↓ **Lecture 17** [26.06.25]

7 | Quantum error correction (QEC) protocol:

QEC protocols are classically controlled algorithms (no Hamiltonian dynamics!) with the goal to systematically remove errors from quantum systems to protect quantum information. Their job is to "pump entropy" out of the system.

⊲ Encoded qubit:

$$|\Psi_0\rangle = \alpha|0\rangle + \beta|1\rangle \quad \in \mathcal{C} \tag{5.57}$$

$\alpha, \beta \in \mathbb{C}$: Logical amplitudes (this is the information we want to protect!)

i | Assume that since initialization in $|\Psi_0\rangle$, a few elementary errors occurred on random positions of the chain:

$$|\Psi_0\rangle \xrightarrow[\text{on sites with } x_j = 1]{\text{Unknown errors}} |\tilde{\Psi}_0\rangle = \underbrace{\prod_j (E_j)^{x_j}}_{\equiv E(\boldsymbol{x})} |\Psi_0\rangle \quad \notin \mathcal{C} \tag{5.58}$$

$\boldsymbol{x} = (x_1, \dots, x_L) \in \{0,1\}^L$: unknown error pattern

Our goal is to figure out if and where errors occurred so that we can remove them before they have the chance to accumulate and destroy the encoded qubit [like in Eq. (5.56)].

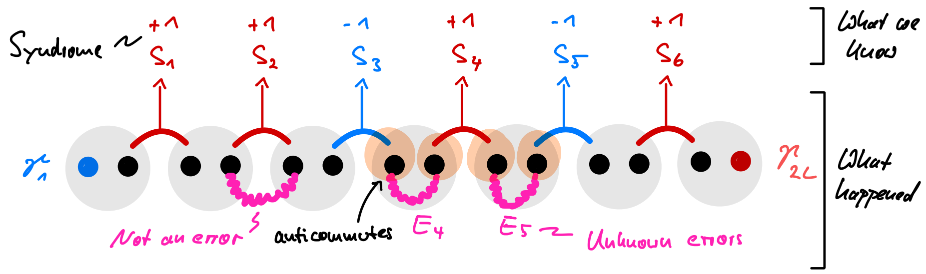
Due to the errors, the state above is no longer in the code space: $|\tilde{\Psi}_0\rangle \notin \mathcal{C}$. [In condensed matter parlance, it is no longer a ground state of Eq. (5.50) but an excited state.] Note, however, that the amplitudes α and β are still hidden in $|\tilde{\Psi}_0\rangle$! The problem is that they were "shuffled around" in an unknown way because of the error operations …

ii | Observation: $S_k E_j = -E_j S_k$ if $k = j-1$ or $k = j$

This follows from Eq. (5.34) and the fact that adjacent errors and syndromes share a *single* Majorana fermion.

→ Measuring S_k yields information (negative eigenvalues) about the locations of errors!

→ Measure all stabilizers S_j:



→ ⁂ *Error syndrome* $\boldsymbol{s} = (s_1, \dots, s_{L-1}) \in \{\pm 1\}^{L-1}$

¡! Since $S_j^2 = \mathbb{1}$ this yields one bit $s_j = \pm 1$ of information per stabilizer generator. It is crucial that [due to Eq. (5.55)] these measurements cannot destroy the encoded qubit [i.e., they cannot reveal the amplitudes α and β in Eq. (5.57)].

Question: Can we use $\boldsymbol{s}$ to compute $\boldsymbol{x}$?

If we *knew* the error pattern $\boldsymbol{x}$, we could simply undo the (unitary) error operators E_j and recover the state $|\Psi_0\rangle$. The hitch is that we *don't know* $\boldsymbol{x}$ (that's what errors are, after all!).

iii | Decoding algorithm:

A decoding algorithm is a *classical algorithm* that tries to guess the actual error pattern based on the syntrome information:

$$\text{Syndrome } \boldsymbol{s} \xrightarrow{\text{Decoding}} \text{Predicted error pattern } \boldsymbol{x}^? = \mathcal{D}(\boldsymbol{s}) \tag{5.59}$$

$\mathcal{D}$: (Classical) decoding algorithm (to be constructed)

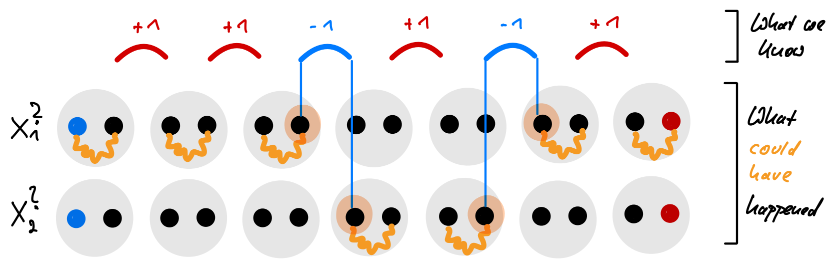
- ¡! Note that $\mathcal{D} : \{\pm 1\}^{L-1} \to \{0,1\}^L$ cannot be surjective because there are only $L-1$ bits of syndrome data but L bits in an error pattern. This immediately tells us that there must be error patterns that the decoder cannot predict because they give rise to the same syndrome data. Hence there must be situations (= error patterns) where the decoder *fails* and the encoded quantum information is lost. This is not bad luck but intrinsic to any quantum error correction protocol. (Because physical errors can always conspire to act as a logical operation that one cannot detect without destroying the encoded qubit.)
- This can be seen from Eq. (5.56) were we showed that the error pattern $\boldsymbol{x} = (1, \dots, 1)$ leads to a logical Σ^z operation. But $[\Sigma^z, S_j] = 0$ for all $j = 1, \dots, L-1$ so that syndrome measurements cannot detect this type of error: $\boldsymbol{s}(\Sigma^z) = (+1, \dots, +1)$. [Had no error occurred, the syndrome would be the same: $\boldsymbol{s}(\mathbb{1}) = (+1, \dots, +1)$.] A decoder has to make a decision whether to decode $\boldsymbol{s} = (+1, \dots, +1)$ to $\boldsymbol{x}^? = (0, \dots, 0)$ or $\boldsymbol{x}^? = (1, \dots, 1)$ (if errors are rare, the first choice is the better one!). In any case, there will be situations where the decoder chooses wrong and fails to predict the actual error string: $\boldsymbol{x}^? \neq \boldsymbol{x}$.

So which decoding algorithm should be pick for our "Majorana chain quantum code"?

For a given quantum code, there are many possible decoding algorithms. Which one to pick depends on many factors: the probability distribution of errors, how efficient the algorithm runs (on classical hardware), and, most importantly, its ↑ *threshold* (the microscopic error rate that must be reached for the QEC scheme to become useful).

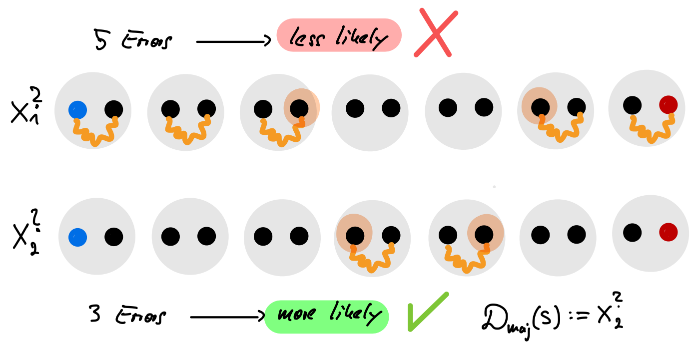
Suggestion: "Majority voting" $\mathcal{D}_{\text{maj}}$:

Step **1**: Construct the (only) *two* error patterns $\boldsymbol{x}_1^?$ and $\boldsymbol{x}_2^?$ consistent with the given syndrome $\boldsymbol{s}$:



¡! Note that syndromes adjacent to *two* errors yield $+1$ when measured as they anticommute *twice* and the minus cancels. This means that syndrome measurements detect *boundaries of error chains*.

Step **2**: Return the pattern with *fewer* errors:



- The rationale of this coice is that for low error probabilities (actually: $p < 0.5$) the error pattern with the smaller number of errors is the more likely one.
- $\mathcal{D}_{\text{maj}}$ is provably optimal for the Majorana chain quantum code if syndrome measurements are perfect (it is the so called ↑ *maximum-likelihood decoder*).

iv | Apply corrective operations:

As a last step, we apply the inverse error operations on the locations predicted by $\boldsymbol{x}^?$ (since here errors are Hermitian, it is $E_j^{-1} = E_j^\dagger = E_j$):

$$|\tilde{\Psi}_0\rangle \xrightarrow[\text{= Undo errors}]{\text{Apply corrections}} E^{-1}(\boldsymbol{x}^?)|\tilde{\Psi}_0\rangle \tag{5.60a}$$

$$= E^{-1}(\boldsymbol{x}^?)E(\boldsymbol{x})|\Psi_0\rangle \tag{5.60b}$$

$$= E(\boldsymbol{x}^? \oplus \boldsymbol{x})|\Psi_0\rangle \tag{5.60c}$$

$$\begin{cases} \overset{\boldsymbol{x}^?=\boldsymbol{x}}{=} & |\Psi_0\rangle \quad \to \text{Success} \checkmark ☺ \\ \underset{5.56}{\overset{\boldsymbol{x}^?=\bar{\boldsymbol{x}}}{=}} & -\Sigma^z|\Psi_0\rangle \quad \to \text{Failure} ✗ ☹ \end{cases} \tag{5.60d}$$

Here $\bar{\boldsymbol{x}}$ denotes the comlementary pattern obtained from $\boldsymbol{x}$ by exchanging $0 \leftrightarrow 1$, and $\oplus$ denotes bit-wise modulo-2 addition, i.e., $\bar{\boldsymbol{x}} \oplus \boldsymbol{x} = (1, 1, \dots, 1)$.

- ¡! Note that the quantum memory controller does *not know* whether the correction was successful or not. Otherwise, it could have applied the "correct correction" in the first place. A failed QEC cycle therefore leads to a "silent" logical operation on the encoded qubit which (most likely) screws up the quantum algorithm that follows.
- If the decoding is successful, the QEC protocol gains at no point knowledge about the encoded amplitudes α and β. That this is possible in principle was the seminal insight by Peter Shor in 1995 [148]. It is one of the foundations upon which the promise of scalable quantum computing rests (and, by know, several billions of market cap).
- In reality, it is often more convenient to compute and accumulate all correction strings and adapt only the final read-out stage of the quantum circuit (or apply the corrections in classical post-processing). If possible, this is advantageous because applying corrective unitaries takes time and can introduce new errors.

Together, these four steps make up a ⁂ *quantum error correction cycle* and are repeated periodically (in the range of microseconds to milliseconds) to ensure that the number of errors accumulated between consecutive correction cycles remains small (which is necessary for the decoder to function properly).

This means in particular that the decoding algorithm must be *computationally efficient*. This becomes a formidable task for larger and more complicated quantum error correction codes (like the → *toric code*), and explains why quite a lot of effort is put into making decoders both better at guessing $\boldsymbol{x}$ and doing so *quickly* and without much classical overhead [149].

8 | This procedure hints at a much more general recipe to construct quantum error correction codes:

⊲ Topologically ordered phase of matter:

$$\boxed{\begin{array}{rcl} \text{Robustly degenerate ground state space} & \to & \text{Code space} \\ \text{Excitations of Hamiltonian} & \to & \text{Errors} \\ \text{Local Hermitian terms in Hamiltonian} & \to & \text{Syndrome observables} \end{array}}$$

→ ⁂ *Topological Quantum Error Correction (TQEC)*

We will study a two-dimensional topological quantum code that does not rely on fermion parity as a symmetry in ?? (→ *toric code*).

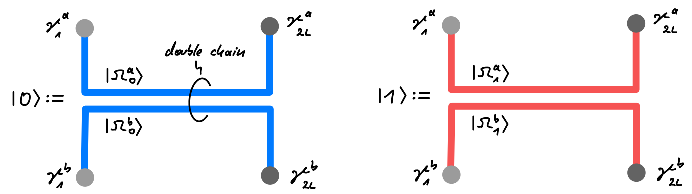
The concept of TQEC belongs to the fascinating intersection of condensed matter physics, topology, and quantum information theory teased in the Venn diagram of Section 0.2.

9 | Comments:

- Pairs of chains:

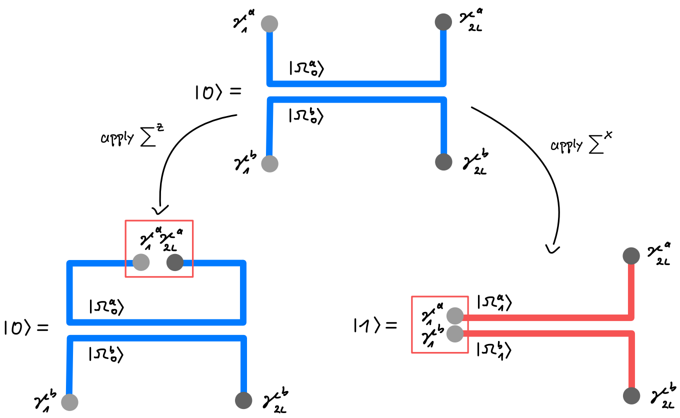
  Recall that we assumed the logical operators Σ^x and Σ^y to be forbidded as physical errors due to fermionic parity symmetry (which might or might not be a good assumption in a particular setting). But if parity-violating unitaries cannot be physically realized, then *we* as operators should also not be capable of these operations! We can't have our cake and eat it too!

  So in an "honest" setting with fermion parity symmetry, we should implement *all* logical operators as parity-symmetric operators as well. The trick is to encode a single logical qubit not in one but in a *pair of chains*, e.g., by associating the negative parity of both to $|0\rangle$ and postive parity of both to $|1\rangle$:

  Importantly, the *total* fermion parity of the system is now fixed (even, in this case) and does not have to be changed when flipping the qubit state. To make sure that no logical operators are affected by errors, we must now completely rely on the *locality* argument. This means in particular, that "storage mode" is achieved by moving the endpoints of all chains far apart from each other to prevent local (parity-preserving) operators from coupling them (as shown above).

  Logical operators can then be applied by moving two enpoints close together and applying (or measuring) a parity-symmetric (!) product of two Majorana modes:

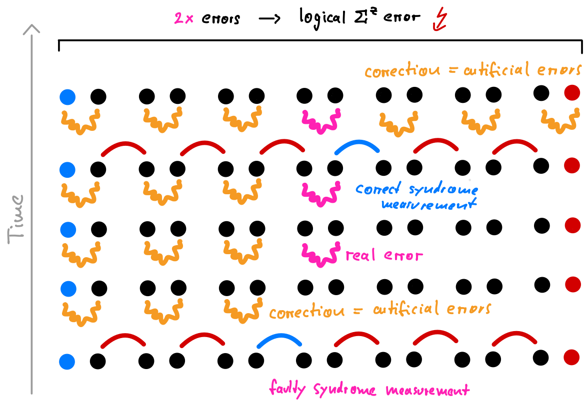


Crucially, the application of $\Sigma^x \propto \gamma_1^a \gamma_1^b$ (which essentially tunnels fermions from one chain to the other) does *not* violate fermion parity but switches the "subsystem parities" of the two chains (thereby flipping the logical qubit). (Note that applying a local pair of Majoranas between the two chains *away* from the endpoints creates quasiparticle excitations in both chains – which can be detected by syndrome measurements.) Encoding qubits in the sub-parities of multiple chains is the basic principle of Majorana-based quantum computing architectures [147, 150, 151].

- Imperfect stabilizer measurements:

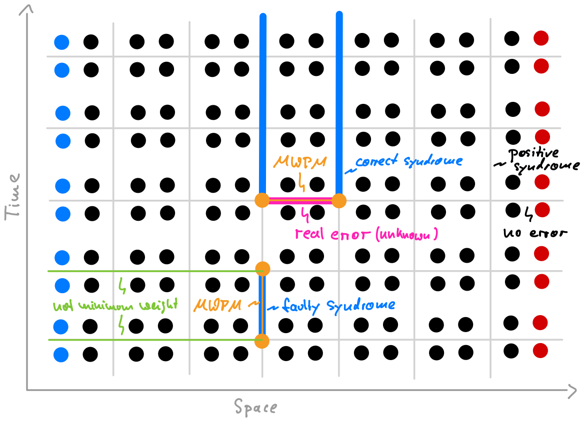
  In a realistic setup, there is noise *everywhere*, in particular, the syndrome measurements *themselves* are not always correct: Sometimes a measurement might return $s_j = -1$, although no error occurred on the chain. Conversely, a measurement might miss an actual error and return $s_j = +1$. Projective measurements in a quantum experiment *always* come with a rate that quantifies how noisy they are (and this rate is non-zero)!

  As we will see now, this is not just a minor inconvenience that can be "abstracted away." That one must take these additional errors into account can be seen from the following process that is triggered by only *two* errors (one on the chain and one affecting the syndrome). Most importantly, the process (and therefore its probability) is *independent* of the length of the chain:



  The crucial point is that a *single* faulty syndrome measurement can trick our decoder into applying an extensive number of artificial errors ("corrections"). A single true error is then enough to tip the scales and destroy the encoded information.

  Every realistic quantum error correction protocol must be designed to withstand noise not only on the quantum code itself but also on the syndrome measurement routine. To achieve this, one can employ decoding algorithms that operate on "spacetime" by taking into account not only the current pattern of syndromes but also their history:



For the Majorana chain code, it is convenient to draw a square lattice in spacetime where syndrome measurements are associated to vertical edges and $s_j = -1$ outcomes are highlighted. Next, one identifies the *endpoints* of these highlighted paths (orange above). To reconstruct the (unknown) error pattern (now including both errors on the quantum code and faulty stabilizer measurements), the decoder performs a procedure called ↑ *minimum-weight perfect matching* (MWPM) [152, 153]. The idea is to connect all endpoints pairwise (via paths that can include horizontal and vertical edges and can terminate on the boundaries) such that no endpoints remain unpaired (this is the "perfect matching"). Every path is assigned a "weight" computed from the probabilities of errors and faulty stabilizer measurements, such that smaller weights correspond to more likely paths (for low error probabilities, these are typically the shortest paths). The perfect matching with the smallest total weight ("minimum-weight") is then selected and used to guess the errors that occurred: every *horizontal* line traversed by the paths that connect the endpoints corresponds to an *error* that occurred, and every *vertical* line to a *faulty stabilizer measurement*.

A similar approach can be used to decode two-dimensional topological codes like the → *toric code* [154–156] (the spacetime pattern is then three-dimensional). Note that the MWPM decoder sketched above is no longer the maximum-likelihood decoder [157, 158], i.e., it does not necessarily construct the corrective operations that are most likely correct (this is not obvious).

- The Majorana chain can be mapped via a → *Jordan-Wigner transformation* from fermions to spins (➔ Problemset 8). This elucidates its relation to the ← *transverse-field Ising model* discussed in Section 0.3. On the level of quantum codes [and in the language of stabilizers, Eq. (5.51)] this mapping yields a degenerate version of the → *toric code* (on a lattice of size $L \times 1$) [149], which is *not* a quantum- but a classical memory (a ↑ *repetition code*). The Majorana chain code is therefore a "topological quantum memory with caveats:" One type of error (phase errors) are kept in check due to the locality structure of the code – this is the hallmark of *topological* quantum codes. By contrast, the other type of errors (bit-flip errors) cannot be corrected. In the "spin-world" of the toric code, this cannot be argued away and one is left with a classical error correction code. In the "fermion world" of the Majorana chain, one can – on purely physical grounds – argue that such errors violate fermion parity symmetry and are therefore suppressed. True topological quantum codes (like the toric code) do not rely on such symmetry-protection arguments. The price to pay is that such codes are at least two-dimensional (because there is no topological order in one-dimensional spin systems [6]).
- Starting from Majorana fermions, one can construct more general quantum error correction codes called ↑ *Majorana fermion codes* [159].
- The concepts we explored suggest an intriguing possibility: The Majorana chain Hamiltonian

Eq. (5.50) is a quantum phase with a ground state manifold that has the properties of a quantum error correction code! The local terms in the Hamiltonian correspond to syndrome operators, and a non-trivial syndrome (indicating the presence of an error) corresponds to an *excitation* of this Hamiltonian. This motivates the following question:

Can we suppress *errors energetically (by lowering the temperature)*
instead of applying active error corrections (by a classical decoding algorithm)?

A Hamiltonian with these properties is called a ↑ *self-correcting quantum memory*. Unfortunately, the Majorana chain is *not* self-correcting for the same reason the one-dimensional classical Ising model has no phase transition at finite temperatures: While the initial *creation* of an error indeed costs energy, its subsequent *movement* is not energetically penalized (in the classical Ising model, the creation of a domain wall costs energy, but its proliferation through the chain does not). This mechanism prevents a thermodynamically stable phase at finite temperature in which the quantum information encoded in the thermal Gibbs state $\rho = e^{-\beta H}/Z$ (density matrix!) would survive exponentially long in the system size.

The quest for finding a truly self-correcting system in three spatial dimensions or less is an active area of research. For example, it is known that a wide class of systems based on stabilizer codes (under some additional constraints) *cannot* be self-correcting [160, 161] (due to the presence of point-like excitations). There are interesting proposals with partially self-correcting properties [162, 163]; however, to the best of my knowledge, all of them have some drawbacks and do not qualify as true self-correcting systems.

Fun fact: In *four* spatial dimensions, a self-correcting quantum memory is known to exist, namely the 4D generalization of the toric code [164]. The problem is that our world is not four dimensional ☹.

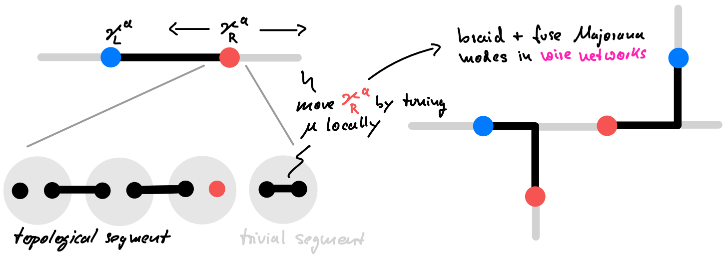
- Braiding in wire networks:

  Majorana modes located at extrinsic defects can exhibit → *non-abelian anyonic statistics* (so called ↑ *(projective) Ising anyons* [165, 166]). Note that these are not quasiparticle excitations but high-energy deformations of the Hamiltonian! As we have seen in this chapter, Majorana modes naturally occur on the endpoints of p-wave superconducting wires (they can also appear in the vortices of two-dimensional $p_x + i p_y$ superconductors [167]).

  Measuring the parity of a fermion mode given by two Majorana modes (recall $\Sigma^z \propto \gamma_{2L}\gamma_1$) can then be interpreted as the ↑ *fusion* of two "Ising anyons." The non-abelian nature of these anyons is reflected in the fact that there are two consistent outcomes of this measurement: the fermion mode can be empty or occupied. Formally, one writes $\sigma \times \sigma = 1 + \Psi$ where σ denotes an Ising anyon (realized by a Majorana mode), 1 corresponds to an empty fermion mode and Ψ to an occupied fermion mode.

  It turns out that moving Ising anyons adiabatically (= slowly) around each other effects non-trivial unitary operations on the degenerate subspace that encodes the different fusion outcomes. This process is called ↑ *braiding* and can be used to manipulate the encoded qubits (like the ones in our Majorana chain quantum code) *without decoding them*. This is the rationale of → *topological quantum computation*, an intrinsically robust quantum computing architecture.

  But how can one braid the Majorana modes at the endpoints of Majorana chains around each other? The idea is to use *wire networks* with locally tunable chemical potentials (by applying local gate voltages) [150]. By tuning the chemical potential, one can make segments of the wire network topological, while other parts remain in the trivial phase [recall Eq. (5.22)]. The boundaries between topological and trivial segments then host Majorana modes that can be shuffled around by changing the local gate voltages. If one adds T-junctions to connect these wires, one can start braiding Majorana modes around each other:



This is the basic idea behind a Majorana-based topological quantum computer. However, there are two caveats to be aware of: First, the braiding rules of Ising anyons cannot realize a universal gate set [168] so that one needs additional (non-topological) gates to construct a full-fledged quantum computer. And second, from an engineering perspective, it is simpler to replace the dynamical *braiding* by meticulously designed sequences of *projective measurements* [147, 169]; this architecture is known as ↑ *measurement-based topological quantum computing* and is actively pursued by Microsoft [151].

5.7. ‡ Experiments

- The first evidence for Majorana zero modes at the boundaries of quantum wires was reported in 2012 by Mourik *et al.* [141]. They fabricated a semiconducting nanowire with strong spin-orbit coupling that opens a band gap when a magnetic field is applied (to enter a "spinless" regime). This nanowire is then coupled to a normal s-wave superconductor which induces effective p-wave pairing in the nanowire [170, 171]. The emergence of zero-energy Majorana modes can than be probed by ↑ *tunnel spectroscopy*. These results were later substantiated by many follow-up studies (e.g. [172, 173]), see also Ref. [174] for a review.

  Characterizing the topological nature of Majorana zero modes is notorioly difficult because their signatures are ofthen hard to distinguish from non-topological phenomena. This has lead to several controversial reports, including complete retractions of papers [142].

- As discussed in Section 5.6, Majorana chains can in principle be used as quantum memories. Beyond that, "braiding" Majorana zero modes (either by adiabatically moving them around each other or projectively measuring them) can be used to affect unitary gates on the encoded qubits (these unitaries are not universal, though). This led to proposals for Majorana-chain based quantum computing architectures [147, 150] which are actively pursued by Microsoft's quantum computing division. In 2025, first experimental results of parity measurements of Majorana qubits were reported [151] – and immediately criticized as unreliable [175].

  The future will tell whether Majorana modes are a feasible approach to build a quantum computer …