Tensor Network Decoders for Quantum Error Correcting Codes

Varun Seshadri

February 2, 2023

Contents

Todo list

remove todo list when doc is complete

Chapter 1

Introduction

Tensor Networks in the context of Quantum Error Correction have been used in two distinct places:

1. Approximate decoding of QEC Codes

2. Code construction that come with a natural tensor network decoder

1.1 Mathematical Preliminaries

1.1.1 Group Theoretic concepts

Definition 1: Abelian and Non-Abelian Group

A group G is said to Abelian, iff all the elements of the group pairwise commute with each other under the group operation. It is said to be non Abelian otherwise.
Examples

1. The group of integers $\mathbb{Z}$ is Abelian under the addition operation.

2. The set of 2×2 matrices under matrix multiplication form a non-Abelian group.

Pauli Group

The Pauli matrices I, X, Y, Z form a group. The n-qubit Pauli group is defined as follows

Definition 2: n-Qubit Pauli group

$$\mathcal{G}_n = \{f : f = c\sigma_1 \otimes \sigma_2 \otimes \cdots \otimes \sigma_n\} \quad \text{where} \quad \sigma_j \in \{I, X, Y, Z\} \text{ and } c \in \{\pm 1, \pm i\} \tag{1.1}$$

Equivalenty single qubit Pauli operators are also defined on a index notation with the following definition

$$\sigma^0 = I \quad \sigma^1 = X \quad \sigma^2 = Y \quad \sigma^3 = Z \tag{1.2}$$

Stabilizer Group

Coset

Lagrange's Theorem

This is for partitioning the stabilizer group in the disjoint subsets

1.2 Stabilizer Quantum Error Correction

1.2.1 Stabilizer Codes

We consider stabilizer quantum error correcting codes, where the codespace is determined as the subspace of the states which are simultaneously the +1 eigenstates of the stabilizers. The stabilizers are a subset of the n-qubit Pauli group, $\mathcal{G}_n$, i.e; $\mathcal{S} \subset \mathcal{P}_n$. The codespace $mathcalH_0$ can be expressed mathematically as follows

$$\mathcal{H}_0 = \psi : S \left|\psi\right\rangle = \left|\psi\right\rangle \forall S \in \mathcal{S} \tag{1.3}$$

Consider the set of r independent generators of $\mathcal{S}$, then $dim(\mathcal{H}_0) = 2^(n - r)$, where n is the number of physical qubits. In other words, the number of logical qubits is $k = n - r$.

Definition 3: Logical Operators of a Stabilizer Code

The logical operators $\mathcal{L} \subset \mathcal{G}_n$ form a non-Abelian. Given a stabilizer code $\mathcal{S}$, it's logical operators must obey the following

1. It has two sets of generators, X-type and Z-type generators, denoted as Z_α, X_α, with $\alpha = 1, \ldots, k$
2. For any $L \in \mathcal{L}$ and $S \in \mathcal{S}$, $LS = SL$
3. $X_\alpha Z_\beta = (-1)^{\delta_{\alpha,\beta}} Z_\beta, X_\alpha$

Definition 4: Destabilisers or Pure Errors

Another group of abelian operators $P \in (G)_n$ are called pure errors (or) destabilisers. They are characterized as follows

1. It is a set of $n - k$ operators P_i and $P_i^2 = \nVdash$
2. $P_i S_j = (-1)^{\delta_{i,j}} S_j P_i$ for $S_j \in \mathcal{S}$ and $P_i \in \mathcal{P}$

The stabilizers, destabilisers and logical operators together form the n-qubit Pauli group.

1.3 Tensor Network

1.3.1 Tensors

1.3.2 MPS

1.3.3 PEPS

1.3.4 Quantum Circuits as Tensors

Chapter 2

QEC Code Construction

This chapter will contain details from both Tensor Network Codes and Quantum Legos

2.1 Tensor Network description of stabilizer codes

2.1.1 Example: Surface Code

2.1.2 Example: Bacon Shor Code

2.1.3 Example: Concatenated Codes

2.1.4 Simulating the code

2.2 Quantum Lego description of QECC Codes

2.2.1 Example: Surface Code

2.2.2 Example: Bacon Shor Code

2.2.3 Example: Concatenated Codes

2.2.4 Simulating the code

2.3 Tensor Network Decoder

Chapter 3

Decoding of Quantum Error Correcting Codes

3.1 Formal definition of the decoding problem

3.1.1 BSV definiton of Maximum Likelihood decoders

In this section we formally define MLD. We consider a quantum memory composed of n physical qubits. Let $H = (\mathrm{C}^2)^{\otimes n}$ be the full n-qubit Hilbert space and $\mathcal{P}$ be the group of n-qubit Pauli operators**??**.

A quantum code of stabilizer type is defined by an abelian stabilizer group $\mathcal{G} \subset \mathcal{P}$ such that $-I \notin \mathcal{G}$. Quantum codewords are n-qubit states invariant under the action of any element of $\mathcal{G}$. Such states define a codespace $\mathcal{H}_0 = \{\psi in\mathcal{H} : g\psi = \psi \forall g \in \mathcal{G}\}$. The encoding step amounts to initializing the memory in some (unknown) stateρ supported on the codespace $\mathcal{H}_0$. We shall consider a stochastic Pauli noise described bya linear map

$$\mathcal{N}(\rho) = \sum_{f \in \mathcal{P}} \pi(f) f \rho f^\dagger$$

where π is some normalized probability distribution on the Pauli group $\mathcal{P}$. Since the initial state ρ is supported on the codespace $\mathcal{H}_0$, one has $f\rho f^\dagger = \rho$for any $f \in \mathcal{G}$. By the same token, $f\rho f^\dagger = h\rho h^\dagger$whenever $f\mathcal{G} = h\mathcal{G}$. Given a Pauli operator $f \in \mathcal{P}$, a subset $f\mathcal{G} \equiv \{fg : g \in \mathcal{G}\}$is called a coset of $\mathcal{G}$. Clearly, $\mathcal{P}$ is a disjoint union of cosets $C^\alpha = f^\alpha \mathcal{G}$, where f^α is some fixed representative of C^α. The above shows that errors in the same coset have the same action on the codespace. Thus,

$$\mathcal{N}(\rho) = \sum_{\alpha} \pi(f^\alpha \mathcal{G}) f^\alpha \rho f^{\alpha\dagger}$$

where the sum ranges over all cosets of $\mathcal{G}$ and

$$\pi(f\mathcal{G}) = \sum_{g \in \mathcal{G}} \pi(fg)$$

For simplicity, here we assumed that all coset representatives $\{^\alpha$ are hermitian operators. We shall refer to the quantity $\pi(f\mathcal{G})$ as a coset probability. At the decoding step one attempts to guess the coset of the stabilizer group that contains the actual error based on a partial information about the error known as a syndrome. More precisely, let $g^1, g^2, \ldots, g^n \in \mathcal{G}$ be some fixed set of generators of $\mathcal{G}$. Since the generators $\}^i$ pairwise commute, they can be diagonalized simultaneously.

A configuration of eigenvalues $g^i = \pm 1$can be described by a syndrome $s \in \{0,1\}^m$ such that $g^i = (-1)^{s_i} \forall i = 1, dots, m$. Assuming that the generators g^i are independent, there are 2^m possible syndromes. The full Hilbert space can be decomposed into a direct sum of syndrome subspaces

$$\mathcal{H} = \bigoplus_{s \in \{0,1\}^m} \mathcal{H}_s$$

where $\mathcal{H}_s = \{\psi in \mathcal{H} : g^i \psi = (-1)^{s_i} \psi \forall i\}$. Note that the codespace $\mathcal{H}_0$ corresponds to the zero syndrome. A Pauli operator $f \in \mathcal{P}$ is said to have a syndromes iff $fg^i = (-1)^{s_i} g^i f$for all $i = 1, \ldots, m$. Equivalently, f has a syndromes iff $f\mathcal{H}_0 = \mathcal{H}_s$. For each syndrome s let us choose some fixed Pauli operator $f(s)$ with the syndrome s. One can easily check that the set of all Pauli operators with a syndromes coincides with the coset $f(s)\mathcal{C}(\mathcal{G})$, where

$$\mathcal{C}(\mathcal{G}) = \{f \in \mathcal{P} : fg = gf \forall g \in \mathcal{G}\}$$

is a group known as the centralizer of $\mathcal{G}$. Note that $\mathcal{G} \subseteq \mathcal{C}(\mathcal{G})$. Thus each coset of $\mathcal{C}(\mathcal{G})$ can be partitioned into a disjoint union of several cosets of $\mathcal{G}$. Here we only consider stabilizer codes with a single logical qubit. Let $X, Y, Z \in \mathcal{C}(\mathcal{G})\ \mathcal{G}$ the logical Pauli operators on the encoded qubit. Then each coset of $\mathcal{C}(\mathcal{G})$ consists of four disjoint cosets of $\mathcal{G}$, namely,

$$f(s)\mathcal{C}(\mathcal{G}) = C_I^s \cup C_X^s \cup C_Y^s \cup C_Z^s$$

where,

$$C_I^s = f(s)\mathcal{G}, \quad C_X^s = f(s)\bar{X}\mathcal{G}$$
$$C_Y^s = f(s)\bar{Y}\mathcal{G}, \quad C_Z^s = f(s)\bar{Z}\mathcal{G}$$

The decoding step starts by a syndrome measurement that projects the corrupted state $\mathcal{N}(\rho)$ onto one of the syndrome subspaces $\mathcal{H}_f$. The above arguments show that$f\rho f^\dagger$ has support on $\mathcal{H}_s$ iff $f \in f(s)\mathcal{C}(\mathcal{G})$. Thus the syndrome measurement reveals the coset of $\mathcal{C}(\mathcal{G})$ that contains the error f,whereas our goal is to determine which coset of $\mathcal{G}$ contains f. Using the above equations, one can write the post-measurement (unnormalized) state as

$$\begin{aligned} \rho(s) = \pi(C_I^s).f(s)\rho f(s) \\ +\pi(C_X^s).f(s)\bar{X}\rho\bar{X}f(s) \\ +\pi(C_Y^s).f(s)\bar{Y}\rho\bar{Y}f(s) \\ +\pi(C_Z^s).f(s)\bar{Z}\rho\bar{Z}f(s) \end{aligned}$$

where s is the observed syndrome. Here we assumed for simplicity that $f(s)$ and the logical operators $\bar{X}, \bar{Y}, \bar{Z}$are hermitian. This shows that the effective noise model conditioned on the syndrome can be described by applying one of the four Pauli errors $f(s), f(s)\bar{X}, f(s)\bar{Y}, f(s)\bar{Z}$ with probabilities $\pi(C_I^s), \pi(C_X^s), \pi(C_Y^s), \pi(C_Z^s)$respectively. Clearly, the best possible error correction algorithm for this effective noise model is to choose a recovery operator as the most likely of the four errors.Equivalently, we should choose a recovery operator as any Pauli operator that belongs to the most likely of the four cosets $C_I^s, C_X^s, C_Y^s, C_Z^s$which we denoteCsML. These steps can be summarized as follows.

Algorithm 1: ML-Decoder

```
Input: syndrome $s \in \{0,1\}^m$
Output: recovery operator $g \in \mathcal{P}$

$f(s) \leftarrow$ any Pauli operator with a syndrome $s$
$C_{ML}^s \leftarrow$ arg $max_{\mathcal{C}}\ \pi(\mathcal{C})$, where $\mathcal{C} \in \{C_I^s, C_X^s, C_Y^s, C_Z^s\}$
```

The final step of the decoding is to apply the optimal recovery operator g.It results in a state $g\rho(s)g^\dagger$.We conclude that MLD correctly identifies the coset of $\mathcal{G}$ that contains the actual error and maps the corrupted state $\mathcal{N}(\rho)$ back to the encoded state ρ with a probability

$$P_{success} = \sum_{s\in\{0,1\}^m} \pi(\mathcal{C}^s_{ML})$$

In what follows we shall always ignore overall phase factors of Pauli operators. Such phase factors are irrelevant for our purposes since they do not change the outcome of error correction.

3.1.2 Exact decoding definition of Poulin

Here we provide more detail on how to compute the logical channel for a round of error correction. Note that, via the Choi-Jamiolkowski isomorphism, a qubit channel $\mathcal{E}$ is completely described by the 4×4 process matrix

$$C_{ij} = Tr([P_i \otimes P_j][(\mathcal{E} \otimes \mathcal{I})(\left|\Psi^+\middle\rangle\middle\langle\Psi^+\right|)])$$

where P_i for $i = 0; 1; 2; 3$ are the identity and Pauli X, Y, Z matrices respectively. This represents the state obtained when the channel is applied to the first qubit of a Bell state $|\Psi^+\rangle = |00\rangle + |11\rangle$, when expressed in the Pauli basis. Here we will show how to compute C_ij for the logical channel in the case of surface-code error correction, which will depend on the noise map, the syndrome and the decoder. The simulated error correction process can be decomposed into three parts

$$\mathcal{E} = \mathcal{D}_s \circ \mathcal{R}_s \circ \mathcal{N}$$

where $\mathcal{N}$ is the physical noise map acting on the N qubits of the code, $\mathcal{R}_s$ is the recovery map and $\mathcal{D}_s$ is the decoder correction. The recovery map returns the noisy state to the code space (without performing any classical processing of the syndrome) and can further be decomposed into

$$\mathcal{R}_s = T_s \Pi_s \rho \Pi_s T_s$$

where $\Pi_s$is a projector onto the subspace corresponding to the syndrome s and T_s is a Pauli operator which returns a state in the image of $\Pi_s$to the codespace. We have de nedTsas a product of Paulis that connects the flipped X checks to the top boundary and flipped Z checks to the left boundary. The decoder correction $\mathcal{D}_s$ is the correction selected from $\{\bar{I}, \bar{X}, \bar{Y}, \bar{Z}\}$ by the decoding algorithm to minimise the logical error.

For now, assume that the decoder does nothing so that $\mathcal{E} = \mathcal{R}_s \circ \mathcal{N}$. The logical channel $\mathcal{E}$ is a map from the codespace to itself, so it is effectively a single-qubit map. The process matrix describing the transformation of the logical information given the syndromes is thus

$$C_{ij} = Tr([P_i \otimes P_j][((\mathcal{R}_s \circ \mathcal{N}) \otimes \mathcal{I})(\left|\Psi^+\middle\rangle\middle\langle\Psi^+\right|)])$$

where $|\Psi^+\rangle$is a Bell state of the form $|\Psi+ = |0\rangle_L |0\rangle_a + |1\rangle_L |1\rangle_l\rangle$, where the first qubit is encoded into the surface code and the second qubit is an unencoded ancilla qubit that is assumed to be noise free. Using the cyclic property of the trace and the fact that $T_s\Pi_S = \Pi_0 T_s$, where Π_0 is the projection onto the code space, we can write the above expression more explicitly as

$$C_{ij} = Tr([T_s P_i T_s \Pi_s \otimes P_j][(\mathcal{N} \otimes \mathcal{I})(\left|\Psi^+\middle\rangle\middle\langle\Psi^+\right|)])$$

In the main text we have shown how to represent the Bell state as a tensor network, and how this tensor-network description changes as local noise $\mathcal{N}$, and syndrome projectors Π_s are applied to the state. Using this, the trace over the physical indices only

$$A_i = Tr_L([T_s P_i T_s \Pi_s \otimes P_j][(\mathcal{N} \otimes \mathcal{I})(|\Psi^+\rangle\langle\Psi^+|)])$$

can be represented as a square lattice tensor network, as in 1(d), and can be contracted using the method described in the text. Here Tr_L indicates that the trace is taken over the physical indices of the logical qubit of $|\Psi^+\rangle$and the ancilla qubit is left uncontracted, and untouched by the other operators in the trace. Therefore the expression above has two uncontracted indices (the ancilla indices), i.e. it is a 2×2 matrix. We use these free indices to compute the target quantities C_ij via $C_ij = Tr(P_j A_i)$. Computing C_ij in this way would seem to require computing four full square lattice tensor contractions: one for each A_i, i.e. each row of C_ij. Having computed the effective logical channel $\mathcal{R}_s \circ \mathcal{N}$, a decoder correction $\mathcal{D}_s$ can be incorporated simply by composing the logical channel with the chosen Pauli operator. Here we have shown how to compute the logical channel for a given syndrome and noise model. In the following section we will describe a decoding algorithm based on the above calculations.

Exact Decoding

In the previous section we showed how to compute the logical channel $\mathcal{E}$ for a given noise model and syndrome. This same calculation allows us to perform exact decoding i.e. to select a optimal correction for a given syndrome. Assume that we are given a syndrome s and a noise model $\mathcal{N}$. We can compute $\mathcal{R}_f \circ \mathcal{N}$ using the method described above. Then computing the optimal correction simply amounts to choosing the $\mathcal{D}_s$ from $\{I; X; Y; Z\}$ that minimises the distance of the effective channel from the identity

$$d(\mathcal{D}_s \circ \mathcal{R}_s \circ \mathcal{N}, I)$$

where d can be any distance between operators e.g. diamond distance, delity, or 2-norm distance. The 2-norm distance is usually used due to its easy computability. If $\mathcal{R}_s \circ \mathcal{N}$ is calculated exactly, this decoding algorithm is optimal, in the sense that it exactly chooses the correction that minimises the distance of the logical channel from the identity. It involves the exact calculation of $\mathcal{R}_s \circ \mathcal{N}$ and is therefore exponential, as described above. However, if $\mathcal{R}_s \circ \mathcal{N}$ is already calculated, the remaining calculations involve manipulations of small matrices and take constant time. Thus choosing an optimal correction requires negligible extra work compared to exactly simulating error correction, and therefore we have used this above optimal decoding in all of our exact simulations. When the approximate contraction algorithm is used, the decoder functions almost identically, except that it uses an approximation of $\mathcal{R}_s \circ \mathcal{N}$, rather than the exact channel. In this case, due to possible errors in the approximation, the decoding is not guaranteed to be optimal. However it would be interesting to see whether such a decoding algorithm is sufficiently accurate for practical purposes.

3.2 Tensor Network Construction of the decoding problem in Quantum Error Correction

3.2.1 MPS construction: BSV

Let $f\mathcal{G}$ be one of the cosets $\{C_I^s, C_X^s, C_Y^s, C_Z^s\}$ defined earlier. Our goal is to compute the coset probability $\pi(f\mathcal{G})$. Let π_1 be any probability distribution on the single-qubit Pauli group. For example, $\pi_1(X) = \pi_1(Y) = \pi_1(Z) = \frac{\epsilon}{3}$ and $\pi_1(I) = 1 - \epsilon$ for the depolarizing noise with a rate ϵ. By definition,

$$\pi(f\mathcal{G}) = \sum_{g \in \mathcal{G}} \prod_e \pi_1(f_e g_e),$$

where the product ranges over all edges of the surface code lattice. Let us parameterize $g \in \mathcal{G}$ by binary variables $\alpha_u, \beta_p \in \{0,1\}$ associated with sites u and plaquettes p such that

$$g(\alpha;\beta) = \prod_u (A_u)^{\alpha_u} . \prod_p (B_p)^{\beta_p}$$

Here we used a convention $(B_p)^0 \equiv I$ and $(A_u)^0 \equiv I$. Let e be some edge of the surface code lattice with endpoints $u(e), v(e)$ and adjacent plaquettes $p(e), q(e)$, see Fig. 5.Let g_e be the restriction of g onto the qubit e. Clearly g_e depends only on the bits $\alpha_{u(e)}, \alpha_{v(e)}$ and $\beta_{p(e)}, \beta_{q(e)}$. Thus we can write

$$g_e(\alpha;\beta) = g_e(\alpha_{u(e)}, \alpha_{v(e)}; \beta_{p(e)}, \beta_{q(e)})$$

,where $g_e(i,j;k,l)$ is a function of just four binary variables $i,j,k,l \in \{0,1\}$. For horizontal edges located at the left or the right boundary of the lattice the variable $\alpha_{u(e)}$ or $\alpha_{v(e)}$ respectively is missing. Likewise, for horizontal edges located at the top or the bottom boundary the variable $\beta_{p(e)}$ or $\beta_{q(e)}$ respectively is missing. We arrive at

$$\pi(f\mathcal{G}) = \sum_\alpha \sum_\beta T(\alpha;\beta)$$

where the sums range over binary strings $\alpha, \beta \in \{0,1\}^{d(d-1)}$ corresponding to all possible configurations of variables α_u, β_p and

$$T(\alpha;\beta) = \prod_e \pi_1(f_e g_e(\alpha_{u(e)}, \alpha_{v(e)}; \beta_{p(e)}, \beta_{q(e)}))$$

The righthand side of Eq. (35) coincides with the contraction value of a properly defined tensor network on a two-dimensional grid. To define this tensor network, consider the extended surface code lattice shown on Fig. 6. The extended lattice has three types of nodes which we call s-nodes, h-nodes, and v-nodes. Each s-node represents a location of a stabilizer (either a site stabilizer $A_u$or plaquette stabilizer B_p) while h-nodes and v-nodes represent code qubits located on horizontal and vertical edges of the original surface code lattice respectively. We shall refer to edges of the extended lattice as links todistinguish them from edges of the original surface code lattice. Consider any configuration of variables α, β and the corresponding term $T(\alpha;\beta)$ from above. For each site stabilizer A_u, let us copy the corresponding variable $\alpha_u$to all links incident to the s-node u. Likewise, for each plaquette stabilizer B_p, let us copy the corresponding variable β_p to all links incident to the s-node p. We obtain a labeling of the links by binary variables $\gamma(\alpha;\beta)$ with the property that all links incident to any s-node have the same label. Let us call such a link labeling valid. By definition,$T(\alpha;\beta)$ is a product of terms

$$T_e(\alpha;\beta) = \pi_1(f_e g_e(\alpha_{u(e)}, \alpha_{v(e)}; \beta_{p(e)}, \beta_{q(e)}))$$

associated with h-nodes and v-nodes e of the extended lattice. Since α and β are uniquely determined by the link labeling $\gamma(\alpha;\beta)$, we can also write $T_e(\alpha;\beta)$ as a function of γ, that is, $T_e(\alpha;\beta) = T_e(\gamma)$. This shows that

$$\pi(f\mathcal{G}) = \sum_{valid \gamma} \prod_{e \in h,v} T_e(\gamma)$$

where the product is over all h-nodes and v-nodes and the sum ranges over all valid link labelings. We can now extend the sum in the equation above to all link labelings γ by adding extra terms $T_e(\gamma) \in \{0,1\}$ associated with s-nodes e such that $T_e(\gamma) = 1$ iff all links incident to e have the same label and $T_e(\gamma) = 0$ otherwise. We arrive at

$$\pi(f\mathcal{G}) = \sum_{\gamma} \prod_{e \in h,v} T_e(\gamma)$$

Now the product ranges over all nodes of the extended lattice and the sum ranges over all link labelings. Furthermore, by construction, each term $T_e(\gamma)$ depends only on the labels of links incident to the node e.The expression in the righthand side of the above equation is known as a contraction value of the tensor network defined by the collection of tensors $T_e(\gamma)$. Tensor networks are usually represented by diagrams like the one shown on Fig. 6 such that each box on the diagram carries a tensor with several indexes. Indexes of a tensor are associated with the links emanating from the corresponding box. Diagrams representing the tensors $T_e(\gamma)$ are shown on Eqs. (39,40,41). All tensor indexes i, j, k, l on these diagrams take values $0, 1$. For tensors located at the boundary some of the indexes may be missing. Note that the order of arguments of g_e is interchanged in Eqs. (40,41). This is simply because the qubits located on horizontal edges (h-nodes)have site stabilizers on the left and on the right whereasqubits located on vertical edges (v-nodes) have site stabilizers on the top and on the bottom.

3.2.2 PEPS Construction

We define an Nparticle PEPS $|\psi\rangle$ to be a quantum state whose tensor of coefficients $\psi_{i_1,i_2,\ldots,i_N} = \langle i_1, \ldots i_N|\psi\rangle$ is the contraction of a network of Ntensors $\mathcal{T} = \{A^{(1)}_{i_1,\alpha_1}, A^{(2)}_{i_2,\alpha_2}, \ldots, A^{(N)}_{i_N,\alpha_N}\}$, each of which has one physical index labeled $i$and some number of virtual indices labeled α. We assume that the physical particles have dimension d, that each virtual index has bond dimension D,and that the number of virtual indices per tensor in $\mathcal{T}$is less than a constant n. This implies that the tensor-network description of $|\psi\rangle$ is memory efficient: While the tensor $\psi_{i_1,i_2,\ldots,i_N}$ has d^N entries, each tensor in $\mathcal{T}$ has at most dD^n entries, and thus the whole set $\mathcal{T}$ can be specified with only NdD^n complex numbers.

We consider the optimized layout of the surface code (add figure), wherem qubits are placed on the vertices of a $W \times L$ rectangular lattice, and x-check operators $A_f = \prod_{i \in f} X_i$and z-check operators $B_f = \prod_{i \in f} Z_i$ are defined on alternating faces of the lattice in a checkerboard pattern. This layout is illustrated in (add figure). The logical qubit state $|0\rangle_L$ is defined to be the simultaneous $+1$ eigenspace of all check operators and of the logical operator¯$\bar{Z}$, where¯$\bar{Z}$ is a string of $Z's$ along the left boundary of the code. Likewise, the state $|1\rangle_L$ is fixed by all check operators but is a -1 eigenstate of¯$\bar{Z}$;it can be obtained as¯$\bar{X}\,|0\rangle_L$, where¯$\bar{X}$ is a product of X operators along the bottom boundary of the code. Given that the product state $|0\rangle^{\otimes N}$ has nonzero overlap with $|0\rangle_L$, and that $|0\rangle^{\otimes N}$ is a $+1$ eigenstate of every $B_f$and ¯$\bar{Z}$,we have

$$|0\rangle_L \propto \prod_f \frac{1}{2}(I + A_f)\,|0\rangle^{\otimes N}$$

where $\frac{1}{2}(I + A_f)$ is the projector onto the $+1$ eigenspace of A_f and the product is taken over all x-checks

Let $W_{i,i',\mathbf{0}}(C)$ be a tensor with two physical indices i, i' and an arbitrary number of virtual indices $\alpha = \alpha_1, \alpha_2, \ldots$, which depends on a 2×2 matrix C. Each index of $W(C)$ has bond dimension two, and the only nonzero entries of $W(C)$ are

$$W_{i,i',\mathbf{0}}(C) = \delta_{i,i'} \quad \text{and} \quad W_{i,i',\mathbf{1}}(C) = C_{i,i'}$$

where $0(1)$ means that all virtual indices are set to $0(1)$, and the symbol $\delta_{i,i'}$ denotes the Kronecker delta. For convenience, we define $Q^{\pm} = W(\pm X)$ and $R^{\pm} = W(\pm Z)$.

Consider the projector $\frac{1}{2}(I+A_f)$ acting on particles 1,2,3 and 4 ordered clockwise around a face. It can easily be verified that the tensor $\langle i_1, i_2, i_3, i_4| \, I+A_f \left| i_1^{'}, i_2^{'}, i_3^{'}, i_4^{'} \right\rangle$ can be expressed a contraction of four tensors,

$$\sum_{\alpha_1,\alpha_2,\alpha_3} Q^{+}_{i_1,i_1^{'},\alpha_1} Q^{+}_{i_2,i_2^{'},\alpha_1,\alpha_2} Q^{+}_{i_3,i_3^{'},\alpha_2,\alpha_3} Q^{+}_{i_4,i_4^{'},\alpha_3}$$

We remark that the tensor description for the projection onto the -1 eigenspace of A_f is identical to Eq.(3) but with any one of the Q^{+} tensors replaced with Q^{-}. Projections onto eigenspaces of B_f operators can be defined analogously by replacing all $Q's$ with $R's$. The tensor network corresponding to the product of projectors in Eq.(1) isobtained by overlapping the tensor projector in Eq.(3) overthe whole lattice.

The state $|0\rangle_L$ is then obtained by applying this projector to the state $|0\rangle^{\otimes N}$, which, in the tensor-network picture, corresponds to fixing the second index to zero, thus effectively removing it. A square-lattice tensor network $\mathcal{T} = \{A^{(1)}_{i_1,\alpha_1}, A^{(2)}_{i_2,\alpha_2}, \ldots, A^{(N)}_{i_N,\alpha_N}\}$ for the state $|0\rangle_L$ is thusformed from contractions of Q^{+} tensors. While this tensor network describes the logical $|0\rangle_L$ state, we need to be able to represent other states in orderto fully characterize the transformation of the encoded information during error correction. Specifically, we want to compute the logical channel $\mathcal{E}_L$ that is applied to the logical qubit during a round of error correction. By the Choi-Jamiolkowski isomorphism, this channel can be inferred directly from the resulting output when a Bell state of the form $|\Psi + = |0\rangle_L |0\rangle_a + |1\rangle_L |1\rangle_l\rangle$ input, where the first qubit is encoded in a surface code and is subject to error correction, while the second qubit i sunencoded and assumed to be noise free. We can obtaina tensor-network description for $|\Psi^{+}\rangle$ by a simple modification to the tensor network describing $|0\rangle_L$. Consider the tensor

$$C\bar{N}OT = \sum_{\alpha_1,\alpha_2,\alpha_3} Q^{+}_{i_1,i_1^{'},\alpha_1} Q^{+}_{i_2,i_2^{'},\alpha_1,\alpha_2} \cdots Q^{+}_{i_4,i_4^{'},\alpha_L,\alpha_a}$$

which can be thought of as an L qubit operator (expressed in tensor form) with a single uncontracted virtual index α_a, which we call the ancilla index. If α_a is set to 0, then $C\bar{N}OT$ is simply the identity on its physical indices, while if $\alpha_a = 1$, $C\bar{N}OT$ is an L-fold tensor product of X. Therefore,if $C\bar{N}OT$ is applied to the bottom row of the tensor network for $|0\rangle_L$, the resulting tensor-network state $\psi_{i_1,i_2,\ldots,i_N,\alpha_a} = \langle i_1, i_2, \ldots, i_N, \alpha_a|\psi\rangle$ (which includes the ancilla index as a physical particle) will be the desired Bell state $\left|\Psi + = |0\rangle_L |0\rangle_a + (\bar{X} |0\rangle_L) |1\rangle_l \right\rangle$

3.2.3 Circuit Construction: Ferris Poulin

The notes from the Ferris Poulin Paper

3.2.4 MLD of concatenated Block codes

Concatenated Block codes can be constructed using Tensor Networks. This section will have the proof that tensor network contraction decoding is the same as the maximum likelihood decoding.

Appendix

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.