Topological Concatenation of 2D Color Codes

Alexandre Guernut    Christophe Vuillot

June 19, 2021
1. 2D triangular color codes

2. Topological concatenation of color codes

3. Characteristics of concatenated codes

4. Decoding the concatenated color codes

5. Further prospects
2D triangular color codes
A topological stabilizer code

- Stabilizers are associated to the faces of a tiling of a sphere

(a) Sphere tiling
A topological stabilizer code

- Stabilizers are associated to the faces of a tiling of a sphere
- Qubits are at the vertices of the tiling

(a) Sphere tiling
A topological stabilizer code

- Stabilizers are associated to the faces of a tiling of a sphere
- Qubits are at the vertices of the tiling
- Each face is associated to an X and a Z stabilizer acting on the vertices
A topological stabilizer code

- Stabilizers are associated to the faces of a tiling of a sphere
- Qubits are at the vertices of the tiling
- Each face is associated to an X and a Z stabilizer acting on the vertices
- One vertex and three faces are removed to obtain one logical qubit

(a) Sphere tiling
A topological stabilizer code

- Stabilizers are associated to the faces of a tiling of a sphere
- Qubits are at the vertices of the tiling
- Each face is associated to an X and a Z stabilizer acting on the vertices
- One vertex and three faces are removed to obtain one logical qubit

(a) Sphere tiling
(b) 2D Triangular color code
The dual graph is often more useful
Dual representation

- The dual graph is often more useful
- Stabilizers are now vertices that act on triangular faces (the qubits)
Dual representation

- The dual graph is often more useful
- Stabilizers are now vertices that act on triangular faces (the qubits)
- Colored boundaries are now boundary nodes
Study cases

- Any 3-colorable, 3-valent tiling of a sphere could be used
Study cases

- Any 3-colorable, 3-valent tiling of a sphere could be used
- We limit our interest to two particular regular tilings
Study cases

- Any 3-colorable, 3-valent tiling of a sphere could be used
- We limit our interest to two particular regular tilings

(a) $6 - 6 - 6 : [n = \frac{3}{4}d^2 + \frac{1}{4}, 1, d]$
Study cases

- Any 3-colorable, 3-valent tiling of a sphere could be used
- We limit our interest to two particular regular tilings

(a) $6 - 6 - 6 : \left[ n = \frac{3}{4} d^2 + \frac{1}{4}, 1, d \right]$

(b) $4 - 8 - 8 : \left[ n = \frac{1}{2} d^2 + d - \frac{1}{2}, 1, d \right]$
Study cases

- Any 3-colorable, 3-valent tiling of a sphere could be used
- We limit our interest to two particular regular tilings
- They are the only ones whose leading coefficient in the number of qubits representation as a function of the distance is less than 1

(a) \(6 - 6 - 6: \left[ n = \frac{3}{4}d^2 + \frac{1}{4}, 1, d \right] \)

(b) \(4 - 8 - 8: \left[ n = \frac{1}{2}d^2 + d - \frac{1}{2}, 1, d \right] \)
A colored string is a sequence of edges of the same color.
A colored string is a sequence of edges of the same color

Three strings crossing at the same point form a non-trivial logical operator if they start on the boundaries
A colored string is a sequence of edges of the same color

Three strings crossing at the same point form a non-trivial logical operator if they start on the boundaries
Y operators

- A colored string is a sequence of edges of the same color
- Three strings crossing at the same point form a non-trivial logical operator if they start on the boundaries
Topological concatenation of color codes
Once one has both a quantum code and a decoding procedure whose threshold is above the physical error rate.

---

Once one has both a quantum code and a decoding procedure whose threshold is above the physical error rate

There are two ways of improving the logical error rate

Once one has both a quantum code and a decoding procedure whose threshold is above the physical error rate.

There are two ways of improving the logical error rate.

1. Doing usual concatenation, which can make non-planar stabilizers appear.

---

Once one has both a quantum code and a decoding procedure whose threshold is above the physical error rate,

There are two ways of improving the logical error rate:

1. Doing usual concatenation, which can make non-planar stabilizers appear.

2. In the case of topological codes, we can increase the distance, with a quadratic cost in the number of qubits used (BPT bound).

---

Note: The BPT bound refers to the Bravyi-Preskill-Terhal bound, which is a theoretical limit on the error rate that can be tolerated by a quantum error correction code. It is named after its contributors: Sergey Bravyi, David Preskill, and Bruno Terhal.
Once one has both a quantum code and a decoding procedure whose threshold is above the physical error rate

There are two ways of improving the logical error rate

Doing usual concatenation\(^1\), which can make non-planar stabilizers appear

In the case of topological codes, we can increase the distance, with a quadratic cost in the number of qubits used (BPT bound)

Topological concatenation is a hybrid of these two methods

\(^1\)Daniel Gottesman, A Theory of Fault-Tolerant Quantum Computation (1997)
Plain surgery merges topological codes while keeping encoded qubits.
Plain surgery merges topological codes while keeping encoded qubits.

The merge process can be tuned to trade between distance and ease of measurement.
Plain surgery merges topological codes while keeping encoded qubits.

The merge process can be tuned to trade between distance and ease of measurement.
Merging two color codes

- It is not obvious which form the stabilizers should have while concatenating two triangular color codes.
Merging two color codes

- It is not obvious which form the stabilizers should have while concatenating two triangular color codes.
- The dual view makes it clearer, as stabilizers are vertices and qubits are triangles.
Merging two color codes

- It is not obvious which form the stabilizers should have while concatenating two triangular color codes.
- The dual view makes it clearer, as stabilizers are vertices and qubits triangles.
- Boundary nodes are merged and stabilizers in the overlap regions are linked.
Merging two color codes

- It is not obvious which form the stabilizers should have while concatenating two triangular color codes.
- The dual view makes it clearer, as stabilizers are vertices and qubits triangles.
- Boundary nodes are merged and stabilizers in the overlap regions are linked.

\[ 1 + \alpha \]
\[ 1 - \alpha \]
Non-trivial measurements

- Y operators on several qubits can be combined to measure product operators
Non-trivial measurements

- Y operators on several qubits can be combined to measure product operators

Schematic of logical operators
Non-trivial measurements

- Y operators on several qubits can be combined to measure product operators

Schematic of logical operators
Non-trivial measurements

- Y operators on several qubits can be combined to measure product operators

![Schematic of logical operators](image-url)
Non-trivial measurements

- Y operators on several qubits can be combined to measure product operators

Schematic of logical operators
Non-trivial measurements

- Y operators on several qubits can be combined to measure product operators
- The product of the two logical operators can be represented by a red-string from left to right:
Non-trivial measurements
Upper level stabilizers

Stabilizers are product operators on some qubits:
Upper level stabilizers

We can have a look at the dual view:
We can then replace the physical qubits by logical ones:
Upper level stabilizers

We can apply the merge procedure to neighboring qubits:
Upper level stabilizers

In the primal view:
Upper level stabilizers

The large stabilizer can me measured by measuring colored edges in its surroundings:
Repeating the process around all the stabilizers with all qubits considered:
Characteristics of concatenated codes
Any kind of 2D triangular color code can be used at any level of encoding.
Characteristics of concatenated codes

Theoretical expectations

- Any kind of 2D triangular color code can be used at any level of encoding
- Suppose we use a $[n_1, 1, d_1]$ code as an upper-level template and $[n_0, 1, d_0]$ codes to encode physical qubits
Any kind of 2D triangular color code can be used at any level of encoding.

Suppose we use a $[n_1, 1, d_1]$ code as an upper-level template and $[n_0, 1, d_0]$ codes to encode physical qubits.

Suppose we use $\alpha$ to parametrize the concatenation.
Any kind of 2D triangular color code can be used at any level of encoding

Suppose we use a $[n_1, 1, d_1]$ code as an upper-level template and $[n_0, 1, d_0]$ codes to encode physical qubits

Suppose we use $\alpha$ to parametrize the concatenation

What do we expect to get for the concatenated code?
Theoretical expectations

- Any kind of 2D triangular color code can be used at any level of encoding.
- Suppose we use a $[n_1, 1, d_1]$ code as an upper-level template and $[n_0, 1, d_0]$ codes to encode physical qubits.
- Suppose we use $\alpha$ to parametrize the concatenation.
- What do we expect to get for the concatenated code?
Theoretical expectations

- Any kind of 2D triangular color code can be used at any level of encoding.
- Suppose we use a \([n_1, 1, d_1]\) code as an upper-level template and \([n_0, 1, d_0]\) codes to encode physical qubits.
- Suppose we use \(\alpha\) to parametrize the concatenation.
- What do we expect to get for the concatenated code?

\[
\begin{bmatrix}
n_1 & n_0
\end{bmatrix}
\]
- Any kind of 2D triangular color code can be used at any level of encoding
- Suppose we use a $[n_1, 1, d_1]$ code as an upper-level template and $[n_0, 1, d_0]$ codes to encode physical qubits
- Suppose we use $\alpha$ to parametrize the concatenation
- What do we expect to get for the concatenated code?

$$[[n_1 n_0, 1]]$$
Any kind of 2D triangular color code can be used at any level of encoding.

Suppose we use a $[[n_1, 1, d_1]]$ code as an upper-level template and $[[n_0, 1, d_0]]$ codes to encode physical qubits.

Suppose we use $\alpha$ to parametrize the concatenation.

What do we expect to get for the concatenated code?

$$[[n_1 n_0, 1, ?]]$$
Theoretical expectations

- Any kind of 2D triangular color code can be used at any level of encoding.
- Suppose we use a $[n_1, 1, d_1]$ code as an upper-level template and $[n_0, 1, d_0]$ codes to encode physical qubits.
- Suppose we use $\alpha$ to parametrize the concatenation.
- What do we expect to get for the concatenated code?

\[
[n_1 n_0, 1, f(\alpha, d_0, d_1)]
\]
Determining the distance of a code is hard as we need to find the smallest-weighted non-trivial logical operator.
- Determining the distance of a code is hard as we need to find the smallest-weighted non-trivial logical operator.
- The minimal weight can be upper bounded by the weight of the edge operator, which is easy to compute.
- Determining the distance of a code is hard as we need to find the smallest-weighted non-trivial logical operator.
- The minimal weight can be upper bounded by the weight of the edge operator, which is easy to compute.
- Determining the distance of a code is hard as we need to find the smallest-weighted non-trivial logical operator.
- The minimal weight can be upper bounded by the weight of the edge operator, which is easy to compute.

\[ d_0 \left( (d_1 - 1) \frac{\alpha + 1}{2} + 1 \right) \]
### Characteristics of concatenated codes

#### Distance evaluation

<table>
<thead>
<tr>
<th>Lattice type</th>
<th>$d_0 = d_1$</th>
<th>$\alpha$</th>
<th>Concatenated distance</th>
<th>$n$</th>
<th>Qubit gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>6-6-6</td>
<td>11</td>
<td>$\frac{3}{11}$</td>
<td>81</td>
<td>8281</td>
<td>+68%</td>
</tr>
<tr>
<td>6-6-6</td>
<td>11</td>
<td>$\frac{7}{11}$</td>
<td>101</td>
<td>8281</td>
<td>+8%</td>
</tr>
<tr>
<td>6-6-6</td>
<td>11</td>
<td>$\frac{9}{11}$</td>
<td>111</td>
<td>8281</td>
<td>-11%</td>
</tr>
<tr>
<td>6-6-6</td>
<td>111</td>
<td>$\frac{81}{111}$</td>
<td>10671</td>
<td>$85.10^6$</td>
<td>+0%</td>
</tr>
<tr>
<td>6-6-6</td>
<td>111</td>
<td>$\frac{91}{111}$</td>
<td>11221</td>
<td>$85.10^6$</td>
<td>-10%</td>
</tr>
<tr>
<td>4-8-8</td>
<td>11</td>
<td>$\frac{3}{11}$</td>
<td>81</td>
<td>5041</td>
<td>+50%</td>
</tr>
<tr>
<td>4-8-8</td>
<td>11</td>
<td>$\frac{7}{11}$</td>
<td>101</td>
<td>5041</td>
<td>-3%</td>
</tr>
<tr>
<td>4-8-8</td>
<td>11</td>
<td>$\frac{9}{11}$</td>
<td>111</td>
<td>5041</td>
<td>-20%</td>
</tr>
<tr>
<td>4-8-8</td>
<td>111</td>
<td>$\frac{49}{111}$</td>
<td>10671</td>
<td>$40.10^6$</td>
<td>-1%</td>
</tr>
<tr>
<td>4-8-8</td>
<td>111</td>
<td>$\frac{91}{111}$</td>
<td>11221</td>
<td>$40.10^6$</td>
<td>-37%</td>
</tr>
</tbody>
</table>
Decoding the concatenated color codes
A computationally efficient decoder for triangular color codes as been presented by \(^2\)

\(^2\) Chamberland, Triangular color codes on trivalent graphs with flag qubits (2020)
\(^3\) https://github.com/networkx/networkx
\(^4\) https://github.com/oscarhiggott/PyMatching
A computationally efficient decoder for triangular color codes as been presented by \(^2\)

Our Python implementation uses NetworkX\(^3\) for graphs and PyMatching\(^4\) for syndrome pairings.

\(^2\)Chamberland, Triangular color codes on trivalent graphs with flag qubits (2020)

\(^3\)https://github.com/networkx/networkx

\(^4\)https://github.com/oscarhiggott/PyMatching
Decoding the concatenated color codes

Decoder limitations
Logical error rate (non concatenated case)
Decoding the concatenated color codes

Logical error rate (concatenated case)

Concatenation of a distance 21 code with itself (39601 qubits)
Further prospects
Further prospects

- We might want to try using a better decoder.
Further prospects

- We might want to try using a better decoder
- We might want to try decoding recursively
Further prospects

- We might want to try using a better decoder
- We might want to try decoding recursively
- We might want to try choosing a different geometry for the upper level code (toric geometry)