Post-selection-free preparation of high-quality physical qubits

Ben Barber, Neil I. Gillespie, and J. M. Taylor

Riverlane, Cambridge, UK

Rapidly improving gate fidelities for coherent operations mean that errors in state preparation and measurement (SPAM) may become a dominant source of error for fault-tolerant operation of quantum computers. This is particularly acute in superconducting systems, where tradeoffs in measurement fidelity and qubit lifetimes have limited overall performance. Fortunately, the essentially classical nature of preparation and measurement enables a wide variety of techniques for improving quality using auxiliary qubits combined with classical control and post-selection. In practice, however, post-selection greatly complicates the scheduling of processes such as syndrome extraction. Here we present a family of quantum circuits that prepare high-quality $|0\rangle$ states without post-selection, instead using CNOT and Toffoli gates to non-linearly permute the computational basis. We find meaningful performance enhancements when two-qubit gate fidelities errors go below 0.2%, and even better performance when native Toffoli gates are available.

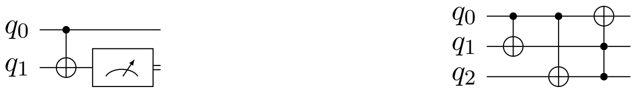
Key physical implementations of quantum computers now reliably achieve two-qubit gate fidelities approaching 99.8% [23], with multiple systems reporting fidelities ranging from 99.2% to 99.6% [1, 10, 19, 26]. This leaves SPAM (state preparation and measurement) errors as a dominant source of error in many, particularly superconducting, machines. Google's Sycamore processor experienced measurement error rates greater than 3% [1], and IBM report similar levels of measurement error [16] in their devices. Recent improvements to standard superconducting measurement error rates have not yet reduced them below 1% [20], although more is possible using multilevel qubit encodings [8]. In contrast, for ions, neutral atoms, and spins in silicon measurement error rates can be much lower (cf. IonQ's 99.3% [27] and HRL's 99.75% [2]), though achieving low rates in dense arrays remain a substantial challenge [28].

There are three broad approaches to reducing the impact of SPAM errors. The first is to improve the physical processes of preparation and measurement. In most architectures there are tradeoffs between SPAM errors and speed, as longer integration or preparation times can improve performance but take longer; see for example [8] for the benefits of measuring slowly up to T_1 limits. The second is error mitigation techniques which calibrate

Ben Barber: ben.barber@riverlane.com

Neil I. Gillespie: neil.gillespie@riverlane.com

J. M. Taylor: jake.taylor@riverlane.com, current address Joint Center for Quantum Information and Computer Science



(a) post-selection

(b) A $(3, 1, 1)$ post-selection-free purification circuit.

Figure 1: Two approaches to $|0\rangle$ state production.

and compensate for errors by running the circuit many times and reconstructing the expected outputs post-facto. This is helpful for NISQ applications [13, 25] but does little to improve the entropy extraction critical for quantum error correction, and the extension of error mitigation into the early fault-tolerant regime [24] requires SPAM errors with similar performance to two-qubit gate fidelities. Here we examine the third approach: algorithmically improving the quality of preparation and measurement using quantum logic gates.

Algorithmic improvement for measurement is straightforward. The simplest example is used in, e.g., the double species ion clock (see [15, 17], and [5] for a recent example), where one readout qubit is used multiple times to estimate the clock qubit—a simple type of repetition code. More generally, CNOT gates can encode a set of measurement outcomes in a classical error correcting code on auxiliary measurement qubits, allowing recovery from sufficiently few measurement errors [14]. As these codes have very good performance and efficient decoders, we recommend their use but do not elaborate on them further here.

A conventional approach to improving preparation using post-selection is the circuit in Figure 1a. We prepare two qubits in state $|0\rangle$, apply a CNOT from the first to the second, then measure the second in the computational basis. If the measurement result is 0, we output the first qubit as a high-quality $|0\rangle$. If not, we reject and begin the process again.

Suppose for now that gates are perfect, but that each preparation incorrectly produces $|1\rangle$ with probability p_0, and that the measurement outcome is also incorrect with probability p_0, with all errors independent. Then the measurement result is 0 precisely when there are an even number of errors across the two preparations and one measurement. We will correctly output $|0\rangle$ if there are no errors, or two errors which occur on the second preparation and the measurement. Otherwise we will either output $|1\rangle$ or reject. Conditional on acceptance, this circuit produces post-selected $|0\rangle$ states with an error rate $2p_0^2 + O(p_0^3)$.

Post-selection is an acceptable method for production of complex logical states such as T-magic states, where the scheduling difficulties it presents are the price we pay for implementing a logical non-Clifford gate fault-tolerantly. However, it is much less satisfactory for the large number of simple physical $|0\rangle$ states required for every round of syndrome extraction. Specific challenges induced by post-selection include

- the need to schedule multiple attempts in case the first attempt fails;
- the need to wait for measurement results and classical processing, which in some systems can take appreciable time;
- the risk of measurement inducing cross-talk errors on adjacent qubits;
- limited advantage when measurement errors are substantially worse than preparation errors.

	result	
total errors	post-selection	purification
0	$\vert 0\rangle$	$\vert 0\rangle$
1	reject	$\vert 0\rangle$
2	$\begin{cases} \vert 0\rangle & \text{with probability } 1/3 \\ \vert 1\rangle & \text{with probability } 2/3 \end{cases}$	$\vert 1\rangle$
3	reject	$\vert 1\rangle$

Table 1: Comparison of two approaches to $|0\rangle$ state production.

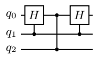
We propose an alternative approach avoiding post-selection, with the simplest example shown in Figure 1b. This circuit fixes $|000\rangle$, $|010\rangle$ and $|001\rangle$, and takes $|100\rangle$ to $|011\rangle$, so outputs $|0\rangle$ on the first wire precisely when there is at most one error on the input. Thus it produces $|0\rangle$ states with slightly higher error rate $3p_0^2 + O(p_0^3)$, but does so without the use of post-selection.

The outcomes for each circuit are summarised in Table 1. Our post-selection-free approach has a higher yield of $|0\rangle$ states, correctly processing all cases with 1 error rather than one of the cases with 2 errors. It greatly simplifies scheduling of $|0\rangle$ production, and avoids delays on architectures where measurement is significantly slower than applying gates. On the other hand it has higher resource requirements, requiring one additional qubit and more gates, including a physical Toffoli.[1]

We call the circuit in Figure 1b a $(3, 1, 1)$ purification circuit because it uses three qubits to prepare one high-quality $|0\rangle$ and can tolerate one error; we define the notation formally at the beginning of Section 1. For the $(3, 1, 1)$ circuit and a selection of other circuits described in more detail later, Figure 2 shows the output error rate as a function of the input error rate, assuming that gates depolarise qubits with probability 0.003 and idle qubits depolarise with probability 0.001 in each round. The $(3, 1, 1)$ circuit improves a 2% preparation error rate to a 0.5% error rate and a 1% preparation error rate to a 0.4% error rate.

Figure 3 shows the thresholds a gate set must meet in order for the circuit of Figure 1b to improve preparation quality at fixed idle depolarisation rate 0.001. Contours correspond to preparation error rates. If your CNOT and Toffoli depolarisation rates place you to the left of a contour then you obtain an improvement from the $(3, 1, 1)$ circuit.

In the rest of this paper we propose and assess the performance of a range of circuits analogous to that of Figure 1b, and quantify the trade-off in which more complicated circuits provide greater protection against preparation errors but are more vulnerable to gate errors. In Section 1 we define and describe the general properties of purification circuits, explain how to find small examples computationally, and consider ways to combine these small examples to form larger circuits. In Section 2 we describe a particular construction based on graphs inspired by examples discovered in Section 1. The simplest versions of

[1] A variety of constructions could instead be used to approximate a Toffoli gate up to relative phases. For example, the circuit 

q_1-conditionally conjugates the q_2-conditional Z gate to an X, resulting in an operation differing from the Toffoli with target q_0 by -1 precisely on the basis state $|101\rangle$. In our model (see Section 1.1) the state is a mixture of computational basis states so these relative phases are unobservable.

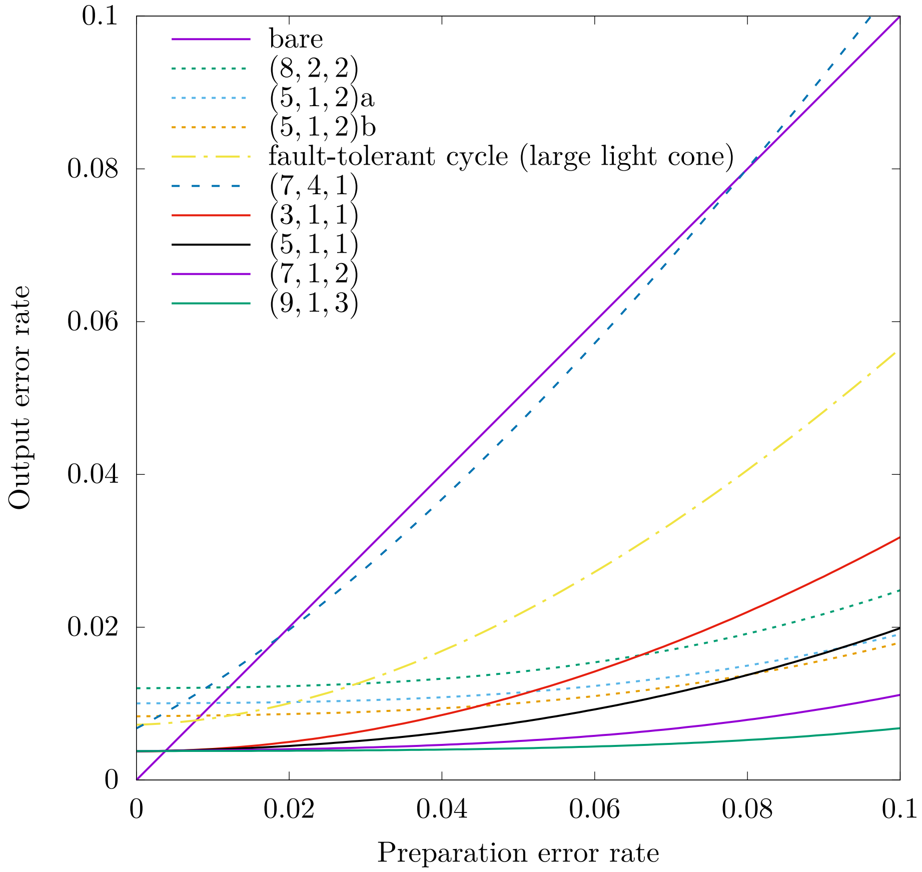


Figure 2: Output error rate as a function of input error rate when gates depolarise with probability 0.003 and idle qubits depolarise with probability 0.001. The curve labelled $(3, 1, 1)$ corresponds to the circuit of Figure 1b.

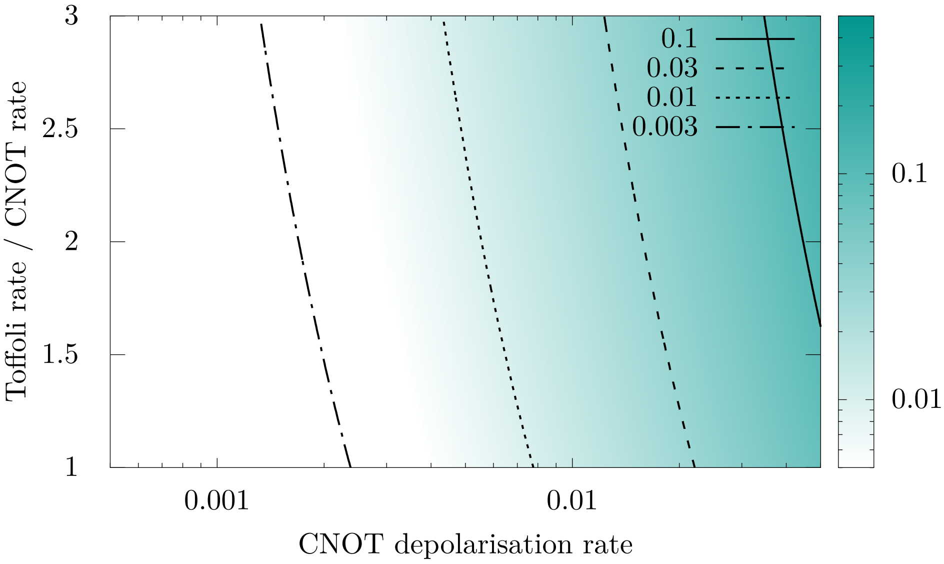


Figure 3: Gate thresholds for the $(3, 1, 1)$ circuit.

these circuits have favourable performance and resource requirements and have a natural planar layout. In Section 3 we present data on the performance of these circuits.

Various schemes making repeated applications of the idea behind Figure 1a have been studied under the name 'algorithmic cooling' [3, 4, 9, 11, 18, 22]. Their analysis is typically in terms of entropy, tracking which qubits are 'hot' or 'cold'. Our approach is combinatorial, instead tracking what happens to individual error patterns as the circuit permutes the computational basis.

1 Purification circuits

We define an (n, k, e) *purification circuit* (or (n, k, e) *circuit*) to be a CNOT and Toffoli circuit mapping the n-long bitstrings of weight at most e into the space of n-long bitstrings that are zero in k nominated positions. If each value in the input is intended to be 0, but experiences an independent error probability p_0 of being flipped to a 1, then each of the k outputs of an (n, k, e) purification circuit experiences a reduced error rate $O(p_0^{e+1})$.

The parameter e is defined in terms of protection against adversarial errors and, as with the distance of a (quantum or classical) error correcting code, does not necessarily give a complete picture of the degree of protection against random errors. The worst case output error rate of an (n, k, e) purification circuit is $\binom{n}{e+1}p_0^{e+1} + O(p_0^{e+2})$. In many cases only a small fraction of the sets of $e + 1$ preparation errors lead to an error on the output, so the coefficient of p_0^{e+1} will be much smaller. This applies, for example, to the family of circuits that we consider in Section 2.

1.1 Error model

An (n, k, e) purification circuit can be viewed as a purely classical object operating on classical bitstrings. To interpret it as a quantum circuit running on noisy hardware we adopt the following error model for preparation, idle and gate errors.

The basic error events are:

- a qubit is incorrectly prepared as $|1\rangle$ rather than $|0\rangle$ with probability p_0;
- a qubit not involved in the current round of gates depolarises with probability p_I;
- a CNOT gate depolarises the set of qubits it acts on with probability p_C;
- a Toffoli gate depolarises the set of qubits it acts on with probability p_T.

All of these events are independent.

By 'a set Q of qubits depolarises' we mean any of the following equivalent things.

- the qubits in Q are replaced by a uniform mixture of the computational basis;
- each qubit in Q experiences a Pauli I, X, Y or Z error chosen uniformly and independently at random;
- each qubit in Q experiences a Pauli X error with probability $1/2$ and a Pauli Z error with probability $1/2$, with all choices made independently.

Note that the parameterisation for preparation errors differs from that for idle and gate errors; preparing $|1\rangle$ rather than $|0\rangle$ with probability $p_0 < 1/2$ corresponds to correctly preparing $|0\rangle$ then depolarising with probability $2p_0$.

Since failures in either preparation or application of a gate in this model replace qubits by mixtures of computational basis states, the state of the system at any point is fully described by a probability distribution over the computational basis. Each probability is a polynomial in p_0, p_I, p_C, p_T, which can be computed precisely for circuits of moderate size. See Appendix E for more details.

This simple model has the significant advantage of being easy to compute with. The disadvantage is that it does not capture all possible error processes within a quantum computer. To give just one example, suppose that your CNOT gate is composed of CZ and Hadamard gates, and that a more accurate error model is that each component gate has an independent probability of depolarising the qubits it acts on. This is not the same as the combined CNOT gate having some probability of depolarising the qubits it acts on; it doesn't even have the property that a system with this noise model can be described as a mixture of computational basis states. Fully realistic noise models are of course more complicated again.

1.2 Existence

We can view an (n,k,e) purification circuit as a permutation of $\mathbb{F}_2^n$ which maps the set of vectors of weight at most e into the set of vectors that vanish in k nominated positions. This places a size constraint on n,k,e. The necessary condition is also sufficient.

Proposition 1. *Let $n \geq 1$ and $k, e \in \{1,\ldots,n\}$. If*

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{e} \leq 2^{n-k}, \tag{1}$$

then there is an (n,k,e) purification circuit consisting of CNOT and Toffoli gates.

Note that this is false if we restrict to circuits containing only CNOTs (which act linearly on $\mathbb{F}_2^n$) or to circuits containing only Toffolis (which fix the set of states of weight at most 1). Any purification circuit for which (1) is equality is optimal in the sense that, for $p_0 < 1/2$, it maps the most likely 2^{n-k} basis states to the most useful 2^{n-k} states.

We prove Proposition 1 in Appendix A. The argument is group-theoretic, and does not provide an efficient procedure to construct purification circuits with given parameters. For the special case of $(2^{m+1}-1, 1, 2^m-1)$ circuits we describe an explicit if impractical construction in Appendix B.

1.3 Finding small circuits

An $(n,1,1)$ purification circuit must have $n \geq 3$. It is straightforward to check that no one or two gate circuit is a $(3,1,1)$ circuit, so the $(3,1,1)$ circuit in Figure 1b is the smallest example that can defend against a single preparation error. (There is one other family of examples, obtained by replacing the second CNOT gate by a Toffoli.) Similarly, an $(n,1,2)$ circuit must have $n \geq 5$. An exhaustive search reveals that the smallest $(5,1,2)$ circuits have 9 gates; two examples are shown in Figure 4. Equivalent circuits can be obtained by permuting the bottom four wires, or by interchanging consecutive commuting

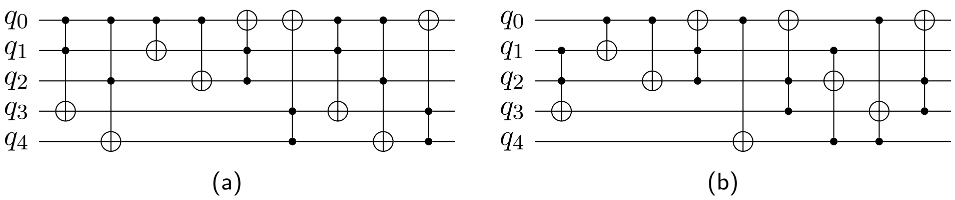


Figure 4: Examples of the two classes of $(5, 1, 2)$ purification circuits of length 9.

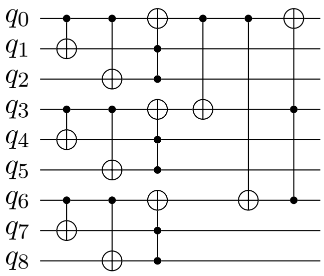


Figure 5: Composing $(3, 1, 1)$ circuits to obtain a $(9, 1, 3)$ circuit.

gates. All 384 $(5, 1, 2)$ circuits of length 9 which output on the top wire can be obtained from these examples in this way: 288 variations of the first type and 96 variations of the second.

There are $2^5 = 32$ basis states of 5 qubits, so a set of states (such as the low weight states, or the states that are $|0\rangle$ on the first qubit) can be represented by a string of 32 bits. By working forward from the initial set of states, backward from the target set of states and meeting in the middle, we can find $(5, 1, 2)$ circuits very quickly, in around $\sqrt{2^{32}} = 2^{16}$ time and space; see Appendix D for more details.

To find a $(7, 1, 3)$ circuit in this way the comparable figure is $\sqrt{2^{2^7}} = 2^{64}$ time and space, so we are already at the limit of what can be achieved without special insight into the problem. In the next section we present techniques for combining purification circuits. The results do not have optimal parameters (n, k, e), but do have relatively simple structures and avoid the requirement to do large amounts of work upfront to discover them.

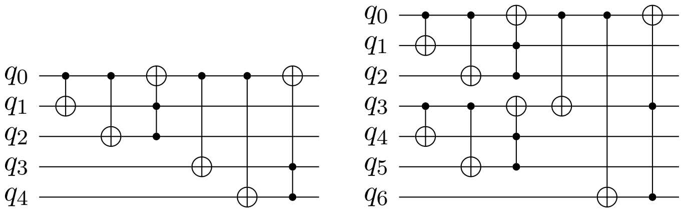
1.4 New purification circuits from old

In this section we describe two techniques for constructing new purification circuits from existing ones.

1.4.1 Composition

The simplest and most general technique is to compose circuits, of similar or different types, to obtain larger ones. Figure 5 shows three $(3, 1, 1)$ circuits feeding their outputs into a fourth $(3, 1, 1)$ circuit. An error on the output requires errors on at least two of the inputs of the fourth circuit, which requires at least two errors on at least two of the first three circuits, so the composed circuit has parameters $(9, 1, 3)$. By Proposition 1, a $(9, 1, e)$ purification circuit could in principle have $e = 4$, so we have given up something on the achievable protection in exchange for a concrete circuit with a simple structure.

By the same argument we obtain the following.



(a) $(5, 1, 1)$ (b) $(7, 1, 2)$

Figure 6: Interpolating between $(3, 1, 1)$ and $(9, 1, 3)$.

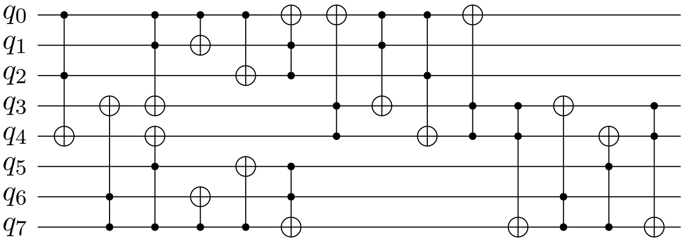


Figure 7: An $(8, 2, 2)$ circuit obtained by overlapping two $(5, 1, 2)$ circuits. Output is on the first and last wires.

Proposition 2. *Feeding the outputs of n_2 copies of an $(n_1, 1, e_1)$ purification circuit with g_1 gates into an $(n_2, 1, e_2)$ purification circuit with g_2 gates produces an $(n_1 n_2, 1, (e_1 + 1)(e_2 + 1) - 1)$ circuit with $n_2 g_1 + g_2$ gates.* □

Observe that feeding $(5, 1, 2)$ circuits into a $(3, 1, 1)$ circuit or feeding $(3, 1, 1)$ circuits into a $(5, 1, 2)$ both result in $(15, 1, 5)$ circuits, but 5-into-3 uses 30 gates and 3-into-5 uses 20 gates, so there is typically an incentive to use simpler circuits closer to the raw input.

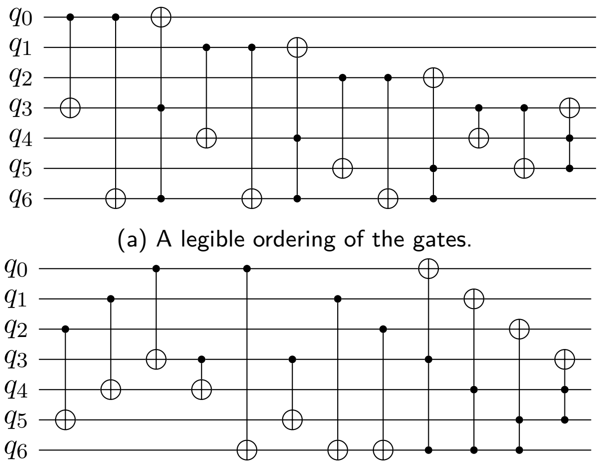
There is no requirement that the circuits in the first stage are identical. For example, we might replace some of the first round $(3, 1, 1)$ circuits in the $(9, 1, 3)$ circuit by naive preparations. Changing only some of the inputs in this way, as shown in Figure 6, allows us to interpolate between the performance and resource requirements of the $(3, 1, 1)$ circuit and the $(9, 1, 3)$ circuit.

1.4.2 Juxtaposition and overlapping

So far all of our circuits have produced a single output. The simplest way to get multiple outputs is to use multiple circuits; m copies of an (n, k, e) circuit can be viewed as a single (mn, mk, e) circuit. This is very far from optimal, as we are over-protected against sets of e errors split between the m circuits. In favourable situations we can exploit this fact by overlapping circuits to re-use qubits. For example, the circuit in Figure 7 comprises two copies of the $(5, 1, 2)$ circuit from Figure 4a sharing qubits q_3 and q_4. We can check that this circuit retains the full protection of both original copies, for an overall parameter set $(8, 2, 2)$. This circuit has been drawn to highlight its symmetry, but its depth can be reduced by 1 to 13 by performing the first four gates over two rounds.

1.5 Relation to classical codes

A $(2e+1, 1, e)$ purification circuit can be viewed as a decoder for the $(2e+1)$-bit classical repetition code. The output bit contains the majority vote of the input bits, and the other



(a) A legible ordering of the gates.

(b) A logically equivalent circuit that can be scheduled over 8 rounds.

Figure 8: $(7, 4, 1)$ purification circuits based on the Hamming code.

bits contain the syndrome information required to reconstruct the input. Any CNOT and Toffoli circuit for decoding a classical $[n, k, 2e + 1]$ code can be viewed as an (n, k, e) purification circuit in the same way.

This is generally more than we need; we only require that strings close to the zero codeword are correctly decoded. Take for example the $[7, 4, 3]$ Hamming code. By Proposition 1, there is a CNOT and Toffoli circuit bijecting the ball of radius 1 about 0 with the space of vectors that vanish in 4 nominated positions. Since there are only $\binom{128}{8} \approx 2^{40}$ sets of 8 basis states on 7 qubits, searching for these circuits is feasible, unlike the situation for $(7, 1, 3)$ circuits described in Section 1.3. There are 508 022 784 such circuits. By inspecting a small number of these and rearranging gates by hand we arrive at the circuit presented in Figure 8a. We emphasise that this is not a full decoding circuit for the Hamming code.

We can interpret this circuit as four consecutive $(3, 1, 1)$ circuits on sets of wires $\{0, 3, 6\}$, $\{1, 4, 6\}$, $\{2, 5, 6\}$, $\{3, 4, 5\}$. The first three circuits share the auxiliary qubit q_6. The final circuit re-uses the remaining auxiliary qubits from the first three circuits. Figure 8b shows a logically equivalent circuit obtained by commuting gates past each other greedily that can be scheduled over 8 rounds. This is the form of the circuit that we use in simulations.

In the next section we present a very general construction inspired by circuits like that in Figure 8a.

2 Purification circuits from graphs

With perfect gates, an (n, k, e) purification circuit improves a preparation error rate p_0 to $O(p_0^{e+1})$. In practice purification circuits will themselves be subject to error, meaning that the dominant term in the output error probability is likely to be due to gate failures, for example of the final Toffoli. There is therefore limited advantage to increasing e. In this section we focus on increasing the *rate* k/n, presenting a general construction, based on graphs, which allows us to tune the rate against other parameters like circuit depth.

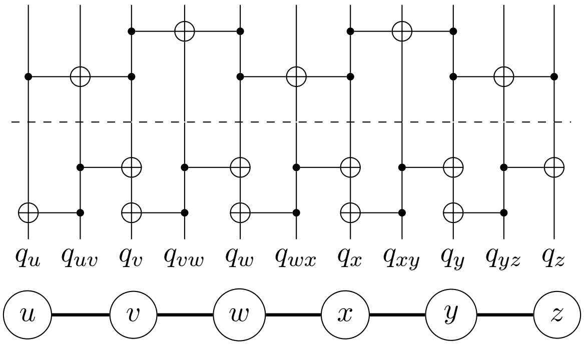


Figure 9: In the graph construction, each edge becomes a $(3, 1, 1)$ circuit. In the first stage, CNOTs copy an error on any edge to its endpoints. In the second stage, Toffolis correct the error on any edge which marked its endpoints in this way. The gates within each stage commute and can be applied in any order.

2.1 Reinterpreting the $(3, 1, 1)$ circuit

We can interpret the operation of the $(3, 1, 1)$ circuit as follows. Think of the first wire as a data qubit, and the other two wires as auxiliary qubits. When the single allowed error is on the data, the two CNOTs mark the auxiliary qubits. This double mark is detected by the Toffoli, which clears the data error. When the error is on an auxiliary qubit, it can't spread back through the CNOTs, and it isn't able to activate both controls of the Toffoli, so the data remains unchanged. If instead we have a auxiliary qubits then we have $\binom{a}{2}$ pairs available for marking in this way, which can be used to protect up to $\binom{a}{2}$ data qubits against a single preparation error. This construction is naturally expressed in the language of graphs.

2.2 The general construction

Given a graph G with r vertices and s edges, we can obtain an $(r + s, s, 1)$ purification circuit as follows. Associate one data qubit q_{uv} to each edge uv, and one auxiliary qubit q_v to each vertex v.

(detect stage) For each edge uv, apply CNOTs controlled on q_{uv} and targeting q_u, q_v.

(correct stage) For each edge uv, apply Toffolis controlled on q_u, q_v and targeting q_{uv}.

The gates within each stage commute, so can be performed in any order. The circuit has $2s$ CNOTs and s Toffolis, and can be scheduled over at most $3\chi_e(G)$ rounds, where $\chi_e(G)$ is the edge-chromatic number of G. See Figure 9 for the result of applying this process to a path of length 5.

There is a superficial resemblance between these circuits and those implementing a surface code (or repetition code, in the case where G is a path or cycle). Data qubits are associated with each edge, and 'syndrome extraction' consists of CNOTs implementing a boundary operator, revealing those vertices incident to an odd number of edges with data errors. The two processes then diverge, as a true surface code makes corrections based on considering a global syndrome; we instead make local corrections to edges both of whose endpoints are marked in the syndrome. This works well for isolated errors, but not for more complicated error configurations.

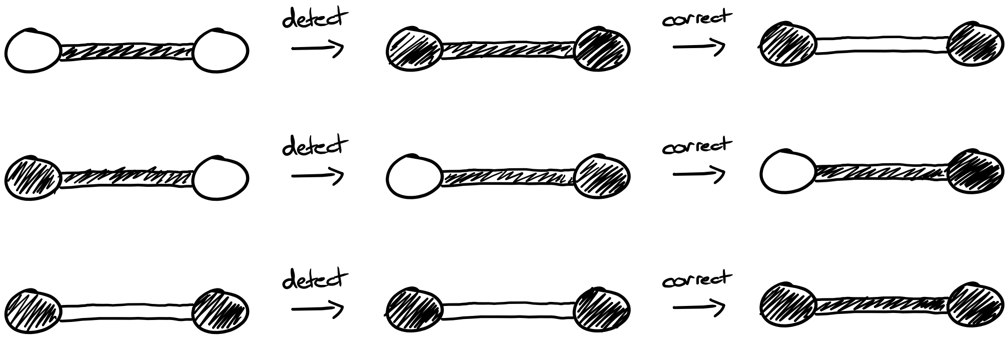


Figure 10: Correct operation of the $(3, 1, 1)$ circuit viewed as a graph and the two different ways in which it fails to handle two errors.

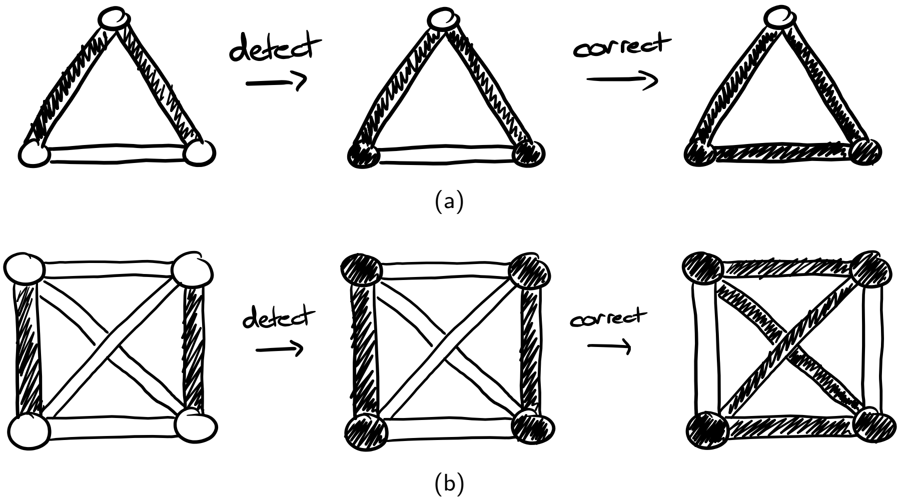
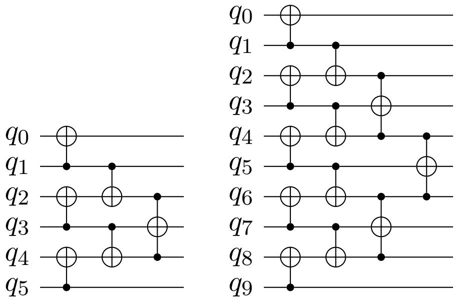


Figure 11: Failure of the complete graph construction with data errors on two different edges.



(a) Output on q_3. (b) Output on q_5.

Figure 12: Small and large light cones over a long even cycle. When viewed as parts of the same circuit over a path or cycle, there is a further idle step following the short light cone.

Figure 10 shows the correct operation of the $(3, 1, 1)$ circuit when there is a single error on the data edge, and the two ways it can fail when there are two errors spread across the data edge and auxiliary vertices. In larger graphs there are other modes of failure. Errors on two incident edges cancel out part of their boundary, leading to neither error being cleared and inducing an error on any edge spanning their other two endpoints (Figure 11a). Errors on two disjoint edges are successfully cleared by the circuit, but cause errors on any other edges induced by these vertices (Figure 11b).

Taking G to be the complete graph K_r produces an $(r + \binom{r}{2}, \binom{r}{2}, 1)$ purification circuit. If we write $k = \binom{r}{2}$, then the parameters become $(k+O(\sqrt{k}), k, 1)$; that is, we can protect a set of k qubits against a single preparation error with only $O(\sqrt{k})$ overhead. The optimal overhead is at least $\log_2 k$ (as we must be able to map every single qubit error to an error pattern supported on the auxiliary qubits), so this is not too far from best possible.

2.3 Paths and cycles

In addition to failing for many sets of two preparation errors (Figure 11), the circuit for K_r has the disadvantage that it requires $\Omega(r)$ rounds of gates. These disadvantages can be addressed by choosing a sparser graph, such a long cycle. If G is a cycle C_k on k vertices, then the circuit can be scheduled over 4 (if k is even) or 5 (if k is odd) rounds, and has a natural planar layout (cf. Figure 9). It also significantly outperforms its parameters.

A $(2k, k, 1)$ circuit defined over a cycle protects completely against any single error. Two or more errors on the input might lead to an error on the output, but only if those errors are sufficiently close. By the *light cone* of an output qubit we mean the subset of the gates and input qubits that can causally affect it. The state of the output qubit depends only on the preparation and gate errors on this part of the circuit. Over a long even cycle there are only two types of output qubit up to isomorphism (those in the first round of Toffolis and those in the second round) and so two possible light cones, shown in Figure 12. This allows us to analyse the output quality of the circuit over any sufficiently long even cycle by examining only two light cones. Since the larger light cone only reaches 10 qubits, 'sufficiently long' means $k \geq 10$.

With preparation error rate p_0 and perfect gates, each output qubit experiences errors with probability $O(p_0^2)$. Errors on output qubits are independent if they are sufficiently separated that their light cones are disjoint, but errors on nearby output qubits are correlated. This might lead to the following situation. Suppose that some set of a errors on the input leads to $b > a$ errors on the output. Then an event which should have probability

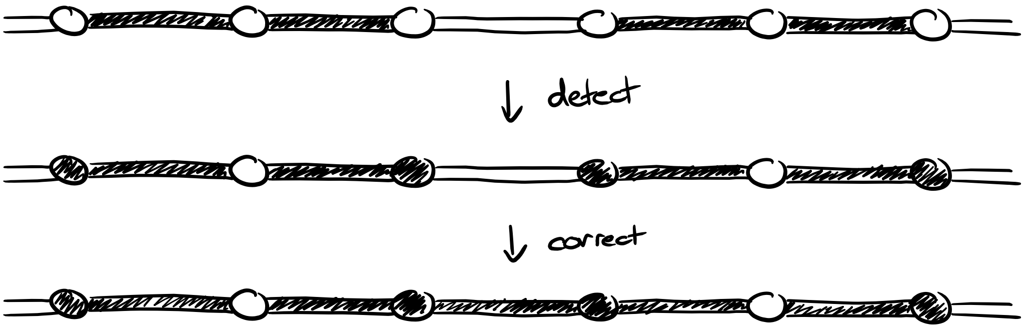


Figure 13: Smallest non-fault tolerant error pattern on a path or long cycle.

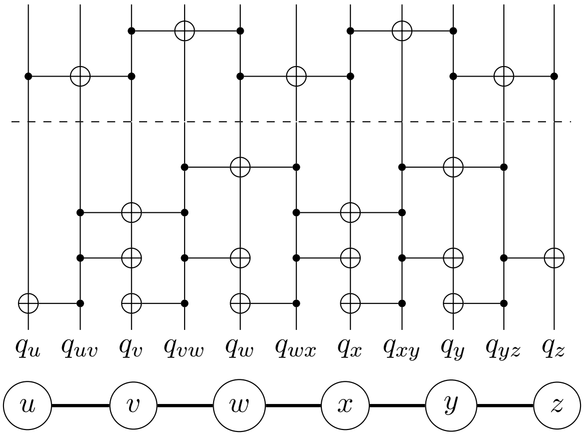


Figure 14: Graph circuits arising from paths or cycles can be made fully fault-tolerant by adding additional Toffolis to ensure that the shared vertex is set to $|1\rangle$ when there are errors on consecutive edges.

$O(p_0^b)$ in fact occurs with probability $\Omega(p_0^a)$. Depending on the intended use of the output qubits, this might be problematic.

We call a purification circuit *combinatorially fault-tolerant up to b preparation errors* if, for any $a \leq b$, any set of a errors on the input leads to at most a errors on the output. For example, one can check that the circuits corresponding to long cycles are combinatorially fault-tolerant up to 3 preparation errors and that there is exactly one pattern of 4 errors on the input leading to more than 4 errors on the output (Figure 13). This pattern works as follows. Input errors on two adjacent edges are frozen in place as their shared auxiliary qubit is first set to $|1\rangle$ then reset to $|0\rangle$, so does not activate the Toffolis. Two of these frozen patterns placed one edge apart preserve the original errors but also causes a new error on the separating edge.

One way to stop this behaviour is to prevent errors on adjacent pairs of edges from being frozen in place. We can do this by adding an extra round of Toffolis to the detect stage.

(detect stage) For each edge uv, apply CNOTs controlled on q_{uv} and targeting q_u, q_v.

(detect′ stage) For each pair of consecutive edges uv, vw, apply Toffolis controlled on q_{uv}, q_{vw} and targeting q_v.

(correct stage) For each edge uv, apply Toffolis controlled on q_u, q_v and targeting q_{uv}.

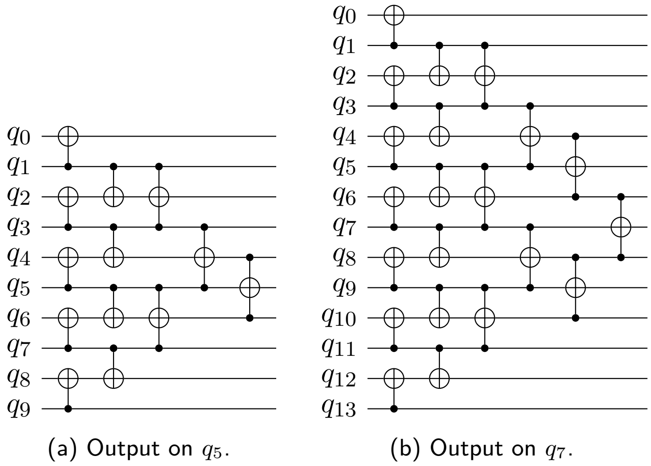


Figure 15: Small and large light cones for the fault-tolerant circuit over a long even cycle. When viewed as parts of the same circuit over a path or cycle, there is a further idle step following the short light cone.

The gates in both detect stages all commute with each other, so can be performed in any order. One choice of circuit is shown in Figure 14, with corresponding light cones in Figure 15.

The idea behind this extended circuit is that, if there are no input errors on the vertices, then the detect stage marks *all* of the endpoints of each edge with an input error, with no cancellation arising from errors on consecutive edges. Then an input error on an edge is successfully cleared unless there is also an input error on one of its endpoints; and an edge with no input error experiences an error on the output only if, for both of its endpoints, there is an input error on either that vertex or the next edge.

Theorem 3. *The extended purification circuit defined over a path or cycle is combinatorially fault-tolerant for any number of preparation errors.*

We give the elementary proof in Appendix C.

We have now described all of our constructions. In the next section we present data on their performance.

3 Performance of purification circuits

For each circuit or light cone discussed in Sections 1 and 2 we computed the distribution of the output qubits. The 2^k probabilities in this output distribution are polynomials in p_0, p_I, p_C, p_T. Rather than perform Monte Carlo experiments we calculated enough terms of these polynomials to obtain the desired accuracy across the parameter range of interest; we give more details in Appendix E.

The circuits we simulate are listed in Table 2 along with some basic parameters. To simulate a circuit with idle errors we need to decide how to schedule the gates into rounds. For the cycle-based constructions the gates were scheduled manually as described in Figures 12 and 15. For the other constructions the gates were scheduled greedily, commuting gates past each other when necessary. The circuit diagrams in this paper were programmatically generated from the scheduled circuits as simulated, and typeset using yquant [6]. The typeset circuits respect the scheduled gate order, but to avoid visual clutter we have

name	leading order error probability	n/k	gates/k	depth	figure
bare	p_0	1	0	0	
$(3,1,1)$	$3p_0^2 + 4p_0(p_I/2) + (p_I/2)^2 + 3(p_C/4) + 4(p_T/8)$	3	3	3	1b
$(5,1,1)$	$p_0^2 + 8p_0(p_I/2) + 16(p_I/2)^2 + 3(p_C/4) + 4(p_T/8)$	5	6	6	6a
$(7,1,2)$	$6p_0^3 + p_0(p_I/2) + 4(p_I/2)^2 + 3(p_C/4) + 4(p_T/8)$	7	9	6	6b
$(9,1,3)$	$27p_0^4 + 12p_0^2(p_I/2) + (p_I/2)^2 + 3(p_C/4) + 4(p_T/8)$	9	12	6	5
$(5,1,2)$a	$10p_0^3 + 5p_0(p_I/2) + 9(p_I/2)^2 + 3(p_C/4) + 21(p_T/8)$	5	9	9	4a
$(5,1,2)$b	$10p_0^3 + (p_I/2) + 3(p_C/4) + 15(p_T/8)$	5	9	9	4b
$(8,2,2)$	$(29p_0^3 + 8(p_I/2) + 6(p_C/4) + 42(p_T/8))/2$	4	9	13	7
$(7,4,1)$	$(2p_0 + 18(p_I/2) + 14(p_C/4) + 20(p_T/8))/4$	7/4	3	8	8b
cycle (small light cone)	$8p_0^2 + (p_I/2) + 3(p_C/4) + 4(p_T/8)$	2	3	4	12a
cycle (large light cone)	$8p_0^2 + (p_I/2) + 3(p_C/4) + 4(p_T/8)$	2	3	4	12b
fault-tolerant cycle (small light cone)	$6p_0^2 + (p_I/2) + 3(p_C/4) + 12(p_T/8)$	2	4	6	15a
fault-tolerant cycle (large light cone)	$6p_0^2 + (p_I/2) + 3(p_C/4) + 12(p_T/8)$	2	4	6	15b

Table 2: Leading order error probability on each output qubit. When $k > 1$, this is interpreted as the expected number of errors on the output divided by k.

not forced them to respect the scheduling of gates into rounds. The exact sequence of idle, CNOT and Toffoli gates simulated can be generated from the published source code [21]. For example, the $(3,1,1)$ circuit simulated is

```
Circuit {outputBits = [0],
         gatesOf = [CNOT 0 1,Idle 2,CNOT 0 2,Idle 1,Toffoli 1 2 0]}.
```

For a circuit or light cone we write $p_{\text{out}}(p_0, p_I, p_C, p_T)$ for the expected number of errors on the output, divided by the number of outputs. When there is a single output this is the probability of an error on that qubit; when there are multiple outputs this is the average probability of an error on each qubit.

Let g_1, g_2, g_3 be the number of idle, CNOT and Toffoli gates in a circuit. The polynomial $p_{\text{out}}(p_0, p_I, p_C, p_T)$ is a sum of terms of the form

$$p_0^{f_0}(1-p_0)^{n-f_0}(p_I/2)^{f_1}(1-p_I)^{g_1-f_1}(p_C/4)^{f_2}(1-p_C)^{g_2-f_2}(p_T/8)^{f_3}(1-p_T)^{g_3-f_3},$$

each corresponding to a run of the circuit in which there were f_0 preparation errors, f_1 idle errors, f_2 CNOT errors and f_3 Toffoli errors. In particular,

$$p_{\text{out}}(p_0, 0, 0, 0) = \sum_{i=1}^{n} a_i p_0^i (1-p_0)^{n-i} \tag{2}$$

for some $a_i \geq 0$, where the $i = 0$ term is 0 as there are no errors on the output if there are no errors on the input. We plot these polynomials for each circuit in Figure 16. Note that the small and large light cones of each cycle construction, and the two $(5,1,2)$ circuits, are equivalent in the absence of idle and gate errors, so only one representative of each type is plotted in Figure 16.

As expected, near 0 the performance of the circuits is determined by the leading order term of their error probability. We include the plot for the full range of preparation errors to point out some general features.

- All curves pass through $(0,0)$, as the zero state is fixed by CNOT and Toffoli gates.
- All curves pass through $(0.5,0.5)$, as a uniform mixture of the basis states after preparation is preserved by every gate. For the majority vote-based $(2e+1,1,e)$ circuits, the curves are symmetric about this point.
- At $p_0 = 1$ the output is deterministic. For most circuits, such as the majority vote $(2e+1,1,e)$ circuits, the output is $|1\rangle$, an error. For the cycle construction, an all $|1\rangle$ input happens to produce an all $|0\rangle$ output. (For $p_0 < 0.5$ the error rate at $1-p_0$ is greater than the error rate at p_0, so this isn't an argument in favour of preparing $|1\rangle$ states deliberately.) The $(7,4,1)$ circuit passes through $(1,0.75)$, meaning that an all $|1\rangle$ input produces one $|0\rangle$ and three $|1\rangle$ states on the output.

3.1 Noisy gates

Throughout this section we fix an idle error rate $p_I = 0.001$.

Figure 17 shows the performance of the various circuits with representative gate noise parameters $p_C = p_T = 0.003$. The main difference from Figure 16 is that there can now be errors on the output even without errors on the input. This limits the performance of circuits, moving the y-intercepts in the plot away from 0. For p_0 close to 0 we have $p_{\text{out}} > p_0$, as high quality qubits are damaged by noisy gates. Then for each circuit there is an interval of p_0 for which the circuit reduces the output error rate below p_0. This interval might extend as far as $p_0 = 0.5$ (at which no purification circuit can lead to an improvement) or, as in the case of the $(7,4,1)$ circuit in Figure 17, end much earlier.

These intervals are sections through the region of parameter space on which using each circuit leads to an improvement in average qubit quality. Figure 18 illustrates this region for the $(7,4,1)$ circuit. Figure 19 illustrates this region for the $(3,1,1)$ circuit. Note that these regions are genuinely different in form. For example, immediately below the plane $p_0 = 0.5$ there is no improvement from running the $(7,4,1)$ circuit, but there is an improvement from running the $(3,1,1)$ circuit.

Let $\theta(p_I,p_C,p_T) = \inf\{p_0 : p_{\text{out}}(p_0,p_I,p_C,p_T) = p_0\}$ be the lower threshold preparation error rate for obtaining an improvement. In the setting of Figures 18a and 19, θ corresponds to the lowest intersection point of each vertical line with the boundary surface. In particular, θ is not necessarily continuous.

Figure 3 shows the threshold $\theta(0.001,p_C,p_T)$ for the $(3,1,1)$ circuit. The x-axis shows p_C, the y-axis shows the ratio p_T/p_C of Toffoli to CNOT error rate and the z-axis, represented by colour, shows θ. We include contour lines for $p_0 \in \{0.003, 0.01, 0.03, 0.1\}$. If your CNOT and Toffoli error rates place you on a contour, running the $(3,1,1)$ circuit produces output of the same quality as the input at that value of p_0. If your CNOT and Toffoli error rates place you below-left of a contour, you expect to obtain an improvement from running a circuit at that value of p_0. Examination of plots like Figures 18a and 19 for each circuit show that this holds in practice. It would be interesting to find general conditions under which output error rate is an increasing function of gate error rate. We return to this question in Section 3.2.

Figure 3 shows that the $(3,1,1)$ circuit can improve output qubit quality even for high gate failure rates; for example, a CNOT failure rate of 1% and a Toffoli failure rate of 3% suffices to improve a preparation error rate of 3% (although in practice you would like lower gate errors to obtain a meaningful improvement). The $(5,1,1)$, $(7,1,2)$ and $(9,1,3)$

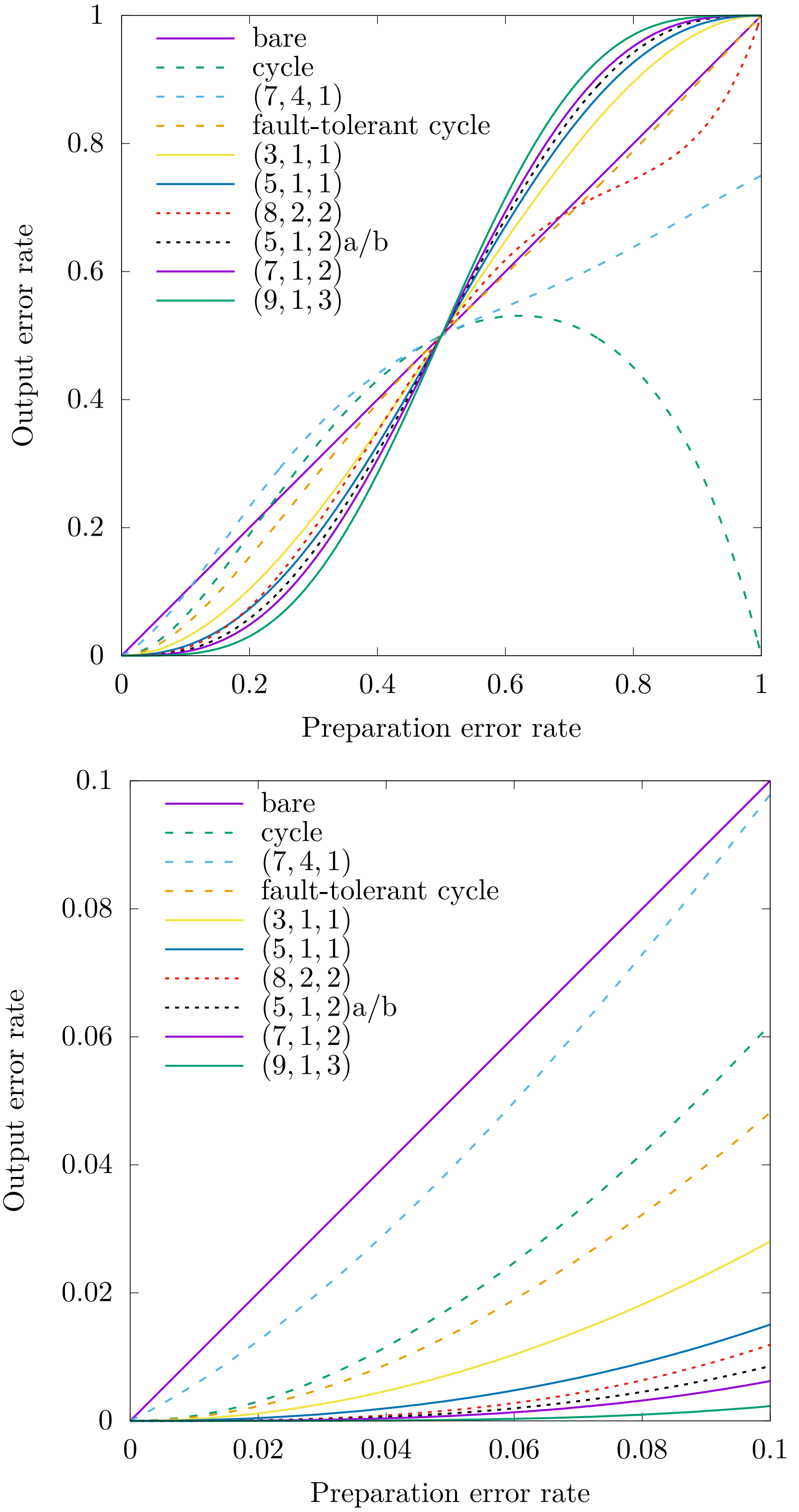


Figure 16: Circuit performance with perfect gates. When $k > 1$, output error rate is interpreted as expected number of errors divided by k.

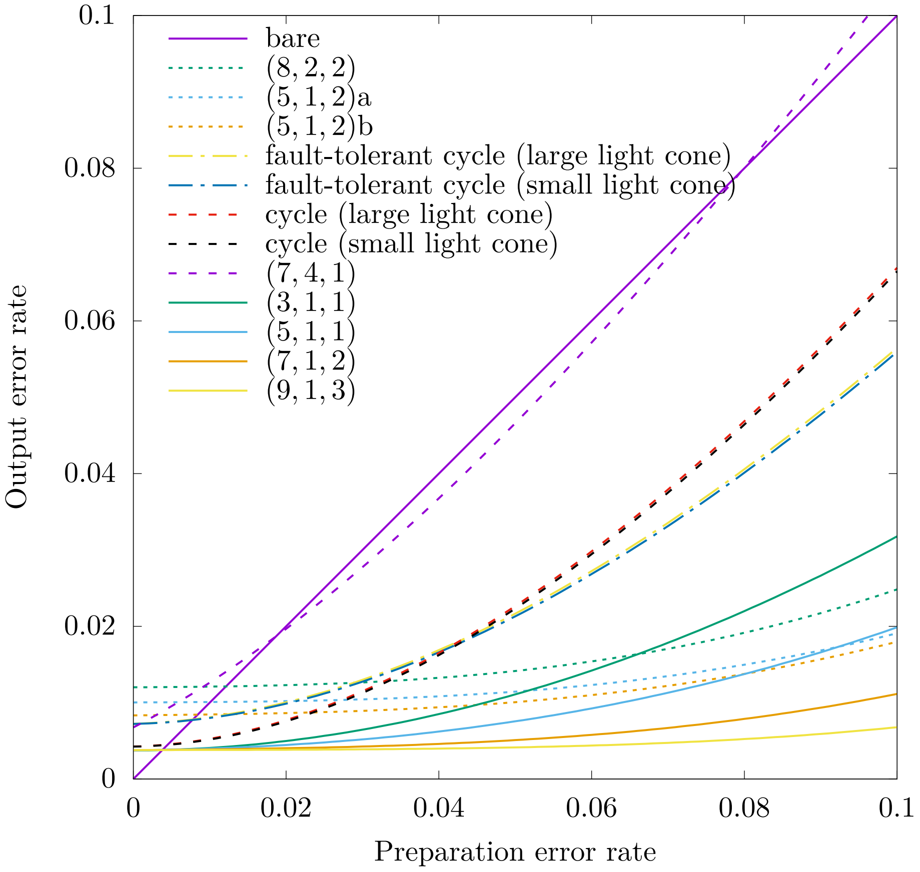
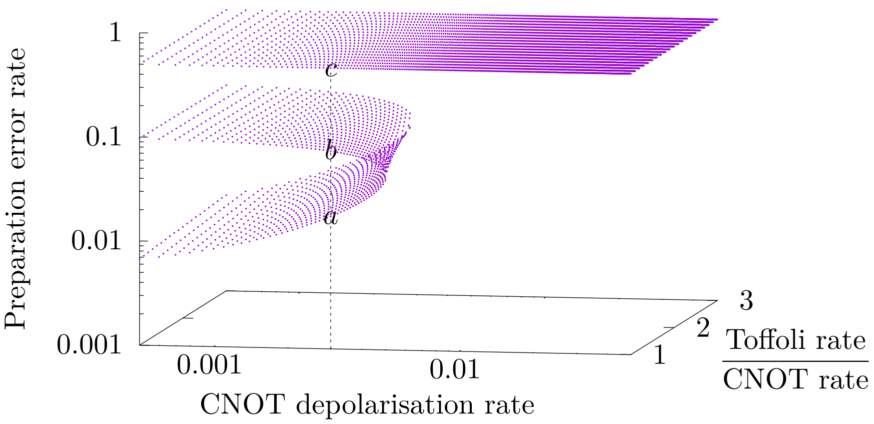
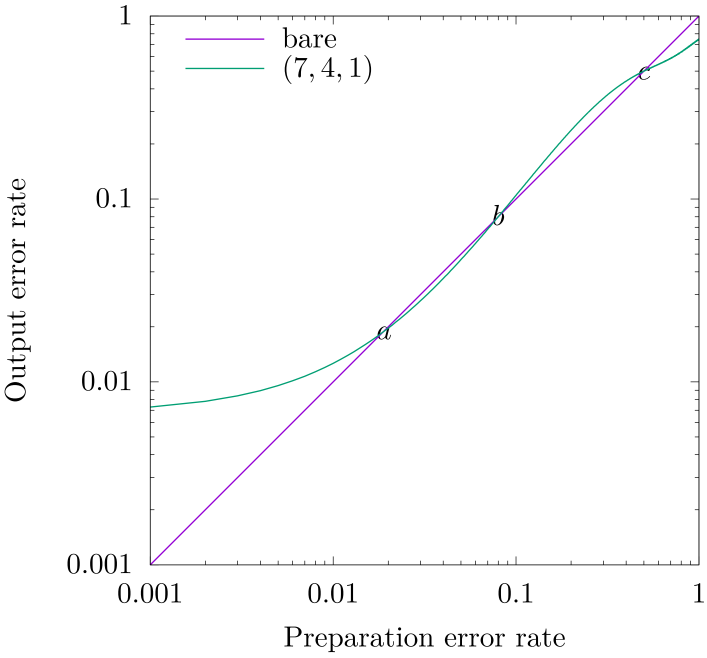


Figure 17: Circuit performance with noisy gates ($p_I = 0.001, p_C = p_T = 0.003$). When $k > 1$, output error rate is interpreted as expected number of errors divided by k.



(a) The region of parameter space ($0 \leq p_0 \leq 0.5$, $p_I = 0.001$, $0 \leq p_C \leq 0.05$, $1 \leq p_T/p_C \leq 3$) in which using the $(7, 4, 1)$ circuit leads to an improvement in average qubit quality. Each data point represents a set of parameters for which $p_{\text{out}} = p_0$. The plane at $p_0 = 0.5$ corresponds to the maximum entropy state in which every computational basis state is equally likely. The curved surface bounds the region to its left in which running the circuit provides an advantage over naive preparation.



(b) Each vertical line through Figure 18a intersects the surface exactly where the corresponding curve $p_{\text{out}}(p_0) = p_{\text{out}}(p_0, 0.001, p_C, p_T)$ crosses the line $y = x$. The marked points $a \approx (0.019, 0.019)$, $b \approx (0.080, 0.080)$ and $c = (0.5, 0.5)$ correspond to the identically marked points in Figure 18a.

Figure 18

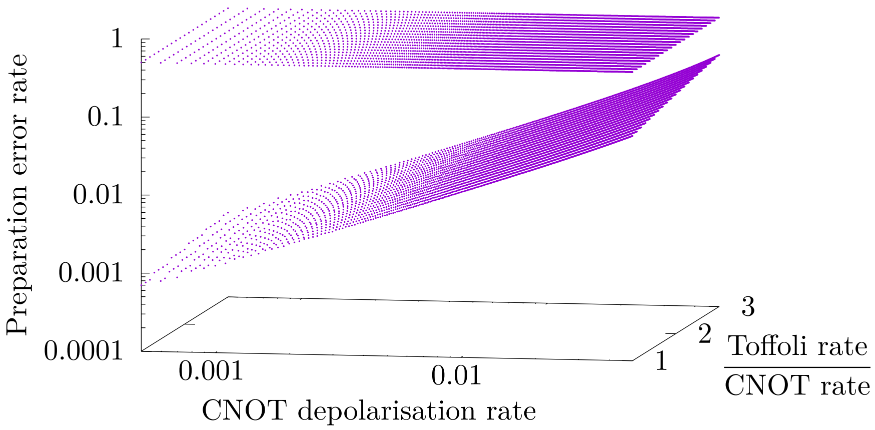


Figure 19: Parameter sets between these two surfaces lead to improvements when running the $(3, 1, 1)$ circuit.

circuits have the same thresholds as the $(3, 1, 1)$ circuit: if applying the $(3, 1, 1)$ circuit reduces error rates, then applying it multiple times also reduces error rates.

Thresholds for the remaining eight circuits are plotted in Figure 20. Note that many of the thresholds exhibit the discontinuity—a sharp step up to 0.5—that we should expect from Figure 18a. Figures 17 and 20 suggest that the basic path/cycle construction of Figure 9 provides very good performance at a modest factor of two overhead in number of qubits.

3.2 Monotonicity

We might expect that running a purification circuit with lower gate and preparation error rates always produces a higher quality output, but Figure 16 shows that this is false for preparation errors and the circuit defined over a cycle.

It is, however, true near 0. It follows from (2) that $p_{\text{out}}(p_0, 0, 0, 0)$ is increasing for $p_0 < 1/n$. A similar argument shows that $p_{\text{out}}(0, p_I, 0, 0)$, $p_{\text{out}}(0, 0, p_C, 0)$ and $p_{\text{out}}(0, 0, 0, p_T)$ are increasing when each failure probability is less than one over the corresponding number of gates. Since p_{out} is a polynomial, continuity of the partial derivatives shows that it is increasing in some neighbourhood of 0.

As remarked earlier, when $\left|\binom{n}{0} + \cdots + \binom{n}{e}\right| = 2^{n-k}$ (so that inequality (1) is tight) and $p_0 < 0.5$, every acceptable input state (pattern of preparation errors that leads to clean output) has higher probability than every unacceptable input state. This means that any averaging process, for example an arbitrary sequence of depolarisations, is bad for the output error rate. Again, this is not enough to show that p_{out} is increasing in each error rate, as successive depolarisations do not necessarily make things progressively worse.

We conjecture that it should be possible to prove monotonicity under some reasonable set of conditions. For instance, suppose that $\left|\binom{n}{0} + \cdots + \binom{n}{e}\right| = 2^{n-k}$ and $p_0 < 0.5$. Is $p_{\text{out}}(p_0, p_I, p_C, p_T)$ an increasing function of p_0, p_I, p_C and p_T? We leave this as an open question.

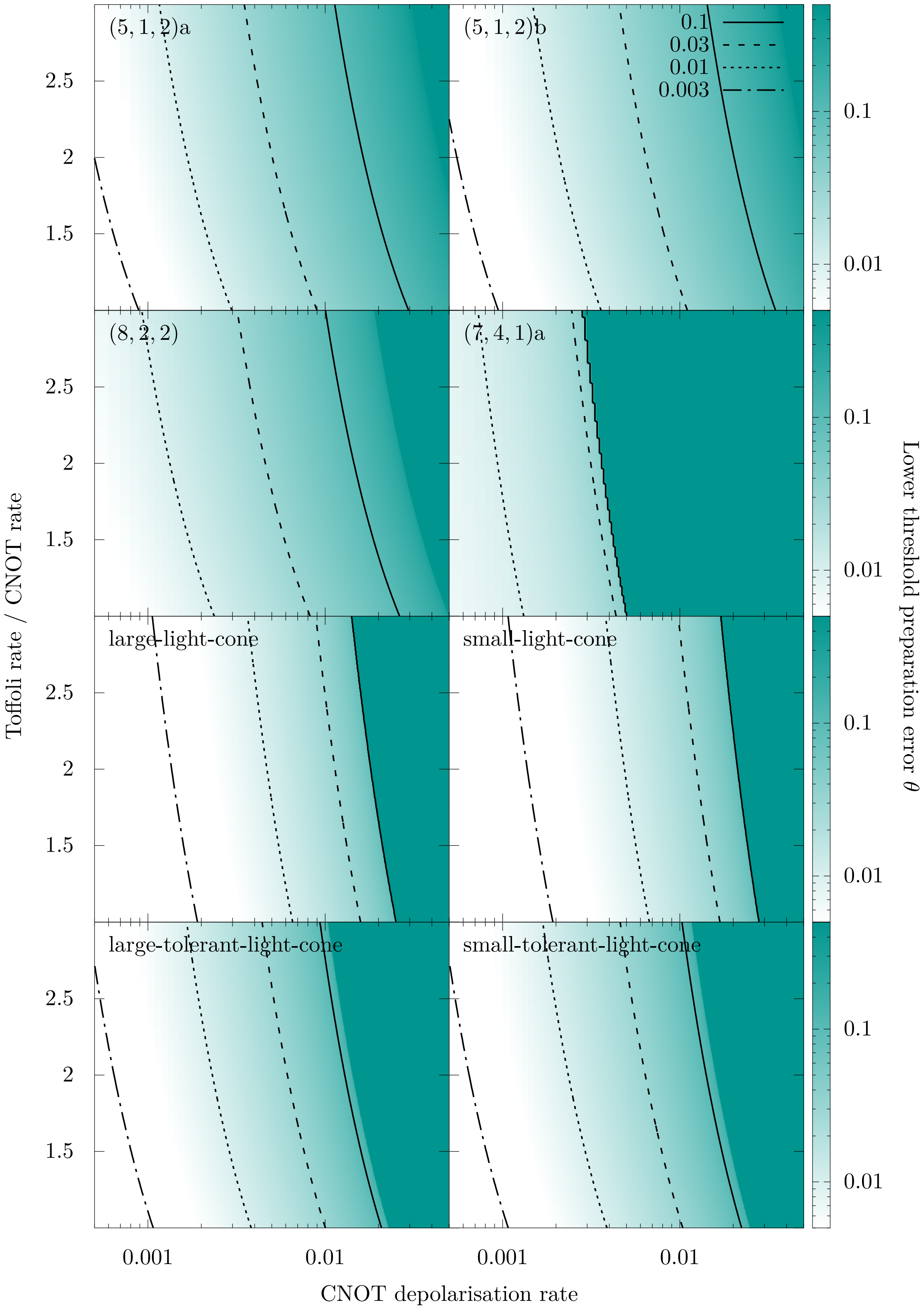


Figure 20: Thresholds for the eight circuits not obtained by composing $(3, 1, 1)$ circuits. Each subplot shows, for each value of p_C and p_T/p_C, the least p_0 beyond which running the corresponding circuit can produce output of higher quality than the input. If your gate errors places you down and to the left of a contour, then you expect an improvement from running the corresponding circuit. See Section 3.1 for a complete description.

4 Outlook

We have described a variety of relatively convenient low depth circuits compatible with 1D arrays of qubits which substantially improve $|0\rangle$ preparation when two-qubit errors are well below preparation errors. We believe non-Clifford circuits are also worth exploring for pre- and post-processing of the measurement subsystems that will be used in syndrome extraction and elsewhere in quantum computers.

The practical use of these circuits remains an open question, as tradeoffs in qubit cost and connectivity, as well as the cost in time and energy for physical resets of unused qubits, depend substantially upon the physical architecture. We would very much like to see data on the practical effect of running our circuits on real quantum computing hardware.

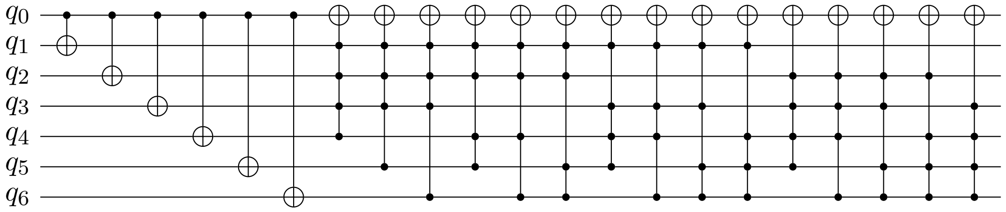


Figure 21: A $(7,1,3)$ purification circuit presented using multiply controlled Toffoli gates.

A Existence of purification circuits

In this section we prove Proposition 1 on the existence of purification circuits with given parameters. We use the following algebraic fact.

Lemma 4. *Let G_n be the group generated by CNOT and Toffoli gates on n qubits. For $n \geq 4$, G_n acts as the alternating group on $\mathbb{F}_2^n \setminus \{0\}$. For $n \leq 3$, G_n acts as the symmetric group on $\mathbb{F}_2^n \setminus \{0\}$.*

Proof of Proposition 1. By (1), $1 + n \leq 2^{n-k}$, so we may assume that $n > 1$. Then $3 \leq 2^{n-k}$, and so $k \leq n-2$.

Let σ be any permutation of $\mathbb{F}_2^n \setminus \{0\}$ which maps the vectors of weight at most e into the vectors that vanish in the first k positions. Let τ the transposition that swaps the standard basis vectors e_n and e_{n-1}. Since $k \leq n-2$, $\tau\sigma$ also maps the vectors of weight at most e into the vectors that vanish in the first k positions. One of these permutations is even, so by Lemma 4 can be expressed in terms of CNOTs and Toffolis. □

Proof of Lemma 4. For $n = 1$, all the groups in question are trivial.

For $n \geq 2$, let C_n be the group generated by CNOTs on n qubits. C_n is isomorphic to $\mathrm{GL}(n,2)$ acting naturally on $\mathbb{F}_2^n \setminus \{0\}$. For $n = 2$, this achieves the action of the full symmetric group on $\mathbb{F}_2^2 \setminus \{0\}$.

For $n \geq 3$, let T_n be the group generated by Toffolis on n qubits. T_n acts on the vectors of weight at least 2. For $n = 3$, it acts as the symmetric group $\mathrm{Sym}(4)$; for $n \geq 4$ it acts as the alternating group $\mathrm{Alt}(2^n - n - 1)$ [12, Theorem 1.1.4].

Now consider the group $G_n = \langle C_n, T_n \rangle$.

- The action of G_n on $\mathbb{F}_2^n \setminus \{0\}$ is 2-transitive (because $\mathrm{GL}(n,2)$ is 2-transitive) and therefore primitive.
- The action of G_n on $\mathbb{F}_2^n \setminus \{0\}$ contains a 3-cycle (2-cycle) if $n \geq 4$ ($n = 3$), since T_n does.

As G_n acts primitively on $\mathbb{F}_2^n \setminus \{0\}$ and contains a 3-cycle (2-cycle) if $n \geq 4$ ($n = 3$), it contains the alternating group (symmetric group) on $\mathbb{F}_2^2 \setminus \{0\}$ [7, Theorem 3.3A]. A CNOT or Toffoli gate on at least 4 qubits is an even permutation of the basis states, so this containment is equality. □

B An explicit construction

In this section we give an explicit construction of purification circuits with parameters $(2^{m+1} - 1, 1, 2^m - 1)$.

Let $n = 2^{m+1} - 1$ and $e = 2^m - 1$. Let q_0 be an output qubit and $q_1, \dots, q_{n-1}$ be auxiliary qubits. Form a circuit in two stages.

- In the first stage, perform $n-1$ CNOTs controlled on q_0 and targeting each other q_i.
- In the second stage, perform $\binom{n-1}{e+1}$ Toffolis targeting q_0 and controlled on each set of $e+1$ qubits from $\{q_1, \dots, q_{n-1}\}$.

For $m = 1$ this is the $(3, 1, 1)$ circuit in Figure 1b. For $m = 2$ it is the $(7, 1, 3)$ circuit shown in Figure 21; recall from Section 1.3 that searching for a shortest $(7, 1, 3)$ circuit by brute force was out of reach.

As presented, this circuit already has a large number of gates. If the multi-controlled Toffolis are expanded to conventional Toffolis, this number will increase further, the exact increase depending on how the expansion is performed and the original order of the multi-controlled Toffolis.

Proposition 5. *This circuit is an $(n, 1, e)$ purification circuit.*

Proof. Let $0 \leq r \leq e$ and let v be a vector of weight r. We must show that the circuit maps v to a vector which is 0 in the first position.

If $v_0 = 0$, then every gate of the circuit fixes v, so the output 0 is preserved.

If $v_0 = 1$, then all of the CNOTs activate, mapping v to a vector w with $w_0 = 1$ and $(n-1) - (r-1) = 2e + 1 - r \geq e + 1$ of bits $1, \dots, n-1$ set. This will activate

$$\binom{2e+1-r}{e+1} = \binom{2e+1-r}{e-r} = \binom{2^m + s}{s}$$

of the Toffolis, where $0 \leq s = e - r \leq e < 2^m$. We claim that this number is odd, from which the result follows.

We use induction on s. For $s = 0$ the number is 1, so odd. For $s > 0$, we have

$$\binom{2^m + s}{s} = \frac{2^m + s}{s} \binom{2^m + s - 1}{s - 1},$$

where by induction the binomial coefficient on the right-hand side is odd. Since $s < 2^m$, the number of powers of 2 dividing the numerator and denominator of the fraction agree, so it has the form a/b where both a and b are odd. Hence the left-hand side is odd, as required. □

C Fault-tolerance of the enhanced cycle construction

In this section we prove Theorem 3, that the enhanced cycle construction is fully fault-tolerant.

Proof of Theorem 3. A path on two vertices only has one output, so is automatically fault-tolerant. Otherwise a path behaves like a cycle on the same number of vertices where one nominated edge never has an error on the input, so we prove the result for cycles.

Let V be the set of vertices with an error on the input, E the set of edges with an error on the input and F the set of edges with an error on the output. We must show that $|F| \leq |V| + |E|$, or equivalently that

$$|F \setminus E| \leq |V| + |E \setminus F|. \tag{3}$$

Fix an orientation of the cycle. We will show that, starting from any element of $F \setminus E$ and walking round the cycle clockwise, we encounter an element of $V \cup (E \setminus F)$ before another element of $F \setminus E$. This suffices to prove (3).

Let xyz be a sequence of three consecutive vertices moving clockwise around the cycle. Suppose that $xy \in F \setminus E$ but $y \notin V$. For xy to have an error on the output but not the input, y must be in state $|1\rangle$ after the detect$'$ stage. Since y and xy have no errors on the input, yz must have an error on the input. If $yz \in E \setminus F$ then we are done, so assume that $yz \in E \cap F$. But now, by construction, an input error on an edge cannot persist in the output if there are no input errors on its endpoints. There is no input error on y, so we must have $z \in V$. □

D Searching for circuits

As described in Section 1.2 and Appendix A, CNOT and Toffoli gates on a set of n qubits act as permutations on the computational basis, which we identify with $\mathbb{F}_2^n$. Let Π_n be the set of these CNOT and Toffoli permutations. An (n,k,e) purification circuit is a composition of elements of Π_n which maps the Hamming ball $B(0,e)$ of radius e about 0 into the codimension k subspace V_{n-k} of $\mathbb{F}_2^n$ comprising the vectors which vanish in the first k positions. By Proposition 1, such a composition exists provided $|B(0,e)| \leq |V_{n-k}|$.

We can find such compositions as follows. Let $\mathcal{P}(\mathbb{F}_2^n)$ be the power set of $\mathbb{F}_2^n$ and, for a set S, let $\binom{S}{m}$ be the set of subsets of S of size m. For sets of states $S \in \mathcal{P}(\mathbb{F}_2^n)$, sets of sets of states $\mathcal{S} \subseteq \mathcal{P}(\mathbb{F}_2^n)$ and permutations $\pi \in \Pi_n$, let

- $\pi(S) = \{\pi(s) : s \in S\}$
- $\Pi_n(\mathcal{S}) = \{\pi(S) : \pi \in \Pi_n, S \in \mathcal{S}\}$.

If $\Pi_n^{t_1}(\{B(0,e)\})$ intersects $\Pi_n^{t_2}(\binom{V_{n-k}}{|B(0,e)|})$ then, for some $\pi_i, \rho_i \in \Pi_n$ and some $S \subseteq V_{n-k}$ of size $|B(0,e)|$,

$$\pi_1 \cdots \pi_{t_1}(B(0,e)) = \rho_1 \cdots \rho_{t_2}(S)$$

whence, using the fact that every element of Π_n is self-inverse,

$$\rho_{t_2} \cdots \rho_1 \pi_1 \cdots \pi_{t_1}(B(0,e)) = S \subseteq V_{n-k}.$$

That is, there is an (n,k,e) purification circuit of length $t_1 + t_2$.

We expect to find such an intersection point once

$$|\Pi_n^{t_1}(\{B(0,e)\})| \times |\Pi_n^{t_2}(\tbinom{V_{n-k}}{|B(0,e)|})| \approx \tbinom{2^n}{|B(0,e)|},$$

so time and memory requirements scale roughly as the square root of $\binom{2^n}{|B(0,e)|}$. This contrasts with a naive search over circuits of length $t_1 + t_2$, which requires at least an expected $\binom{2^n}{|B(0,e)|}$ time but constant memory.

The first method has an additional advantage over a naive search. There are typically multiple circuits taking one set of states to another set of states. This is in part due to the existence of multiple circuits implementing the same permutation (obtained, for example, by commuting gates past each other), and in part due to the fact that distinct permutations of states can have the same action on sets of states (for example, the two inequivalent circuits in Figure 4 which both act as $(5,1,2)$ purification circuits). By tracking reachable sets of states rather than circuits, the first method experiences a limited overhead from this phenomenon. A brute force search by contrast might be able to avoid the most obvious cases of repetition (for example, by not applying a pair of commuting gates in both orders), but greater care would be required to avoid duplicated effort.

In [21], `Search.hs` implements this method by applying generic pathfinding functions from `Pathfinding.hs` to the gate set described in `Circuits.hs`.

E Assessing circuits

We want to assess the performance of a purification circuit with n qubits, g_1 idle gates, g_2 CNOTs and g_3 Toffolis. With the error model described in Section 1.1, at every point during the operation of the circuit the quantum state can be described by a probability distribution π over the computational basis states, which we identify with $\mathbb{F}_2^n$. Immediately after preparing all n qubits this distribution is

$$\mu_0(x) = p_0^{\|x\|}(1-p_0)^{n-\|x\|}, \tag{4}$$

where $\|x\|$ is the Hamming weight of x, the number of coordinates in which it takes the value 1.

Suppose that we have applied m gates, achieving a distribution μ_m over $\mathbb{F}_2^n$, and are about to apply a gate G acting on the set of qubits Q. Let p_G be the probability that G depolarises the qubits in Q. Then

$$\mu_m(x) = (1-p_G)\mu_m(G^{-1}(x)) + \frac{p_G}{2^{|Q|}} \sum_{v \in \mathbb{F}_2^Q} \mu_m(x+v). \tag{5}$$

Performing this iteration computes the distribution over the computational basis at the end of the circuit in $O(2^n(g_1+g_2+g_3))$ polynomial arithmetic operations. The polynomials in question have up to $(g_1+1)(g_2+1)(g_3+1)$ terms, so in practice the additional overhead from polynomial arithmetic can be significant.

To reduce the overhead we make two choices. From (4) and (5) it follows that μ_m is a sum of terms of the form

$$p_0^{f_0}(1-p_0)^{n-f_0}(p_I/2)^{f_1}(1-p_I)^{g_1'-f_1}(p_C/4)^{f_2}(1-p_C)^{g_2'-f_2}(p_T/8)^{f_3}(1-p_T)^{g_3'-f_3},$$

where g_1', g_2' and g_3' are the number of idle, CNOT and Toffoli gates applied so far (thus $m = g_1' + g_2' + g_3'$). Using this basis rather than the standard monomial basis means that the first polynomial multiplication in (5) is implicit, and the second is an increment of f_1, f_2 or f_3.

The second choice is to observe that large values of f_i correspond to events that are unlikely, depending on the values of $n, g_1, g_2, g_3, p_0, p_I, p_C, p_T$ and our desired accuracy. So we need only record terms where each f_i is in some restricted range.

In [21], `Simulate.hs` implements this method over the description of circuits in `Circuits.hs`.

References

[1] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. DOI: 10.1038/s41586-019-1666-5.

[2] Jacob Z. Blumoff, Andrew S. Pan, Tyler E. Keating, Reed W. Andrews, David W. Barnes, Teresa L. Brecht, Edward T. Croke, Larken E. Euliss, Jacob A. Fast, Clayton A. C. Jackson, Aaron M. Jones, Joseph Kerckhoff, Robert K. Lanza, Kate Raach, Bryan J. Thomas, Roland Velunta, Aaron J. Weinstein, Thaddeus D. Ladd, Kevin Eng, Matthew G. Borselli, Andrew T. Hunter, and Matthew T. Rakher. Fast and high-fidelity state preparation and measurement in triple-quantum-dot spin qubits. *PRX Quantum*, 3:010352, Mar 2022. DOI: 10.1103/PRXQuantum.3.010352. URL `https://link.aps.org/doi/10.1103/PRXQuantum.3.010352`.

[3] P. Oscar Boykin, Tal Mor, Vwani Roychowdhury, Farrokh Vatan, and Rutger Vrijen. Algorithmic cooling and scalable NMR quantum computers. *Proceedings of the National Academy of Sciences*, 99(6):3388–3393, 2002. DOI: 10.1073/pnas.241641898.

[4] Gilles Brassard, Yuval Elias, Tal Mor, and Yossi Weinstein. Prospects and limitations of algorithmic cooling. *The European Physical Journal Plus*, 129(11):1–16, 2014. DOI: 10.1140/epjp/i2014-14258-0.

[5] S. M. Brewer, J.-S. Chen, A. M. Hankin, E. R. Clements, C. W. Chou, D. J. Wineland, D. B. Hume, and D. R. Leibrandt. $^{27}\mathrm{Al}^{+}$ quantum-logic clock with a systematic uncertainty below 10^{-18}. *Phys. Rev. Lett.*, 123:033201, Jul 2019. DOI: 10.1103/PhysRevLett.123.033201. URL `https://link.aps.org/doi/10.1103/PhysRevLett.123.033201`.

[6] Benjamin Desef. Yquant: Typesetting quantum circuits in a human-readable language. 2020. DOI: 10.48550/ARXIV.2007.12931. URL `https://arxiv.org/abs/2007.12931`. arXiv:2007.12931.

[7] John D. Dixon and Brian Mortimer. *Permutation groups*. Springer, New York, NY, 1996. DOI: 10.1007/978-1-4612-0731-3.

[8] Salvatore S. Elder, Christopher S. Wang, Philip Reinhold, Connor T. Hann, Kevin S. Chou, Brian J. Lester, Serge Rosenblum, Luigi Frunzio, Liang Jiang, and Robert J. Schoelkopf. High-fidelity measurement of qubits encoded in multilevel superconducting circuits. *Phys. Rev. X*, 10:011001, Jan 2020. DOI: 10.1103/PhysRevX.10.011001. URL `https://link.aps.org/doi/10.1103/PhysRevX.10.011001`.

[9] Yuval Elias, Tal Mor, and Yossi Weinstein. Semioptimal practicable algorithmic cooling. *Phys. Rev. A*, 83:042340, Apr 2011. DOI: 10.1103/PhysRevA.83.042340. URL `https://link.aps.org/doi/10.1103/PhysRevA.83.042340`.

[10] Alexander Erhard, Joel J. Wallman, Lukas Postler, Michael Meth, Roman Stricker, Esteban A. Martinez, Philipp Schindler, Thomas Monz, Joseph Emerson, and Rainer Blatt. Characterizing large-scale quantum computers via cycle benchmarking. *Nature Communications*, 10(1):1–7, 2019. DOI: 10.1038/s41467-019-13068-7.

[11] José M. Fernandez, Seth Lloyd, Tal Mor, and Vwani Roychowdhury. Algorithmic cooling of spins: A practicable method for increasing polarization. *International Journal of Quantum Information*, 02(04):461–477, 2004. DOI: 10.1142/S0219749904000419. URL `https://doi.org/10.1142/S0219749904000419`.

[12] David Gajewski. *Analysis of Groups Generated by Quantum Gates*. PhD thesis, University of Toledo, 2009.

[13] Michael R. Geller and Mingyu Sun. Toward efficient correction of multiqubit measurement errors: pair correlation method. *Quantum Science and Technology*, 6(2): 025009, Feb 2021. DOI: 10.1088/2058-9565/abd5c9. URL `https://doi.org/10.1088/2058-9565/abd5c9`.

[14] Rebecca Hicks, Bryce Kobrin, Christian W. Bauer, and Benjamin Nachman. Active

readout-error mitigation. *Phys. Rev. A*, 105:012419, Jan 2022. DOI: 10.1103/PhysRevA.105.012419. URL `https://link.aps.org/doi/10.1103/PhysRevA.105.012419`.

[15] D. B. Hume, T. Rosenband, and D. J. Wineland. High-fidelity adaptive qubit detection through repetitive quantum nondemolition measurements. *Phys. Rev. Lett.*, 99:120502, Sep 2007. DOI: 10.1103/PhysRevLett.99.120502. URL `https://link.aps.org/doi/10.1103/PhysRevLett.99.120502`.

[16] IBM. Rising above the noise: Quantum-limited amplifiers empower the readout of IBM quantum systems. IBM Research Blog, Jan 2020. URL `https://www.ibm.com/blogs/research/2020/01/quantum-limited-amplifiers/`. https://www.ibm.com/blogs/research/2020/01/quantum-limited-amplifiers/.

[17] L. Jiang, J. S. Hodges, J. R. Maze, P. Maurer, J. M. Taylor, D. G. Cory, P. R. Hemmer, R. L. Walsworth, A. Yacoby, A. S. Zibrov, and M. D. Lukin. Repetitive readout of a single electronic spin via quantum logic with nuclear spin ancillae. *Science*, 326 (5950):267–272, 2009. DOI: 10.1126/science.1176496. URL `https://www.science.org/doi/abs/10.1126/science.1176496`.

[18] Raymond Laflamme, Junan Lin, and Tal Mor. Algorithmic cooling for resolving state preparation and measurement errors in quantum computing. *Physical Review A*, 106 (1):012439, 2022. DOI: 10.1103/PhysRevA.106.012439.

[19] Ilya N. Moskalenko, Ilya A. Simakov, Nikolay N. Abramov, Alexander A. Grigorev, Dmitry O. Moskalev, Anastasiya A. Pishchimova, Nikita S. Smirnov, Evgeniy V. Zikiy, Ilya A. Rodionov, and Ilya S. Besedin. High fidelity two-qubit gates on fluxoniums using a tunable coupler. *npj Quantum Information*, 8(1):130, 2022. DOI: 10.1038/s41534-022-00644-x.

[20] A. Opremcak, C. H. Liu, C. Wilen, K. Okubo, B. G. Christensen, D. Sank, T. C. White, A. Vainsencher, M. Giustina, A. Megrant, B. Burkett, B. L. T. Plourde, and R. McDermott. High-fidelity measurement of a superconducting qubit using an on-chip microwave photon counter. *Phys. Rev. X*, 11:011027, Feb 2021. DOI: 10.1103/PhysRevX.11.011027. URL `https://link.aps.org/doi/10.1103/PhysRevX.11.011027`.

[21] Riverlane. Source code and data behind this paper. Github, Aug 2022. URL `https://github.com/riverlane/purification-without-post-selection`. https://github.com/riverlane/purification-without-post-selection.

[22] Leonard J. Schulman and Umesh V. Vazirani. Molecular scale heat engines and scalable quantum computation. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, STOC '99, page 322–329, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130678. DOI: 10.1145/301250.301332. URL `https://doi.org/10.1145/301250.301332`.

[23] Youngkyu Sung, Leon Ding, Jochen Braumüller, Antti Vepsäläinen, Bharath Kannan, Morten Kjaergaard, Ami Greene, Gabriel O. Samach, Chris McNally, David Kim, Alexander Melville, Bethany M. Niedzielski, Mollie E. Schwartz, Jonilyn L. Yoder, Terry P. Orlando, Simon Gustavsson, and William D. Oliver. Realization of high-fidelity CZ and ZZ-free iSWAP gates with a tunable coupler. *Phys. Rev. X*, 11:021058, Jun 2021. DOI: 10.1103/PhysRevX.11.021058. URL `https://link.aps.org/doi/10.1103/PhysRevX.11.021058`.

[24] Yasunari Suzuki, Suguru Endo, Keisuke Fujii, and Yuuki Tokunaga. Quantum error mitigation as a universal error reduction technique: Applications from the NISQ to

the fault-tolerant quantum computing eras. *PRX Quantum*, 3:010345, Mar 2022. DOI: 10.1103/PRXQuantum.3.010345. URL https://link.aps.org/doi/10.1103/PRXQuantum.3.010345.

[25] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error mitigation for short-depth quantum circuits. *Phys. Rev. Lett.*, 119:180509, Nov 2017. DOI: 10.1103/PhysRevLett.119.180509. URL https://link.aps.org/doi/10.1103/PhysRevLett.119.180509.

[26] Ye Wang, Stephen Crain, Chao Fang, Bichen Zhang, Shilin Huang, Qiyao Liang, Pak Hong Leung, Kenneth R. Brown, and Jungsang Kim. High-fidelity two-qubit gates using a microelectromechanical-system-based beam steering system for individual qubit addressing. *Phys. Rev. Lett.*, 125:150505, Oct 2020. DOI: 10.1103/PhysRevLett.125.150505. URL https://link.aps.org/doi/10.1103/PhysRevLett.125.150505.

[27] Kenneth Wright, Kristin M. Beck, Sea Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, et al. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 10(1):1–6, 2019. DOI: 10.1038/s41467-019-13534-2.

[28] Wenchao Xu, Aditya V. Venkatramani, Sergio H. Cantú, Tamara Šumarac, Valentin Klüsener, Mikhail D. Lukin, and Vladan Vuletić. Fast preparation and detection of a Rydberg qubit using atomic ensembles. *Phys. Rev. Lett.*, 127:050501, Jul 2021. DOI: 10.1103/PhysRevLett.127.050501. URL https://link.aps.org/doi/10.1103/PhysRevLett.127.050501.