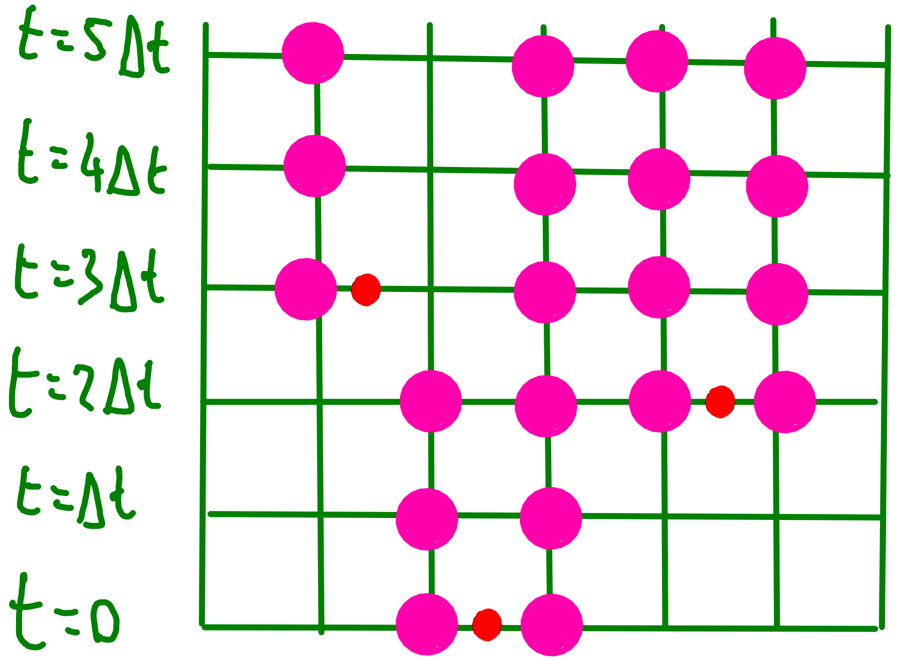
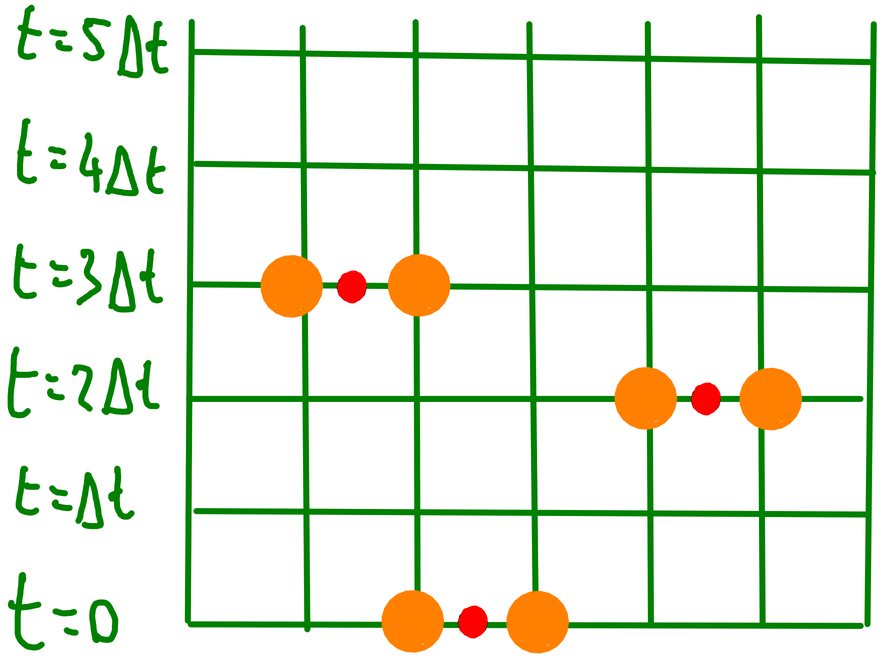
# Perfect syndrome measurements

At each time slice, the syndrome tells us the anyon config at that time

By looking at the anyon config, and comparing it for different times, we want to be able to determine which errors occured

To do this we look at the change in the anyon config



The time slices then correspond to independent decoding problems, each equivalent to the one shot case

Error model is (for charge syndrome)

$$\mathcal{E}(\rho) = \sum_j (1-p)\rho + p \; \sigma_z^j \rho \sigma_z^j, \qquad p = p_z(\Delta t) + p_y(\Delta t)$$

The exponential suppression of the logical error rate in the one-shot case then become an exponentially long lifetime for continuous error correction

$$P(\Delta t) = e^{-\alpha(p) L^\beta} \implies P(T) < \frac{T}{\Delta t} e^{-\alpha(p) L^\beta} \qquad \therefore T(P) = P \Delta t \, e^{\alpha(p) L^\beta}$$

The time before the logical error rate reaches some unacceptable value can be made arbitrarily long by increasing the code size

# Noisy Syndrome Measurements

What about the more realistic case, where syndrome measurements are noisy

Simplest way to model this is to simply say that there is some probability, q, that the measurement lies

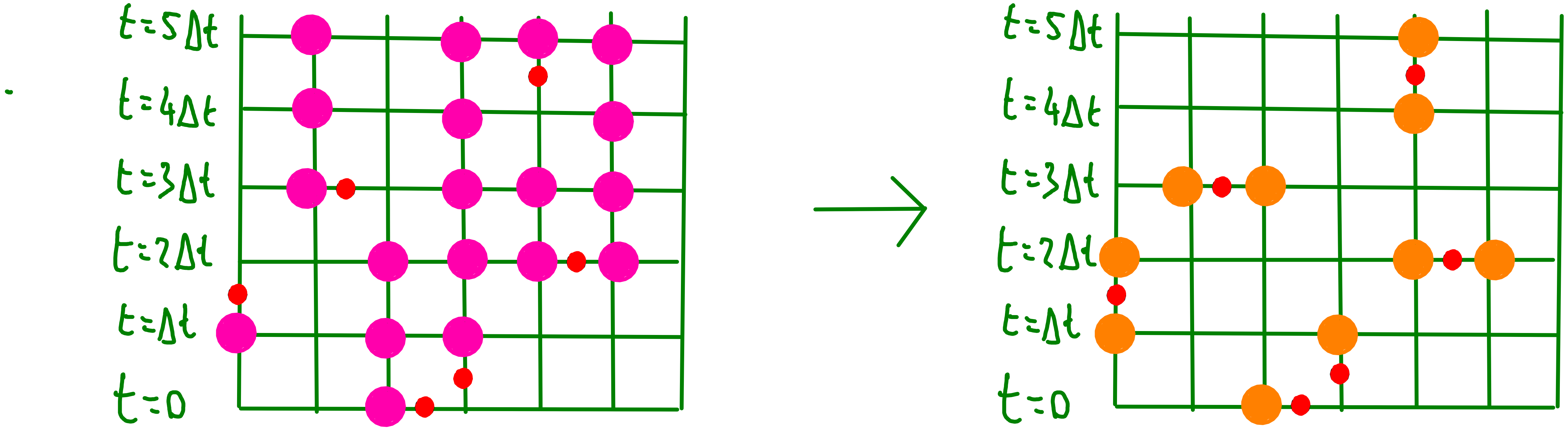
- tells us there's an anyon when there isn't
- tells us there isn't when there is

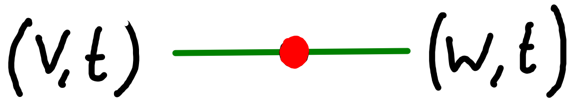
For $q>0$, we can no longer use the same methods as for $q=0$

Attempting to correct anyons that aren't really there, and not correcting ones that we cannot see, will lead to a high probability of logical errors.

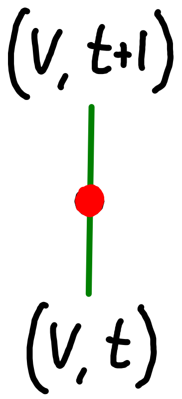
So how do we deal with measurement errors?

Exactly the same as before: look at changes in syndrome




This denotes a z error on the qubit between vertices v and w between rounds t-1 and t. This will cause these vertices v to give unexpected results at measurement round t.




This denotes that the measurement of vertex v at time t gave the wrong value, and therefore the result is unexpected.
Since the false result is expected again at t+1, but result will actually be the true one (without further errors), this will also be unexpected.

Again we find that errors create pairs of 'defects' (we can't really call them anyons any more)

Both errors on qubits and measurement errors do this

We no longer have many independent 2D syndromes

Instead we get a 3D syndrome of defect pairs

Probability of errors on space-like edges given by $p_z + p_y$

Probability for time-like edges given by $q$

The example here looks at the vertex defects, but there is also an equivalent 3D syndrome for plaqutte defects

Probability for errors on space-like edges in this case is

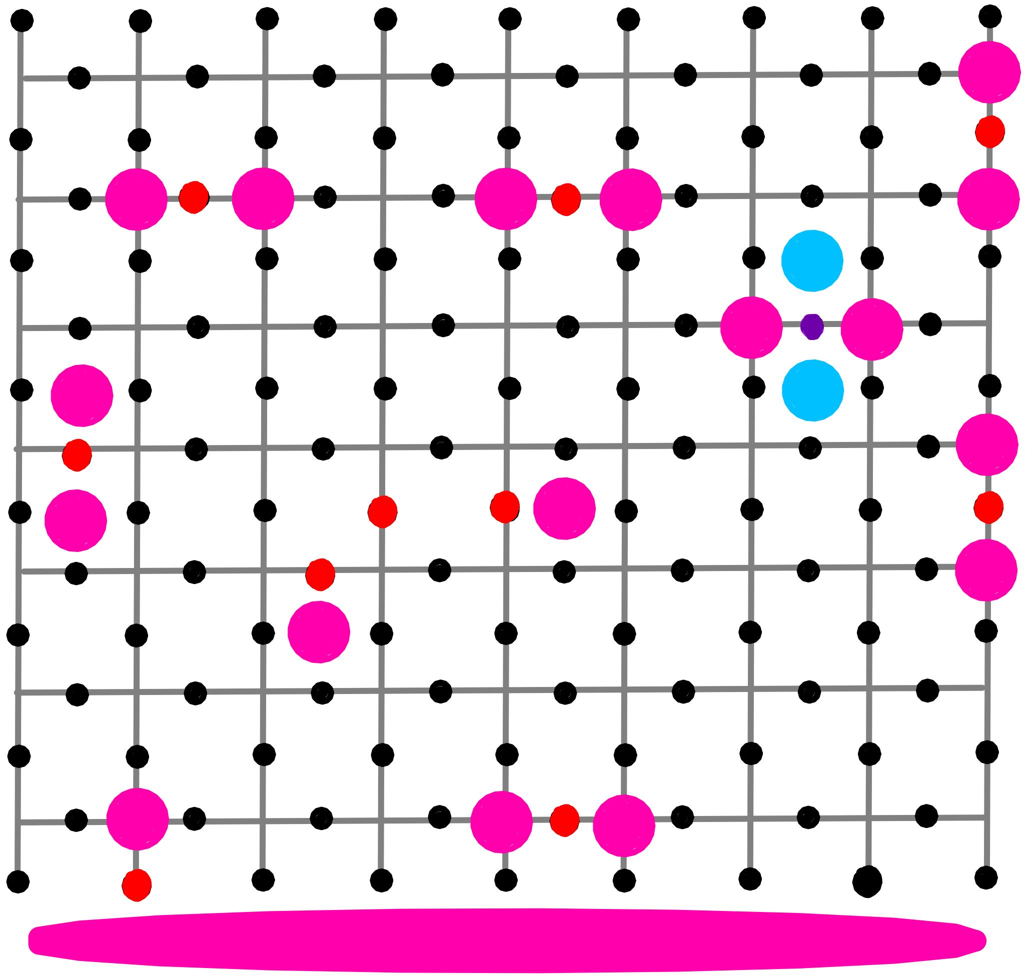
$p_x + p_y$

# Time boundaries

For e anyons (created by z errors) we know that the left and right boundaries are 'hard' and the top and bottom are 'soft'.

This means we can have isolated e anyons on the top and bottom edges, because the edge can absorb the antiparticle.

But for left and right edges, this can't be done



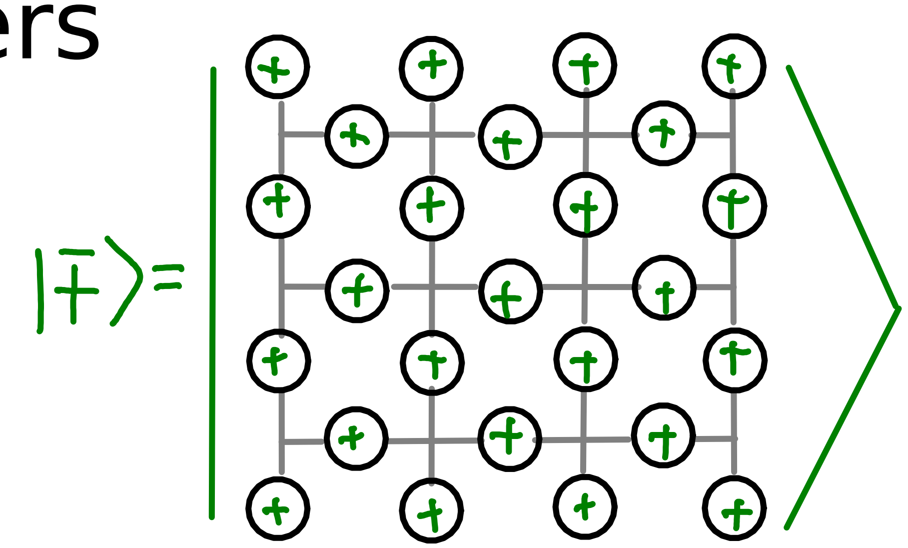
For our 3D syndrome of defects, we have a new dimension, and hence new edges: initial and final

Whether these edges are hard or soft depends on initial and final conditions

Consider a qubit initialized in the + state (+1 eigenstate of logical X), and finally measured later in the X basis

Can be initialized using a product state that is

- +1 eigenstate of all vertex stabilizers
- +1 eigenstate of logical X



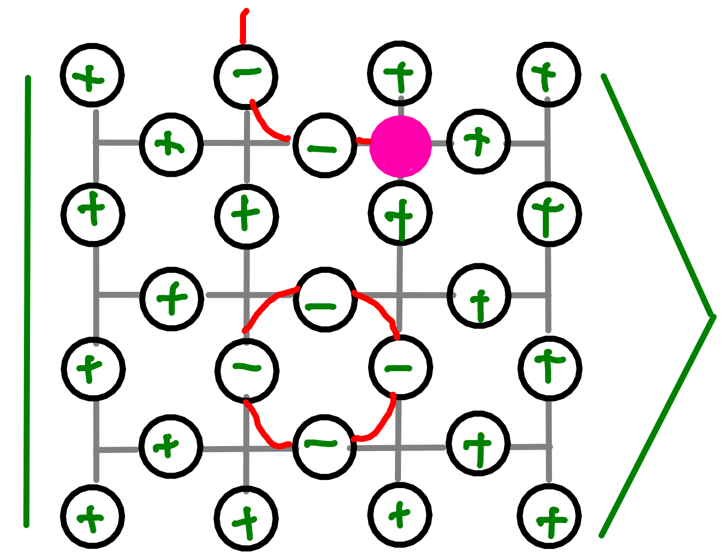
Both A and B stabilizers then measured in usual ancilla assisted way for many rounds

Given that we know the initial value of the A stabilizers with certainty (all +1), the bottom time boundary is hard to vertex defects

But it is soft to plaquette defects, because the initial value for these stabilizers is random

When we want to measure X, we measure all code qubits in the x basis

Result will be something like



With this info we can infer the X eigenvalue the number of -'s on the top line

We can also infer one last syndrome for the vertex stabilizers (the ones needed to correct X)

These measurements will also be noisy (lie with prob. q), but we can imagine them as perfect measurements following bit flips with prob q.

Since measurements are 'perfect', there will be no time-like edges to cross the top boundary. So it will also be hard.

Note that no vertex measurements can be made in this round. The last round of vertex measurements are the normal faulty ones. So the final time boundary will be soft for vertex defects.

What does this imply for logical operators?

Minimum logical X must place a plaquette defect on the top boundary (on which we have defined logical X), and leave no trace on the bulk.

Since the only other soft boundary is on the right, this requires us to act on a chain of qubits from top to bottom.

Code distance (for Z errors) is therefore still L

Logical X errors, however, could take a short cut into the soft final boundary

But it is only soft because we are making an X measurement
Since X errors would not affect the result, their effect is trivial

Since L paulis must be applied to L qubits to make a logical operation, we need errors chains of length at least L/2 to fool a decoder

MWPM can still be used, just in 3D instead of 2

This gives more possible error paths, and hence a lower threshold

$$p < \frac{1}{(2 \times 5)^2} = 1\%$$

# Finding the threshold

For a given decoder, it is always good to prove a finite lower bound for the threshold

But usually these are quite far from the true threshold that the decoder corrects up to. Typically these must be found numerically

Under threshold we expect

$$P \to 0 \quad \text{as} \quad L \to \infty \qquad \text{for} \quad p < p_c$$

Above threshold we have little idea of what's going on. Decoding using a 'good' method is not much different than a 'bad' method, like just throwing all m anyons off the right edge and all e anyons off the bottom.

The probability that this causes a logical error increases with system size before converging

$$P = \text{Prob}(\text{odd } \# \ \sigma_z \text{ errors on top line}) = \frac{1-(1-2p)^L}{2}$$

Using this behaviour we can determine the threshold numerically:

1. generate a random error, E, according to the error model (a list of what error happened on each qubit)

2. determine the corresponding syndrome, S

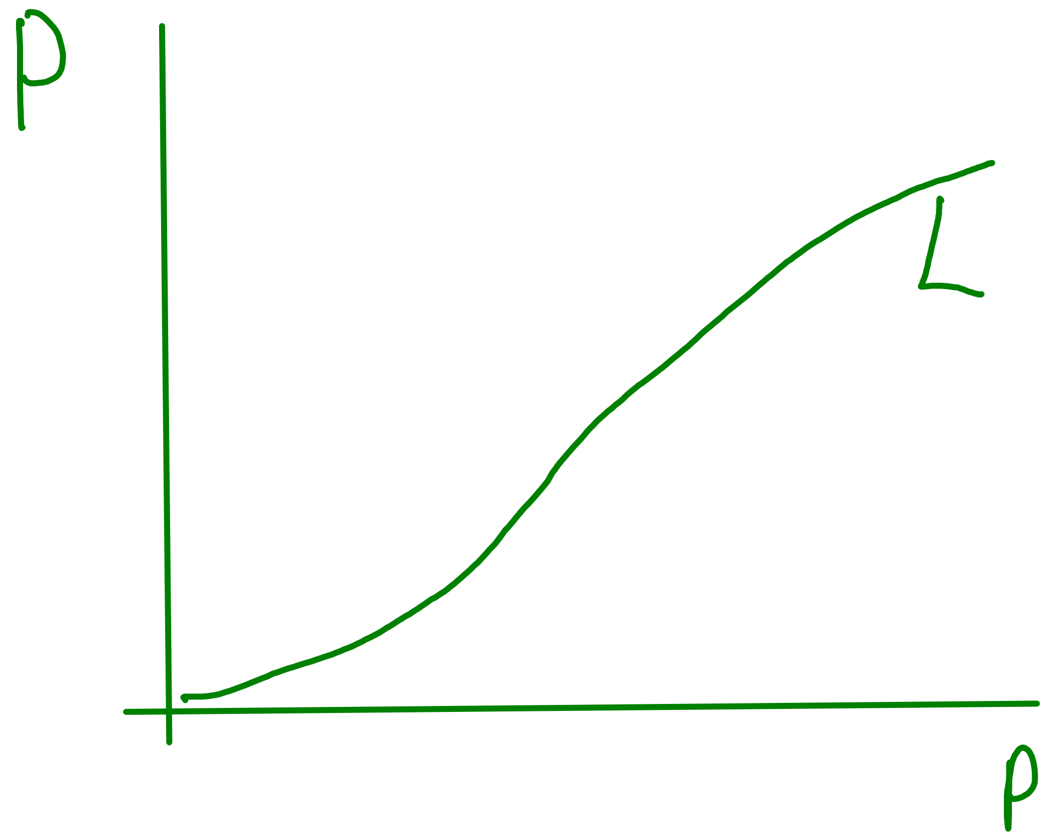
3. run the decoder to determine the corresponding correction operator $E_c$

4. Repeat for many samples, N, and count the number of times the decoding fails (when $E_c E \sim Z$)

$$P = \frac{\#\text{failures}}{N}$$

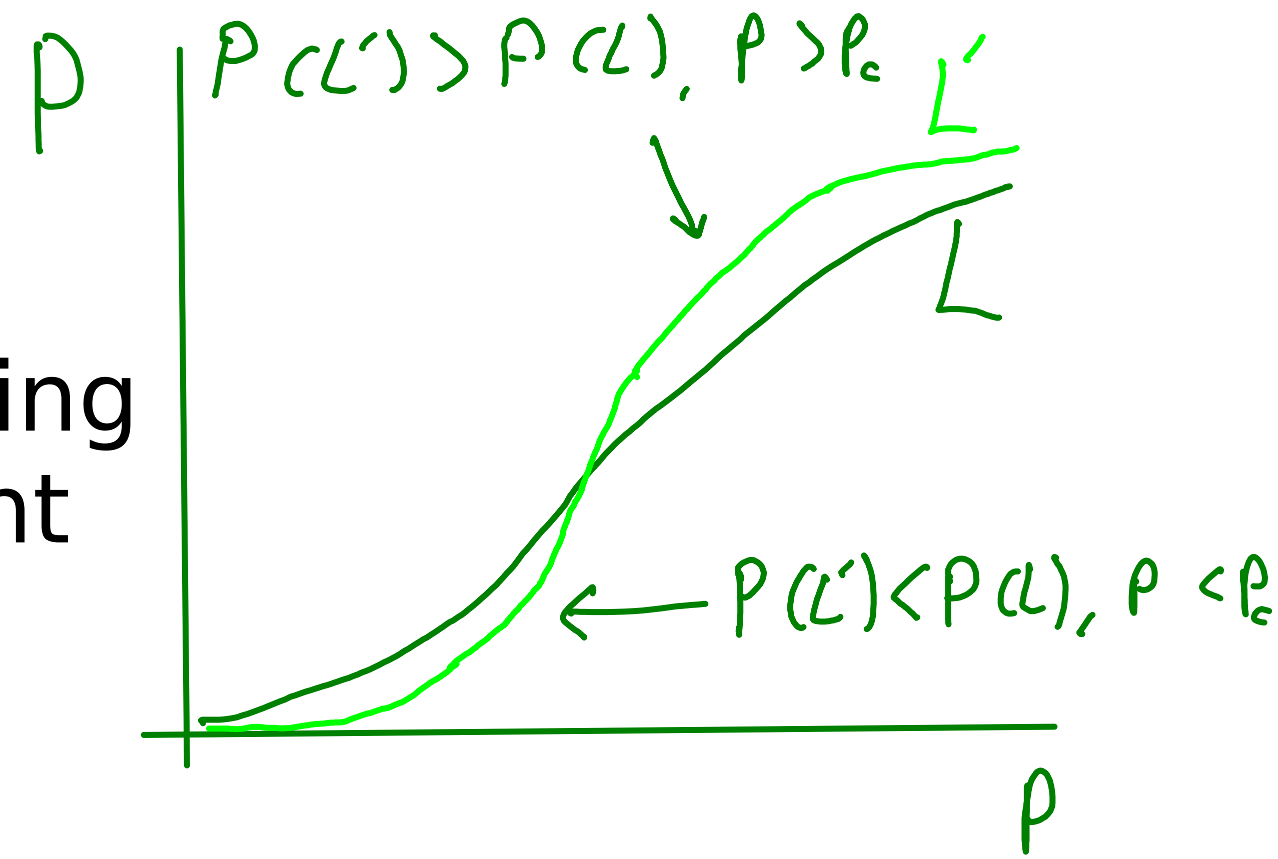
Determine this for many different $p$ and $L$

Plot $P$ for different $\rho$ and a single L

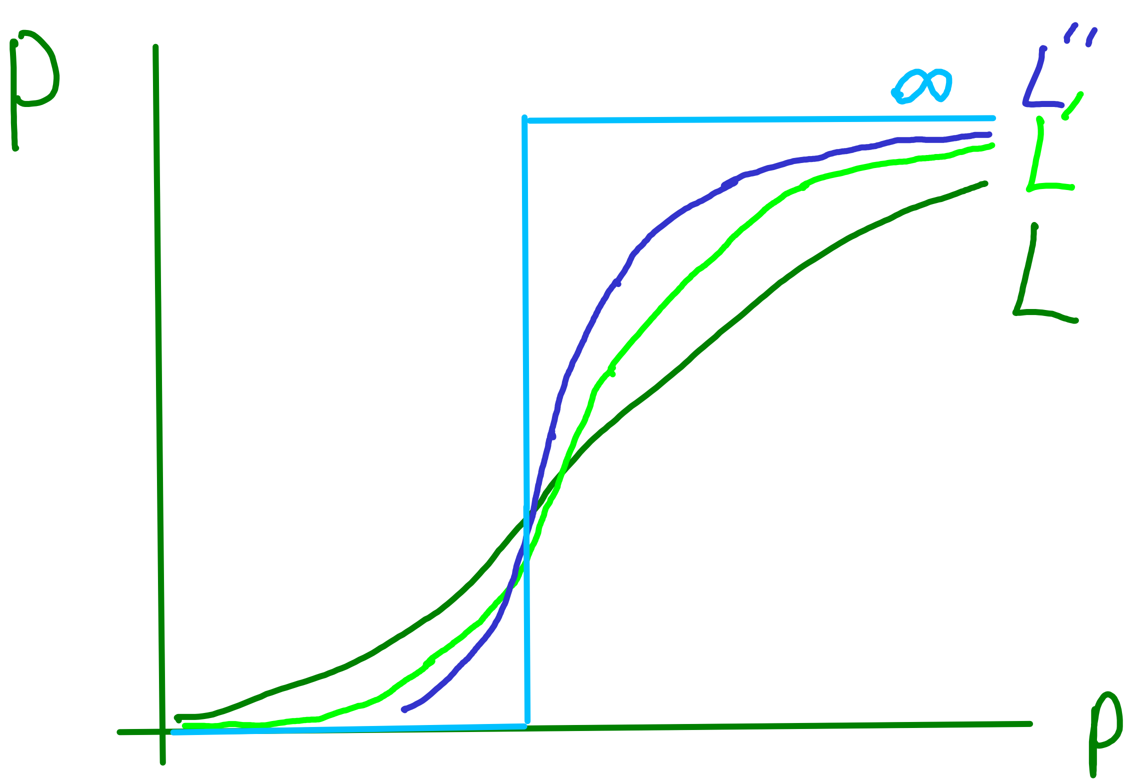



Increases monotonically, but tells us nothing about the threshold

Plot also the results for L'>L




We can expect $\rho_c$ to be the crossing point, but finite size effects might mean it is not correct




So we add a few more, and maybe analyze how the crossing scales with L to find the true $\rho_c$

For MWPM and perfect measurements, the threshold is

$$p_c \approx 10.5\%$$

This is much better than the bound of 1/36 that we proved

However, it is worse that the threshold we would get from brute force. This is

$$p_c \approx 11\%$$

Including measurement errors (q=p) typically causes these thresholds to lose an order of magnitude