

# Magic States Distillation Using Quantum LDPC Codes.

David Ponarovsky

March 30, 2024

## 1 Current Status.

1. Section 5 - Correct. In any CSS code, one can find a large subspace  $\Lambda \subset C_X$  with a dimension that is linear in  $n$  and this subspace also satisfies the required relation for distillation. Specifically, for any  $x \in \Lambda, y, z \in C_X$ , it holds that  $xy = 0$  and  $xyz = 0$ .
2. Sections 6 and 7 - Incorrect. Initially, I believed that assuming the code is LDPC, one could encode the state  $C_Z^\perp$  in constant depth. However, this idea turned out to be incorrect both in calculation and in contrast to the fact that synthesizing the ground state of the Toric code requires  $\Omega(\log n)$  depth.

## 2 Punctured Polynomial Codes.

For  $\Delta = 4^c, \Delta < q$ , We have that

$$\sum_{x \in \mathbb{F}_\Delta} x^i = \Delta \in \{0, \Delta/2\}$$

$$\sum_{\substack{x \in \mathbb{F}_\Delta \\ x < \Delta}} x^{i+j} = \Delta \sum_{x \in \mathbb{F}_\Delta} x^{i+j} = \Delta \begin{cases} 0 & i+j \neq \Delta-1 \\ \Delta-1 & \text{else} \end{cases}$$

So the punctured  $d$ -degree polynomial code is orthogonal for the punctured  $n-1-d$  polynomial code. So we can take  $d = q/2 - 1$ , and  $\Delta = \alpha q$  to have  $[\alpha q, q/2 - 1, q/2 - (1 - \alpha)q]$  code. For example we can take  $\alpha = 7/8$  and have  $[7/8q, q/2 - 1, 3/8q]$ . The rate of the code is

$$\sim \frac{1}{2} \frac{7}{8} = \frac{4}{7} > \frac{1}{2}$$

**Claim 2.1.** For any  $\Delta > 5$  there are good LDPC family  $C$  such that for any  $x, y \in C$  it holds that  $x \cdot y =_{(\Delta-1)} 0$ .

*Proof.* Consider the Tanner code defined by using the  $\Delta$ -punctured polynomial code as  $C_0$ , where the rate of  $C_0$  is strictly greater than  $\frac{1}{2}$ . Then we have for any  $x, y \in C$ :

$$x \cdot y =_{(\Delta-1)} \sum_{v \in V^+} x|_v \cdot y|_v =_{(\Delta-1)} 0$$

□

### 3 Candidate For Triorthogonal LDPC Code.

Consider the Tanner **Graph**, such that the graph  $G$  is bipartite, and every two checks overlap on the  $i$ th bucket,  $\Delta$ -size, bits. So for any two checks, we have that

$$\begin{aligned} \sum_{x=i \cdot \Delta}^{(i+1)\Delta} x^j &=_{\Delta} \sum_{x'=(i-1) \cdot \Delta}^{i\Delta} (x' + \Delta)^j \\ &=_{\Delta} \sum_{x=(i-1) \cdot \Delta}^{i\Delta} x'^j = \sum_{x \in \mathbb{F}_{\Delta}} x^j \\ &= \sum_{x \in \mathbb{F}_{\Delta}} (x + a\Delta)^i (x + b\Delta)^j = \sum_{x \in \mathbb{F}_{\Delta}} x^{i+j} \end{aligned}$$

So it's left to show that if we take the bipartite graph to be an expander graph then we have a good code.

Let  $G$  be a bipartite graph  $G = (L, R, E)$  that is a  $(n, m, \gamma, \alpha)$  expander. This means that for any subset  $S \subset V(G)$  with  $|S| < \gamma n$ , the size of the group of neighbors of  $S$  is at least  $\Gamma(|S|) > \alpha|S|$ . Consider the graph  $G' = (\Delta \times L, R, E')$  defined as follows:

$$E' = \{ \{ (i, v), u \} : i \in [\Delta], \{ u, v \} \in E \}$$

Thus for any  $S \subset \Delta \times L$  if  $|S|/\Delta < \gamma n$  we have that:  $\Gamma'(S) < \Gamma(|S|/\Delta)$ .

Therefore, if  $S$  is the set of vertices associated with the non-trivial symbols induced by the assignment of a codeword on the vertices, then if  $|S| < \gamma n$ , we have:

$$\frac{|S|}{\Gamma'(|S|)} \leq \frac{|S|}{\Gamma(|S|/\Delta)} \leq \frac{\Delta}{\alpha}$$

So there is a check that sees on his local view less than  $\Delta/\alpha$  non-trivial bits  $< d(C_0)$ .

### 4 Hyprproduct Code of two Triorthogonal Codes.

Suppose that  $H$  is a parity check matrix such that  $h_i h_j =_{\Delta} \{ \Delta, 0, \Delta/2 \}$  for any two rows. Is that true that the same property holds for the following check matrix?

$$H' \leftarrow [H \otimes I \mid I \otimes H]$$

$$H'_i H'_j = (H \otimes I)_i (H \otimes I)_j + (I \otimes H)_i (I \otimes H)_j$$

Denote  $i = (i_1, i_2)$  and  $j = (j_1, j_2)$ . So:

$$(H \otimes I)_i (H \otimes I)_j = \delta_{i_2, j_2} H_{i_1} H_{j_1}$$

and

$$(I \otimes H)_i (I \otimes H)_j = \delta_{i_1, j_1} H_{i_2} H_{j_2}$$

### 5 The problem with the above.

The code that is obtained by the polynomial tanner is (almost) self dual code, module  $\Delta$  the multiplication  $x \cdot x$  belongs to  $\{0, \Delta/2\}$ . While what we actually want to have is  $x \cdot x =_4 1$ . An idea how to correct that, sets the checks such only two of them don't commute. After taking the Hyprproduct code, they will turned to  $\Theta(\sqrt{n})$  that don't commute. So if we have a perfect  $\Theta(\sqrt{n})$  T states, we can cancel their phase before the encoding.

## 6 Good Codes With Large $\Lambda$ .

**Claim 6.1.** Let  $v_1, v_2 \dots v_k$  vectors in  $\mathbb{F}_2^n$ , then there are  $u_1, u_2 \dots u_{k'}$  for  $k' > k/2$ . Such  $\text{span}\{u_1, u_2 \dots u_{k'}\} \subset \text{span}\{v_1, v_2 \dots v_k\}$  and for any  $i, j$  it holds that  $u_i u_j = 0$ .

```

1 Let  $J \leftarrow \emptyset$ 
2 for  $i \in [k/2]$  do
3    $J \leftarrow J \cup \{v_{2i-1}, v_{2i}\}$ 
4   for  $S \subset J$  do
5     Compute the vector  $m_S$ 
6     define as  $m_{S,j} = u_j \sum_{w \in S} w$ 
7   end
8   Pick  $S$  such  $m_S = 0$  and set
       $u_i \leftarrow \sum_{w \in S} w$ 
9   Choose randomly  $w \in S$  and set
       $J \leftarrow J/w$ 
10 end
    : Find commuted vectors  $u_1, u_2, \dots u_{k'}$ 

```

```

1 Let  $J \leftarrow \emptyset$ 
2 for  $i \in [k/3]$  do
3    $J \leftarrow J \cup \{v_{3i-2}, v_{3i-1}, v_{3i}\}$ 
4   for  $S \subset J$  do
5     Compute the vector  $m_S$ 
6     define as
         $m_{S,j,j'} = u_{j'} u_j \sum_{w \in S} w$ 
7   end
8   Pick  $S$  such  $m_S = 0$  and set
         $u_i \leftarrow \sum_{w \in S} w$ 
9   Choose randomly  $w \in S$  and set
         $J \leftarrow J/w$ 
10 end
    : Find commuted vectors  $u_1, u_2, \dots u_{k'}$ 

```

*Proof.* Consider Algorithm 1a, We are going to prove that at line number (8) the alg always finds a subset  $S$  that satisfies the equality. Assume not. On one hand, the number of possible values that  $m_S$  can have is  $2^i - 1$ . On the other hand, since  $J$  contains  $i + 1$  vectors on the  $i$ th iteration, it follows that the number of subsets is  $2^{i+1} - 1 \geq 2^i$ .

Therefore, there must be at least two different subsets  $S$  and  $S'$  such that  $u_S = u_{S'}$ . However, this means that

$$\begin{aligned}
 m_{S \Delta S', j} &= u_j \sum_{w \in S \Delta S'} w = u_j \left( \sum_{w \in S \Delta S'} w + 2 \sum_{w \in S \cap S'} w \right) \\
 &= m_{S,j} + m_{S',j} = 0
 \end{aligned}$$

Thus,  $m_{S \Delta S'} = 0$ . Additionally, it is clear that the rank does not decrease, as for  $u_i$ , there exists one  $v_j$  such that only  $u_i$  is supported by  $v_j$ .  $\square$

**Claim 6.2.** Let  $v_1, v_2 \dots v_k$  vectors in  $\mathbb{F}_2^n$  and  $m$  be an integer  $m < k$ , then there are  $u_1, u_2 \dots u_{k'}$  for  $k' > k/2 - m$ . Such  $\text{span}\{u_1, u_2 \dots u_{k'}\} \subset \text{span}\{v_{m+1}, v_{m+2} \dots v_k\}$ , for any  $i, j$  it holds that  $u_i u_j = 0$  and for any  $i \in [k']$ ,  $j \leq m$  it holds that  $u_i v_j = 0$ .

*Proof.* Modify the Algorithm 1a as follows, Initialize  $u_1, \dots, u_m$  to be  $v_1, \dots, v_m$  and  $J = \{v_{m+1}, \dots, v_{2m+2}\}$ . Notice that in the  $i$ th iteration, for the counting argument to work in the proof of Claim 6.1, we have to ensure that:

$$\begin{aligned}
 |J| &\geq m + i + 1, \text{ So } m + i + 1 \leq k - m - i \\
 \Rightarrow i &\leq k/2 - m - \frac{1}{2}
 \end{aligned}$$

In the end,  $u_{m+1}, u_{m+2}, \dots, u_{k'}$  will satisfy the equations.  $\square$

**Claim 6.3.** Let  $v_1, v_2 \dots v_k$  vectors in  $\mathbb{F}_2^n$ , then there are  $u_1, u_2 \dots u_{k'}$  for  $k' > k/4$ . Such  $\text{span}\{u_1, u_2 \dots u_{k'}\} \subset \text{span}\{v_1, v_2 \dots v_k\}$ . And for any  $i, j$   $\sum u_{i,k} u_{j,k} =_4 0$ .

*Proof.* Use the Algorithm 1a twice. However, in the second iteration, define  $m_{S,j}$  to be the product of module 4. Note that  $m_{S,j}$  must be either  $4n$  or  $4n + 2$ . Thus, we can follow the proof of Claim 6.1.  $\square$

**Claim 6.4.** *[COMMENT] Complete for the above the version, which handle triples. number of options is  $(2^i)^2 = 2^{2i}$  and therefore we have the correctness if  $|J| > 2i + 1$ .*

**Claim 6.5.** *Consider the Left-Right  $(\Delta, n)$ -Complex  $\Gamma$ .  $\dim C_X / C_Z^\perp \cap C_Z / C_X^\perp$  is linear in  $n$ .*

*Proof.* The rates of both  $C_X / C_Z^\perp$  and  $C_Z / C_X^\perp$  are  $(2\rho - 1)^2$ , where  $\rho$  can be any number in the range  $(0, 1)$  [LZ22]. Consider choosing  $\rho$  such that the rates of the quotient spaces are strictly greater than  $\frac{1}{2} + \alpha$ . This implies that the rate of their intersection is greater than  $2\alpha$ .  $\square$

**Corollary 6.1.** *Fix the rate of the small codes  $C_A$  and  $C_B$  to  $\rho = \frac{1}{2} + \alpha$ . There is a subspace  $\Lambda \subset C_X / C_Z^\perp$  at rate  $\frac{1}{4} \cdot 2\alpha$  such that for any  $x \in \Lambda$  and  $y, z \in C_Z^\perp \cup \Lambda$  it holds that:*

1.  $xy =_4 0$
2.  $xyz =_4 \sum_i x_i y_i z_i =_4 0$

**Claim 6.6.** *Consider  $C, \Lambda$  and  $C', \Lambda'$  defined in ?? . Denote by  $\bar{\Lambda}$  the subspace  $C / \Lambda$ . Then:*

$$d(C' / \bar{\Lambda}') \geq d(C / \bar{\Lambda})$$

*Proof.* The way we perform Guess elimination is critical. We want to make sure that we do not add an  $\Lambda$  row to a  $\bar{\Lambda}$  row. *[COMMENT] Continue, Easy. Just need to perform the row reduction when rows of  $\Lambda$  at bottom, and then rotate the matrix  $\curvearrowright$*

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \curvearrowright \begin{bmatrix} D & C \\ B & A \end{bmatrix}$$

$\square$

**Claim 6.7** (Not Formal). *It is easy to see that by using concatenation again, one can obtain the code  $\dim \Lambda' \leftarrow \frac{1}{2} \dim \Lambda'$ . For any  $x \in \text{gen } \Lambda'$ ,  $|x|_4 = 1$ , and for any  $x \in C' / \Lambda'$ , we have  $|x|_4 = 0$ .*

*Proof.* *[COMMENT] We will do it by iterating the generators of  $C$  after performing rows reduction to the generator matrix. Now we will concatenate the  $i$  coordinate to complete the weight of the  $i$ th row to satisfy the requirements.*  $\square$

## 7 Compute $|C_Z^\perp\rangle$ In Constant Depth. *[COMMENT] Wrong Section.*

Let  $C_0$  be a  $\Delta$ -length error linear binary code,  $\Gamma$  a  $\Delta$ -regular bipartite graph, and let  $C_Z$  be the Tanner code defined by  $C_0$  and  $\Gamma$ . We are about to prove that the uniform superposition over  $C_Z^\perp$  codewords can be computed with constant probability at a depth dependent only on  $\Delta$ , in particular independent of the  $C_Z^\perp$ -length. For this, we are going to use Proposition 10 in [MN98], which states that both the encoder and the decoder of any stabilizer  $m$ -length code can be implemented by a circuit at depth  $\Theta(\log m)$  with  $\Theta(m^2)$  ancillae.

**Claim 7.1.** *Let  $G$  be a  $\Delta$ -regular bipartite graph, and denote by  $C_Z^\perp$  the dual-tanner code  $\mathcal{T}(G, C_0^\perp)^\perp$ . Then there is a circuit that with constant probability computes the state  $|C_Z^\perp\rangle$  at  $\Theta(\log \Delta)$  depth, and  $\Theta(\Delta^2)n$  ancillary qubits.*

*Proof.* Let  $E_v$  and  $D_v$  be the encoder and the decoder of  $C_0$  over the local view of vertex  $v$ . By [MN98] we have that both have depth  $\Theta(\log \Delta)$  and require  $\Delta^2$  ancillae. Since  $\Gamma$  is bipartite, we can decompose  $V$  into  $V^-$  and  $V^+$  such that the local views of any two vertices in  $V^\pm$  are disjoint. Therefore, for any two different vertices  $v, u \in V^\pm$ , the encoders  $E_v$  and  $E_u$  act on disjoint subsets of qubits, each corresponding to the local view of either  $v$  or  $u$ . Consider the following algorithm:

For any  $v \in V$ , let  $|z_v\rangle$  be the superposition of codewords in  $C_0$  supported by the local view of  $v$ . Similarly, for any subset of vertices  $W \subset V$ , let  $|z_W\rangle$  be the uniform superposition over the subspace spanned by the generators supported by the vertices in  $W$ . In other words:

$$|z_W\rangle = \left| \sum_{v \in W} z_v \right\rangle$$

- 1 Initialize  $2n$  qubits.
- 2 Call the left and right segments  $L$  and  $R$ .
- 3 Apply  $E_v$  in parallel on  $L$  for any  $v \in V^+$ .
- 4 Apply  $E_v$  in parallel on  $R$  for any  $v \in V^-$ .
- 5 XOR  $R$  into  $L$  by applying CNOT from the  $i$ th bit of  $R$  to the  $i$ th bit of  $L$ .
- 6 Apply  $D_v$  in parallel on  $R$  for any  $v \in V^-$ .
- 7 Apply  $H^k$  on  $L$ . And measure.
- 8 Accept if the result in  $C_Z$

**Algorithm 1:** Compute  $|C_Z^\perp\rangle$

Using the notation, applying the encoders  $E_v, E_u$  for any pair of vertices with disjoint local view become:

$$\begin{aligned} E_v \cup E_u |0\rangle^n &= E_v |0 + z_u\rangle = E_v |0_{/u\text{'s view}}\rangle \otimes |z_u\rangle \\ &= |z_v\rangle |z_u\rangle = |z_u + z_v\rangle = |z_{\{u,v\}}\rangle \end{aligned}$$

So applying all the encoders  $E_v$  at once over the positive vertices results in:

$$(\cup_{v \in V^+} E_v) |0\rangle^n = (\cup_{v \in V^+/v_0} E_v) |z_{v_0} + 0\rangle = |z_{V^+}\rangle$$

Thus the whole computation sum up into:

$$\begin{aligned} (\cup_{v \in V^+} E_v) \otimes (\cup_{v \in V^+} E_v) & |0\rangle^n \otimes |0\rangle^n \mapsto \\ \text{CNOT} \sum_{z \in A} \sum_{z' \in B} & |z_{V^+}\rangle |z_{V^-}\rangle \mapsto \\ I \otimes H^k \sum_{z \in A} \sum_{z' \in B} & |z + z'\rangle |z'\rangle \mapsto \\ \sum_{z \in A} \sum_{z' \in B} & |z + z'\rangle (-1)^{wz'} |w\rangle \mapsto \end{aligned}$$

So if  $w \in C_Z$  then clearly  $z'w = 0$ . The probability for that to occur is

$$\Pr[w \in C_Z] = \frac{|C_Z|}{\mathbb{F}_2^n} = 2^{(\rho-1)n}$$

□

## 8 Distillate $|\Lambda + C_Z^\perp\rangle$ Into Magic.

Let  $|f\rangle$  be a codeword in  $C_X$ , and let  $\hat{X}_g$  be the indicator that equals 1 if  $f$  has support on generator  $g$ , and 0 otherwise. Observe that applying  $T^\otimes$  on  $|f\rangle$  yields the state:

$$\begin{aligned} T^{\otimes n} |f\rangle &= T^{\otimes n} \left| \sum_g \hat{X}_g g \right\rangle = \exp \left( i\pi/4 \sum_g \hat{X}_g |g| - 2 \cdot i\pi/4 \sum_{g,h} \hat{X}_g \hat{X}_h |g \cdot h| \right. \\ &\quad \left. + 4 \cdot i\pi/4 \sum_{g,h} \hat{X}_g \hat{X}_h \hat{X}_l |g \cdot h \cdot l| - 8 \cdot i\pi/4 \cdot \text{integers} \right) |f\rangle \\ &= \exp \left( i\pi/4 \sum_g \hat{X}_g |g| - 2 \cdot \pi/4 \sum_{g,h} \hat{X}_g \hat{X}_h |g \cdot h| + 4 \cdot i\pi/4 \sum_{g,h} \hat{X}_g \hat{X}_h \hat{X}_l |g \cdot h \cdot l| \right) |f\rangle \end{aligned}$$

So in our case:

$$\begin{aligned}
T^{\otimes n} |f\rangle &= \\
&= \exp \left( i\pi/4 \sum_{g \in \text{gen } \Lambda} \hat{X}_g \right. \\
&\quad \left. - 2 \cdot \pi/4 \sum_{g,h \in \text{gen } C_Z^\perp} \hat{X}_g \hat{X}_h |g \cdot h| \right. \\
&\quad \left. + 4 \cdot i\pi/4 \sum_{g,h \in \text{gen } C_Z^\perp} \hat{X}_g \hat{X}_h \hat{X}_l |g \cdot h \cdot l| \right) |f\rangle
\end{aligned}$$

So eventually, we have a product of gates when non-Clifford gates are applied on only on generators of  $C_Z^\perp$ .

$$T^n |f\rangle = \prod_{g \in \text{gen } \Lambda} T_g \prod_{g,h \in \text{gen } C_Z^\perp} \{CS_{g,h}|CZ_{g,h}|I\} \prod_{g,h,l \in \text{gen } C_Z^\perp} \{CCZ_{g,h,l}|I\} |f\rangle$$

Decompose  $f = f_1 + f_2$ , where  $f_1$  is supported only on  $C_X/C_Z^\perp$  and  $f_2$  is supported only on  $C_Z^\perp$ . By using commuting relations, the above can be turned into.

$$\begin{aligned}
T^n |f\rangle &= \prod_{g \in \text{gen } \Lambda} T_g X_{f_1} \\
&\quad \prod_{g,h \in \text{gen } C_Z^\perp} \{CS_{g,h}|CZ_{g,h}|I\} \prod_{g,h,l \in \text{gen } C_Z^\perp} \{CCZ_{g,h,l}|I\} |f_2\rangle
\end{aligned}$$

Denote by  $M_1, M_2$  the gates:

$$\begin{aligned}
M_1 &= \prod_{g \in \text{gen } \Lambda, h} \{CZ_{g,h}|I\} \\
M_2 &= \prod_{g,h \in \text{gen } C_Z^\perp} \{CS_{g,h}|CZ_{g,h}|I\} \prod_{g,h,l \in \text{gen } C_Z^\perp} \{CCZ_{g,h,l}|I\}
\end{aligned}$$

And then we get that

$$\begin{aligned}
\prod_{g \in \text{gen } \Lambda} T_g |f\rangle &= M_1^\dagger T^n M_2^\dagger |f\rangle \\
\prod_{g \in \text{gen } \Lambda} T_g |f\rangle &= M_1^\dagger T^n E_L[M_2^\dagger] |L[f]\rangle
\end{aligned}$$

**Claim 8.1.** Let  $v \in V^-$ , and let  $g_1$  be the generator supported by  $v$ , which matches an assignment of a codeword in  $C_A \otimes C_B$  on the local view of  $v$ . Denote by  $U_{v,g_1}$  the control-gate which, depending on the control bit  $(v, 1)$ , turns on  $g_1$  over the edges associated with the local view of  $v$  in the graph  $G$ . Then, the depth of  $U_{v,g_1}$  depend only on  $\Delta$ .

**Claim 8.2.** Let  $(v, g_1)$  and  $(u, g_2)$  be control wires for two different generators in the graph  $G$ . Then  $U_{v,g_1}$  and  $U_{u,g_2}$  **[COMMENT]** There must be a claim about the relationship between two different generators intersection, But I don't sure exactly why.

**Definition 8.1.** We say that a quantum circuit  $\mathcal{C}$  is well error spreading if the light cone define by any  $T$ .

*Proof.* Denote by  $U_v$  the gate which turn on all the generators supported on  $v$ . As any of them is just of a code word of  $C_A \otimes C_B$ , namely turning on generator require touching at most constant number of qubits combing  $\square$

**Claim 8.3.** *The state:*

$$\sum_{z \in C_Z^\perp} \exp \left( -2 \cdot \pi/4 \sum_{g, h \in \text{gen } C_Z^\perp} \hat{X}_g \hat{X}_h |g \cdot h| \right. \\ \left. + 4 \cdot i\pi/4 \sum_{g, h \in \text{gen } C_Z^\perp} \hat{X}_g \hat{X}_h \hat{X}_l |g \cdot h \cdot l| \right) |z\rangle$$

*Can be computed such that any*

**Claim 8.4.** *The state  $(M_2^\dagger \otimes I) |C_Z^\perp + \Lambda\rangle |0\rangle$  can be computed, such that the light cone depth of any non-clifford gate is bounded by constant.*

*Proof.*

$$\begin{aligned} (I \otimes H_X) C X_{n \rightarrow n} (E \otimes E) I \otimes L[M_2^\dagger] & \prod_{\substack{J \in \{\text{gen } \Lambda, g \in J \\ \text{gen } C_Z^\perp\}}} \prod_{g \in J} (I + X_{L[g]}) & |0\rangle |0\rangle \\ = (I \otimes H_X) C X_{n \rightarrow n} \sum_{\substack{z \in C_Z^\perp \\ x \in \Lambda}} e^{\varphi(z)} & |x\rangle |z\rangle \\ = \sum_{\substack{z \in C_Z^\perp \\ x \in \Lambda}} e^{\varphi(z)} & |x+z\rangle |0\rangle \\ = \sum_{\substack{z \in C_Z^\perp \\ x \in \Lambda}} (M_2^\dagger \otimes I) & |x+z\rangle |0\rangle \\ = (M_2^\dagger \otimes I) & |C_Z^\perp + \Lambda\rangle |0\rangle \end{aligned}$$

□

Denote by  $p \in [0, 1]$  the error rate of input magic states, and let  $|A\rangle$  be an ancilla initialized to a one-qubit magic state. This  $|A\rangle$  can be used to compute the  $T$  gate, with a probability of  $Z$  error occurring with a probability of  $p$  [BH12].

**Claim 8.5.** *There are constant numbers  $\zeta_\Delta, \xi_\Delta$ , and a circuit  $\mathcal{C}$  such that:*

1. *In the no-noise setting, The circuit compute the state*

$$\mathcal{C} |0\rangle^{\Theta(n)} \otimes |A\rangle^{\Theta(n)} \rightarrow \prod_{g \in \text{gen } \Lambda} T_g |C_Z^\perp + \Lambda\rangle$$

2. *Otherwise, the circuit computes the state*

$$\mathcal{C} |0\rangle^{\Theta(n)} \otimes |A\rangle^{\Theta(n)} \rightarrow Z^e \prod_{g \in \text{gen } \Lambda} T_g |C_Z^\perp + \Lambda\rangle$$

, where the probability that  $e_i = 1$  is less than  $\zeta_\Delta \cdot p$ . Additionally, for any  $i$ , there are at most  $\xi_\Delta$  indices  $j$  such that  $e_i$  and  $e_j$  are dependent.

*Proof.* Concatenate the  $T^n \otimes I$  with the gate in Claim 8.4.

□

**Claim 8.6.** *For any  $\alpha \in (0, 1)$  the probability that  $|e| > (1 + \alpha)p\zeta_\Delta$  is less than:*

$$\Pr[|e| > (1 + \alpha)\mathbf{E}[|e|]] < \frac{1 \cdot \xi_\Delta n}{\alpha^2 \zeta_\Delta^2 p^2 n^2} = o(1/n)$$

*Proof.* By the Chebyshev inequality, notice that the number for which  $\mathbf{E}[e_i e_j] - \mathbf{E}[e_i] \mathbf{E}[e_j] \neq 0$  is less than  $\xi_\Delta n$ .  $\square$

**Definition 8.2.** We will say that a decoder  $\mathcal{D}$  for the good quantum LDPC code is an good-local decoder if

1. There is a threshold  $\mu n$  such that if the error size is less than  $|e| < \mu n$  then  $\mathcal{D}$  correct  $e$  in constant number of rounds. With probability  $1 - o(1/n)$ .
2. In any rounds  $\mathcal{D}$  performs at most  $O(n)$  work (depth  $\times$  width).
3. The above is true in operation-noisy settings, where there is a probability of  $p$  for an error to occur after acting on a qubit. ( $\star$ )

$\star$  The motivation for this is that if the decoder does not act on the qubit, then it also does not apply a  $T$  gate on it. Therefore, in the distillation setting, there is zero chance for an error to occur.

**Claim 8.7.** Suppose there is a good local decoder  $\mathcal{D}$  for the good qLDPC code. Then, there exists  $p_0$  such that for any sufficiently large  $n$ , there is a distillation protocol that, given  $\Theta(n)$  magic states at an error rate  $p < p_0$ , successfully distills  $\Theta(n)$  perfect magic states with a probability of  $1 - o(1/n)$ . Furthermore, the protocol's space and time complexity (both quantum and classical) are  $\Theta(n)$  and  $\Theta(n^2)$ , respectively.

## References

- [MN98] Cristopher Moore and Martin Nilsson. *Parallel Quantum Computation and Quantum Codes*. 1998. arXiv: [quant-ph/9808027 \[quant-ph\]](#).
- [BH12] Sergey Bravyi and Jeongwan Haah. “Magic-state distillation with low overhead”. In: *Physical Review A* 86.5 (2012), p. 052329.
- [LZ22] Anthony Leverrier and Gilles Zémor. *Quantum Tanner codes*. 2022. arXiv: [2202.13641 \[quant-ph\]](#).