

Optimizing fault-tolerant cat state preparation

Tom Peham^{1,*}, Erik Weilandt^{1,†} and Robert Wille^{1,2,‡}

¹*Chair for Design Automation, Technical University of Munich, Munich, Germany*

²*Munich Quantum Software Company, 80799 Munich, Germany*



(Received 10 January 2026; revised 20 April 2026; accepted 29 April 2026; published 9 June 2026)

Cat states are an important resource for fault-tolerant quantum computing, where they serve as building blocks for a variety of fault-tolerant primitives. Consequently, the ability to prepare high-quality cat states at large fault distances is essential. While optimizations for low fault distances or small numbers of qubits exist, higher fault distances can be achieved via generalized constructions with potentially suboptimal circuit sizes. In this work, we propose a cat state preparation scheme based on preparing two cat states with low-depth circuits, followed by a transversal CNOT and measurement of one of the states. This scheme prepares w -qubit cat states fault tolerantly up to fault distances of 9 using $\lceil \log_2 w \rceil + 1$ depth and at most $3w - 2$ CNOTs and $2w$ qubits. We discuss that the combinatorially challenging aspect of this construction is the precise wiring of the transversal CNOT and propose three methods for finding these: two based on Satisfiability Modulo Theory solving and one heuristic search based on a local repair strategy. Numerical evaluations show that our circuits achieve a high fault distance while requiring fewer resources than generalized constructions.

DOI: [10.1103/13v7-n843](https://doi.org/10.1103/13v7-n843)

I. INTRODUCTION

Quantum error correction (QEC) and Fault-tolerant quantum computing (FTQC) [1–4] enable a pathway towards utility-scale quantum computing, where billions of operations need to be performed reliably on encoded quantum information [5]. By redundantly encoding information into quantum error-correcting codes and repeatedly extracting error syndromes, the lifetime of logical qubits can be extended far beyond what is possible for physical qubits. Recent experimental advances show that current hardware platforms exhibit the capabilities required for FTQC, like real-time, below-threshold decoding [6], encoded universal gate sets [7], and first encoded algorithms [8].

Realizing large-scale FTQC requires that every step of a computation be implemented in a fault-tolerant manner, so that physical errors do not spread uncontrollably through the system. Instead of viewing fault tolerance at the level of an entire algorithm, it is common to decompose computations into subcircuits or *gadgets* [9], each of which is itself fault tolerant. If errors are removed quickly via decoding and error correction, they do not accumulate, and information can be protected with high probability. From this perspective, a central circuit-design task in FTQC is to devise and optimize fault-tolerant gadgets for tasks such as syndrome extraction

[10–13], ancilla state preparation [14–16], magic-state preparation [17–19], code switching [20,21], or measurement-free protocols [22,23], among others. While such circuits are often designed manually or for specific code instances, developing general automated compilation and optimization techniques for fault-tolerant circuit constructions has started to gain more and more traction [14,15,24–27].

Despite first demonstrations of fault tolerance, state-of-the-art quantum hardware remains strongly resource constrained [6,8,28–30]. This means that successful near-term fault-tolerant quantum computing requires careful optimization of fault-tolerant circuits employed during computation. Even large-scale computations that reuse the same subcircuits often greatly benefit from circuit optimizations, as any optimization to a gadget has an immediate multiplicative impact on resource savings.

This work focuses on optimizing fault-tolerant preparation circuits of specific resource states, so-called “cat states,” which are w -qubit states of the form $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes w} + |1\rangle^{\otimes w})$ (also referred to as Greenberger-Horne-Zeilinger states [31]) or $\frac{1}{\sqrt{2}}(|+\rangle^{\otimes w} + |-\rangle^{\otimes w})$. Such states are a useful resource for FTQC since they can be used for syndrome extraction [2,32,33], magic-state preparation [17], or fault-tolerant encoders [34].

Given the conceptually simple and useful nature of these states, it is not surprising that fault-tolerant circuit designs for such states have already been investigated. Reference [11] showed that distance-three cat state preparation can be done with logarithmic overhead, but the overhead increases for preparing cat states with a higher fault distance. Reference [26] gives a construction for arbitrary-sized cat states requiring logarithmic depth and linearithmic two-qubit gates and qubits. Another approach is via flag-fault-tolerant constructions, as any higher-distance flag-fault-tolerant syndrome

*Contact author: tom.peham@tum.de

†Contact author: erik.weilandt@tum.de

‡Contact author: robert.wille@tum.de

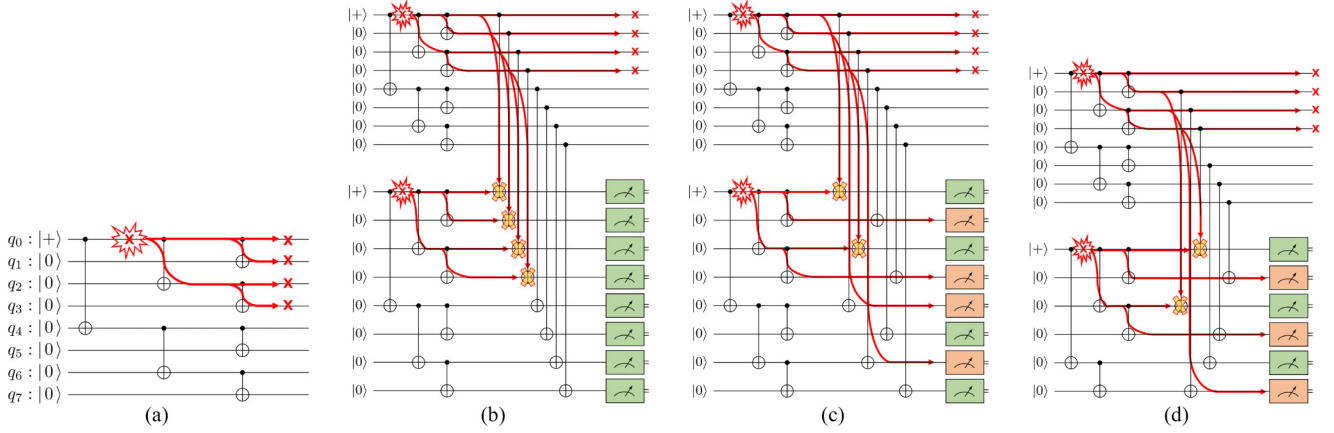


FIG. 1. Fault-tolerant cat state preparation. (a) Non-fault-tolerant log-depth preparation of an 8-qubit cat state. A single weight-one error propagates to a weight-four error on the prepared state. (b) Errors on the data cat state can be detected by an ancilla cat state by copying the X errors to the ancilla and measuring in the Z basis. Here this is not fault tolerant as two errors are undetected and lead to a weight-four error on the data qubits. (c) The error detection gadget can be made fault tolerant against 4 errors by changing the connectivity of the transversal CNOT. (d) Qubit overhead can be reduced by using a smaller ancillary cat state; a 6-qubit cat state with an appropriately connected transversal CNOT is sufficient.

measurement scheme can be turned into a fault-tolerant cat state preparation scheme requiring worst-case linear depth, two-qubit gate, and qubit overhead [10,16].

In this work, we aim to further optimize the resource requirements for fault-tolerant cat state preparation. We propose a scheme for the fault-tolerant preparation of cat states using two (potentially differently sized) cat states, copying errors from one state to the other, measuring the ancilla state, and postselecting on the error-free measurement result (see Fig. 1). This construction prepares a cat state using logarithmic depth and a linear number of two-qubit gates and qubits. The challenging part of this construction is finding certain qubit permutations to avoid errors canceling in such a way as to violate fault tolerance of the preparation schemes.

Our main contributions are as follows:

- (1) We propose a fault-tolerant cat state preparation scheme that uses two low-depth cat states, a single transversal CNOT layer, and postselection. This yields w -qubit cat states with logarithmic CNOT depth and linear overhead in qubits and two-qubit gates.
- (2) We develop several combinatorial search methods for determining suitable transversal CNOTs that ensure fault tolerance, including a counterexample-guided abstraction refinement (CEGAR)-based search, a satisfiability/satisfiability modulo theories (SAT/SMT) formulation, and a heuristic local-repair strategy.
- (3) We apply these methods to synthesize fault-tolerant cat state preparation circuits for up to 49 qubits for a fault distance of 4 and 19 qubits for a fault distance of 9.
- (4) We evaluate the constructed circuits using circuit-level noise simulations. The resulting circuits exhibit reduced qubit and gate overheads, as well as improved acceptance and error rates, compared to existing general-purpose constructions.

The remainder of this work is structured as follows. Section II reviews the necessary background on cat states and fault tolerance. In Sec. III, we introduce the fault-tolerant cat state preparation scheme studied in this work and formalize the conditions under which a (partial) transver-

sal CNOT preserves fault tolerance. Section IV reviews existing fault-tolerant cat state preparation constructions. In Sec. V, we analyze the error-propagation structure of balanced-tree cat state preparation circuits and show how this structure can be exploited to restrict the search space for fault-tolerant constructions and efficiently checking the fault tolerance of a given candidate CNOT. Section VI presents the combinatorial search methods used to synthesize fault-tolerant transversal CNOTs. Finally, Sec. VII evaluates the resulting circuits, and Sec. VIII concludes the paper.

II. BACKGROUND

Throughout this work we use $[i] = \{1, 2, \dots, i\} \subseteq \mathbb{Z}$. The Pauli matrices are given by

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix},$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix},$$

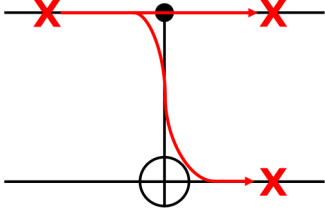
and together form the single-qubit Pauli group, which naturally extends to the Pauli group on multiple qubits. In this work, we consider Pauli operators that are pure X or pure Z type, i.e., w -qubit operators of the form $\prod_{i=1}^w \sigma_i$ with $\sigma_i \in \{I, X\}$ or $\sigma_i \in \{I, Z\}$.

We will focus on the preparation of w -qubit Greenberger-Horne-Zeilinger (GHZ) states, that is, states of the form $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes w} + |1\rangle^{\otimes w})$. Preparation of $\frac{1}{\sqrt{2}}(|+\rangle^{\otimes w} + |-\rangle^{\otimes w})$ works analogously. GHZ states are stabilizer states on w qubits defined by the stabilizers $X^{\otimes w}$ and $Z_i \otimes Z_{i+1}$, $i \in [w-1]$.

Any Z -type operator with an odd support size anticommutes with the $X^{\otimes w}$ operator and an arbitrary X -type operator anticommutes with at least one of the Z stabilizers. Therefore, one can detect up to $w-1$ X (bit flip) errors and any odd number of Z (phase flip) errors on the state. Since any

two-qubit Z error is a stabilizer, any Z error is stabilizer equivalent to a trivial error or a single-qubit error.

A *fault* is an error event occurring during the execution of a circuit, and these faults can propagate through the circuit. X faults propagate through the control of a CNOT gate, i.e., if the control qubit flips, the target qubit does too.



An *error* is the fault propagated to the end of the circuit. We denote an arbitrary w -qubit X error as an $\mathbb{F}_2^w$ vector where the vector has nonzero entries exactly on the indices corresponding to the support of the error, e.g., the vector $[1,0,1]$ corresponds to the operator XIX . For a GHZ state, any X error $e \in \mathbb{F}_2^w$ is equivalent to the error $\mathbb{1}^{\otimes w} + e$ obtained by flipping all bits in e since $X^{\otimes w}$ is a stabilizer of the state. The *weight* of an error e on a cat state (denoted $\text{wt}(e)$) is defined as the number of nonzero entries in e or $\mathbb{1} + e$, whichever is smaller. Multiple faults can occur simultaneously in a circuit, causing multiple errors. If two faults cause errors e_1 and e_2 , the resulting combined error is given by $e_1 + e_2$. The set of errors caused by up to t faults in a circuit $\mathcal{C}$ is denoted as $\mathcal{E}_t(\mathcal{C})$, which we call the (t -) *fault set* of the circuit $\mathcal{C}$. If the circuit in question is clear from context, we will often only write $\mathcal{E}_t$ for the t -fault set.

Errors in a circuit can be detected via certain measurements that do not disturb the state.

Definition 1. We say that a cat state preparation scheme is fault tolerant up to t errors, FT^t , if any combination of weight w (up to t faults) in the circuit causing an error $e \in \mathcal{E}_t$ on the resulting cat state with weight greater than w is detected.

III. CAT STATE VERIFICATION VIA TRANSVERSAL CNOTS

Since cat states are Calderbank-Shor-Steane (CSS) states, they can be prepared by initializing qubits in $|+\rangle$ followed by a CNOT circuit [15,35]. Specifically, since cat states have only a single X -type stabilizer, one qubit is prepared in $|+\rangle$ and the rest in $|0\rangle$. The following CNOT circuit must then propagate the initial X stabilizer to the other qubits. This can be done in $\lceil \log_2 w \rceil$ depth with a “balanced tree” CNOT circuit as shown in Fig. 2(a), where the cat state is doubled in size at every layer by entangling the current state with the same number of $|0\rangle$ qubits. If w is not a power of 2, the last layer only entangles $2^{\lceil \log_2 w \rceil} - w$ qubits. We refer to this circuit structure as $\mathcal{B}_w$.

This preparation is not even FT^1 in general, as single errors on a control in the circuit propagate to an X error on the entire subtree, as shown in the preparation of an 8-qubit cat state using the circuit $\mathcal{B}_8$ in Fig. 2(a).

In general, such high-weight propagated errors can be detected using an ancilla cat state by copying the errors via a transversal CNOT and measuring the ancilla cat state in the Z basis. The transversal CNOT acts trivially on the cat state

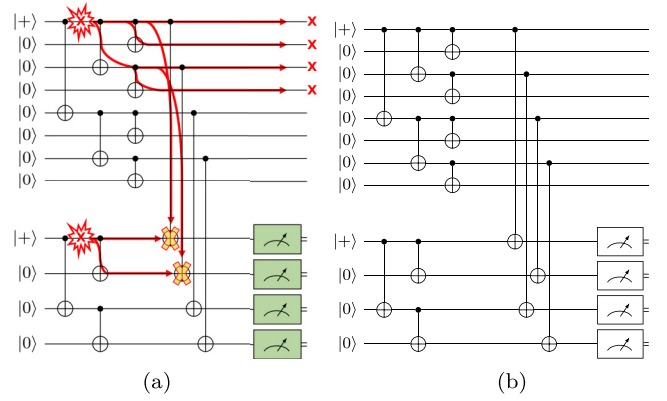


FIG. 2. FT^2 cat state preparation. (a) Non- FT^2 permutation. Two faults lead to a weight four fault on the data. (b) FT^2 permutation.

as the $X^{\otimes w}$ stabilizers of the “control” cat state propagate to a stabilizer of the ancilla cat state, and similarly for the Z -type stabilizers. Assuming the ancilla cat state is error-free, copying an error will flip bits in the cat state according to the support of that error. In the absence of errors, measuring the ancillary cat state in the Z basis yields either bit string 0^w or 1^w . Any flipped bit in the measurement result, therefore, reveals the presence of an error.

However, if the ancilla cat state is not error-free, low-weight faults in the circuits may lead to high-weight errors going undetected. In particular, if both cat states are prepared using the same circuit structure, any two faults occurring in the same location in both circuits will cancel out after the transversal CNOT.

Example 1. Consider the circuit in Fig. 1(b). It prepares two 8-qubit cat states using the circuit $\mathcal{B}_8$. A transversal CNOT is used to copy errors from the first 8 qubits onto the second set of 8 qubits. In the depicted scenario, the fault $X_1X_2X_3X_4$ is copied to the ancilla cat state but remains undetected because the same fault occurs during the preparation of the ancilla cat state. Since the resulting residual fault on the prepared cat state has weight 4, the cat state preparation in Fig. 1 is not FT^2 .

Error cancellations can be avoided by permuting the wiring of the transversal CNOT. Since any permutation of the qubits of a cat state leaves the state invariant, any transversal CNOT with $CX_{i,\sigma(w)}$ for arbitrary permutations σ in the symmetric group S_w acts trivially on the cat states while copying errors from one state to the other. This fact can be exploited to render the error detection fault tolerant. Intuitively, as long as it can be ensured that high-weight faults do not cancel out or combine to the stabilizer $X^{\otimes w}$ after permuting the qubits, all problematic errors will be detected.

Detecting errors on a cat state using a transversal CNOT has the benefit of only adding a single layer of CNOTs to the preparation. If the states are prepared using $\mathcal{B}_w$, then preparation has CNOT depth $\lceil \log_2 w \rceil + 1$ using $3w - 2$ CNOTs and postselects on w qubits.

Example 2. Rewiring the transversal CNOT in Fig. 1(b) to get the circuit in Fig. 1(c) according to

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 5 & 3 & 7 & 2 & 6 & 8 & 4 \end{pmatrix},$$

prevents the faults from canceling out. In fact, no combination of up to $t < 4$ faults in this construction cancels out such that an error of weight greater than t is undetected. Thus, the resulting circuit is FT^4 .

This CNOT and qubit overhead can be further reduced by using a smaller ancillary system. If the ancilla state has $1 < w' < w$ qubits, then a transversal CNOT between any w' qubits in the first circuit and the second circuit acts trivially on the system. We can characterize such a partial transversal CNOT by how it connects qubits of the first and second circuits. Concretely, it is given by a set of CNOT pairs

$$\overline{\text{CX}} \subseteq [w] \times [w']$$

with $|\overline{\text{CX}}| = w'$ such that no index appears more than once on either side. Equivalently, $\overline{\text{CX}}$ defines a bijection between the w' ancilla qubits and a subset of w' data qubits. Such a CNOT has an associated *mapping* $f : \mathbb{F}_2^w \rightarrow \mathbb{F}_2^{w'}$ of errors on w qubits to errors on w' qubits defined on the i th single-qubit error $\hat{e}_i$ as

$$f(\hat{e}_i) = \begin{cases} \hat{e}_j & \text{if } (i, j) \in \overline{\text{CX}} \\ \mathbf{0} & \text{else.} \end{cases}$$

Error cancellations must still be avoided by connecting the qubits of the two cat states appropriately. To define what it means for such a cat state preparation circuit to be FT^t , we introduce the following definition.

Definition 2. Let $\mathcal{C}, \mathcal{C}'$ be two cat state preparation circuits over w qubits and $w' \leq w$ qubits, respectively. A (partial) transversal CNOT $\overline{\text{CX}}$ with associated mapping $f : \mathbb{F}_2^w \rightarrow \mathbb{F}_2^{w'}$ is called *T -fault-tolerant* (or FT^T) for $T > 0 \in \mathbb{Z}$ if for all $t, t' \geq 0$ with $t + t' \leq T$ there do not exist

$$e \in \mathcal{E}_t(\mathcal{C}), \quad e' \in \mathcal{E}_{t'}(\mathcal{C}')$$

such that the combined error

$$f(e) + e' \in \{I^{\otimes w'}, X^{\otimes w'}\}$$

while

$$\text{wt}(e) > T.$$

Example 3. Consider again the FT^4 preparation circuit for an 8-qubit cat state in Fig. 1(c). As it turns out, a 6-qubit ancilla state is sufficient to make this construction FT^4 by choosing the partial transversal CNOT

$$\{(2, 3), (3, 6), (4, 1), (6, 5), (7, 4), (8, 2)\}. \quad (1)$$

The resulting circuit is shown in Fig. 1(d). Six qubits is also the smallest possible ancilla state to make this construction FT^4 .

The objective of this work is precisely to construct such fault-tolerant cat state preparation schemes using (partial) transversal CNOTS. Note that the size of the circuit only depends on the size of the ancillary state. Since we want to minimize resource costs, i.e., depth, CNOT count, and postselection overhead, we want to find the ancilla with the smallest number of qubits possible.

Before discussing how we tackle this optimization problem, we will briefly review related fault-tolerant cat state preparation constructions.

TABLE I. Resource scaling of different $\text{FT}^t w$ -qubit cat state preparation methods.

Method	Gate count	Depth	Qubits
This method	$O(w)$	$O(\log w)$	$O(w)$
Ref. [26]	$O(w \log w)$	$O(\log w)$	$O(w \log w)$
Refs. [10,16]	$O(w)$	$O(w)$	$O(w)$

IV. RELATED WORK

Recently, Ref. [26] has derived a w -fault-tolerant cat state preparation scheme based on rigorous *distance-preserving* rewrite rules on ZX diagrams [36,37]. The construction works by recursively constructing w -qubit cat states from $w/2$ qubit cat states until the base case of 4 qubits is reached, for which a simple construction using 5 qubits, 5 CNOTS, and 1 measurement is taken. The combination works by measuring $w/2$ XX parity checks between pairs of qubits of cat states of size $w/2$. Therefore, every combination step requires $w/2$ extra qubits, w CNOT gates, and postselects on $w/2$ measurements. In total this method requires $\frac{w}{2} \log_2 w + \frac{w}{4}$ qubits, $w \log_2 w - \frac{3}{4}w$ CNOTS, and a depth of $2 \log_2 w + 1$ if w is a power of 2. If the desired cat state preparation should be FT^t with $t < w$, then the requirements are more relaxed, as at most t parity checks need to be made in the recursive combination.

Another construction of note is based on flag-fault-tolerant measurements [10]. Flag-based constructions work by detecting errors between pairs of CNOTS interacting with an ancilla qubit that act trivially on the system in the absence of errors. This method requires initializing ancillas using a depth $w - 1$ circuit where the qubit initialized in $|+\rangle$ acts as the control qubit of all CNOTS. Recently, Ref. [16] has introduced a method for finding optimal error-detecting flag gadgets. While the construction has been introduced in the context of state preparation of arbitrary CSS codes, it can also be used for cat state preparation. They manage to find circuits for up to $t = 5$ faults and up to 29 qubits. Fully fault-tolerant cat state initialization is therefore possible for up to 10-qubit cat states (since we want to protect against up to $t = 5$ faults). In that case, the optimal construction requires 16 qubits (10 for the cat state and 6 for the flags), 21 CNOTS (9 for the cat state preparation and 12 for the flag construction), and has linear depth. While this construction postselects on fewer measurements for $w = 10$ than the proposed transversal construction, the depth overhead is much higher. Furthermore, we aim to construct larger cat states, which require fault tolerance beyond $t = 5$. Reference [10] also gives a general w -flag construction for this flag-based linear cat state preparation that requires $2w - 1$ qubits with resets ($3w - 1$ without resets), $3w - 3$ CNOTS, and depth $7w - 15$ (for larger w). This means that the general construction (almost) matches the transversal cat state preparation except for the linear depth overhead.

A brief comparison of the resource scaling is shown in Table I. This shows that asymptotically, preparing cat states using transversal verification scales most favorably in all considered metrics. In Sec. VII we will see that this is not only an asymptotic advantage but that the proposed cat state preparation method uses fewer resources than the recursive construction from Ref. [26] in almost all cases, even in the

very low qubit regime. However, the suggested improvements hinge on the fact that such t -FT permutations can actually be constructed.

Fault-tolerant state preparation using multiple copies of a state and a transversal is a well-known construction [38–40]. In fact our setting is very similar to Ref. [39] and Ref. [38] where fault-tolerant preparation of logical basis states of the quantum Reed Muller codes and the Golay code are achieved by constructing multiple copies of a state and finding an appropriate permutation of the transversal CNOTs. The codes considered in those works do not allow all possible permutations of qubits and are more restrictive. Furthermore, unlike in our case, the ancilla states in those constructions are fixed as no smaller states can be used for verification. The constructions in the two works are limited to distance 7 and 9 codes, which translates to a fault distance of 3 and 4, respectively. We aim to construct circuits with very high fault distance and utilize as few qubits as possible. To achieve this, we need to develop efficient combinatorial search algorithms to construct the desired cat state preparation circuits. In the next sections, we outline how we achieve this goal.

V. CONSTRUCTING FAULT-TOLERANT CAT STATE PREPARATION CIRCUITS

Given two cat state preparation circuits $\mathcal{C}, \mathcal{C}'$ of size w, w' with $w' \leq w$, the task is to construct a partial transversal CNOT that is fault tolerant in the sense of Definition 2. Since all w' qubits of the ancillary state serve as targets in this construction, the transversal CNOT can be characterized by

- (1) a choice of w' control qubits on the data cat state, and
- (2) a permutation of the target qubits on the ancilla cat state.

Only certain combinations of these two choices avoid error cancellations that would violate fault tolerance. Because the number of possible fault combinations grows rapidly with the fault distance, and because the joint search space of size $\binom{w}{w'} w'!$ is enormous, we rely on structural properties of the preparation circuit to guide the construction.

A. Fault structure of the preparation circuit $\mathcal{B}_w$

A key benefit of using the balanced-tree preparation circuit $\mathcal{B}_w$ is that its fault structure is highly regular. Any single X error occurring before a control in $\mathcal{B}_w$ propagates to the entire binary subtree below it. For $w = 8$, for example, the circuit has the 1-fault set

$$\begin{aligned} \mathcal{E}_1 = \{ & X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, \\ & X_1X_2, X_3X_4, X_5X_6, X_7X_8, \\ & X_1X_2X_3X_4, X_5X_6X_7X_8 \}. \end{aligned}$$

In general, $\mathcal{E}_1(\mathcal{B}_w)$ consists of all X errors supported on dyadic intervals of $[w]$, and thus has size $2w - 2$. The higher-order fault sets $\mathcal{E}_t(\mathcal{B}_w)$ are formed by adding these error patterns over $\mathbb{F}_2$. Due to the regular structure, many such combinations will coincide, thereby avoiding the storage of many redundant elements in memory. Nevertheless, the size of the t -fault set still grows exponentially in t .

The fault set structure is essential for two reasons:

- (1) it allows us to reason about which errors can propagate to the ancilla and cancel there, and
- (2) it enables an efficient check of whether a candidate partial transversal CNOT is fault tolerant.

B. Determining w' and a choice of control qubits

In the following, we assume that both $\mathcal{C}$ and $\mathcal{C}'$ are implemented using the balanced-tree circuit $\mathcal{B}_w$ (or $\mathcal{B}_{w'}$), as introduced in Sec. III. Before searching for a permutation of the ancilla targets, we must decide how many ancilla qubits are needed and which data qubits should act as controls. This is hard to decide in general, but we can narrow down the search due to the structure of the possible faults.

Let us consider the simplest case first, namely, constructing FT^2 cat state preparation circuits. If the ancilla had fewer than $w/2$ qubits, then two qubits participating in the same CNOT of the last layer of the data circuit would not be connected to the ancilla, making certain weight-2 propagated errors completely undetectable. Thus $w' \geq w/2$ is necessary, and in fact already sufficient. With $w/2$ controls, each size-2 subtree is connected to the ancilla, and the cancellations shown in Fig. 2 can be avoided by ensuring that entire subtrees never map to entire subtrees.

The size of a subtree can also be used to determine the minimal sizes of the required ancilla states. Consider, for example, a subtree with 4 qubits, which means that a single error can lead to a weight-4 error on the data qubits. If only one or two of the qubits are connected to the ancilla state, up to two single-qubit errors are sufficient to cancel the copied errors on the ancilla. This is completely independent of the target permutation. Therefore, to achieve FT^3 , at least three of those data qubits must be connected to the ancilla state. Since this has to be true for a size-4 subtree, we know that FT^3 cat state preparation requires an ancilla state of size at least $(3/4)w$.

This argument straightforwardly generalizes to larger subtrees as well. If there is a subtree of size 8, then FT^7 preparation requires that at least 7 of those qubits be connected to the ancilla state, and FT^7 preparation of (large enough) cat states therefore requires at least a $(7/8)w$ qubit ancilla.

Slightly larger ancillas may be needed once propagated ancilla faults are also taken into account, but these lower bounds greatly simplify the search for a possible choice of ancilla size and control qubits.

The structured nature of these fault sets allows for explicit constructions for specific cases. In Appendix B we sketch a general FT^4 construction that does not rely on combinatorial search.

C. Checking fault tolerance for a fixed set of controls

Once a candidate ancilla size w' and a set of control positions have been selected, the remaining task is to find a permutation of the w' ancilla qubits that yields an FT^t transversal CNOT. To search efficiently over permutations, we need a fast check for fault tolerance.

Let $\pi : \mathbb{F}_2^w \rightarrow \mathbb{F}_2^{w'}$ be the projection of an error onto the chosen control qubits. For each k , we precompute the projected fault set

$$\mathcal{P}_k = \pi(\mathcal{E}_k).$$

Many errors in $\mathcal{E}_k$ might have the same projection, so for each projected pattern $e' \in \mathcal{P}_k$, we retain only a maximal-weight representative

$$\mathcal{R}_k = \{\arg \max_{e \in \pi^{-1}(e')} \text{wt}(e) \mid e' \in \mathcal{P}_k\}.$$

These representatives capture exactly the error patterns that could compromise fault tolerance. Storing only $\mathcal{R}_k$ reduces the memory overhead significantly, as many errors in $\mathcal{E}_k$ are projected identically, and we only keep the worst-case preimage of each projection.

An error $e \in \mathcal{R}_k$ violates fault tolerance precisely if

- (1) fewer than $\text{wt}(e) - k - 1$ faults in $\mathcal{C}'$ are sufficient to cancel $\pi(e)$, and
- (2) that many faults are still allowed to occur, i.e., $\text{wt}(e) - k - 1 \leq t - k$.

This means that fault tolerance is violated if $\pi(e)$ can be canceled by at most

$$h_k(e) := \min[t - k, \text{wt}(e) - k - 1]$$

faults on the ancilla.

We therefore precompute, for each $e \in \mathcal{R}_k$, the set of *bad images*:

$$\text{Bad}(e) = \left\{ e' \in \bigcup_{k'=1}^{h_k(e)} \mathcal{E}_{k'}(\mathcal{C}') \mid \text{wt}(e') = \text{wt}(e) \right\},$$

which are the errors in the ancilla state that could potentially cancel a projected data error under permutation.

Given this set, checking whether a permutation σ is FT' reduces to verifying that

$$\sigma[\pi(e)] \notin \text{Bad}(e) \quad \text{for all } k < t \text{ and all } e \in \mathcal{R}_k.$$

As long as we can store the sets $\mathcal{R}_k$ for each k and $\text{Bad}(e)$ for each error in these sets, checking whether a permutation is fault tolerant can be done efficiently for each error.

Example 4. Consider the preparation of a $w = 6$ qubit cat state shown in Fig. 3. Here, the qubits 0, 2, 3, and 4 have already been selected as control qubits. The fault set (up to stabilizer) of that 6-qubit preparation circuit is

$$\begin{aligned} & \{[0, 0, 0, 0, 0, 1], [0, 0, 0, 0, 1, 0], [0, 0, 0, 1, 0, 0], \\ & [0, 0, 1, 0, 0, 0], [0, 1, 0, 0, 0, 0], [1, 0, 0, 0, 0, 0], \\ & [1, 1, 0, 0, 0, 0], [0, 0, 0, 1, 1, 0], [1, 1, 1, 0, 0, 0]\}. \end{aligned}$$

Projecting this onto the controls gives the projected fault set

$$\{[1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1], [1, 1, 0, 0]\}.$$

The only high-weight fault in this projected set is $[1, 1, 0, 0]$, which is the projection of $[0, 0, 0, 1, 1, 0]$ (drawn in blue in Fig. 3) or $[1, 1, 1, 0, 0, 0]$ (drawn in red in Fig. 3). This means that in this case $\mathcal{R}_1$ is simply $\{[1, 1, 1, 0, 0, 0]\}$. The projection of $[1, 1, 1, 0, 0, 0]$ has weight two, so the bad images are all

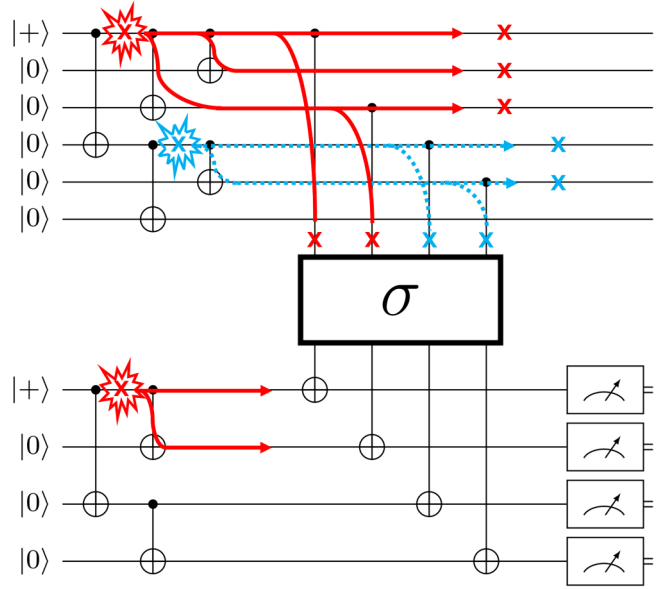


FIG. 3. Fault-tolerance check: precomputing what errors on the ancilla can cancel with the worst-case errors on the data qubit allows for fast checking of fault tolerance based on the permutation σ .

weight-two faults in the single-qubit fault set of the ancilla preparation circuit:

$$\text{Bad}([1, 1, 1, 0, 0, 0]) = \{[1, 1, 0, 0], [0, 0, 1, 1]\}.$$

With this, we can quickly check candidates for the permutation σ that makes the preparation FT². For example, the permutation

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 3 & 2 \end{pmatrix}$$

does not work because σ projects $[1, 1, 1, 0, 0, 0]$ onto $[1, 1, 0, 0] \in \text{Bad}([1, 1, 1, 0, 0, 0])$. On the other hand,

$$\sigma = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 2 & 1 & 3 \end{pmatrix}$$

does work.

With a way to narrow down the search for the choice of controls and an efficient method for checking the fault tolerance of a candidate permutation, we now turn to the task of actually constructing FT' partial transversal CNOTs.

VI. COMBINATORIAL SEARCH METHODS

While we are particularly interested in fault-tolerant cat state preparation circuits using a balanced binary-tree structure, the methods presented in this section apply to arbitrary unitary cat state preparation circuits $\mathcal{C}, \mathcal{C}'$. The only information required is the single-qubit fault sets $\mathcal{E}_1(\mathcal{C})$ and $\mathcal{E}_1(\mathcal{C}')$ of the two cat state preparation circuits, as well as the target fault distance t . Any binary tree on w qubits can be turned into a cat state preparation circuit, and the single-qubit fault set is immediately given by the set of proper subtrees.

While the circuit structure $\mathcal{B}_w$ has the lowest possible depth for a unitary cat state preparation circuit, additional (e.g., hardware) constraints might necessitate other preparation

circuits. Cat states can also be prepared in constant depth using dynamic circuits with midcircuit measurement and feedback [41,42]. Engineered dissipation provides another nonunitary route in which the target GHZ state is obtained as the unique steady state of engineered open-system dynamics rather than via a finite-depth unitary circuit [43,44]. Since the dynamic correction is based on the parity of the measurement bits and since a single measurement might be involved in a linear number of corrections, these circuits are also not directly fault tolerant.

Based on single-qubit fault sets of some unitary cat state preparation circuit, we can attempt to search for fault-tolerant partial CNOTs.

A. Direct search for target permutation using satisfiability modulo theories

For a fixed choice of the w' control positions on the w -qubit data cat state, the task is to determine how these controls are wired to the w' -qubit ancilla cat state. We encode this problem as a satisfiability instance over integer variables and solve it using an SMT solver [45,46].

The wiring is represented by a symbolic permutation $\sigma = (\sigma_1, \dots, \sigma_{w'})$, where each variable σ_i takes values in $[w']$ and specifies the ancilla qubit targeted by the i th control. To ensure that each ancilla qubit is targeted exactly once, the tuple $(\sigma_1, \dots, \sigma_{w'})$ is constrained to be a bijection:

$$\bigwedge_{i \in [w']} \bigwedge_{j \in [w'] \setminus \{i\}} \sigma_i \neq \sigma_j.$$

We now describe how a projected error pattern on the control qubits propagates through such a permutation. Any projected error is represented as a bit string $e \in \{0, 1\}^{w'}$, with support $S(e) = \{i \mid e_i = 1\}$ and weight $|e|$. Given an assignment to the permutation variables σ , the induced error on the ancilla is another bit string $e^\sigma \in \{0, 1\}^{w'}$, whose entries are defined by

$$e_j^\sigma \Leftrightarrow \bigvee_{i \in S(e)} (\sigma_i = j), \quad j = 1, \dots, w'.$$

That is, the j th ancilla qubit is affected if at least one control qubit in the support of e is wired to it. Since σ is a permutation, this mapping preserves Hamming weight:

$$\sum_{j=1}^{w'} e_j^\sigma = |e|.$$

For each error e , let $\text{Bad}(e) \subseteq \{0, 1\}^{w'}$ denote the set of bad images as defined in Sec. VC, each of which has weight $|e|$. For each $e' \in \text{Bad}(e)$, we add the clause

$$\bigvee_{j=1}^{w'} (e'_j \neq e_j^\sigma),$$

which forbids the solver from mapping e to any image in $\text{Bad}(e)$.

The downside of this encoding is that all forbidden propagations must be encoded up front, even if a small subset of these propagated faults is sufficient to constrain the choice of permutation. Furthermore, the encoding is specific to the

choice of control qubits and it might be necessary to solve multiple instances for different choices of control qubits.

B. Simultaneous search for control qubits and target permutation with CEGAR

We now describe a combined exact approach that simultaneously searches over the choice of w' control qubits among the w data qubits and the wiring to the w' ancilla qubits, without encoding all fault propagations up front. Because the resulting search space is enormous, we adopt a CEGAR strategy.

CEGAR [47] is an iterative approach in which one first solves a coarse abstraction of the problem and then checks whether the candidate solution satisfies the full specification. If a counterexample is found, it is used to refine the abstraction by adding constraints that rule out the offending behavior. This *solve-check-refine* loop is repeated until either a valid solution is obtained or unsatisfiability is proved.

In our setting, the initial abstraction captures only the structural constraints:

- (1) exactly w' data qubits are selected as controls, and
- (2) the selected controls are wired bijectively to the w' ancilla qubits.

For each data qubit $q \in [w]$, we introduce a Boolean variable ctrl_q indicating whether q is selected as a control, and an integer variable $\text{trgt}_q \in \{1, \dots, w'\}$ specifying the ancilla qubit targeted by q if $\text{ctrl}_q = 1$. A valid partial transversal CNOT is enforced by the constraints

$$\sum_{q=1}^w \text{ctrl}_q = w',$$

which ensures that exactly w' controls are chosen, and

$$\bigwedge_{u < v} (\text{ctrl}_u = 0 \vee \text{ctrl}_v = 0 \vee \text{trgt}_u \neq \text{trgt}_v),$$

which enforces that no two selected control qubits are connected to the same ancilla target.

Given a candidate assignment $(\text{ctrl}, \text{trgt})$, we construct the corresponding partial transversal CNOT $\overline{\text{CX}}$ and its induced error mapping f , and check whether the resulting construction is FT^t . If no violation is found, the search terminates with an FT^t partial transversal CNOT (Fig. 4).

If the checker identifies a counterexample, that is, a projected error e whose induced image $f(e) \in \text{Bad}(e)$ ¹ under the current assignment $(\text{ctrl}, \text{trgt})$, we refine the abstraction by adding a *blocking clause* that excludes this specific assignment.

Concretely, for each ancilla index $j = 1, \dots, w'$, we introduce the Boolean formula

$$b_j(e) := \bigvee_{\substack{q \in [w] \\ \text{ctrl}_q = 1, e_q = 1}} (\text{trgt}_q = j),$$

¹Since the choice of control qubits is determined jointly with the wiring, the sets $\text{Bad}(e)$ cannot be fully precomputed. In practice, computing and caching them on demand works well, as valid control selections are typically found early.

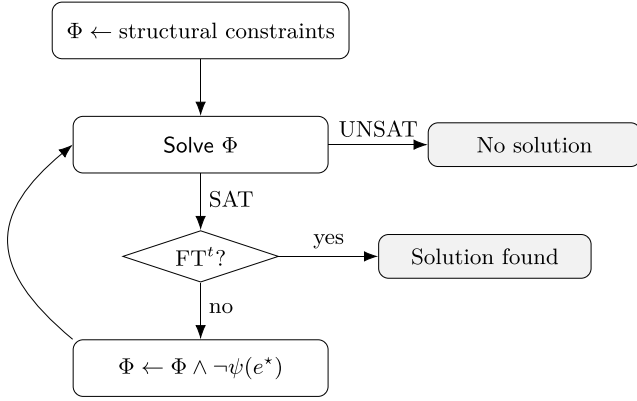


FIG. 4. Constructing FT^t partial transversal CNOTs for cat state preparation using CEGAR. An initial SAT formula Φ is iteratively refined by adding blocking clauses forbidding counterexamples until satisfiability is decided.

which evaluates to true if and only if the projected error has support on ancilla qubit j . To prevent this projection from recurring, we add the clause

$$\bigvee_{j=1}^{w'} (f(e)_j \neq b_j(e)),$$

thereby excluding the current assignment.

This refinement process accumulates blocking clauses as new counterexamples are discovered, progressively shrinking the search space until either an FT^t partial transversal CNOT is found or unsatisfiability is established.

The advantage of this CEGAR formulation is that it avoids encoding all forbidden projections up front, and instead learns only those constraints that are required to eliminate observed counterexamples. In some instances, however, a large number of refinements may be needed before convergence. In such cases, it can be advantageous to fix the choice of control qubits and directly search for a fault-tolerant wiring, as discussed in the previous section.

C. Heuristic local search

In addition to the exact SMT encoding(s), we also employ a heuristic local search strategy that operates directly on the space of ancilla permutations. Remember that for a fixed choice of the w' control positions, the task is to find a permutation of the w' ancilla qubits such that any error e is not projected onto an error in $\text{Bad}(e)$.

The procedure begins with a random permutation of the w' ancilla positions. For a given permutation, we check whether the permutation of any projected error is in the precomputed bad images $\text{Bad}(e)$. If such an e exists, we attempt to adjust the permutation so that e is no longer a violation. This can be done by swapping qubits in the support of e with qubits outside the support. Note that this is not guaranteed to converge, but in practice, it often eliminates the conflicts quickly. If the search becomes stuck while trying to eliminate the violation, occasional random permutations are performed in the hope of breaking the cycle in the search.

Pseudocode for this algorithm is shown in Algorithm 1. This randomized repair process continues until either a valid permutation is found or the maximum number of iterations is reached. While the method does not guarantee completeness—there may exist valid permutations that it fails to discover—it often succeeds in practice for moderate system sizes.

A key challenge of this approach is that repairing a conflict for one error e does not guarantee that previously repaired conflicts remain satisfied. In other words, a swap that eliminates $f(e) \in \text{Bad}(e)$ may introduce a new violation for a different error e' . Our procedure is inspired by the resampling algorithm of Moser and Tardos [48], originally developed as a constructive proof of the Lovász local lemma, which describes the conditions under which a search procedure for such rare combinatorial objects is successful. Although the formal conditions of the lemma do not hold in our setting, the same randomized repair philosophy often succeeds in practice. Repeated swaps and occasional random perturbations typically shuffle the permutation into a globally valid configuration before the iteration budget is exhausted.

D. Comparison of the three search approaches

Table II shows an overview of the benefits and downsides of each of the proposed methods. The CEGAR approach is the most general as it does not presuppose a selection of control qubits for the transversal CNOT. It can certify both the existence and the nonexistence of a fault-tolerant partial transversal CNOT for a given w' , and can often solve SAT instances much faster than the direct SMT encoding. UNSAT instances, on the other hand, are often slow to solve due to the large number of iterations required. The SAT/SMT encoding with fixed controls is narrower in scope, but if an instance is UNSAT, this approach can often prove this very quickly in practice. If the number of different choices of control qubits can be narrowed down, this direct encoding can be very effective for unsatisfiable instances. The local search heuristic is by

ALGORITHM 1. Heuristic search for fault-tolerant partial CNOT.

Input: Fault set $\mathcal{E}$ on w qubits, forbidden images $\text{Bad}(e)$ for all $e \in \mathcal{E}$, maximum iterations N_{iter} , a projection $\pi : \mathbb{F}_2^w \rightarrow \mathbb{F}_2^{w'}$

Output: Permutation σ making the overall CNOT FT^t or FAIL

```

1   $\sigma \leftarrow$  random permutation;
2  for  $k = 1$  to  $N_{\text{iter}}$  do
3    Find  $e \in \mathcal{E}$  with  $\sigma(\pi(e)) \in \text{Bad}(e)$ ;
4    if no such  $e$  then
5      return  $\sigma$ ;
6     $\text{ones} \leftarrow \text{supp}(\pi(e))$ ;
7     $\text{zeros} \leftarrow [w'] \setminus \text{supp}(\pi(e))$ ;
8    while  $\sigma(\pi(e)) \in \text{Bad}(e)$  do
9      randomly pick  $i \in \text{ones}$ ;
10     randomly pick  $j \in \text{zeros}$ ;
11      $\sigma \leftarrow \sigma \circ (i\ j)$ ;
12 return FAIL;
  
```

TABLE II. Comparison of the three search strategies.

Method (scope)	Summary
SAT/SMT <i>Permutation</i>	Pros: Exact; often proves UNSAT quickly; direct encoding of forbidden errors. Cons: Must be repeated for many control sets; typically slowest on SAT instances.
CEGAR <i>Controls + permutation</i>	Pros: Can prove SAT/UNSAT; refinement keeps formulas small. Cons: May require many refinements to prove UNSAT.
Local search <i>Permutation</i>	Pros: Fast; often finds a valid permutation quickly in practice. Cons: Heuristic; no completeness or UNSAT guarantees.

far the fastest and often succeeds quickly, but it cannot certify unsatisfiability and may fail to find a fault-tolerant CNOT even if one exists.

In practice, we combine these methods. The solver first attempts the joint CEGAR search, and only if this times out do we iterate over control selections and apply the SAT/SMT encoding together with the local search. Since the direct SMT encoding is fast for UNSAT instances, we use it to quickly filter out choices of controls that do not work, thereby avoiding the need for heuristic search to iterate over permutations unnecessarily if the instance is UNSAT. This layered strategy balances generality, exactness, and speed: the most powerful method is tried first, while the faster but less reliable methods act as fallbacks when exact approaches run out of time.

VII. EVALUATION

The proposed search algorithms for FT^r transversal CNOTs have been implemented in PYTHON and used to construct fault-tolerant cat state preparation schemes for up to 49 qubits and fault distances of up to 9. In this section, we show the resource costs and circuit-level simulation results of the proposed scheme in comparison with the general recursive cat state preparation construction from [26]. The PYTHON implementation, along with all generated circuits, is available via the open-source MQT QECC repository [49], which is part of the Munich Quantum Toolkit [50].

A. Simulation setup

The circuits were generated using different machines running UBUNTU 22.04 and required at most 128 GB.

The circuits were simulated using the open-source stabilizer simulator STIM [51] with a standard depolarizing noise model [10,15,22] parametrized by a single noise parameter p :²

(1) All 2-qubit gates are followed by 2-qubit depolarizing noise of strength p .

²While only bit-flip noise is relevant for our purposes, we employ a standard noise model often used in circuit-level noise simulations to make our simulations more easily comparable to other settings.

(2) All initializations flip with a probability $2/3 p$.

(3) All measurements flip with a probability $2/3 p$.

For the proposed transversal verification scheme, we simulate the entire circuit and postselect on the all-0 or all-1 Z -basis measurement of the ancilla. If no error is detected on the ancilla, the data is measured out fault-free to obtain a bit string $b \in \{0, 1\}^w$. Let n be the number of shots simulated, n_{acc} the number of accepted shots, and B be the set of measured data bitstrings of the accepted shots. Based on this data, we estimate the acceptance rate R_{acc} and the probability P_k that k bits are flipped on the data as

$$R_{\text{acc}} = \frac{n_{\text{acc}}}{n},$$

$$P_k = \frac{1}{|B|} \sum_{b \in B} \left[\left(\sum_{i \in [w]} b_i \right) = k \right],$$

where $b_i \in \{0, 1\}$ is the i th bit of the bit string b and $[\cdot]$ denotes the *Iverson Bracket*, which interprets Booleans as integers.

The simulation of the recursive cat state construction proceeds in a similar fashion.

B. Discussion

A comparison of the resulting generated circuits in terms of the number of CNOTs and qubits required to fault-tolerantly prepare a data cat state of a certain size is shown in Fig. 5. We observe that, on both accounts, the proposed transversal construction utilizes fewer resources than the recursive construction. For larger cat state sizes, up to 20 CNOT gates and qubits can be saved.

This also translates to an improved acceptance rate for larger cat states and higher fault distances as can be seen in Fig. 6, which shows the acceptance rates for simulation runs at physical error rates of 1% (Fig. 6) and 0.1% [Fig. 6(b)]. While for high physical error rates of 1% the difference in acceptance rates is as high as 8%, for higher-weight cat states, the difference in acceptance rate can still be as high as 2.5% even for physical error rates of 0.1%. The difference is less significant for lower error rates.

In contrast to the acceptance rate, the difference is more pronounced when it comes to errors on the data cat state after postselection. Consider the error rate plots in Fig. 7, which show simulated error rates for FT^3 cat states at error rates $p = 1\%$. Since the circuits are constructed to be only fault tolerant against 3 errors, we expect the probability that we observe 4 errors to be about as high as the probability that we observe 3 errors. This is also what we see in the simulations of the recursive construction. However, the transversal verification actually performs much better than that. We observe that P_4 remains approximately an order of magnitude lower than P_3 . Although P_5 is not a magnitude lower than P_4 for smaller cat states, the scaling is quite favorable compared to the recursive construction. The same phenomenon can be observed for the FT^4 (Fig. 8) cat states, where 5 bitflips are more likely for the recursive construction compared to the transversal construction.

The reason for this is that while the transversal construction is not guaranteed to catch all problematic combinations of 4 errors in the FT^3 case, there are only very few combinations

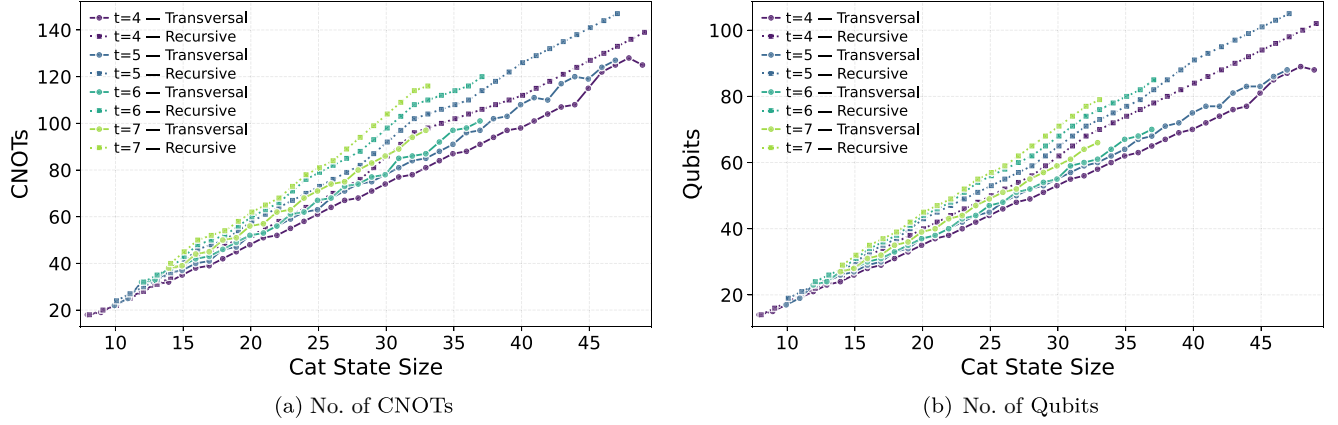


FIG. 5. Comparison of the required number of CNOT gates (a) and qubits (b) for cat state preparation circuits for different cat state sizes and fault distance. “Transversal” corresponds to the proposed construction and “recursive” corresponds to the construction from Ref. [26].

that pass verification undetected. The recursive construction only checks 3 qubits of each state when combining them with ZZ measurements and so might miss errors on the other qubits.

Therefore, even if the overhead savings in acceptance rate, depth, qubits, or CNOTs are not substantial for small fault distances, the transversal construction may still provide higher-quality cat states.

More details about the constructed circuits are listed in Appendix A.

Note that the particulars of the acceptance and error rates are dependent on the utilized error model. A comparatively high measurement error present on some hardware platforms

[28] for example will have a detrimental impact on the acceptance rate as any flipped measurement will result in the state being discarded. More complex noise like crosstalk might also significantly impact the error rate of the produced states. Feasibility of the proposed constructions will have to be evaluated for a given hardware-specific error model to draw definite conclusions.

VIII. CONCLUSION

In this work, we propose a method for the fault-tolerant construction of cat states using two cat states: a transversal CNOT gate and postselection. We demonstrate that fault toler-

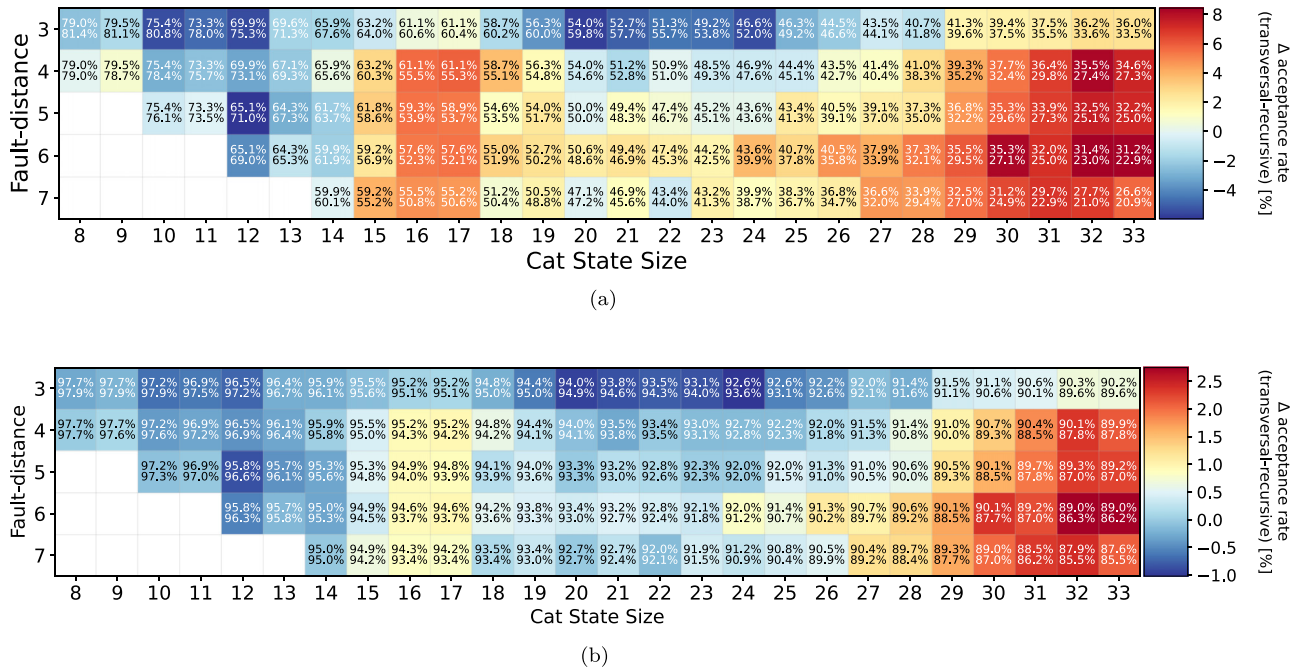


FIG. 6. Comparison of acceptance rates for cat state preparation. “Transversal” corresponds to the proposed construction and “recursive” corresponds to the construction from Ref. [26]. Each cell shows the absolute acceptance rate for the transversal (top) and recursive (bottom) constructions. The hue indicates the difference in acceptance rates of the two constructions. (a) Acceptance rates for simulations using a physical error rate of $p = 1\%$. (b) Acceptance rates for simulations using a physical error rate of $p = 0.1\%$.

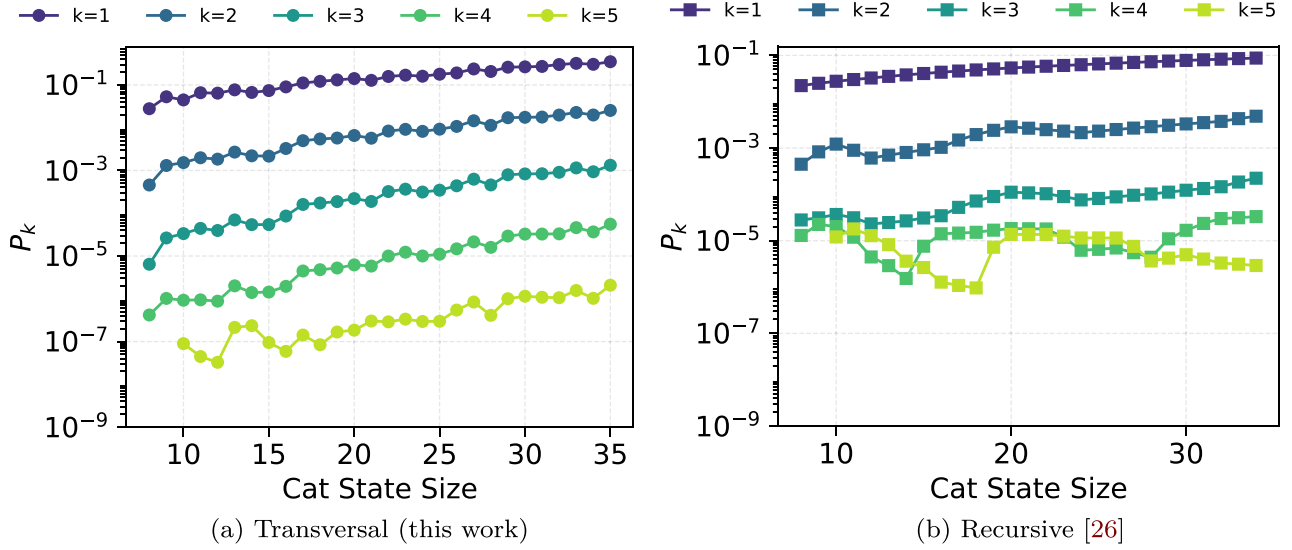


FIG. 7. Error rates for FT^3 cat state preparation at $p = 1\%$. P_k is the probability that k bitflips are measured on the data after postselection. (a) Error rates of circuits constructed using the proposed methods. (b) Error rates of circuits synthesized using the construction from Ref. [26].

ance of this construction depends on the precise connection between the two cat states and the transversal CNOT, and propose three different approaches for finding fault-tolerant CNOTS. We show that for the parameter ranges for which we are able to solve this combinatorial search problem, our construction outperforms the log-depth construction from Ref. [26] in terms of depth, qubit, and CNOT overhead. Furthermore, circuit-level noise simulations demonstrate that our construction exhibits higher acceptance rates and lower error rates compared to Ref. [26].

While the constructed circuits themselves might have applications in certain Shor-style syndrome extraction schemes [2] or as a resource for other QEC schemes [33], the proposed search methods might have applications for finding other fault-tolerant constructions. The CEGAR search in particular might prove useful for other “brute-force” fault-tolerance

constructions as it does not require encoding all constraints up front.

While the approaches of this work and Ref. [26] are very different—the former substituting the formal elegance of the latter with raw compute—they complement each other well as the circuits constructed in this work can be used as a starting point for further fault-tolerance construction in fault-tolerant rewriting. For example, to construct even larger cat states, the optimized circuits of the proposed constructions can be used as part of the recursive construction in Ref. [26].

ACKNOWLEDGMENTS

The authors would like to thank Lucas Berent and Ludwig Schmid for valuable feedback on the manuscript.

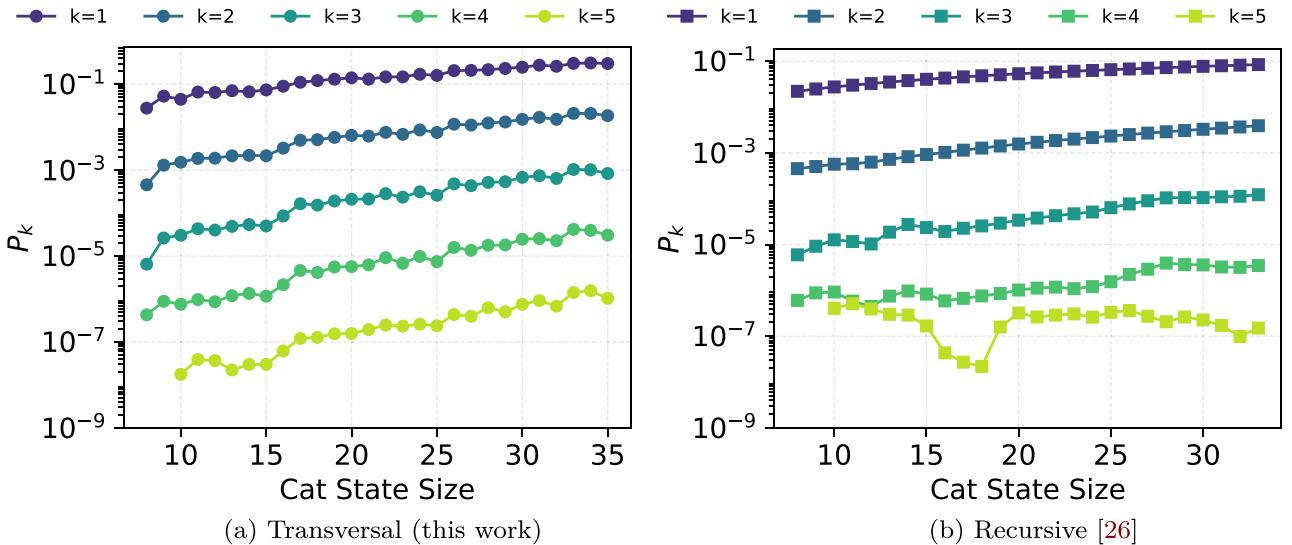


FIG. 8. Error rates for FT^4 cat state preparation at $p = 1\%$. P_k is the probability that k bitflips are measured on the data after postselection. (a) Error rates of circuits constructed using the proposed methods. (b) Error rates of circuits synthesized using the construction from Ref. [26].

The authors acknowledge funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (Grant Agreement No. 101001318) and Millionion (Grant Agreement No. 101114305). This work was part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus. This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, Grant Agreement No. 563402549).

DATA AVAILABILITY

The data that support the findings of this article are openly available [49].

APPENDIX A: FULL LIST OF CIRCUIT CONSTRUCTIONS

In the following, we list acceptance rates (R_A), depth (d), number of CNOT gates (CX), and number of qubits (Q) for all cat state preparation circuits constructed using the proposed methods as well as circuits constructed using the recursive construction from [26]. The Δ columns list the difference between the circuits obtained using the proposed methods and Ref. [26]. The results are listed in Tables III, IV, V, VI, VII, VIII, and IX.

TABLE III. Comparison of FT³ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
8	97.66	5	18	14	97.94	7	16	13	-0.29	-2	2	1
9	97.72	6	19	15	97.91	7	18	15	-0.18	-1	1	0
10	97.20	6	22	17	97.87	7	20	17	-0.66	-1	2	0
11	96.93	6	25	19	97.53	7	23	19	-0.60	-1	2	0
12	96.47	6	28	21	97.19	7	26	21	-0.72	-1	2	0
13	96.41	6	29	22	96.66	8	29	23	-0.24	-2	0	-1
14	95.89	6	32	24	96.13	8	32	25	-0.24	-2	0	-1
15	95.49	6	35	26	95.61	8	35	27	-0.12	-2	0	-1
16	95.16	6	38	28	95.09	8	38	29	0.08	-2	0	-1
17	95.16	7	39	29	95.05	8	40	31	0.11	-1	-1	-2
18	94.78	7	42	31	95.01	8	42	33	-0.23	-1	0	-2
19	94.39	7	45	33	94.97	8	44	35	-0.58	-1	1	-2
20	94.00	7	48	35	94.94	8	46	37	-0.93	-1	2	-2
21	93.77	7	49	36	94.61	8	49	39	-0.84	-1	0	-3
22	93.50	7	52	38	94.28	8	52	41	-0.78	-1	0	-3
23	93.12	7	55	40	93.96	8	55	43	-0.84	-1	0	-3
24	92.61	7	58	42	93.63	8	58	45	-1.02	-1	0	-3
25	92.56	7	59	43	93.12	9	61	47	-0.56	-2	-2	-4
26	92.18	7	62	45	92.61	9	64	49	-0.43	-2	-2	-4
27	91.98	7	65	47	92.11	9	67	51	-0.12	-2	-2	-4
28	91.36	7	68	49	91.60	9	70	53	-0.24	-2	-2	-4
29	91.49	7	69	50	91.11	9	73	55	0.39	-2	-4	-5
30	91.06	7	72	52	90.61	9	76	57	0.45	-2	-4	-5
31	90.62	7	75	54	90.11	9	79	59	0.51	-2	-4	-5
32	90.31	7	78	56	89.63	9	82	61	0.68	-2	-4	-5
33	90.25	8	79	57	89.59	9	84	63	0.66	-1	-5	-6
34	89.71	8	82	59	89.55	9	86	65	0.15	-1	-4	-6

TABLE IV. Comparison of FT⁴ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
8	97.66	5	18	14	97.66	8	18	14	-0.00	-3	0	0
9	97.72	6	19	15	97.62	8	20	16	0.10	-2	-1	-1
10	97.20	6	22	17	97.58	8	22	18	-0.38	-2	0	-1
11	96.93	6	25	19	97.24	8	25	20	-0.31	-2	0	-1
12	96.47	6	28	21	96.90	8	28	22	-0.44	-2	0	-1
13	96.07	6	31	23	96.38	8	31	24	-0.31	-2	0	-1
14	95.89	6	32	24	95.85	8	34	26	0.04	-2	-2	-2
15	95.49	6	35	26	95.05	9	39	29	0.45	-3	-4	-3
16	95.16	6	38	28	94.26	9	44	32	0.91	-3	-6	-4
17	95.16	7	39	29	94.22	9	46	34	0.95	-2	-7	-5
18	94.78	7	42	31	94.18	9	48	36	0.61	-2	-6	-5
19	94.39	7	45	33	94.14	9	50	38	0.25	-2	-5	-5
20	94.00	7	48	35	94.10	9	52	40	-0.10	-2	-4	-5
21	93.49	7	51	37	93.78	9	55	42	-0.29	-2	-4	-5
22	93.44	7	52	38	93.45	9	58	44	-0.01	-2	-6	-6
23	93.00	7	55	40	93.13	9	61	46	-0.14	-2	-6	-6
24	92.67	7	58	42	92.81	9	64	48	-0.14	-2	-6	-6
25	92.17	7	61	44	92.30	9	67	50	-0.13	-2	-6	-6
26	91.97	7	64	46	91.80	9	70	52	0.17	-2	-6	-6
27	91.53	7	67	48	91.30	9	73	54	0.23	-2	-6	-6
28	91.42	7	68	49	90.80	9	76	56	0.62	-2	-8	-7
29	91.04	7	71	51	90.04	10	81	59	1.00	-3	-10	-8
30	90.67	7	74	53	89.29	10	86	62	1.38	-3	-12	-9
31	90.35	7	77	55	88.54	10	91	65	1.81	-3	-14	-10
32	90.13	7	78	56	87.80	10	96	68	2.32	-3	-18	-12
33	89.88	8	81	58	87.77	10	98	70	2.11	-2	-17	-12

APPENDIX B: A GENERAL FT⁴ TRANSVERSAL CAT STATE CONSTRUCTION

The combinatorial search approach from Sec. V applies to arbitrary cat state preparation circuits. Its scalability, however, is limited by the size of the fault sets induced by the chosen preparation circuit. For highly structured circuits, one can instead hope for explicit constructions that work for arbitrary cat state sizes. In this section, we give such a construction for FT⁴ cat state verification when the cat state size is a power of 2.

We begin with the simpler FT³ case. Let $w = 2^n$, and let the data cat state be prepared using the balanced binary-tree circuit $\mathcal{B}_w$. As discussed in Sec. VA, any propagated single fault in $\mathcal{B}_w$ has support equal to the leaf set of a rooted subtree. In particular, every propagated fault has weight 2^k for some k , depending on the level at which the fault occurs.

To obtain an FT³ partial transversal CNOT to an ancilla cat state, every propagated weight-four fault in the data must intersect the set of control qubits in at least 3 positions. Indeed, if only 2 qubits of such a weight-four subtree were connected to the ancilla, then two independent single faults could already cancel on the ancilla side while leaving a nontrivial residual error on the data. It follows that from each size-four subtree of the data cat state, at least 3 qubits must be chosen as controls. Since there are 2^{n-2} such subtrees, the ancilla must therefore

TABLE V. Comparison of FT⁵ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
10	97.20	6	22	17	97.29	8	24	19	-0.09	-2	-2	-2
11	96.93	6	25	19	96.96	8	27	21	-0.03	-2	-2	-2
12	95.77	6	32	23	96.62	8	30	23	-0.85	-2	2	0
13	95.66	6	33	24	96.09	9	33	25	-0.44	-3	0	-1
14	95.27	6	36	26	95.57	9	36	27	-0.31	-3	0	-1
15	95.28	6	37	27	94.77	9	41	30	0.51	-3	-4	-3
16	94.88	6	40	29	93.98	9	46	33	0.91	-3	-6	-4
17	94.82	7	41	30	93.94	9	48	35	0.88	-2	-7	-5
18	94.10	7	46	33	93.90	9	50	37	0.20	-2	-4	-4
19	93.99	7	47	34	93.59	9	54	40	0.40	-2	-7	-6
20	93.27	7	52	37	93.28	9	58	43	-0.01	-2	-6	-6
21	93.16	7	53	38	92.96	9	61	45	0.20	-2	-8	-7
22	92.77	7	56	40	92.64	9	64	47	0.14	-2	-8	-7
23	92.33	7	59	42	92.31	9	67	49	0.01	-2	-8	-7
24	92.01	7	62	44	92.00	9	70	51	0.01	-2	-8	-7
25	91.96	7	63	45	91.49	10	73	53	0.47	-3	-10	-8
26	91.31	7	68	48	91.00	10	76	55	0.31	-3	-8	-7
27	91.00	7	71	50	90.50	10	79	57	0.50	-3	-8	-7
28	90.56	7	74	52	90.01	10	82	59	0.56	-3	-8	-7
29	90.45	7	75	53	89.25	10	87	62	1.20	-3	-12	-9
30	90.08	7	78	55	88.51	10	92	65	1.57	-3	-14	-10
31	89.70	7	81	57	87.77	10	97	68	1.94	-3	-16	-11
32	89.34	7	84	59	87.03	10	102	71	2.30	-3	-18	-12
33	89.23	8	85	60	87.00	10	104	73	2.23	-2	-19	-13

TABLE VI. Comparison of FT⁶ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
12	95.78	6	32	23	96.34	8	32	24	-0.56	-2	0	-1
13	95.66	6	33	24	95.81	9	35	26	-0.15	-3	-2	-2
14	94.99	6	38	27	95.29	9	38	28	-0.30	-3	0	-1
15	94.87	6	39	28	94.49	9	43	31	0.38	-3	-4	-3
16	94.61	6	42	30	93.70	9	48	34	0.90	-3	-6	-4
17	94.61	7	43	31	93.67	9	50	36	0.94	-2	-7	-5
18	94.17	7	46	33	93.63	9	52	38	0.54	-2	-6	-5
19	93.78	7	49	35	93.32	9	56	41	0.46	-2	-7	-6
20	93.39	7	52	37	93.01	9	60	44	0.39	-2	-8	-7
21	93.16	7	53	38	92.69	9	63	46	0.47	-2	-10	-8
22	92.77	7	56	40	92.36	9	66	48	0.41	-2	-10	-8
23	92.12	7	61	43	91.77	9	71	51	0.35	-2	-10	-8
24	92.01	7	62	44	91.19	9	76	54	0.82	-2	-14	-10
25	91.36	7	67	47	90.69	10	79	56	0.67	-3	-12	-9
26	91.31	7	68	48	90.20	10	82	58	1.11	-3	-14	-10
27	90.73	7	73	51	89.71	10	85	60	1.02	-3	-12	-9
28	90.56	7	74	52	89.22	10	88	62	1.34	-3	-14	-10
29	90.13	7	77	54	88.47	10	93	65	1.65	-3	-16	-11
30	90.08	7	78	55	87.73	10	98	68	2.35	-3	-20	-13
31	89.18	7	85	59	87.00	10	103	71	2.19	-3	-18	-12
32	89.02	7	86	60	86.27	10	108	74	2.75	-3	-22	-14
33	88.97	8	87	61	86.24	10	110	76	2.73	-2	-23	-15

TABLE VII. Comparison of FT⁷ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
14	94.99	6	38	27	95.01	9	40	29	-0.02	-3	-2	-2
15	94.87	6	39	28	94.22	9	45	32	0.66	-3	-6	-4
16	94.27	6	44	31	93.43	9	50	35	0.84	-3	-6	-4
17	94.20	7	45	32	93.39	9	52	37	0.81	-2	-7	-5
18	93.49	7	50	35	93.35	9	54	39	0.14	-2	-4	-4
19	93.38	7	51	36	93.05	9	58	42	0.33	-2	-7	-6
20	92.72	7	56	39	92.74	9	62	45	-0.01	-2	-6	-6
21	92.67	7	57	40	92.41	9	65	47	0.26	-2	-8	-7
22	91.96	7	62	43	92.09	9	68	49	-0.14	-2	-6	-6
23	91.91	7	63	44	91.51	9	73	52	0.40	-2	-10	-8
24	91.20	7	68	47	90.92	9	78	55	0.28	-2	-10	-8
25	90.82	7	71	49	90.43	10	81	57	0.40	-3	-10	-8
26	90.45	7	74	51	89.93	10	84	59	0.52	-3	-10	-8
27	90.40	7	75	52	89.18	10	89	62	1.22	-3	-14	-10
28	89.71	7	80	55	88.44	10	94	65	1.27	-3	-14	-10
29	89.34	7	83	57	87.70	10	99	68	1.64	-3	-16	-11
30	88.97	7	86	59	86.96	10	104	71	2.01	-3	-18	-12
31	88.54	7	89	61	86.23	10	109	74	2.31	-3	-20	-13
32	87.92	7	94	64	85.51	10	114	77	2.40	-3	-20	-13
33	87.56	8	97	66	85.48	10	116	79	2.08	-2	-19	-13

have size

$$w' = 3 \times 2^{n-2}.$$

We now make this more explicit. For $k \in \{0, \dots, n\}$ and $i \in \{0, \dots, 2^k - 1\}$, let

$$S_{k,i} = \{i \times 2^{n-k}, \dots, (i+1) \times 2^{n-k} - 1\}$$

denote the set of leaves in the i th level- k subtree of $\mathcal{B}_w$. Thus the level- $(n-2)$ subtrees are precisely the size-four blocks

$$\{0, 1, 2, 3\}, \{4, 5, 6, 7\}, \dots, \{w-4, w-3, w-2, w-1\}.$$

From each such block we choose the first three qubits as controls. This yields the control set

$$C = \{4j, 4j+1, 4j+2 : j = 0, \dots, 2^{n-2} - 1\},$$

which has cardinality $|C| = 3 \times 2^{n-2} = w'$.

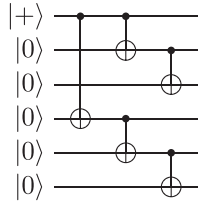
TABLE VIII. Comparison of FT⁸ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
16	94.27	6	44	31	93.16	10	52	36	1.11	-4	-8	-5
17	94.20	7	45	32	93.12	9	54	38	1.09	-2	-9	-6
18	93.49	7	50	35	93.08	9	56	40	0.41	-2	-6	-5
19	93.04	7	53	37	92.77	9	60	43	0.27	-2	-7	-6
20	92.39	7	58	40	92.46	9	64	46	-0.07	-2	-6	-6
24	90.87	7	70	48	90.66	9	80	56	0.22	-2	-10	-8
25	90.50	7	73	50	90.16	10	83	58	0.34	-3	-10	-8

TABLE IX. Comparison of FT⁹ cat state preparation protocols at physical error rate $p = 0.1\%$.

w	This work				Recursive [26]				Δ			
	R_A	d	CX	Q	R_A	d	CX	Q	R_A	d	CX	Q
18	93.49	7	50	35	92.81	10	58	41	0.68	-3	-8	-6
19	93.05	7	53	37	92.50	10	62	44	0.55	-3	-9	-7

The remaining task is to choose an ancilla cat state preparation circuit and a permutation mapping the control qubits in C to the ancilla qubits. For the ancilla, we use a tree structure $\mathcal{T}_{w'}$ that is analogous to $\mathcal{B}_w$, except that at the lowest level the leaves are grouped into triples rather than pairs. Equivalently, $\mathcal{T}_{w'}$ is obtained by first forming binary trees on each triple of leaves and then recursively combining these triple blocks in a balanced binary tree. For example, $\mathcal{T}_6$ is given by



We now construct the desired permutation in two steps.

Step 1: Local scrambling. Within each pair of adjacent control triples, we swap the first qubit of the first triple with the first qubit of the second triple. Writing the control qubits in the order induced by C , this gives the permutation

$$\sigma_1 = \prod_{k=0}^{2^{n-3}-1} (6k, 6k+3).$$

The role of σ_1 is to ensure that the three control qubits originating from a single size-four subtree of the data are not mapped into a single size-three subtree of the ancilla. Otherwise, a propagated weight-four data error and a propagated weight-three ancilla error could already cancel in a way that violates FT³.

A natural but incorrect idea is to apply this same local scrambling recursively at higher levels of the tree. Figure 9 illustrates why this fails for $w = 32$: although no highlighted subtree on the left is mapped directly onto a single subtree on the right, a size-16 subtree of the data is mapped into two size-6 subtrees of the ancilla. Consequently, one can find two weight-6 errors on the ancilla side that cancels the propagated error while leaving a residual weight-16 error on the data. Thus this recursive version of σ_1 is not FT³.

Step 2: Global distribution of triples. The previous example suggests the correct design principle: the image of every data subtree should be distributed across the ancilla tree as evenly as possible. After applying σ_1 , we therefore redistribute the control triples so that triples from the first half of the data are mapped to the even-numbered triple blocks of the ancilla, while triples from the second half are mapped to the odd-numbered triple blocks.

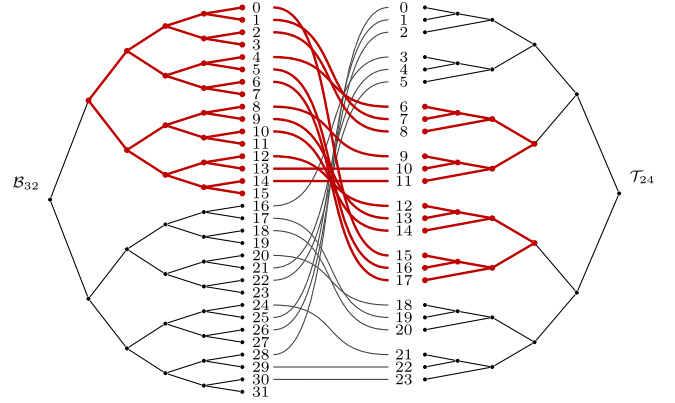


FIG. 9. Non-FT³ partial transversal CNOT. A size-16 subtree is mapped to two different size-6 subtrees giving an error configuration that would cancel on the CNOT.

Formally, let the control indices be numbered $0, \dots, w' - 1$. We define

$$\sigma_2(i) = \begin{cases} i + 3 \left\lfloor \frac{i}{3} \right\rfloor, & i < 3 \times 2^{n-3} \\ i - 3 \times 2^{n-3} + 3 \left(\left\lfloor \frac{i - 3 \times 2^{n-3}}{3} \right\rfloor + 1 \right), & i \geq 3 \times 2^{n-3}. \end{cases}$$

This maps the first half of the control triples to triple blocks

$$0, 2, 4, \dots, 2^{n-2} - 2$$

of the ancilla and the second half to triple blocks

$$1, 3, 5, \dots, 2^{n-2} - 1.$$

For $w = 32$, this permutation is shown in Fig. 10(a).

The permutation σ_2 eliminates the higher-level clustering problem described above: the three control qubits from any given data size-four subtree are sent into ancilla triple blocks that belong to different larger subtrees. After applying σ_2 , the only remaining problematic cancellations are the obvious local ones, namely, those where a data triple is mapped directly to a single ancilla triple block. These are precisely the cancellations removed by σ_1 .

We therefore define the final permutation as

$$\sigma = \sigma_1 \circ \sigma_2.$$

For $w = 32$, the combined permutation is shown in Fig. 10(b).

σ actually also already ensures that no combination of four subtrees cancel out, which makes this construction FT⁴ as well.

The above construction is already sufficient for FT⁴. Intuitively, the distribution enforced by σ_2 prevents large subtrees of the data from concentrating inside too few subtrees of the ancilla, while σ_1 rules out the remaining local weight-three cancellations. As a result, no combination of four propagated subtree faults can cancel on the ancilla side while leaving an uncorrectable residual error on the data. Hence the partial transversal CNOT defined by σ is FT⁴.

For cat state sizes that are not powers of 2, one may obtain a fault-tolerant construction by truncating the corresponding trees, i.e., by removing leaves from $\mathcal{B}_w$ and $\mathcal{T}_{w'}$ to obtain the required sizes.

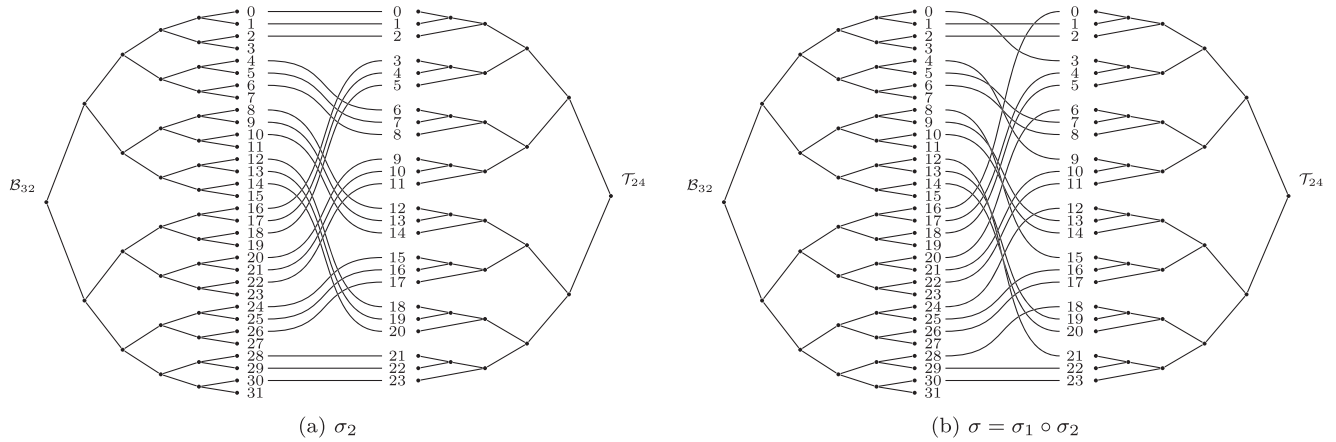


FIG. 10. The permutations σ_2 and σ for $w = 32$. (a) The permutation σ_2 for the $w = 32$ case distributes the three groups from the first half of the data cat state to the even-numbered three groups of the ancilla and those from the second half to the odd-numbered three groups. (b) Concatenating σ_1 with σ_2 results in the permutation on the right, which is FT^4 .

Finally, note that the circuits produced by the combinatorial search of Sec. V do not in general use ancilla states prepared with the $\mathcal{T}_{w'}$ structure. Instead, the search finds the smallest ancilla size w' for which there exists a partial transversal CNOT to an ancilla prepared using the bal-

anced binary-tree circuit $\mathcal{B}_{w'}$. This is why the search-based constructions can sometimes require fewer ancilla qubits than the explicit construction given here. For $w = 2^n$, however, the circuits found by the search match the qubit count of the present construction.

- [1] D. Gottesman, Theory of fault-tolerant quantum computation, *Phys. Rev. A* **57**, 127 (1998).
- [2] P. Shor, Fault-tolerant quantum computation, in *Proceedings of the 37th Conference on Foundations of Computer Science, Burlington, VT* (IEEE, Piscataway, NJ, 1996), pp. 56–65.
- [3] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* **43**, 4452 (2002).
- [4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).
- [5] C. Gidney, How to factor 2048 bit RSA integers with less than a million noisy qubits, [arXiv:2505.15917](#).
- [6] Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, *Nature (London)* **638**, 920 (2025).
- [7] I. Pogorelov, F. Butt, L. Postler, C. D. Marciniak, P. Schindler, M. Müller, and T. Monz, Experimental fault-tolerant code switching, [arXiv:2403.13732](#).
- [8] B. W. Reichardt *et al.*, Fault-tolerant quantum computation with a neutral atom processor, [arXiv:2411.11822](#).
- [9] D. Gottesman, An introduction to quantum error correction and fault-tolerant quantum computation, [arXiv:0904.2557](#).
- [10] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, *Quantum* **2**, 53 (2018).
- [11] P. Prabhu and B. W. Reichardt, Fault-tolerant syndrome extraction and cat state preparation with fewer qubits, *Quantum* **7**, 1154 (2023).
- [12] P.-H. Liou and C.-Y. Lai, Reducing quantum error correction overhead with versatile flag-sharing syndrome extraction circuits, *IEEE Trans. Quantum Eng.* **6**, 1 (2025).
- [13] P.-J. H. S. Derks, A. Townsend-Teague, A. G. Burchards, and J. Eisert, Designing fault-tolerant circuits using detector error models, *Quantum* **9**, 1905 (2024).
- [14] R. Zen, J. Olle, L. Colmenarez, M. Puviani, M. Müller, and F. Marquardt, Quantum circuit discovery for fault-tolerant logical state preparation with reinforcement learning, *Phys. Rev. X* **15**, 041012 (2025).
- [15] T. Peham, L. Schmid, L. Berent, M. Müller, and R. Wille, Automated synthesis of fault-tolerant state preparation circuits for quantum error-correction codes, *PRX Quantum* **6**, 020330 (2025).
- [16] D. Forlivesi and D. Amaro, Flag at origin: A modular fault-tolerant preparation for CSS codes, [arXiv:2508.14200](#).
- [17] C. Chamberland and K. Noh, Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits, *npj Quantum Inf.* **6**, 91 (2020).
- [18] C. Gidney, N. Shutty, and C. Jones, Magic state cultivation: Growing T states as cheap as CNOT gates, [arXiv:2409.17595](#).
- [19] K. Sahay, P.-K. Tsai, K. Chang, Q. Su, T. B. Smith, S. Singh, and S. Puri, Fold-transversal surface code cultivation, [arXiv:2509.05212](#).
- [20] F. Butt, S. Heußen, M. Rispler, and M. Müller, Fault-tolerant code-switching protocols for near-term quantum processors, *PRX Quantum* **5**, 020345 (2024).
- [21] S. Heußen and J. Hilder, Efficient fault-tolerant code switching via one-way transversal CNOT gates, *Quantum* **9**, 1846 (2025).
- [22] S. Heußen, D. F. Locher, and M. Müller, Measurement-free fault-tolerant quantum error correction in near-term devices, *PRX Quantum* **5**, 010333 (2024).
- [23] H. Goto, Y. Ho, and T. Kanao, Measurement-free fault-tolerant logical-zero-state encoding of the distance-three nine-qubit surface code in a one-dimensional qubit array, *Phys. Rev. Res.* **5**, 043137 (2023).
- [24] L. Schmid, T. Peham, L. Berent, M. Müller, and R. Wille, Deterministic fault-tolerant state preparation for near-

- term quantum error correction: Automatic synthesis using Boolean satisfiability, in *Proceedings of the 2025 Design, Automation & Test in Europe Conference (DATE), Lyon, France* (IEEE, Piscataway, NJ, 2025), pp. 1–7.
- [25] N. Shutty and C. Chamberland, Decoding merged color-surface codes and finding fault-tolerant Clifford circuits using solvers for satisfiability modulo theories, *Phys. Rev. Appl.* **18**, 014072 (2022).
- [26] B. Rodatz, B. Poór, and A. Kissinger, Fault tolerance by construction, [arXiv:2506.17181](https://arxiv.org/abs/2506.17181).
- [27] B. Rodatz, B. Poór, and A. Kissinger, Floquetifying stabiliser codes with distance-preserving rewrites, [arXiv:2410.17240](https://arxiv.org/abs/2410.17240).
- [28] D. Bluvstein *et al.*, Logical quantum processor based on reconfigurable atom arrays, *Nature (London)* **626**, 58 (2024).
- [29] L. Postler, F. Butt, I. Pogorelov, C. D. Marciniak, S. Heußen, R. Blatt, P. Schindler, M. Rispler, M. Müller, and T. Monz, Demonstration of fault-tolerant Steane quantum error correction, *PRX Quantum* **5**, 030326 (2024).
- [30] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, Demonstration of fault-tolerant universal quantum gate operations, *Nature (London)* **605**, 675 (2022).
- [31] D. M. Greenberger, M. A. Horne, and A. Zeilinger, Going beyond Bell's theorem, in *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, edited by M. Kafatos, *Fundamental Theories of Physics* Vol. 37 (Springer, Dordrecht, 1989).
- [32] T. Tansuwannont, B. Pato, and K. R. Brown, Adaptive syndrome measurements for Shor-style error correction, *Quantum* **7**, 1075 (2023).
- [33] G. Escobar-Arrieta and M. Gutiérrez, Improved performance of the Bacon-Shor code with Steane's syndrome extraction method, *Phys. Rev. A* **111**, 032427 (2025).
- [34] A. K. Sharma and S. S. Garani, Fault-tolerant quantum LDPC encoders, [arXiv:2405.07242](https://arxiv.org/abs/2405.07242).
- [35] A. Kissinger, Phase-free ZX diagrams are CSS codes (...or how to graphically grok the surface code), [arXiv:2204.14038](https://arxiv.org/abs/2204.14038).
- [36] B. Coecke and A. Kissinger, Picturing quantum processes, in *Diagrammatic Representation and Inference*, edited by P. Chapman, G. Stapleton, A. Moktefi, S. Perez-Kriz, and F. Bellucci (Springer Nature, Berlin, 2018).
- [37] J. van de Wetering, ZX-calculus for the working quantum computer scientist, [arXiv:2012.13966](https://arxiv.org/abs/2012.13966).
- [38] A. Paetznick and B. W. Reichardt, Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit Golay code, [arXiv:1106.2190](https://arxiv.org/abs/1106.2190).
- [39] A. Gong and J. M. Renes, Computation with quantum Reed-Muller codes and their mapping onto 2D atom arrays, [arXiv:2410.23263](https://arxiv.org/abs/2410.23263).
- [40] R. Takagi, T. J. Yoder, and I. L. Chuang, Error rates and resource overheads of encoded three-qubit gates, *Phys. Rev. A* **96**, 042302 (2017).
- [41] E. Bäumer, V. Tripathi, D. S. Wang, P. Rall, E. H. Chen, S. Majumder, A. Seif, and Z. K. Mineev, Efficient long-range entanglement using dynamic circuits, *PRX Quantum* **5**, 030339 (2024).
- [42] E. Bäumer and S. Woerner, Measurement-based long-range entangling gates in constant depth, *Phys. Rev. Res.* **7**, 023120 (2025).
- [43] F. Reiter, D. Reeb, and A. S. Sørensen, Scalable dissipative preparation of many-body entanglement, *Phys. Rev. Lett.* **117**, 040501 (2016).
- [44] X. Q. Shao, J. H. Wu, X. X. Yi, and G.-L. Long, Dissipative preparation of steady Greenberger-Horne-Zeilinger states for Rydberg atoms with quantum Zeno dynamics, *Phys. Rev. A* **96**, 062315 (2017).
- [45] A. Biere, M. Heule, H. van Maaren, and T. Walsh, *Handbook of Satisfiability* (IOS Press, Amsterdam, 2009).
- [46] L. de Moura and N. Björner, Z3: An efficient SMT solver, in *Tools and Algorithms for the Construction and Analysis of Systems* (Springer, New York, 2008), pp. 337–340.
- [47] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, Counterexample-guided abstraction refinement for symbolic model checking, *J. ACM* **50**, 752 (2003).
- [48] R. A. Moser and G. Tardos, A constructive proof of the general Lovász local lemma, *J. ACM* **57**, 1 (2010).
- [49] L. Berent, T. Peham, L. Burgholzer, L. Schmid, P.-J. H. S. Derks, T. Grurl, and C. Pichler, MQT QECC, GitHub, <https://github.com/munich-quantum-toolkit/qecc/> (2025).
- [50] R. Wille, L. Berent, T. Forster, J. Kunasaikaran, K. Mato, T. Peham, N. Quetschlich, D. Rovara, A. Sander, L. Schmid, D. Schönberger, Y. Stade, and L. Burgholzer, The MQT handbook: A summary of design automation tools and software for quantum computing, in *Proceedings of the 2024 IEEE International Conference on Quantum Software (QSW)* (IEEE, Piscataway, NJ, 2024).
- [51] C. Gidney, STIM: A fast stabilizer circuit simulator, *Quantum* **5**, 497 (2021).