Error Correction for non-Abelian anyons

James R. Wootton, Jan Burri, Daniel Loss
Universität Basel

Sofyan Iblisdir
Universitat de Barcelona

Phys. Rev. X 4, 011051 (2014)

Adrian Hutter, Daniel Loss, James R. Wootton
Universität Basel

arXiv:1410.4478

The fusion space

Any anyon model is first defined by the list of **particle types**

$$\{1, a_1, a_2, ...\}$$

Two (or more) anyons can be **fused** into a single composite. Particle type of composite given by **fusion rules**

$$a_i \times a_j = a_k \qquad\qquad e \times e = 1$$

For **non-Abelian anyons**, result of fusion depends on more than just local degrees of freedom (such as particle types of components)

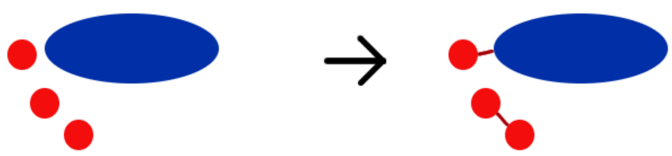
$$a_i \times a_j = a_k + a_l + \ldots$$

Depends also on non-local degrees properties of many-anyon states:
The fusion space

Naturally a good candidate to store quantum information: **quantum memory**

Decoding non-Abelian quantum memories

Abelian model with $e \times e = 1$ fusion rule.
information stored in anyon occupancy of
hole (blue)



Best operator for correction is unambiguous:
Errors moved an anyon into the hole, so move
antiparticle in to cancel the effect

Non-Abelian model with $\tau \times \tau = 1 + \tau$
information stored in fusion outcome of
computational anyons (blue)

Best operator for correction is ambiguous:
Same probability that errors fused with
Computational anyons and didn't!

If method to remove anyons depends on their positions alone, it will cause a logical error with high probability

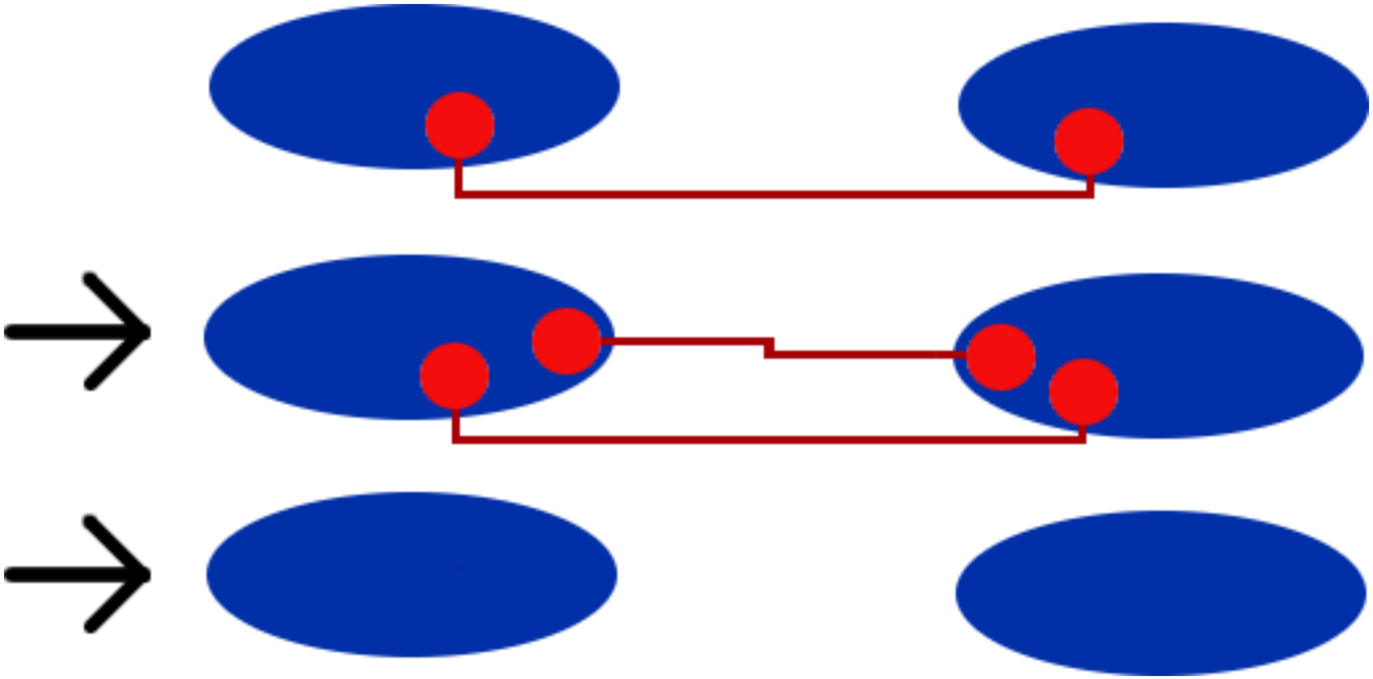
Anyon configuration is not sufficient syndrome information for error correction

We must measure the fusion space!

Decoding non-Abelian quantum memories

Another danger: **uncorrectable logical errors**

In the Abelian case, any known logical error can be corrected



Just send in the antiparticles!

For the non-Abelian case this doesn’t have a uniquely defined result

It will not always undo the error, and will not tell us if it has

Can cause problems if we fuse the wrong things

Decoding algorithm

To obtain enough information to decode, **fusion measurements** must be made

Pairs (or clusters) of anyons should be combined and the composite particle type determined

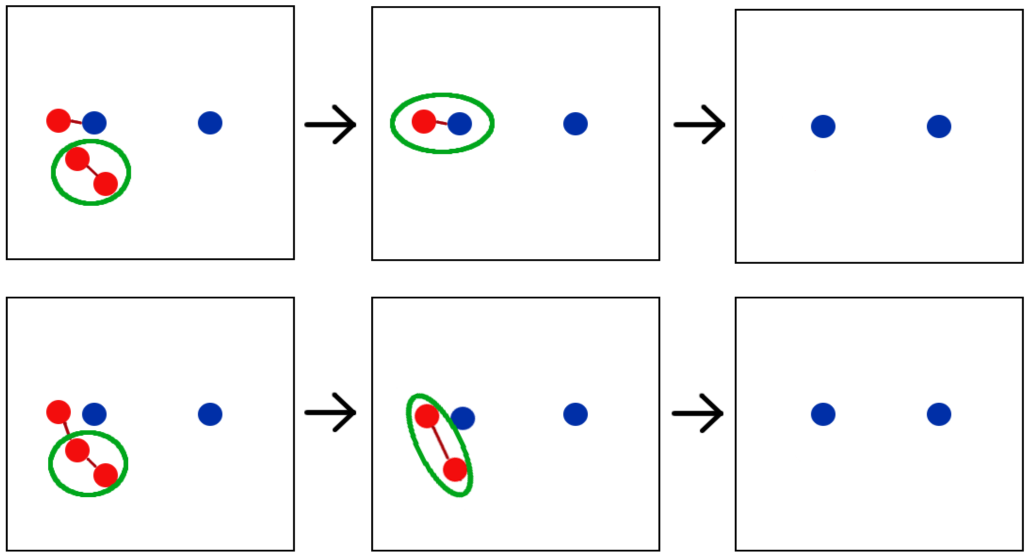
Doing this blindly can lead to **uncorrectable errors**!

The fusion process is also **irreversible**

Best candidate: pairs created by the same error chain

So the decoder needs to:

1. Find clusters of anyons
2. Fuse them
3. Repeat until all have annihilated



Known HDRG Decoders

We already know decoders that work like this, even for the Abelian case

Each decoder defines a search distance $D(n)$, depending on the round of iteration.

Bravyi-Haah 2011

- $D(n)=2^n$, (L_∞ norm)
- Clusters consist of connected components of the syndrome

Anwar-Brown-Campbell-Browne 2013

- $D(n)=n+1$, (combination of L_1 and L_∞)
- Clusters consist of connected components of the syndrome

Expanding Diamonds (Dennis Thesis)

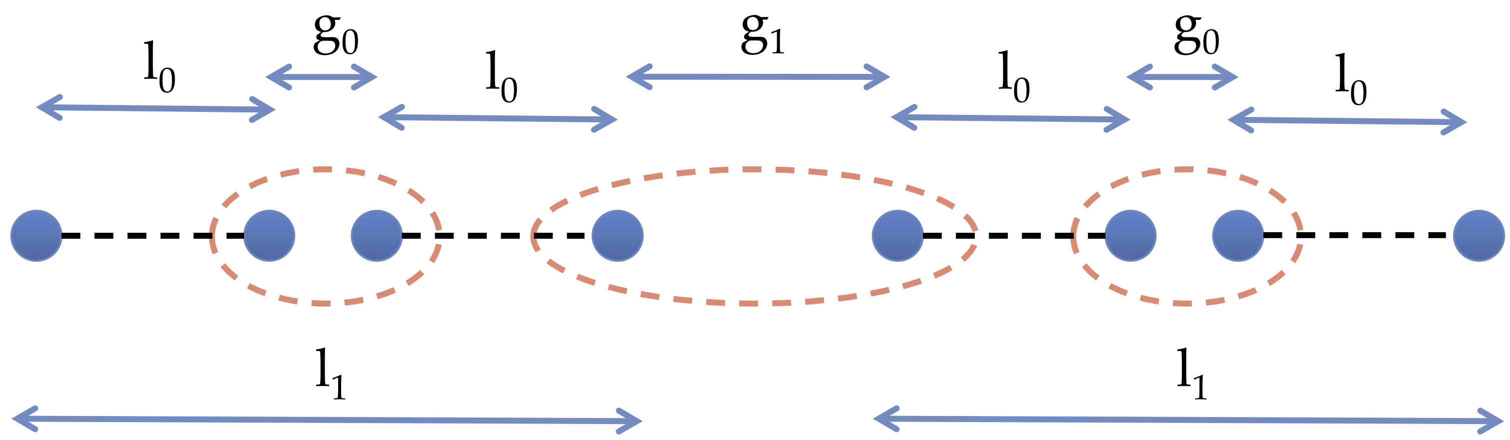
- $D(n)=n+1$, Manhattan distance (L_1 norm)
- New clusters are built out of pairs of mutual nearest neighbors of previous clusters

These all have an additional property:

Clusters that fuse to vacuum (neutral clusters) are completely forgotten about once they are identified.

HDRG Decoders

Let's look at the kinds of error that are a problem for these decoders



Anyons are continually paired incorrectly

They are then forgotten about

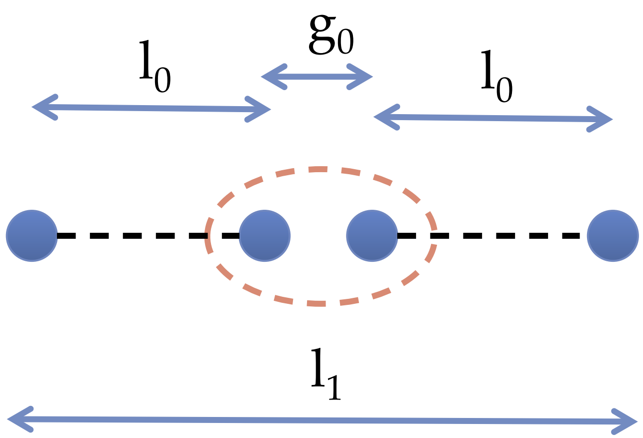
Structure required to fool ED and ABCB needs only $\sim L^{0.63}$ errors

For BH $\sim L^{0.67}$ errors are needed. Both fall far short of the linear ideal

Shortcuts

This might never have happened if we kept the full syndrome in mind

How do we do this such that the method remains efficient?



Distance between left and right anyons should reflect the number of errors needed to connect them

This is not $2l_0 + l_1$, it is $2l_0$

Distance should be allowed to take a 'shortcut' via the neutral cluster

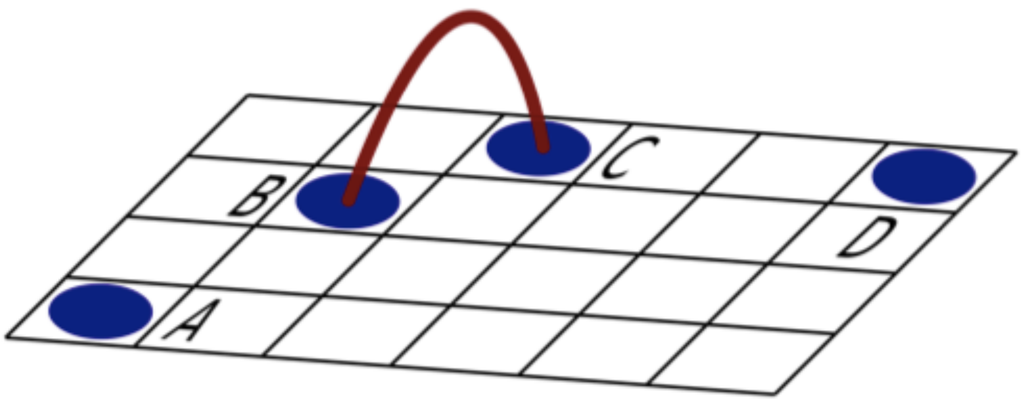
Shortcuts

When a neutral cluster is found:

- Remove it from the syndrome, but...
- Update distances between all other clusters according to

$$d_{jk} \to \min\left(d_{jk},\ \min_{c_m, c_n \in C} (d_{jc_m} + d_{c_m k})\right)$$

We effectively add wormholes to the lattice, to keep the memory of neutral clusters alive



These allow mistakes caused by false neutrality to be undone

Number of errors required then becomes $L^{1-\varepsilon}$

Matching based clustering

These shortcuts can be used along with your favourite clustering method

But can we develop a better one?

Now we can avoid false neutrality problems, we can try to extract as much syndrome information as possible

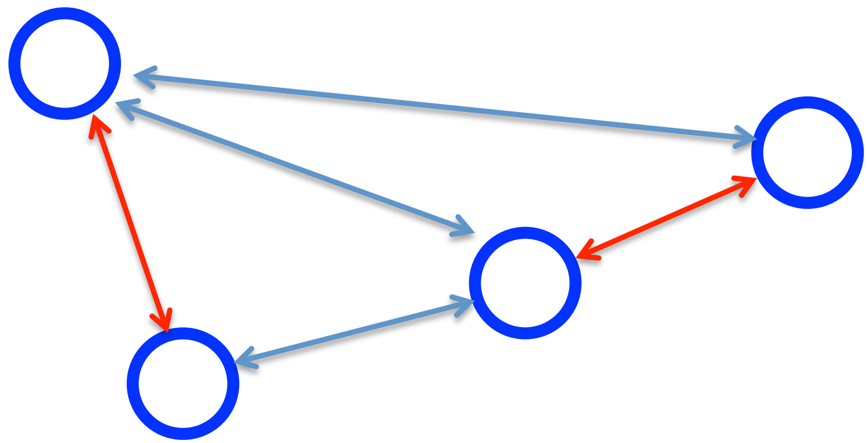
We can consider methods, like expanding diamonds, that only form new clusters out of two non-neutral ones

This sounds like a job for minimum weight perfect matching!

Matching based clustering

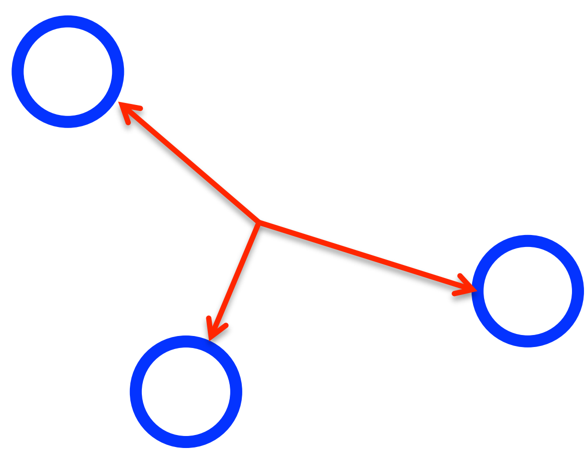
MWPM can be used when errors create pairs of anyons (Toric Code)

Each anyon pair is assigned a weight according the number of errors needed to create them



MWPM finds minimum weight pairing, and so minimum error chain

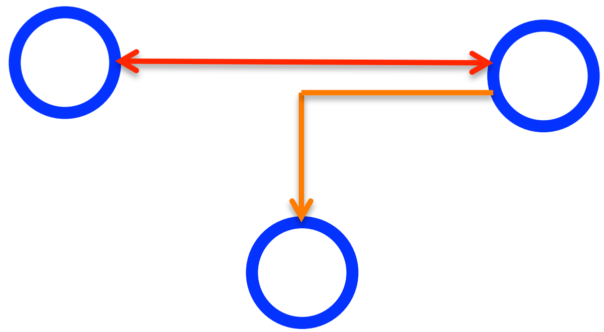
But this cannot be used for general anyon models



Matching based clustering: I

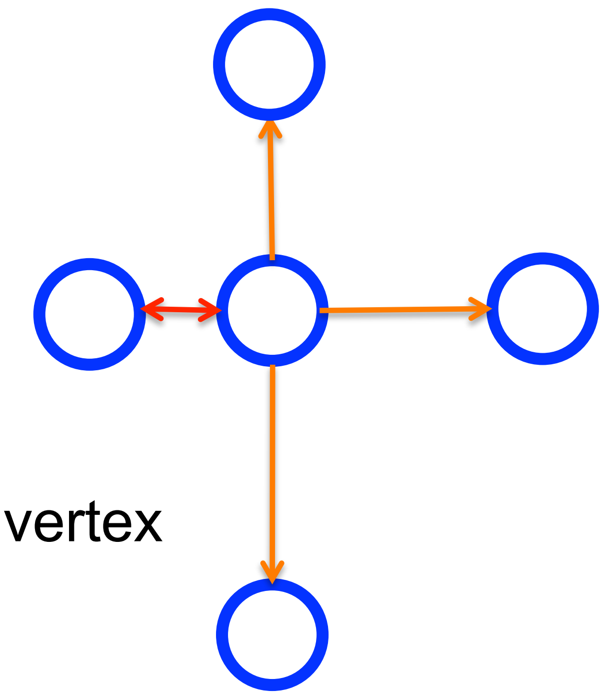
For simplicity, consider only error chains that split at anyons

Number of errors required for decays is too high in general, due to chains splitting only at anyons



Any such minimal error chain will be a star graph
- Single internal vertex
- Internal vertex is nn of all external ones

We can interpret these errors in terms of a single pair creation, followed by a set of decays



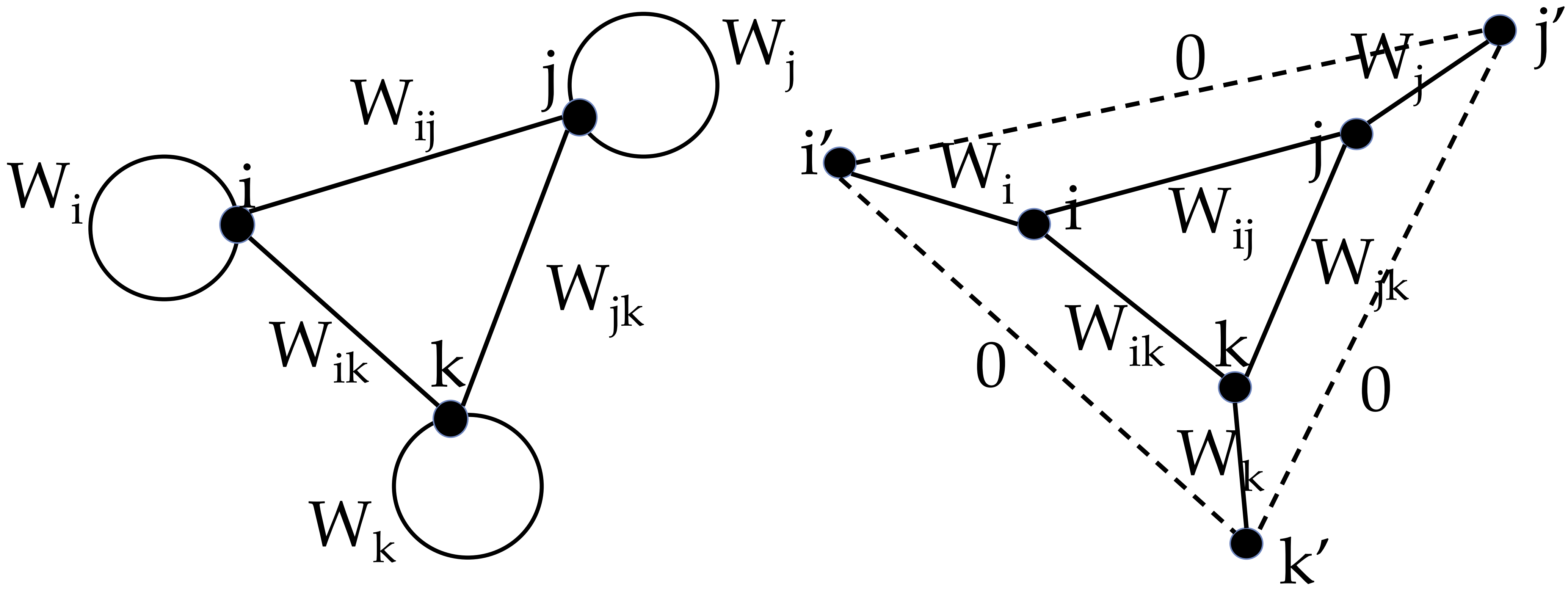
The pair consists of the internal vertex and one external vertex

Matching based clustering: I

Minimal error chains may be found using a minimum weight (non-perfect) matching problem, which can also be solved with Blossom

Anyons can be paired, with a pairing weight W_{ij}, or not paired with a weight W_j

For $W_j = d_{j,nn(j)}$, the non-paired anyons correspond to decays from nn's



Since W_j is too high in general, maybe a lower weight would work better

Matching based clustering: II

Good candidate for pairs: mutual nearest neighbours

$$d_{j,nn(j)} = d_{k,nn(k)}$$

If we use $W_j = d_{j,nn(j)}/2$, the matching will only pair mutual nearest neighbours

But method doesn't care if it gets pairs or not, and we'd like lots of pairs

Maybe it would be better to use a higher W_j, penalizing non-pairing

Together, the two methods suggest we use $W_j = k\, d_{j,nn(j)}$ for $0.5 < k < 1$

The value k is defined simply as the one that works best

We also include many degeneracy based terms in the weights

This decoder can also be used for decoding Abelian codes
See poster by Adrian Hutter later

Phi-Lambda Model

For further study, we need a concrete anyon model

We choose the $\Phi - \Lambda$ model:

Submodel of the universal $D(S_3)$ anyon model (Mochon '04)

Efficiently simulable

Has fusion behaviour similar to Fibonacci and Ising

$$\Phi \times \Phi = 1 + \Lambda + \Phi, \quad \Lambda \times \Lambda = 1$$

Fibonacci: $\tau \times \tau = 1 + \tau$

Ising: $\sigma \times \sigma = 1 + \psi$

Error model

Simulation is done on a spin lattice, based on the (Abelian) $D(Z_6)$ model

We consider a quantum memory based on these edge occupancies

Two types of error:

those that create Φ pairs
those that create Λ pairs

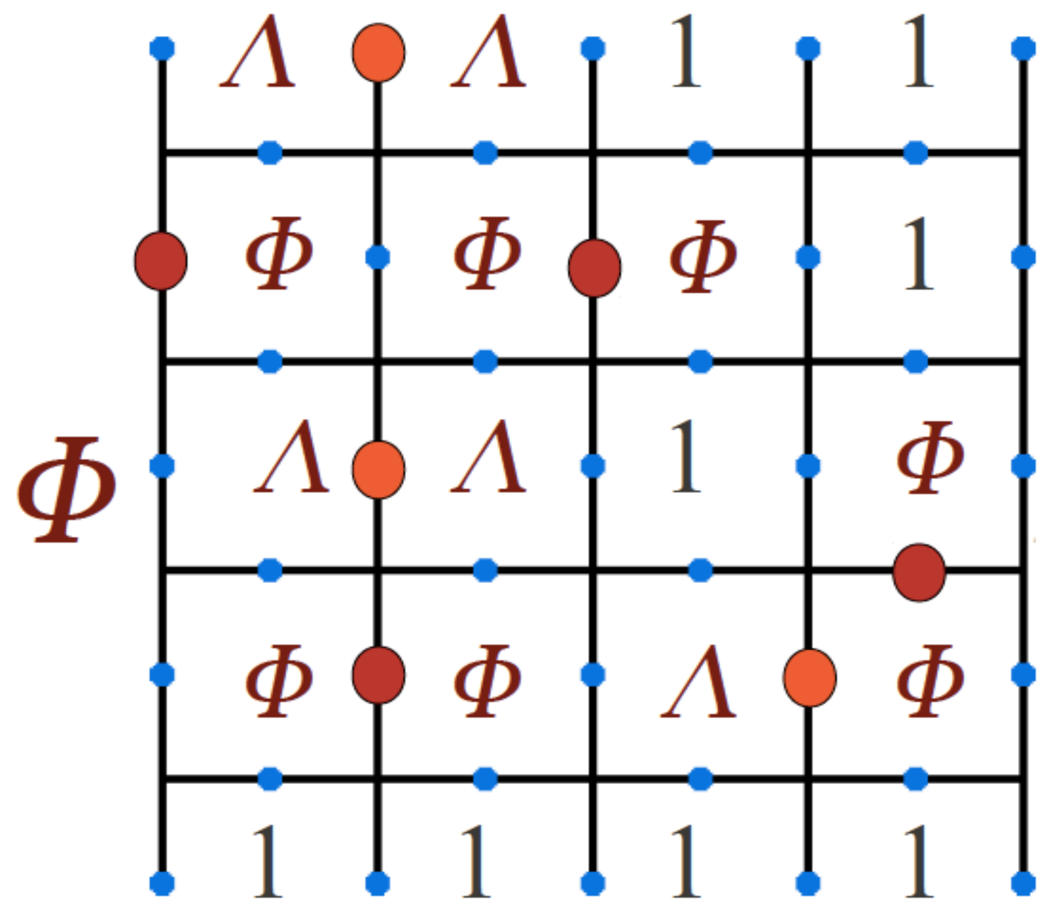
i.i.d. noise model with

$$p_\Phi = p_\Lambda = p/2$$

Measurements are assumed to be perfect

No noise acts during decoding

(Unrealistic, but first step)



Decoder first deals with Φ anyons, and then Λ anyons

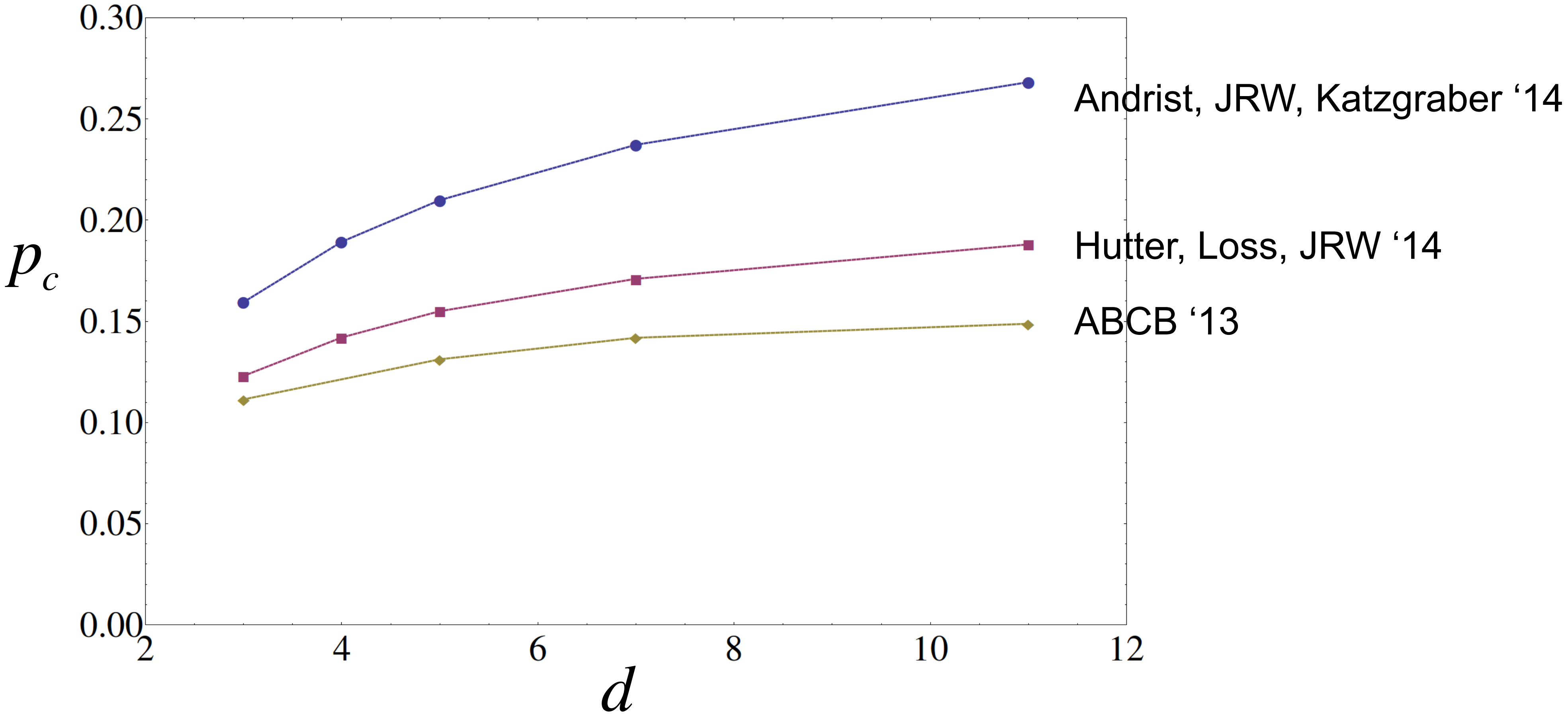
Results

With expanding diamonds (without shortcuts) we found a threshold $p_c \approx 7\%$

First indication that FTQC is actually possible using non-Abelian anyons

With our improved decoder we find $p_c \approx 15\%$

Similar improvements found for qudit codes



Outlook: Continuous Decoding

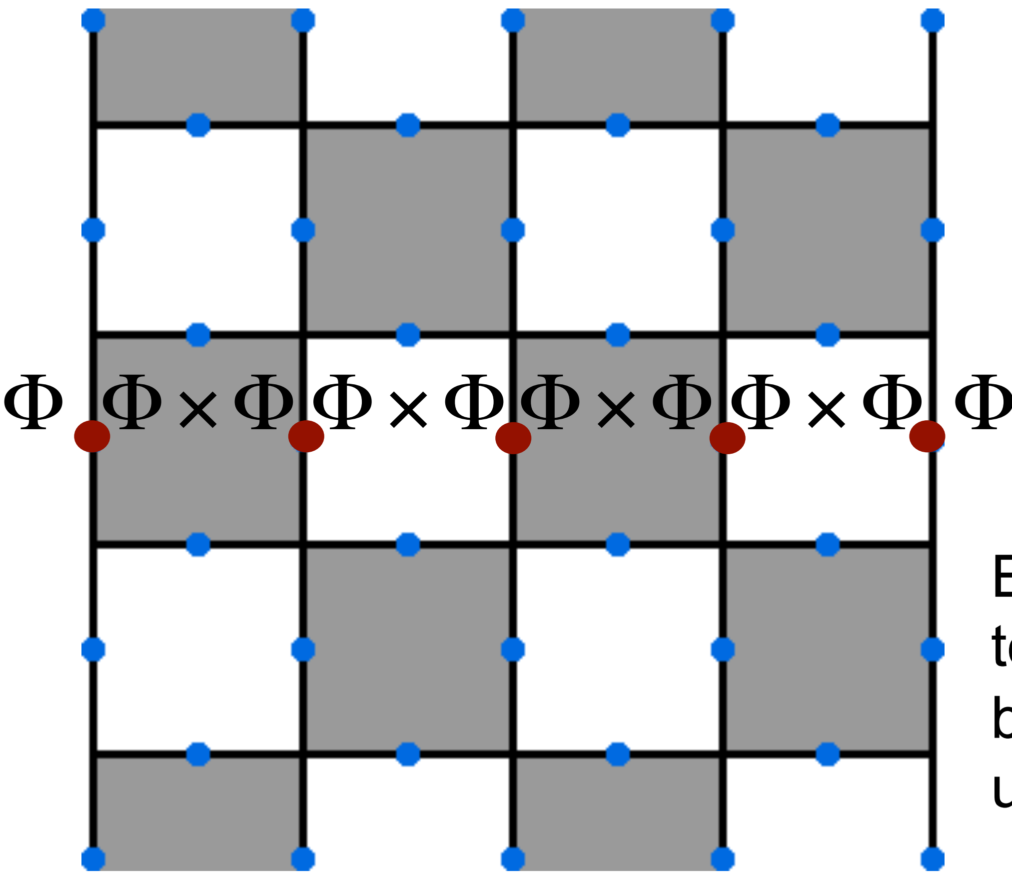
More realistic cases must be considered before we have a true **threshold theorem**

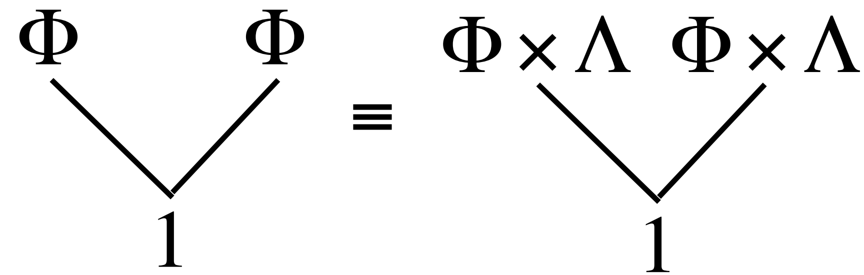
One important case is **continuous decoding**: code is measured periodically to keep errors suppressed for a long time

Especially important is the case of **measurement errors**

Trade-off between waiting long enough to do sensible fusions, and not allowing anyons to build up over time

Outlook: Continuous Decoding





Equivalent to Λ creating error applied to every error spin with probability ½ before the Φ creating errors are applied: uncorrectable

Thanks for your attention