Efficient magic state factories with a catalyzed $|CCZ\rangle \to 2|T\rangle$ transformation

Craig Gidney and Austin G. Fowler

Google Inc., Santa Barbara, California 93117, USA

April 30, 2019

arXiv:1812.01238v3 [quant-ph] 26 Apr 2019

We present magic state factory constructions for producing $|CCZ\rangle$ states and $|T\rangle$ states. For the $|CCZ\rangle$ factory we apply the surface code lattice surgery construction techniques described in [15] to the fault-tolerant Toffoli [21, 12]. The resulting factory has a footprint of $12d \times 6d$ (where d is the code distance) and produces one $|CCZ\rangle$ every $5.5d$ surface code cycles. Our $|T\rangle$ state factory uses the $|CCZ\rangle$ factory's output and a catalyst $|T\rangle$ state to exactly transform one $|CCZ\rangle$ state into two $|T\rangle$ states. It has a footprint 25% smaller than the factory in [15] but outputs $|T\rangle$ states twice as quickly. We show how to generalize the catalyzed transformation to arbitrary phase angles, and note that the case $\theta = 22.5°$ produces a particularly efficient circuit for producing $|\sqrt{T}\rangle$ states. Compared to using the $12d \times 8d \times 6.5d$ $|T\rangle$ factory of [15], our $|CCZ\rangle$ factory can quintuple the speed of algorithms that are dominated by the cost of applying Toffoli gates, including Shor's algorithm [31] and the chemistry algorithm of Babbush et al. [1]. Assuming a physical gate error rate of 10^{-3}, our CCZ factory can produce $\sim 10^{10}$ states on average before an error occurs. This is sufficient for classically intractable instantiations of the chemistry algorithm, but for more demanding algorithms such as Shor's algorithm the mean number of states until failure can be increased to $\sim 10^{12}$ by increasing the factory footprint $\sim 20\%$.

1 Introduction

In fault-tolerant quantum computation based on the surface code (a likely component of future error corrected quantum computers due to the surface code's comparatively high threshold and planar connectivity requirements [3, 11, 29, 30, 13]), the cost of a quantum algorithm is well approximated by the number of non-Clifford operations. This is due to the fact that non-Clifford operations are performed via magic state distillation [5], and the cost of state distillation is large. For example, the spacetime volume (qubit-seconds) of the T state factory from [15] is two orders of magnitude larger than the volume of a CNOT operation between adjacent qubits [19]. The non-Clifford gate count will likely be particularly significant for the earliest error corrected quantum computers, which will not have enough space to distill magic states in parallel.

Over the past decade, thanks to techniques such as block codes [4, 16], bridge compression [14], and many others [19, 8, 9, 25], the cost of magic state distillation has steadily decreased. This paper adds catalyzed phasing to the pile of known techniques, continuing the tradition of gradually chipping away at the convenient approximation that magic states are the dominant cost in error-corrected quantum computation.

Note that, in this paper, we focus on optimizing the cost of distillation in the *single-factory regime*. For example, we do not investigate whether there are block code factories that can use catalyzation. We focus on the single-factory regime because we are interested in estimating the minimum number of physical qubits needed to run classically intractable instances of various quantum algorithms at a reasonable rate, and the single-factory regime is the relevant one for these kinds of estimates. In Figure 1 and Figure 2, we give a high-level view of this paper's improvements in footprint and spacetime volume, over previous factories in the single-factory regime.

The paper is organized as follows. Section 1 provides an overview of the paper, and explains the various notation and diagram conventions we will be using. In Section 2, we explain how to construct an efficient $|CCZ\rangle$ factory by applying the techniques of [15] to the construction of [21, 12]. In Section 3, we construct a circuit which can transform a $|CCZ\rangle$ state into two $|T\rangle$ states if a catalyst $|T\rangle$ state is present. In Section 4, we show that this catalyzed circuit generalizes to other phase angles and note that this generalized circuit can produce two $|\sqrt{T}\rangle$ states using only five $|T\rangle$ states. In Section 5, we combine constructions from the previous sections into an efficient $|T\rangle$ factory. Finally, in Section 6, we discuss applications of our constructions,

Craig Gidney: craiggidney@google.com

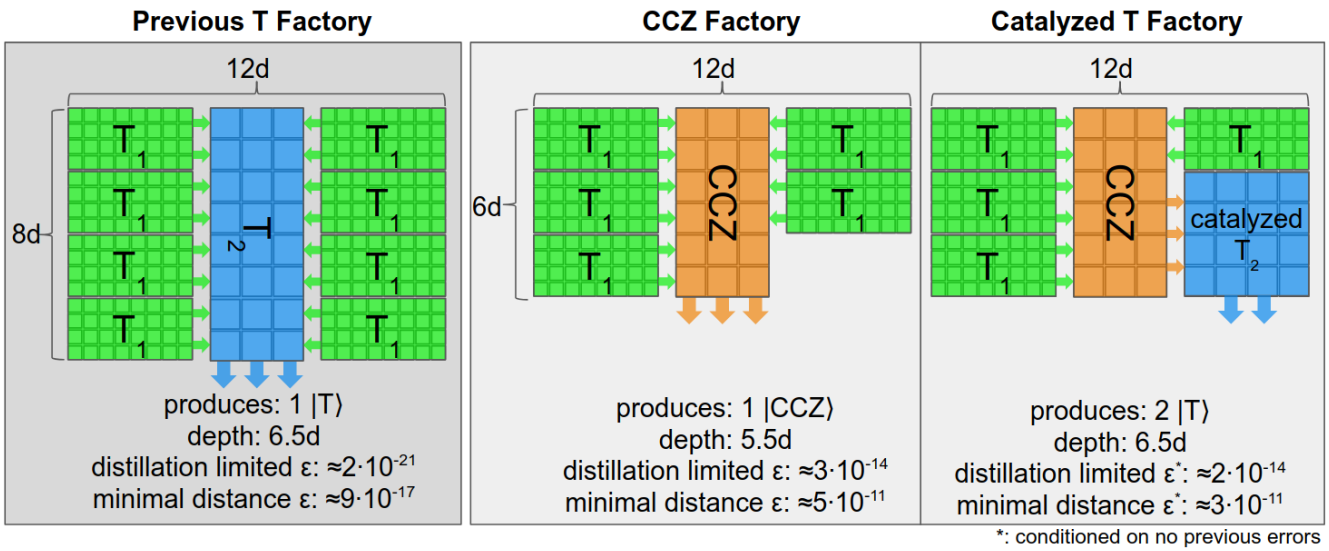

Figure 1: Overview of the spatial layout and data flow of the $15|T\rangle \overset{35\epsilon^3}{\rightarrow} |T\rangle$ factory construction from [15] (left), our $8|T\rangle \overset{28\epsilon^2}{\rightarrow} |CCZ\rangle$ factory construction (middle), and our $|T\rangle$-catalyzed $8|T\rangle \overset{28\epsilon^2}{\rightarrow} 2|T\rangle$ factory (right). The level 1 T factories (green) are effectively the same as in [15], and are performed at half code distance to balance the contributions from distillation error and code error. The distillation limited error rates assume a physical gate error rate of 10^{-3}, that the injection technique of [24] can create level 0 $|T\rangle$ states with approximately that level of error, and that the code distance is large enough for the dominant source of error in the outputs to be distillation error. The minimal distance error rates include the effects of topological errors in the surface code itself, with a code distance of 7 for level 0 $|T\rangle$ state injection, code distance 15 for level 1 factories, and code distance 31 for everything else. The factory from [15] has significantly better error suppression, but the amount of suppression is overkill unless one wants to run century-long computations without a single error. Our factories have smaller footprints, faster output, and an amount of suppression sufficient to run proposed algorithms beyond the classically simulable regime (e.g. [1]). The error rates of the catalyzed T factory have asterisks because its errors are correlated: if one error occurs it can poison the catalyst state and cause many more errors. This means that this factory should be used in contexts where a single error is already considered a complete failure (e.g. at the level of an entire algorithm, not as an input to further distillations).

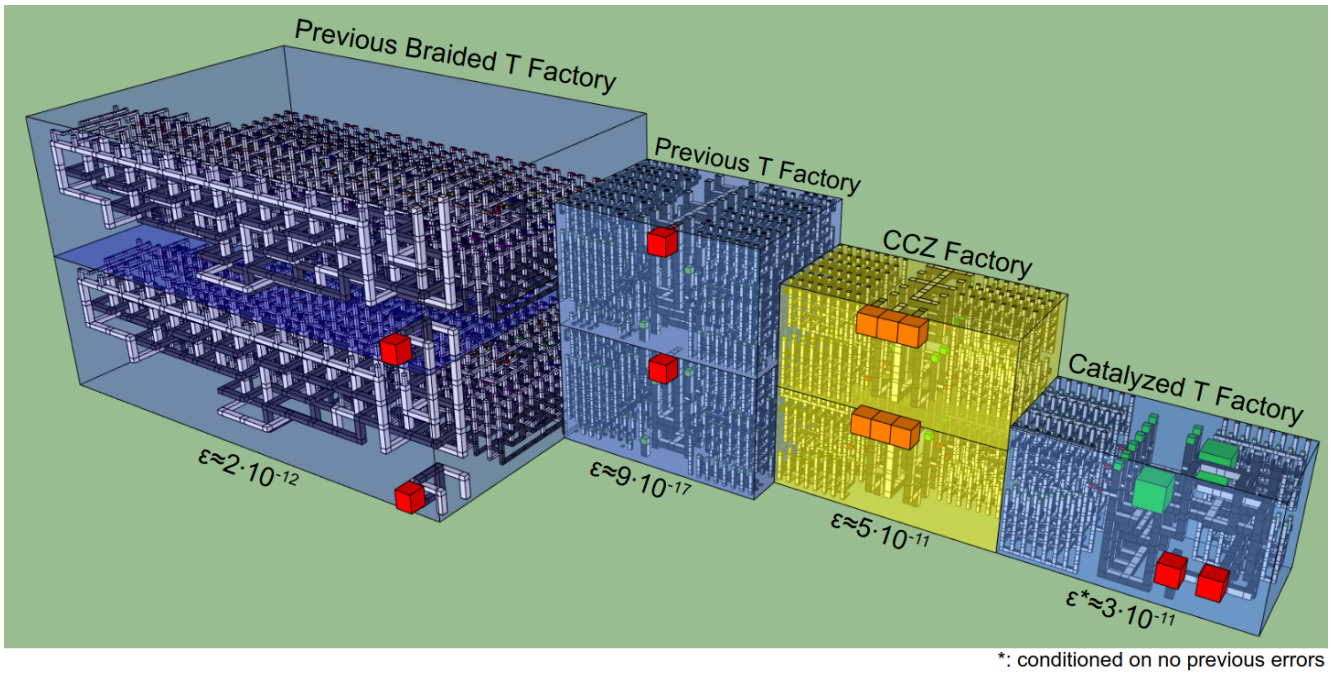

Figure 2: Size comparison of various factories producing two magic states, including output error rates. Error rates are computed assuming a physical gate error rate of 10^{-3}, and include topological errors from the surface code itself. Includes the $15|T\rangle \overset{35\epsilon^3}{\rightarrow} |T\rangle$ factory construction using braids [14] (left) and lattice surgery [15] (middle left), as well as our $8|T\rangle \overset{28\epsilon^2}{\rightarrow} |CCZ\rangle$ factory construction (middle right) and our $|T\rangle$-catalyzed $8|T\rangle \overset{28\epsilon^2}{\rightarrow} 2|T\rangle$ factory (right). $|T\rangle$ output events are indicated with red cubes. $|CCZ\rangle$ output events are indicated with a triplet of orange cubes. The braided factory has been scaled to account for the fact that it uses the unrotated surface code instead of the rotated surface code [19]. The braided T factory's error rate is significantly higher because it uses an older injection technique, resulting in the level 0 T gates having an error rate of 10^{-2} instead of $2 \cdot 10^{-3}$. The error rate of the catalyzed T factory has an asterisk because its errors are correlated: if one error occurs it can poison the catalyst state and cause many more errors. This means that this factory should be used in contexts where a single error is already considered a complete failure (e.g. at the level of an entire algorithm, not as an input to further distillations).

ALGORITHM PARAMETERS		Chemistry N=54	Chemistry N=250	Shor 1024	Shor 4096 d1=19
T count:	1.00E+08	1.00E+05	1.00E+07	5.12E+04	2.05E+05
Toffoli count:	1.00E+08	4.47E+06	2.63E+08	1.29E+10	8.25E+11
Qubit count:	100	123	341	3082	12298
Error budget:	1.00%	2.00%	2.00%	50.00%	50.00%
DEVICE PARAMETERS					
Characteristic physical gate error rate:	0.10%	0.10%	0.10%	0.10%	0.10%
Surface code cycle time (microseconds):	1	1	1	1	1
ADVANCED PARAMETERS					
L1 distillation code distance (d1):	15	15	15	15	19
L2 distillation code distance (d2):	31	31	31	31	31
Routing overhead	50.00%	50.00%	50.00%	50.00%	50.00%
OUTPUT					
Chance of failure:	**0.84%**	**1.45%**	**1.66%**	**78.88%**	**41.11%**
Physical qubit count:	**4.00E+05**	**3.43E+05**	**1.01E+06**	**over error budget**	**4.54E+07**
Duration (hours):	**7.53E+00**	**2.14E-01**	**1.27E+01**	**over error budget**	**4.91E+04**

Figure 3: Screenshot of the resource estimation spreadsheet include in the supplementary materials of this paper (file name "calculator-CCZ-2T-resources.ods"), with various interesting cases pre-entered. Assuming a physical gate error rate of 10^{-3}, and minimal code distances, the $|CCZ\rangle$ factory is unlikely to fail when producing on the order of 10^{10} states. This is sufficient to run classically intractable chemistry algorithms [1], but not quite sufficient to factor a 1024 bit number with a 50% success rate (assuming that factoring an n bit number requires $12n^3$ Toffoli gates and $3n$ space [32]). However, if the physical gate error rate is improved slightly or (more plausibly) the factory is made slightly larger by increasing the level 1 code distance from 15 to 19, then the number of states that can be produced increases to be on the order of 10^{12}. This allows 4096 bit numbers to be factored (though we do not recommend using a single factory for this task, since it would take 5 years to produce the necessary magic states assuming a surface code cycle time of 1 microsecond).

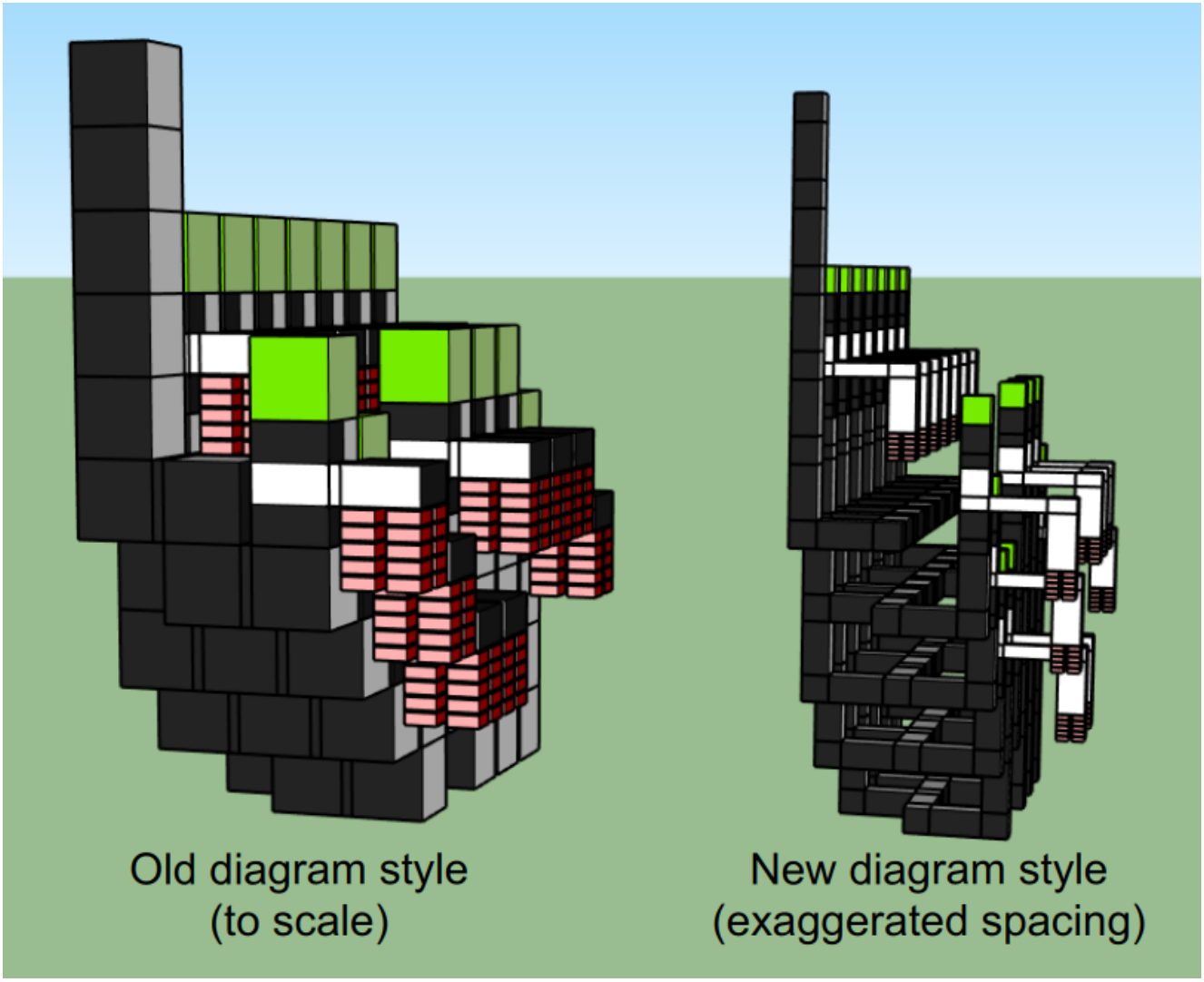

Figure 4: Comparison of the to-scale diagram style from [15] with the exaggerated-spacing diagram style used by this paper. The to-scale diagram style emphasizes how things fit together and is ideal when reasoning geometrically. The exaggerated-spacing diagram style emphasize how things connect together and is ideal when reasoning topologically. An even more abstract diagram style for lattice surgery is the ZX calculus [10]. In Figure 9 we show how translate a topological diagram into a ZX calculus graph.

summarize our contributions, and point towards future work.

In this paper we will refer to factories using the notation "$|\text{In}\rangle \overset{f(\epsilon)}{\rightarrow} |\text{Out}\rangle$ factory". The left hand side is the state input into the factory, the right hand side is the state output from the factory, and the function above the arrow indicates the amount of error suppression up to leading terms (i.e. the $f(\epsilon)$ above the arrow is shorthand for the true suppression $f(\epsilon) + O(\epsilon f(\epsilon))$). For example, we will refer to the $|T\rangle$ state distillation based on the 15-qubit Reed-Muller code [5] as the $15|T\rangle \overset{35\epsilon^3}{\rightarrow} |T\rangle$ factory.

We use three main types of diagram in this paper: circuit diagrams, time slice diagrams, and 3D topological diagrams. The circuit diagrams demonstrate the functionality that the 3D topological diagrams are supposed to be implementing, and the time slice diagrams are a sequence of slices through the 3D topological diagrams, showing boundary information and which patches are being merged or split. We often provide multiple diagrams of the same construction, with common labelling between the diagrams. For example, Figure 5, Figure 7, and Figure 8 are all diagrams of our CCZ factory. Discerning readers can use the labels common to all three diagrams to verify that they agree with each other. In particular, those three diagrams all have three output qubits labelled 1 through 3, eight ancillae qubits labelled a through h, and four "stabilizer qubits" labelled by the stabilizer measurement they correspond to.

To make it possible to see the internal topological structure of the 3D topological diagrams, we have chosen to significantly exaggerate the amount of space between events. We draw operations as if they had linear $O(d)$ separation (where d is the code distance), but on actual hardware the operations have a constant $O(1)$ separation. This exaggeration of separation does not change the topology, so interpreting the figures as if they were to scale will still produce the correct computation. But it is important to account for the distortion when computing the footprint or depth of the computation. Figure 4 shows a comparison between the old to-scale diagram style and our new exaggerated spacing style diagrams.

We will sometimes refer to multi-qubit stabilizers using a concatenated-subscript notation such as Z_{123}. Each subscript refers to a separate qubit, i.e. $Z_{123} = Z_1 Z_2 Z_3$.

We will often refer to $|T\rangle$ states as having a particular "level", e.g. "a level 1 $|T\rangle$ state" or equivalently "a $|T_1\rangle$ state". The level refers to the number of distillation steps used to produce the state. We will also refer to factories by the level of their output. For example, our starting point is level 0 $|T\rangle$ states produced using the post-selected state injection of Li [24]. These $|T_0\rangle$ states are then distilled by the level 1 T factory from [15] into $|T_1\rangle$ states, which we can then feed into our $|CCZ\rangle$ factory.

Lastly, we wish to point out the useful supplementary materials included with this paper. First, because it is significantly easier to understand 3D diagrams when one is able to move the camera, the supplementary materials include SketchUp files storing the models shown in the 3D topological diagrams. Second, the supplementary materials include a spreadsheet (file name "calculator-CCZ-2T-resources.ods") that can compute the overhead of computations that use our factories. Interested readers can estimate the running time and number of physical qubits required by their algorithms by entering into the spreadsheet the number of T and Toffoli gates performed by the algorithm, how many qubits the algorithm uses, and an error budget. Figure 3 shows a screenshot of the spreadsheet.

2 Lattice surgery construction of the $8|T\rangle \overset{28\epsilon^2}{\rightarrow} |CCZ\rangle$ factory

A key technique introduced in [15] is a single-layer stabilizer measurement involving an arbitrary numbers of qubits. We use this technique in order to quickly measure the 4 stabilizers of the error-detecting Toffoli distillation protocol [21, 12]. See Figure 5 for a circuit diagram of the CCZ-distillation process. The operations in the circuit are chosen in a way that trivially translates into lattice surgery. In Figure 6 we show time slices of one possible translation of the circuit into lattice surgery (with matching qubit labels and operation labels), and then in Figure 7 show the time slices of our CCZ factory (corresponding to two interleaved translations of the circuit). We also provide an annotated 3D topological diagram of the CCZ factory (see Figure 8).

Our $|CCZ\rangle$ factory has a naive depth of 4 (stabilizer measurements) + 1.5 (T state injections) + 1 (X or Y basis measurement, depending on T injection measurements) + 2 (detect errors) = 8.5. We use the same technique as in [15] to partially overlap executions of the factory, resulting in an effective depth of 5.5. The T state injections take $1.5d$ layers because they are performed at half code distance and it takes $0.5d$ layers to move the black side into position for a parity measurement, then $0.5d$ layers to perform the parity measurement, then $0.5d$ layers to return the black side to its original position. It is acceptable to inject at half code distance because the incoming T states have an error rate larger than the topological error incurred from an injection at this distance.

Our $|CCZ\rangle$ factory produces magic states fast enough that algorithms will bottleneck on routing instead of

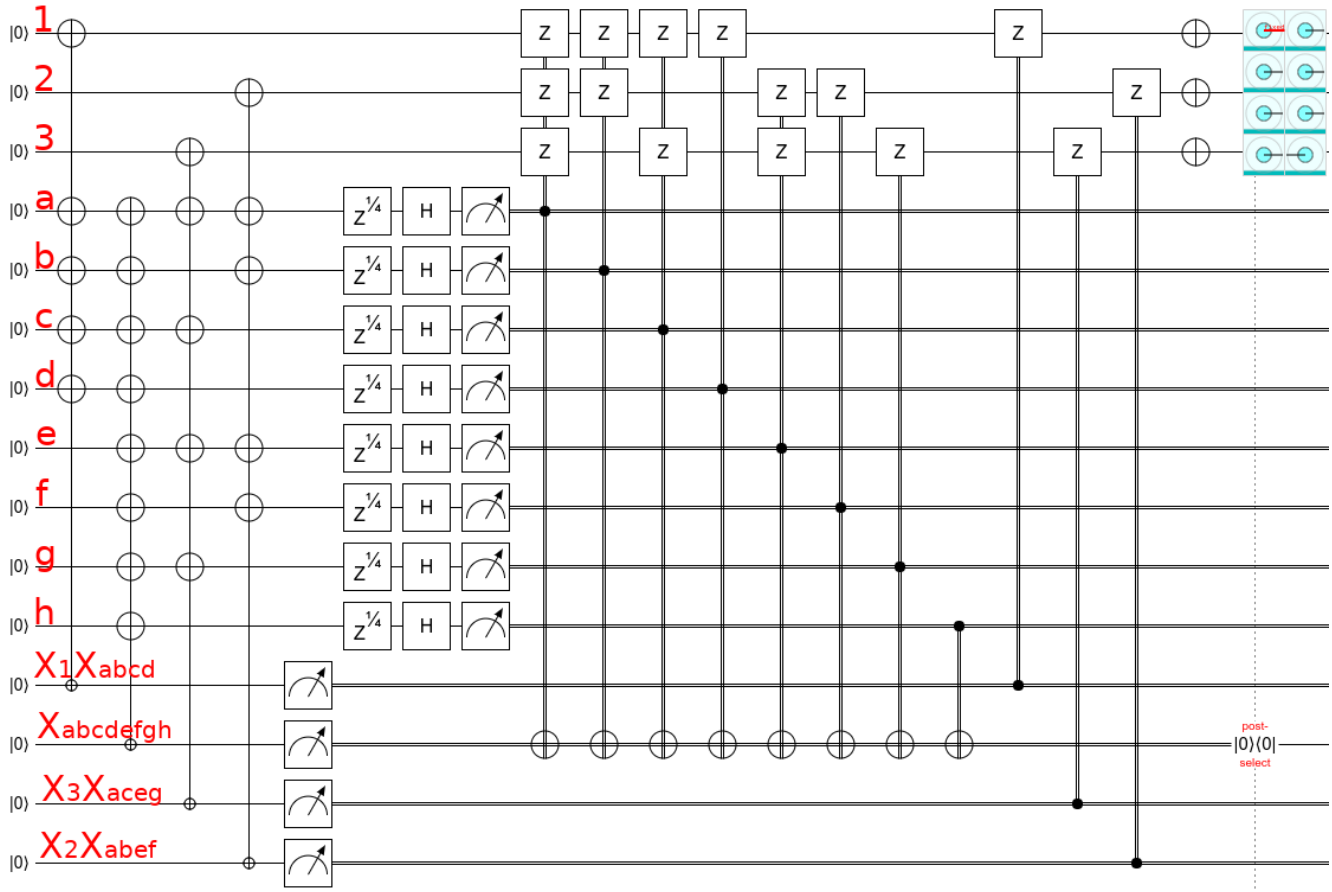

Figure 5: Quantum circuit for the $8|T\rangle \xrightarrow{28\epsilon^2} |CCZ\rangle$ factory. A rewrite of figure 3 from [21]. The box with blue circles in the top right is a state display from the online simulator Quirk, with each circle representing an amplitude (the radius of the colored circle indicates the amplitude's magnitude, and the angle of the line rooted at the center of the circle indicates the phase). The state display is showing that the output state is a $|CCZ\rangle$ state. The small circled pluses in the circuit are X-axis controls (equivalent to a normal control surrounded by Hadamard gates); whenever one of these controls directly precedes a measurement the measurement corresponds to a Pauli product measurement. The post-selection operation represents the classical control software determining if the an error was detected; if it fails the output must be discarded. Pauli operations and classically-controlled Pauli operations appear here, but not in Figure 8, because they are performed entirely within classical control software. The circuit can be opened in Quirk by following this link. Discerning readers can follow the link and edit the circuit in order to confirm that adding a single Z error by any T gate is caught by the post-selection, and also that all possible pairs of Z errors escape detection.

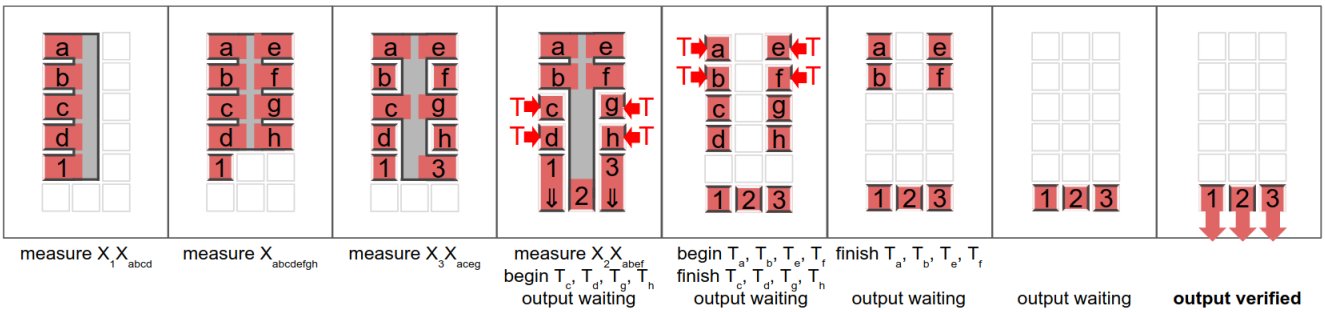

Figure 6: Time slices of lattice surgery activity during production of a single $|CCZ\rangle$ state. Each red square corresponds to a qubit, and the label inside the red square identifies the qubit from Figure 5 that the square corresponds to. Gray rectangles correspond to X stabilizer measurements between sets of qubits. The red arrows labelled "T" correspond to a noisy T state entering the system. It is possible to double the throughput shown here by interleaving the production of two states (shown in Figure 7).

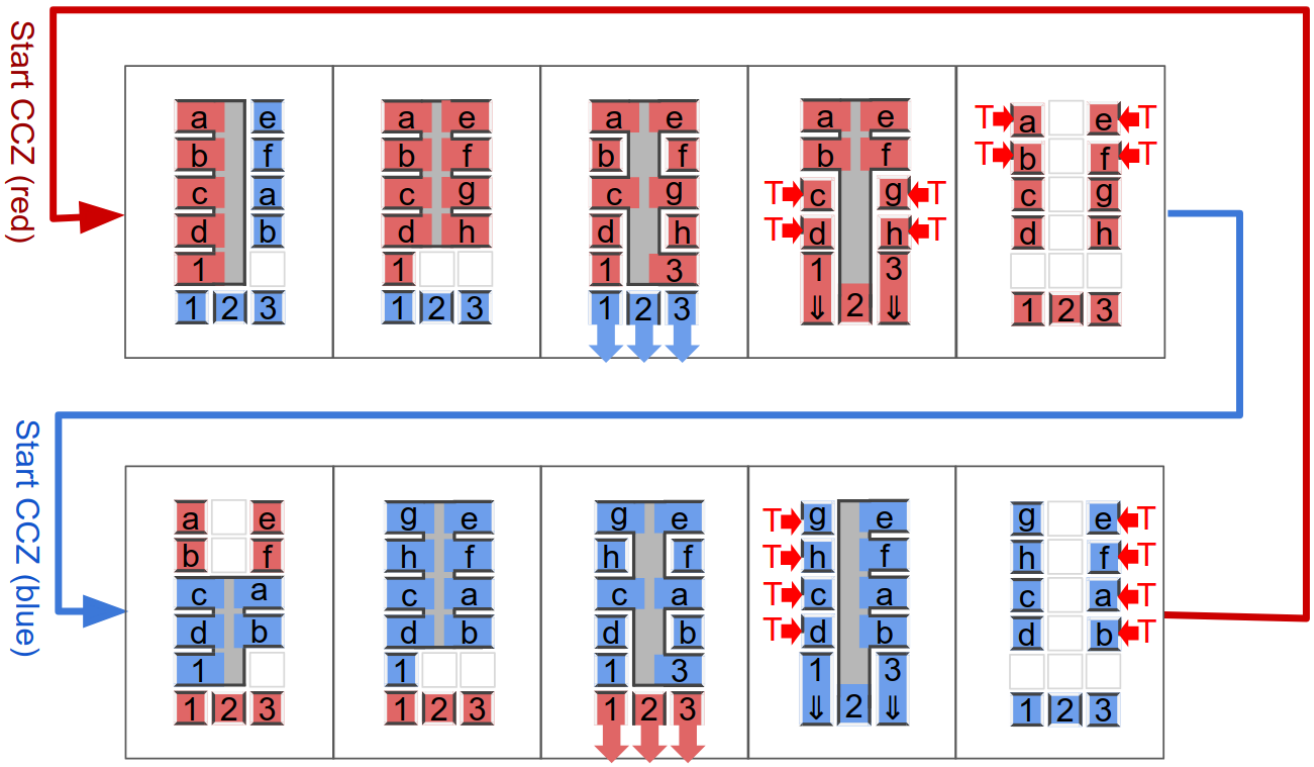

Figure 7: Time slices of lattice surgery activity during production of $|CCZ\rangle$ states by the $8|T\rangle \xrightarrow{28\epsilon^2} |CCZ\rangle$ factory. To maximize utilization, two states are produced concurrently. Each red or blue square corresponds to a qubit, and the label inside the square identifies the qubit from Figure 5 that the square corresponds to. Gray rectangles correspond to X stabilizer measurements between sets of qubits. The red arrows labelled "T" correspond to a noisy T state entering the system. Blue squares correspond to qubits involved in producing one of the states, and red squares correspond to qubits involved in producing the other state. The red squares in each step are exactly identical to the red squares shown in the matching step of Figure 6. See Figure 8 for a 3D topological diagram corresponding to the time slices.

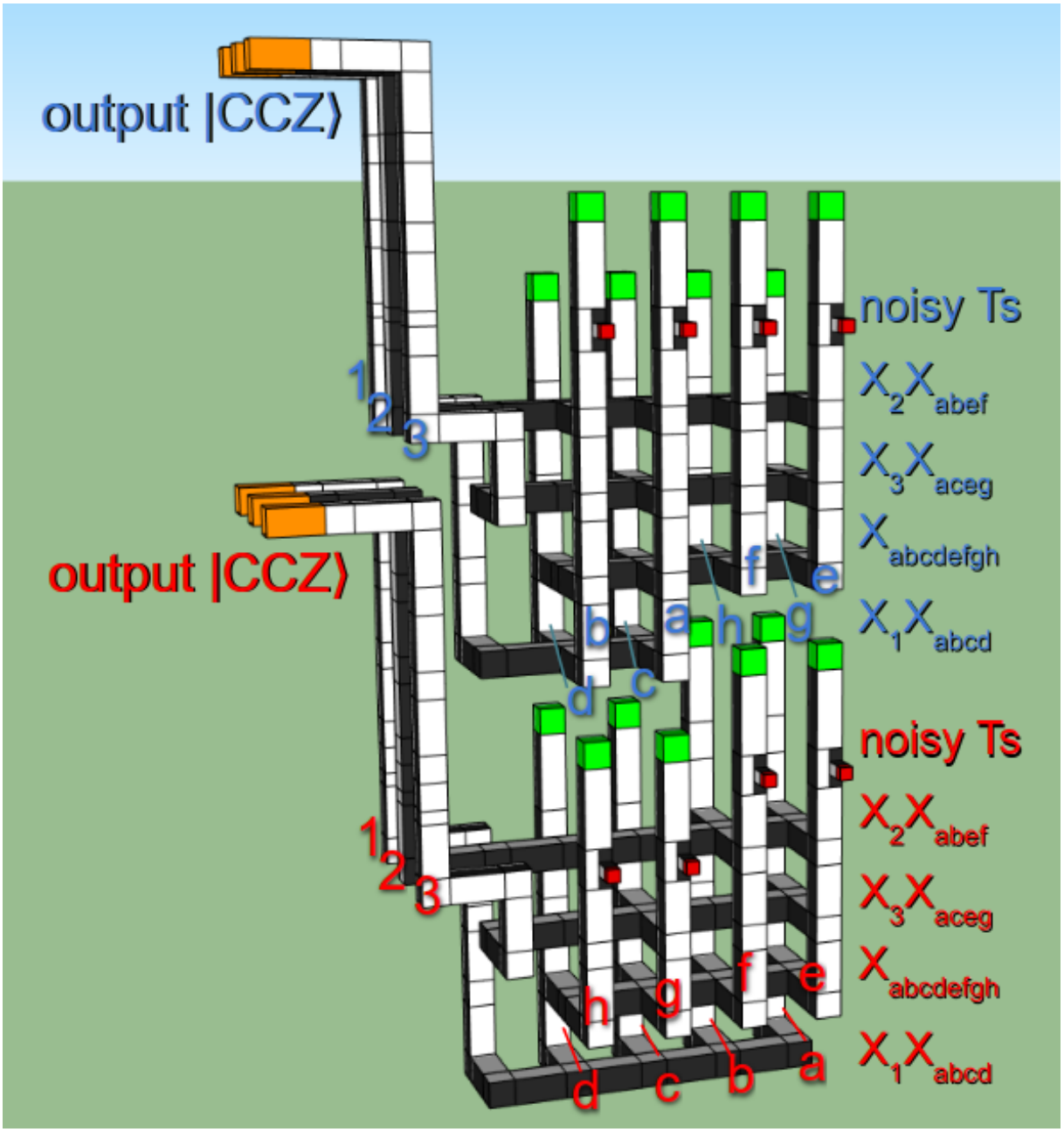

Figure 8: 3D topological diagram for our construction of the $8|T\rangle \xrightarrow{28\epsilon^2} |CCZ\rangle$ factory. The spacing between qubits has been increased to make it possible to see the internal structure. White (black) surfaces correspond to boundaries where chains of Z (X) errors can terminate. Corresponds to the time slices from Figure 7. Time increases from bottom to top. The vertical poles correspond to qubits from Figure 5. Red boxes indicate connection points for noisy $|T_1\rangle$ states produced by a level 1 T factory. The green boxes atop the columns are performing either an X or Y basis measurement at half code distance, as described in [15], by including or omitting an S gate performed using twists [6]. Using half code distance is acceptable because, at the location in the factory where these operations are performed (i.e. after the T injections), individual errors are detected as distillation failures. The labels along the right hand side indicate the stabilizer measurements occurring at each time. The red/blue coloring of labels matches the red/blue coloring of Figure 7. Note that inserting the $|T_1\rangle$ state has a depth of 1.5, unlike the other steps which have depth 1. Each horizontal bar linking several vertical poles is a stabilizer measurement of a product of logical X observables. The groups of three qubits highlighted orange and exiting left are the $|CCZ\rangle$ states being output (note that the middle pole of each $|CCZ\rangle$ state is rotated with respect to the others, with white on top instead of black on top). The two instances of the factory that are shown differ slightly. Their qubits have been permuted so that each factory's top layer fits into a void at the bottom of the following factory, saving a layer of depth.

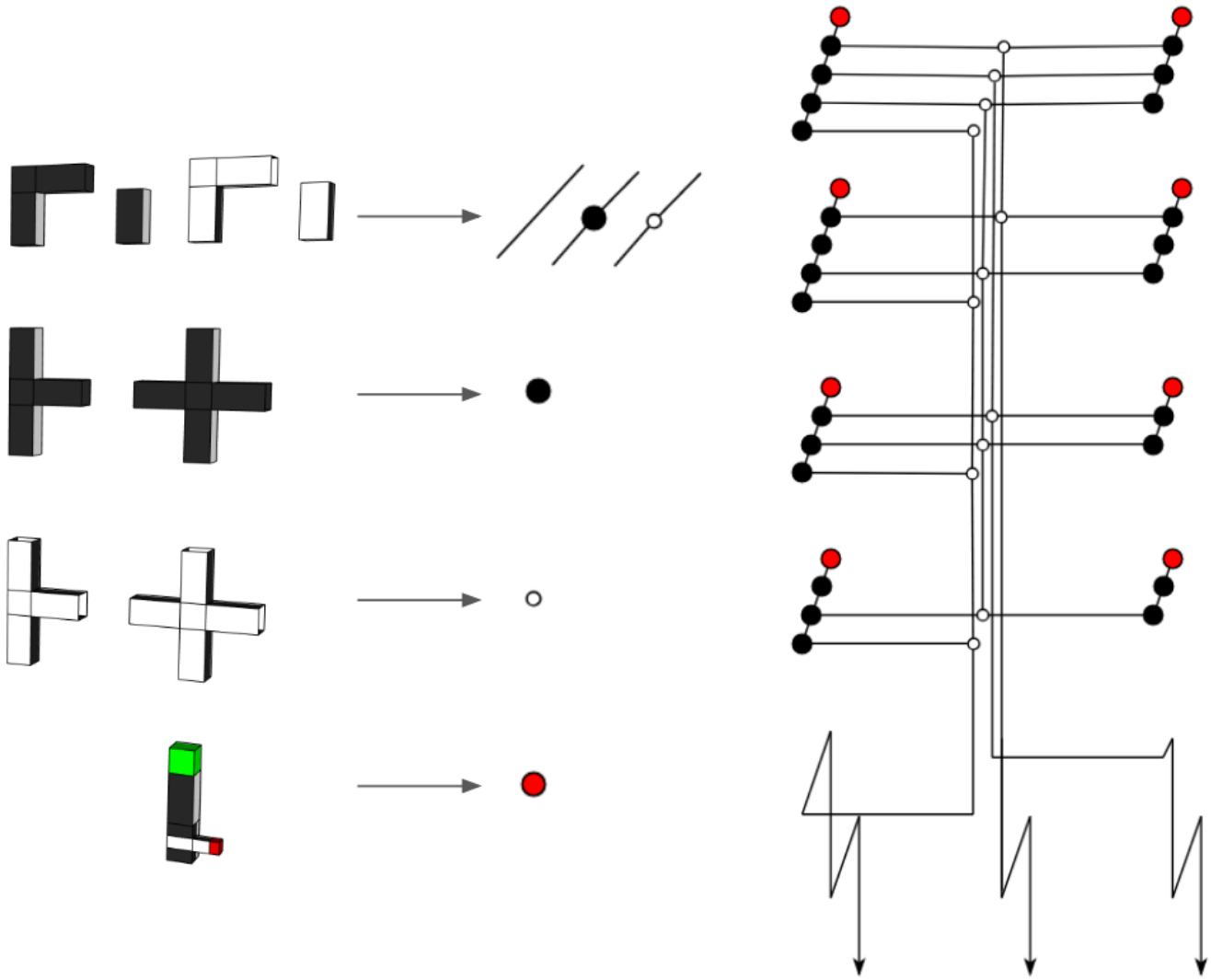

Figure 9: A substitution procedure (left) for translating our 3D topological diagrams into (nearly) the ZX calculus [10], as well as an example translation of one of the CCZ factories from Figure 8 (right). We use black (white) nodes instead of green (red) nodes (the usual notation for the ZX calculus) so that the node colors match the boundary colors in the 3D topological diagrams. Pieces with two ports are translated into edges or degree-2 nodes of either color. Pieces with three or more ports are translated into a node of matching color. The $Z \otimes Z$ measurement of a qubit vs a $|T\rangle$ state followed by measuring the qubit in the X or Y basis depending on the outcome of the parity measurement is translated into a (non-standard) red node. The red node can be expanded into a proper ZX calculus construction, but we do not attempt to do so. The ZX calculus graph is more amenable to verification than the 3D diagram. The reverse translation, from ZX calculus graph to 3D topological diagram, is often more difficult.

magic state production unless special care is taken. For example, suppose there are several Toffoli operations to perform on qubits all placed in a common area; a common area with exactly one entrance capable of allowing exactly one qubit to enter or leave every d cycles. Because a new $|CCZ\rangle$ state is produced every $5.5d$ cycles, and each such state involves three qubits, the entrance will be occupied for $3d$ out of every $5.5d$ cycles moving magic state qubits into the common area to meet target qubits (or vice versa). This leaves only $2.5d$ cycles for other work requiring the entrance. Furthermore, the $|CCZ\rangle$ teleportation process requires classically controlled CZ and CNOT operations. If these operations also block the entrance, and are not done in a way that minimizes depth, they will use up the remaining $2.5d$ cycles and cause a routing bottleneck.

We see three ways for algorithms to avoid bottlenecking on routing and keep up with our $|CCZ\rangle$ factory:

1. Increase the amount of space dedicated to routing. Play it safe; do not have areas with narrow entrances or hallways that can only accommodate one qubit per d cycles. This strategy is simple and effective, but costly.

2. Carefully distribute logical qubits across multiple disjoint areas with the goal of ensuring that Toffolis rarely target multiple qubits in the same area. This avoids the bottleneck by having the magic state qubits pass through multiple different entrances, instead of one common entrance. This strategy will not work for all algorithms, but it will work for some algorithms.

3. Use generalized CCZ operations capable of targeting arbitrary stabilizers instead of individual qubits, and move Clifford work into the classical control system. The generalized CCZ is performed in the same way that [25] performs generalized T gates targeting arbitrary stabilizers. The gate teleportation process is modified; replacing each $Z_t \otimes Z_m$ parity measurement between a target qubit t and the magic state qubit m with a many-body stabilizer measurement $P \otimes Z_m$ where P is a vector of Pauli operations possibly involving every logical data qubit in the computation. The main drawback of this approach is that there is 2x space overhead associated with ensuring it is always fast to access the X, Y, and Z observable of every qubit. This can likely be avoided by interleaving single-qubit work between the Toffoli operations, but requires careful algorithm-by-algorithm consideration.

Note that our CCZ factory's footprint includes an unused 2x4 area, adjacent to where the $|CCZ\rangle$ state exits the factory (see Figure 1). This area can be used to hold target qubits waiting for a Toffoli operation, which helps with the routing overhead. Our overhead spreadsheet assumes this space will be used in this manner.

In order to produce a $|CCZ\rangle$ state every $5.5d$ cycles, we need enough level 1 T factories to create 8 $|T\rangle$ states every $5.5d$ cycles. The half-code-distance level 1 T factory from [15] produces a $|T\rangle$ state every $3.25d$ cycles, except when distillation errors are detected. Assuming a physical gate error rate of 10^{-3} and a level 1 code distance of 15, distillation errors will be detected approximately 3% of the time (the $|T_0\rangle$ states have $\sim 10^{-3}$ error when injected, gain $\sim 10^{-3}$ error while the level 0 T gates are performed at distance 7, there are fifteen of them, and the most likely case is that a single one fails: $2 \cdot 10^{-3} \cdot 15 = 3\%$). These failures reduce the effective output rate to a $|T\rangle$ state every $3.35d$ cycles, so five of these factories will produce ~ 8.2 $|T\rangle$ states every $5.5d$ cycles, which is sufficient to keep up with the $|CCZ\rangle$ factory. We accumulate a buffer of surplus level 1 $|T\rangle$ states in the small hallways between the $|CCZ\rangle$ factory and the level 1 $|T\rangle$ factories so that a single level 1 T factory failure does not delay the entire $|CCZ\rangle$ factory. As shown in Figure 1, the five level 1 factories are placed to either side of the $|CCZ\rangle$ factory. Note that it is occasionally necessary to route the fifth factory's output to the opposite side, and that there is enough contiguous unused volume in the factory to do this when needed.

We compute the error rate of the $|CCZ\rangle$ states being produced by our factory in two different regimes: the large code distance regime where the factory is distillation limited, and the minimal code distance regime where the factory may be limited by topological errors in the surface code. We assume a physical gate error rate of 10^{-3} in both cases, and assume that the post-selected state injection of Li [24] creates $|T_0\rangle$ states with approximately this probability of error. In the distillation limited regime, we run these states through the $15|T\rangle \overset{35\epsilon^3}{\rightarrow} |T\rangle$ factory and then through our $8|T\rangle \overset{28\epsilon^2}{\rightarrow} |CCZ\rangle$ factory producing intermediate $|T_1\rangle$ states with error rate $\sim 3.5 \cdot 10^{-8}$ and then $|CCZ\rangle$ states with error rate $\sim 3.4 \cdot 10^{-14}$. In the minimal code distance regime, we must account for topological error introduced while performing T gates and the Clifford operations making up the factory. For example, we assume that the error rate of the $|T_0\rangle$ states doubles while performing a level 0 T gate at distance 7. This increases the effective error of the $|T_1\rangle$ states, but this contribution is overshadowed by the large size and proportionally small code distance of the level 1 T factory operating on these states. The factory adds approximately 10^{-6} error to the output error, which is three to four times more than the distillation error. We sum the two error rates, resulting in an estimated error rate for $|T_1\rangle$ states of $\sim 1.4 \cdot 10^{-6}$. This is forty times more error

than in the distillation limited case. The CCZ factory has a code distance large enough that we are distillation limited, and the error rate of the final $|CCZ\rangle$ states is correspondingly $\sim 5.3 \cdot 10^{-11}$.

As shown in figure Figure 3, the minimal distance factory causes errors in more than 50% of runs when attempting to factor a 1024 bit number, but can comfortably run classically intractable chemistry algorithms. However, if one increases the level 1 code distance from 15 to 19 (increasing the footprint of the factory by roughly 20%), then the level 1 error improves so much that it's possible to factor 4096 bit numbers.

3 The $|T\rangle$-catalyzed $|CCZ\rangle \rightarrow 2|T\rangle$ factory

In [21], it is shown how to perform a Toffoli gate by using Clifford operations, measurement, and four T gates. That circuit can be rewritten into an inline circuit that transforms three $|+\rangle$ states into a $|CCZ\rangle$ state via Clifford operations and four T gates [17]. Then, by diagonalizing that circuit's stabilizer table, said circuit can be rewritten into a form where three of the T gates apply directly to an input $|+\rangle$ state. Those three T gates can then be replaced by three $|T\rangle$ state inputs, resulting in a circuit that maps $|T\rangle^{\otimes 3}$ to $|CCZ\rangle$ using Clifford gates and one T gate. This circuit contains no measurement, and therefore can be inverted. The inverse circuit (shown in Figure 10) maps a $|CCZ\rangle$ state to three $|T\rangle$ states using Clifford gates and one T gate.

Because $|T\rangle$ states can be used to perform T gates, the T gate used to transform the $|CCZ\rangle$ into three $|T\rangle$ states can be powered by a $|T\rangle$ state output from a previous iteration of the circuit. If we keep feeding a $|T\rangle$ state output from iteration k into iteration $k+1$, then we effectively have a circuit that takes a $|CCZ\rangle$ state and outputs two $|T\rangle$ states. Under this interpretation of the circuit, the third $|T\rangle$ state is an ancillary state that is necessary for the transformation to be possible, but is not consumed by the transformation. Thus, in keeping with terminology for ancillary states that enable LOCC communication tasks without being consumed [20], and previous work [7], we refer to the third $|T\rangle$ as a catalyst. We refer to the circuit as a whole as the $|T\rangle$-catalyzed $|CCZ\rangle \rightarrow 2|T\rangle$ factory, or "C2T factory" for short.

Beware that, although the catalyst $|T\rangle$ state is not consumed by the C2T factory, it does accumulate noise from the incoming $|CCZ\rangle$ states. If a catalyst $|T\rangle$ has cycled through n iterations of the C2T factory, and there is a probability ϵ of each $|CCZ\rangle$ containing an error, then there is an $\Theta(n\epsilon)$ chance that the catalyst has been poisoned and is causing the factory to produce bad outputs. However, because every error in the catalyst ultimately traces back to an error in a $|CCZ\rangle$ state, the chance of there being *any* error grows like $\Theta(n\epsilon)$, instead of $\Theta(n^2\epsilon)$ as would be expected from a naive calculation assuming uncorrelated errors.

Distillation protocols usually require inputs with uncorrelated errors, so it is important that we only use the C2T factory as the last step in a distillation chain. In a sense, because of how we use the C2T factory, the correlation between errors is beneficial to us instead of detrimental. It means that when we run an algorithm many times there will be a small number of runs with many errors, instead of many runs with a small number of errors. We experience quadratically fewer whole-algorithm failures than would be expected from the fact that the expected number of errors is growing like $\Theta(n^2\epsilon)$. For other examples of correlation between errors being beneficial, we recommend reviewing hat guessing games [28].

The C2T factory circuit shown in Figure 10 is compact, but not in an ideal form for embedding into lattice surgery. Figure 11 fixes this by providing an equivalent circuit that, although it appears much more complicated, trivially translates into lattice surgery. We show the result of this translation in Figure 12, which has time slices of the lattice surgery operations occurring as the factory operates. And finally Figure 13 shows an annotated 3D topological diagram of the process.

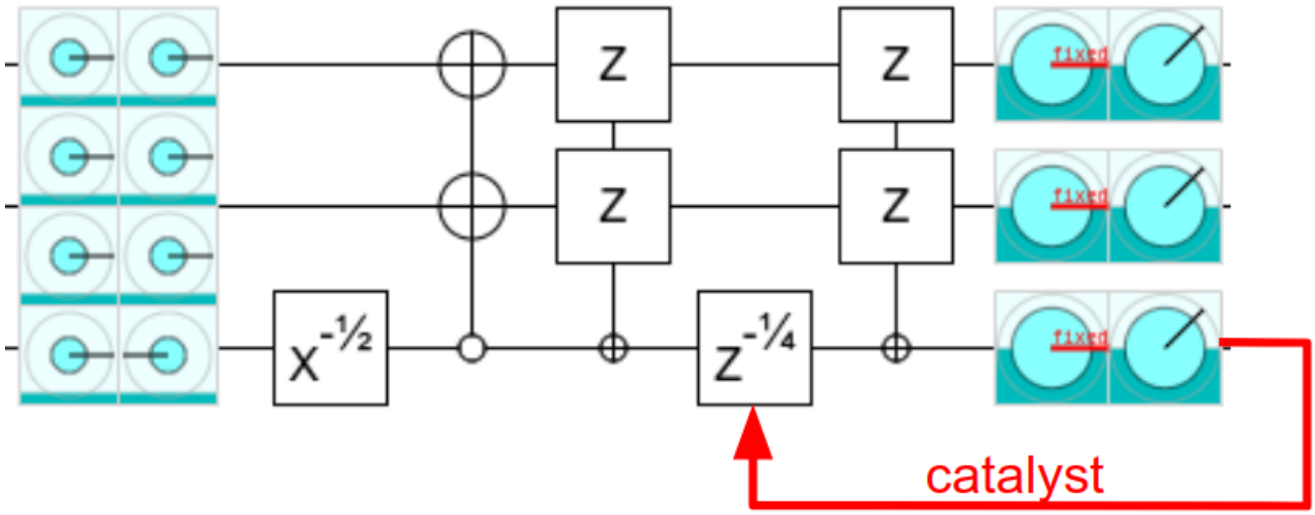

Figure 10: A circuit that transforms a $|CCZ\rangle$ state into three $|T\rangle$ states by applying Clifford operations and a single T gate. By using one of the outputs to fuel the next iteration, the circuit can be re-interpreted as a circuit that turns one $|CCZ\rangle$ into two $|T\rangle$ states when catalyzed by one $|T\rangle$ state. The boxes with blue circles are state displays from the online simulator Quirk, with each circle representing an amplitude (the radius of the colored circle indicates the amplitude's magnitude, and the angle of the line rooted at the center of the circle indicates the phase). The state displays are showing that the input state is a $|CCZ\rangle$ and the output states are $|T\rangle$ states. The small circled pluses are X-axis controls (equivalent to a normal control surrounded by Hadamard gates). The circuit can be opened in Quirk by following this link.

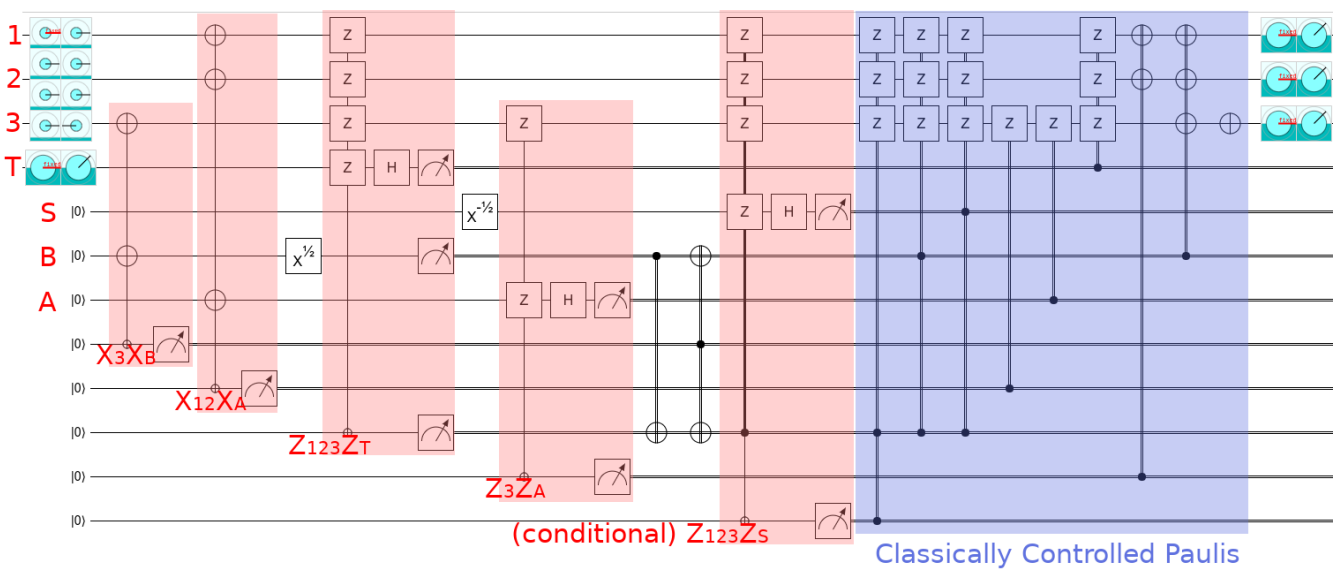

Figure 11: A circuit for catalyzed $|T\rangle$ state production, specialized for lattice surgery. Given a $|CCZ\rangle$ state (first three qubits) and a $|T\rangle$ state (fourth qubit), produces three $|T\rangle$ states. Red areas correspond to a product-of-Paulis measurement. The blue area happens entirely within classical control software. The S ancilla is preparing an $|S\rangle$ state that can be used to correct the T gate teleportation used to perform the $Z^{-1/4}$ gate from Figure 10. The B ancilla is being used to perform the $X^{-1/2}$ gate from Figure 10. The A ancilla is being used to perform the multi-target CNOT from Figure 10. The boxes with blue circles, at the beginning and end of the circuit, are state displays from the online simulator Quirk. Each circle represents an amplitude (the radius of the colored circle indicates the amplitude's magnitude, and the angle of the line rooted at the center of the circle indicates the phase). The state displays are showing that the input and output states are $|CCZ\rangle$ and $|T\rangle$ states as described. The small circled pluses in the circuit are X-axis controls (equivalent to a normal control surrounded by Hadamard gates); whenever one of these controls directly precedes a measurement the measurement corresponds to a Pauli product measurement. The circuit can be opened in Quirk by following this link.

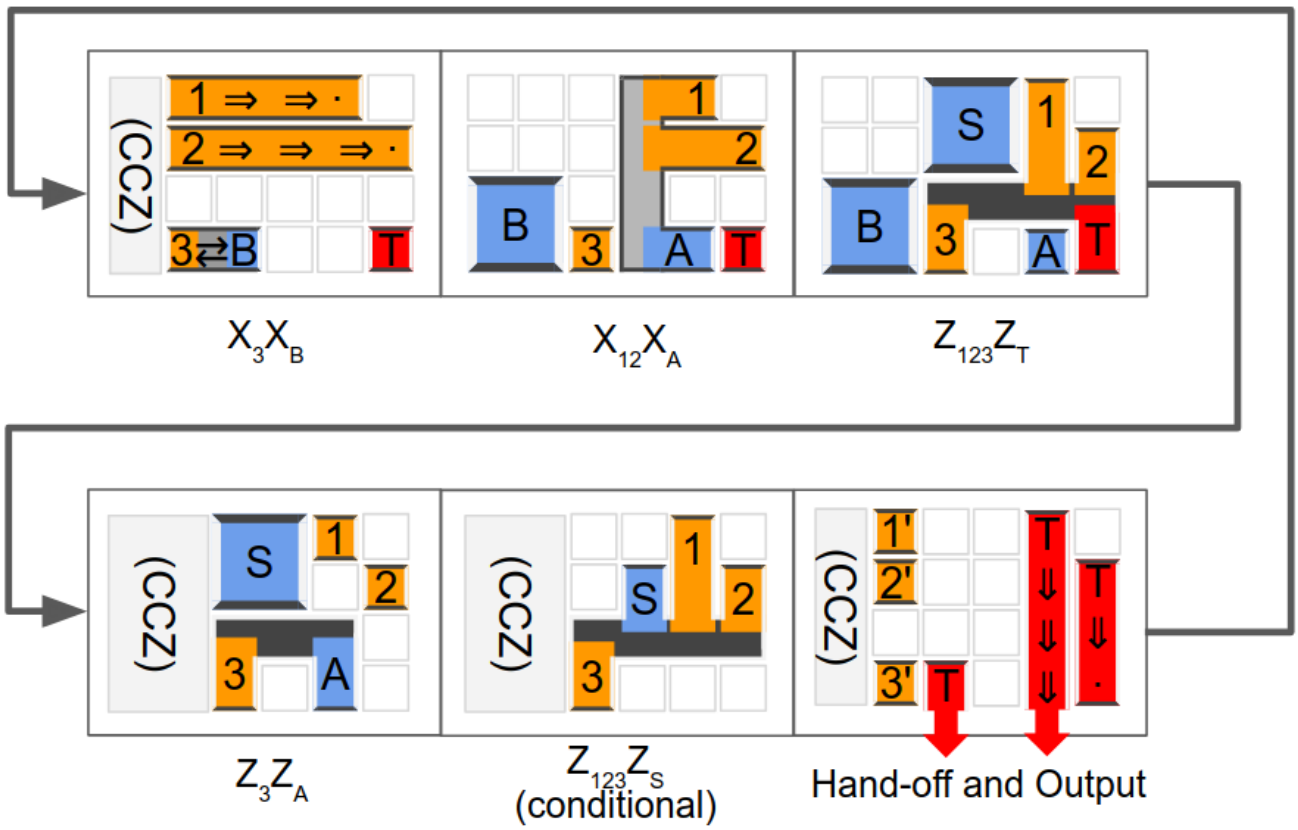

Figure 12: Time slices of lattice surgery activity during transformation of a $|CCZ\rangle$ state (orange qubits labelled 1, 2, 3) into three $|T\rangle$ states (shown in red in last slice), catalyzed by a $|T\rangle$ state (bottom right qubit in red). Black and dark gray bars correspond to stabilizer measurements. Ancillae qubits are shown in blue. The code distance of the ancillae qubits is doubled when single-qubit Clifford operations are being applied, to ensure there is sufficient suppression of errors. The light gray "(CCZ)" box to the left will be used by the CCZ factory producing $|CCZ\rangle$ states to be transformed. See Figure 13 for a 3D topological diagram corresponding to the time slices. Every step being performed can be matched up with a step from Figure 11, and the qubit labels shown here correspond to the qubit labels there.

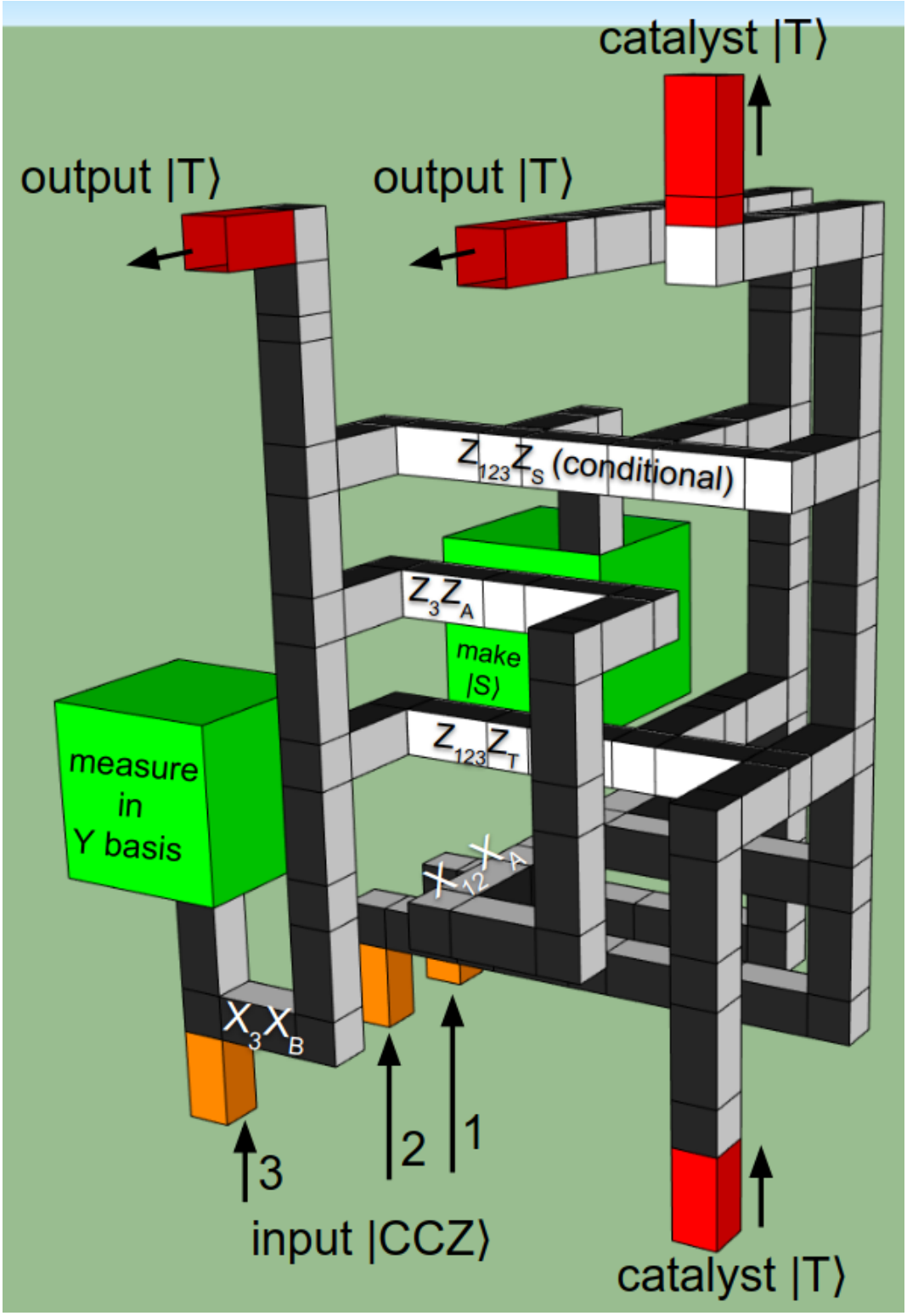

Figure 13: 3D topological diagram of a lattice surgery circuit transforming a $|CCZ\rangle$ state (orange-tipped inputs at bottom) and a $|T\rangle$ state (bottom right red-tipped input) into three $|T\rangle$ states (red-tipped outputs at top). We conservatively assume that the green boxes are large enough to perform any single-qubit Clifford with negligible error. See Figure 8 for details about how to interpret the diagram.

4 Arbitrary-Angle Phase Catalysis

The catalysis technique used in the C2T factory from the previous section generalizes to phasing angles other than the T gate's 45°. In Figure 14, we show a generalization of Figure 10 that works for an arbitrary angle θ. This circuit performs two Z^θ operations by performing cheap stabilizer operations, performing one Toffoli gate, performing one $Z^{2\theta}$ operation, and being catalyzed by one $Z^\theta|+\rangle$ state. Contrast with gate teleportation [18], which consumes a previously prepared $Z^\theta|+\rangle$ state in order to perform one Z^θ operation, with a 50% chance of requiring a fixup $Z^{-2\theta}$ operation.

One way to discover the generalized phase catalysis circuit is to start from the phase-gradient-via-addition circuit [22, 17, 27], which performs a series of rotations Z, S, T, $\sqrt{T}$, $\sqrt{\sqrt{T}}$, etc by adding a register containing the target qubits into a phase gradient catalyst state. Include a carry bit input in the addition of the phase-gradient-via-addition circuit, truncate the circuit after the first ripple-carry step by using the correct fixup operation, and the result is a phase catalysis circuit for an angle $\theta = \pi/2^k$ which trivially generalizes to arbitrary angles. The catalysis circuit can likely also be derived from synthillation parity-check circuits [9], which use similar magic states and have a similar structure but are used to perform distillation of existing states instead of producing additional states.

Specializing the generalized phase catalysis circuit to $\theta = 22.5°$, i.e. to the $\sqrt{T}$ gate, produces the circuit shown in Figure 15. This specialized circuit creates two $|\sqrt{T}\rangle$ states by performing one Toffoli operation and one T gate. This is significantly more efficient than previous techniques we were able to find and adapt to the task of producing $|\sqrt{T}\rangle$ states [23, 2, 26, 22, 17, 27], assuming a physical gate error rate of 10^{-3} and a target error rate of 10^{-10}. For example, according to figure 5 of [2], repeat-until-success circuits use ≈ 45 T gates to approximate a $\sqrt{T}$ gate to within precision $\epsilon = 10^{-10}$. As another example, according to table III of [26], direct synthesis of $\sqrt{T}$ state uses ≈ 25 times more volume than direct synthesis of $|T\rangle$ states (though this ratio improves as the physical gate error rate improves). A final example: the phase-gradient-via-addition operation described in [22, 17] can perform a $\sqrt{T}$ gate with a 4-bit adder (which requires 3 $|CCZ\rangle$ states). Phase-gradient-via-addition is the closest to competing with phase catalysis, which is perhaps not surprising since phase catalysis is an optimized form of this technique. Other techniques appear to be very far behind; requiring an order of magnitude more spacetime volume.

5 Lattice surgery construction of the $8|T\rangle \xrightarrow{28\epsilon^2} 2|T\rangle$ factory

We now combine the $|CCZ\rangle$ factory from Section 2 with the C2T factory from Section 3, producing a $|T\rangle$-catalyzed T factory that transforms eight noisy $|T\rangle$ states into two $|T\rangle$ states with quadratically less noise. Note that this means we achieve a 4:1 ratio of input $|T\rangle$ states to output $|T\rangle$ state, which is competitive with the 3:1 ratio of block codes [4]. This is surprising, because normally one has to work with a larger number of $|T\rangle$ states in order to achieve good ratios.

Note that we do not use exactly the same CCZ factory as in Section 2. We re-order the stabilizer measurements and place the output qubits in a different location, so that it fits into the C2T factory from Section 3. Furthermore, we do not bother interleaving the factory with itself anymore. There's no point; we need five T_1 factories to run at the rate achieved by interleaving but now only have four factories (recall Figure 1).

The details of the combined factory are covered in Figure 17, which shows the parallel operation of C2T factory and CCZ factory. Recall that the qubit labels can be matched up with Figure 5 for verification that the correct stabilizers are being measured (though in a different order). Our penultimate figure, Figure 18, shows a 3D topological diagram of the factory. Note that the figure omits the level 1 T factories feeding in noisy $|T_1\rangle$ states, and exaggerates the spacing between qubits in order to make internal structures visible, but is otherwise complete.

To bootstrap the factory, an initial catalyst $|T\rangle$ state is made "the hard way", using some less efficient $|T\rangle$ factory that can output $|T\rangle$ states with error no higher than the error rate of the $|CCZ\rangle$ factory. Bootstrapping occurs once at the start of the computation, and any time the catalyst $|T\rangle$ state is lost. Specifically, note that the $|CCZ\rangle$ state produced by the CCZ part of the factory is being consumed before it's known if it contained a distillation error. Therefore, when a detected distillation error does occur, the $|T\rangle$ state catalyst must be discarded. This has a negligible effect on the effective depth of the factory, because it occurs so rarely (approximately once per hundred thousand distillations). There is a space towards the top right of the factory where a spare $|T\rangle$ catalyst could be placed, to be used as a backup when the main catalyst is lost.

The primary bottleneck on the output of this factory is the rate at which $|T_1\rangle$ states are produced. As shown in Figure 1, we assume there are four $|T_1\rangle$ factories present (one beside each pair of qubits require $|T_1\rangle$ states. When functioning perfectly, each of these factories produces a pair of noisy $|T_1\rangle$ states every $6.5d$ cycles, which is just enough to feed the catalyzed T fac-

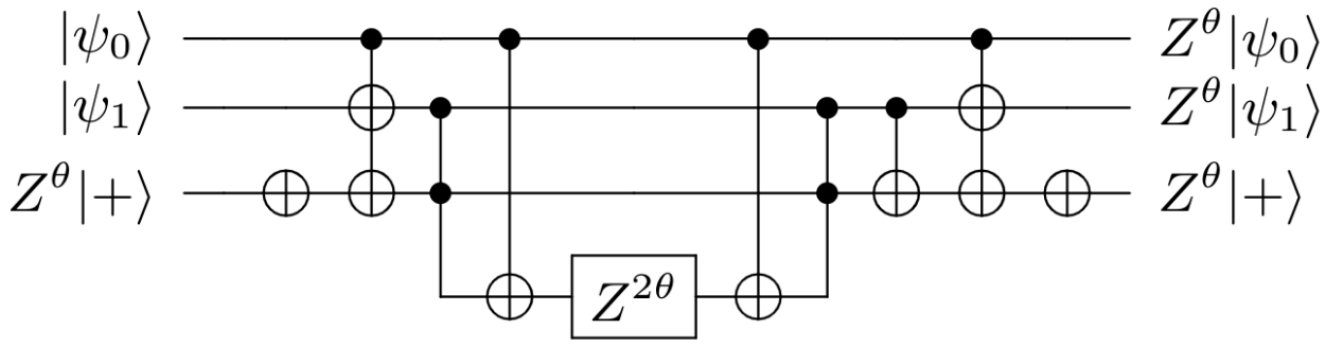

Figure 14: Generalized phase catalysis circuit. Given a $Z^\theta|+\rangle$ catalyst, two Z^θ operations can be applied via stabilizer gates, one AND computation gate (notation from [17]), and one $Z^{2\theta}$ gate.

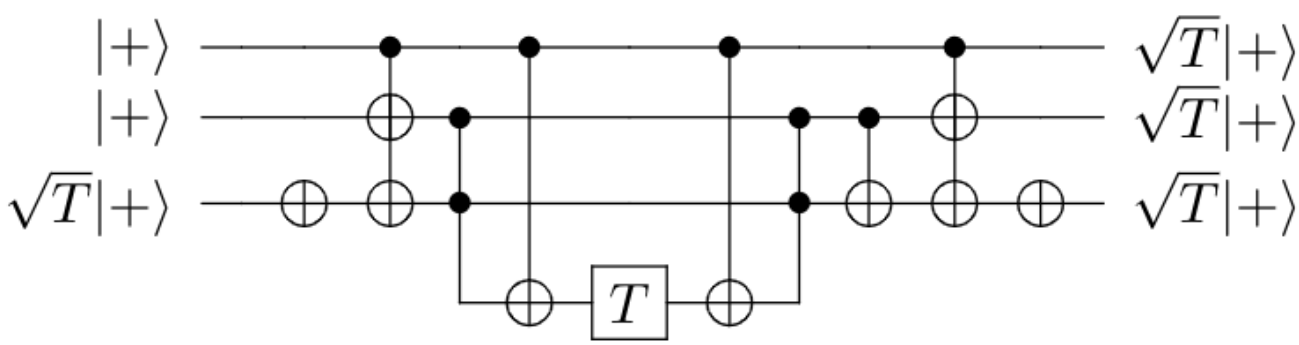

Figure 15: Using a catalyst $|\sqrt{T}\rangle$ state to create 2 additional $|\sqrt{T}\rangle$ states using cheap stabilizer operations, one T gate, and one AND computation. Has a T-cost of 5 [21, 17], implying the T-cost of a $\sqrt{T}$ state is at most 2.5.

tory and keep it producing a pair of $|T_2\rangle$ states every $6.5d$ cycles.

Of course, the $|T_1\rangle$ factories do not always function perfectly. They discard their output roughly 3% of the time due to detecting an error (computed in Section 2). In order to actually achieve a depth of $6.5d$ for the $8|T\rangle \overset{28\epsilon^2}{\rightarrow} 2|T\rangle$ factory, it is necessary increase the $|T_1\rangle$ factory output rate by more than 3% to compensate. There are many ways to achieve such a small gain, and Figure 16 sketches one way to do so. Therefore the T_1 factories can keep up with the catalyzed T factory producing a pair of $|T_2\rangle$ states every $6.5d$ cycles.

6 Conclusions

In this paper we presented two factories: a $|CCZ\rangle$ factory and a catalyzed $|T\rangle$ factory. We compiled these factories all the way down to 3d topological diagrams (see Figure 19) and gave detailed estimates of their spacetime volume, footprint, and error rates. We also showed how to generalize the phase catalysis technique used by our $|T\rangle$ factory to apply to arbitrary angles, including the particularly-efficient angle of $\theta = 22.5°$. Finally, we slightly improved the output rate of the level 1 T factories from [15], gave a simple procedure for transforming topological diagrams into ZX calculus graphs, provided a resource estimator spreadsheet, and gave working simulator links for verifying most of our circuit constructions.

Because it takes four $|T\rangle$ states to perform a Toffoli gate, but only one $|CCZ\rangle$ state to do the same, algorithms dominated by applying Toffolis, such as Shor's algorithm and the chemistry algorithm in [1], run five times as fast when using our $|CCZ\rangle$ factory instead of the $|T\rangle$ factory from [15]. However, we caution that it is often necessary to rework these algorithms' circuits to account for the much faster Toffoli rate. Assuming that such a reworking is possible for [1], the runtimes at classically intractable sizes would be reduced from ∼10 hours (see table VII of [1]) to ∼2 hours. For algorithms dominated by performing T gates, our catalyzed T factory provides a more modest 2× speedup.

We believe it is possible to further decrease the volume of our factories. For example, we suspect that the level 1 T state injection at the end of each factory can be partially merged with that factory's final stabilizer measurement. If that is true, then the depth of the factories could be reduced by $1d$. However, the effect of this optimization on the topological error rate is difficult to predict and we will use simulation to check the optimization's correctness before claiming it.

Another possible optimization is to eagerly route $|CCZ\rangle$ qubits emerging from the CCZ factory to their final destination (in preparation for a parity measurement), instead of holding them next to the factory until they are verified. Removing the output-holding area reduces the CCZ factory's footprint by over 20%, which is a large gain, but it is important to keep in mind that this is not a true reduction in volume but rather a reclassification of some of the factory's volume as routing volume.

Yet another possible optimization would be to carefully analyze how topological errors within the surface code propagate through the factory. At any location where an error chain between two boundaries would result in a detected failure, the boundaries can be moved closer together.

A final idea that should be investigated is estimating logical error probabilities from the observed pattern of detection events produced by the surface code's stabilizer measurements. For example, if there were a sudden burst of detection events crossing between two boundaries during the execution of a factory, the factory's output could be cautiously discarded even if the logical measurement results indicate there is no problem. Assuming there is some metric that can be derived from the raw detection events, that reliably correlates with the true failure probability, this would allow us to reduce the number of false negatives (where an undetected error escapes the factory) at the cost increasing the number of false positives (where a run with no error is discarded).

In this paper we focused on making a low-volume factory in the single-factory regime, but it is also important to consider factories optimized to have a tiny footprint. Early quantum computers will have limited space; it's worth sacrificing depth if it means the factory actually fits on the machine. By combining techniques from this paper and low-footprint distillation techniques mentioned in [25], it should be possible to create factories covering fewer qubits but with roughly the same volume as ours.

Another interesting avenue to explore is the high-footprint / multi-factory regime, where factories based on block codes become possible. Block factories should be able to outperform the efficiency of our factory, assuming enough states are being distilled in parallel. But this raises the question of whether block factories can also be improved by catalysis; are there catalyzed block factories? We don't know the answer to this question.

We suspect that the space of quantum circuits contains many other gems akin to the catalyzed phasing circuit. We consider finding these circuit to be important, because they can be surprisingly efficient at their tasks. It would be particularly useful to have a general framework for finding catalyzed circuits, to better understand what makes them efficient, and to understand

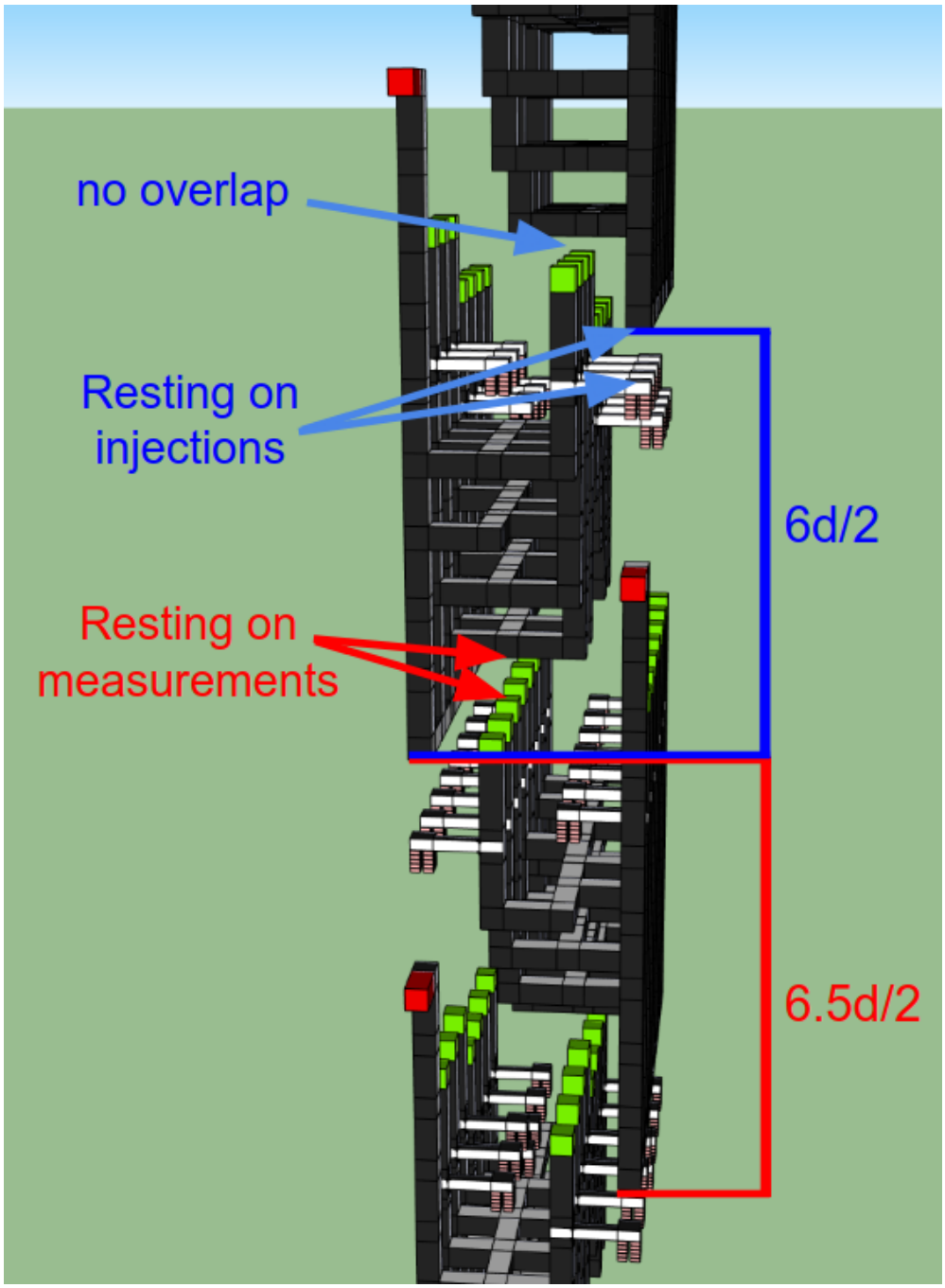

Figure 16: 3D topological diagram of a re-arrangement of the level 1 T factory from [15]. Quantities are quoted in units of $d/2$ instead of d because the factory is performed at half code distance. Improves the depth from $13d/2$ to $12.5d/2$, increasing the output rate by roughly 4%, which ensures four level 1 T factories is sufficient to feed our T-catalyzed factory. There are two variants of the factory: the one shown with an output on the left (A) and the one shown with an output on the right (B). The qubits of A and B have been permuted so that their first stabilizer measurement involves qubits that are all on the same side, allowing the stabilizer measurement to be performed without using a central bar. The second measurement of B (the first measurement using the central bar) is over the back 8 qubits and the last measurement of A is over the front 8 qubits. This allows B to be lowered by half of $d/2$, so that B rests on the level 0 $|T\rangle$ injections to the right of A. The transition back from B to A cannot be lowered quite as far, because the top of B would intersect the first central bar used in A. Overall this optimization saves $0.5d/2$ depth relative to the interleaving technique used in [15].

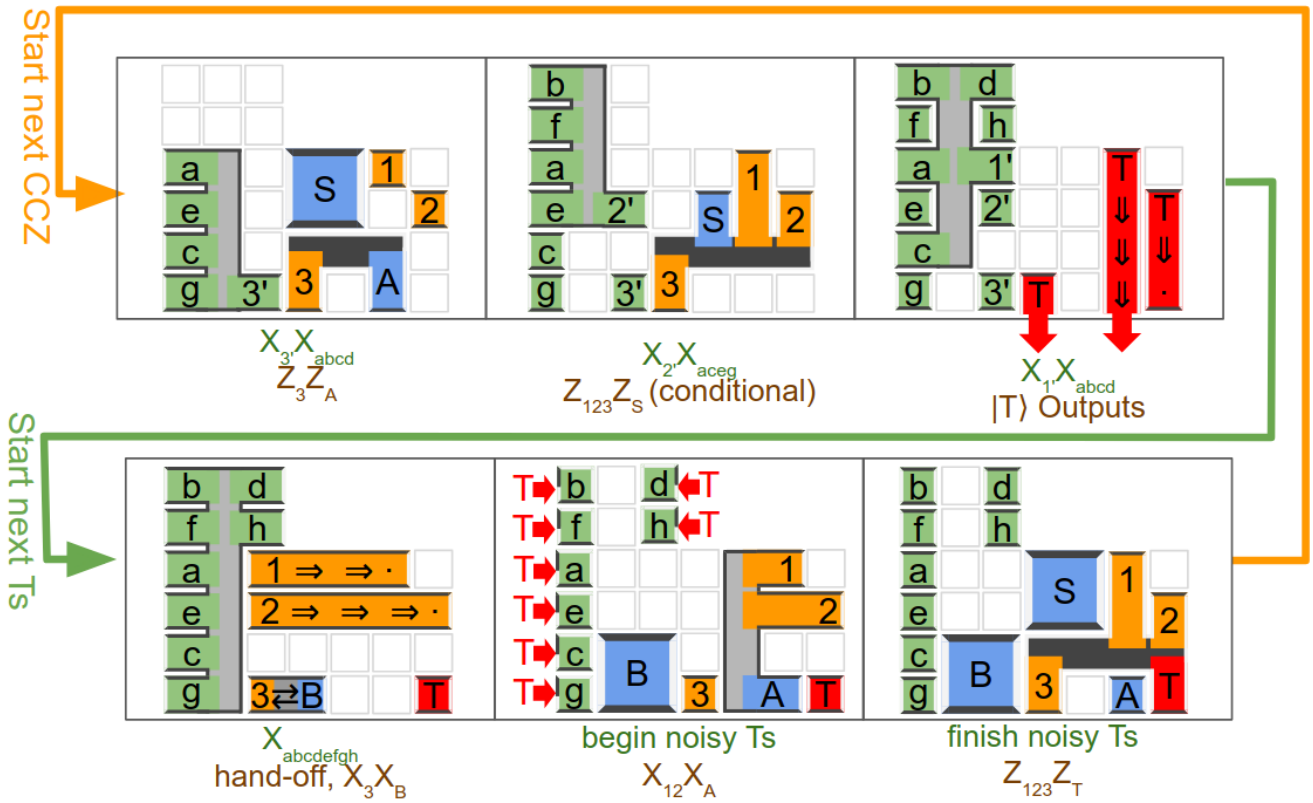

Figure 17: Time slices of activity during catalyzed $|T\rangle$ state distillation. Every step being performed can be matched up with Figure 5 and Figure 12. See Figure 18 for the 3D topological diagram these time slices come from.

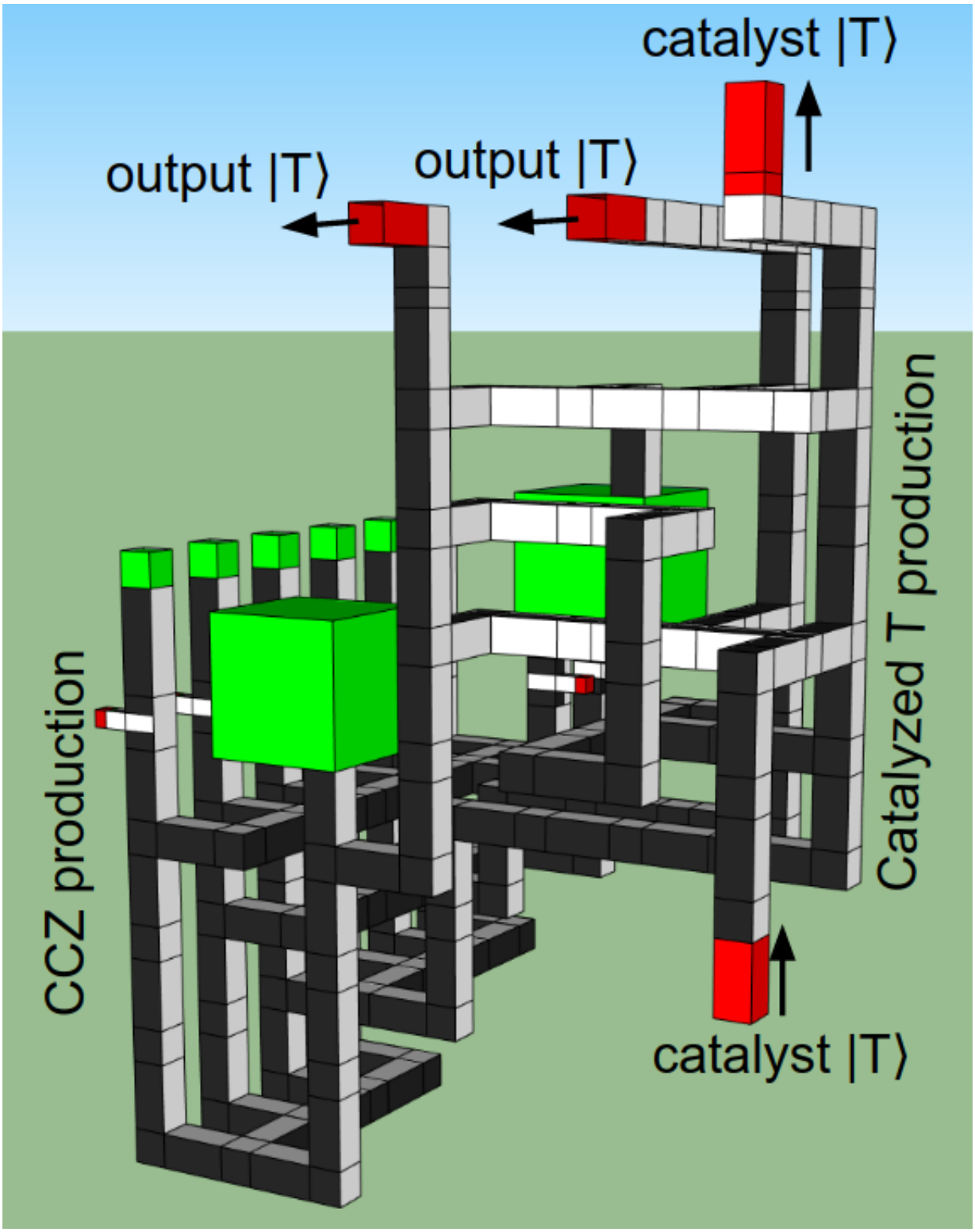

Figure 18: 3D topological diagram of the full $|T\rangle$-catalyzed $8|T\rangle \overset{28\epsilon^2}{\to} 2|T\rangle$ factory. Contrast with the time slices from Figure 17, and the circuit in Figure 5 combined with the circuit in Figure 10. Single-qubit Clifford gates that would affect the catalyst $|T\rangle$ if they failed are performed with extremely conservative code distance (large green boxes).

the connection with related constructions such as distillation via parity-checks [9], synthillation [8], and phase gradient kickbacks [22, 17, 27]. Our guess as to the nature of the connection between these constructions is that there are a small number of circuit identities underlying all these related but different techniques, and that each technique is rewriting and interpreting the underlying circuit identities in a different way. If the connection is actually of this form, then perhaps it is possible to write code that takes a circuit using one of these techniques, derives the identity the circuit is using, and then produces a whole related family of interesting circuits (perhaps including circuits that use catalysis).

7 Acknowledgements

We thank Earl T. Campbell for reviewing a draft of this paper and suggesting useful references.

References

[1] Ryan Babbush, Craig Gidney, Dominic W Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding electronic spectra in quantum circuits with linear t complexity. *arXiv preprint arXiv:1805.03662*, 2018. DOI: 10.1103/PhysRevX.8.041015.

[2] Alex Bocharov, Martin Roetteler, and Krysta M Svore. Efficient synthesis of universal repeat-until-success quantum circuits. *Physical review letters*, 114(8):080502, 2015. DOI: 10.1103/PhysRevLett.114.080502.

[3] S. B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary. *quant-ph/9811052*, 1998. URL https://arxiv.org/abs/quant-ph/9811052.

[4] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Physical Review A*, 86(5):052329, 2012. DOI: 10.1103/PhysRevA.86.052329.

[5] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, 2005. DOI: 10.1103/PhysRevA.71.022316.

[6] Benjamin J Brown, Katharina Laubscher, Markus S Kesselring, and James R Wootton. Poking holes and cutting corners to achieve clifford gates with the surface code. *Physical Review X*, 7(2):021029, 2017. DOI: 10.1103/PhysRevX.7.021029.

[7] Earl T Campbell. Catalysis and activation of magic states in fault-tolerant architectures. *Physical Review A*, 83(3):032317, 2011. DOI: 10.1103/PhysRevA.83.032317.

[8] Earl T Campbell and Mark Howard. Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost. *Physical Review A*, 95(2):022316, 2017. DOI: 10.1103/PhysRevA.95.022316.

[9] Earl T Campbell and Mark Howard. Magic state parity-checker with pre-distilled components. *Quantum*, 2:56, 2018. DOI: 10.22331/q-2018-03-14-56.

[10] Niel de Beaudrap and Dominic Horsman. The zx calculus is a language for surface code lattice surgery. *arXiv preprint arXiv:1704.08670*, 2017. URL https://arxiv.org/abs/1704.08670.

[11] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. *J. Math. Phys.*, 43: 4452–4505, 2002. DOI: 10.1063/1.1499754. quant-ph/0110143.

[12] Bryan Eastin. Distilling one-qubit magic states into toffoli states. *Physical Review A*, 87(3):032321, 2013. DOI: 10.1103/PhysRevA.87.032321.

[13] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, 2012. DOI: 10.1103/physreva.86.032324. arXiv:1208.0928.

[14] Austin G Fowler and Simon J Devitt. A bridge to lower overhead quantum computation. *arXiv preprint arXiv:1209.0510*, 2012. URL https://arxiv.org/abs/1209.0510.

[15] Austin G Fowler and Craig Gidney. Low overhead quantum computation using lattice surgery. *arXiv preprint arXiv:1808.06709*, 2018. URL https://arxiv.org/abs/1808.06709.

[16] Austin G Fowler, Simon J Devitt, and Cody Jones. Surface code implementation of block code state distillation. *Scientific reports*, 3:1939, 2013. DOI: 10.1038/srep01939.

[17] Craig Gidney. Halving the cost of quantum addition. *Quantum*, 2:74, 2018. DOI: 10.22331/q-2018-06-18-74.

[18] Daniel Gottesman and Isaac L Chuang. Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations. *Nature*, 402(6760):390, 1999. DOI: 10.1038/46503.

[19] Clare Horsman, Austin G Fowler, Simon Devitt, and Rodney Van Meter. Surface code quantum computing by lattice surgery. *New Journal of Physics*, 14(12):123011, 2012. DOI: 10.1088/1367-2630/14/12/123011.

[20] Daniel Jonathan and Martin B Plenio. Entanglement-assisted local manipulation of pure

quantum states. *Physical Review Letters*, 83(17): 3566, 1999. DOI: 10.1103/PhysRevLett.83.3566.

[21] Cody Jones. Low-overhead constructions for the fault-tolerant toffoli gate. *Physical Review A*, 87(2):022328, 2013. DOI: 10.1103/PhysRevA.87.022328.

[22] Alexei Yu Kitaev, Alexander Shen, Mikhail N Vyalyi, and Mikhail N Vyalyi. *Classical and quantum computation*. American Mathematical Soc., 2002.

[23] Andrew J Landahl and Chris Cesare. Complex instruction set computing architecture for performing accurate quantum z rotations with less magic. *arXiv preprint arXiv:1302.3240*, 2013. URL https://arxiv.org/abs/1302.3240.

[24] Ying Li. A magic states fidelity can be superior to the operations that created it. *New Journal of Physics*, 17(2):023037, 2015. DOI: 10.1088/1367-2630/17/2/023037.

[25] Daniel Litinski. A game of surface codes: Large-scale quantum computing with lattice surgery. *Quantum*, 3:128, 2019. DOI: 10.22331/q-2019-03-05-128.

[26] Prashant Mishra and Austin Fowler. Resource comparison of two surface code implementations of small angle z rotations. *arXiv preprint arXiv:1406.4948*, 2014. URL https://arxiv.org/abs/1406.4948.

[27] Yunseong Nam, Yuan Su, and Dmitri Maslov. Approximate quantum fourier transform with o(n log(n)) t gates. *arXiv preprint arXiv:1803.04933*, 2018. URL https://arxiv.org/abs/1803.04933.

[28] Maura B Paterson and Douglas R Stinson. Yet another hat game. *the electronic journal of combinatorics*, 17(1):R86, 2010. URL https://www.combinatorics.org/ojs/index.php/eljc/article/view/v17i1r86.

[29] R. Raussendorf and J. Harrington. Fault-tolerant quantum computation with high threshold in two dimensions. *Phys. Rev. Lett.*, 98:190504, 2007. DOI: 10.1103/PhysRevLett.98.190504. URL https://doi.org/10.1103/PhysRevLett.98.190504. quant-ph/0610082.

[30] R. Raussendorf, J. Harrington, and K. Goyal. Topological fault-tolerance in cluster state quantum computation. *New J. Phys.*, 9:199, 2007. DOI: 10.1088/1367-2630/9/6/199. URL https://doi.org/10.1088/1367-2630/9/6/199. quant-ph/0703143.

[31] Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. Ieee, 1994. DOI: 10.1109/SFCS.1994.365700.

[32] Christof Zalka. Fast versions of shor's quantum factoring algorithm. *arXiv preprint quant-ph/9806084*, 1998. URL https://arxiv.org/abs/quant-ph/9806084.

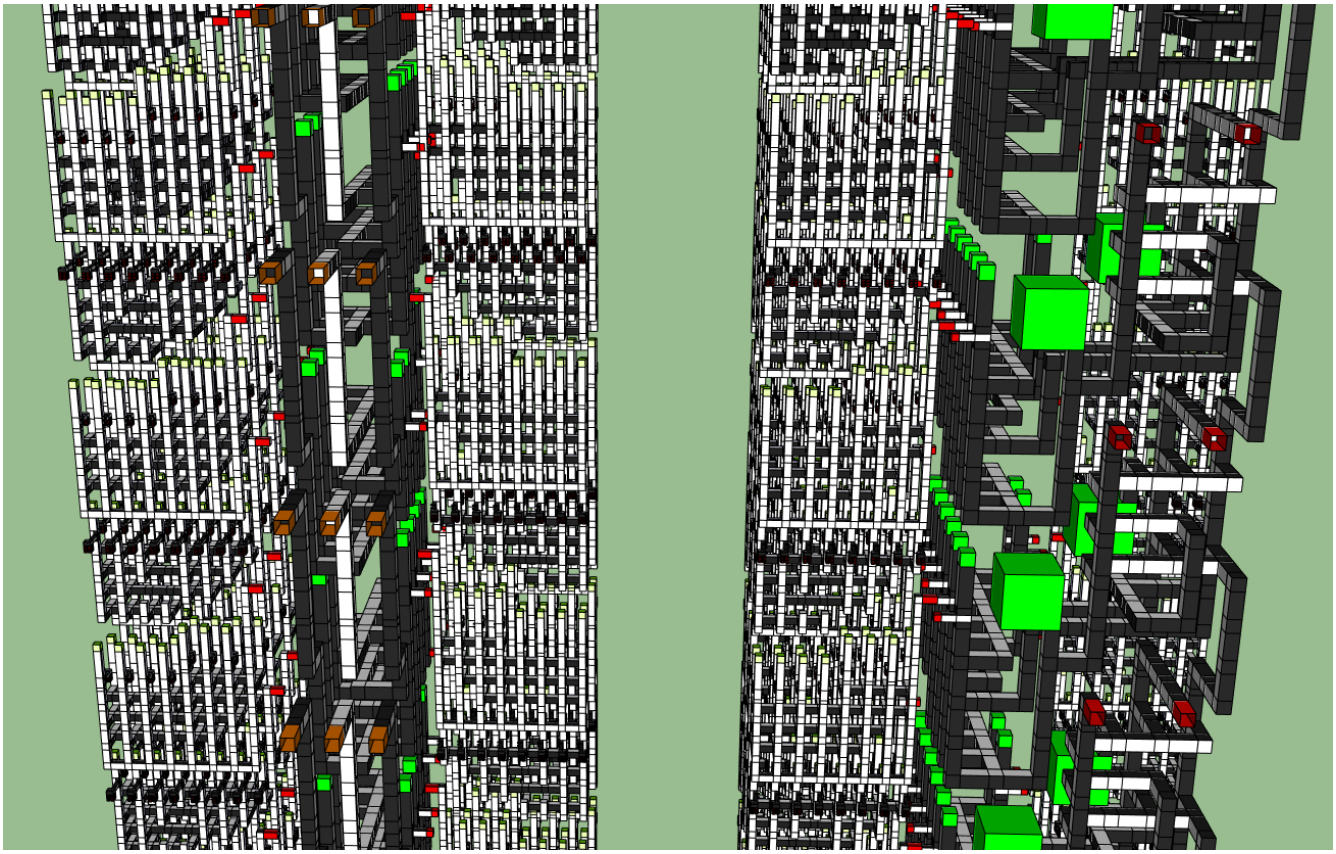

Figure 19: 3D topological diagram showing tiled operation of our CCZ factory (left) and our catalyzed T factory (right). Includes the half-distance level 1 T factories feeding noisy states into the larger factories.