Research Notes for Kim Lab

Sanchayan Dutta

22 September 2022

Table of Contents

Minimum weight perfect matching vs. union-find decoder

The standard decoder for topological codes is minimum weight perfect matching (MWPM) and it has a worst case complexity of between $O(n^3)$ and $O(n^7)$ depending on the implementation.

Delfosse-Nickerson '21 introduce an almost-linear time decoder based on transforming Pauli error chain into equivalent syndrome erasure chains using the union-find data structure.

Table 1: Comparison of the Union-Find and MWPM decoders' thresholds under phase-flip error.

	UF decoder	MWPM decoder
2d-toric code	9.9%	10.3% [20]
2+1d-toric code	2.6%	2.9% [62]
2d-hexagonal color code	8.4%	8.7% [17]

Quantum Erasure Channel

Delfosse-Nickerson consider a simplified *quantum erasure channel* such that each qubit is subjected to a Pauli Z error with a probability of $\frac{1}{2}$.

We should note here that Delfosse-Zemor '20 in PRR.2.033042 defined the quantum erasure channel as replacing a qubit with the fully mixed state $\frac{I}{2} = \frac{1}{4}(\rho + X\rho X + Y\rho Y + Z\rho Z)$.

More terminology: *Leakage* and *loss* are two ways in which a qubit can be erased. Loss means a qubit has left the computational subspace (for example, the two lowest ground states of an atom are no longer being used and some other energy state is attained). Leakage means that a qubit has been physically lost.

Union-find decoder for surface codes

The algorithm is motivated by the fact that erasure errors are much simpler to decode than Pauli errors. If erasure is that only source of error, then the decoding problem is much simplified as now errors can only be present within the erasure. The peeling decoder introduced by Delfosse-Zemor '20 uses this fact to find a correction in the case when only erasure errors are present. However, if in addition to erasures, Pauli errors are also present, then this algorithm will necessarily fail as the correction will now have support outside the erasures.

Union-find decoder for surface codes

Step I: The decoder takes a syndrome generated by both Pauli errors and erasure $\mathcal{E}$, and from this generate a modified erasure $\mathcal{E}'$ such that there is a valid correction operator supported entirely in $\mathcal{E}$'. This stage is called *syndrome validation*.

Step II: The peeling decoder (DZ '20) is applied on this modified erasure $\mathcal{E}'$. Since this modified erasure with have the same syndrome as the original erasure $\mathcal{E}$ and the Pauli errors, this will return a valid correction operator.

Union-find decoder for surface codes

To perform syndrome validation in Step I, we identify ‘invalid’ clusters of erasures (to be defined) and iteratively grow them until the updated state is correctable by the peeling decoder.

We’ll now see what an invalid cluster is and for that the following lemma is needed (straightforward to prove).

Lemma 1: Let $\mathcal{E}$ be a connected set of edges and let σ be a set of syndrome vertices included in $\mathcal{E}$. There exists a Z error $E_Z \subset \mathcal{E}$ with syndrome σ if and only if the cardinality of σ is even.

Union-find decoder for surface codes

We define a cluster to be a connected component of erased qubits in the subgraph $(V, \mathcal{E})$. These clusters can either be a connected subgraph induced by a subset of erased edges or an isolated vertex. If no Pauli error is present it must be the case that the cluster supports an even number of syndrome bits.

If somehow the cluster contains an odd number of syndrome bits we call that cluster invalid – it implies that there is at least one Z error chain that that lies outside the erasure cluster and terminates in this cluster.

Union-find decoder for surface codes

Since we cannot apply the erasure decoder to odd clusters directly, instead we can grow these clusters by adding edges to the erasure, until they connect with another odd cluster. When two odd clusters merge, the resulting cluster is even.

We note that a single vertex outside the erasure supporting a syndrome bit is an odd cluster.

The following algorithm will describe the merge and update procedure and our previously stated Lemma 1 guarantees that we can apply the peeling decoder on the newly formed even erasure clusters.

Union-find decoder for surface codes

Algorithm 1: Union-Find decoder – Naive version

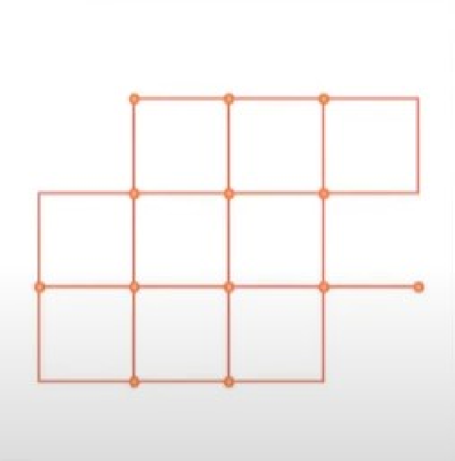
input : The set of erased positions $\varepsilon \subset E$ and the syndrome $\sigma \subset V$ of an error E_Z.

output: An estimation $\mathcal{C}$ of E_Z up to a stabilizer.

```
Create the list of all odd clusters C_1, ..., C_m,
  and initialize the modified erasure ε' = ε.
while there exists an odd cluster do
    for all odd cluster C_i do
        Grow C_i by increasing its radius by one
          half-edge.
        If C_i meets another cluster, fuse and
          update parity.
        If C_i is even, remove it from the odd
          cluster list.
Add full edges that are in the grown clusters to
  ε'.
Apply the peeling decoder to the erasure to
  find C.
```

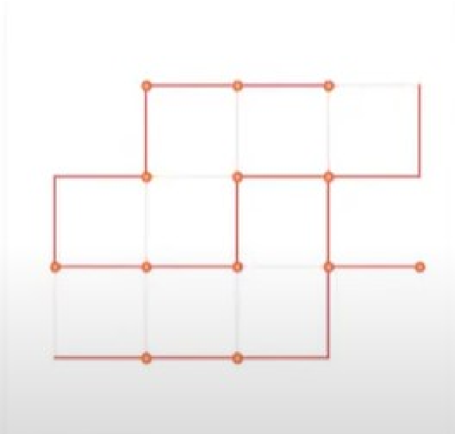
Illustrations for the peeling decoder

Connected subset of erased edges $\mathcal{E}$.



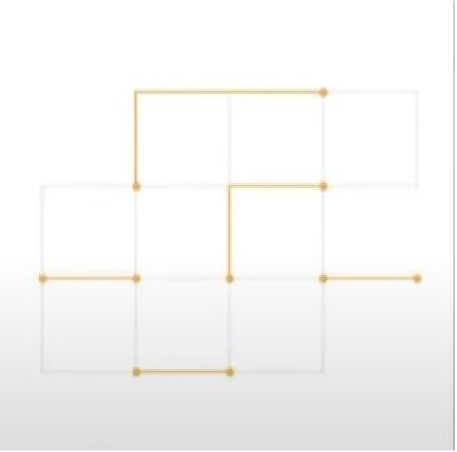
Illustrations for the peeling decoder

Construct a connected spanning forest $F_{\mathcal{E}}$.



Illustrations for the peeling decoder

Peel from the forest to get a minimal correction set A. This set exists and is unique (DZ '20). The peeling decoder is applicable for any even-parity $\mathcal{E}$, owing to Lemma 1.



Schematic Representation of Syndrome Validation

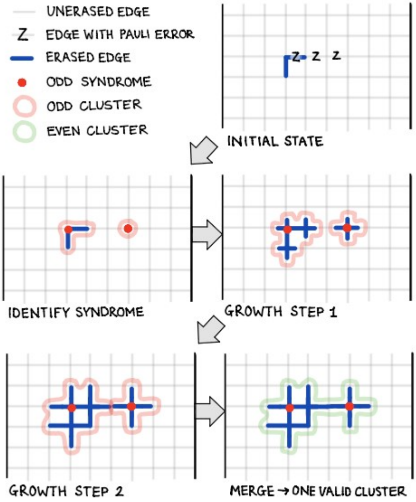


Figure 3: Schematic representation of syndrome validation.

Modified Erasure Decoder Performance

We now discuss the performance of the decoder in the low error regime, and show that it performs as well as the most likely error (MLE) decoder below the minimum distance. MLE decoder returns the most likely error configuration given the observed syndrome.
The MWPM-decoder is an implementation of MLE decoder. The modified erasure decoder here is not exactly a MLE decoder, but it has similar performance below the minimum distance d.

Modified Erasure Decoder Performance

For any code, the MLE decoder can correct any error configuration, E_Z with a weight up to $(d-1)/2$, where d is the minimum distance of the code. For the erasure channel, any erasure pattern of up to $d-1$ qubits can be corrected (**why?**). Moreover, both of these bounds are tight. Reaching these bounds with an efficient decoding algorithm is evidence of good performance.

In our mixed noise model, MLE decoder can correct any combination of t erased qubits and s Z-errors (outside the erased set) as long as $t + 2s < d$.

Modified Erasure Decoder Performance

Theorem 1: If $t + 2s < d$, algorithm 1 can correct any combination of t erased qubits and s Z-error.

Proof. For the case of erasures, the proof is straightforward since the erasure decoder can be called immediately without going through steps 1 to 7.
Consider the other extreme where only Pauli errors occur and denote the error by E_Z. A cluster C grows in algorithm 1 (step 4) if it contains an odd number of syndrome vertices. We recall that a single vertex with syndrome can be considered an odd cluster.

Modified Erasure Decoder Performance

When a cluster grows, following algorithm 1, at least one new half-edge of E_Z is covered. After at most $2s$ rounds of growth, the grown cluster covers the entire error E_Z and there can be no more odd clusters left to grow. At the end of the growing procedure, the diameter of the largest erased cluster is at most $2s$ edges ($2s + 2s$ half-edges). By erasing this cluster, we obtain a erasure pattern that covers E_Z and does not cover a non-trivial logical error since $2s < d$. When the peeling decoder is run in the final step it must therefore succeed at identifying E_Z up to a stabilizer.

Modified Erasure Decoder Performance

The general argument for a combination of s errors and t erasures is similar. Growing the clusters increases the diameter of the largest cluster by at most $2s$. It is then upper bounded by $t + 2s < d$.
Just as in the case of Pauli errors, the final step returns an error equivalent to E_Z, up to a stabilizer. $\square$

Losses and computational errors

Stace-Barrett-Doherty ('09): Losses are both *detectable* and *locatable* suggesting that they should be easier to rectify than computational errors.

Indeed, quantum communication channels can tolerate a much higher rate of loss (~ 0.5) than its largest estimates for computational error threshold ($\sim 10^{-2}$).

The point of the paper is that surface codes can be made robust against both computational errors and loss errors.

Non-uniqueness of logical operators

If a $L \times L$ has periodic boundary conditions, there are $2L^2$ physical qubits and $2(L^2 - 1)$ independent stabilizers. The two remaining degrees of freedom are capable of encoding two logical qubits $\bar{q}_i (i \in \{1, 2\})$.

A logical $\bar{Z}_i$ operator corresponds to a product of Z (X) operators along a homologically non-trivial cycle (i.e., spanning the lattice), shown in the following figure. $\bar{Z}_i$ and $\bar{X}_i$ commute with all the stabilizers but are not contained within the stabilizer group.

Non-uniqueness of logical operators

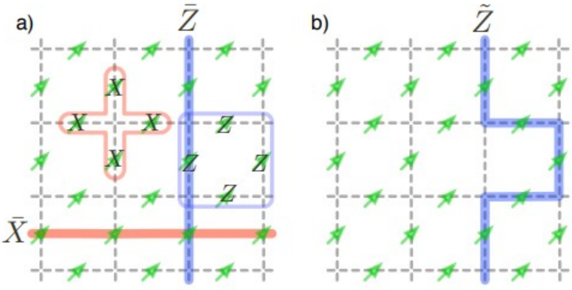


FIG. 1: (a) Physical qubits (arrows) reside on the edges of a square lattice (dashed). Also depicted are a plaquette operator, a star operator, the logical $\bar{Z}$ operator and the logical $\bar{X}$ operator. (b) In the event of a qubit loss, an equivalent logical operator $\tilde{Z}$ can be routed around the loss.

The error model

For the purposes of analysis, the error model considered is local and uncorrelated. Each physical qubit is lost with probability p_{loss}. Losses are presumed to be detectable: a projector onto a computational basis of a given qubit $\Pi_i = |0\rangle_i\langle 0| + |1\rangle_i\langle 1|$ is an observable indicating only whether the state of the qubit has leaked outside of the computational basis.

The remaining qubits are subject to independent bit-flip (X) and phase-flip (Z) errors each with probability p_{com}. Both errors are handled in an analogous way in the surface code, so here we confine our attention to X errors, noting that the thresholds for Z errors will be identical. Aside from these errors, we assume that other quantum operations (e.g., syndrome measurements) can be implemented perfectly.

Logical operators in the presence of losses

The logical operators are not unique; e.g. multiplying $\bar{Z}_i$ by a plaquette stabilizer yeilds a new operator $\tilde{Z}_i = P_p\bar{Z}_i$ that acts identically on the code space, $\tilde{Z}_i|C\rangle = P_p\bar{Z}_i|C\rangle = \bar{Z}_iP_p|C\rangle = \bar{Z}_i|C\rangle$. Thus, there are many homologically equivalent cycles with which to "measure" (what does this mean?) each logical operator, as was shown in the previous figure.

This redundancy allows us to obtain the loss threshold for the case $p_{\text{com}} = 0$; if only a few physical qubits are lost, it is likely that each logical operator can be reliably measured by finding a homologically non-trivial cycle that avoids all lost qubits.

Logical operators in the presence of losses

If p_{loss} is too high, there is likely to be a *percolated* region of losses spanning the entire lattice, in which case there are no homologically non-trivial cycles with which to measure (?) the logical operators.

As $L \to \infty$, there is a sharp boundary between recoverable and non-recoverable errors corresponding to the *bond percolation threshold* on the square lattice: for $p_{\text{loss}} < 0.5$ loss recovery almost entirely succeeds, whereas for $p_{\text{loss}} > 0.5$ loss recovery almost surely fails. Notably, this threshold saturates the bound on p_{loss} imposed by the no-cloning theorem.

Recovery in the presence of computational errors

The case p_{loss} and $p_{\text{com}} > 0$ has been well studied. Briefly, physical bit-flip errors lead to logical bit-flip ($\bar{X}_i$) errors but not logical phase errors and vice-versa. An error chain E is a set of lattice edges (i.e., physical qubits) where a bit-flip has occurred. The plaquette operator eigenvalues change to -1 only at the boundary ∂E of the chain. Measuring the plaquettes operators therefore yeilds information about the endpoints of the connected sub-chains of E.

Recovery in the presence of computational errors

If E crosses $\bar{Z}_i$ an odd number of times, then the logical qubit suffers a $\bar{X}_i$ error. These errors may be corrected if the endpoints, ∂E, can be matched by a correction chain E' such that the closed chain $C = E + E'$ crosses $\bar{Z}_i$ an odd number of times, i.e., C is homologically trivial. The error rate below which the correction chain E' may be successfully constructed is closely related to the phase boundary of the random bond Ising model (RBIM).

If $p_{\text{com}} > p_{c0}$ then in the limit $L \to \infty$, the chain is homologically trivial only $\sim 25\%$ of the time and recovery fails.

Boundary of correctability

We can think of the above results as the endpoints of a boundary of correctability: $(p_{\text{loss}}, p_{\text{com}}) = (0.5, 0)$ and $(0, 0.104)$ respectively. In the following, we (SBD '09) will demonstrate that toric codes are robust against both loss and computational errors, with a graceful tradeoff between the two.

We will describe how losses can be corrected by forming new stabilizer generators which are aggregates of plaquettes or stars, called super plauqettes and superstars, respectively. The superstar and superplaquette eigenvalues then reveal the error syndromes, and a perfect matching algorithm is used to find an error correction chain E'. We illustrate the efficacy of the single round error correction protocol by numerically (*) calculating the boundary of correctability in the $(p_{\text{loss}}, p_{\text{com}})$ parameter space.

Superplaquettes and superstars

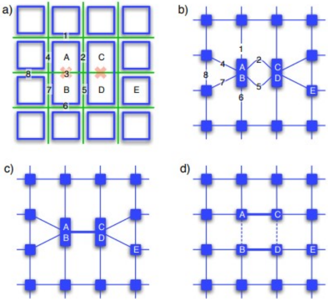


FIG. 2: (a) Lattice with two lost qubits (crosses). Representative qubits and plaquettes are labeled 1 to 8, and A to E respectively. (b) Plaquettes sharing a lost qubit ($\{A, B\}$ and $\{C, D\}$) become superplaquettes AB and CD, and may be multiply connected (i.e. share more than one qubit). (c) Degraded lattice showing superedges (thick lines). (d) Restored lattice with zero weight edges (dotted) and irregular weights (thick).

Superplaquettes and superstars

Consider the lattice shown in Fig 2(a) which is damaged by the loss of two physical qubits, marked by the crosses. The loss of qubit 3 affects the plaquette stabilizers: $P_A = Z_1 Z_2 Z_3 Z_4$ and $P_B = Z_3 Z_5 Z_6 Z_7$ rendering them unmeasurable. However, the superplaquette $P_{AB} = P_A P_B = Z_1 Z_2 Z_4 Z_5 Z_{6_{Z_7}}$ is independent of the qubit at site 3, so stabilizes the remaining qubits.

Without errors, P_{AB} has an eigenvalue of $+1$. An error chain ending within the superplaquette AB changes the eigenvalue of P_{AB} to -1. It follows that the syndrome associated with a superplaquette is determined by the parity of the number of error chains that cross its boundary. The fact that superplaquette operators yeild syndrome information with which to construct an error correction chain E', is the basis for our loss-tolerant error-correction scheme.

Stabilizers in lossy lattice

In general, given any set of lost qubits, we can form a complete set of stabilizers on the damaged lattice in the following way: for each lost qubit q, which participate in the neighbouring (super)plaquettes P_q and P'_q we form a superplaquette operator $P_q P'_q$ which is independent of Z_q. In the same way, we form superstar operators from products of star operators. As discussed earlier, we can also form new logical $\bar{X}_i$ and $\bar{Z}_i$ operators by deforming the original logical operators to conform to the boundaries of the newly formed superplaquettes.

The junk qubit

We note that in Fig 2(a) there is a damaged plaquette operator $\bar{Z}_J = Z_3 P_A = Z_1 Z_2 Z_4$ (or, equivalently $Z_3 P_B = Z_5 Z_6 Z_7$) associated with the lost qubit 3, which commutes with all the Z-stabilizers and X-stabilizers on the damaged lattice, but whose eigenvalue (being a multi-Pauli) is indeterminate ± 1. Likewise, the damaged star operator $\bar{X}_J = X_4 X_7 X_8$ also has indeterminate eigenvalue and commutes with the new stabilizers on the damaged lattice.

The junk qubit

Having indeterminate eigenvalues, $\bar{Z}_J$ and $\bar{X}_J$, which mutually anti-commute, define a two-dimensional degree of freedom in an uncertain state. They therefore describe a completely mixed junk qubit, J, which is a consequence of entanglement between the lost qubit and the remaining qubits. Since $\bar{Z}_J$ and $\bar{X}_J$ each commute with the new stabilizers, and with the deformed logical operators: $|\psi\rangle\langle\psi|_{\bar{q}_1} \otimes |\phi\rangle\langle\phi|_{\bar{q}_2} \otimes \mathbb{I}_{\mathcal{J}/2}$, and so the loss does not affect the logical qubits.

The modified graph

When analyzing the pattern of syndromes on the plaquettes and superplaquettes, we construct a new graph, depicted in Fig 2(b), in which a superplaquette is represented by a node, and superplaquette share a bond on the new graph whenever they share a physical qubit in common. Thus P_{AB} and P_{CD} share the qubits 2 and 5 and this is represented as two edges between the superplaquette nodes labelled AB and CD.

The modified graph

The error correction syndrome ∂E, arising from an error chain on the graph in Fig 2(b) is determined by the (super)plaquettes that have an eigenvalue of -1. To correct the errors, we follow the procedure described by Harrington et al to find the most likely chain given rise to ∂E. The probability of a given error chain is modified by the presence of losses.

With no loss, the probability of error on a qubit, $\ell = \{P, P'\}$ between two neighbouring plaquettes is uniform, $p_l = p_{\text{com}}$. With loss, superplaquettes may share multiple physical qubits in common, as shown in Fig 2(b), where the superplaquettes AB and CD have qubits 2 and 5 in common.

The modified graph

A non-trivial syndrome arises only if either qubit 2 or qubit 5 suffer an error, but not both. By extension, for a pair of neighbouring superplaquettes, $\ell = \{P, P'\}$ sharing n_l physical qubits, a non-trivial syndrome arises only if there are an odd number of errors on the n_l qubits, which happens with probability

$$p_l = \sum_{\text{m odd}} \binom{n_l}{m} p_{\text{com}}^m (1 - p_{\text{com}})^{n_l - m} = \frac{1 - (1 - 2p_{\text{com}})^{n_l}}{2}$$

The modified graph

We therefore degrade the graph shown in Fig 2(b), replacing multi-edges with superedges whose error rates depend on the number of physical qubits shared between neighbouring plaquettes. The degraded lattice is shown in Fig 2(c), in which there are no multi-edges, but the error probabilities are no longer constant.
On this degraded lattice we may now assign a probability for any hypothetical error chain $E' = E + C$, where C is a closed chain. This probability, which is conditioned on the measured syndrome, ∂E is

$$P(E'|\partial E) = \mathcal{N} \prod_{\forall \ell} e^{J_\ell u_\ell^{E'}}$$

where $u_\ell = -1$ if $\ell \in C$ and $+1$ if $\ell \notin C$ for a chain C.
$\mathcal{N} = \prod_{\forall \ell} \sqrt{p_\ell(1 - p_{ell})}$ is a normalization constant and $e^{2J_\ell} = 1/p_\ell - 1$.

The correction chain for loss

The chain E' that maximizes $P(E'|\partial E)$ also minimizes $\sum_{\ell \in E'} J_\ell$. This minimization may be accomplished by using Edmonds' minimum weight perfect matching algorithm. For $p_{\text{loss}} = 0$ this simply minimizes the total metropolis length of the matching path and is the same procedure implemented in the previous studies. For $p_{\text{loss}} > 0$, the edge weights are not uniform since p_l depends on the number of physical qubits n_l shared between adjacent superplaquettes.

To determine the homology classes

For the purposes of simulation, it is easier to determine homology classes on a square lattice rather than the degraded lattice. We therefore restore the square lattice by dividing the superplaquettes into their constituent plaquettes in the following way: (1) An edge between two plaquettes within a single superplaquette is assigned a weight of 0. (2) An edge between plaquettes in two neighbouring superplaquettes is given the weight of the superedge in the degraded lattice as shown in Fig 2(b).

These transformations do not change the weighted distance between any pair of syndromes and so a minimum weight perfect matching on the restored lattice is also a minimum weight perfect matching on the degraded lattice. Determining the homology class is accomplished by counting crossings of vertical and horizontal test lines in the dual lattice.

The simulation procedure

In order to test the efficacy of their loss-tolerant error correction scheme SBD '09 generate random losses on a periodic lattice with rate p_{loss}. On the remaining qubits they generate a set of errors, E, with rate p_{com}. Applying Edmonds' algorithm to ∂E on the weighted lattice yeilds the maximum-likelihood error correction chain E'. The homology class of the chain $E + E'$ then determines whether error correction succeeds.

The simulation procedure

For each value of p_{loss} they simulate the protocol for different values of p_{com} on lattices of size $L = 16, 24$ and 32. For given values of p_{com} and L, the failure rate p_{fail} is calculated by averaging over 10^4 trials. We seek a threshold p_t (depending on p_{loss}) such that $dp_{\text{fail}}/dL < 0$ when $p_{\text{com}} < p_t$ and $dp_{\text{fail}}/dL > 0$ when $p_{\text{com}} > p_t$. That is, for each value of p_{loss}, we fit the simulated failure rate to the universal scaling law $p_{\text{fail}} = f[x]$ where $x = (p_{\text{com}} - p_t)L^{1/\mu_0}$ with fitting parameters p_t, μ_0, a and b.

The region of correctability

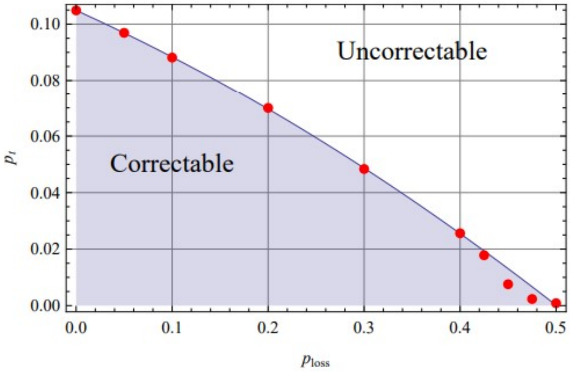


FIG. 3: Correctability phase diagram. The shaded region is correctable in the limit $L \to \infty$. The threshold, p_t, is calculated by fitting the universal scaling law $p_{fail} = f[(p_{com} - p_t)L^{1/\nu_0}]$. The curve is a quadratic fit to the points for which $p_{loss} \leq 0.4$ (where universal scaling is unaffected by the finite lattice size). It extrapolates through $(p_{loss}, p_t) = (0.5, 0)$.

A summary of the protocol

The protocol described in this paper for dealing with losses in a surface code relies on several important properties of the stabilizers:

(1) If a physical qubit q in the logical qubit operator chain $\bar{Z}_i$ is lost, then there is a plaquette P_q, such that $\tilde{Z}_i = \bar{Z}_i P_q$ is independent of q (likewise $\bar{X}_i \to \tilde{X}_i$). Thus, logical error chains can be rerouted around the lost site.

(2) There is another plaquette P'_q such that the superplaquette $P'_q P_q$ is independent of q. Thus, superplaquettes may be constructed to locate the endpoints of the error chains (likewise for superstars).

(3) Newly formed junk qubits are uncorrelated with the logical qubits.

The union-find decoder as a percolation problem

Their paper Thresholds for topological codes in the presence of loss analytically shows that the maximum tolerable qubit loss rate is 0.5 which is the same as the bond percolation threshold on a square lattice. But this is not exactly our problem even in the union-find decoder setting, because for given qubit erasure rate p we don't know how large the modified erasure cluster ε' can be after the merge and join process.

The main barrier to interpreting the union-find decoder as a percolation-like problem seems to be that we do know really know much much the clusters can grow when joining and merging.

Open question: Is there a way around this or can we use a percolation framework for calculating p_{com} at all?