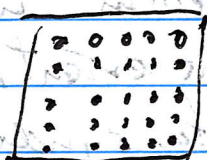


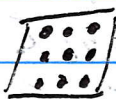
# Dance of the Tweezers

$$H_x = (A \otimes I_n, I_r \otimes A^T)$$

Suppose  $A$  is  $3 \times 5$



Register 1  
5x5



Register 2  
3x3

$$(A \otimes I_5)_{ik,jl} = A_{ij} \delta_{kl}$$

$i \in \{1, 2, 3\}$  3 rows of  $A$

$j \in \{1, 2, 3, 4, 5\}$  5 columns of  $A$

Three checks acting on each

row of 5x5 array

Each check weight  $\leq 5$

$$(I_3 \otimes A^T)_{ik,jl} = \delta_{ij} A_{kl}^T$$

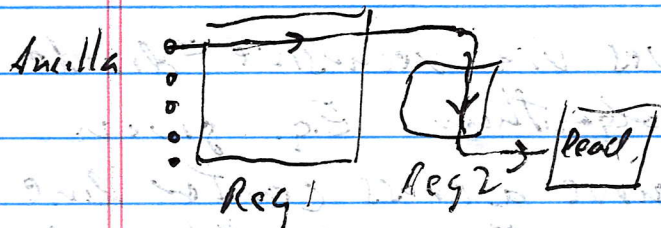
$k \in \{1, 2, \dots, 5\}$  5 columns of  $A$

$l \in \{1, 2, 3\}$  3 rows of  $A$

Five checks act on each column of 3x3 array

Each check weight  $\leq 3$

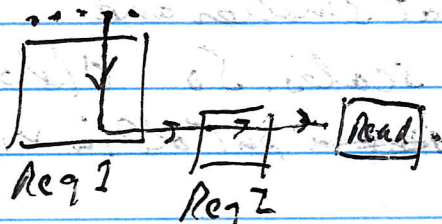
$H_x$  has 15 checks, each supported on one row of Register 1 array and one column of Register 2 array.



In principle, a mobile ancilla qubit could interact with a row of Register 1 and a column of Register 2 before proceeding to readout zone

We also need to measure

$$H_z = (I_5 \otimes A, A^T \otimes I_3)$$



Each mobile ancilla interacts with a column of Reg. 1 and a row of Reg. 2

29 Apr 2026

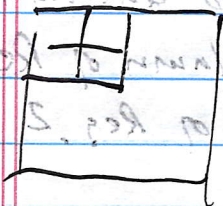
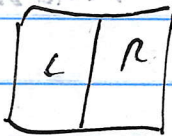
②

There are further considerations in scheduling. We want ancillas to be next to target data qubits when Rydberg laser flashes. We want to extract syndrome data with high parallelism. We want to minimize movement times to the extent possible, since movement dominates the time for syndrome extraction.

Some helpful ideas - Xu et al. 2308.08648.

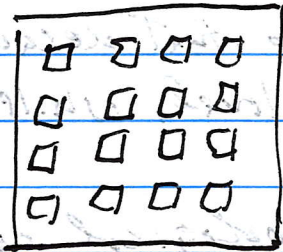
① Coloration scheduling. Each data qubit and each ancilla qubit interacts with only one other qubit at a time. For both  $H_x$  and  $H_z$ , we color the edges of the bipartite Tanner graph so that each qubit is connected to others via edges of different colors. The number of colors is max degree of bipartite graph. In each time step, gates are performed on pairs connected by edges of a specified color.

② Reduce number of moves via recursive "divide and conquer" movement algorithm. E.g. given target locations, first move across center line so each qubit is in targeted L or R rectangle. Do the same for U and D. Then



repeat at smaller and smaller scales. So  $O(\log n)$  moves ~~can~~ put all in right position

29 Apr 2026  
PH/CS 219 (3)



In 2D codes each vertex of the base graph becomes a fiber of  $L$  qubits, arranged e.g. in a square array, perhaps with vacancies.

Monomial  $x^p$  at each entry of  $A$  specifies which of the  $L$  qubits participates in each of  $L$  checks arising from each row of  $H_x$  or  $H_z$ . That needs to be included in the "tweezer dance". Optimizing the movement algorithm will be an important element of operating a neutral atom FTQC at scale.

### Syndrome Decoding

An important part of FTQC is the classical algorithm for syndrome decoding, which must be robust against errors in syndrome bits. In particular, we wish to verify that there is an "accuracy threshold" such that when error rate in 2Q gates is less than  $\epsilon_0$ , the prob of a logical error decays exponentially with increasing block size and distance. This requires a circuit level analysis of how errors propagate, including for gadgets that execute universal gates on logical qubits.

For ~~concatenated~~ concatenated codes, verifying the threshold is relatively simple, at least conceptually. We can analyze decoding performed "level by level".

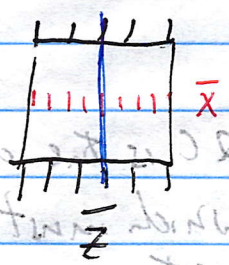
WCS rAPS  
 (3) PLS 2014

129 Apr 2026 (4)

There is an effective error rate that decays double-exponentially with encoding level, while the code length expands exponentially.

For other codes like topological codes and high-rate codes, we need different methods. To get started, consider our favorite example: the 2D Toric code. For now, consider only memory, not computation, and "phenomenological" noise rather than circuit-based noise.

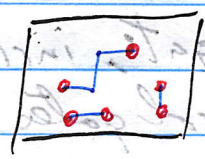
Consider the planar unrotated surface code,  $L \times L$  with qubits on edges.



The distance is  $d=L$ .

Min weight logical ops are strings,  $\bar{Z}$  running from top to bottom,  $\bar{X}$  running from left to right.

Errors define a  $\mathbb{Z}$ -chain,  $E$ , the set of edges where errors occur. Suppose at first that syndrome information is perfect. Consider  $Z$  errors and  $X$ -type stabilizer generators. The syndrome



is a zero-chain, a set of sites  $S$

where  $X_{site} = -1$ . Hence

$$S = \partial E \quad \text{syndrome } 0\text{-chain}$$

is boundary of error  $\mathbb{Z}$ -chain

The decoding algorithm specifies a way to return to the code space, a recovery chain  $E'$  such that  $\partial E' = S$ .

29 Apr 2026 (5)

Then  $E + E'$  is a cycle, with trivial syndrome.  
A noise model assigns a probability to each possible error 1-chain  $E$ , denoted  $\text{Prob}(E)$ .

Once the syndrome is known, choose a canonical  $E_0$  such that  $\partial E_0 = S$ . The general 1-chain with boundary  $S$  is

$$E_0 + C \text{ where } C \text{ is a cycle, } \partial C = 0 \quad (\text{Cycle is defined relative to boundary})$$

All cycles in the same homology class are equivalent. If  $C$  and  $C'$  are both in class  $h$ , then  $C + C'$  is in trivial class - that is can be "tiled by plaquettes" and hence is in the stabilizer.

Given syndrome and noise model, we can assign a probability to each homology class

$$\text{Prob}(h | S) = \frac{\sum_{C \in h} \text{Prob}(E_0 + C)}{\sum_C \text{Prob}(E_0 + C)}$$

Denominator = Prob of syndrome  $S = \partial E_0$ .

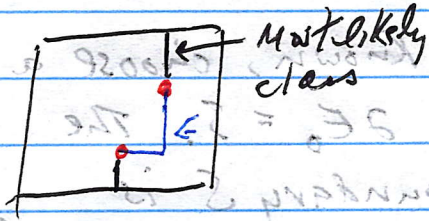
Numerator = " " " " and  $E_0 + E' \in h$  where  $E' = E_0 + C$ .

Optimal recovery: choose  $h$  to be the most likely homology class (and recovery chain to be any representative of that class).

Then the prob of a decoding failure is

$$\text{Failure Probability} = \sum_E \text{Prob}(E) \text{Prob}(E + E_0 \notin h_{\max} | E)$$

Prob( $E$ ) is determined by noise model, and the other factor is <sup>prob.</sup> that  $E$  is not in the most likely class given syndrome  $S = \partial E$ .



If noise is i.i.d with error probability  $\epsilon$ , then

$$\text{Prob}(E) = \epsilon^{|E|} (1-\epsilon)^{n-|E|}$$

Computing the most likely homology class can be

expensive — it is more feasible to find the most likely error  $\mathbb{Z}_2$ -chain! Which for the i.i.d.

Noise model is a  $\mathbb{Z}_2$ -chain  $E$  with minimal error weight  $|E|$  given the observed syndrome.

This is done by the minimal-weight-perfect-matching algorithm (relative to the boundary).

This works pretty well, but might not be optimal, because the most likely error might not belong to the most likely class (stat mech analogy: this is like minimizing the energy,

ignoring entropy — we should minimize free energy instead. Dennis et al. quant-ph/0110143.)

For the i.i.d. model, the probability of a decoding failure is the probability that the actual error  $E$ , and the most likely error  $E'$  satisfy

$$E + E' = C \in \text{non-trivial homology class}$$

29 Apr 2026 (7)

where the most likely error is determined by Min Wt Perfect Matching (MWPM).



For a  $Z$  error in unrotated planar surface code, this is the probability that  $C = E + E'$  contains an odd number of "strings" connecting the bottom and top boundaries.

There is at least one such path in  $E + E'$ , which has length (# of edges)  $\geq L$ . Furthermore  $E$  must contain at least half of the edges in this path, otherwise, we could reduce the weight of  $E'$  by choosing  $E$  instead, which has the same boundary. Because MWPM found  $E'$ , that can't be the case.

Denote this length- $l$  path by  $C_l$ . The number of errors in  $E$  on this path is at least  $l/2$ , and the number of ways to choose which edges in  $C_l$  are contained in  $E$  is at most  $2^l$  (each of  $l$  edges either contained or not). Hence prob  $C_l$  contained in  $E + E'$ :

$$\text{Prob}(C_l) \leq 2^l \epsilon^{l/2} = (4\epsilon)^{l/2}$$

Let  $N_l = \#$  of connected paths of length  $l$  connecting top and bottom boundaries.

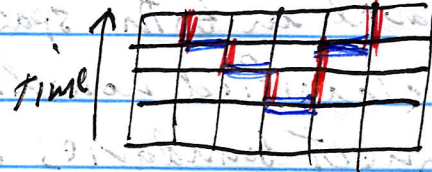
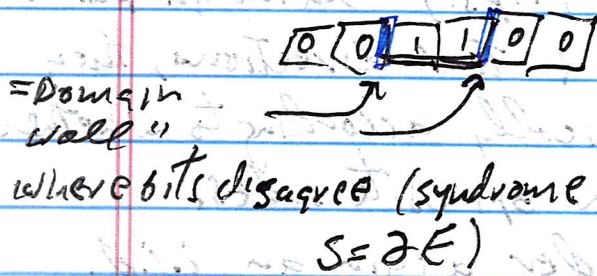
Union bound  $\Rightarrow$

$$\text{Prob}(\text{failure}) \leq \sum_{l=L}^{\infty} N_l (4\epsilon)^{l/2}$$

Now we upper bound  $N_l$ .



To visualize the syndrome history in spacetime, add a dimension representing time. Consider the easier case of the 1D repetition code. Bits are on plaquettes, checks are on vertical edges. Errors that occur between syndrome measurement rounds (where a bit flips) are indicated as horizontal edges.

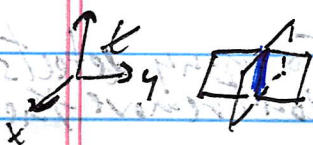


Bit flips cause syndrome to hop

Now, if syndrome is perfect (no meas errors)  $S + E$  has no boundary

If the syndrome has errors, let's consider  $E$  to contain syndrome errors as well as data bit-flip errors, so that  $E$  becomes a  $\mathbb{Z}$ -chain containing both horizontal and vertical edges, and let  $S$  be the observed syndrome  $\mathbb{Z}$ -chain (vertical edges only). The observed syndrome + syndrome errors is the same as ideal syndrome. Therefore it is still true that  $S + E$  has no boundary when  $E$  contains syndrome errors and  $S$  is the observed syndrome.

For the 2D Toric code, the syndrome history is 3D. Again consider qubits on plaquettes. In  $xy$  and  $yz$  planes



syndrome on vertical edges where 4 such plaquettes meet.

Planes

data qubit

Again errors between rounds indicated on horizontal edges, causing ideal syndrome to hop. Again  $\partial(S+E)=0$  where  $S$  is observed syndrome history and  $E$  includes syndrome measurement errors.

A phenomenological noise model assigns a probability to each error  $\mathbb{Z}$ -chain (including data flips and syndrome flips.) Although paths now travel in both space + time directions, they can be classified homologically according to whether they connect boundaries on opposite sides.

For simplicity consider again an i.i.d. noise model. In principle even probs on vertical (syndrome) edges and horizontal (data) edges could differ, but in practice they are comparable so suppose  $\epsilon$  for both.

We can apply MWPM to the spacetime syndrome  $S$ , and bound the probability that  $E+S$  contains a path of length  $L$  or greater.

The analysis is as before except for two changes.

- the number of paths of length  $L$  scales linearly with elapsed time  $T$ , so we are estimating the prob of logical error per unit time (per syndrome measurement round)

- the path has 4 ways to turn so # of ways to add an edge is 5 instead of 3  $\Rightarrow 5(4\epsilon)^{L/2} = (100\epsilon)^{L/2}$  and

$\epsilon_0 = .01$ . MC simulations show  $\epsilon_0 \approx .0293$

- we'll want  $T \gg L$ . otherwise too many defects get matched to the future boundary & we don't remove them.