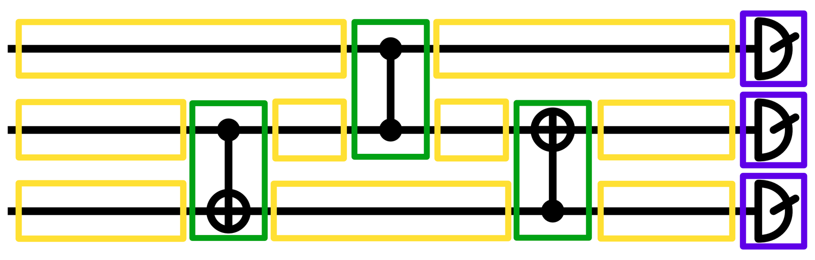
Reinforcement Learning Approach to Decoding for Fault Tolerant Quantum Computation

Ryan Sweke, Markus S. Kesselring, Jens Eisert
Freie Universitaet Berlin

Fault Tolerant Quantum Computation (FTQC)

Any quantum circuit can be rewritten such that only three types of blocks are needed:

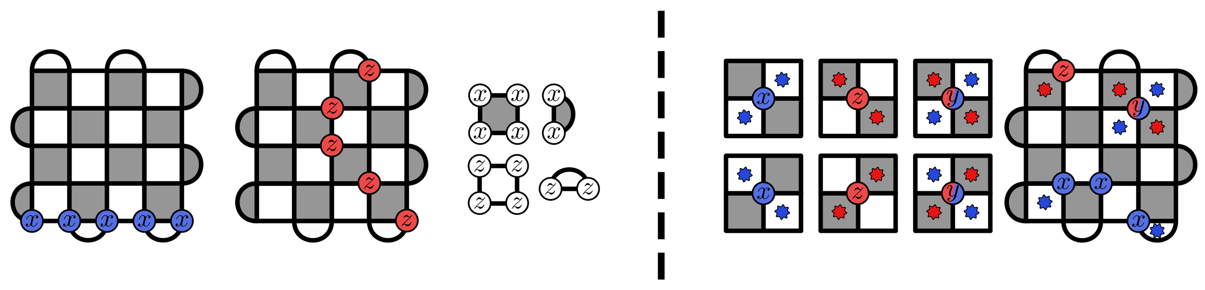
1. qubit idling/arbitary length identity gates (yellow)
2. controlled operations (green)
3. measurements (purple)



FTQC requires being able to implement all types of blocks in the presence of both noise and imperfect measurements.
Current strategy: Encode logical qubits in the ground state space of a quantum code and perform periodic decoding.
Green blocks, and yellow blocks followed by green blocks, are particularly challenging.

The Surface Code

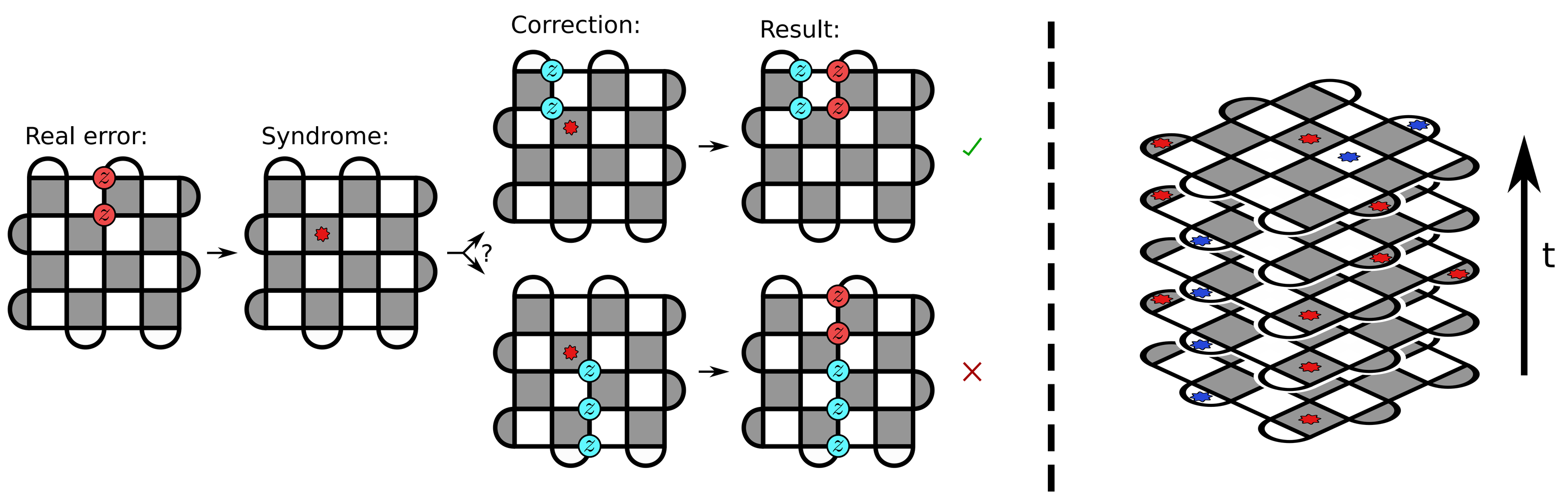
Qubits are arranged on the vertices of a square lattice.
Logical operators span the whole code, from left to right (X_L) or from top to bottom (Z_L).
Two types of local stabilizer operators: X plaquettes (grey) and Z plaquettes (white):



Measurement of all stabilizers generates a syndrome, in which anyons appear on the plaquettes whose stabilizer anticommutes with the error configuration:

The Decoding Problem

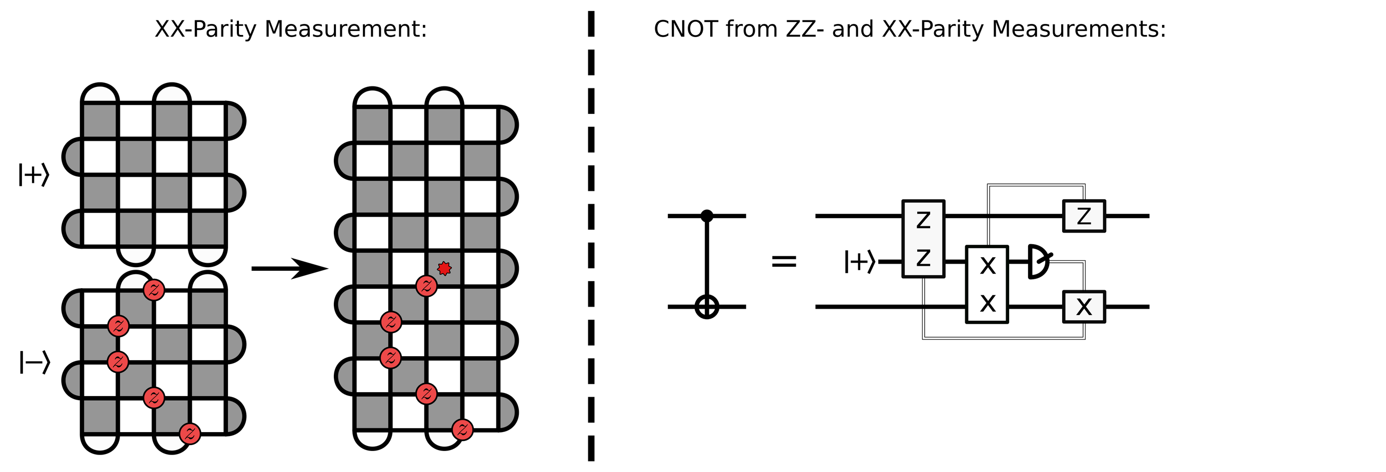
Any decoding algorithm has access only to the syndrome measurements (anyon configuration), and *not* the underlying error configuration.
From a given syndrome the decoder should suggest a sequence of operations which brings the code back into the ground state space (i.e no anyons), but without also performing a logical operation! Unfortunately, many error configurations result in the same syndrome!



In the left example above, the upper correction correctly puts the code into the ground state space without introducing a logical operation, while the lower correction results in a logical operation.
In a more realistic setting stabilizer measurements are also noisy, and even the syndrome information can be imperfect! In this setting, shown above right, many rounds of syndrome measurement are conducted, and the decoding algorithm has access to a sequence of syndrome measurements.

Lattice Surgery for Controlled Operations

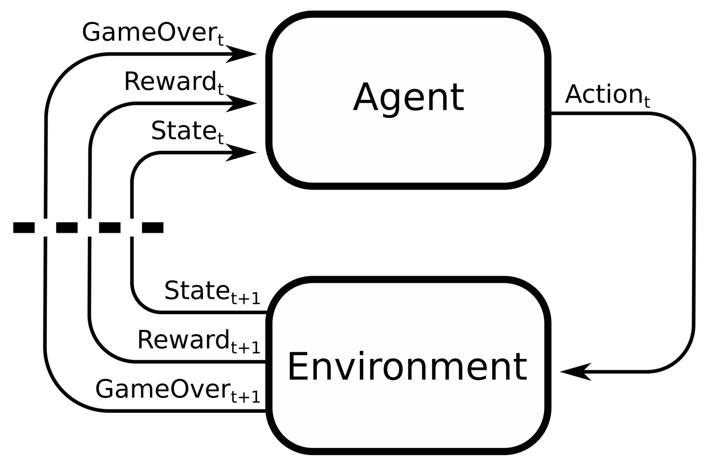
Lattice surgery can be used to implement two qubit controlled operations (green blocks) in a fault tolerant fashion. In the process the two surface codes are joined, and the parity of the encoded qubits can be obtained through measurement of new stabilizers along the joint.



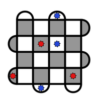
Above left, two logical qubits are in states $|+\rangle$ and $|-\rangle$ respectively, and the odd XX-Parity can be deduced from the odd number of violated stabilizers on the joint. On the right, a circuit is shown which performs a CNOT gate, using only single qubit and two-qubit parity measurements. Of course, in the presence of noise, and after qubit idling, decoding is necessary to ensure (a) the correct parity measurement and (b) to return the logical qubits in the ground state of the respective codes.

Decoding as a Game

We view decoding as a game in which an Agent (the decoder) plays against a noisy environment, with the code lattice as the board.

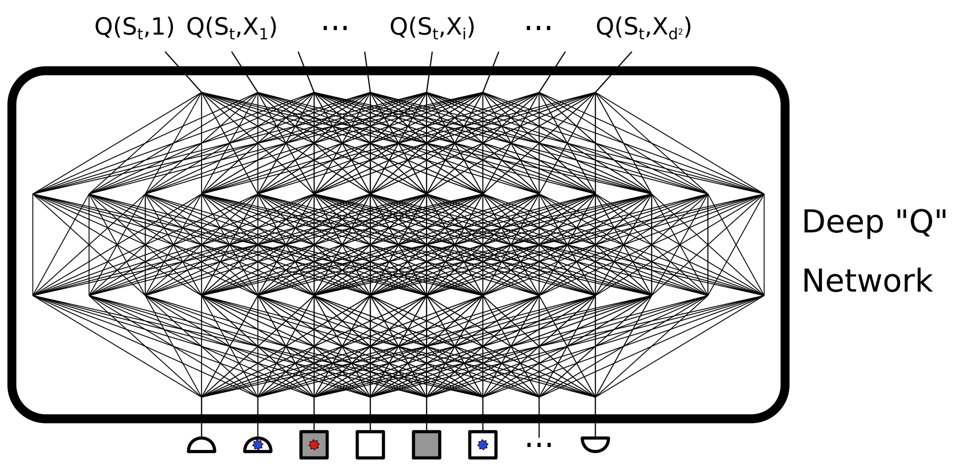


At each step the agent can perform a single gate and the environment updates the visible syndrome, while periodically introducing errors according to its internal noise model. More specifically:

- State$_t$ = syndrome: 
- Action$_t$ = $\{1, X_1, \ldots, X_{d^2}\}$
- Reward$_t$ = $\begin{cases} 1, & \text{if no anyons} \wedge \text{no logical errors} \\ 0, & \text{otherwise.} \end{cases}$
- GameOver$_t$ = $\begin{cases} \texttt{True}, & \text{if State}_t \text{ can be decoded} \\ \texttt{False}, & \text{otherwise.} \end{cases}$

The Agent

We obtain such an agent through "Q-learning"- a reinforcement learning framework in which through repeated interactions with the environment (game play), the agent tries to learn a good Q function, parameterized by a deep neural network.



State$_t$ = unrolled Syndrome

Specifically, the Q function gives a measure of how good a specific action is, from a given state, in terms of expected future reward. Precisely:

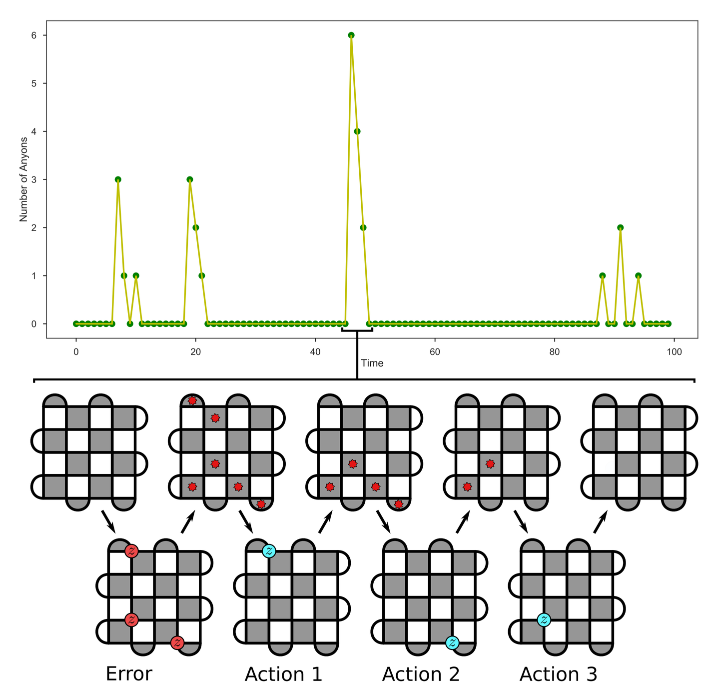
$$Q(S_t, A_t) = \mathbb{E}\left(\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}\right)$$

Once learned, the optimal action from any state can be obtained by taking the argmax over the Q function:

$$A_t = \text{argmax}_{A_j}\left(Q\left(S_t, A_j\right)\right)$$

Game Time!

We find that our Agent is capable of learning a Q function representing an effective decoding strategy, allowing it to continuously compete with the environment, and keep the code in the ground state space, without introducing logical errors.



So, can we train an agent to compete against environments providing faulty syndromes?

If yes, such an agent would provide an effective decoder for the fault tolerant setting, without requiring final data qubit measurements, and hence a technique for performing arbitrary length qubit idling (yellow blocks) in the fault tolerant setting. Combined with lattice surgery (for green blocks) this provides a promising toolbox for FTQC.