
Quantum error correction with the surface code

The authors are grateful to Earl Campbell, Andrew Cross, and John Preskill for reviewing this chapter.

Rough overview (in words)

To protect quantum information from the detrimental effects of noise, we can encode it into a code space of some quantum error correcting (QEC) code [935, 952]. Oftentimes, we choose to work with stabilizer codes [444]. By definition, a code space of a stabilizer code is the simultaneous (+1)-eigenspace of a set of commuting Pauli operators, commonly referred to as parity checks.

The surface code [626, 196, 343] is one of the most studied stabilizer codes. It can be implemented with a planar layout of qubits and entangling gates only between neighboring qubits. For that reason, the surface code is particularly appealing for quantum hardware architectures with restricted qubit layout and connectivity, such as superconducting circuits [348, 155]. The most common realization of the surface code uses $n = L^2$ data qubits to encode $k = 1$ logical qubit and has code distance $d = L$, where L is the linear size of the $L \times L$ square lattice with open boundary conditions. Additionally, $n_A = L^2 - 1$ ancilla qubits are typically used to measure parity checks; see Fig. 26.1(a).

In order to perform QEC, we have to be able to detect errors without revealing the encoded information. For stabilizer codes, we can achieve that by measuring their parity checks to obtain the error syndrome (which comprises the measurement outcomes returning -1). Then, the error syndrome is processed by specialized classical algorithms, also known as “decoders,” to find an appropriate recovery operator that attempts to remove errors afflicting the encoded information. For generic stabilizer codes, the problem of optimal decoding is computationally hard, even for simple noise models [563]. However,

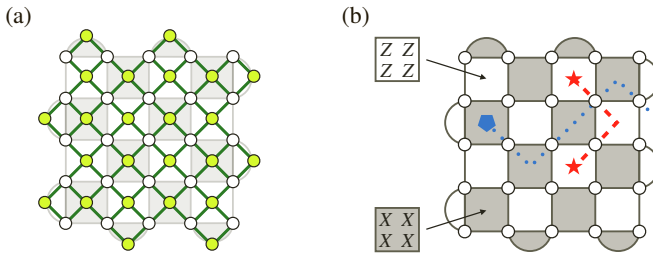


Figure 26.1 (a) A planar layout of data and ancilla qubits (empty and filled dots, respectively) with entangling gates (thick edges) only between neighboring qubits. This layout gives rise to the $L \times L$ square lattice with open boundary conditions, where $L = 5$ here. (b) The surface code can be realized by measuring Pauli Z- and X-type parity checks (light and dark faces, respectively). The error syndrome (stars and pentagons) can be interpreted as the endpoints of string-like Pauli X and Z errors (dashed and dotted lines, respectively).

for QEC codes with some underlying structure, such as the surface code, there exist a variety of computationally efficient (albeit not optimal) decoding algorithms. In particular, the three most popular classes of decoders for the surface code are as follows:

- Matching decoders, including the minimum-weight perfect matching algorithm [343] and its follow-up improvements, such as the belief-matching algorithm [320, 530]. These decoders phrase the problem of surface code decoding as a graph-theoretic problem of perfect matching, which can be efficiently solved [372].
- Clustering decoders, such as the renormalization-group decoder [364, 42] and the union-find decoder [338]. These decoders primarily exploit the structure of the error syndrome in the surface code; see Fig. 26.1(b).
- Tensor network decoders [192, 331, 291]. These decoders phrase the problem of surface code decoding as a numerical problem of contracting tensor networks.

In order to assess the usefulness of decoders, one usually considers two criteria: runtime and accuracy. The first criterion, runtime, is defined as the time needed for the decoder to process the error syndrome. It is crucial that any practical decoder is able to operate at a rate compatible with the rate of parity check measurements; otherwise, the error syndrome will start to accumulate, leading to the backlog problem [985]. The second criterion, accuracy, is typically defined for a given noise model in terms of the logical error rate, that is, the failure rate of the decoder to successfully undo the effects of noise on

the encoded information. From the perspective of reducing runtime and improving accuracy, matching and clustering decoders stand out. Namely, they can achieve almost-linear runtime [529, 338] and good accuracy. Combined with the techniques of parallelizable real-time decoding [942, 170], these decoders will likely play a key role in scalable QEC with the surface code. To achieve optimal accuracy, one can use tensor network decoders, however, they are often not computationally efficient, with runtime that scales unfavorably.

Rough overview (in math)

In addition to being compatible with planar layouts of qubits and admitting computationally efficient decoders with good accuracy, the surface code also exhibits one of the highest QEC thresholds. Recall that a QEC threshold is specified for the following triple: a QEC code family of growing distance d , a decoder, and a noise model. It is defined as the highest value p_{th} such that for any error rate $p < p_{\text{th}}$ the probability that the decoder fails to undo the effects of noise goes to zero as d goes to infinity. For example, the QEC threshold for the surface code, using the minimum-weight perfect matching algorithm for decoding, with a circuit noise model based on depolarizing noise, is around 1% [1017, 530].

Typically, if the error rate p describing noise is sufficiently low and below the threshold p_{th} , then the logical error rate p_{fail} roughly scales as

$$p_{\text{fail}} \sim \left(\frac{p}{p_{\text{th}}} \right)^{\lceil \frac{d}{2} \rceil}. \quad (26.1)$$

This implies that in order to achieve the target error rate ϵ , it suffices to implement the surface code with code distance $d = O(\log(1/\epsilon)/\log(p_{\text{th}}/p))$ using $n + n_A = O(d^2) = O(\log^2(1/\epsilon)/\log^2(p_{\text{th}}/p))$ data and ancilla qubits. Subsequently, qubit overhead associated with QEC based on the surface code scales polylogarithmically in the inverse target error rate $1/\epsilon$.

Dominant resource cost (gates/qubits)

Performing reliable QEC in the presence of measurement errors becomes challenging since the error syndrome can be corrupted. A straightforward solution to the problem of unreliable error syndrome is to repeatedly measure the parity checks in order to gain enough confidence in their measurement outcomes [936, 343]. If this approach is applied to the surface code with code distance d , then one needs to perform $O(d)$ rounds of parity check measurements, incurring relatively large time overhead.

To reduce time overhead, one can pursue single-shot QEC [166], which does not require repeated measurement rounds. It is possible to realize single-shot

QEC with the surface code [224, 65, 339], however, in addition to the parity checks in Fig. 26.1(b), one would need to measure nonlocal high-weight parity checks, which is a serious limitation. A more streamlined approach is to consider a different realization of the surface code, the 3D subsystem toric code [652, 202], which can be implemented with qubits arranged on the cubic lattice and local low-weight parity checks. Although this approach is natively defined in three spatial dimensions, it can be emulated with planar layouts of qubits and either a limited number of nonlocal gates or the ability to reshuffle qubits (which is available with, e.g., Rydberg atoms [890, 203]). In order to realize code distance d , one incurs qubit overhead of $O(d^3)$ (compared to qubit overhead of $O(d^2)$ for the surface code). From that perspective, single-shot QEC with the subsystem toric code can be viewed as trading time overhead for qubit overhead.

Caveats

There have been efforts to improve surface code decoders by incorporating various machine learning methods, including neural networks [992, 746, 251, 1003, 107] and reinforcement learning [973]. At the current stage, decoders solely based on machine learning methods seem to be of limited applicability, mostly due to high training costs and scalability issues. Nevertheless, these approaches are likely to be immensely beneficial for QEC in the settings where (possibly correlated) noise is unknown and may have to be learned first.

Typically, in QEC analysis one considers simple Pauli noise, such as depolarizing noise acting independently and identically on each qubit. If noise exhibits bias between the X , Y , and Z components of Pauli noise, then this structure can be exploited, leading to dramatically increased QEC thresholds, as exemplified by variants of the surface code [999, 173, 363]. Similarly, noise that is biased toward erasure errors can be beneficial from the perspective of QEC [947, 1057, 654, 467]. On the other hand, realistic noise may be coherent or correlated and thus not only difficult to correct, but also to numerically simulate. For instance, the logical error rates for coherent noise may be orders of magnitude higher than the estimates of the logical error rates for simple Pauli noise (assuming both types of noise have the same error rate) [564]. In certain cases of coherent noise, however, QEC with the surface code may be efficiently simulable [193].

In addition to the 3D subsystem toric code, one can also consider other higher-dimensional versions of the surface code. With these codes, roughly speaking, one improves the QEC capabilities at the expense of increased qubit overhead. Moreover, for the higher-dimensional surface code, it may suffice to use arguably the least complex decoders that are based on cellular automata

(which, by definition, are parallelizable and only use local information about the error syndrome) [343, 200, 651, 1005].

Example use cases

- Decoders for the surface code can be used for other QEC code families, such as the color code [169, 168, 647]. In fact, due to a close connection between the color codes and the surface codes [171, 653], any surface code decoder can be used as a subroutine in the restriction decoder for any color code (in two or more spatial dimensions) [649, 1004].

Further reading

- The seminal paper by Dennis et al. [343] is a thorough introduction to QEC with the surface code.
- A recent perspective [205] discusses how to use matching decoders to decode stabilizer codes.
- Open-source software packages have been developed for implementing QEC with the surface code, such as Stim [422] and PyMatching [528].