Supplemental Material to "Twins Percolation for Qubit Losses in Topological Color Codes"

D. Vodola,[1] D. Amaro,[1] M.A. Martin-Delgado,[2] M. Müller,[1]
[1]*Department of Physics, Swansea University, Singleton Park, Swansea SA2 8PP, United Kingdom.*
[2]*Departamento de Física Teórica I, Universidad Complutense, 28040 Madrid, Spain.*

In this supplemental material we present important technical details as well as additional numerical results that are not shown in the main text, which underline the functioning and effectiveness of the qubit loss recovery algorithm for a variety of 2D color code lattices. Specifically, we start in Sec. I by providing background information on the 2D lattices and geometries we study numerically, and present in Sec. II schematic examples of the different methods used to identify equivalent logical operators. In Sec. III we sketch the basic idea of protocol that has been proposed in Ref. [S1] for the correction of qubit losses in Kitaev's toric code, in order to highlight in more detail the fundamental differences with the protocol we introduce in the main text for the color code. In Sec. IV we provide the details of why excitations of the merged and reduced stabilizers as a result of the lattice reconstruction algorithm are correlated and can be removed deterministically and without the use of a decoding algorithm. Finally, the algorithm used to compute the qubit loss thresholds is described in more detail in Sec. V and the numerical results obtained for 2D color codes defined on hexagonal (6.6.6) lattices, as well as for 6.6.6 and 4.8.8. color codes on planar triangular lattices, are presented and discussed.

I. COLOR CODE AND LATTICES

In this section we provide background information about color codes and about the lattices whose tolerance to qubit loss has been benchmarked: color codes defined on 4.8.8 and 6.6.6 lattices in their triangular and square versions [Fig. S1].

As described in the main text, color codes [S2] are defined on 2D lattices where qubits are located on the vertices. The underlying lattices are necessarily three-colorable and trivalent, so the plaquettes and their edges can be colored with colors $c = R, G, B$ in such a way that adjacent plaquettes have different colors c, and the color of each edge is complementary to the two colors of the two plaquettes that share the edge. Instances of regular, translationally invariant lattices are the 4.8.8 lattice [Figs. S1(a,b)], where plaquettes in the bulk are squares and octagons, and the 6.6.6 lattice [Figs. S1(c,d)], i.e. a honeycomb lattice. The number of physical qubits N_q forming these lattices is computed as a function of the distance d: $N_q = 2(d-1)^2 + 2$ for the square 4.8.8 lattice [Fig. S1(a)]; $N_q = (d^2-1)/2 + d$ for the triangular 4.8.8 lattice [Fig. S1(b)]; $N_q = 3d^2/2 - 2d + 2$ for the square 6.6.6 lattice [Fig. S1(c)]; $N_q = (3d^2+1)/4$ for the triangular 6.6.6 lattice [Fig. S1(d)].

Whereas these are translationally invariant lattices, the qubit loss correction algorithm and associated protocols introduced in the main text also apply to more general lattices called colexes [S3], which may not have any discrete symmetry.

The code space C where logical qubits are encoded is determined by a set of stabilizer operators defined on the lattice. For each plaquette P there are two stabilizers: $S_P^\sigma = \prod_{j\in P}\sigma_j$, where σ is a Pauli X or Z operator acting on all the qubits belonging to a given plaquette. All stabilizer operators mutually commute given that every plaquette contains an even number of qubits and different plaquettes share two or zero qubits. Thus, the stabilizer group $\mathcal{S}$ is generated by the set of independent stabilizer generators and the code space C is the common $+1$ eigenspace of all stabilizers.

The topology of the surface where the lattice is embedded uniquely determines the number k of logical qubits in C, independently of the system size and the bulk structure of the lattice hosting the code, via $k = 4 - 2\chi$ where χ is the Euler characteristic of the surface. For instance, color codes defined on square lattices like the ones shown in Figs. S1(a,c) encode two logical qubits, while the color codes on triangular lattices as the ones displayed in Figs. S1(b,d) encode one logical qubit.

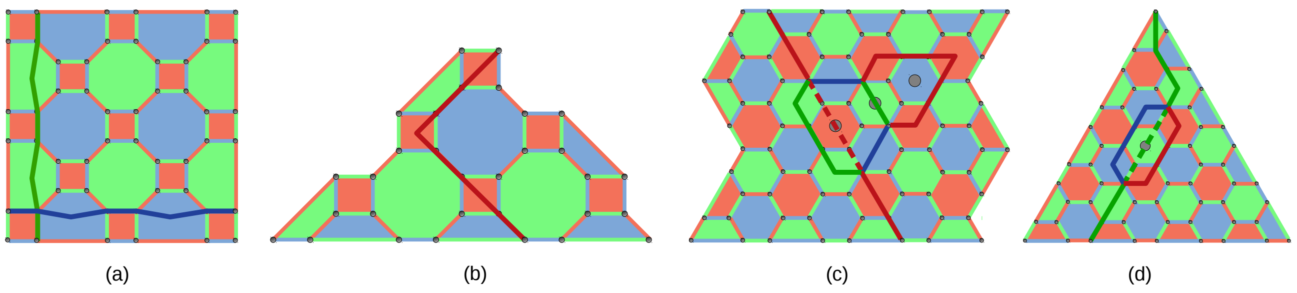

FIG. S1. Color code lattices studied for the qubit loss problem. (a) Square 4.8.8. lattice of logical distance $d = 6$ where the logical generators can be defined along a blue and a green path traversing the lattice from the left to the right, and from the upper to the lower boundaries, correspondingly. (b) Triangular 4.8.8. lattice of distance $d = 7$, with one red path. (c) Square 6.6.6 lattice of distance $d = 8$ where the red path has a second level branching due to the action of the plaquettes marked with a grey circle. (d) Triangular 6.6.6. lattice of distance $d = 9$ where the green path has a first level branching obtained by combining it with the green plaquette marked with a grey circle.

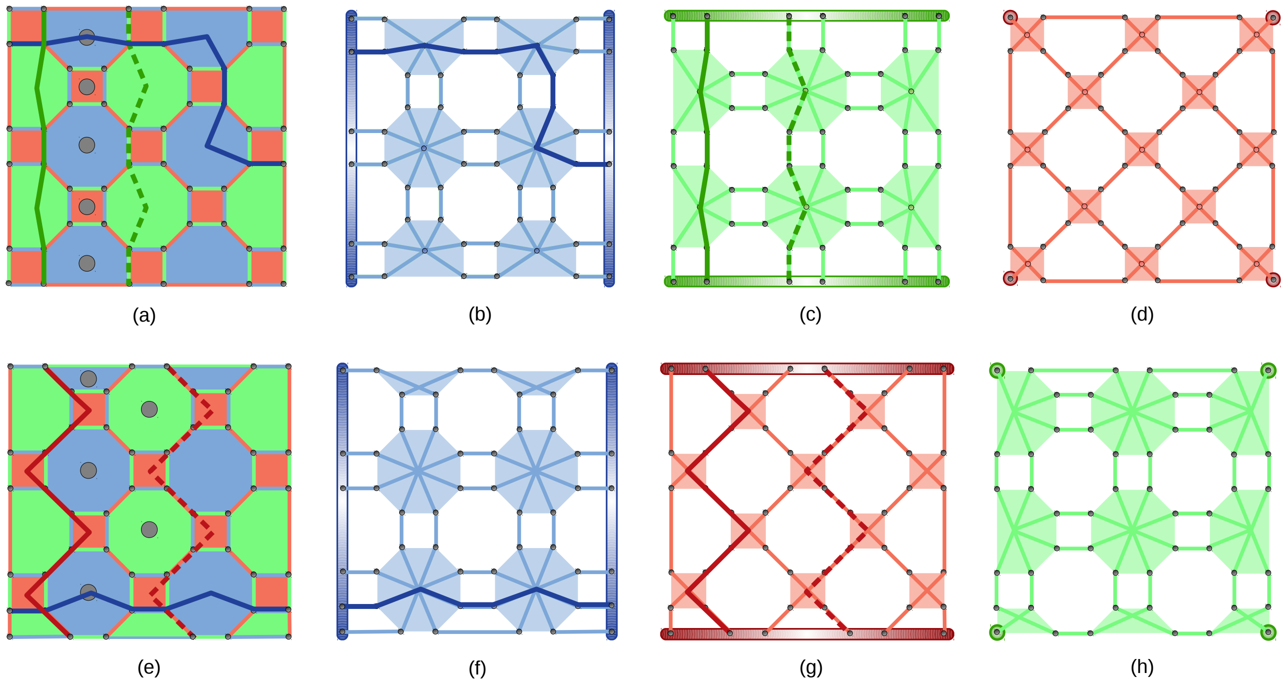

FIG. S2. Two instances of a 4.8.8 square color code with distance $d = 6$ and their shrunk lattices. (a) Square 4.8.8. color code lattice where blue and green logical operators are defined. The dashed green path has been deformed into the solid green line by the action of the plaquettes marked in grey. (b) Blue shrunk lattice with borders at the left and right sides of the lattice. The blue path goes from one border to the other. (c) Green shrunk lattice where borders are at the top and bottom sides of the lattice and the two greeen paths go from one to the other. (d) Red shrunk lattice. (e-h) A different instance of a square 4.8.8. color code, where the logical operators belong to the blue and red shrunk lattices. We notice that in both instances in the blue and green shrunk lattices of a 4.8.8 color code, each plaquette is connected to another (or to a border) by two links, while for the red shrunk lattice each plaquette is connected to another (or to a border) by only one link. The different structure of the red lattice as compared to the blue and green shrunk lattices is related to a lower threshold for the existence of a logical operator given by a percolating string [method (I) in the main text)].

In order to define the logical operators (generators) that generate the algebra of the encoded qubits it is convenient to introduce the concept of a shrunk lattice. A color code has three shrunk lattices, one for each color. For instance, the green shrunk lattice is obtained by placing a node at the centre of every green plaquette and connecting them through links [Figs. S2(c,h)]. Note that every link in the shrunk lattice corresponds to an edge of the original lattice of the same color. In planar codes, every shrunk lattice has at least two borders. The borders of a shrunk lattice consist of the qubits connected only through one link. For instance, the blue borders (the borders of the blue shrunk lattice) in Figs. S2(b,f) are the left and the right sides of the lattice. Borders can also consist of one qubit only. This is the case of the top right qubit in the triangular code of Fig. S1(b) and of the four corners of the square lattice in Fig. S2(d).

Having the definition of border in mind, we call any set of qubits that goes from one border to the other in the shrunk lattice of the color c [Fig. S2] a path Q_c. All paths have in common that (1) they share an even number of qubits with every plaquette, that (2) they share zero or an even number of qubits with paths of the same color. For each of the paths Q_c, we can define the operators $O_c^\sigma = \prod_{i\in Q_c}\sigma_i$ for $\sigma = X, Z$ that commute with the stabilizer group because of property (1). The $2k$ logical generators T_μ^σ required for the k logical qubits, can be then identified by choosing among all the O_c^σ the couples that satisfy the appropriate commutation and anticommutation relation for Pauli operators.

Importantly, two logical operators are equivalent if they differ by any element of the stabilizer $\mathcal{S}$, i.e., if there exists a subset $\mathcal{V} \subseteq \mathcal{S}$ of stabilizer operators such that

$$\tilde{T}_\mu^\sigma = T_\mu^\sigma \prod_{S\in\mathcal{V}} S. \tag{S1}$$

Equivalent logical operators satisfy the same (anti)commutation relations and have the same effect on the encoded data, but they have support on different sets of physical qubits $\tilde{Q}_c$. In particular, if a qubit i is in Q_c and a σ-stabilizer operator S_P^σ has support also on i, then $\tilde{T}_\mu^\sigma = T_\mu^\sigma S_P^\sigma$ will not have support on i given that for Pauli operators σ^2 is the identity.

In deforming a path Q_c of a given color c, two general cases are found by choosing different $\mathcal{V}$. If $\mathcal{V}$ is formed by plaquettes of color $c' \neq c$, the modified qubit set $\tilde{Q}_c$ will still be a colored path inside the shrunk lattice of the same color c and we say that $\tilde{Q}_c$ is a *deformation* of Q_c. For instance, in Fig. S2(a), the green dashed path has been deformed by red and blue plaquettes marked with grey circles.

If instead $\mathcal{V}$ is made by plaquettes of the same color as c we say that the path $\tilde{Q}_c$ undergoes *branching*. For example, a branched green path splits into a red and a blue string, which belong to their respective shrunk lattices. We call that a first level of branching [see Fig. S1(d)]. In the same way, red and blue strings can branch again, like in Fig. S1(c), where the red string branches into a green and a blue string, and we call this a second level of branching. This freedom in choosing $\mathcal{V}$ has been used in the protocols described in the main text for finding a modified $\tilde{T}^{\sigma}_{\mu}$ that does not have support on lost and twin qubits.

II. EXISTENCE CHECKS FOR EQUIVALENT LOGICAL OPERATORS

In this section we provide details about the three ways (I), (II) and (III) described in the main text to find a modified logical operator $\tilde{T}^{\sigma}_{\mu}$ by Eq. (S1) such that it does not have support on the set M of lost and twin qubits. We notice first that multiplying T^{σ}_{μ} by stabilizers $S^{\sigma'}_{P}$ of the other type $\sigma' \neq \sigma$ is of no help for avoiding missing qubits. Therefore, it suffices to consider only the multiplication with stabilizers of the same type. In this way we can simply consider the path Q_c that defines the logical operator T^{σ}_{μ} and conclude that T^{σ}_{μ} is still defined if there exists a set of plaquettes $\{P_j\}$ such that the modified path $\tilde{Q}_c$ obtained by

$$\tilde{Q}_c = Q_c \bigoplus_j P_j \tag{S2}$$

does not include any missing qubits. The symbol $\oplus$ denotes the symmetric difference, that is defined as $A \oplus B := (A \cup B) \setminus (A \cap B)$ on sets A and B. It is introduced because, for σ Pauli operators, σ^2 is the identity and thus it ensures that $\tilde{Q}_c$ does not contain qubits that appear an even number of times in the collection $\{Q_c, P_j\}$.

In Sec. I, we have seen that deformation and branching are the two strategies that paths have in order to avoid missing qubits. For instance, in Fig. S3(d), the red path is deformed to avoid the two losses that fall on it, while in Fig. S3(c), the blue path must

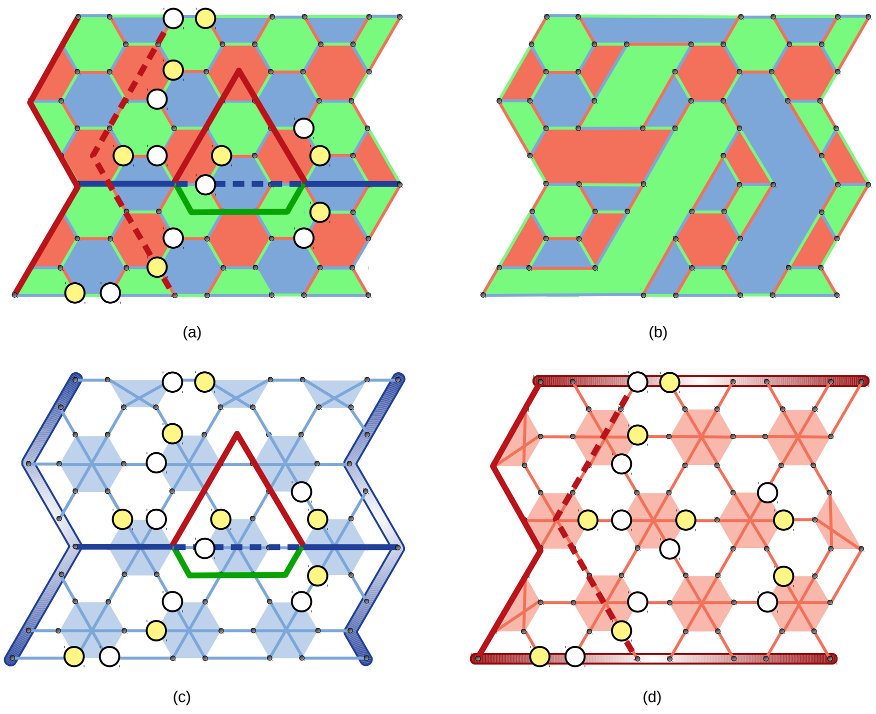

FIG. S3. Square 6.6.6. lattice undergoing the loss of the qubits marked by white circles. (a) In this particular example, for each qubit that is lost, a twin qubit marked by a yellow circle is chosen and also removed from the lattice. The dashed red and blue paths contain missing qubits, so the red path can be deformed into the solid red line and the blue can undergo branching to avoid the missing qubits. (b) Reconstructed lattice obtained with the protocol described in the main text. (c) Blue shrunk lattice. One can check that there is no blue path that goes from one border to the other without touching a missing qubit, which means that the blue shrunk lattice does not percolate. However, a blue logical operator manages to reach the other border by branching into the red and green shrunk lattices, and thereby without visiting any of the missing qubits (losses and twin qubits). (d) Red shrunk lattice where the red dashed path contains two missing qubits, but it can be deformed into the red solid path, which does not contain any losses or twin qubits. Therefore, for the shown scenario, this red shrunk lattice percolates.

branch to reach the other border. In order to study the robustness of color codes against losses, these two strategies were studied in three different methods described in the main text: (I) simple deformation, i.e., when paths of a color c can be deformed only inside the shrunk lattice of the same color, which is equivalent to a standard percolation check; (II) deformation and simple branching, i.e., situations in which paths of a color c can be modified by only the first and second level of branching; (III) searching exhaustively by use of Eq. (S2) also among all higher-level deformations and branching process, and therefore among all equivalent paths generated by the action of the whole stabilizer group $\mathcal{S}$.

Even though the number of equivalent paths grows exponentially with the number n of plaquettes, the search in (III) can be efficiently performed by mapping the plaquette sets $\{P_j\}$ and the paths Q_c into binary matrices and checking the solvability of a system of linear equations that we are going to introduce below. To this end, recall that N is the number of physical qubits, n is the number of independent plaquettes and

- let $\mathbb{Q}_c = (q_1, \ldots, q_N)^T$ be a binary column vector where $q_i \in \{0, 1\}$ is chosen such that $q_i = 1$ if the physical qubit i appears in the path Q_c, otherwise $q_i = 0$;
- let $\mathbb{P}_j = (p_{j1}, \ldots, p_{jN})^T$ be a binary column vector for $j = 1, \ldots, n$ with $p_{ji} \in \{0, 1\}$ and $p_{ji} = 1$ if the physical qubit i appears in the plaquette P_j, otherwise $p_{ji} = 0$;
- let $\mathbb{A}$ be a $N \times n$ binary matrix whose j-th column is equal to the column vector $\mathbb{P}_j$;
- let $\mathbb{M} = (m_1, \ldots, m_N)^T$ be a binary column vector where $m_i = 1$ if the i-th qubit is a lost or a twin qubit, otherwise $m_i = 0$.

The symmetric difference between Q_c and P_j is then mapped, for the binary vectors, to $\mathbb{Q}_c \oplus \mathbb{P}_j$ where $\oplus$ stands for sum modulo 2 and all the operations among binary matrices and vectors will be also performed modulo 2. Solving Eq. (S2) is then equivalent to finding a row binary vector x such that the binary column $\tilde{\mathbb{Q}}_c = \mathbb{A}x \oplus \mathbb{Q}_c$ that represents the modified path $\tilde{Q}_c$ will satisfy $\mathbb{M} \circ \tilde{\mathbb{Q}}_c = 0$, where the symbol $\circ$ denotes element-wise multiplication. This can be translated into the linear system

$$(\mathbb{M} \circ \mathbb{A})x = \mathbb{M} \circ \mathbb{Q}_c \tag{S3}$$

where the existence of solutions for x can be found efficiently by by standard Gaussian elimination algorithms, for which the runtime scales as $\sim N^3$ or less.

III. QUBIT LOSS IN THE KITAEV SURFACE CODE

In this section we briefly summarize main elements of the protocol pioneered in Refs. [S1, S4] to deal with qubit losses in the Kitaev code [S5]. We recall that in Kitaev's toric or surface code [Fig. S4(a)] the physical qubits reside on the edges of a lattice, the stabilizer group is generated by four-qubit Z-type plaquette (purple) and X-type vertex (yellow) operators and the logical operators correspond to product of Z (black) and X (gray) operator along non-trivial strings extending across the entire lattice.

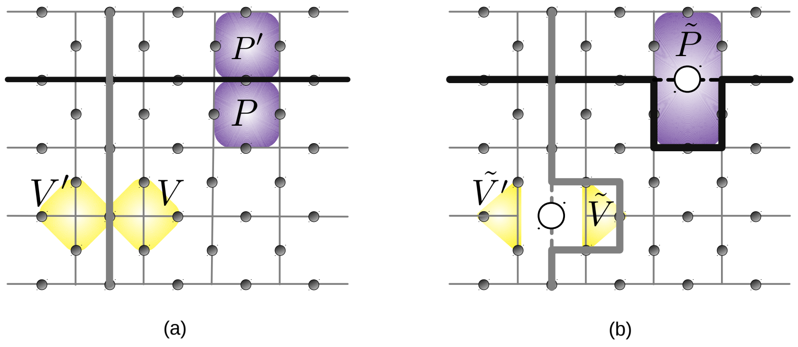

FIG. S4. Lattice structure of Kitaev's toric or surface code. (a) Physical qubits (gray circles) reside on the links of a square lattice and the stabilizers are defined on the plaquettes and vertices. Plaquette stabilizers are product of Pauli Z operators acting on the qubits in the plaquettes like P and P' marked in purple. Vertex stabilizers act with Pauli X operators on sets of neighbouring qubits (like V and V' marked in yellow). The Z-type logical operator is defined on the black path going from the left to the right border and the X-type logical operator is defined on the grey path going from the top to the bottom border. (b) According to the algorithm put forward by Stace *et al.* [S1], the lattice hosting the toric/surface code can be reconstructed by merging the plaquettes P, P' sharing a lost qubit into a super-plaquette $\tilde{P}$ like the one marked in purple, and modifying the vertex sets V, V' into $\tilde{V}, \tilde{V}'$ respectively to avoid the lost qubit. The X and Z logical operators are deformed by the action of the plaquette P and the vertex V respectively to avoid the positions (edges) corresponding to losses.

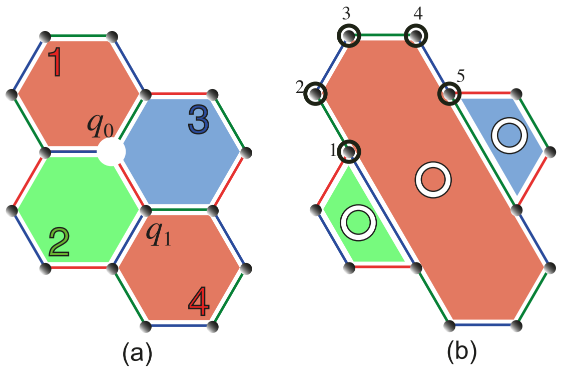

FIG. S5. (a) Subset Σ of the stabilizer group $\mathcal{S}$ showing four Z and X-stabilizers. The lost qubit q_0 is depicted as a white circle and the twin qubit is q_1. (b) After the lattice is reconstructed the plaquettes are in an undefined state and excitations (represented as hollow circles) can be present and must be removed. One possible correction for bringing the system back into the code space is a product of Pauli operators acting on the qubits shown as black circles: this string of Pauli operators shares an odd number of qubits with each of the three new plaquettes, therefore it anticommutes with the associated X/Z stabilizer generators of the complementary type Z/X, respectively.

Consider the lattice in Fig. S4(b), which is damaged by two losses marked as white circles. For each lost qubit q, which is shared by two plaquettes P and P', a new so-called super-plaquette $\tilde{P} = P \oplus P'$ that does not contain the lost qubit can be introduced, instead of P and P'. The symbol $\oplus$ indicates the symmetric difference between sets $A \oplus B := (A \cup B) \setminus (A \cap B)$. Additionally, the two vertex operators [V and V' shown in yellow in Fig. S4(a)], originally containing the lost qubit q, shrink and are converted into three-qubit X-type stabilizers defined on the qubit sets $\tilde{V} = V \setminus q$ and $\tilde{V}' = V' \setminus q$ that do not contain q. Overall, this defines a complete set of new stabilizers on the damaged lattice.

If a logical operator, like the one depicted in Fig. S4(a) with a solid line, has support on a loss, it can be deformed by the action of one or more plaquettes [in Fig. S4(b) it is deformed by P] such that it percolates through the entire lattice. In the case of the grey logical operator, it is deformed by the action of the vertex set V. This procedure will fail if finding a percolating path is not possible anymore. Note that logical operators never undergo branching in the surface/toric code. Therefore, the boundary between recoverable and non-recoverable losses is given in this case by the bond percolation threshold on a 2D square lattice, that is known to be $p_c = 1/2$ [S6].

IV. REMOVAL OF EXCITATIONS AND MODIFIED STABILIZER GROUP

After our algorithm for dealing with losses has been applied and the existence of logical operators has been checked so that the logical information is not corrupted [steps (i, ii, iii) of the algorithm as outlined in the main text], the code will be in an undetermined state as it might be affected by excitations (i.e. -1 eigenstates of the newly defined merged or shrunk X and Z-type stabilizer generators). If needed, these can be annihilated by physically applying a Pauli correction along a chain of qubits connecting the three new plaquettes, as shown in Fig. S5(b), or by a corresponding Pauli frame update on a software level [S7, S8]. This excitation removal, to re-initialize the QEC on the reconstructed lattice within the code space, does not require a decoder, is fully deterministic and iteratively applicable to more than a single qubit loss.

In this section, we will show how the excitations affect the stabilizers group $\mathcal{S}$ and what will be new stabilizers $\mathcal{S}'$ after removing them. One example of a reconstructed lattice affected by multiple losses is depicted in Fig. S3(b). We will describe the case of one single qubit loss, as it is straightforward to generalize it to more than one lost qubit.

Consider the subset $\Sigma = \{B_P^X, B_P^Z\}_{P=1...4} \subset \mathcal{S}$ for the section of the code reported in Fig. S5(a) and, say, the couple lost-twin qubits is represented by the pair (q_0, q_1). In order to end up in a valid code space C defined by independent and mutually commuting stabilizers, we have to build a new set Σ' that has no support on q_0 and q_1. Let us define dimer operators $D^\sigma = \sigma_{q_0}\sigma_{q_1}$ with $\sigma = X, Z$. Since D^X anticommutes with B_1^Z and B_4^Z, its eigenvalue $\epsilon_X = \pm 1$ is undetermined and for updating the generator set we replace B_1^Z with the product $B_1^Z B_4^Z$ and B_4^Z with $\epsilon_X D^X$. In the same way, the operator $D^Z = Z_{q_0} Z_{q_1}$ anticommutes with B_1^X and B_4^X, so we replace B_1^X with the product $B_1^X B_4^X$ and B_4^X with $\epsilon_Z D^Z$. We can now use the dimer operators D^X, D^Z to decouple the generators from the qubits q_0 and q_1. To do so, we multiply all the B_P^σ that contain q_0 and q_1 by $\epsilon_\sigma D^\sigma$. The final set of generators is

$$\Sigma = \{\epsilon_\sigma D^\sigma, \epsilon_\sigma D^\sigma B_1^\sigma B_4^\sigma, \epsilon_\sigma D^\sigma B_2^\sigma, \epsilon_\sigma D^\sigma B_3^\sigma, \sigma = X, Z\} \tag{S4}$$

and corresponds to the lattice reconstructed by the steps (i, ii) as well as the dimer formed by the lost and the twin qubit. This set of generators Σ will be then split into two disjoint subsets, one containing the two dimer operators $\{\epsilon_\sigma D^\sigma\}$, the other Σ'

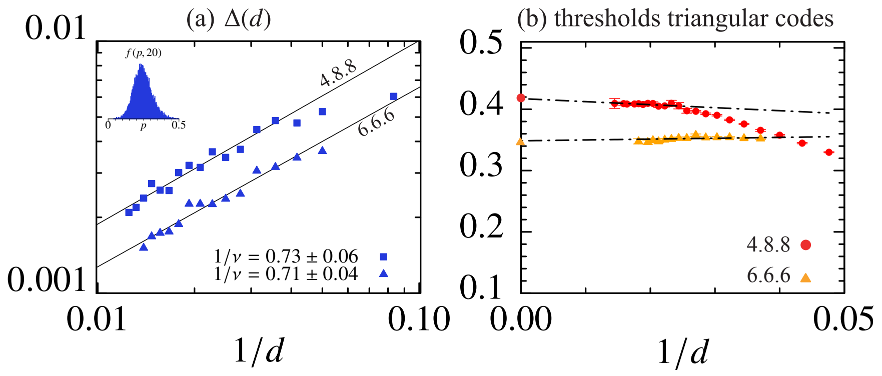

FIG. S6. (a) Standard deviation $\Delta(d)$ of the distribution $f(p,d)$ as a function of $1/d$ for the blue shrunk lattices of the 4.8.8 and the 6.6.6 lattices. For a code of distance d with a fraction p of losses, the quantity $f(p,d)\mathrm{d}p$ gives the probability that the lattice ceases to show a percolating path for the first time [S6] An example of $f(p,d)$ is represented in the inset for $d = 20$. For both the shrunk lattices, $\Delta(d) \propto 1/d^{1/\nu}$ with $1/\nu \approx 3/4$. The points are fit by the solid lines scaling as $1/d^{3/4}$. (b) Critical threshold scaling in the limit $d \to \infty$ for the triangular 4.8.8 and 6.6.6 lattices computed by solving Eq. (1) of the main text [equivalent to Eq. (S3)]. The values in the thermodynamic limit are represent by point on the vertical axis and coincide with the ones computed for the corresponding square lattice [See Fig. 3(a) of the main text and Fig. S7].

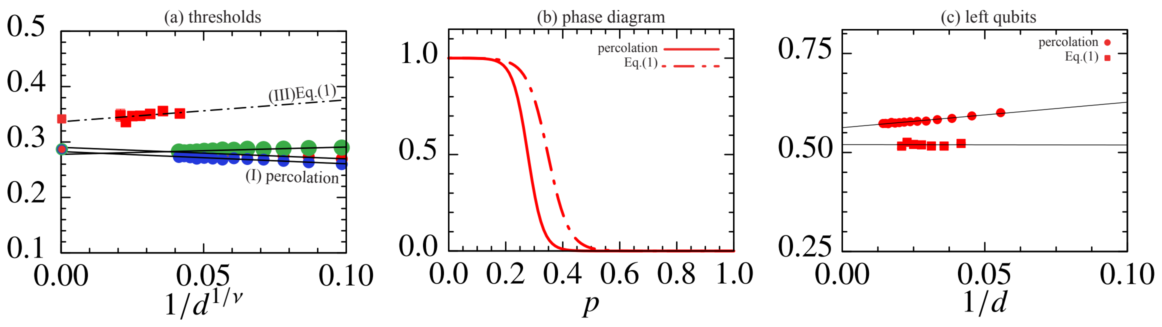

FIG. S7. Numerical results for the critical threshold p_∞ for the 6.6.6 square lattice. (a) Critical loss rate p_c as a function of the lattice distance d. Circles represent the percolating critical loss rate for the three shrunk lattices (points are fit by the solid lines scaling as $1/d^{3/4}$) while squares represent the fundamental thresholds from Eq. (1) of the main text for the red shrunk lattice (points are fit by the solid line scaling as $1/d$). (b) Phase diagram as a function of the loss rate p for a lattice with $d = 48$ representing the probability of finding a percolating path (solid line) and a solution of Eq. (1) of the main text (dot-dashed line). (c) Fraction of left qubits as a function of $1/d$. Again, squares represent results obtained from the algebraic method of Eq. (1) and circles are obtained with the percolating method.

having support on the remaining qubits of the reconstructed lattice [Fig. S5(b)]. The set Σ' contains generators that are in an undetermined state. In order to fix the state of the code, as all the three redefined σ-type generators show correlated excitations, one has to measure only one generator for each σ and thus detect whether an excitation $\epsilon_\sigma = -1$ is present. If it is the case, we then need to remove the excitations by bringing the state back to the code space. For example, if $\epsilon_Z = -1$, an X-type excitation will affect the new-defined plaquettes and one possible correction will be the product of X-Pauli operators on the three qubits as Fig. S5(b) shows. For more than a single loss, this argument can be easily iterated, with several binary unknowns ϵ being introduced, one for each of removed dimer.

V. ALGORITHM FOR COMPUTING THE CRITICAL LOSS RATE AND NUMERICAL RESULTS

In this section we explain in more detail the algorithm used to obtain the loss thresholds at finite code distance d and in the thermodynamic limit $d \to \infty$. We also present the results for the triangular and the 6.6.6 lattices that are not discussed in the main text. For each of the lattices in Fig. S1, we fix a logical distance d and find the critical loss rate $p_c(d)$ at which a logical operator can no longer be found according to each of the three methods explained in the main text and with more detail in Sec. II of this Supplemental Material.

We start by drawing a set of losses randomly with a rate $p_1 = 1/2$. Then, we reconstruct the lattice by following the protocol explained in the main text and after that we check if a logical operator still exists. If it is the case, the process is repeated, but using a bigger loss rate $p_2 = p_1 + 1/4$, otherwise, the loss rate is reduced by the same factor $p_2 = p_1 - 1/4$. For each round, the loss rate is increased (reduced) by half of the previous factor if we can (cannot) find an equivalent logical operator. The algorithm scales only logarithmically with the number of qubits in the lattice and halts when the number of lost qubits in two successive rounds does not change anymore. This algorithm gives a distribution $f(p,d)$ of loss rates at which a logical operator is found for the first time for a given fixed code distance. An example $f(p,d)$ is plotted in the inset of Fig. S6. The distribution $f(p,d)$ can also be interpreted as the probability that the logical operator ceases to be defined if the rate of losses is increased from p to $p + \mathrm{d}p$. Finally, we take the mean value of $f(p,d)$ as the critical loss rate $p_c(d)$ corresponding to the fixed distance d.

Regarding the methods (I) and (II) described in the main text and in Sec. II, in order to extract the value p_∞ of the threshold for $d \to \infty$, we use a finite-size scaling analysis borrowed from percolation theory [S6]. For p near the critical value p_∞, the correlation length ξ, characterizing the typical size of a path connecting two random plaquettes, is known to diverge as $\xi \propto |p - p_\infty|^{-\nu}$ for $d \to \infty$ with an exponent ν that percolation theory predicts to depend only on the dimensionality of the system. All the quantities showing a scaling law near p_∞ are controlled by the ratio d/ξ, i.e. by a scaling variable $z = (p - p_\infty)d^{1/\nu}$. In particular, let $F(p,d)$ be the probability that a distance-d code affected by a rate p of losses shows a percolating path. Clearly, the distribution $f(p,d)$ we computed numerically satisfies $f(p,d) = \mathrm{d}F/\mathrm{d}p$.

For $d \to \infty$, F will approach a step function with a discontinuity at p_∞ that separates a region ($p < p_\infty$) where a logical operator is always found from another ($p > p_\infty$) where it does not exist, while f will become a Dirac delta peaked around p_∞. For sufficiently large, but finite d and for p close enough to p_∞, $F(p,d)$ is expected to be a function only of z, i.e.$F(p,d) = F(z)$ and $f(p,d) = d^{1/\nu}\mathrm{d}F(z)/\mathrm{d}z$. From the last equation, one can see that $p_c(d)$, i.e. the mean value of $f(p,d)$, differs from p_∞ according to

$$p_c(d) - p_\infty \propto \frac{1}{d^{1/\nu}}, \tag{S5}$$

while the standard deviation of $f(p,d)$ will scale as $\Delta(d) \propto d^{-1/\nu}$. We use these last two equations to compute ν and p_∞ and thereby to determine the thresholds for qubit losses for the color codes when using a deformed or a branched logical operator.

Figure S6(a) shows $\Delta(d)$ as a function of $1/d$ in log-log scale for the blue shrunk lattice of the square 4.8.8 and the 6.6.6 codes. Similar plots are found for the other colored shrunk lattice. The slope of the curves gives the exponent $1/\nu$. Interestingly, one can see that $1/\nu$ for this generalized percolation problem takes the value of 3/4 expected for the standard percolation [S6].

When instead we determine the threshold by solving Eq. (1) of the main text [equivalent to Eq. (S3)], we also use Eq. (S5), but with an exponent $\nu = 1$ as no natural scaling law and thus critical exponent are expected.

A. Numerical results for the triangular codes and the 6.6.6 lattices

For the 4.8.8 and the 6.6.6 triangular codes, Fig. S6(b) shows the threshold rates obtained by solving Eq. (1) of the main text [equivalent to Eq. (S3)]. The dot-dashed lines represent fits scaling as $p_\infty + b/d$. For both the lattices, p_∞ can be read on the vertical axis.

Data for the 6.6.6 square lattice are reported in Fig. S7(a) that shows $p_c(d)$ as a function of $1/d^{1/\nu}$ computed via the methods (I) and (III) described in the main text and in Sec. S3. Linear fits whose intercept yield p_∞ are also drawn. As for the 4.8.8 lattice the method (III), the one based on checking whether the system in Eq. (1) of the main text admits solutions, yields a higher threshold. For a lattice with $d = 48$, Fig. S7(b) shows the probability of finding a red logical operator by using method (I), i.e., simple percolation and method (III). It then represents the phase diagram describing the boundary separating a region where the qubit associated to the logical red operator can be successfully recovered from a region where it is completely destroyed. For completeness, Fig. S7(c) shows the fraction of qubits left in the lattice as a function of $1/d$. Notably, this number approaches the limiting value of 50% when considering the solution of system of Eq. (1), demonstrating the high robustness of color codes against qubit loss.

[S1] T. M. Stace, S. D. Barrett, and A. C. Doherty, Phys. Rev. Lett. **102**, 200501 (2009).
[S2] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).
[S3] H. Bombin and M. A. Martin-Delgado, Phys. Rev. B **75**, 075103 (2007).
[S4] T. M. Stace and S. D. Barrett, Phys. Rev. A **81**, 022317 (2010).
[S5] A. Kitaev, Ann. Phys. **303**, 2 (2003).
[S6] D. Stauffer, *Introduction to percolation theory* (Taylor & Francis, London, 1985).
[S7] E. Knill, Nature **434**, 39 EP (2005).
[S8] P. Aliferis, D. Gottesman, and J. Preskill, Quantum Info. Comput. **6**, 97 (2006).