be separated from the optical frequency comb using frequency filters to then be detected. These filters are schematically shown as ring resonators, which are among the most compact and lowest-loss options for frequency separation [60] negligibly impacting the pass-through frequency modes, and demonstrating very high transmission efficiencies of the filtered mode. The delay is used for time multiplexing before a final balanced beamsplitter is used to interfere frequency modes in (b) with the delayed modes in (a) to produce the final cluster state. With this resource state CVQC

operations can be applied to generate cat states by performing PhANTM, squeezed cats using a squeezing gate, or GKP states by adaptive breeding.

The architecture implements quantum error correction (QEC) by embedding the bred GKP sensor states from the photonic integrated circuit in Fig. 3 into the RHG cluster state [56], the MBQC analog of the surface code. Large scale entanglement is generated using a macronode based approach [27] where each qubit of the RHG lattice is replaced by four optical modes. Entanglement links in the RHG lattice are replaced by GKP Bell pairs and the final cluster state is then generated by interfering all modes of the macronode with passive linear optics.

Logical gates can be implemented on the RHG cluster state using lattice surgery [61] and magic states. Magic states can be embedded in the macronode RHG lattice by replacing a GKP Bell pair with the result of interfering a magic state with squeezed vacuum [27]. More detail on the lattice construction is given in Sec. 7.

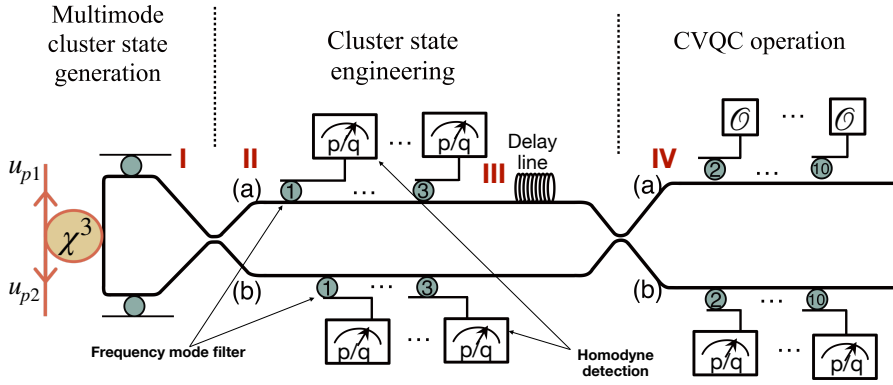


Figure 3: Chip design of GKP generator to create multiple cat states in parallel from a time-frequency encoded Gaussian cluster state. The required squeezing is produced via four wave mixing in a ring resonator (χ^3). Cat states are bred within the same chip to generate GKP state. The shape of the cluster state can be adapted as function of the operation applied in stage **IV**. u_{p1} and u_{p2} stand for the optical pump beams at frequency respectively ν_{p1} and ν_{p1} . Boxes with p/q measurement indicate circuits where homodyne detection and corrective displacement are performed while boxes denoted \mathcal{O} represent circuit where multiple photons subtraction are applied as well as displacement and homodyne detection. The green circles denote frequency filters that transmit given frequency modes. The labels within the circles correspond to the frequencies mode number consistent with those indicated in Fig. 4.

4 Cat States on Dual-Rail Quantum Wire

Cat states are a fundamental resource in our CVQC architecture. They are used as a resource state to generate both sensor states via breeding and magic states via implementing CV-QEC on input cat states. Methods for creating cat states in CV photonics need at a minimum a combination of Gaussian and non-Gaussian resource states, or photon number detection along with Gaussian resource states. A method was proposed to create cat states by applying PNR on squeezed vacuum using machine learning to control some of the experimental parameters [62]. This method is efficient for state generation, requiring only a few iterations of photon detection, but requires high photon number resolution. Another proposal used high-photon-number Fock states (with 10 or more photons) along with photon number resolution to create cat states deterministically [63]. Creating high photon-number Fock states presents a high experimental challenge [64]. In PhANTM [55], the method used in the present architecture, applications of probabilistic photon subtraction and teleportation build up non-Gaussianity over time. This requires a Gaussian cluster as a resource to repeatedly apply the method. One of the key advantages of PhANTM lies in its ability to generate sufficiently large cat states that can subsequently

be bred while distributing the total number of photons subtracted over multiple repetitions of the operation.

Fig. 4 shows the steps and cluster engineering using homodyne measurements, leading up to many copies of cats generated in parallel from a single frequency-space-time entangled cluster state. The frequencies are denoted by ν or arabic numerals, spatial rails are denoted by (a) and (b), and time steps are denoted by τ_i . Starting with many pairs of two-mode entangled states in **I** interfered at the first beamsplitter of Fig. 3, we arrive at the entangled frequency chain also known as the dual-rail quantum wire shown in Fig. 4 **II**. We apply p and q homodyne measurements in region **II** of Fig. 3 and Fig. 4 to convert a dual rail wire into multiple wires in frequency-space as shown in region **III** of Fig. 4. Going through a time delay then entangles the two-mode wires in time to create many dual-rail wires in frequency-space-time as shown in Fig. 4, on which PhANTM can be applied in parallel, e.g., on modes $2_{a,b}$ and $10_{a,b}$ in Fig. 4 at time τ_i . Since the cluster state modes that undergo homodyne detections or PhANTM are pre-determined, no active switching is required.

The application of PhANTM on the dual-rail wire in Fig. 4 **IV** corresponds to repeating the circuit in Fig. 5 in time. The rotation on the output mode can be manipulated through the phase

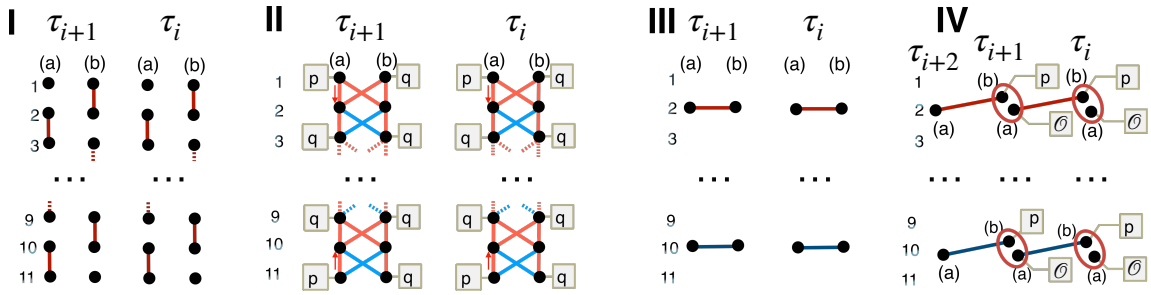


Figure 4: Cluster preparation for PhANTM application and parallel dual-rails. **I**, **II**, **III** and **IV** refer to locations on the chip in Fig. 3. In stages **I** to **III**, the cluster state is generated and then reshaped using homodyne measurements to form dual rails independent of each other. In **IV**, PhANTM is applied as the circuit in Fig. 5 shows: at the time step τ_i modes ν_a and ν_b are respectively mode 2 and 1 in the circuit. Then the output mode in the dual rail is the mode ν_b of the τ_{i+1} time step. Dark blue and maroon segment represent effective C_Z gates with weights of respectively -1 and $+1$, coming from applications of two-mode-squeezing and beamsplitter operations, while red circles represent the second beamsplitters in Fig. 3, where a phase can be applied to correct any rotation induced by cluster engineering. Orange and blue segments represent C_Z gates with respective weights of $+1/2$ and $-1/2$. The red arrows in **II** show the direction of the teleportation.

of the beamsplitter and homodyne detection basis. The beamsplitter is defined by two parameters, θ and ϕ , which are related to the reflectivity and transmissivity such that $r = e^{i\phi} \sin(\theta)$ and $t = \cos(\theta)$. We have chosen the phase $\phi = \pi/2$ in Fig. 5 such that a single application of the circuit at τ_i produces an output cat with the same orientation as the input cat, allowing repeated applications of PhANTM at τ_{i+1} and later timesteps. Note that there are two homodyne measurements applied in a unit round of PhANTM (one on ν_a and another on ν_b) before the output mode is ready for the next round. Also note that given the beamsplitters and phases involved in the circuit in Fig. 5, the bases used for either mode teleportation or deletion differ from the usual p or q bases for a canonical cluster state [65]. The effective C_Z gate in Fig. 5 results from the application of two-mode squeezing, beamsplitters, and homodyne measurements as shown in Fig. 4. The modes that undergo operation \mathcal{O} in Fig. 3 have multiple photon subtractions followed by homodyne detection applied to them, while the rest of the modes in **IV** have p basis homodyne measurement applied to them. Active displacements are needed in between repeated applications of PhANTM due to random homodyne measurement results. The value of the displacement is the output of the homodyne detector fed directly back to a standard displacement circuit with a local oscillator for the next temporal mode [66]. The timescale of these active displacements must be shorter than the time delay between successive

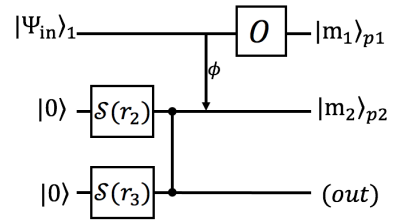


Figure 5: Application of PhANTM at stage **IV**. The top two modes are entangled through a beamsplitter (downward arrow) while the bottom two are entangled through a C_Z gate (a line with two nodes at the ends) which is implemented by two-mode squeezing, beamsplitter, and homodyne measurements for cluster engineering. Conceptually, the C_Z gate happens first, followed by the beamsplitter, and then the measurements from top to bottom. O stands for multiple photon subtraction operations.

temporal modes in the cluster state.

Monte Carlo simulation of several rounds of PhANTM were conducted using the unit cell shown in Fig. 5, sampling the number of photons detected in each photon subtraction. The parameters of the cat states that we aim to optimize are their size and squeezing, both of which are stochastic. The larger these parameters, the higher the effective squeezing of the resulting GKP states, leading to lower error rates for QEC codes. To facilitate analysis and comparison across various conditions, a single parameter is defined: the amplitude of the cat state with zero squeezing (α_c , also see Sec. 5). Fig. 6 shows the mean of α_c as a function of cluster squeezing for different number of PhANTM

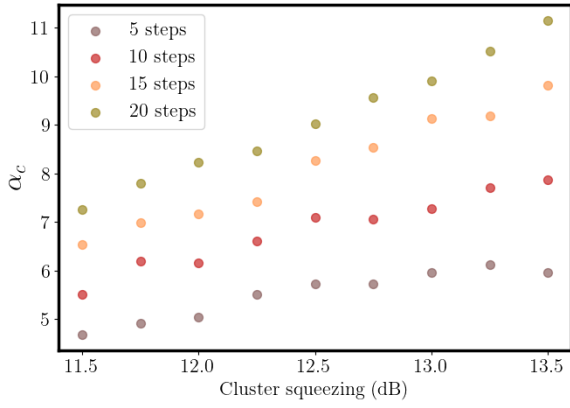


Figure 6: Mean α_c as a function of cluster squeezing. Each color corresponds to a given number of PhANTM steps. The means are estimated from 500 Monte Carlo iterations of PhANTM. Details of these simulation are given in Appendix. The plot shows the positive correlation of α_c with cluster squeezing and the total number of PhANTM steps.

rounds assuming $r_2 = r_3$ (see also Sec. 11.3 in Appendix). In these simulations, eight sequential photon subtraction attempts were performed at each time step. The beamsplitter angles for each photon subtraction are chosen such that no more than 10 photons are detected per PNR detector. The subtraction probability goes down for later subtraction events, as continued subtraction attempts de-amplify the quantum signal. We apply a reflectivity gradient (higher angle in later subtraction attempts) to ensure that the distribution of subtracted photons among the PNR detectors is roughly uniform (see Sec. 11.4 in Appendix). Increasing the cluster squeezing enhances the potential number of photons that can be subtracted in a single subtraction event, resulting in cats with larger amplitudes. Furthermore, increasing the number of PhANTM steps raises the total number of photons subtracted across all cumulative PhANTM events, leading to larger amplitudes for cat states. The additional squeezing value also depends on the reflection and transmission coefficients of each beamsplitter utilized in the photon subtraction process.

The success probability of each step of PhANTM, and the final cat size, can be tuned given the hardware constraints of the photon number resolution ceiling and the losses in the photonic circuit resulting from repeated teleportations in time, leading to many possible optimizations.

5 Generation of GKP States

There are two main approaches to creating GKP states, which we describe below for comparison with our approach. In the first approach, single shot methods are devised to directly create GKP states. In this approach, Gaussian Boson Sampling (GBS)-based approaches to GKP state generation [49] produce probabilities of 2% to 7% with fidelities of around 0.65 for GKP states with 10 dB effective squeezing with 12 dB of Gaussian squeezing. The probabilities are of $O(10^{-5})$ if the target fidelities are close to 1. These probabilities are low for practical quantum computation. Bifurcation-based methods proposed creating 10 dB GKP states with maximum photon number resolution of 40 photons, Gaussian squeezing of up to 19.4 dB with much higher success probabilities for very high fidelity GKP states [57]. The overall probabilities, while impressive, are still far from deterministic and resolving 40 photons would require Transition Edge Sensors (TESs) working at cold cryogenic temperatures of 0.1 K and low repetition rates of 100 KHz [67]. Very high squeezing required also poses a challenge in physical implementation of the protocol. In another proposal, single-photon states are used in conjunction with GBS circuits to increase the fidelity to 0.95 and success probability from 10% to 40% [68], not including the below-unity probability of generating single photons. In the second approach GKP states are generated from a set of cat states with a fixed α and parity by teleporting the cats through each other, also known as “breeding” [54]. In a practical implementation, deterministically creating photonic cat states with the same amplitude and parity is difficult. In short, GKP generation schemes either have low success probability, require high photon number resolution, use Fock states that are difficult to prepare with high fidelities and probabilities, or impose conditions on cat state parameters that are difficult to satisfy in a hardware implementation.

We focus on creating GKP states above the FT threshold from breeding a supply of probabilistic cat states from PhANTM. Since PhANTM is a probabilistic process, fixed- α and fixed-parity cats are difficult to engineer. In addition, with a low probability the cat states produced from PhANTM have amplitudes that are too small for breeding. Finally, given a finite

amount of squeezing in the cluster state, some cats from PhANTM are too large to be squeezed to the amplitude required for breeding. In the following, we present an adaptive breeding protocol that overcomes the probabilistic nature of PhANTM to create GKP states with a high effective squeezing.

The pseudoalgorithm for adaptive breeding is shown in Algorithm 1. The algorithm takes a set of cat states embedded in a cluster state. In order to breed GKP or sensor states with the correct spacing, the cat amplitudes must be adjusted using a squeezing operator. Once the amplitudes are adjusted, the cats are simply measured in the p basis. Because the cats are embedded in a cluster state with effective C_Z operations applied between the cluster modes, squeezing and breeding is implemented via measurements only [55].

The adaptive breeding protocol takes several inputs: the number of desired breeding rounds, M ; the squeezing present in the cluster state, $r_{cluster}$; a number of input cats which depends on the number of breeding rounds, 2^M ; the input cats' unsqueezed amplitudes α , their present squeezing level after PhANTM, r_{cat} , and their parity P ; the desired GKP or sensor state spacing in phase space, ξ ; the required cat state amplitude for breeding, α_b , which depends on the spacing as $\alpha_b = \xi 2^{(M-3)/2}$ [54]; and the lower bound on cat squeezing that will determine its suitability for breeding, r_{lb} . All homodyne measurements are performed in the appropriate basis depending on whether an input cat is used in breeding or replaced with a squeezed vacuum. As a first requirement for the adaptive breeding to proceed, the amplitude α of the cats output from PhANTM must be known. The output amplitude and squeezing, r_{cat} depends on the photon subtraction vector obtained from PhANTM. The parity P of each cat also depends on the total number of photons subtracted during PhANTM. As each cat can only breed with another cat of the same parity, small displacements can be used to adjust the odd cat parities to even ones. This displacement causes an effective, known displacement in the first breeding round output, which can be kept track of but does not have to be actively fixed until breeding is complete. In general, the cat amplitudes are adjusted via squeezing operations of magnitude s_{α_b} , which depends

Algorithm1 GKP adaptive breeding algorithm

Input cluster squeezing $r_{cluster}$

Input M breeding steps

Input 2^M outputs from PhANTM, each with: unsqueezed cat amplitude α , parity P , cat squeezing r_{cat}

Input desired GKP state spacing ξ

Input desired cat α_b in [54] $\alpha_b = \xi 2^{(M-3)/2}$

Input lower limit of acceptable cat squeezing r_{lb}

Input measurement bases p and q

for each PhANTM output **do**

if P is odd **then**

displace cat to change phase by $\pi/2$

end if

compute $s_{\alpha_b} = \ln(\alpha_b/\alpha)$ and

$s_{2\alpha_b} = \ln(2\alpha_b/\alpha)$

if $s_{\alpha_b} < r_{cluster}$ and $s_{\alpha_b} > r_{lb}$ **then**

squeeze cat with $s_{\alpha_b} - r_{cat}$

teleport to next time step

else if $s_{2\alpha_b} < r_{cluster}$ and $s_{2\alpha_b} > r_{lb}$ **then**

squeeze cat with $s_{2\alpha_b} - r_{cat}$

teleport to next time step

else

replace with momentum squeezed state

end if

end for

begin Breeding Procedure:

for each output pair produced above **do**

select unmeasured modes on the cluster

state on which to teleport output bred states

breed and entangle outputs by measuring

one mode of pair with p

end for

for each unmeasured mode remaining above

do

select unmeasured modes on the cluster

state on which to teleport output bred states

breed and entangle outputs by measuring

one mode of pair with p

end for

repeat Breeding Procedure until only one

mode remains unmeasured

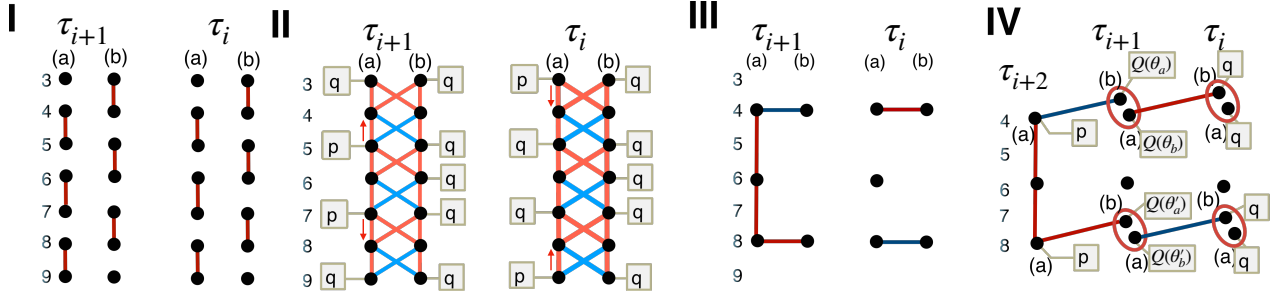


Figure 7: Cluster engineering for adaptive breeding. While the figure follows the same structure as in Fig. 4, a notable difference is the connection between dual rails of different frequency modes required here. Breeding is applied between states in modes 4_a and 8_a at the time step τ_{i+2} in **IV**. The output state of the breeding protocol is in mode 6_a . Also at stage **IV** and prior to breeding, squeezing gates are applied at time step τ_{i+1} by setting angles of the homodyne detections. Note that the physical components on the chip create the cluster state in **I** and **II**, which is then engineered for different processes in **III** and **IV** by changing homodyne measurements.

on the desired spacing ξ and the number of breeding steps M .

The extra teleportation step in the breeding protocol allows us to introduce momentum squeezed states at $r_{cluster}$ squeezing in two scenarios. If the cat amplitude produced by PhANTM is too small to be used in GKP generation such that no amount squeezing smaller than $r_{cluster}$ can be used to adjust its amplitude to α_b , it can be replaced by a momentum squeezed state before breeding. Adding a squeezed vacuum state in place of a cat increases the squeezing in one quadrature at the expense of the squeezing in the other quadrature. In the second scenario, arbitrarily large cat states can be produced in PhANTM that are too big to be squeezed to amplitude α_b needed in breeding because the squeezing required to do so would exceed $r_{cluster}$. These cats can be squeezed to an even multiple of the required amplitude for the breeding protocol to proceed ($2\alpha_b$ is used in Algorithm 1). If the squeezing of these large cats, $s_{2\alpha_b}$, is smaller than the desired lower bound, r_{lb} , then these cats can once again be replaced by momentum squeezed states (and likewise if the required squeezing would exceed $r_{cluster}$). The squeezing of cats or cluster is the dominant term that determines the effective squeezing of the GKP states in one quadrature, while the number of breeding rounds is the dominant term for effective squeezing in the other direction. Note that using a cat that is an even multiple of the required cat squeezing also has the effect of increasing the effective squeezing in one of the quadratures, giving us flexibility in trading off the squeezing in the GKP quadratures by

combining mixtures of cats with different amplitudes and momentum squeezed states.

Fig. 7 illustrates the cluster engineering method required to apply adaptive breeding in Algorithm 1 to cat states generated on different dual rails of the cluster state (see also Appendix Sec. 11.5). At stage **II**, the mode arrangement is modified compared with Fig. 4. Specifically, prior to applying any breeding operation, distinct frequency modes need to be connected via an effective C_Z interaction. This connection is shown in stage **III**. Breeding of cat states in mode 4 and 8 is applied in stage **IV** at the time step τ_{i+2} . Prior to adaptive breeding, squeezing gates are employed to ensure that the size of each cat state matches α_b or an even multiple of α_b in the pseudoalgorithm above. This operation requires one single time step and consumes two spatial modes of the same frequency as it is represented between time step τ_{i+1} and τ_{i+2} in Fig. 7. The squeezing parameter of the gate is set by the homodyne measurement angles θ_a and θ_b [58]. More details are discussed in Sec. 11.5 of the Appendix. Note that we can swap between sensor states and GKP qubits by changing the amplitudes to which the cats are squeezed.

Notice that applying the squeezing operation or replacing cat states with momentum squeezed states does not require any active switching of the quantum states. Replacement of cat states is accomplished by setting $\theta_a = \pi/2$ and $\theta_b = 0$ in homodyne measurements on the cluster (see Fig. 7). This causes the cluster mode in the next time step to be a momentum squeezed state, which is subsequently used for breeding,

removing the need for active switching with an extra teleportation step.

To quantitatively model the adaptive breeding step, the outputs of the previously described PhANTM simulations are used as input to an adaptive breeding simulation. A Monte Carlo approach is used to obtain statistics about the sensor states obtained from adaptive breeding, with 1000 samples drawn from a set of input cats that were produced with an array of settings for starting cluster squeezing and the number of photon subtraction attempts. The simulation makes use of Qutip [69] for density matrix calculations and Strawberry Fields [70] for homodyne detection simulation. The average sensor state squeezing along with their standard deviations are shown alongside FT threshold simulations in Sec. 8.

6 Generation of Magic States

Magic states are an important ingredient of universal FTQC as they allow implementation of non-Clifford operations while themselves remaining distillable with only Clifford gates [24]. A popular gate set which generates the Clifford group is $\{CNOT, H, S\}$, where $CNOT$ is the controlled NOT gate; H is the Hadamard gate, and S is the phase gate. The non-Clifford T gate usually completes the universal gate set [24]. Provided with a source of physical magic states, one can distill a resource state that enables teleportation of non-Clifford operations into the QEC substrate at arbitrarily low error rates. Improving magic state quality by increasing the fidelity with respect to any of the $|H\rangle$ or $|T\rangle$ states reduces the physical resources required for a given logical error rate in the quantum computation. A promising method to produce a GKP magic state uses GKP error correction on the vacuum state [25], implying that GKP qubits and encoded Clifford operations are sufficient for universal QC [25, 71]. However, simulations show that a reasonable success rate with existing methods requires the effective squeezing of GKP states to be extraordinarily high, approaching 20 dB (see Fig. 8), to create distillable magic states in one error correction round. In other architectures, producing magic states with practical success rates require the GKP states to have very high effective

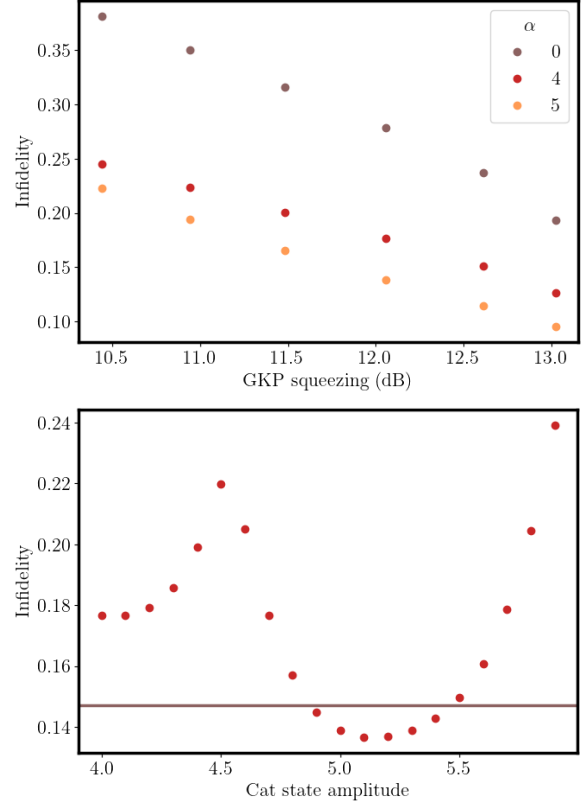


Figure 8: Top: magic state infidelity as a function of squeezing in the GKP states used as a QEC resource for three input cat amplitudes (α), where $\alpha = 0$ corresponds to the vacuum state. Bottom: magic state infidelity as a function of input cat amplitude and GKP squeezing of 13 dB. The horizontal line represents the distillation threshold for H-type states.

squeezing [25, 57], well above the FT threshold. In this section we provide a method to circumvent this issue and produce magic states with GKP resources that have effective squeezing levels near the FT threshold.

The method used to create magic states with significant probability using much lower squeezed GKP states in the present architecture is shown in Fig. 9. We replace the input vacuum with a cat state ($|\pm\alpha_4\rangle$ in Fig. 9) on which to apply QEC with GKP states. This substitution opens up the possibility to optimize the input cat amplitude, as well as all operations in the GKP breeding process prior to CV-QEC. We also replace the C_Z operations during CV-QEC with beamsplitters, which allows us to use separate photonic chip components to produce GKP and cat states.

This substitution requires a single squeezing operation to be performed on one of the GKP

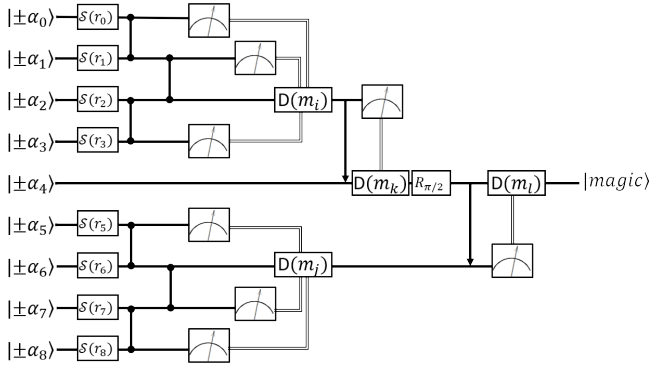


Figure 9: Magic state generation protocol. Double lines represent feed-forward operations. The gauge boxes represent p basis measurements, except for the final measurement which is in the q basis. The unsqueezed input cats labeled $|\pm\alpha_i\rangle$ can also be replaced with prior cascaded breeding steps in order to increase the quality of the bred GKP states before applying CV-QEC.

resource states, which is easily commuted to the state preparation stage before breeding the GKP state. A simple implementation of the protocol is shown in Fig. 9, resulting in a magic state generation circuit using nine cat states. This contains the equivalent of two GKP breeding rounds for each of the two GKP resource states that are consumed; the circuit with 3 GKP breeding rounds would contain 17 input cats. The cat input $|\pm\alpha_4\rangle$ breeds successively with the output of two cat trees, each of which produces high quality GKP states.

The distillation infidelity threshold for H -type states is approximately 14.7%. The plots in Fig. 8 show a deterministic simulation of CV-QEC on input vacuum or cat states with post-selection on quadrature measurement outcomes of zero. Note that this post-selection, done to ease simulations, will slightly underestimate the output squeezing of the bred GKP states [54]. We see that when performing CV-QEC on the vacuum mode (purple data points) one obtains infidelities above the distillation threshold (of 14.7%) when using GKP states with squeezing of 13dB or less as the resource state for CV-QEC. On the other hand, with effective squeezing greater than 12dB and cat states with amplitude of ~ 5.1 , magic states with infidelities below the distillation threshold can be obtained.

In practice, the cat states produced by PhANTM have random amplitudes, and they must be squeezed in order to obtain the correct

spacing for GKP breeding before magic state generation. This means that our GKP states have a random squeezing distribution. Further, as Fig. 8 shows, the optimal input cat amplitude for the error-corrected mode is between 4.9 and 5.5. Because our cats have random amplitudes and squeezing levels, only a few will hit this required target, although PhANTM can be engineered to emit cat states with mean squeezing levels and amplitudes centered on the requirements. Monte Carlo simulations were used to determine the success rate of producing a magic state when random cats from PhANTM were used to provide the resource states, both for GKP breeding and the cat state upon which CV QEC was performed. The simulations use Qutip [69] for density matrix calculations and Strawberry Fields [70] for homodyne detection simulation. Fig. 10 shows histograms of the infidelities obtained from each magic state generation attempt, where the fidelity is measured with respect to an ideal H -type magic state. The cat states were produced using Monte Carlo PhANTM simulations with a cluster state squeezing of 13.5 dB, 20 PhANTM steps, and 8 photon-subtraction attempts made before homodyne detection and teleportation to the next PhANTM time step. When using vacuum states as the error-corrected mode, we see a success rate of 0.003 after 1000 iterations (brown counts in the histogram). On the other hand, when using cat states as the error-corrected mode, we see a success rate of 0.048 after 1000 magic state generation attempts, an improvement of more than factor of 10, which reduces resources required to produce a magic state by a factor 10 relative to using vacuum alone. This simulation specifically targeted H -type states, which enable teleportation of T gates within the architecture. These success rates mean that magic factories will not require the majority of qubit resources in surface code substrates using lattice surgery [72].

7 Encoding Logical Qubits in a Surface Code

GKP encoding can protect against small displacement errors, but larger displacements ($> \frac{\pi}{2}$) will result in an error. Depending on the quadrature, this can be either a Pauli X error (q quadrature) or a Pauli Z error (p quadrature). These

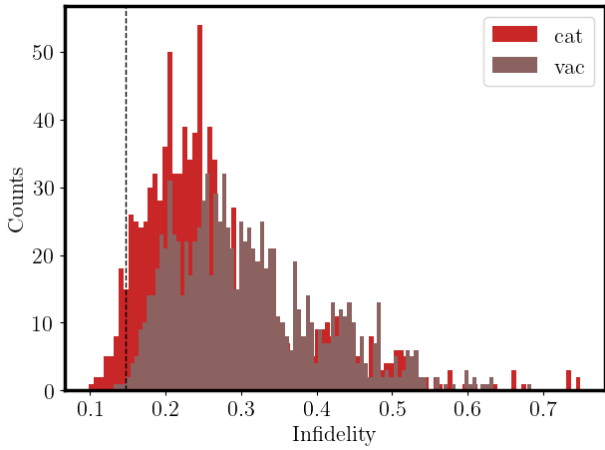


Figure 10: Histograms of fidelities obtained from Monte Carlo simulation of magic state generation for CV QEC on vacuum modes (brown) or cat states (red).

logical qubit errors require additional qubit-level error-correction. Many error correcting codes have been studied in the context of GKP qubits [26, 73, 74, 75]. We will focus on topological error correction based on a family of codes known as surface codes, which have been shown to have high thresholds ($\approx 1\%$ under circuit level noise [76] or $\approx 10\text{dB}$ with GKP states [26]) and well-studied logical gates [77].

As mentioned earlier, photonic CVQC is well-suited to MBQC. Circuit model codes known as Calderbank-Steane-Short(CSS) codes, where X and Z errors are detected independently, can be translated into a cluster state for fault-tolerant MBQC via a process known as foliation [78], which loosely maps the data qubits in the circuit model code to linear cluster states. Surface codes are examples of CSS codes, and the resulting foliated 3D cluster state is often called the Raussendorf-Harrington-Goyal (RHG) lattice after the authors who first introduced it [79]. The unit cell of this cluster state is shown in Fig. 11. The product of X -basis measurements (equivalently p -basis homodyne measurements for GKP qubits) on all face qubits is a *stabilizer* of the cluster state. In the error free case, measuring stabilizers will result in $+1$ eigenvalue and errors will result in -1 measurement results for any stabilizer that they anti-commute with. A second unit cell can be defined by shifting the base unit cell by $1/2$ in all directions. Here, the edge qubits of the initial unit cell become face qubits of the second unit cell. These two lattices are known as

primal and dual lattices, respectively. By measuring the stabilizers of both lattices, errors on all qubits can be detected. Logical operators in foliated cluster states are formed by correlation surfaces that map logical operators from the underlying code from one end of the cluster state to the other. To implement logical X and Z operations, correlation surfaces of the RHG lattice are given by 2D sheets of qubits which span the lattice and intersect primal and dual unit cells, respectively. The code distance is the shortest chain of errors that can result in a logical error, which is equivalent to the shortest error chain that can span the RHG lattice. The threshold theorem states that provided the physical error rate of the underlying qubits is below some threshold, we can arbitrarily suppress the logical error rate by increasing the code distance [80].

The popular Clifford + T gate set maps naturally to the surface code. Clifford gates and teleportation based T gates can be implemented by lattice surgery [77]. This approach defines logical qubits as patches, which are subsections of the overall surface code lattice, with logical operators identified on the edges. In MBQC, this corresponds to volumes of the RHG lattice with correlation surfaces on the boundaries. Patches can be merged by measuring stabilizer checks between edges of the patches, thus performing a joint logical measurement of the two patches. Clifford gates and the T gate can be implemented with joint measurements including ancilla qubits or magic states. In fact, in Ref. [77], it was shown how all Clifford gates can be commuted through the T gates and measurements, resulting in circuits that just consist of multi-qubit joint measurements including a magic state. Lattice surgery operations can also be mapped to changing measurement bases (homodyne angles) in fault-tolerant MBQC [61].

This gate set is well studied and optimized, however our architecture can be adapted to other gate sets [77, 81] and other error correcting codes [82]. In this paper we aim to quantify Gaussian squeezing resources needed to demonstrate FT thresholds with the RHG lattice serving as the example code. We leave optimization of the error correcting code and gate set for future work.

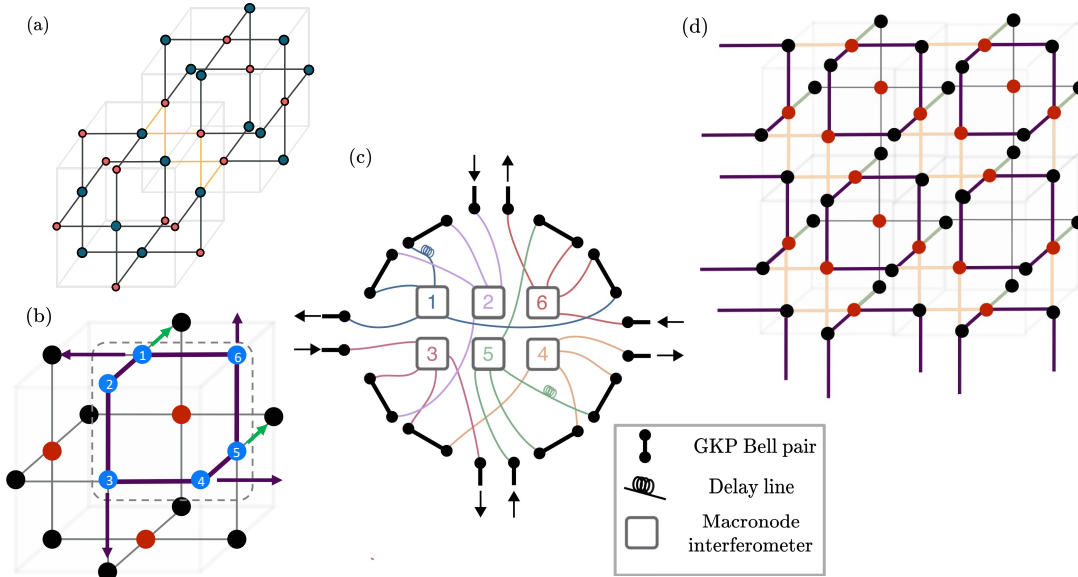


Figure 11: (a) RHG unit cells. Primal unit cell shown with blue qubits on faces and red qubits on edges. Products of X -basis measurement on face qubits are a stabilizer of the cluster state. Dual unit cell shown shifted by half a unit in all directions. Now red qubits are on faces and blue qubits are on edges. By measuring all primal and dual stabilizers, Pauli errors on all qubits can be corrected. (b) Six macronode tilable unit cells shown relative to the RHG unit cell. Purple arrows show where GKP nodes are sent and green arrows dictate delayed modes. (c) Macronode layout within the unit cell. 12 GKP Bell pairs are created per unit cell with four modes being sent to neighboring unit cells and four being received. Temporal connections between macronodes are created using delay lines. The colored links represent fiber optic connections between separate optical chips for the GKP generation and macronode interferometer circuits. (d) Example of the tileability of the unit cell into a larger RHG lattice. Orange links denote Bell pairs shared between unit cells and green edges indicate temporal links. Bell pairs can be generated by interfering sensor states output from the GKP generation algorithm on a beamsplitter.

7.1 Macronode GKP lattice construction

With GKP qubits there exists more than one method for constructing the RHG lattice in Fig 11(a).

We can construct the lattice directly, where the nodes are GKP $|+\rangle$ states and the edges are CV C_Z gates, which can be implemented deterministically with linear optics and squeezing. This approach has two main drawbacks. Firstly, while it has been demonstrated previously [83], in-line squeezing is generally difficult to realize experimentally. Second, ideal C_Z gates are not physical, requiring infinite squeezing. Approximating the gate with finite squeezing will add noise to the GKP qubits. Alternatively, any lattice can be constructed with passive interferometers starting with GKP Bell pairs, where the number of physical modes per macronode is the valency of the underlying *canonical* graph state and the GKP Bell pairs are distributed to match the entanglement in the canonical graph state. Relating measurements on the physical modes

with effective measurements on the underlying lattice requires a list of corrections known as a *dictionary protocol* [58], which we describe in Appendix 11.6. Note, that in the following we use the terms GKP state and sensor state interchangeably, always meaning either a sensor state or one half of a GKP Bell pair formed by interfering sensor states.

We wish to design a modular construction of the required lattice. To this end, we identify a unit cell consisting of 24 sensor states or 12 GKP Bell pairs that can be tiled to construct a macronode RHG lattice of arbitrary size. From this unit cell, four GKP nodes are sent to neighboring unit cells, and correspondingly four are received. Two of the modes are delayed to form temporal links. This is illustrated in Fig. 11. As the primal and dual layers are created in a single time-step, we avoid the requirement for adaptive optics that have often been used to account for the different entanglement structure in primal and dual layers [26, 28]. To teleport non-Clifford operations onto logical qubits within the surface

code, physical magic states are injected into the RHG lattice [72, 61]. The physical magic state is interfered with a vacuum state at a beamsplitter and is used in place of one GKP Bell pair in the RHG macronode.

We note here that due to the inherent randomness in the measurements, there are several feed forward corrections that need to be accounted for. Some of them, such as the corrections due to the macronode construction and the random outcomes on the effective Pauli measurements on the RHG lattice, do not need to be physically implemented and can be tracked in software. While certain operations do require physically implementing the feedforward, such as changing measurement bases due to magic state injection, changing the measurement basis corresponds to changing the homodyne phase for GKP based architectures. In this way, the required adaptivity at the QEC level does not require any fast modulators on the qubit modes.

8 FTQC Threshold Simulation

Several CV FTQC architectures based on GKP qubits and the surface code exist in the literature and have been characterised by their GKP effective squeezing threshold. Refs. [84] and [26] used constructions based on C_Z gates and found thresholds of 10 dB and 10.5 dB respectively, however these ignored the additional noise from finite squeezing in physical gates. In Ref. [28] found a threshold of 12.7 dB while accounting for finite squeezing. Ref. [27] proposed a macronode-based architecture which did away with the need for C_Z gates and found a threshold of 10.1 dB. Refs. [26, 28, 27] all require fast reconfigurable optics either in the computation or in the preparation of GKP states with a sufficiently high success probability.

In the following section, we analyze the fault-tolerant performance of our architecture. In order to allow comparison with other architectures, we first use a phenomenological error model used widely in the literature [84, 85, 28], which models realistic GKP states as ideal GKP states acted on by a Gaussian displacement channel.

Formally, this is represented as [27]

$$\varepsilon(\rho, \Delta) = \frac{1}{\sqrt{\pi}\Delta^2} \int d\alpha \exp\left(-\frac{1}{\Delta^2} |\alpha|^2\right) \hat{D}(\alpha) \rho \hat{D}^\dagger(\alpha) \quad (1)$$

Which has the effect of replacing the delta function teeth of an ideal GKP state with Gaussian distributions with a standard deviation, Δ , where Δ describes the same broadening of the GKP teeth as Δ_D described in Sec. 2 [26]. We can link this to the GKP effective squeezing in dB as $S_{\text{dB}} = -\log_{10}(\Delta^2)$ [26]. Despite its simplicity, this error model can describe many common noise sources such as finite squeezing and photon loss [27, 26, 28, 86]. We then move on to quantify the fault-tolerance performance in terms of the input Gaussian resource for PhANTM - the squeezing level of the multi-mode cluster state. The input for these simulations are distributions of GKP squeezings obtained from simulations of the proposed GKP generation pipeline.

In both cases, to determine the logical error rate we track how the GKP effective squeezing propagates through the optical circuit used to generate the macronode lattice and use the propagated noise variances to simulate noisy homodyne measurements. We apply the corresponding post-processing to reduce the physical macronode measurements to the effective noisy measurements on the canonical lattice, as shown in Appendix 11.6. With these noisy measurements we determine the probability of a qubit level error on each node of the canonical lattice, see Appendix 11.7 for more detail. These probabilities are used to sample a set of errors and generate the corresponding syndrome of the RHG lattice. This syndrome is then decoded using the pyMatching implementation of the minimum weight perfect matching decoder [87]. This simulation method is described in more detail in Appendix 11.8 and is similar to the method used in Ref [26], adapted to the macronode approach in Ref. [27] and for our specific GKP generation method.

8.1 Phenomenological error model

Here, we present results which tailor the phenomenological noise model to the PhANTM and adaptive breeding approach to generate GKP states. As described in Sec. 5, generating GKP states from PHANTM and breeding results in

GKP states that are inherently asymmetric due to the different physical processes that determine the squeezing in the GKP quadratures. Note that this is equivalent to grid states with asymmetric spacing in the two quadratures [73] which has also been studied in [82]. This feature was incorporated into our simulation method by defining p and q noise variances separately for each GKP state. While we can also set each GKP state independently, for simplicity we assume that the p and q noise are identical for each GKP state. To determine the 2D threshold curve for unbalanced quadrature noise we employ the method used in Ref. [13]. By expressing the p and q noise in terms of a dummy variable, $\Delta_p = c_p \times x$, $\Delta_q = c_q \times x$, we fix the ratio of p and q noise via c_p/c_q and use x to find the threshold for a given ratio. Fig. 12 shows the resultant surface, where we have shaded the correctable region. The surface is symmetric with marginal squeezing thresholds of 7.2dB. With balanced squeezing in each quadrature we recover a threshold of 10.2dB, which is inline with similar error models [27].

A second adaptation stems from the fact that the size of the cat states from PhANTM depend on the number of photons subtracted and so are inherently probabilistic. This results in GKP states that do not have a single squeezing value for each quadrature, but rather are described by a distribution of effective squeezing. To include this in our simulation framework we employ a two-stage sampling procedure where for each mode and for each Monte Carlo iteration we sample first the squeezing variance for each mode from a Gaussian distribution with standard deviation given by σ_d and then we sample the noisy homodyne value. Fig. 12(b) shows how the squeezing threshold depends on σ_d .

8.2 End-to-end simulations

As in many similar architectures [28, 84, 88, 26, 27], the previous section abstracted the thresholds in terms of GKP effective squeezing. While this allows a comparison against different architectures, it does not give an estimate of the physical requirements for our approach. A key resource for our architecture is the squeezing level of the multi-mode Gaussian cluster state. In this section we aim to determine the threshold in terms of

this resource. To this end we perform simulations of PhANTM and adaptive breeding and then use the resultant GKP states as the inputs to the logical-level simulations. We fix the PhANTM and breeding setup to use eight PNR detectors per frequency mode with gradient beamsplitters, see Sec. 4. We chose three breeding rounds and employ a threshold for cat squeezing of $r_{lb} = 0.5$ (4.34dB), below which a momentum squeezed state from the cluster state is swapped in, as described in Sec. 5. We then sweep the cluster squeezing and simulate the logical error rate for several code distances, repeating this simulation for a range of PhANTM steps.

We find thresholds for cluster squeezing needed for PhANTM and teleportation-based squeezing gate of 12.1, 12.7, and 13 dB for 20, 15 and 10 PhANTM steps, respectively. We see that the required cluster squeezing increases as the number of PhANTM steps, and therefore the mean cat state amplitude α , decreases. The lower panels in Fig. 13 show the mean and standard deviation of the GKP effective squeezing distributions output from the breeding process. We also compare the adaptive breeding protocol, used in Fig. 13, to standard breeding, where cat states must be squeezed exactly to the GKP spacing. With standard breeding, no threshold was found below 13.5 dB, indicating that adaptive breeding can be used to reduce the threshold by *at least* 1.5 dB and possibly more. While our simulation takes into account all generated GKP states, not just those above threshold, we can quantify the proportion of GKP states that fall above threshold, allowing us to not only compare the Gaussian squeezing resources required but also the generation probability with other GKP generation protocols. To this end, we determine the likelihood that a GKP state drawn randomly from the generated distribution has a squeezing level above the threshold, that is the combination of squeezing in each quadrature puts it in the 'correctable' region of Fig. 12. This likelihood can then be taken as the success probability of our protocol. Some examples are shown in table 1.

While direct simulations of the required Gaussian squeezing thresholds for other GKP generation methods are missing from the literature, we can compare to several approaches that benchmark themselves by generating 10 dB GKP

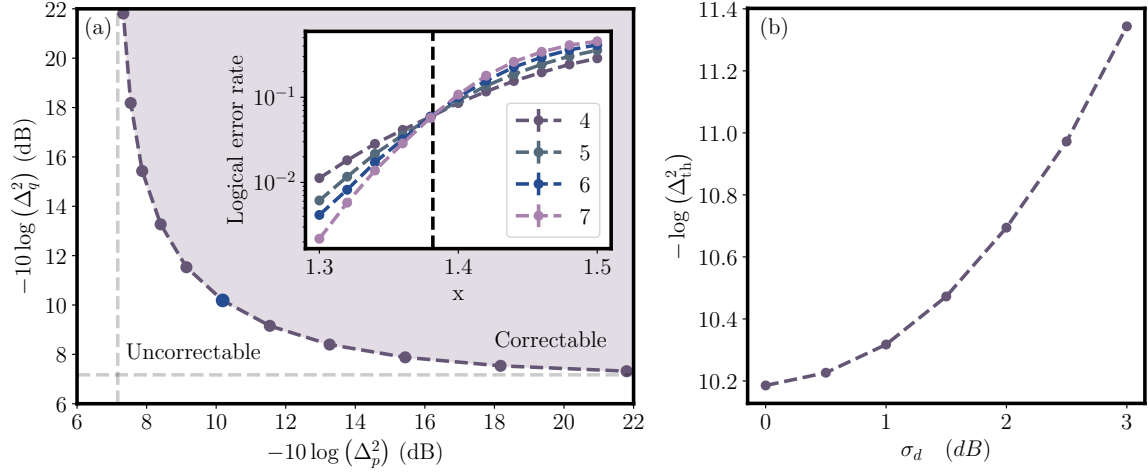


Figure 12: (a) 2D threshold plot for independent noise in the GKP quadratures, in terms of decibel squeezing, $S_{\text{dB}} = -10 \log(\Delta^2)$. Combinations of squeezing values above the line are correctable and those below are not. Dotted lines show the marginals with infinite squeezing in one of the quadratures. The inset shows an example threshold plot for a single point, shown in blue, in the main panel. Each point is averaged 10^5 times. (b) Plot showing the dependence of the squeezing threshold on the distribution of GKP effective squeezing. On each Monte Carlo run, first the GKP effective squeezing for each mode from a normal distribution with standard deviation σ_d before the noisy homodyne values are generated from the sampled squeezing value.

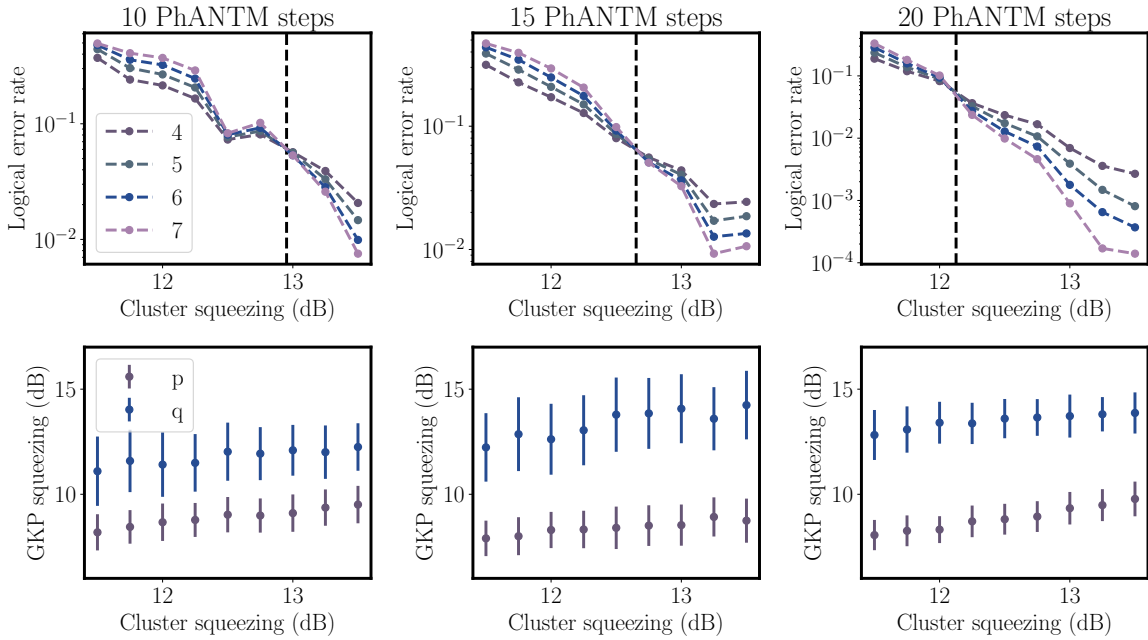


Figure 13: Fault-tolerant threshold for cluster squeezing for a range of PhANTM step numbers. Bottom panels show the corresponding mean and standard deviations of GKP effective squeezing distributions input to the logical-level simulation. These results used 10^3 PhANTM iterations, 10^3 adaptive breeding iterations and 10^5 logical error rate trials.

states, which are at the threshold for several fault-tolerant architectures [27, 89, 84]. The highest success probability for single-shot GKP generation protocols are found in Ref. [57]; however they require input 18–20 dB of Gaussian squeezing, and photon number resolution up to 40 pho-

tons. On the other hand, lower input squeezing resources can be used as in Ref. [49], requiring 12 dB, and Ref. [63] is as low as 6dB. However, in the former, success probabilities are on the order of 10^{-5} and the latter requires the generation of large, $n = 10$, Fock states which are experimen-

tally challenging .

9 Discussion

We have presented an architecture for CV FTQC that uses the time-frequency-space degrees of freedom to create GKP states above the FT threshold with high probabilities, creates distillable magic states with a supply of realistically squeezed GKP states and cat states, and can be implemented on low-loss photonic chips in a modular architecture. This work proposes a static state generation process that produces a distribution of GKP effective squeezing, affecting the FT threshold only modestly compared to idealized GKP states with a set squeezing target, as shown in Fig. 12. The results presented in the paper require Gaussian squeezing levels of 12 dB to 13 dB. Low photon number resolution is needed for cat state generation after which all steps require homodyne detections and linear optical CV operations. The architecture can be implemented without on-chip switches or quantum memories, resulting in a fully-passive architecture.

The experimental hardware needed for the architecture is ambitious but feasible. The squeezing level needed is below the state-of-the-art levels achieved in free-space [90]. In recent years, the squeezing measured from on-chip sources has steadily improved, reaching 8.3 dB [91], with efforts underway to reach levels seen in free-space. The other major resource needed in our approach is photon number resolving detection. While TES devices offer resolutions in excess of 100 photons [92] and have traditionally been used in CVQC, SMSPDs [30] with low data acquisition requirements operating at higher temperatures have been shown to detect up to 10 photons, sufficient for our architecture. Simulta-

neously, chip-integrated room-temperature PNR technologies capable of detecting up to 10 photons at MHz scales are also in development [93]. Finally, while losses in photonic circuit components need to be minimized for implementing our architecture on photonic chips, Silicon nitride platforms currently provide some of the lowest losses available for passive components [29] at levels that look promising for FTQC.

Multiple improvements, optimizations, and validations of the presented architecture remain to be explored. Higher probability methods for cat state generation via optimizing the squeezing and photon number resolution are possible. Adaptive breeding, used in both sensor state and magic state generation protocols, can be optimized with advanced adaptation algorithms. In the case of magic states, targeting a higher success probability while simultaneously reducing the quality of GKP states needed for magic generation looks possible with further development of the CV-QEC process with non-Gaussian states. The squeezing requirements of the architecture could also be lowered by moving to cluster states with a lower valency [94] and employing more sophisticated GKP decoders, such as the ones taking into account correlations introduced by linear optics [82].

While the effect of losses in PhANTM has been investigated earlier [55], appendix 11.9 provides simulations and a discussion of the effect of losses on the adaptive breeding and QEC protocols. We briefly note that PhANTM is more sensitive to propagation and homodyne losses than to photon number resolution losses, and shows significant reduction in the Wigner negativity of cat states for losses beyond 1%. As for breeding, the effective squeezing of GKP states from breeding is reduced by less than 1 dB with 3% homodyne loss. Similarly, the GKP effective squeezing needed to be at 10.2 dB threshold for fault tolerance is increased to about 11.6 dB with a 5% loss. This demonstrates that losses are less important for adaptive breeding and QEC than for cat generation through PhANTM. Estimating the loss and noise targets especially with further design improvements would require more comprehensive error models and simulations, including the effects of Gaussian noise on effective squeezing and optical loss on Wigner negativity at each step of

Cluster squeezing (dB)	PhANTM steps	
	10	20
12.0	0.26	0.45
13.0	0.56	0.91

Table 1: Success rates for which adaptive breeding produces sensor states with squeezing levels above the FT threshold for different levels of cluster squeezing and PhANTM steps.

the architecture. An end-to-end simulation incorporating losses is left for future work.

While many improvements could be made and the performance under losses are yet to be fully quantified, we believe that the modular and passive architecture for photonic CV FTQC presented in the paper will constitute a foundational design for fault tolerant photonic quantum computers in the future.

10 Acknowledgements

Part of the QEC simulation code was funded under DARPA Contract HR00112390051. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. We acknowledge insightful conversations with QC82 co-founders Olivier Pfister, Xu Yi, Joe Campbell, and Andreas Beling on theoretical and experimental CVQC, squeezing sources, homodyne detection, photon number resolution, and many other topics. We are thankful to Leandre Brunel for sharing some of his work on GKP lattice construction methods.

References

- [1] Richard P. Feynman. “Simulating physics with computers”. *International Journal of Theoretical Physics* **21**, 467–488 (1982).
- [2] John Preskill. “Quantum computing in the NISQ era and beyond”. *Quantum* **2**, 79 (2018).
- [3] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe. “Demonstration of a small programmable quantum computer with atomic qubits”. *Nature* **536**, 63–66 (2016).
- [4] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. *Nature* **549**, 242–246 (2017).
- [5] Loïc Henriët, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. “Quantum computing with neutral atoms”. *Quantum* **4**, 327 (2020).
- [6] P.J.J. O’Malley, R. Babbush, I.D. Kivlichan, J. Romero, J.R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A.G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J.Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T.C. White, P.V. Coveney, P.J. Love, H. Neven, A. Aspuru-Guzik, and J.M. Martinis. “Scalable Quantum Simulation of Molecular Energies”. *Physical Review X* **6**, 031007 (2016).
- [7] Isaac H Kim, Ye-Hua Liu, Sam Pallister, William Pol, Sam Roberts, and Eunseok Lee. “Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules”. *Physical Review Research* **4**, 023019 (2022).
- [8] Craig Gidney and Martin Ekerå. “How to factor 2048 bit rsa integers in 8 hours using 20 million noisy qubits”. *Quantum* **5**, 433 (2021).
- [9] Daniel Litinski. “How to compute a 256-bit elliptic curve private key with only 50 million toffoli gates” (2023).
- [10] Srujan Meesala, David Lake, Steven Wood, Piero Chiappina, Changchun Zhong, Andrew D. Beyer, Matthew D. Shaw, Liang Jiang, and Oskar Painter. “Quantum Entanglement between Optical and Microwave Photonic Qubits”. *Physical Review X* **14**, 031055 (2024).
- [11] M. Saffman. “Quantum computing with atomic qubits and Rydberg interactions: progress and challenges”. *Journal of Physics B: Atomic, Molecular and Optical Physics* **49**, 202001 (2016).
- [12] Pieter Kok, W. J. Munro, Kae Nemoto, T. C. Ralph, Jonathan P. Dowling, and G. J. Milburn. “Linear optical quantum computing with photonic qubits”. *Rev. Mod. Phys.* **79**, 135–174 (2007).
- [13] Sara Bartolucci, Patrick Birchall, Damien Bonneau, Hugo Cable, Mercedes Gimeno-

- Segovia, Konrad Kieling, Naomi Nickerson, Terry Rudolph, and Chris Sparrow. “Switch networks for photonic fusion-based quantum computing” (2021). [arXiv:2109.13760](#).
- [14] Seth Lloyd and Samuel L. Braunstein. “Quantum computation over continuous variables”. *Phys. Rev. Lett.* **82**, 1784 (1999).
- [15] C. Fabre and N. Treps. “Modes and states in quantum optics”. *Rev. Mod. Phys.* **92**, 035005 (2020).
- [16] Christian Weedbrook, Stefano Pirandola, Raúl García-Patrón, Nicolas J. Cerf, Timothy C. Ralph, Jeffrey H. Shapiro, and Seth Lloyd. “Gaussian quantum information”. *Rev. Mod. Phys.* **84**, 621–669 (2012).
- [17] Nicolas C. Menicucci. “Fault-tolerant measurement-based quantum computing with continuous-variable cluster states”. *Phys. Rev. Lett.* **112**, 120504 (2014).
- [18] Warit Asavanant, Yu Shiozawa, Shota Yokoyama, Baramée Charoensombutamon, Hiroki Emura, Rafael N. Alexander, Shuntaro Takeda, Jun-ichi Yoshikawa, Nicolas C. Menicucci, Hidehiro Yonezawa, and Akira Furusawa. “Generation of time-domain-multiplexed two-dimensional cluster state”. *Science* **366**, 373–376 (2019).
- [19] N. C. Menicucci, P. van Loock, M. Gu, C. Weedbrook, T. C. Ralph, and M. A. Nielsen. “Universal quantum computation with continuous-variable cluster states”. *Phys. Rev. Lett.* **97**, 110501 (2006).
- [20] Daniel Gottesman, Alexei Kitaev, and John Preskill. “Encoding a qubit in an oscillator”. *Phys. Rev. A* **64**, 012310 (2001).
- [21] Weizhou Cai, Yuwei Ma, Weiting Wang, Chang-Ling Zou, and Luyan Sun. “Bosonic quantum error correction codes in superconducting quantum circuits”. *Fundamental Research* **1**, 50–67 (2021).
- [22] Qian Xu, Guo Zheng, Yu-Xin Wang, Peter Zoller, Aashish A. Clerk, and Liang Jiang. “Autonomous quantum error correction and fault-tolerant quantum computation with squeezed cat qubits”. *npj Quantum Information* **9**, 1–11 (2023).
- [23] M. Ohliger. “Limitations of quantum computing with Gaussian cluster states”. *Physical Review A* **82** (2010).
- [24] Sergey Bravyi and Alexei Kitaev. “Universal quantum computation with ideal clifford gates and noisy ancillas”. *Phys. Rev. A* **71**, 022316 (2005).
- [25] Ben Q Baragiola, Giacomo Pantaleoni, Rafael N Alexander, Angela Karanjai, and Nicolas C Menicucci. “All-gaussian universality and fault tolerance with the Gottesman-Kitaev-Preskill code”. *Physical Review Letters* **123**, 200502 (2019).
- [26] J. Eli Bourassa, Rafael N. Alexander, Michael Vasmer, Ashlesha Patil, Ilan Tzitrin, Takaya Matsuura, Daiqin Su, Ben Q. Baragiola, Saikat Guha, Guillaume Dauphinais, Krishna K. Sabapathy, Nicolas C. Menicucci, and Ish Dhand. “Blueprint for a Scalable Photonic Fault-Tolerant Quantum Computer”. *Quantum* **5**, 392 (2021).
- [27] Ilan Tzitrin, Takaya Matsuura, Rafael N Alexander, Guillaume Dauphinais, J Eli Bourassa, Krishna K Sabapathy, Nicolas C Menicucci, and Ish Dhand. “Fault-tolerant quantum computation with static linear optics”. *PRX Quantum* **2**, 040353 (2021).
- [28] Mikkel V Larsen, Christopher Chamberland, Kyungjoo Noh, Jonas S Neergaard-Nielsen, and Ulrik L Andersen. “Fault-tolerant continuous-variable measurement-based quantum computation architecture”. *PRX Quantum* **2**, 030325 (2021).
- [29] PsiQuantum Team. “A manufacturable platform for photonic quantum computing”. *Nature* **641**, 1–3 (2025).
- [30] L. D. Kong, T. Z. Zhang, X. Y. Liu, H. Li, Z. Wang, X. M. Xie, and L. X. You. “Large-inductance superconducting microstrip photon detector enabling 10 photon-number resolution”. *Adv. Photonics* **6**, 016004 (2024).
- [31] R Nehra, C. H. Chang, Q. Yu, A Beling, and O Pfister. “Photon-number-resolving segmented detectors based on single-photon avalanche-photodiodes”. *Optics Express* **28**, 3660 (2020).
- [32] Roman Schnabel. “Squeezed states of light and their applications in laser interferometers”. *Physics Reports* **684**, 1–51 (2017).
- [33] Olivier Pfister. “Continuous-variable quantum computing in the quantum optical frequency comb”. *Journal of Physics B*:

- Atomic, Molecular and Optical Physics **53**, 012001 (2019).
- [34] Michael A Nielsen. “Cluster-state quantum computation”. *Reports on Mathematical Physics* **57**, 147–161 (2006).
- [35] R. Raussendorf and H. J. Briegel. “A one-way quantum computer”. *Phys. Rev. Lett.* **86**, 5188 (2001).
- [36] S. T. Flammia, N. C. Menicucci, and O. Pfister. “The optical frequency comb as a one-way quantum computer”. *J. Phys. B*, **42**, 114009 (2009).
- [37] Nicolas C. Menicucci. “Temporal-mode continuous-variable cluster states using linear optics”. *Phys. Rev. A* **83**, 062314 (2011).
- [38] Peter van Loock, Christian Weedbrook, and Mile Gu. “Building gaussian cluster states by linear optics”. *Physical Review A—Atomic, Molecular, and Optical Physics* **76**, 032321 (2007).
- [39] Blayne W Walshe, Rafael N Alexander, Nicolas C Menicucci, and Ben Q Baragiola. “Streamlined quantum computing with macronode cluster states”. *Physical Review A* **104**, 062427 (2021).
- [40] Rafael N. Alexander and Nicolas C. Menicucci. “Flexible quantum circuits using scalable continuous-variable cluster states”. *Phys. Rev. A* **93**, 062326 (2016).
- [41] M. Pysher, R. Bloomer, C. M. Kaleva, T. D. Roberts, P. Battle, and O. Pfister. “Broadband amplitude squeezing in a periodically poled KTiOPO₄ waveguide”. *Opt. Lett.* **34**, 256 (2009).
- [42] Mile Gu, Christian Weedbrook, Nicolas C. Menicucci, Timothy C. Ralph, and Peter van Loock. “Quantum computing with continuous-variable clusters”. *Phys. Rev. A* **79**, 062318 (2009).
- [43] N. C. Menicucci, S. T. Flammia, H. Zaidi, and O. Pfister. “Ultracompact generation of continuous-variable cluster states”. *Phys. Rev. A* **76**, 010302(R) (2007).
- [44] A. I. Lvovsky and M. G. Raymer. “Continuous-variable optical quantum-state tomography”. *Rev. Mod. Phys.* **81**, 299–332 (2009).
- [45] Stephen D. Bartlett, Barry C. Sanders, Samuel L. Braunstein, and Kae Nemoto. “Efficient classical simulation of continuous variable quantum information processes”. *Phys. Rev. Lett.* **88**, 097904 (2002).
- [46] Alexei Ourjoumtsev, Rosa Tualle-Brouri, Julien Laurat, and Philippe Grangier. “Generating optical schrodinger kittens for quantum information processing”. *Science* **312**, 83–86 (2006).
- [47] Mohamed F Melalkia, Tecla Gabbrielli, Antoine Petitjean, Léandre Brunel, Alessandro Zavatta, Sébastien Tanzilli, Jean Etesse, and Virginia D’Auria. “Plug-and-play generation of non-gaussian states of light at a telecom wavelength”. *Optics Express* **30**, 45195–45201 (2022).
- [48] David S. Schlegel, Fabrizio Minganti, and Vincenzo Savona. “Quantum error correction using squeezed schrödinger cat states”. *Phys. Rev. A* **106**, 022431 (2022).
- [49] Ilan Tzitrin, J Eli Bourassa, Nicolas C Menicucci, and Krishna Kumar Sabapathy. “Progress towards practical qubit computation using approximate Gottesman-Kitaev-Preskill codes”. *Physical Review A* **101**, 032315 (2020).
- [50] Lucas J Mensen, Ben Q Baragiola, and Nicolas C Menicucci. “Phase-space methods for representing, manipulating, and correcting Gottesman-Kitaev-Preskill qubits”. *Physical Review A* **104**, 022408 (2021).
- [51] Arne L Grimsmo and Shruti Puri. “Quantum error correction with the Gottesman-Kitaev-Preskill code”. *PRX Quantum* **2**, 020101 (2021).
- [52] Blayne W Walshe, Ben Q Baragiola, Rafael N Alexander, and Nicolas C Menicucci. “Continuous-variable gate teleportation and bosonic-code error correction”. *Physical Review A* **102**, 062411 (2020).
- [53] Takaya Matsuura, Hayata Yamasaki, and Masato Koashi. “Equivalence of approximate Gottesman-Kitaev-Preskill codes”. *Physical Review A* **102**, 032408 (2020).
- [54] Daniel J. Weigand and Barbara M. Terhal. “Generating grid states from Schrödinger-cat states without postselection”. *Phys. Rev. A* **97**, 022341 (2018).

- [55] Miller Eaton, Carlos González-Arciniegas, Rafael N. Alexander, Nicolas C. Menicucci, and Pfister Pfister. “Measurement-based generation and preservation of cat and grid states within a continuous-variable cluster state”. *Quantum* **6**, 769 (2022).
- [56] R Raussendorf, J Harrington, and K Goyal. “A fault-tolerant one-way quantum computer”. *Ann. Phys. (NY)* **321**, 2242–2270 (2006).
- [57] Kan Takase, Fumiya Hanamura, Hironari Nagayoshi, J. Eli Bourassa, Rafael N. Alexander, Akito Kawasaki, Warit Asavanant, Mamoru Endo, and Akira Furusawa. “Generation of flying logical qubits using generalized photon subtraction with adaptive gaussian operations”. *Phys. Rev. A* **110**, 012436 (2024).
- [58] Rafael N. Alexander, Seiji C. Armstrong, Ryuji Ukai, and Nicolas C. Menicucci. “Noise analysis of single-mode gaussian operations using continuous-variable cluster states”. *Phys. Rev. A* **90**, 062324 (2014).
- [59] Rafael N. Alexander, Pei Wang, Niranjana Sridhar, Moran Chen, Olivier Pfister, and Nicolas C. Menicucci. “One-way quantum computing with arbitrarily large time-frequency continuous-variable cluster states from a single optical parametric oscillator”. *Phys. Rev. A* **94**, 032327 (2016).
- [60] Junqiu Liu, Guanhao Huang, Rui Ning Wang, Jijun He, Arslan S. Raja, Tianyi Liu, Nils J. Engelsens, and Tobias J. Kippenberg. “High-yield, wafer-scale fabrication of ultralow-loss, dispersion-engineered silicon nitride photonic circuits”. *Nature Communications* **12**, 2236 (2021).
- [61] Daniel Herr, Alexandru Paler, Simon J Devitt, and Franco Nori. “Lattice surgery on the raussendorf lattice”. *Quantum Science and Technology* **3**, 035011 (2018).
- [62] Amanuel Anteneh, Léandre Brunel, and Olivier Pfister. “Machine learning for efficient generation of universal photonic quantum computing resources”. *Optica Quantum* **2**, 296–302 (2024).
- [63] Matthew S. Winnel, Joshua J. Guanzon, Deepesh Singh, and Timothy C. Ralph. “Deterministic preparation of optical squeezed cat and gottesman-kitaev-preskill states”. *Phys. Rev. Lett* **132**, 230602 (2024).
- [64] M. Uria, P. Solano, and C. Hermann-Avigliano. “Deterministic Generation of Large Fock States”. *Physical Review Letters* **125**, 093603 (2020).
- [65] Robert Raussendorf. “Measurement-based quantum computation with cluster states”. PhD thesis. Ludwig–Maximilians–Universität München. (2003).
- [66] M. Eaton. “Measurement-based non-gaussian quantum state engineering”. PhD thesis. University of Virginia. (2022).
- [67] Thomas Gerrits, Adriana Lita, Brice Calkins, and Sae Woo Nam. “Superconducting transition edge sensors for quantum optics”. In *Superconducting devices in quantum optics*. Pages 31–60. Springer (2016).
- [68] Valerio Crescimanna, Aaron Z. Goldberg, and Khabat Heshami. “Seeding gaussian boson samplers with single photons for enhanced state generation”. *Phys. Rev. A* **109**, 023717 (2024).
- [69] J.R. Johansson, P.D. Nation, and Franco Nori. “QuTiP 2: A Python framework for the dynamics of open quantum systems”. *Computer Physics Communications* **184**, 1234–1240 (2013).
- [70] Nathan Killoran, Josh Izaac, Nicolás Quesada, Ville Bergholm, Matthew Amy, and Christian Weedbrook. “Strawberry Fields: A Software Platform for Photonic Quantum Computing”. *Quantum* **3**, 129 (2019).
- [71] Cameron Calcluth, Alessandro Ferraro, and Giulia Ferrini. “Vacuum provides quantum advantage to otherwise simulatable architectures”. *Physical Review A* **107**, 062414 (2023).
- [72] Daniel Litinski. “Magic state distillation: Not as costly as you think”. *Quantum* **3**, 205 (2019).
- [73] Matthew P Stafford and Nicolas C Menicucci. “Biased gottesman-kitaev-preskill repetition code”. *Physical Review A* **108**, 052428 (2023).
- [74] Nithin Raveendran, Narayanan Rengaswamy, Filip Rozpedek, Ankur Raina, Liang Jiang, and Bane Vasić. “Finite Rate

- QLDPC-GKP Coding Scheme that Surpasses the CSS Hamming Bound”. *Quantum* **6**, 767 (2022).
- [75] Jiakuan Zhang, Jian Zhao, Yu-Chun Wu, and Guo-Ping Guo. “Quantum error correction with the color-gottesman-kitaev-preskill code”. *Physical Review A* **104**, 062434 (2021).
- [76] David S Wang, Austin G Fowler, and Lloyd CL Hollenberg. “Surface code quantum computing with error rates over 1%”. *Physical Review A* **83**, 020302 (2011).
- [77] Daniel Litinski. “A game of surface codes: Large-scale quantum computing with lattice surgery”. *Quantum* **3**, 128 (2019).
- [78] A Bolt, G Duclos-Cianci, D Poulin, and TM Stace. “Foliated quantum error-correcting codes”. *Physical review letters* **117**, 070501 (2016).
- [79] Robert Raussendorf, Jim Harrington, and Kovid Goyal. “Topological fault-tolerance in cluster state quantum computation”. *New Journal of Physics* **9**, 199 (2007).
- [80] Emanuel Knill, Raymond Laflamme, and Wojciech H Zurek. “Resilient quantum computation”. *Science* **279**, 342–345 (1998).
- [81] Craig Gidney and Austin G Fowler. “Flexible layout of surface code computations using autocccz states” (2019).
- [82] Blayne W. Walshe, Ben Q. Baragiola, Hugo Ferretti, José Gefaell, Michael Vasmer, Ryohei Weil, Takaya Matsuura, Thomas Jaeken, Giacomo Pantaleoni, Zhihua Han, Timo Hillmann, Nicolas C. Menicucci, Ilan Tzitrin, and Rafael N. Alexander. “Linear-optical quantum computation with arbitrary error-correcting codes”. *Phys. Rev. Lett.* **134**, 100602 (2025).
- [83] Jun-ichi Yoshikawa, Yoshichika Miwa, Alexander Huck, Ulrik L. Andersen, Peter van Loock, and Akira Furusawa. “Demonstration of a quantum nondemolition sum gate”. *Phys. Rev. Lett.* **101**, 250501 (2008).
- [84] Kosuke Fukui, Akihisa Tomita, Atsushi Okamoto, and Keisuke Fujii. “High-threshold fault-tolerant quantum computation with analog quantum error correction”. *Phys. Rev. X* **8**, 021054 (2018).
- [85] Kosuke Fukui, Rafael N Alexander, and Peter van Loock. “All-optical long-distance quantum communication with gottesman-kitaev-preskill qubits”. *Physical Review Research* **3**, 033118 (2021).
- [86] Kyungjoo Noh and Christopher Chamberland. “Fault-tolerant bosonic quantum error correction with the surface-gottesman-kitaev-preskill code”. *Physical Review A* **101**, 012316 (2020).
- [87] Oscar Higgott and Craig Gidney. “Sparse blossom: correcting a million errors per core second with minimum-weight matching”. *Quantum* **9**, 1600 (2025).
- [88] Christopher Chamberland and Andrew W Cross. “Fault-tolerant magic state preparation with flag qubits”. *Quantum* **3**, 143 (2019).
- [89] Kosuke Fukui. “High-threshold fault-tolerant quantum computation with the gottesman-kitaev-preskill qubit under noise in an optical setup”. *Physical Review A* **107**, 052414 (2023).
- [90] Henning Vahlbruch, Moritz Mehmet, Karsten Danzmann, and Roman Schnabel. “Detection of 15 dB squeezed states of light and their application for the absolute calibration of photoelectric quantum efficiency”. *Phys. Rev. Lett.* **117**, 110801 (2016).
- [91] Takahiro Kashiwazaki, Taichi Yamashima, Koji Enbutsu, Takushi Kazama, Asuka Inoue, Kosuke Fukui, Mamoru Endo, Takeshi Umeki, and Akira Furusawa. “Over-8-db squeezed light generation by a broadband waveguide optical parametric amplifier toward fault-tolerant ultra-fast quantum computers”. *Applied Physics Letters* **122** (2023).
- [92] M Eaton, A Hossameldin, R Birrittella, P Alsing, C Gerry, H Dong, C Cuevas, and Olivier Pfister. “Resolution of 100 photons and quantum generation of unbiased random numbers”. *Nat. Phot.* **17**, 106–111 (2022).
- [93] Rajveer Nehra, Chun-Hung Chang, Qianhuan Yu, Andreas Beling, and Olivier Pfister. “Photon-number-resolving segmented detectors based on single-photon avalanche photodiodes”. *Optics Express* **28**, 3660–3675 (2020).

- [94] Stefano Paesani and Benjamin J Brown. “High-threshold quantum computing by fusing one-dimensional cluster states”. *Physical Review Letters* **131**, 120603 (2023).
- [95] L. García-Álvarez, A. Ferraro, and G. Ferrini. “From the bloch sphere to phase space representations with the gottesman-kitaev-preskill encoding”. *International Symposium on Mathematics, Quantum Theory, and Cryptography. Mathematics for Industry* **33** (2020).

11 Appendix

11.1 Additional terminology

Wavefunction for the GKP qubits in the q basis can be defined as $|\mu\rangle_G \sim \sum_{n=-\infty}^{\infty} |(2n + \mu)\sqrt{\pi}\rangle_q$ where $\mu \in \{0, 1\}$, and superpositions can be defined as usual: $|\pm\rangle_G = \frac{1}{\sqrt{2}}(|0\rangle_G \pm |1\rangle_G)$ [66]. Any superposition of these states has support in phase space only on integer multiples of $\sqrt{\pi}$, and each state is periodic with period $2\sqrt{\pi}$. Because of this, small shifts in either position or momentum can be corrected. Note that a squeezed cat state has the form of the zeroth-order GKP state, in light of which it is possible to convert between Cat states and GKP states using breeding protocols. The sensor state, which we use to construct logical GKP qubits within a surface code, is defined as

$$|S\rangle \sim \sum_{n=-\infty}^{\infty} |n\sqrt{2\pi}\rangle_q. \quad (2)$$

There are two families of logical magic states, the T-type and the H-type magic states [95], defined as

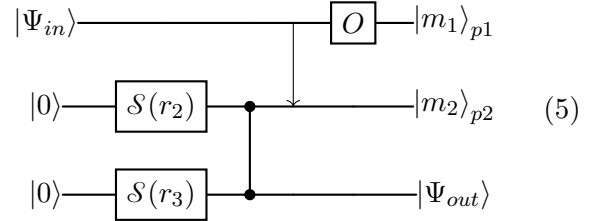
$$|T\rangle = \cos \frac{\theta}{2} |0\rangle_G + e^{i\pi/4} \sin \frac{\theta}{2} |1\rangle_G \quad (3)$$

$$|H\rangle = \frac{1}{\sqrt{2}}(|0\rangle_G + e^{i\pi/2} |1\rangle_G), \quad (4)$$

where $\theta = \arccos(1/\sqrt{3})$. There are 8 T-type and 12 H-type magic states obtainable by applying Clifford operations on $|T\rangle$ and $|H\rangle$.

11.2 Demonstration of PhANTM on dual-rail wire

Eq. 5 represents the PhANTM circuit corresponding to the operation in stage IV of Fig. 4. However, this three-mode circuit is not optimal for numerical simulations. In what follows, we demonstrate how the circuit can be reduced to a two-mode operation efficient for simulation purposes. We have



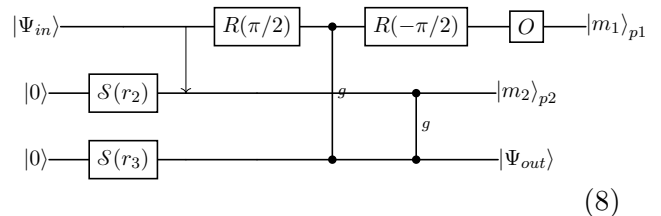
where arrow depicts a balanced beamsplitter with $\pi/2$ phase. Using the definition of the beamsplitter, we can write the relation

$$B_{\frac{\pi}{2},(12)} C_{Z(23)} B_{\frac{\pi}{2},(12)}^\dagger = \frac{1}{\sqrt{2}} e^{\frac{i}{\sqrt{2}}(q_2+p_1)q_3}, \quad (6)$$

where $B_{\frac{\pi}{2},(12)}$ and $C_{Z(23)}$ stand respectively for beamsplitter between mode 1 and 2 with a $\pi/2$ phase and C_Z gate between modes 2 and 3. From this relation we deduce the commutation relation between the operators as

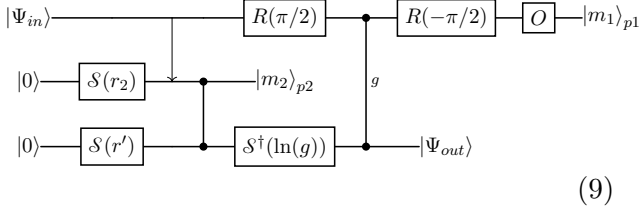
$$B_{\frac{\pi}{2},(12)} C_{Z(23)} = C_{Z(23)} (1/\sqrt{2}) R_1(-\frac{\pi}{2}) C_{Z(13)} (1/\sqrt{2}) R_1^\dagger(-\frac{\pi}{2}) B_{\frac{\pi}{2},(12)}, \quad (7)$$

where R is the rotation operator defined as $R(\theta) = e^{i\theta a^\dagger a}$. This relation can be used in the PhANTM circuit above to get



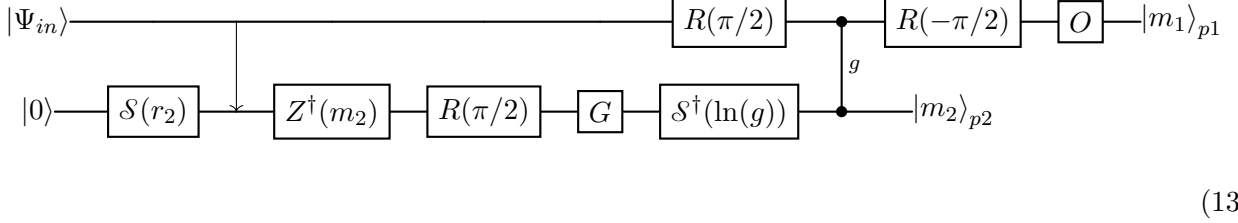
with $g = 1/\sqrt{2}$. The weighting of the C_Z gate can utilize $e^{igq_2q_3} = S^\dagger(\ln(g)) e^{iq_2q_3} S^\dagger(\ln(g))$ [55]. Subsequently we introduce the squeezing parameter r' such that $S(r') = S(\ln(g)) S(r_3)$. Furthermore since the C_Z gates commute, the circuit be-

comes



On the third mode, we can use the expression

$$\mathcal{S}(r')|0\rangle = G|0\rangle_{p3}, \quad (10)$$



From this circuit we can identify operator ζ as

$$\zeta = Z^\dagger(m_2)R(\pi/2)G\mathcal{S}^\dagger(\ln(g)). \quad (14)$$

The circuit in Eq. 13, is the one we use to perform Monte Carlo simulation of PhANTM. G operator has a second order effect and is not included to speed up run times with large Fock dimensions. That's because we take cluster squeezing higher than 11.5 dB for which $2s^2 \approx 15 \gg 1$ in Eq. 11.

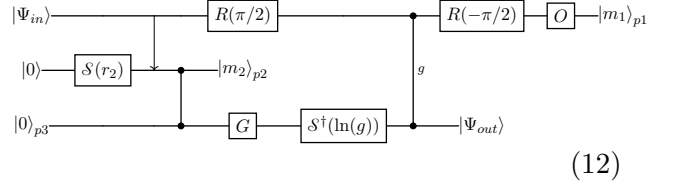
11.3 Monte Carlo simulation of PhANTM

The Monte Carlo methods used for PhANTM simulations make use of Qutip [69] for density matrix calculations and Strawberry Fields [70] for homodyne detection simulation. Fig. 14 shows the results of the PhANTM simulation for various levels of cluster squeezing and 5, 10, 15, and 20 PhANTM steps. The C_Z gate weight is set to 1. The number of photons subtracted at each subtraction event is determined stochastically, based on the density matrix of the state that is about to undergo photons subtraction [55]. The stochastic nature of the photon subtraction results in a

where G is defined as

$$G = \pi^{1/4} \sqrt{\frac{2}{s}} e^{-q^2/2s^2} \quad (11)$$

with $s = e^{r'}$. Then $|0\rangle$ can be replaced by the p momentum squeezed state. Knowing that G can commute with C_Z , the circuit becomes



We recognize the teleportation circuit between mode 2 and 3. Subsequently, we obtain the final circuit presented in Fig. 5.

random cat state size and squeezing after the cumulative steps of PhANTM.

For the simulations, a Fock dimension of 60 is used and 1000 trials are run. Additionally, to enhance computational efficiency, we post-select homodyne measurements at 0, thereby avoiding the needs for further operations to correct the displacements induced at each PhANTM step. Effect of non zero homodyne measurement results on the weight unbalance of the cat states are discussed in [55].

Each output state is fitted with a cat state wavefunction to determine the size and squeezing as well as the parity of the cat state. To isolate dimensional issues in simulations, only states with a fidelity greater than 95% are retained, and these are represented in Fig. 14. These cat states are subsequently utilized for the adaptive breeding stage.

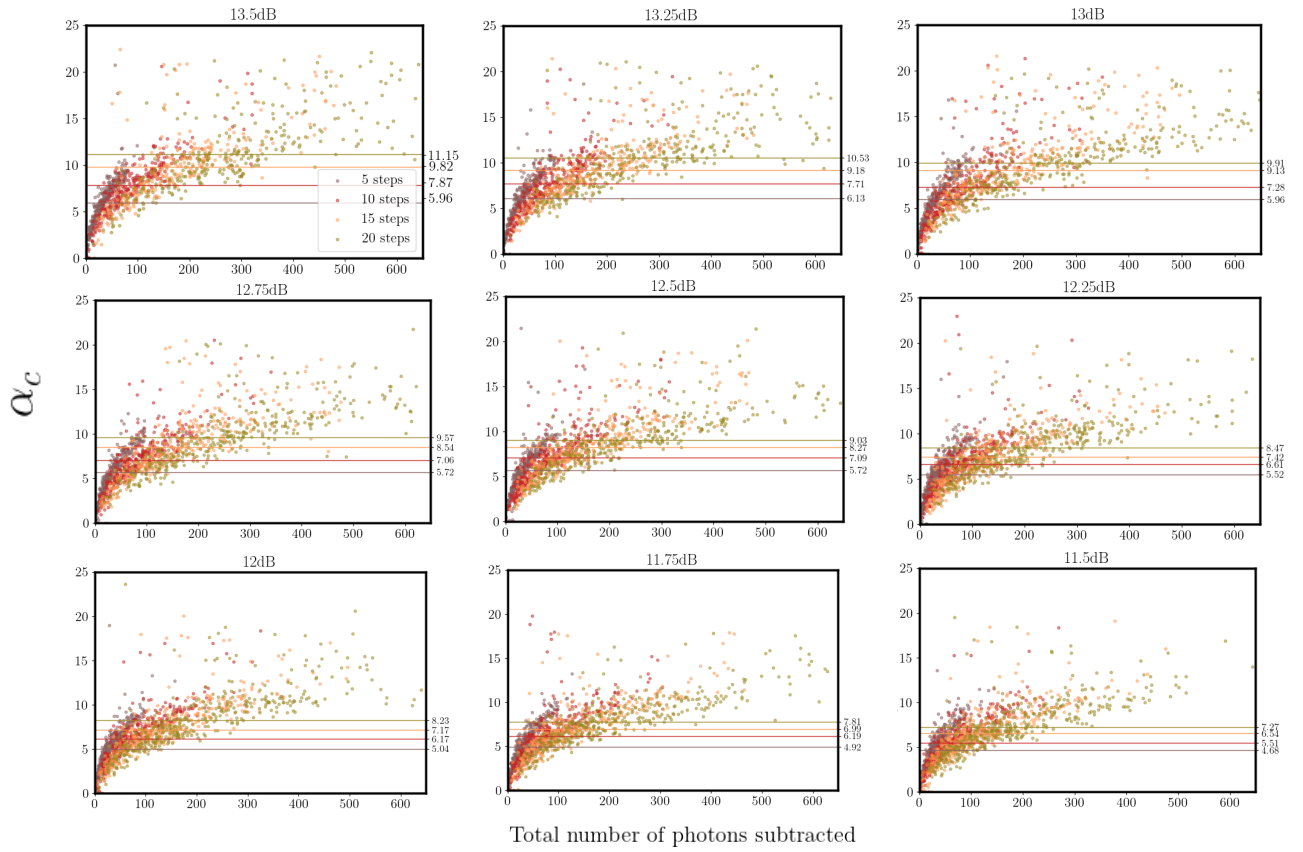


Figure 14: Results of PhANTM simulations (500 iterations for each dataset). Corrected alpha (α_c) as a function of the total number of photons subtracted for different cluster squeezing and different number PhANTM steps. Horizontal line shows mean of α_c for each number of PhANTM step. Mean values are reported on the right axis of each plots

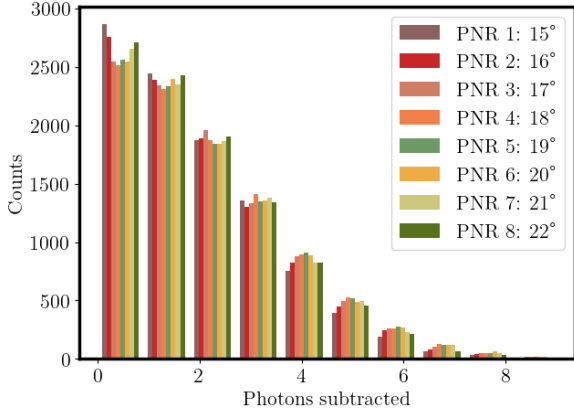


Figure 15: Histogram of photons detected for each of the 8 PNR detectors in the Cat generation process through 20 steps of PhANTM (500 iterations). Data are taken from Monte Carlo simulation of PhANTM with 13.5dB of cluster squeezing. Angle of the beamsplitter for each subtraction is given in the legend.

11.4 Photon number resolution

Simulation results show that only up to 8 photon subtraction events are required to make high quality cat and GKP states. Fig. 15, shows a histogram of the results of photon subtraction measurements assuming eight PNR detectors in PhANTM Monte Carlo simulations. The histograms appear largely uniform due to the gradient applied within the beamsplitter angle evolved in subtractions. CVQC schemes have traditionally needed photon resolutions of 20 to 50 photons to generate GKP states for FTQC with a success probability of a few percent (see Sec. 5).

11.5 Squeezing gate after PhANTM in dual rail cluster

In Fig. 16, it is shown that the squeezing gate is applied on a neighbouring mode of the dual rail where is applied PhANTM. Both operations can't be applied within the same dual rail without utilizing a switch since for PhANTM, a photon subtraction is required. At stage **II** and time step τ_i , mode 3_a is measured in the p basis to create then a C_Z connection between mode 2 and 4. In **IV**, by applying \mathcal{O} in mode 2 at the τ_{i+1} time step, a homodyne measurement is performed in the p basis. Then the quantum state is teleported in mode 4 where corrected displacement can be applied. By choosing correctly local oscillator phase in homodyne detection at the τ_{i+1} time step, the

squeezing gate can be applied from τ_{i+1} to τ_{i+2} in mode 4. Derivation of the squeezing gate can be found in [58, 52] in the case of a beamsplitter with a phase $\phi = 0$. In that case the squeezing gate, \mathcal{S} , can be written as

$$\mathcal{S} = N_m C_m V_i \quad (15)$$

where N_m and C_m are respectively noise and displacement operators. V_i can be expressed as:

$$V_i = R(\theta_{i,+})S(\tan(\theta_{i,-})R(\theta_{i,+})) \quad (16)$$

with

$$\theta_{i,\pm} = \frac{\theta_{i,a} \pm \theta_{i,b}}{2} \quad (17)$$

where $\theta_{i,a}$ and $\theta_{i,b}$ are angles of homodyne detection. By using a beamsplitter with a phase of $\phi = \pi/2$, the formula 15 remains valid, with the distinction that the angle $\theta_{i,a}$ and the input state are both rotated by $\pi/2$.

11.6 Macronode RHG lattice construction

Here, we describe some of the specifics of the reduction from the macronode version to the RHG lattice to the canonical cluster state. A full derivation is given in ref [27] so we will give only a brief introduction here. The protocol is illustrated in Fig. 17. A balanced four-mode splitter is used to distribute the entanglement. To recover the measurements on the canonical RHG lattice we assign one of the four physical nodes as the central mode, which we label as mode 0. This can be measured in either the p or q basis, depending on which basis we would wish to measure the RHG lattice node in. The remaining three auxiliary modes are always measured in the q basis. To map from the physical measurement on the central and auxiliary modes to the equivalent canonical measurement result we require an appropriate dictionary protocol. The dictionary protocol is derived uses the fact that a beam-splitter can be expressed as two C_X gates and two inline squeezing gates. C_X gates with control modes directly before a measurement can be replaced by a q displacement on the target mode, dependent on the measured mode. The inline squeezing operations can be commuted through any remaining C_X gates and displacements to the end of the circuit. The squeezing gates on any auxiliary modes are then absorbed into the q basis measurement.

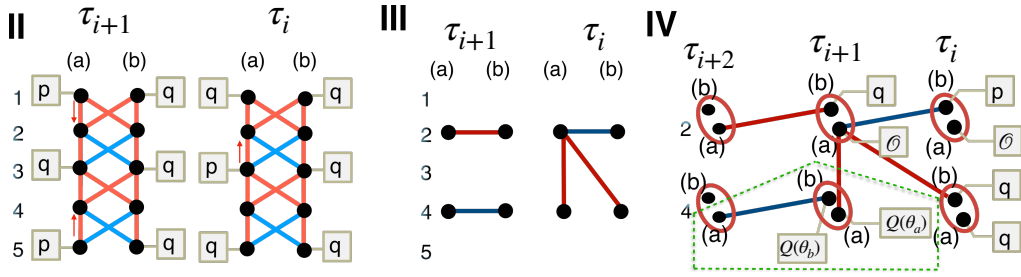


Figure 16: Cluster engineering for squeezing gate application. Stage I is not shown to avoid repetition with Fig. 4. In IV, is illustrated the application of the squeezing gate in mode 4 within the time step τ_{i+1} in the green dash box. The squeezing parameter is determined by the angles θ_a and θ_b of the homodyne detections. In mode 2, is performed a single PhANTM step, then the state is transferred to mode 4 before applying the squeezing gate.

This results in a single squeezing gate on the central mode, which re-scales the measurement results by 2 or 0.5 depending on the measurement basis, and an X correction (q -basis displacement) that depends on the q basis measurement results of the auxiliary modes. The remaining C_X gates are commuted backwards past the GKP Bell pairs which introduces a new set of C_Z gates which mirror those of the original lattice, and a set of Z corrections (p -basis displacements) that depend on the measurement results from the neighboring macronodes. This results in the same entanglement structure as the canonical lattice, up to a squeezing operation and X and Z corrections on the central mode.

11.7 GKP binning

In this section we describe the process of mapping CV displacement errors to qubit-level errors. This is referred to as the "inner decoder" in Refs. [26, 27] as it is the process of decoding the GKP code.

We employ the simple binning process where each homodyne value is rounded to the nearest multiple of $\sqrt{\pi}$. Odd and even multiples define the two logical qubit states. The error probability can be determined by determining how much of the Gaussian distribution of each bin overlaps with the neighboring bins [84]. However, this approach loses some information that can be gained from the position of the homodyne value relative to its bin. For example, a homodyne value found in the center of a bin is much less likely to have come from the neighboring bin than one towards the bin edge. We can quantify the error probability for a specific homodyne

value, z , and a GKP tooth width Δ as [84, 26].

$$p_{\text{err}}(z, \Delta) = \frac{\sum_{n \in \mathbb{Z}} \exp \left[- (z - (2n + 1)/\sqrt{\pi})^2 / \Delta^2 \right]}{\sum_{n \in \mathbb{Z}} \exp \left[- (z - n/\sqrt{\pi})^2 / \Delta^2 \right]} \quad (18)$$

We are interested in the error rate on the effective measurements on the canonical lattice. That is after the physical measurements have been post processed. As we have GKP states on every physical mode in each macronode, it is sufficient to determine the probability that the central mode measurement and each correction is incorrectly binned [27]. This is because, to first order, the total error probability of the macronode is given by the sum of these individual probabilities.

As well as the linear optics which can alter the GKP noise, the measurement result post processing also has an effect on the effective distribution that the combined corrections are sampled from. Ref.[27] includes a full description of both p and q basis measurements (q basis are required for logical gates) and for inclusion of p -squeezed states replacing some GKP states. Here we include description of the process for determining the error on p -basis measurements as these are the only basis required the threshold simulations we perform.

The macronode correction includes a squeezing gate and X and Z corrections. For a p -basis measurement on the central mode resulting in a homodyne value m_{central} , the squeezing gate maps $p \rightarrow 2p$. This maps the GKP effective squeezing $\Delta \rightarrow 2\Delta$, resulting in an er-

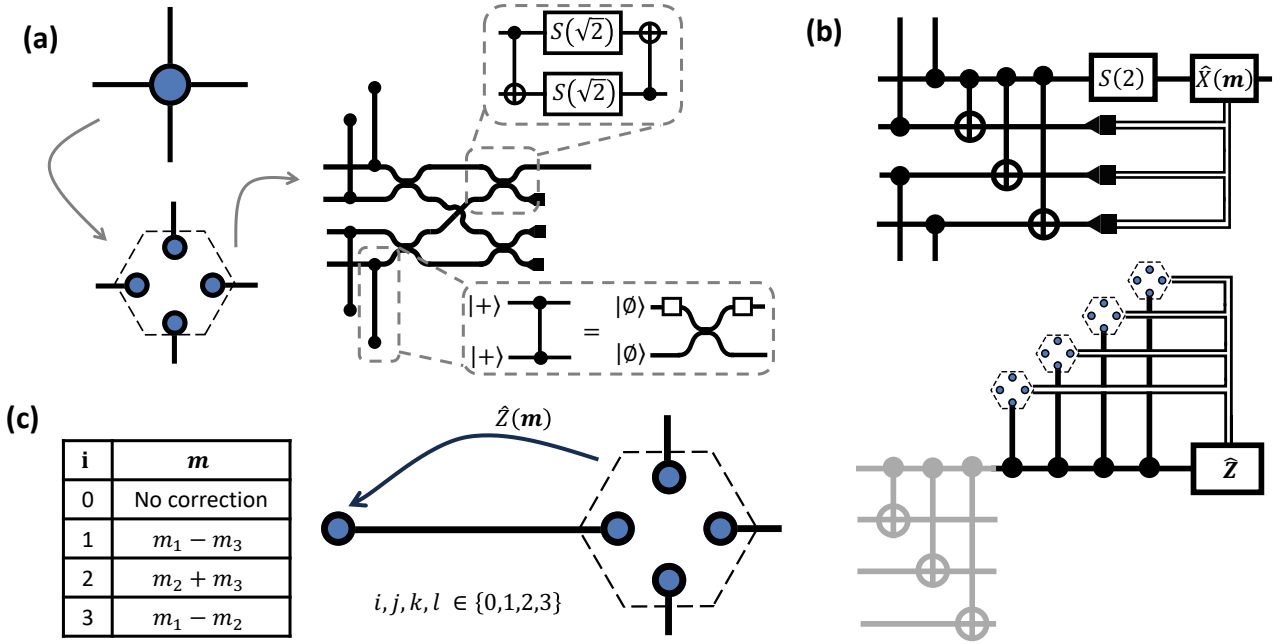


Figure 17: Illustration of the dictionary protocol for the macronode RHG lattice. a) Each node in the RHG lattice is replaced by a four node macronode. Bell pairs can be created by sensor states interfered on a passive beamsplitter (shown in the lower call-out box). A balanced four-mode splitter is used to generate the long range entanglement. Note, that while we have drawn a circuit with C_Z for the Bell pair generation, the physical circuit only consists of beamsplitters and phase rotations (shown with white squares.) The dictionary protocol can be derived by noting that the beamsplitters can be represented as C_X and squeezing gates (shown in the upper call-out box). By commuting these gates through each other we can arrive at the same entanglement structure as the canonical lattice with a squeezing correction and an X and Z corrections, corresponding to a q and p basis displacements, respectively. b) Effective circuit after the X correction has been derived. The X correction depends only measurements from auxiliary modes within the macronode. $\mathbf{m} = \frac{1}{2}(m_2 + m_3 - m_4)$ where m_i is the q -basis measurement result from the i^{th} auxiliary mode. c) Effective circuit after the X and Z corrections have been derived, equivalent to the canonical RHG lattice. Commuting the three C_X gates from (b) through the C_Z gates adds extra C_Z gates with support on the central mode. C_Z gates with support on auxiliary modes can be replaced with measurement based Z corrections. Once commuted through the remaining C_Z gates, the resulting Z corrections on each central mode will now depend on measurement results from neighboring macronodes. The exact form of the correction depends on which mode the in the neighbor had support on the original C_Z link. We show these corrections in the table. i corresponds to mode index in the neighboring macronode. Mode 0 is the central mode.

ror probability of $p_{\text{err}}^{\text{central}} = (2m_{\text{central}}, 2\Delta_{\text{central}})$. The X corrections have no effect on the p -basis measurements so we do not consider them here. The Z corrections are all the sum (or difference) or two independent normally distributed variables. This results in distribution with a variance that is the sum of the two normal distributions. This results in a error probability $p_i^{\text{corr}} = (m_j \pm m_k, \sqrt{\Delta_j^2 + \Delta_k^2})$, where modes j and k are given by the specific correction, as outlined in Sec. 7.1. In the case where a correction is 0, that is the i^{th} macronode is connected to the central mode of the neighboring macronode, $p_{\text{err}}^{\text{corr}} = 0$, as shown in Fig. 17(c).

11.8 Logical error simulation method

We follow a similar method to Ref. [26] where we track the noise on each physical GKP mode. We start by assigning a separate noise variances in the p and q quadratures for every mode. This variance is either an abstract quantity defining the general noise channel in Eq. 8 or the result from circuit-level PhANTM, squeezing, and breeding simulations. To determine the noise on the homodyne values, we propagate these variances using the symplectic representation of the beamsplitter network required to create the RHG lattice. The symplectic form, \hat{S} , transforms the quadrature values as $\mathbf{x}' = \hat{S}\mathbf{x}$, where $\mathbf{x} = (q_1, q_2, \dots, q_N, p_1, p_2, \dots, p_N)$. The noise vari-

ances are then transformed as

$$\Sigma_{\text{out}} = \hat{S}^T \Sigma_{\text{in}} \hat{S}, \quad (19)$$

where Σ is a $2N \times 2N$ diagonal noise matrix, where the first N entries are $\Delta_{q_i}^2$ and the second N entries are $\Delta_{p_i}^2$. We can simulate the noisy GKP measurements by drawing samples from the multi-variate distribution centred at 0 with a covariance matrix given by Σ_{out} . Note that to simplify the simulations, we sample from the diagonal elements of Σ_{out} .

We can convert these noisy measurements of the physical modes into error probabilities for the RHG lattice using the process described in Appendix 11.7.

For a code distance d , we construct a $d \times d \times d$ lattice with periodic boundary conditions in all dimensions. Error correction proceeds using the code's parity check matrix, a binary matrix with elements $H_{ij} = 1$ if qubit i is involved in parity check j and 0 otherwise. We can then use the probabilities calculated above to generate a binary noise array, with element $\epsilon_i = 1$ with probability p_i . An element equal to 1 corresponds to a flipped measurement outcome, e.g. a Pauli Z error on qubit i . The corresponding syndrome can be found by matrix multiplication, modulo 2, $s = H\epsilon$. To decode the syndrome we use a minimum weight perfect matching decoder implemented in python with PyMatching [87]. This aims to return a possible error chain consistent with both the syndrome and parity check matrix. As in previous works, [86, 28, 26, 85], we can utilize the continuous nature of our measurements to increase the success probability of the syndrome decoder. We convert the error probabilities into weights for the decoder as $w_i = -\log(p_i)$. A logical error corresponds to the case where the combination of the original error and the decoder output result in an error chain that spans the lattice. This can be determined by checking the number of intersections of the resultant error chain with the logical operator correlation surface of the lattice. An odd number of intersections corresponds to logical error. Repeating this process many times allows us to determine the logical error rate for a particular GKP effective squeezing and code distance. As the primal and dual lattices can be decoded independently, we only show results for the primal lattice. The procedure for the dual lattice is identical.

11.9 Loss tolerance

As mentioned in the main text, the effects of losses on cat states generated from PhANTM have been explored in [55]. Here, we discuss losses in adaptive breeding and QEC stages of our architecture.

As in PhANTM [55], we model losses applied on an input state ρ_{in} in adaptive breeding by mixing the state with vacuum on a beam splitter with a transmission coefficient η and tracing over the reflective mode of the beam splitter, arriving at the output density matrix ρ_{out} from a loss channel given by

$$\rho_{out} = \sum_{l=0}^{\infty} \mathcal{L}_l \rho_{in} \mathcal{L}_l^\dagger, \quad (20)$$

where each \mathcal{L}_l is a Kraus operator expressed as

$$\mathcal{L}_l = \sqrt{\frac{(1-\eta)^l}{l!\eta!}} a^l e^{\frac{1}{2}a^\dagger a \ln(\eta)}, \quad (21)$$

and l is an index for the Fock basis. Since adaptive breeding is performed on cat states and momentum squeezed states embedded in a cluster, first order losses come from homodyne detections which are modeled by putting a lossy channel before detection. Fig. 18a shows the effective squeezing of the GKP states in the p and q quadratures as a function of homodyne loss after three breeding rounds. To isolate the effect of losses from other random variations in the adaptive breeding protocol, we use a set of eight identical cat states as input and vary the squeezing r_c of the cat states (the amplitude of the cats is fixed by the number of breeding rounds). We observe that the effective squeezing in the q quadrature is not sensitive to the losses up to 5% loss. The effective squeezing in the p quadrature is more sensitive than that in the other quadrature, but changes by less than 0.5 dB up to $\eta_{hd} \approx 0.03$.

For QEC, loss on a GKP state can be mapped to the Gaussian displacement channel by rescaling any homodyne values [85]. For transmission η (representing the propagation loss), this results in a GKP state with a with variance $\Delta_{\text{loss}}^2 = \frac{1-\eta}{2\eta}$. If we have a GKP state with some initial squeezing defined by the variance Δ_{GKP}^2 , then the total variance, including the loss, is $\Delta_{\text{total}}^2 = \Delta_{\text{GKP}}^2 + \Delta_{\text{loss}}^2$. For the case where we have balanced squeezing in both quadratures,

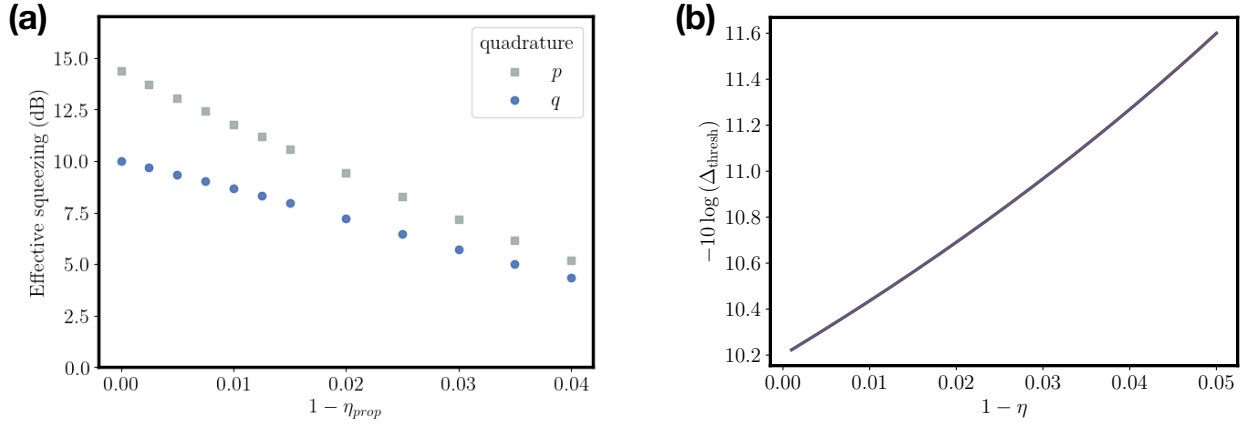


Figure 18: a) Effective squeezing in p (square) and q (circle) quadratures as a function of $1 - \eta_{hd}$ for different cat state squeezing values. Only one type of squeezing is shown for the p -quadrature, as for the other squeezing values the curves overlap with the one already on the plot. b) Loss vs squeezing threshold. We show the GKP effective squeezing that results in a total squeezing of 10.2 dB for given values of η .

our Monte-Carlo simulations produced a squeezing threshold of 10.2 dB. For uniform loss on each physical mode in the macronode lattice the variance on the noisy homodyne values is described by Δ_{total}^2 . We can use this to determine the squeezing threshold in the presence of finite loss by finding Δ_{GKP}^2 that, along with a given transmission η , gives $-10 \log(\Delta_{\text{total}}^2) = 10.2$ dB. Fig. 18b shows the result of this simulation, highlighting that the needed effective squeezing to be at the fault tolerance threshold in the presence of 3% loss is increased by less than 1 dB compared to the ideal case without loss.