Intro to qLDPC Physics

Rio Weil

This document was typeset on November 20, 2025

Introduction:

Lecture notes for the physics of qLDPC codes I gave at group meeting, Frankenstein'ed together from results in recent literature + presentations given by Michael Vasmer, Vedika Khemani, and Chao Yin.

Contents

1 Motivation

Definition: LDPC Codes

A family of stabilizer QECCs $\{Q_i\}$ is LDPC if for every Q_i, there exists a generating set S_i such that:

1. $\forall s \in S_i$ then $\mathrm{Wt}(s) \leq r$ for some constant r (the stabilizers have bounded weight)
2. $\forall q \in [n_i]$, $|\{s \in S_i : q \in \mathrm{supp}(s)\}| \leq c$ for some constant c (qubit degree - each qubit should only be in the support of a constant number of checks).

Useful because they allow for fault-tolerant constructions (bounded stabilizer weight and qubit degree allow for errors to not propagate). Prototypical examples: The Ising model/repetition code in the classical case (where the stabilizers are just Z-checks and the qubits are bits) with $[n,k,d] = [n,1,n]$ and in the quantum case the surface codes with $[[n,k,d]] = [[n,O(1),\Theta(\sqrt{n})]]$.

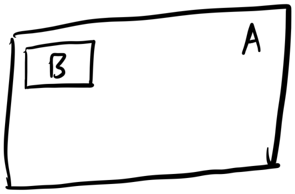
But the search for "good" codes with $k,d = \Theta(n)$ is harder, and limited by locality; the classical/quantum BPT bound 0909.5200 :

$$kd^{1/D} \leq cn \quad \text{(classical)} \tag{1.1}$$

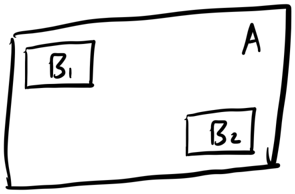
$$kd^{2/(D-1)} \leq cn \quad \text{(quantum)} \tag{1.2}$$

tells us that finite-D Euclidean models cannot correspond to good codes. Indeed, the Ising model and surface code saturate these bounds, even though their encoding rate is zero. Indeed, the limiting factor appears to be the surface area/volume ratio:

Proof sketch. Consider a 2D Euclidean space (can be D-dimensional, but easier to draw):



Then, $|B| = l^D \sim d < d$. If two ground states agree on A, they must agree everywhere. Now, consider two B boxes:



Since the checks are geometrically local, they overlap with at most one B_i block. Even though the size of the two Bs together may be larger than d, since they do not overlap, it is still the case that if the ground states agree on $A \setminus B_1 \setminus B_2$ that they agree everywhere. Now, we can tile the entire system with B blocks. We then observe that:

$$\frac{k}{n} \leq \frac{|A|}{n} \tag{1.3}$$

because we can only choose things independently in A to specify the codewords. But the size of A here is $\frac{|\partial B_i|}{|B|_i}$, and thus:

$$\frac{k}{n} \leq \frac{|A|}{n} \sim \frac{|\partial B_i|}{|B|_i} \sim \frac{1}{l} \sim \frac{1}{d^{1/D}}. \tag{1.4}$$

□

This motivates us to think about non-locality (exactly "how much" locality in the sense of what kind of long-range interactions do you need at minimum appears to be explored in 2409.15203). In particular gives a natural motivation to consider codes on expander graphs, where the surface-to-volume ratio goes to 1; hence we can circumvent above bounds that rely on geometric locality of checks (which we replace with a notion of graph locality/density). In the classical setting, this was done quite a while ago:

- (1960s Gallager) probabilistic construction, (c, b) regular random bipartite graph as the tanner graph yields asymptotically good code.
- (1990s Sipser-Spielman) consider Tanner codes, specified by a sufficiently expandering unerlying graph of fixed degree and a good "local code" on every vertex of the code. Good graph expansion quantified by the Cheeger constant. Defined on graph $G(V, E)$ as:

  $$h(G) = \min \left\{ \frac{|\partial S|}{|S|} : S \subset V, |S| < \frac{n}{2} \right\} \tag{1.5}$$

  where $\partial S \in E$ are the set of edges connecting S to its complement. We can also consider the spectral expansion: eigenvalues of the adjacency matrix:

  $$s = \lambda_1 \geq \lambda_2 \geq \lambda_3 \dots \tag{1.6}$$

  A large spectral gap is characterized by a better expansion:

  $$\lambda_1 - \lambda_2 \leq 2h(G) \tag{1.7}$$

Much more recently, work in developing good qLDPC codes, e.g. 2111.03654 and 2202.13641 asymptotically, as well as 2308.07915 (IBM) for targeting actual realization

Seems a bit theoretical/unrealistic, but experiments with synthetic/designed quantum systems can indeed realize:

- Systems where you can shuttle different components (neutral atoms, trapped ions)
- QED on hyperbolic lattices
- Optical cavities with all-to-all interactions
- IBM staking their entire quantum future on the ability to do long-range interactions with low noise, because their roadmap is based on an instance of a bivariate bicycle code: $[[144, 12, 12]]$ "Gross" code (toric code + additional nonlocal connections) 2308.07915, 2506.03094

Ok lots of background - motivates some questions on the physics side:

1. Can we understand the construction of LDPC codes from a more physical perspective?
2. Local codes have been amenable to study as stable phases of error-correcting matter (e.g. the toric code) - do analogous results hold in k-local settings?
3. What exotic physical properties do good LDPC/qLDPC codes possess?

2 Physical picture of LDPC constructions

Based on the two-part paper by Vedika/Tibor, given in: 2310.16032, 2402.16831.

General upshot of the work is a physical "code factory" where one inputs in graphs, defines classical codes on those graphs, then takes products (of various types) on these graphs, then can perform physical operations (e.g. "modding out" symmetries, gauging, or higgsing) to get quantum codes and models of various types.

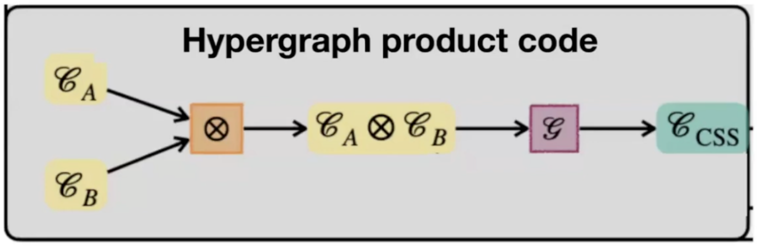
A prominent example; in the CS literature, a very common construction for quantum codes is the "hypergraph product code." Usually, it is formulated coding-theoretically in terms of parity check matrices. But with our physical code factory, we can now interpret it as taking the tensor product of two codes $\mathcal{C}_A, \mathcal{C}_B$ to form $\mathcal{C}_A \otimes \mathcal{C}_B$, and then gauging it to get a CSS code, with X/Z codes given by:

$$\mathcal{C}_X = (\mathcal{C}_A \otimes \mathcal{C}_B)^T, \quad \mathcal{C}_Z = (\mathcal{C}_A^T \otimes \mathcal{C}_B^T)^T \tag{2.1}$$

$$d_X = \min\{d_A, d_B\}, \quad d_Z = \min\left\{d_A^T, d_B^T\right\} \tag{2.2}$$

and overall rate:

$$k_Q = k_A k_B^T + k_A^T k_B \tag{2.3}$$



Let's go through the simplest possible example of this to see how this factory works.

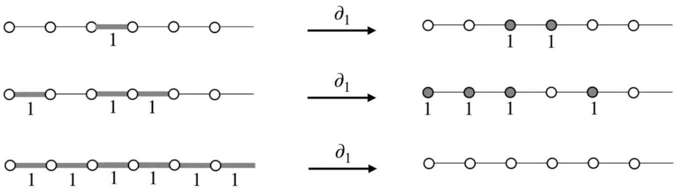
2.1 Ising model as a classical code/chain complex

First - characterization of Ising model in terms of Chain complexes.

Bits define an n-dim vector space $V_0 = \mathbb{F}_2^n$. There is a basis vector for each bit $\{|i\rangle\}_i$, and each vector $\vec{b} = (010100\dots b_n)^T$ picks out a subset of bits.

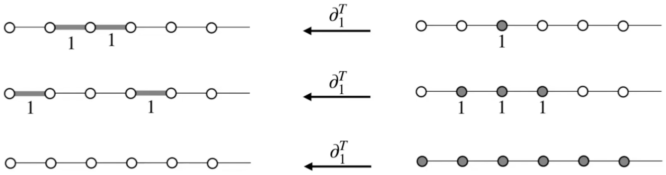
Checks define an m-dimensional vector space $V_1 = \mathbb{F}_2^m$. There is a basis vector for each check $\{|a\rangle\}_a$, and each vector $|c\rangle = (01110\dots c_m)^T$ identifies a subset of checks. The map $\partial_1 : \mathbb{F}_2^m \to \mathbb{F}_2^n$ is the transpose of the parity check matrix, and what it does is $\partial_1|c\rangle = |b\rangle$ where $|b\rangle$ is the subset of bits in the support of the checks $|c\rangle$. The transpose map $\partial_1^T : \mathbb{F}_2^n \to \mathbb{F}_2^m$ (which is just the parity check matrix) does the opposite, mapping from bits to checks $\partial_1^T|b\rangle = |c\rangle$, where $|c\rangle$ are the subset of bits triggered by the bits $|b\rangle$, i.e. the syndrome.

If we look at the 1-D Ising model, we have bits on 0-cells (sites) and checks on 1-cells (edges). We can consider the boundary map ∂_1:



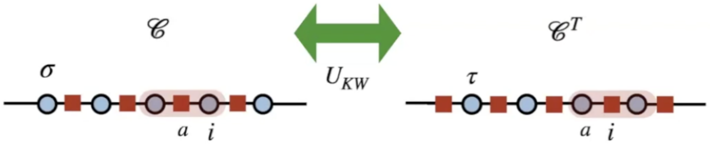
The redundancy is a collection of checks which does not act on any bit (here the product of all checks). ker(∂_1) defines a basis for the vector space of redundancies.

We can also consider the transpose map ∂_1^T:



The symmetry of the model is a collection of (flipped) bits which does not trigger any checks. ker(∂_1^T) defines a basis for the vector space of symmetries.

We can now observe that this transformation is nothing more than a Kramers-Wannier duality! An excitation in the original code becomes a degree of freedom in the transpose code, once we swap the bit and the checks.



$\mathcal{C}$	$\mathcal{C}^T$
m checks (δ): $C_a = \sigma^z_{i-1}\sigma^z_i$	m bits τ^z_a
n bits σ^x_i	n checks (δ^T) $A_i = \tau^x_a\tau^x_{a+1}$
Symmetry $\prod_i \sigma^x_i$	Redundancy $\prod_i A_i = 1$
Redundancy $\prod_a C_a = 1$	Symmetry $\prod_i \tau^z_i$

Note that to define the unitary between the two subspaces, we have to work in the symetric subspace. Then:

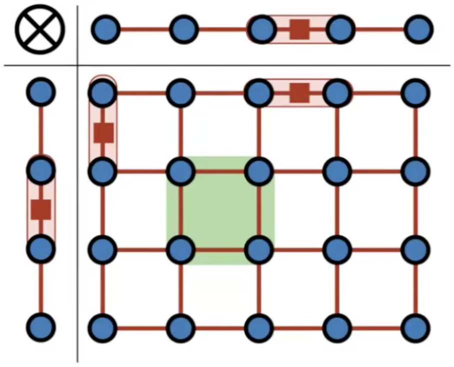
$$\dim(\mathcal{H}_\sigma^{sym}) = 2^{n-k} = 2^{m-k^T} = \dim(\mathcal{H}_\tau^{sym}) \tag{2.4}$$

Gauging generally is exactly this - we want to turn symmetry defects (e.g. domain walls) into dynamical degrees of freedom. Another way to think about it is to "make global symmetries local". As we discussed, we can use the transpose $\sigma \leftrightarrow \sigma^T$ to get a new model with the bits and checks swapped. The KW unitary U_{KW} maps between the symmetric subspaces of H, H^T. The last step to get a full gauge theory in terms of the gauge degrees of freedom is to remove this constraint to map between symmetric subspaces.

Note that for the 1D Ising model, this prescription does not yield a non-trivial gauge theory (it just gives us the 1D Ising model back, with $J \leftrightarrow h$ swapped). Thus we need something extra to get a non-trivial gauge theory out; in particular, we require the degrees of freedom to be extended, rather than point like (loop-like in the simplest case).

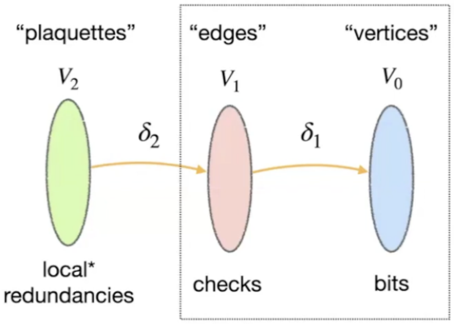
2.2 Taking a product to get the 2D Ising model

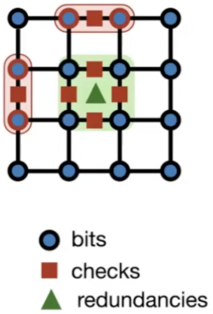
We consider the simplest product construction, the tensor product; we can use this to construct a 2D system out of two 1D systems, e.g. building up the 2D Ising model out of 2 copies of the 1D Ising model.



The tensor product creates local redundancies, increasing the dimension of the chain complex by one $D_c \to D_c + 1$.

We now have a chain complex of dimension 2 - As in our 2D Ising model, we can add to our chain complex a V_2 vector space corresponding to local (few-body, not necessarily geometrically) reduncies of checks.





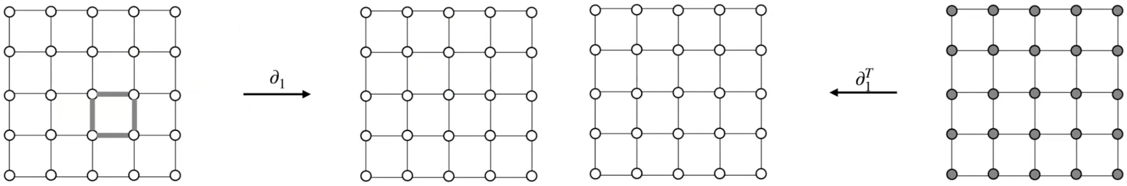
Then:

$$\delta_1 \delta_2 = 0 \tag{2.5}$$

and $\ker(\delta_1)$ defines a basis for the vector space of redundancies (collections of checks that do not flip any bits), with:

$$B_1 \equiv \mathrm{Im}(\delta_2) \subset \ker(\delta_1) \equiv C_1 \tag{2.6}$$

and $\ker(\delta_1^T)$ defines a basis for the vector space of symmetries (collections of flipped bits which do not trigger any checks).

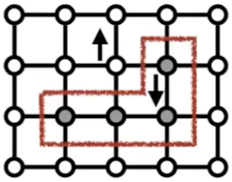


In the 2D Ising model, each plaquette corresponds to a local redundancy $\prod_{a \in p} C_a = +1$. Thus the domain walls in the model are loop like. This gives rise to the thermal stability of the Ising model, as well as the fact that it yields a non-trivial gauge theory upon gauging. In particular, gauging the 2D Ising model gives us a $\mathbb{Z}_2$-gauge theory, which corresponds to the toric code. Pictorially, the uniform superposition over loop configuration in the ground state of the toric code arises from gauging the loop-like degrees of freedom in the Ising model.

The beauty of the formalism is that - even though we used the Ising model here as an example - the formalism only cares about the (boundary) maps and so the gauging procedure applies to LDPC codes and codes on arbitrary geometries.

2.3 Gauging the 2D Ising model to get the Toric code

When we look at the KW duality for the 2D Ising model, we go from the 2D Ising model $\mathcal{C}$ with one global ($\mathbb{Z}_2$) symmetry of the Ising model with many local redundancies to the toric code $\mathcal{C}^T$ with one global redundancy (obtained by multiplying out all star operators/checks $\prod_v A_v = \mathbb{I}$) and many local symmetries (the B_p plaquettes).



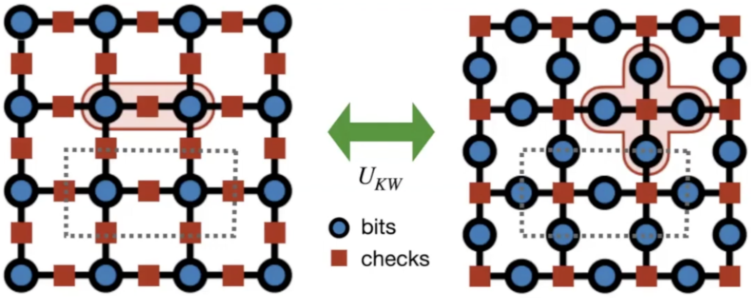
Doing the KW duality, we go from the Ising Hamiltonian:

$$H_{\text{Ising}} = -J \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - g \sum_i \sigma_i^x \tag{2.7}$$

where local redundancies $\implies$ domain walls form closed loops, to the TC Hamiltonian:

$$\tilde{H}_{TC} = -J \sum_{\langle ij \rangle} Z_{ij} - h \sum_i A_i^{(x)} \tag{2.8}$$

where local symmetries being enforced $\implies$ $B_p^z = +1$ (no vortex excitation).



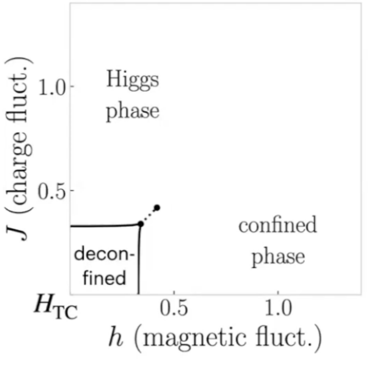
If we want to allow for vortex excitation, we follow the standard minimal coupling prescription:

1. Couple interactions to gauge fields $\sigma_i^z\sigma_j^z \to \sigma_i^z\sigma_j^z \otimes Z_{ij}$. This takes the global Ising symmetry and promotes it to a local symmetry.
2. Add local gauge invariant terms $-K\sum_p B_p, -h\sum_{\langle ij\rangle} X_{ij}$. The latter makes the background gauge fields dynamical and allows them to fluctuate (lifting the no vortex excitation condition). The former is needed for the toric code, and comes from the redundancy of the original Ising model. By the time we take local products, all matter fields/σ cancel out, leaving us with just the B_p.
3. Enforce the local Gauss law $A_i^{(x)}\sigma_i^x = 1$

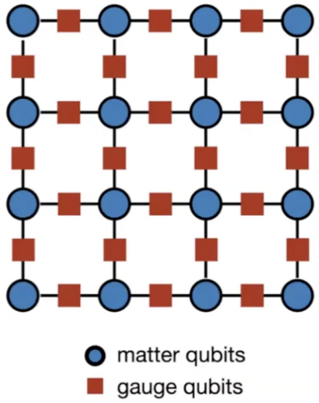
So, promoting our Hamiltonian to:

$$H = -\sum_i \sigma_i^x - h\sum_{\langle ij\rangle} X_{ij} - J\sum_{\langle ij\rangle} \sigma_i^z Z_{ij}\sigma_j^z - \sum_p B_p \tag{2.9}$$

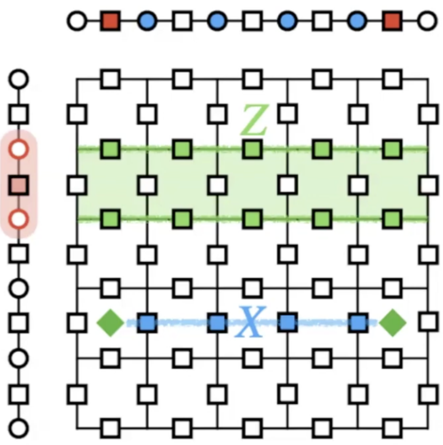
we get a full gauge theory for $\mathbb{Z}_2$ matter coupled to $\mathbb{Z}_2$ gauge fields.



Fradkin, Shenker: PRD 1979



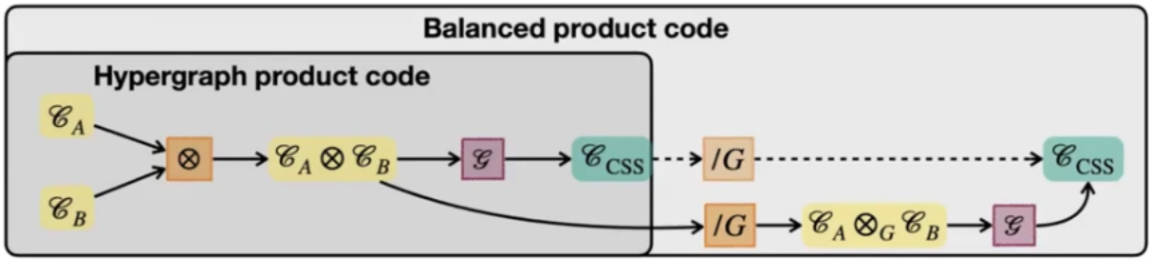
Thus we can understand the toric code as the hypergraph product of two 1D Ising models/repetition codes. The Z logicals are derived from the global redundancies of the 1D Ising model (and the fact that they can be deformed arises from multiplying via a check of the 1D Ising model), while the X logicals are derived from the global symmetries of the 1D Ising model. The deconfined point-like (anyonic e/m) excitations arise from the domain walls of the 1D Ising model.



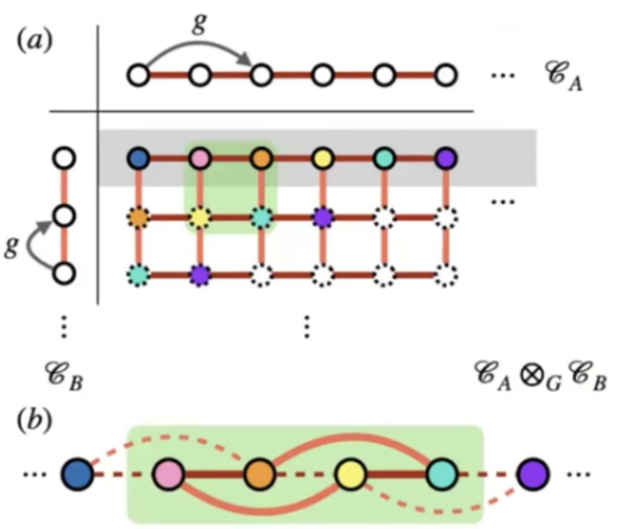
The input code parameters being $[n, 1, n]$ enforces that under the HGP we get code parameters $[n^2, 2, n]$.

2.4 More facets of the factory + good qLDPC codes

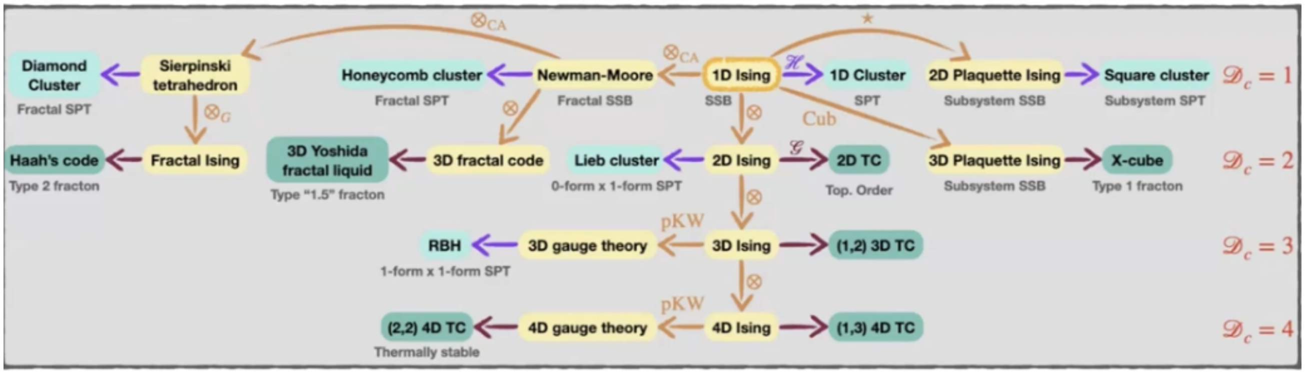
- More types of products; tensor products were straightforwards and introduced local redundancies. Check products or cellular automota products can introduce subsystem symmetries. Cubic products (check products in 3D/taking in 3 classical codes) can introduce both..
- In addition to the "Gauging" map of classical to quantum there is also a "Higgsing" map which takes the tanner graph of the classical code and associates it with the graph state.
- The HGP construction above, if we take two good classical LDPC codes as input, would give $k = \Theta(n)$ and $d \sim \sqrt{n}$. One way to boost this is through the so-called "balance product code", introduced in 2012.09271 and used to construct good qLDPC and locally testable good LDPC codes in 2111.03654 (remark: every good qLDPC code is a "gauge dual" of a locally testable code). Idea is that the input codes have enhanced symmetry (permutation of bits that leave the code invariant - obtained from the Cayley graph of the code). The symmetry is then modded out after taking the product (reducing the number of bits), which boosts the distance (actually the relative distance d/n by reducing n). This is depicted below for the case of translation invariance, where the symmetry can be "modded out" after taking the product, removing bits and making the system quasi-1D with non-local connections.



Key point: boost relative distance d/n by reducing n



Through the work they get quite the zoo of known models related through homology, products, and gauging/higgsing:



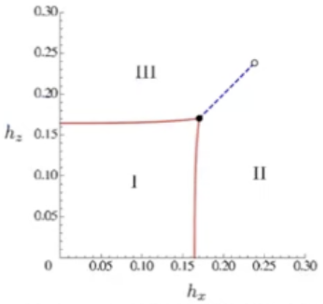
where replacing the 1D Ising model at the root of the tree can yield constructions of non-Euclidean models and phases of matter.

3 qLDPC codes as stable phases of matter

3.1 Stability of Topological Orders

That the toric code has had a stable phase around ithas been known for quite a while; consider perturbed TC Hamiltonian:

$$H_{\text{TC}} + V = -\sum_v A_v - \sum_p B_p - h_x \sum_i X_i - h_z \sum_i Z_i \tag{3.1}$$



where we have a phase around the toric code (I - deconfined) with topological order (vs. magnetically ordered phases (II - confined)/(III - Higgs), wherein the fluxes (m)/charges (e)/condense). Even more generally, it has been known since 2010 (Brayvi/Hastings/Michalakis, 1001.0344, 1001.4363) that topological orders are provably stable to local perturbations $\epsilon \sum_i V_i$, involving (a) a gap that does not close between the ground space and the excited states and (b) exponentially small splitting in the system size between the ground states.

(b) is easy to see intuitively from the topological order condition. All ground states are locally indistinguishable, so to have non-vanishing matrix elements between ground states we require Lth order perturbation theory, where we find splitting of order $\epsilon^L = e^{-L\log(1/\epsilon)}$.

(a) Seems a little more technical. The proof has not been digested, but the overall sketch:

- Spectral gap is stable to relatively bounded perturbations $\|W\psi\| \leq b\|H_0\psi\|$ for $0 \leq b < 1$ (invoking Lieb-Robinson).
- For Hamiltonians with TO, perturbations that are locally block-diagonal (i.e. all terms in the expansion preserve the ground state subspace) are relatively bounded.
- General local perturbations can be shown to be local block diagonal + negligible error.

Beyond my roundhouse, but can allegedly also use QFT arguments[1] which do not work for general geometries/relies on geometric locality.

3.2 Stability of qLDPC codes

So - what of this result when we go to a non-local setting?

- First stab appears to be from 2405.19412, which follow the Brayvi-Hastings proof but relaxing the graph structure. They prove stability so long as $\exists \epsilon_1, \epsilon_2 > 0$ such that the size of balls of radius r on the interaction graph are upper bounded by $\Gamma(r) = O(\exp(r^{1-\epsilon_1}))$ (c.f. $\Gamma(r) = O(\text{poly}(r))$ in the Euclidean case) and balls of radius $O(\log(n)^{1+\epsilon_2})$ are locally indistinguishable. Unfortunately hyperbolic surface code and good LDPC constructions have $\epsilon_1 = 0$ so falls just short. The problem appears to be the use of quasiadiabatic evolution, as LR bounds have exponential decay in the diameter, not the volume of the set, which results in quasipolynomial bounds on energy splitting for expanders (rather than exponential).
- A more physically heuristic argument is given in 2406.15757. I haven't read it in detail, but the idea seems to be (a la Dennis, Kitaev, Preskill) to associate (general) CSS stabilizer Hamiltonians with classical stat mech models with quenched disorder. They consider purely h_x/h_z local magnetic fields (drawing a correspondence to the stat mech models under pure bit/phase flip noise), and suggest the use of decodability success as evidence for an ordered phase (in some sense a bit of an operational definition).

[1] The stability of the spectral gap "can be understood as a consequence of the fact that there exists a stable quantum field theory at long wavelengths that gives a well-defined description of the low-energy physics of geometrically local topological systems in the thermodynamic limit." 2405.19412

The full theorem is given in 2411.01002:

Theorem: Stability of good qLDPC codes

Let H_0 be a stabilizer Hamiltonian of a $[[n,k,d]]$ quantum LDPC code with $\Omega(1)$ gap, with:

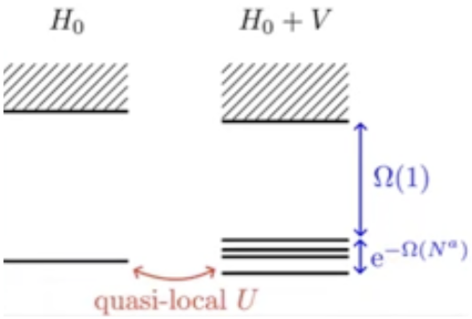
$$d = \Omega(\log n) \tag{3.2}$$

and H_0 has better than quadratic "check soundness" $a < 2$. Then, $\exists \epsilon_0$ such that for V with local strength $\epsilon < \epsilon_0$, $H_0 + V$ still has 2^k low-energy states, separated by gap $\geq 1/2$ from the remainder of the spectrum. The lowest energy states have small splitting:

$$\delta E = \epsilon e^{-\Omega(d)} \tag{3.3}$$

(if $d = n^\alpha$ then this means splitting is exponential) and are connected to the unperturbed ground states via quasi-local unitary U whose generator has $O(\epsilon)$ local strength.
This is known to apply to

1. All known good qLDPC codes with $d = \Theta(n)$
2. hypergraph-product expander codes with $d = \Theta(\sqrt{n})$ (Tillich and Zemor 0903.0566)
3. Finite-dimensional codes with local TO.



where "local strength" for a perturbation $V = \sum_{S,\mathbf{s}} V_{S,\mathbf{s}}$ is:

$$\epsilon = \max_i \sum_{i \in S} \sum_{\mathbf{s}} \left\| V_{S,\mathbf{s}} \right\| e^{\kappa |S|} \tag{3.4}$$

where $V_{S,\mathbf{s}}$ acts on qubit S and has error syndrome $\mathbf{s}$ (all terms in $V_{S,\mathbf{s}}$ anticommute with the same parity checks) and $\kappa = O(1)$. Note that this is a requirement that the perturbations decay with the volume, not the diamater of S.

And a stabilizer code has (d_c, f) check soundness if stabilizers acting on $M < d_c$ sites is equal to a product of $\leq f(M)$ parity checks, where generally we take $f(M) = M^a$ and for general qLDPC stability $a < 2$. In finite dimensions, any polynomial $f(x)$ is ok for stability (so the construction applies to the toric code with $a = 2$ and product of parity checks in a region R acts along ∂R and the area scales as the square of the perimeter). For good qLDPC constructions $a = 1$ so the proof applies.

Full proof is again quite technical, instead just give a sketch. We desire quasi-local U that maps from original Hamiltonian to the new one:

$$U^\dagger H_0 U = H_0 + D_{m_*} + V_{m_*} \tag{3.5}$$

with U quasi-local, D_{m_*}, V_{m_*} quasi-local, D_{m_*} block diagonal between the old/new codespace, $H_0 + D_{m_*}$ retaining the codespace as a gapped 2^k-degenerate ground state subspace, and $\|V_{m*}\| = e^{-\Omega(d)}$ such that $\delta E = \epsilon E^{-\Omega(d)}$.

This is constructed via iterated rotations $U_{m_*} = U_1 U_2 \dots U_{m_*-1}$ - perturbation theory via Schrieffer-Wolff transformation/rotations. For example on the first step:

$$U_1^\dagger (H_0 + V) U_1 = H_0 + D_2 + V_2 \tag{3.6}$$

with D_2 block diagonal, $\|D_2\|_\kappa, \|V_2\|_\kappa \leq 1$ (k-local bounded), with $U_1 \sim e^{[A_{S,\mathbf{s}},\cdot]}$ with $A_{S,\mathbf{s}} \sim \frac{\epsilon}{|\mathbf{s}|} V_{S,\mathbf{s}}$.

Technical point: not using LR bounds (which are not strong enough on expander graphs), instead use cluster expansion machinery (allows for bounding exponentially in volume rather than diameter).

Check soundness is importnat for ensuring the number of nonvanishing terms in the cluster expansion is constrained; more formally in:

$$[D_k, A_k] = \sum_{S,S'} [D_{k,S,\mathbf{s}}, A_{k,S',\mathbf{s}'}] \tag{3.7}$$

check soundness ensures that $S, \mathbf{s}'$ are not too far apart. A violates a syndrome near S, so $\|D_S\|, \|A_S\| \leq e^{-\kappa|S|}$ (exponentially small in volume).

Also, check soundness ensures that the gap does not close when rotating at each term in the PT:

$$\|(D - c_D I)|\psi\rangle\| \leq O(\epsilon^2) \|H_0|\psi\rangle\| \tag{3.8}$$

3.3 More comments on check soundness

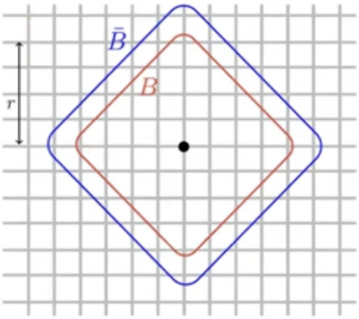
Some comments on the local check-soundness:

What people have considered is the notion of local topological order (1001.0344), which is a stronger version of the error correction condition that is sufficient for stability.

Definition: Local topological order

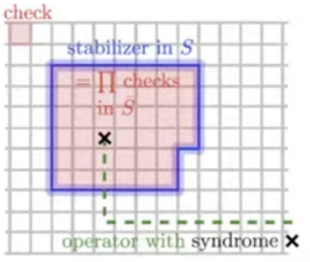
For any ball region B, let $\overline{B} = B \cup \mathrm{Neigh}(B)$. The LTO condition is then that for any ball with $r \ll L$ (with L, say, the code distance), the local ground states $P_{\overline{B}}$ are indistinguishable, with:

$$P_{\overline{B}} O_B P_{\overline{B}} = c P_{\overline{B}} \tag{3.9}$$



This is a stronger version of the error correction condition, as if we take $\overline{B}$ to be the whole system we recover the error correction condition. This rules out scenarios like the "Ising" TC because the $B_p = \pm 1$ subspaces are distinguishable locally. Note that for stabilizer codes that any stabilizer O_B is a product of checks in $\overline{B}$. The authors wanted to generalize this, but not straightforwards. Ball regions are not good, because we can only consider balls with $L \leq \log N$ due to expansion properties. But, we cannot control perturbations $V_1 V_2 \dots V_k$ at order $k \geq \log N$. So, we can only get an inverse polynomial energy splitting, rather than exponential.

Other suggestion - could we consdier general regions with bounded volume instead of diameter? No - LTO does not hold for such general regions, and we can even just consider the toric code:



so instead of trying to fill holes in "infinite dimensions" the authors considered check soundness as an algebraic construction.

Intuition for why check soundness might be good for stability/protect against some finely tuned counterexamples. Imagine we add a bunch of local stabilizers as a perturbation. Check soundness ensures that each is associated to a small number of checks. For a state to gain a lower energy E from violating stabilizers, it needs to violate many checks such that the energy penalty from H_0/the checks is $\gg E$, meaning the gap is stable.

Linear check soundness is equivalent to local testability[2] of the classical code:

$$|\mathbb{H}\mathbf{x}| \geq \alpha \min_{\text{codeword } \mathbf{z}} |\mathbf{x} - \mathbf{z}| \tag{3.10}$$

i.e. energy of a codeword being proportional to the minimal distance between code words. Morally, both are statements are connecting checks (low energy) to bits (low error).

4 Cool LDPC Physics

4.1 Violation of the 3rd law of thermodynamics

Finite rate qLDPC codes (by definition) have Hamiltonians for which the GSD increases exponentially in the system size n, and the stability result of the previous section tells us that this extensive GSD is in fact stable to arbitrary perturbations (c.f. SYK model which does have extensive 0-temperature entropy but is not robust). So we have entire phases of matter with a robus/non-zero entropy density:

$$\lim_{T \to 0} \lim_{n \to \infty} \frac{S(T)}{n} = \frac{k}{n} \log 2 \tag{4.1}$$

open Q: does the 3rd law require locality?

4.2 The NLTS theorem

Theorem: No Low-Energy Trivial States

There exist a family of k-local Hamiltonians such that every low-energy state cannot be generated by a constant depth $\Omega(1)$ circuit.

[2] Called this because if testing whether one is given a codeword or codeword with finite errors takes only local checks given this condition

It was shown in 2206.13228 that there are indeed NLTS Hamiltonians - precisely those corresponding to good qLDPC codes! Big step in complexity theory - related to the QPCP (quantum probablistic proof checking) conjecture (in the sense that QPCP $\implies$ NLTS), the classical analog which has been proven but the quantum proof which has evaded.

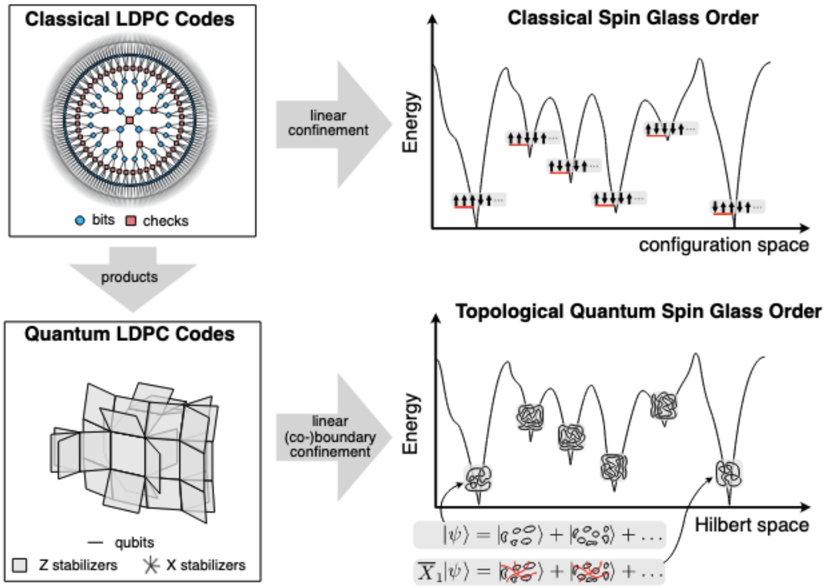
(Proof overview - constant rate and linear distance quantum codes have expansion properties of their check matrices, which implies their approximate code words are clustered. It can be shown for such codes that the classical distributions of the low-energy states of these Hamiltonians are clustered and also well-spread/not concentrated, which gives rise to the circuit lower bound.)

Physically, these are quite interesting because they are " a very strong form of TO" - associating constant/extensive circuit depth with whether a state is trivial/non-trivial, if we are below a certain T_c, NLTS implies *no* state can be prepared by a finite depth circuit. This is very different from the toric code - e.g. the 4D toric code is thermally stable but does not have this NLTS property; indeed the eigenstates below T_c are hard to prepare but we can do a sort of "tiling" argument where we prepare small patches of toric codes on tiles of a larger system. The energy penalty scales with the surface/volume ratio, which can be tuned by the size of the tiling, allowing us to create an arbitrarily low energy state (not an eigenstate) efficiently/in finite time. In the good qLDPC setting, because the ratio of surface area to volume is constant, this kind of construction is not possible.

Another way of phrasing - generally we expect that thermal states to be less complicated, as entanglement decreases when we turn up temperature - but NLTS gives a setting where this is manifestly not true, and all low energy states are still exotic/complex.

4.3 Spin Glasses and MBL

2412.13248 (have not looked into this paper yet)



LDPC codes with local testability have a very rich landscape:

- There are exponentially many (2^k) global mimina at zero energy (codewords), surrounded by linear energy barriers.
- There are exponentially many $\sim 2^k$ local minima at finite energy but zero energy density, also surrounded by linear energy barriers.

- There are exponetnally many $\sim 2^{k'}$ with $k' > k$ local minima at finite energy densities, again surrounded by linear energy barriers.

Classical LDPC realizes spin glass order (given the above energy landscape), with the minima able to store classical information. It turns out that the QLDPC analog is that the minima correspond to topologically ordered quantum states, which can store quantum info.

Note that it is because of this type of landscape that local decoding is so efficient! Takes the form of a bitflip Monte Carlo where we randomly flip local bits and accept if it lowers the energy.

This same feature has been used to give an example of many-body localization, a robust counterexample to the Eigenstate thermalization hypothesis (that $\rho_{\text{eigenstates}} \approx \rho_{\text{thermal}}$ for local probes). 2405.12279 uses these good classical LDPC codes with local testability to argue (via an Anderson localization type argument) to show the MBL property:

Theorem: MBL for LDPC codes

If $\epsilon < C_0$, then with high probability $1 - 2^{-N}$, any eigenstate $|\psi\rangle$ at low energy $E < C_0 N$ is localized in one well $z(\psi)$:

$$\left\| (1 - P_{z(\psi)}) |\psi\rangle \right\| \leq 2^{-N} \tag{4.2}$$

thus we have a robust ETH at low temperature! As a corollary, starting in one well, the state does not escape for any time t.

Proof (sketch).

1. A nice LDPC code exists with exponentially many (required for the localization to be an exponentially small subspace) deep wells. Known from classical coding theory.
2. At low energy, the extensive energy barriers suppress couplings (off-diagonals) among wells (diagonals):
   - $H = H_> + \sum_{\text{wellz}} H_z$ with $\left\| H_> |\text{low } E\rangle \right\| < \epsilon^N$
3. This energy difference is much smaller than the typical energy difference 2^{-N} from detuning. Thus, from a PT argument we can see that the $H_>$ will not mix energy eigenstates.

A Mike Vasmer KITP Talk Notes

Notes based on Michael Vasmer's 2025 KITP talk, found here.

A.1 Literature

- Review: 2103.06309
- Good QLDPC codes:
  - 2111.03654
  - 2202.13641
- Bicycle codes:
  - quant-ph/0304161
  - 1212.6703
  - 2308.07915
- Product codes:
  - 0903.0566
  - 1904.02703
  - 2012.09271
- Logic gates:
  - 2204.10812
  - 2407.18490
  - 2410.03628
  - 2506.03094

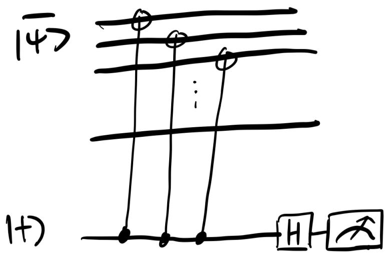
A.2 Defining LDPC Codes

Definition: LDPC Codes

A family of stabilizer QECCs $\{Q_i\}$ is LDPC if for every Q_i, there exists a generating set S_i such that:

1. $\forall s \in S_i$ then $\mathrm{Wt}(s) \leq r$ for some constant r (the stabilizers have bounded weight)
2. $\forall q \in [n_i]$, $|\{s \in S_i : q \in \mathrm{supp}(s)\}| \leq c$ for some constant c (qubit degree - each qubit should only be in the support of a constant number of checks).

Motivation: When we think about QEC, we have some encoded state $|\bar{\psi}\rangle$, and we measure some stabilizer - below $X_1X_2X_3$:



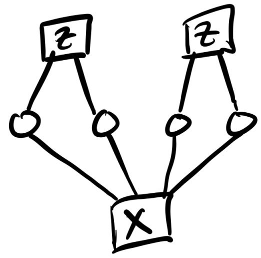
In the real world where each step of this is noisy (the gates), the measurement, etc. the error scales with the weight of the stabilizer. So, it is useful to have a bounded weight so then the noise does not scale with the code size. The same is true for the qubit degree, because if one qubit happened to be part of many checks, then this noise would propagate poorly.

Remark: Note this is not the only way we can do fault-tolerance. E.g. we can do concatenated codes.

Definition: CSS Codes

CSS codes are QECCS for which the stabilizers are exclusively Pauli-X or Z.

Such codes have convenient graphical form in terms of Tanner graphs. For example we can have $S = \langle XXXX, IIZZ, ZZII \rangle$. As a Tanner graph we can draw it as:



in this representation, the LDPC condition can be phrased as the Tanner graph being a constant degree graph, i.e. the degree of each vertex in the graph is bounded by an n-independent constant.

On the physics side of things, qLDPC codes correspond to k-local stabilizer Hamiltonians.

A.3 Families of qLDPC Codes

Recall the shorthand of $[[n, k, d]]$ with n physical, k logical, and d code distance. Asymptotically good codes are those for which $k = \Theta(n)$ and $d = \Theta(n)$. Good stabilizer codes were discovered in the ’90s, but whether there exists good LDPC codes was open for many years.

- Toric code with $[[n, O(1), \Theta(n)]]$n - ’97 by Kitaev. Best known for a while
- Hypergraph product codes with $[[n, \Theta(n), \Theta(\sqrt{n})]]$ - ’14. Already we’ve left the world of geometrically local things. The procedure to construct quantum codes involves inputting two linear codes, and for HGPs we need to put in two expander codes (so we already leave finite dimensionality). But the actual construction is quite simple, e.g. feeding in two classical repetition code gives the toric code.
- Pantaleev-Kalach codes with $[[n, \Theta(n), \Theta(n)]]$. Complicated algebraic/randomized construction, with the n growing extremely fast (PK_2 has $n \sim 10^6$ qubits already) so not practical. So we don’t think about them in fault tolerance settings.

- Quantum Tanner codes $[[n, \Theta(n), \Theta(n)]]$ - based on PK codes, but with more progress towards small constructions.

Let's think about some finite size families - we want to look at $\frac{kd^2}{n}$ as a metric. For Toric code, this is $O(1)$, and indeed for α-dimensional topological local codes we have the bound that $kd^\alpha \leq cn$ for a constant c. So a good figure of merit for "doing better" than the standard toric code.

- (2D) Hyperbolic codes - toric codes on a hyperbolic tiling. This yields $[[n, \Theta(n), \Theta(\log n)]]$. Maybe the log scaling looks bad, but for small code this is actually quite good. Q - Hyperbolic graphs are some kind of instance of expander graphs? so why the difference in the scaling? A - Some precise theorems about expanders must be hidden in the tanner graphs to get good scaling. But not guaranteed - Necessary but not sufficient. Additional comment - we can add more structure to the hyperbolic tiling and this can help us to get better parameters.
- Microsoft construction of "rotating" the self-correcting 4D quantum memory 2506.15130
- Product codes: HGP codes, lifted product codes, balanced product codes. Product codes are generally recipes for inputting types of linear codes, perhaps with some kind of symmetry constraint, then taking the product and factoring out the symmetry gives more efficient constructions. Precursor to PK codes.
- Bicycle codes: Proposed by McKay a while ago, developed by Kovlev/Priyadko, and then develoepd by IBM for bivariate bicyle code, and then trivariate tricycle codes and so on. Central idea: find check matrices $H_x = [A|B], H_z = [B^T|A^T]$ with $[A, B] = 0$. Then, $H_x H_z^T = AB + BA = 2AB = 0$. How do I choose A, B in the right way (good code parameters but also low check weights)? We can use polynomial formalism, machine search over different choices. The one IBM talks about is $[[144, 12, 12]]$ "Gross codes[3]" with $\frac{kd^2}{n} = 12$. Stabilizer code with max weight-6 stabilizers, toric code with some extra long-range connections. Also is translation invariant. IBM showed that even though the stabilizer weight is higher, the threshold is comparable with the toric code $p \sim 0.5\%$ under a circuit noise model, so is a much better memory. (Note - not a threshold in the true sense, more like a pseudothreshold/breakeven point).

A.4 Logical Operations on LDPC Codes

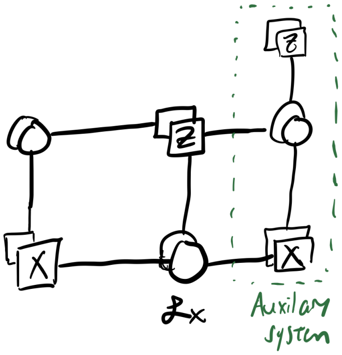
Ok, so we've gone over what qLDPC codes are and some code families people are interested in. How do we do logical operations on these codes? For CSS codes, we have transversal constructions, e.g. a logical `CNOT` is done by doing a physical `CNOT` between every pair of qubits in the code block. But let's look at constructions in some detail.

- Gate teleportation (Gottesman in the 90s) - used to prove that constant space overhead FT was possible. In practice might be useful for concatenated codes, but not for qLDPC.
- Many constructions which generalize gates for topological codes. (start with toric code gate, then generalize to HGP which usually works, and so on).
  - Defects + braiding (Krishna and Poulin)
  - Folding (fold a TC to get a color code, similar logic applies to HGPs - gives us transversal gates and permutations)
  - Dane twists (Cut a torus, twist, and glue back together)

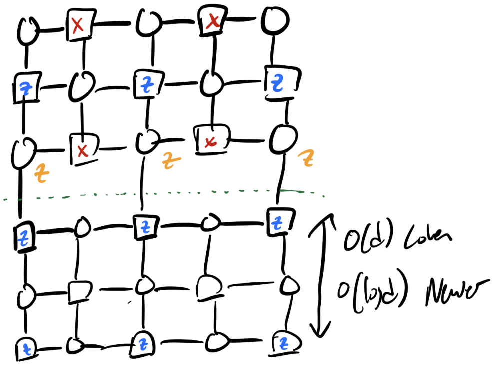
[3] because a dozen dozens is a gross.

- Generalized lattice surgery: If we can do Pauli product measurements, we can generate the Clifford group. This is useful because we can break down any QC into Clifford + Ts, which propagating all Cliffords to the end becomes T/magic states and Pauli product measurements at the end. Since magic state distillation is Clifford, so long as we just have some input magic states, then everything else is Pauli product measurements.

  Suppose we want to measure logical $\overline{X}$ (which will be a high weight operator) - this is noisy. If we break the measurement into smaller pieces (measuring multiple times to combat measurement errors), we can combine the measurement outcomes to get the $\overline{X}$ measurement. The cost of measuring a weight-d logical oeprator: Cohen proposal requires $O(d^2)$ space, $O(d)$ time cost. Improvements by Burton, Xanadu etc. use $O(d \log d)$ space and $O(d)$ time cost. Let's see this at a high-level using a Tanner graph, wherein we can introduce an auxillary system to measure the logical:



  In the toric code, this looks like (for measuring $\overline{Z} = \prod_{i \in \text{string}} Z_i$, with naively d layers to get a fault tolerant measurement):



  To get a sense for numbers, with IBM's [[144, 12, 12]] Gross code they require ∼90 extra qubits to measure all the operators they need.

  Q - What is this in the physics picture? Gauging/anyon condensation picture? Someone raised 2212.00042 as a possible reference, where lattice surgery is described as anyon condensation. Mike also brings up Dominic's work in gauging local operators 2410.02213 which may offer another connection.

  A last comment - the $O(d)$ time overhead doesn't sound too bad, but in IBM's new paper they show that on average they require 18 rounds of measurements of generating Pauli products. Then the overhead is $\sim 18d \ldots$ so although they save space over toric code the time overhead has had to been blown up as a result.

B Vedika Khemani IAS Talks Notes

Notes taken from Vedika Khemani's 3-part lecture series about qLDPC physics at IAS, found here. Based on 2310.16032, 2402.16831.

B.1 Introduction

There is a deep connection between error correcting codes and ordered phases of matters. In particular, stabilizer error correcting codes can be associated with commuting projector Hamiltonians, which are zero-correlation length fixed points of non-trivial phases. The intuition is that when looking at an error correcting code, we want to encode information in a robust and redundant way. One way to do this is to encode information into macroscopically distinct degenerate ground states. Such ground states can arise due to spontaneous symmetry breaking (associated to classical codes) or to topological order (associated with quantum codes) - note that the modern perspective actually unites these two in the sense that topological order is the SSB of higher-form symmetries.

Some examples:

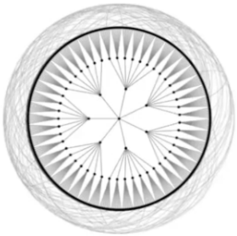
- Kitaev's 2D toric code $\leftrightarrow$ topological order. This TO makes the system locally indistinguishable, and goes hand in hand with the fact that the code is robust to local noise.
- 3D Haah code $\leftrightarrow$ fracton order.
- 2D repetition code $\leftrightarrow$ symmetry breaking order. Classical information can be encoded in the symmetry-breaking ground states of the 2D Ising model, with the ZZ interaction providing the error correction.

It turns out there is a deep relationship between classical and quantum codes - the 2D toric code which is a particular point in a $\mathbb{Z}_2$ gauge theory can be obtained by starting with a 2D Ising model and gauging it.

EC has seen recent breakthroughs - good ($k = \Theta(n), d = \Theta(n)$) qLDPC codes have been constructed (PK codes 2111.03654, Quantum Tanner codes 2202.13641). Also, good locally testable classical LDPC codes have also been constructed (2111.03654, 2203.03581, 2206.07750). Both of these constructions require codes defined on non-Euclidean expander graphs. This is in contrast to the previous examples given, which were all codes given on local, Euclidean lattice models. But these new sets of codes are defined on more exotic geometries - locally tree like with closed boundaries, where the boundary is proportional to the volume. Formally, this is captured by:

$$|\text{Neigh}(S)| \geq \nu |S| \tag{B.1}$$

so long as $|S| \leq n/2$.



This seems perhaps a bit abstract, but indeed such geometries are accessible in new synthetic quantum systems, such as neutral atoms.

The key messages to takeaway from the series:

- LDPC codes furnish an interesting class of stat mech models for many-body physics, leading to novel phenomena in non-Euclidean geometries

- We can understand qLDPC codes as generalized gauge theories. We can formulate symmetry-breaking order, topological orders, and SPT order on general non-Euclidean graphs, and formulate generalized Kramers-Wannier dualities relating these phases.
- We can unpack product constructions in the coding literature in physical terms, such as symmetries, redundancies, and excitations. We will find a unified description of codes via homology.
- LDPC codes can furnish new analytically tractable classical and quantum spin glasses.

Talk outline:

- Classical LDPC codes on expander graphs.
  - What is an LDPC code, and what is a "good code"? Why are expanders necessary?
- Quantum LDPC codes as "products" of classical codes.
  - Unified homological description of quantum codes in terms of chain-complexes. Constrast physical dimension from chain-complex dimension.
  - Gauging and KW dualities
- Time permitting, a discussion of LDPC codes as spin glasses. (did not end up getting here)

B.2 Linear Classical Codes

B.2.1 Parity Checks and LDPC

We begin by reviewing classical coding theory. The goal is to transmit k message, or logical bits. We encode this into $n > k$ physical bits (some form of redundant encoding).

As an example - suppose we want to transmit a single bit $k = 1$. An error channel will come in and try to flip our physical bits with probability p - if the information is unencoded, then the information is lost with probability p. We can then consider the $n = 5$ code where we repeat the input bit 7 times (classical repetition code). Then, so long as the error channel does not flip more than 3 bits, our encoded bit is safe (we can take a majority vote at the end). The codewords are 0000000 and 1111111.

We also introduce the notion of code distance d, which is the Hamming distance between codewords. Here, the code distance is $d = n = 7$. So long as the errors corrupt $< \frac{d}{2}$ bits, then I can distinguish codewords. What we would want is a family of codes such that as we scale up, the probability of successful transmission goes to 1.

More generally, we have n physical bits $x_i = 0/1$. The configuration space is is a vector space $\mathbb{F}_2^n$ with $\mathbb{F}_2 = \{0, 1\}$. The configurations are n-bit vectors in the space, e.g. $(0, 0, 1, 1, 0, 1)^T$. If we write $|i\rangle$ to be the basis vector for the ith bit, a general configuration can be written as $\sum_i x_i |i\rangle$ with $x_i \in \{0, 1\}$. We can consider spins $\sigma_i^z = Z_i = \pm 1$ based on $x_i = \pm 1$ (for now the spins are classical, but we use this suggestive notation as we will soon promote these spins to quantum bits). Now, in order to encode information, we consider a codespace $\mathcal{C}$ which is a subspace (itself a linear vector space) of $\mathbb{F}_2^n$. $\mathcal{C}$ will be composed of 2^k codewords. How do we specify what these codewords are? We can specify them by imposing constraints, known as parity checks (PC). These are encoded in a parity-check matrix $\mathbb{H} \in \mathbb{F}_2^{m \times n}$ of m parity checks. We can then define the codespace as the bitstrings that satisfy all parity checks:

$$\mathcal{C} = \left\{ \mathbf{x} \in \mathbb{F}_2^n : \mathbb{H}\mathbf{x} = 0 \right\} \tag{B.2}$$

As an example, we can consider the parity check matrix:

$$\mathbb{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{B.3}$$

so the codewords $(x_1 x_2 x_3 x_4 x_5 x_6 x_7)^T$ must obey:

$$x_1 \oplus x_2 \oplus x_4 \oplus x_5 = 0 \tag{B.4}$$

$$x_1 \oplus x_3 \oplus x_4 \oplus x_6 = 0 \tag{B.5}$$

$$x_2 \oplus x_3 \oplus x_4 \oplus x_7 = 0 \tag{B.6}$$

where $\oplus$ denotes mod 2 addition.

A couple observations:

- $\mathbf{x}$ is always a codeword.
- $\ker(\mathbb{H})$ gives a basis for codewords.
- Linearity implies that if $\mathbf{w}, \mathbf{v} \in \mathcal{C}$, then $\mathbf{w} + \mathbf{v} \in \mathcal{C}$.
- The number of logical bits k is specified by:

  $$k = n - \text{rank}(\mathbb{H}) \tag{B.7}$$

  with $\text{rank}(\mathbb{H})$ being the number of linear constraints. If you have a CS background, this structure may remind you of satisfiability problems, and indeed they are instances of XOR satisfiability problems.
- The code distance d is determined as the minimum Hamming weight of a non-zero codeword.
- We can think of each of the constraints appearing here as a term in a classical Hamiltonian - for each row of $\mathbb{H}$ we get a term, with:

  $$C_a = \prod_{j:\mathbb{H}_{aj}=1} Z_j \tag{B.8}$$

  with $a = 1, \ldots m$, and the full Hamiltonian is the sum:

  $$H = -\sum_a C_a \tag{B.9}$$

  the observation that $\mathbf{x} = 0^n$ is always a codeword is analogous to the observation that the all-up state is always a ground state in the Hamiltonian language.

With this mapping, we can understand the code parameters $[n, k, d]$ in more physical terms. n is the number of physical bits, or the system size. k is the number of logical bits, or $\log_2(\text{GSD})$. d is the code distance, or the number of flips between ground states.

Definition: LDPC Codes

LDPC codes are low-density parity check codes. These are linear codes for which $\mathbb{H}$ is sparse, which implies:

- The number of bits each check acts on is $O(1)$.
- Each bit is acted on by $O(1)$ checks.

The LDPC condition is stronger than k-locality, which corresponds to the first condition (where each interaction term in the Hamiltonian acts on $k = O(1)$ bits) - here we also require that the bit degree is also bounded.

B.2.2 The Repetition Code, Logical Operators, The Hamming Code

Let us consider a few examples. The first (which we have already alluded to) is the repetition code, or Ising model:

$$H = -\sum_{\langle ij \rangle} Z_i Z_j \tag{B.10}$$

If we have a 1D chain with periodic boundary conditions, we get:

$$\prod_{\langle ij \rangle} C_{ij} = \prod_{\langle ij \rangle} Z_i Z_j = \prod_i Z_i^2 = \mathbb{I} \tag{B.11}$$

In this case, we have $m = n - 1$ independent constraints (because we have n terms, but one of them can be obtained as a product of all other $n - 1$ terms), and so:

$$k = n - (n - 1) = 1 \tag{B.12}$$

So we get two codewords/groundstates, corresponding to $\uparrow^n$ and $\downarrow^n$. The code distance is then $d = n$. The parity check matrix of this code looks like:

$$\mathbb{H} = \begin{pmatrix} 1 & 1 & 0 & 0 & \dots \\ 0 & 1 & 1 & 0 & \dots \\ \vdots & & & & \end{pmatrix} \tag{B.13}$$

Now, we want to introduce the notion of $\uparrow^n$,$\downarrow^n$ as the logical state, and a logical $\overline{Z}$ that denotes the logical bit as being 0 (up)/1 (down). Generally, we want to do operations on the logically encoded information. The logical $\overline{X}$ that flips the logical bit is just the bitflip on all physical spins:

$$\overline{X}_L = \prod_{i=1}^{n} X_i \tag{B.14}$$

This is nothing more than that $\mathbb{Z}_2$ Ising symmetry! So - SSB here yields gives rise to degenerate ($k \geq 1$) and macroscopically distinct ($d = \Theta(n)$) codewords.

If we label the basis states for the codewords as:

$$|\chi_\lambda\rangle = \sum_{i \in \chi_\lambda} |i\rangle \tag{B.15}$$

with $\lambda = 1, \dots, k$, each defines a logical $\overline{X}_\lambda$ operator:

$$\overline{X}_\lambda = \prod_{i \in \chi_\lambda} X_i \tag{B.16}$$

which takes us form the $|\mathbf{0}\rangle$ codeword to the $|\chi_\lambda\rangle$ codeword.

Additionally, for each of these vectors we can introduce a dual vector $\langle \rho_\lambda |$ where:

$$\langle \zeta_\lambda | \chi_{\lambda'} \rangle = \delta_{\lambda\lambda'}. \tag{B.17}$$

We can also consider a logical $\overline{Z}_\lambda$ operator which will tell us which codeword state we are in + have the correct anticommutation relations with the $\overline{X}_\lambda$ operators:

$$\overline{Z}_\lambda = \prod_{i \in \zeta_\lambda} Z_i \tag{B.18}$$

where we can see that $\left\{\overline{X}_\lambda, \overline{Z}_\lambda\right\} = 0$. Now, $\left\langle \overline{Z}_\lambda \right\rangle$ is an order parameter for the SSB, and in particular for each k we will find an order parameter - each codeword defines a $\overline{X}_\lambda$, which is a logical and does not excite any of the checks. Therein they are symmetries of the Hamiltonian, giving rise to a total symmetry group of $G = \mathbb{Z}_2^k$. The codespace is SSB, with a 2^k-fold degeneracy/symmetries being spontaneously broken.

For the classical codes, there is a basis in which $\overline{Z}_\lambda$ can be chosen to be act on a single site i_λ. By the time we finish consistently picking the $\overline{Z}$s, we are left with k single-site bits we can view as our logical bits, which is called putting our code in canonical form. We can think of the first k bits of the code as labelling the logical space, while all the other $n - k$ bits are padding/redundancy.

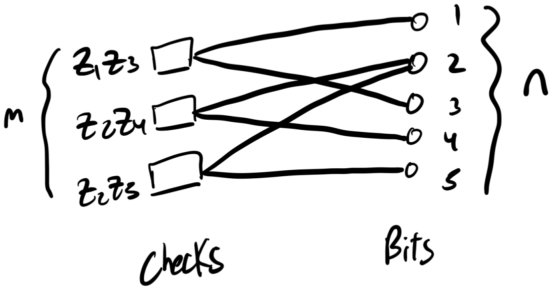
Note that we can either start with the parity check matrix $\mathbb{H}$ or the generator matrix $\mathbb{G}$, and then there will be properties of the parity checks or codewords which will determine whether a local basis can be found for the PCs or not. When looking at LDPC codes, we like to start with the parity check matrix $\mathbb{H}$ because we would like these to be local (in the generalized locality sense).

We can also think about a vector space of checks:

$$\mathbb{F}_2^m = \sum_a \beta_a |a\rangle \tag{B.19}$$

with $\beta_a \in \{0, 1\}$. We can then think of $(\mathbb{H}^T)^{n\times a}$ mapping between checks $|a\rangle$ to collections of bits $\sum_{i\in\mathbb{H}_a} |i\rangle$. In the spin language, if we have a set of interactions, multiplying out all the interactions, which sets of spins are left?

This prescription is useful, because it allows us to consider a Tanner graph representation of code. A Tanner graph is a bipartite graph where on one side we have nodes corresponding to (n) bits, and on the other side we have nodes corresponding to (m) checks. In this formalism, the adjacency matrix of the Tanner graph is exactly the parity check matrix $\mathbb{H}$. In the coding literature, random constructions are quite common - in this setting these constructions are just coming up with some random adjacency/parity check matrix. The LDPC condition is that the adjacency matrix is sparse.



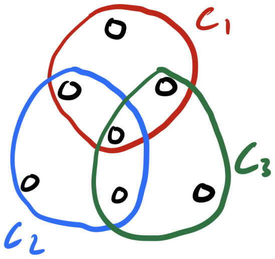
We can consider the [7, 4, 3] Hamming code, which has $2^4 = 16$ codewords. We have already written down the parity check matrix:

$$\mathbb{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{B.20}$$

The corresponding Hamiltonian is that of a generalized Ising model:

$$H = -Z_1Z_2Z_4Z_5 - Z_1Z_3Z_4Z_6 - Z_2Z_3Z_4Z_7 \tag{B.21}$$

We can also consider a hypergraph representation, where edges can act on more than 2 bits at a time. For the Hamming code, this looks like:



B.2.3 Good Codes

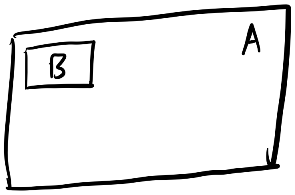
A "good" code is that for which $k, d = \Theta(n)$. This is as good as things can get! This scaling is in fact impossible to achieve in finite D local Euclidean models (checks are on a Euclidean geometry). The (classical) Brayvi-Poulin-Terhal bound tells you that for classical codes in D-dimensions:

$$kd^{1/D} \leq cn \tag{B.22}$$

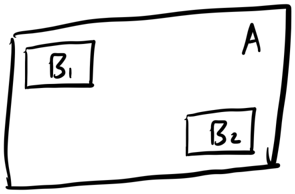
note that indeed the 1D Ising model is $k = 1$ and $d = n$ saturates the bound... this is quite bad. The distance is good but the rate is terrible (asymptotically zero). The associated quantum bound is:

$$kd^{\frac{2}{D-1}} \leq cn \tag{B.23}$$

Proof (sketch). Consider a 2D Euclidean space (can be D-dimensional, but easier to draw):



Then, $|B| = l^D \sim d < d$. If two ground states agree on A, they must agree everywhere. Now, consider two B boxes:



Since the checks are geometrically local, they overlap with at most one B_i block. Even though the size of the two Bs together may be larger than d, since they do not overlap, it is still the case that if the ground states agree on $A \setminus B_1 \setminus B_2$ that they agree everywhere. Now, we can tile the entire system with B blocks. We then observe that:

$$\frac{k}{n} \leq \frac{|A|}{n} \tag{B.24}$$

because we can only choose things independently in A to specify the codewords. But the size of A here is $\frac{|\partial B_i|}{|B|_i}$, and thus:

$$\frac{k}{n} \leq \frac{|A|}{n} \sim \frac{|\partial B_i|}{|B|_i} \sim \frac{1}{l} \sim \frac{1}{d^{1/D}} \tag{B.25}$$

The key figure is the surface-to-volume ratio when we construct the codespace into a division/tiling of smaller blocks. In Euclidean space, the surface scales not proportionally to the volume, but on expander graphs the surface-to-volume ratio goes to one, which ensures we can circumvent bounds that are based on geometric locality of checks (we will replace it with a notion of graph locality). Indeed we can consider expander/hyperbolic graphs where the local structure looks like the local Ising models we have considered today - we will elaborate on these when we discuss Tanner codes in the next section.

B.2.4 Classical expander codes

Definition: Expander codes

A classical linear code is (μ, ν) expanding if $|\mathbf{x}| < \mu n \implies |\mathbb{H}\mathbf{x}| > \nu|\mathbf{x}|$.

In the definition above, $\mathbf{x} \in \mathbb{F}_2^n$ is a configuration and $\mathbf{s} = \mathbb{H}\mathbf{x}$ is the syndrome/vector of violated checks, and $|\cdot|$ denotes the Hamming weight. Expansion ensures that the code distance is linear, with $d \geq \mu n$.

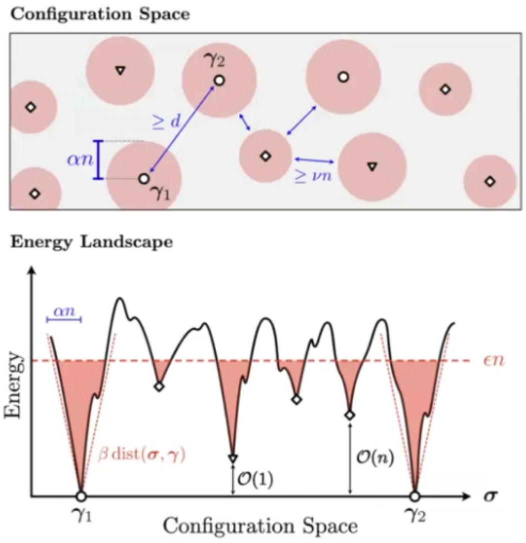
What does this condition say physically? If we start with a codeword, the energy barrier from one codeword to another proportional to the number of spins that we flip, until a finite fraction of spins that have been flipped. This is a very robust energy barrier.

Compare this to the 1D Ising model - if we start with $\uparrow^n$ and flip one spin, we trigger two checks/costs energy two. If we keep flipping (neighbouring) spins, the cost is still energy two (only proportional to the size of the domain wall, which stays a constant size), until we get to the all one state. This is much tamer than the energy barrier that expander codes have.

The energy landscape of expander codes is provably complex - this and connection to spin glass properties are elaborated on in 2412.13248:

- There are exponentially many (2^k) global mimina at zero energy (codewords), surrounded by linear energy barriers.
- There are exponentially many $\sim 2^k$ local minima at finite energy but zero energy density, also surrounded by linear energy barriers.
- There are exponetnally many $\sim 2^{k'}$ with $k' > k$ local minima at finite energy densities, again surrounded by linear energy barriers.

Note that it is because of this type of landscape that local decoding is so efficient! Takes the form of a bitflip Monte Carlo where we randomly flip local bits and accept if it lowers the energy.



B.2.5 Constructions of Classical Expanders

Construction 1 (Gallager, 1960s). We build a regular (c, b) regular random bipartite graph as the Tanner graph. b is the bit degree and c is the check degree (apart from fixing b, c the graph is random). The number of edges in the graph is:

$$N_e = nb = mc \tag{B.26}$$

The rate of the code is:

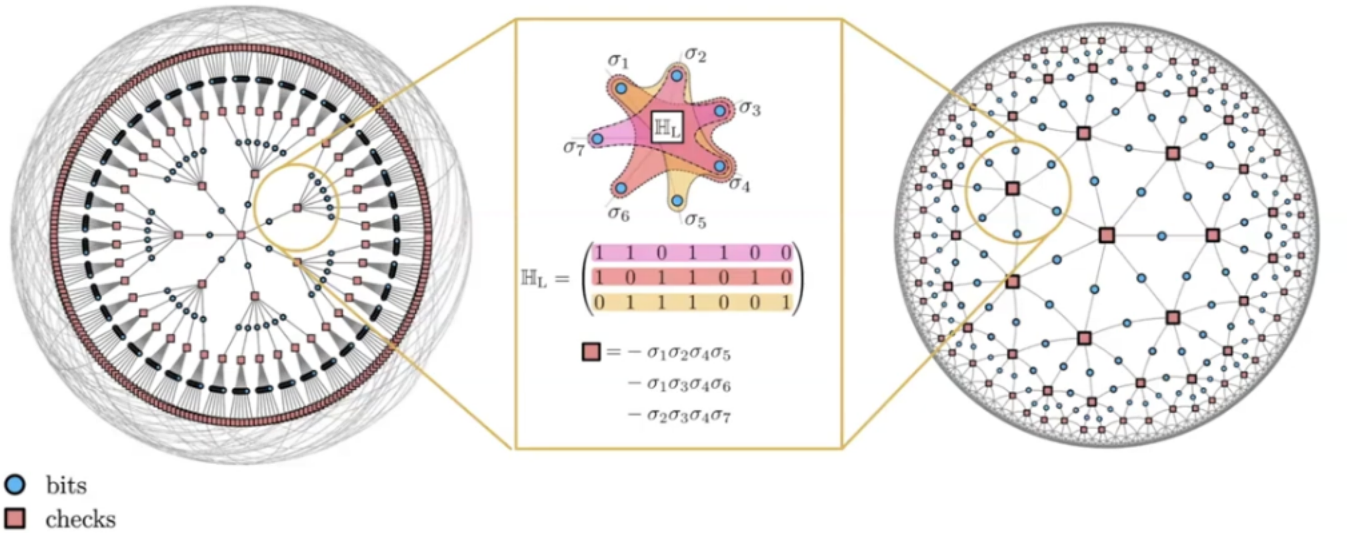
$$\frac{k}{n} \geq \frac{n-m}{n} = 1 - \frac{b}{c} \tag{B.27}$$

the inequality follows from the fact that $k = n - m$ if all constraints m are independent. We can show that with high probability, $\frac{k}{n} \to 1 - \frac{b}{c}$, and $d \propto n$.

Construction 2 (Sipser-Spielman). Consider Tanner codes $\mathcal{T}(G, C_0)$ (note - different from Tanner graphs). These have two ingridients:

- A sufficiently expandering underlying graph of fixed degree s
- A sufficiently good "local code" C_0 with $(n_L = s, k_L, d_L)$ on every vertex.

Examples are taking the Cayley tree/hyperbolic tesselation and associating the Hamming code to the vertices:


σ_1
σ_2
σ_3
σ_4
σ_5
σ_6
σ_7
$\mathbb{H}_L$
$\mathbb{H}_L = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$
$= -\sigma_1\sigma_2\sigma_4\sigma_5$
$-\sigma_1\sigma_3\sigma_4\sigma_6$
$-\sigma_2\sigma_3\sigma_4\sigma_7$
bits
checks

So long as $r_L > \frac{1}{2}$ (rate of local code is high enough) and $d_L \geq \lambda_2$ (rate of local code is high enough, with λ_2 capturing a rate of expansion), then:

$$[n, k, d] = [n, \Theta(n), \Theta(n)] \tag{B.28}$$

Caveat - the model that we've drawn above with the Hamming codes are an illustration, and the actual proofs for good LDPC codes require much larger codes. But, these models do reproduce a lot of phenomenology that is only provable for larger codes.

How do we pick the underlying graphs? We consider graphs with expansion properties quantified by the Cheeger constant. Given a graph $G(V, E)$, we define the Cheeger constant as:

$$h(G) = \min\left\{\frac{|\partial S|}{|S|} : S \subset V, |S| < \frac{n}{2}\right\} \tag{B.29}$$

where $\partial S \in E$ are the set of edges connecting S to its complement. We can also consider the spectral expansion: eigenvalues of the adjacency matrix:

$$s = \lambda_1 \geq \lambda_2 \geq \lambda_3 \ldots \tag{B.30}$$

A large spectral gap is characterized by a better expansion:

$$\lambda_1 - \lambda_2 \leq 2h(G) \tag{B.31}$$

for Ramanujan graphs:

$$\lambda_2 \leq 2\sqrt{s-1} \tag{B.32}$$

and these are graphs with the best possible spectral expansion properties. We can construct these with:

- Random regular graphs
- Cayley graphs of particular groups (PSL$(2, \mathbb{Z}/q\mathbb{Z})$)
- Tilings of the hyperbolic plane which are $\{p, q\}$ regular

Given these sufficiently expanding graphs (quantified by $\lambda_2, h(G)$) and the sufficiently good local code, we have:

$$r = \frac{k}{n} \geq 2r_L - 1 \tag{B.33}$$

$$\frac{d}{n} \geq \frac{d_L - \lambda_2}{s - \lambda_2}\frac{d_L}{s} \approx \left(\frac{d_L}{s}\right)^2 \tag{B.34}$$

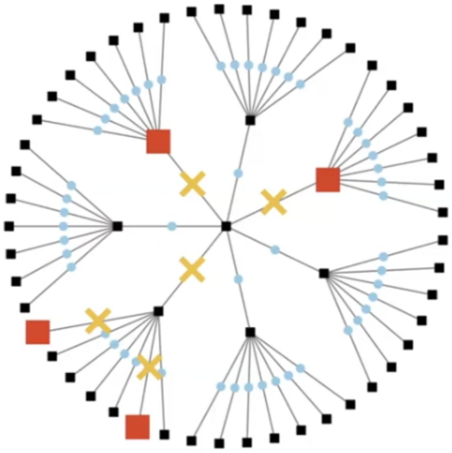
Thus we obtain the desired conditions:

$$r_L > \frac{1}{2}, d_L \geq \lambda_2 \tag{B.35}$$

and this gives us the "good" code properties of $[n, k, d] = [n, \Theta(n), \Theta(n)]$.

Let us give an intuitive picture: How do we go from one codeword to another?

- We start from an all-zero state and start flipping bits. As we continue to flip bits, we get that the internal checks are happy, but then the next layer outward of checks become unhappy. We continue to flip bits, pushing the unhappiness towards the boundary.
- Large local code distance $\implies$ many bits around each vertex must be flipped $\implies$ the global code distance (distance between two global codewords) is large.
- Large local code rate $\implies$ many possibilities to satisfy local codes $\implies$ global code also has many possibilities to satisfy, hence good global rate.



So - by going from Euclidean to non-Euclidean graphs, we were able to get from the $k = 1, d = n$ Ising/repetition code (which saturated the BPT bound) and find classical codes which have $k, d = \Theta(n)$. These codes have been known for many decades, and are used for telecommunications!

B.3 Quantum codes

Now, we want to move onto qLDPC codes. We start with a discussion of quantum codes generally, and we will generally stay within the realm of CSS Stabilizer codes (codes which are described by Pauli stabilizers, where the Paulis do not mix X and Z type).

We consider n physical qubits with Paulis X_i, Y_i, Z_i for $i \in \{1, \ldots, n\}$. The code subspace is the +1 eigenspace of commuting X, Z parity checks:

$$A_1 = X_1 X_2 X_3 \ldots, B_1 = Z_1 Z_2 Z_3 \ldots \tag{B.36}$$

The quantum code is made up of $\mathcal{C}_q = \{\mathcal{C}_x, \mathcal{C}_z\} = \left\{\{A_v\}_v, \left\{B_p\right\}_p\right\}$, where the codewords are the 2^k dimensional code-subspace that satisfy/are the $+1$ eigenvalues of all the X and Z checks. Equivlaently, we can phrase the codewords as the degenerate ground states of the commuting stabilizer Hamiltonian:

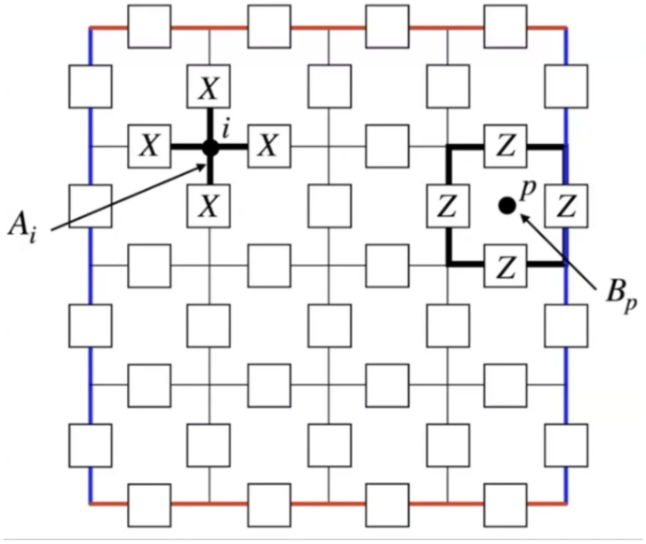
$$H_q = -\sum_v A_v - \sum_p B_p \tag{B.37}$$

with $[A_v, B_p] = 0$.

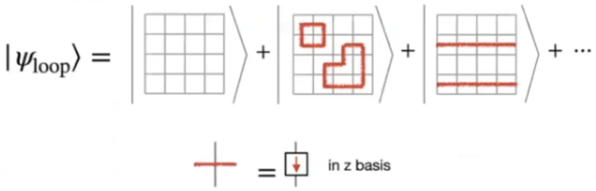
B.3.1 The toric code

The canonical toric code (Kitaev '97) is defined on a square lattice, with n physical qubits living on the edges, and toroidal/periodic boundary conditions. The Hamiltonian is given by the sum of the star and plaquette terms:

$$H_{\mathrm{TC}} = -\sum_v A_v - \sum_p B_p \tag{B.38}$$



The ground state of the toric code is the equal-weight superposition of loops on the dual lattice:

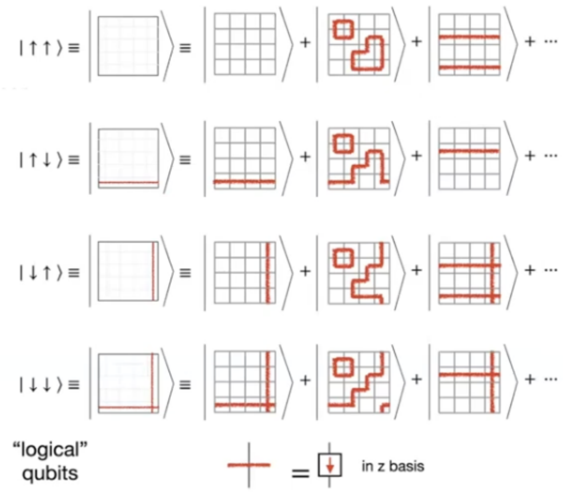


This is because:

$$A_v|\psi_{\text{loop}}\rangle = B_p|\psi_{\text{loop}}\rangle = |\psi_{\text{loop}}\rangle \tag{B.39}$$

for all i, p. This can be seen from the fact that B_p leaves loop configurations invariant (either does not intersect any loops, or touches the loops an even number of times, so either way the product of Zs gives 1) and A_i simply permutes loop configurations on the dual lattice.

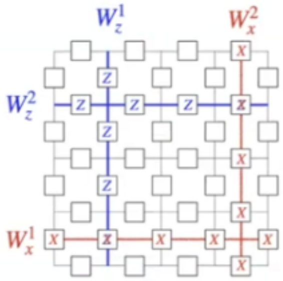
The four degenerate ground states of the toric code Hamiltonian are labelled by whether a non-contractible loop goes around the torus or not. These loops are Wilson loops that commute with all checks but are not generated by any of the checks. Thus we get $k = 2$ logical qubits.



The Wilson loops (1-form symmetries) act as the logical operators for the code:

$$\bar{Z}_1, \bar{Z}_2 = W_z^1, W_z^2 \tag{B.40}$$

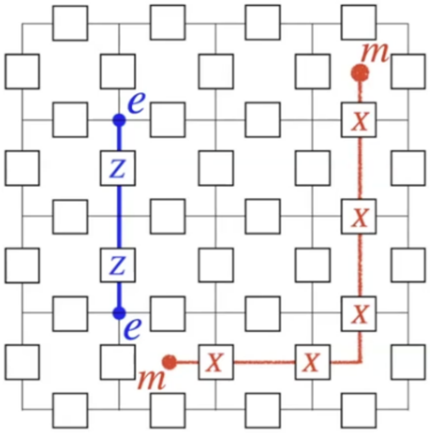
$$\bar{X}_1, \bar{X}_2 = W_x^1, W_x^2 \tag{B.41}$$



You can verify that they follow the appropriate commutation relations as logical Pauli operators, and the $W_x^{1/2}$ operators correctly transform between the different ground states.

Note that the 4 degenerate ground states are macroscopically distinct (because they are distinguished by the macroscopic Wilson loop operator which diverges in the system size), and protect against $d \sim \sqrt{n}$ errors of both X and Z type (bit and phase flips). The ground states are locally indistinguishable, corresponding to the topological order that the states possess.

Note that topological order appears naturally as the deconfined phase of a (discrete) gauge theory, $\mathbb{Z}_2$ in the case of the toric code. We have deconfined charge (e) and flux (m) excitations (pairs of which can be generated by Z or X strings), with non-trivial (anyonic) statistics.



To summarize where we are - we showed that in the classical case, we had the correspondence of SSB with a classical repetition code. We considered the Ising Hamiltonian:

$$H = -\sum_a C_a = -\sum_{\langle ij \rangle} Z_i Z_j \tag{B.42}$$

the 2 degenerate ground states being the 2 codewords of the classical code. We had local bits $\overline{\uparrow} = \uparrow^n$ and $\overline{\downarrow} = \downarrow^n$. The logical $\overline{X} = \prod_i X_i$ corresponded to the Ising symmetry. This was a $k = 1, d = n$ code that saturated the classical BPT bound.

In the quantum case, we have a correspindence of topological order with a quantum code. We considered the toric code Hamiltonian:

$$H_{\mathrm{TC}} = -\sum_v A_v - \sum_p B_p \tag{B.43}$$

the 4 degenerate ground states being the codewords of the code. We encoded 2 logical qubits, distinguished by extensive Wilson loop operators. Such Wilson loop operators also were identified with the logical operators of the code, with:

$$\bar{Z}_1, \bar{Z}_2 = W_z^1, W_z^2 \tag{B.44}$$

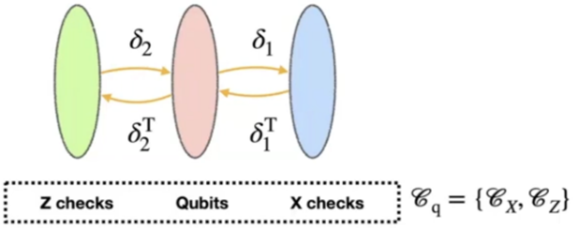
$$\bar{X}_1, \bar{X}_2 = W_x^1, W_x^2. \tag{B.45}$$

This was a $k = 2, d \sim \sqrt{n}$ bound that saturated the quantum BPT bound.

B.3.2 quantum LDPC Codes

Good classical LDPC codes were discovered in the 1960s, while good qLDPC codes were not found until quite recently (2020). What took so long/what was the challenge? Let us recall that a quantum CSS code comprises of two classical codes:

$$H_q = \underbrace{-\sum_v A_v}_{\mathbb{H}_x} \underbrace{-\sum_p B_p}_{\mathbb{H}_z} = -\sum_v \prod_{a \in \delta_1^T(v)} X_a - \sum_p \prod_{a \in \delta_2(p)} Z_a \tag{B.46}$$



One might intuitively think that making a good qLDPC codes would be as simple as taking two good classical LDPC codes and picking them to be $\mathbb{H}_x/\mathbb{H}_z$. The difficulty is that we can't naively pick X/Z checks to come from good classical LDPC codes and then have all of those checks mutually commute. Specifically, a good $\mathcal{C}_z$ will *not* have small logical $\overline{X}$ operators, and hence the $\mathcal{C}_x$ code cannot be LDPC.

We then need some additional structure - it is already present in the figure above. We need $[A_v, B_p] = 0$ and so $\delta_2\delta_1 = 0$; this corresponds to a 2-dimensional chain complex.

Thus, the name of the game is to start with classical code (a 1-dimensional chain complex) and then compose them in such a way that we get a higher dimensional algebraic structure corresponding to a good qLDPC code.

If we look at 2111.03654, 2202.13641, they discuss lifted product constructions, and other ways of "multiplying" two classical codes in such a way to get good qLDPC. The picture in our mind is that we are building up a higher-dimensional (abstract) manifold by taking products of two lower-dimensional manifolds.

B.3.3 Introduction to homology theory

Homology is a very useful language which provides a unified description of *all* CSS stbailizer codes, including toric codes and fracton models in any dimension or manifold (whether Euclidean or not). The code defines a notion of geometry through the chain complex. Note that this geometry may be different from the geometry of the underlying lattice.

Dan Browne has a nice set of lecture notes on Homology theory which can be found here.

Definition: Chain complexes

A chain complex is a sequence of vector spaces with linear "boundary maps" δ in between:

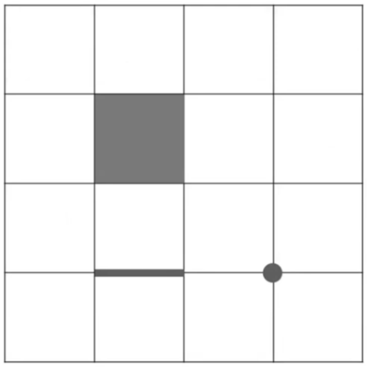
$$V_{D_c} \to \dots V_p \overset{\delta_p}{\to} V_{p-1} \dots V_2 \overset{\delta_2}{\to} V_1 \overset{\delta_1}{\to} V_0 \tag{B.47}$$

with D_c the dimension of the overall chain complex. The maps obey the composition condition:

$$\delta_p \delta_{p+1} = 0 \tag{B.48}$$

Note that the homology theory we discuss will be the combination of topology (cellulation of a manifold) + group theory (for us, $\mathbb{F}_2 = \{0, 1\}$ since we only worry about qubits).

We consider a cellulation ofa 2-dimensional manifold. The 2-cells are plaquttes, the 1-cells are edges, and the 0-cells are vertices.

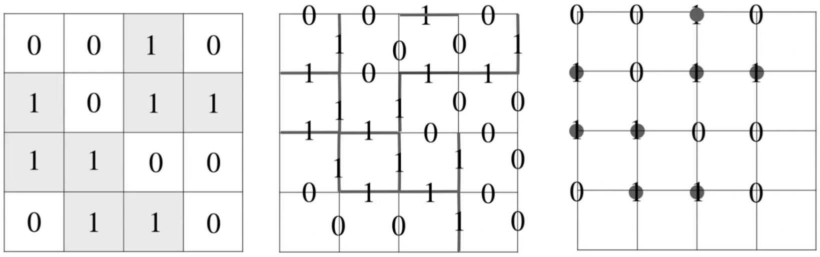


From this, we can define an n-chain:

Definition: n-chain

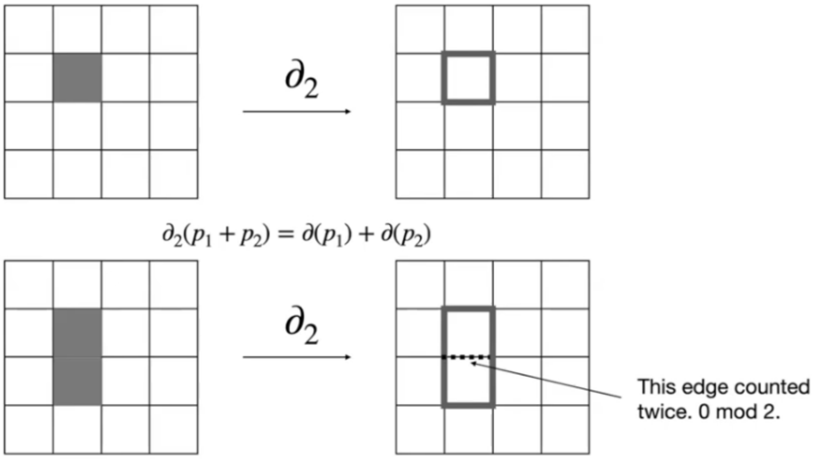
Given a cellulation, an n-chain is the assignment of a group element to every n cell.

The set of n-chains forms avector space V_n over $\mathbb{F}_2$ (under vector addition modulo 2), with a basis vector associated to each n-cell. The identity vector is the zero vector/chain. An example of a 2-chain, 1-chain, and 0-chain are given below.



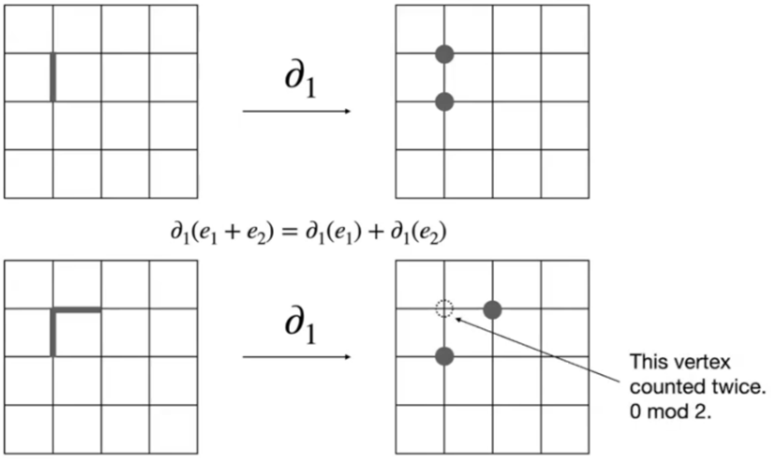
0	0	1	0
1	0	1	1
1	1	0	0
0	1	1	0

The boundary maps ∂_n are linear maps from n-chains to $n-1$ chains. For example ∂_2 takes us from 2-chains (on plaquettes) to 1-chains (on edges):



$$\partial_2(p_1 + p_2) = \partial(p_1) + \partial(p_2)$$

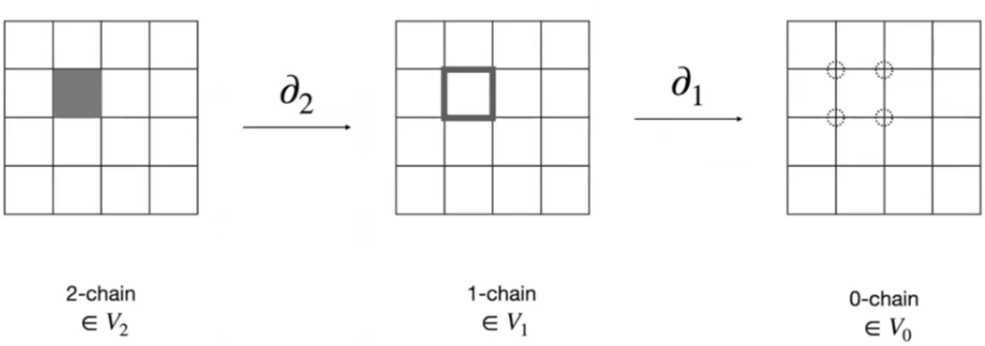
And ∂_1 takes us from 1-chains (on edges) to 0-chains (on vertices):


∂_1
$\partial_1(e_1 + e_2) = \partial_1(e_1) + \partial_1(e_2)$
∂_1
This vertex counted twice. 0 mod 2.

The composition of boundary maps conditions is satisfied, with:

$$\partial_1 \partial_2 = 0 \pmod{2} \tag{B.49}$$

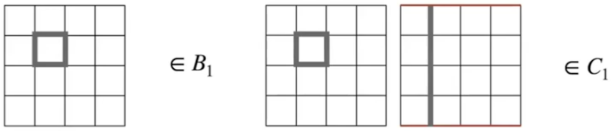
this is illustrated in the simple example below:



and is easy to see intuitively from the fact that the edges around any set of plaquettes involve vertices that are hit by two edges (to form corners).

Definition: *n*-boundaries and cycles

The n-boundary group $B_n \subset V_n$ is the set of n-chains which are boundaries of $(n+1)$ chains.
The n-cycle group $C_n \subset V_n$ is the set of n-chains whose boundary is zero.

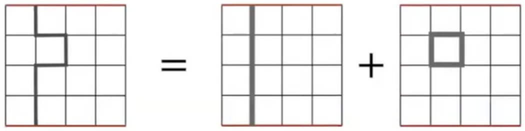


Note that every boundary is a cycle (which follows from the fact that the composition of two boundary maps is zero), but not every cycle is a boundary.

Definition: Homological equivalence

Two chains are homologically equivalent if they are equivalent up to the addition of a boundary.

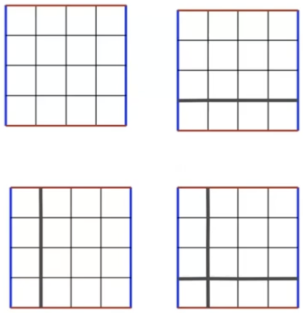
As an example, we can see that the stright line chain is the homologically equivalent to the straight line with a kink:



B.3.4 The Toric Code and Homology/Cohomology

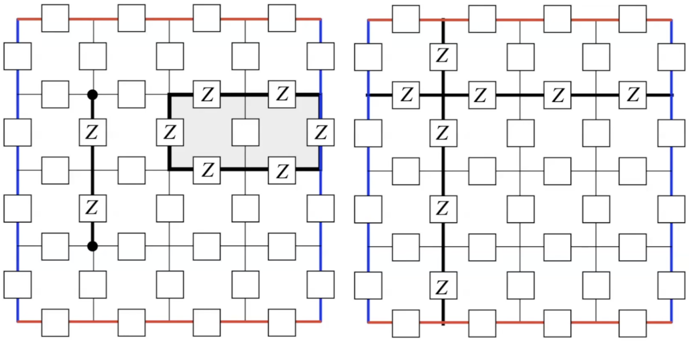
With the mathematical machinery established, let us return to the toric code. First, on the torus we note that there are four equivalence classes of 1-chains. This is characterized by the 1st homology group:

$$H_1 = \frac{C_1}{B_1} \cong \mathbb{Z}_2 \times \mathbb{Z}_2 \tag{B.50}$$

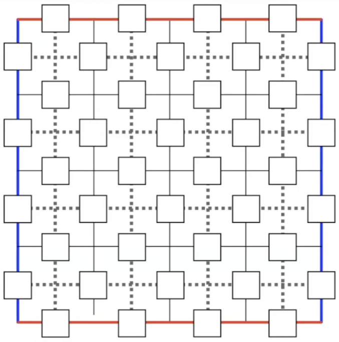


In the homology language, the Z-part of the toric code can be described as follows:

- Qubits live on 1-cells (edges)
- Z operators live on 1-chains
- B_0 (boundaries of 1-chains) correspond to charge syndromes
- B_1 (boundaries of 2-chains/plaquettes) corresponds to the subgroup of Z-stabilizers
- Cycles C_1 commute with all vertex operators
- $H_1 = C_1/B_1$ defines logical Z operators - homological equivalence is equivalence under multiplication by Z stabilizers.

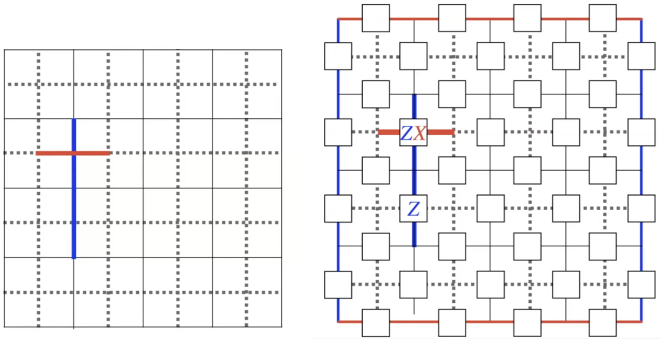


What about the X part of the stabilizer? To this end, we consider the dual to homology, co-homology. We consider the dual lattice, wherein we can define co-chains, co-boundaries, co-cycles, and co-homological equivalence analgously to the primal lattice.



We can consider the inner product between co-n-chains and n-chains. For example $\langle \text{co-1-chain, 1-chain} \rangle \in \mathbb{F}_2$ counts the number of crossings modulo 2. In the same way that Z-operators live on 1-chains, X-operators live on co-1 chains, and their commutation relations are encoded homologically, with:

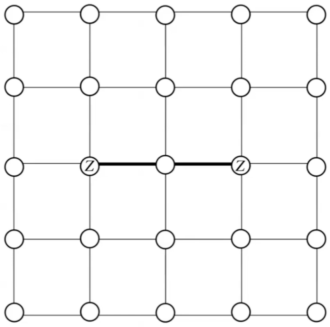
$$[Z(a), X(b)] = (-1)^{\langle b,a \rangle} \tag{B.51}$$



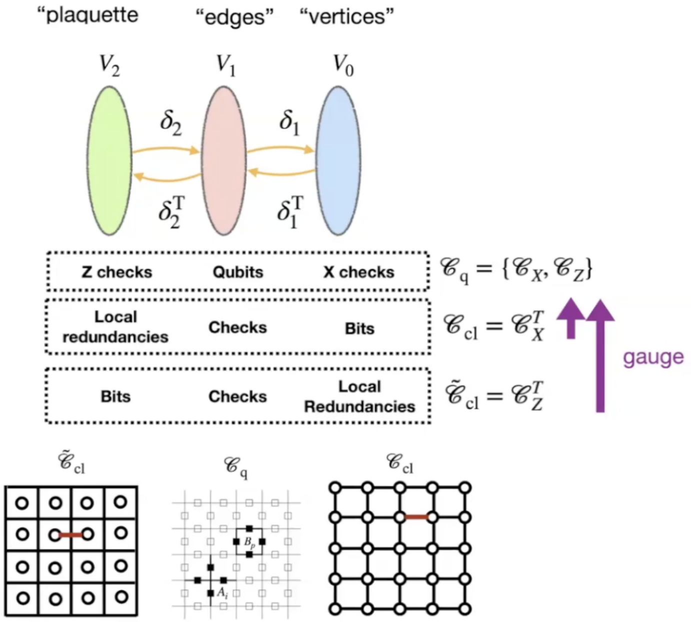
We can view the 2D Ising model with the same lens:

- The bits live on 0-cells/vertices
- Z operators live on 0-chains
- B_0 corresponds to subgroup of Z-stabilizers

- Cycles C_1 are the redundancies of Z-checks; if we have a cycle, we have a product of Z checks that does not act on any vertex.



What is the upshot of this? A chain complex simultaneously defines two classical codes and a quantum code, and we get from the classical to the quantum via gauging.


"plaquette
"edges"
"vertices"
V_2
V_1
V_0
δ_2
δ_1
δ_2^{T}
δ_1^{T}
Z checks
Qubits
X checks
$\mathscr{C}_{\mathrm{q}} = \{\mathscr{C}_X, \mathscr{C}_Z\}$
Local redundancies
Checks
Bits
$\mathscr{C}_{\mathrm{cl}} = \mathscr{C}_X^T$
gauge
Bits
Checks
Local Redundancies
$\tilde{\mathscr{C}}_{\mathrm{cl}} = \mathscr{C}_Z^T$
$\tilde{\mathscr{C}}_{\mathrm{cl}}$
$\mathscr{C}_{\mathrm{q}}$
B_p
A_i
$\mathscr{C}_{\mathrm{cl}}$

The bits, checks ("hyper-edges" - because checks can be more than 2-body terms), and redundancies ("hyper-plaquettes") of the classical code define a notion of geometry (distinct from the geometry of the underlying graph). Although the physical lattice properties/geometry has been appreciated in condensed matter, the code geometry will turn out to be just as important. We will discuss this more in the next lecture.

B.4 Gauging and Boundary Maps

In the last lecture, we will discuss:

- Gauging in terms of boundary maps - what is the property of the classical code living on the chain complex which is being gauged?
- Product constructions as a way to build higher level (e.g. 2D) chain complexes from lower dimensional (e.g. 1D) ones.

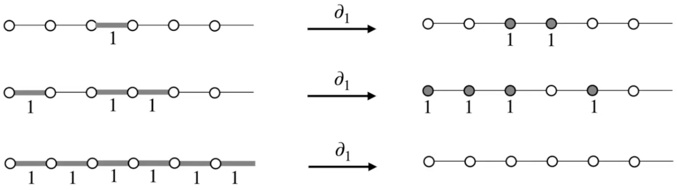
- Product constructions to enrich codes with symmetries (such as subsystem symmetries), starting from simpler codes.

B.4.1 Gauging the 1D Ising Model

Let us review the notion of a boundary map. Bits define an n-dim vector space $V_0 = \mathbb{F}_2^n$. There is a basis vector for each bit $\{|i\rangle\}_i$, and each vector $\vec{b} = (010100\dots b_n)^T$ picks out a subset of bits.

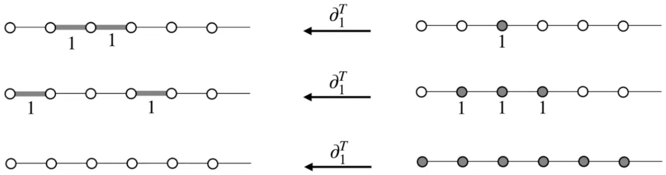
Checks define an m-dimensional vector space $V_1 = \mathbb{F}_2^m$. There is a basis vector for each check $\{|a\rangle\}_a$, and each vector $|c\rangle = (01110\dots c_m)^T$ identifies a subset of checks. The map $\partial_1 : \mathbb{F}_2^m \to \mathbb{F}_2^n$ is the transpose of the parity check matrix, and what it does is $\partial_1|c\rangle = |b\rangle$ where $|b\rangle$ is the subset of bits in the support of the checks $|c\rangle$. The transpose map $\partial_1^T : \mathbb{F}_2^n \to \mathbb{F}_2^m$ (which is just the parity check matrix) does the opposite, mapping from bits to checks $\partial_1^T|b\rangle = |c\rangle$, where $|c\rangle$ are the subset of bits triggered by the bits $|b\rangle$, i.e. the syndrome.

If we look at the 1-D Ising model, we have bits on 0-cells (sites) and checks on 1-cells (edges). We can consider the boundary map ∂_1:



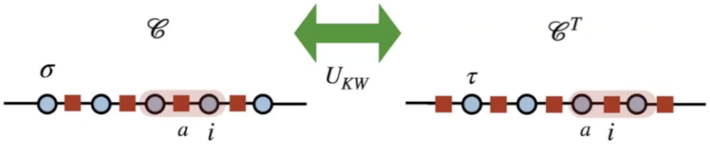
The redundancy is a collection of checks which does not act on any bit (here the product of all checks). ker(∂_1) defines a basis for the vector space of redundancies.

We can also consider the transpose map ∂_1^T:



The symmetry of the model is a collection of (flipped) bits which does not trigger any checks. ker(∂_1^T) defines a basis for the vector space of symmetries.

We can now observe that this transformation is nothing more than a Kramers-Wannier duality! An excitation in the original code becomes a degree of freedom in the transpose code, once we swap the bit and the checks.



$\mathcal{C}$	$\mathcal{C}^T$
m checks (δ): $C_a = \sigma^z_{i-1}\sigma^z_i$	m bits τ^z_a
n bits σ^x_i	n checks (δ^T) $A_i = \tau^x_a\tau^x_{a+1}$
Symmetry $\prod_i \sigma^x_i$	Redundancy $\prod_i A_i = 1$
Redundancy $\prod_a C_a = 1$	Symmetry $\prod_i \tau^z_i$

Note that to define the unitary between the two subspaces, we have to work in the symetric subspace. Then:

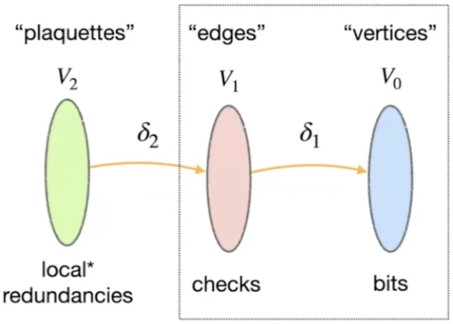
$$\dim(\mathcal{H}_\sigma^{sym}) = 2^{n-k} = 2^{m-k^T} = \dim(\mathcal{H}_\tau^{sym}) \tag{B.52}$$

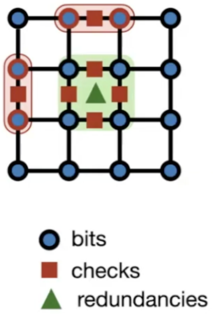
Gauging generally is exactly this - we want to turn symmetry defects (e.g. domain walls) into dynamical degrees of freedom. Another way to think about it is to "make global symmetries local". As we discussed, we can use the transpose $\sigma \leftrightarrow \sigma^T$ to get a new model with the bits and checks swapped. The KW unitary U_{KW} maps between the symmetric subspaces of H, H^T. The last step to get a full gauge theory in terms of the gauge degrees of freedom is to remove this constraint to map between symmetric subspaces.

Note that for the 1D Ising model, this prescription does not yield a non-trivial gauge theory (it just gives us the 1D Ising model back, with $J \leftrightarrow h$ swapped). Thus we need something extra to get a non-trivial gauge theory out; in particular, we require the degrees of freedom to be extended, rather than point like (loop-like in the simplest case).

B.4.2 Gauging the 2D Ising Model

How do we get these extended DoFs? What we will want to do is to take a classical code, but embed it into a chain complex which is larger than just having checks and bits. In particular, let's consider a 2D Ising model, and add to our chain complex a V_2 vector space corresponding to local (few-body, not necessarily geometrically) reduncies of checks.





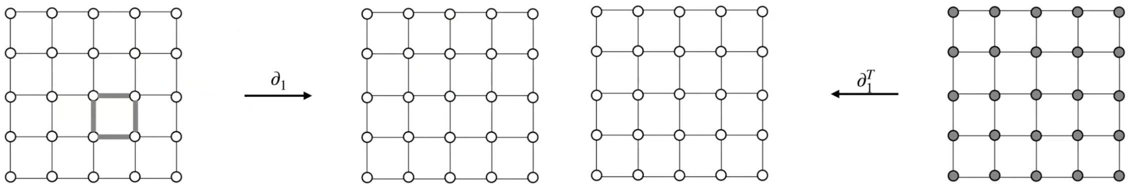
Then:

$$\delta_1 \delta_2 = 0 \tag{B.53}$$

and ker(δ_1) defines a basis for the vector space of redundancies (collections of checks that do not flip any bits), with:

$$B_1 \equiv \text{Im}(\delta_2) \subset \ker(\delta_1) \equiv C_1 \tag{B.54}$$

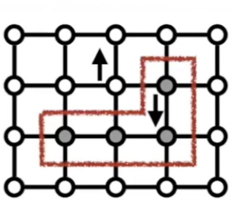
and ker(δ_1^T) defines a basis for the vector space of symmetries (collections of flipped bits which do not trigger any checks).



In the 2D Ising model, each plaquette corresponds to a local redundancy $\prod_{a\in p} C_a = +1$. Thus the domain walls in the model are loop like. This gives rise to the thermal stability of the Ising model, as well as the fact that it yields a non-trivial gauge theory upon gauging. In particular, gauging the 2D Ising model gives us a $\mathbb{Z}_2$-gauge theory, which corresponds to the toric code. Pictorially, the uniform superposition over loop configuration in the ground state of the toric code arises from gauging the loop-like degrees of freedom in the Ising model.

The beauty of the formalism is that - even though we used the Ising model here as an example - the formalism only cares about the (boundary) maps and so the gauging procedure applies to LDPC codes and codes on arbitrary geometries.

When we look at the KW duality for the 2D Ising model, we go from the 2D Ising model $\mathcal{C}$ with one global ($\mathbb{Z}_2$) symmetry of the Ising model with many local redundancies to the toric code $\mathcal{C}^T$ with one global redundancy (obtained by multiplying out all star operators/checks $\prod_v A_v = \mathbb{I}$) and many local symmetries (the B_p plaquettes).



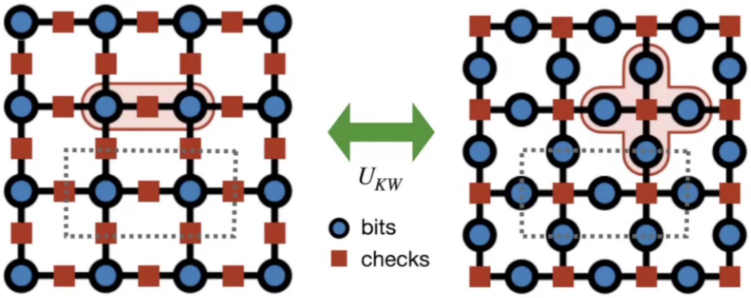
Doing the KW duality, we go from the Ising Hamiltonian:

$$H_{\text{Ising}} = -J \sum_{\langle ij \rangle} \sigma_i^z \sigma_j^z - g \sum_i \sigma_i^x \tag{B.55}$$

where local redundancies $\implies$ domain walls form closed loops, to the TC Hamiltonian:

$$\tilde{H}_{TC} = -J \sum_{\langle ij \rangle} Z_{ij} - h \sum_i A_i^{(x)} \tag{B.56}$$

where local symmetries being enforced $\implies$ $B_p^z = +1$ (no vortex excitation).



If we want to allow for vortex excitation, we follow the standard minimal coupling prescription:

1. Couple interactions to gauge fields $\sigma_i^z \sigma_j^z \to \sigma_i^z \sigma_j^z \otimes Z_{ij}$. This takes the global Ising symmetry and promotes it to a local symmetry.

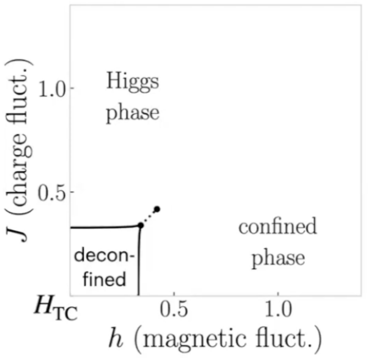
2. Add local gauge invariant terms $-K \sum_p B_p, -h \sum_{\langle ij \rangle} X_{ij}$. The latter makes the background gauge fields dynamical and allows them to fluctuate (lifting the no vortex excitation condition). The former is needed for the toric code, and comes from the redundancy of the original Ising model. By the time we take local products, all matter fields/σ cancel out, leaving us with just the B_p.

3. Enforce the local Gauss law $A_i^{(x)}\sigma_i^x = 1$

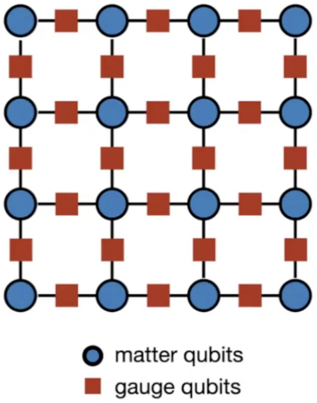
So, promoting our Hamiltonian to:

$$H = -\sum_i \sigma_i^x - h\sum_{\langle ij\rangle} X_{ij} - J\sum_{\langle ij\rangle}\sigma_i^z Z_{ij}\sigma_j^z - \sum_p B_p \tag{B.57}$$

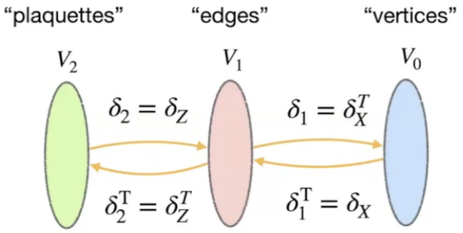
we get a full gauge theory for $\mathbb{Z}_2$ matter coupled to $\mathbb{Z}_2$ gauge fields.



Fradkin, Shenker: PRD 1979



The Ising model was a fruitful example - though we were using it as a guide, boundary maps proviede a completely general gauging prescription.

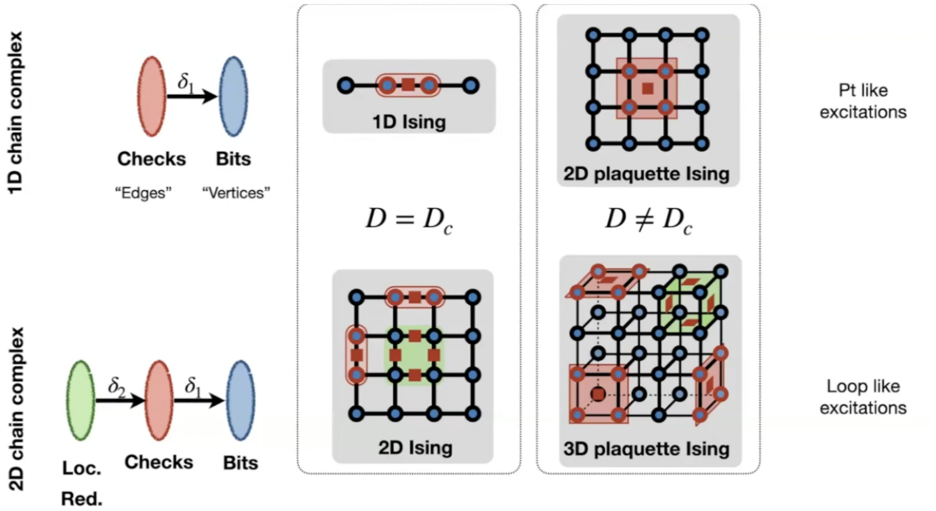


We have the composition relations:

$$\delta_1\delta_2 = 0 \equiv \delta_z\delta_x^T = 0 \tag{B.58}$$

which implies that $[H_z, H_x] = 0$ as we require.

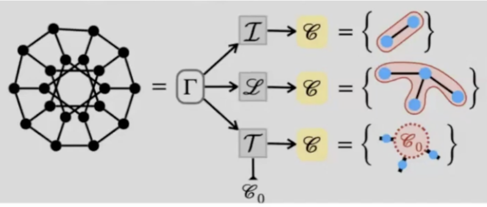
Note that the dimension of the code (characterized by the dimension of the chain complex D_c) does not necessarily coincide with the physical lattice dimension of the code D, for example in the 2D plaquette Ising model (where we can embed a 1-D_c chain complex) or 3D Ising model (where we can embded a 2-D_c chain complex).



B.5 Product Constructions

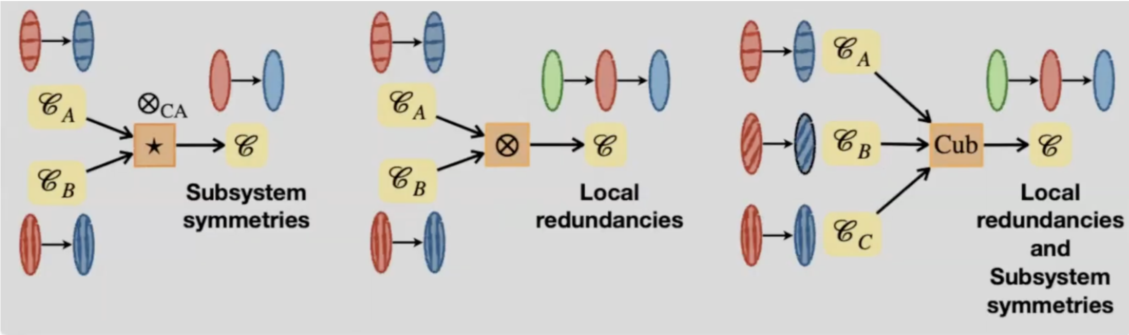
Now, given some classical codes, there are "factories" - systematic constructions we can use to construct quantum codes.

As a first step, considering a given graph structure we can consider constructions of classical code (e.g. Ising, Tanner, or Laplacian (sort of related to Tanner codes)):


$= \Gamma$
$\mathcal{I} \to \mathscr{C} =$
$\mathscr{L} \to \mathscr{C} =$
$\mathcal{T} \to \mathscr{C} =$
$\mathscr{C}_0$
$\mathscr{C}_0$

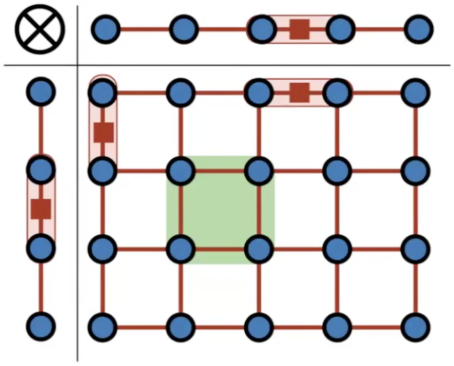
Once we have this, we can start taking products of the classical codes $\mathcal{C}_A, \mathcal{C}_B$. We can now start discussing what these products correspond to physically.

- There are check products, which create subsystem symmetries (which play a large role in fracton models).
- Tensor product, which can create local redundancies,
- A cubic product (which takes in 3 classical codes) and creates a quantum code with both local redundancies and subsystem symmetries.
- Cellular automota product



B.5.1 The Tensor Product

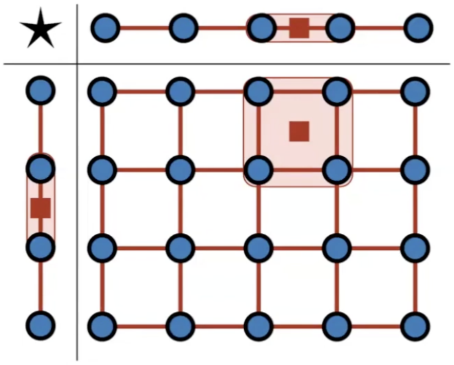
We consider the simplest product construction, the tensor product; we can use this to construct a 2D system out of two 1D systems, e.g. building up the 2D Ising model out of 2 copies of the 1D Ising model.



The tensor product creates local redundancies, increasing the dimension of the chain complex by one $D_c \to D_c + 1$.

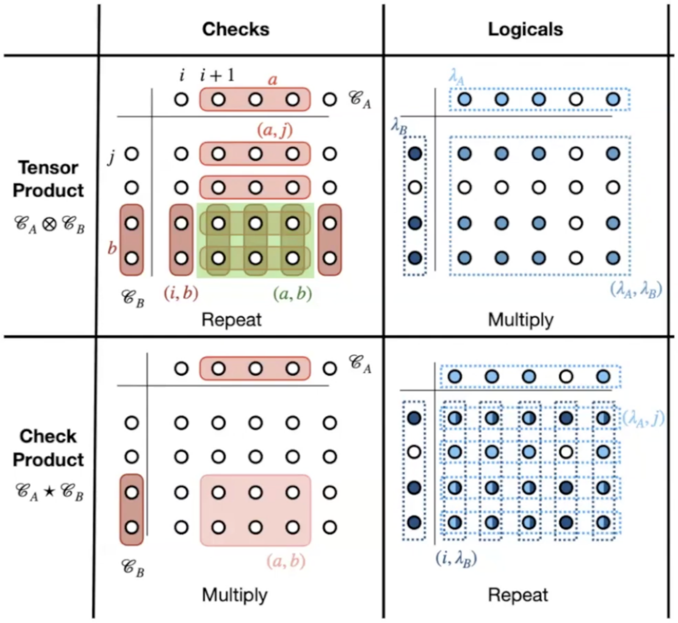
B.5.2 Check product

We can instead consider the check product (which multiplies checks together), which cretes subsystem symmetries - here we see an example of 2 copies of 1D Ising used to construct the Plaquette Ising model:



The check product creates subsystem symmetries, and leaves D_c unchanged.

More generally, we can take any 1-D_c code, lay them out on a line (in 1 spatial dimension), and take each of these products, where the checks/logicals look like:



In the tensor product construction, the code parameters transform as:

$$[n_A,k_A,d_A]\otimes[n_B,k_B,d_B]\cong[n_An_B,k_Ak_B,d_Ad_B] \tag{B.59}$$

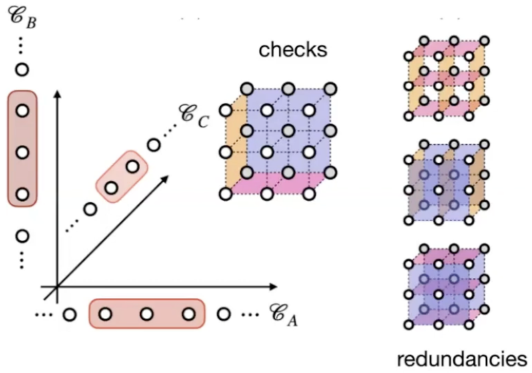
in the check product construction, the code parameters transform as:

$$[n_A,k_A,d_A]*[n_B,k_B,d_B]\cong[n_An_B,k_An_B+n_Ak_B-k_Ak_B,\min(d_A,d_B)] \tag{B.60}$$

Let us explain this one. The definition of the new logicals in the check product construction follows by taking the logicals along each row/column, and then subtracting out the newly added redundancies. The symmetries of the bigger code only act on the lower dimensional planes inherited from the input codes, which is precisely the subsystem symmetry that we get in the new code.

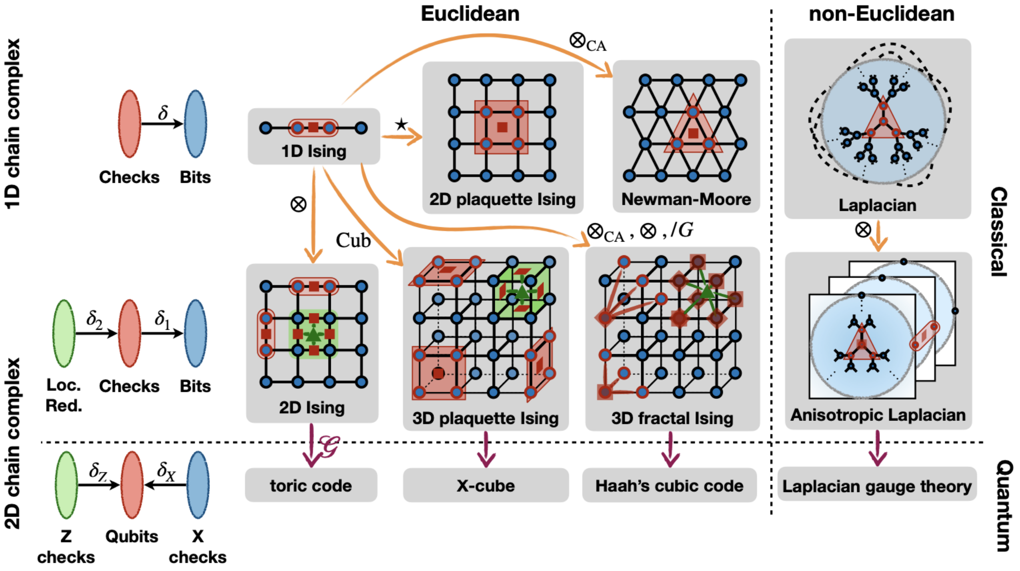
B.5.3 The cubic product

We can also consider the cubic product, which is essentially taking 3 1-D_c classical codes and doing a check product along every plane. In this case we get local redundancies and subsystem symmetries, promoting $D_c \to D_c + 1$.



B.5.4 Gauging classical codes to get quantum codes

Now that we have classical codes with redundancies, we can gauge them to get the quantum codes that we want! The below diagram shows the output of this factory of product construction + gauging:



B.5.5 Hypergraph product code

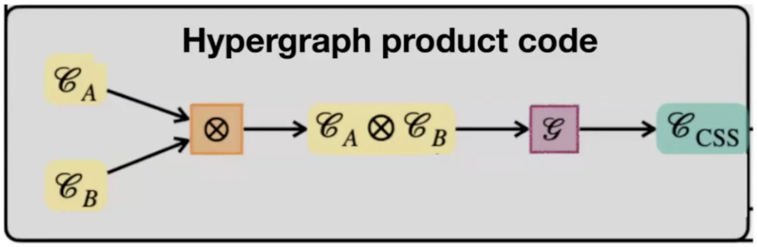
In the CS literature, a very common construction for quantum codes is the "hypergraph product code." Usually, it is formulated coding-theoretically in terms of parity check matrices. But with our physical code factory, we can now interpret it as taking the tensor product of two codes $\mathcal{C}_A, \mathcal{C}_B$ to form $\mathcal{C}_A \otimes \mathcal{C}_B$, and then gauging it to get a CSS code, with X/Z codes given by:

$$\mathcal{C}_X = (\mathcal{C}_A \otimes \mathcal{C}_B)^T, \quad \mathcal{C}_Z = (\mathcal{C}_A^T \otimes \mathcal{C}_B^T)^T \tag{B.61}$$

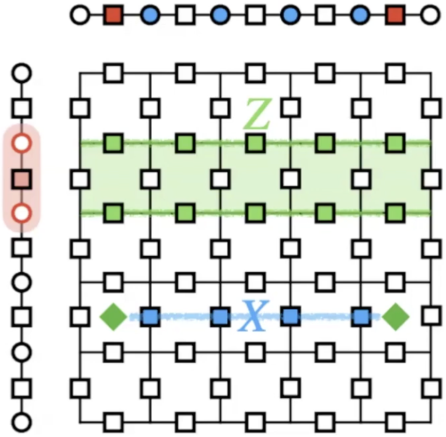
$$d_X = \min\{d_A, d_B\}, \quad d_Z = \min\left\{d_A^T, d_B^T\right\} \tag{B.62}$$

and overall rate:

$$k_Q = k_A k_B^T + k_A^T k_B \tag{B.63}$$



We said that the toric code could be obtained by gauging the 2D Ising model, which itself could be obtained by taking the product of two Ising models - thus we can understand the toric code as the hypergraph product of two 1D Ising models/repetition codes. The Z logicals are derived from the global redundancies of the 1D Ising model (and the fact that they can be deformed arises from multiplying via a check of the 1D Ising model), while the X logicals are derived from the global symmetries of the 1D Ising model. The deconfined point-like (anyonic e/m) excitations arise from the domain walls of the 1D Ising model.



The input code parameters being $[n, 1, n]$ enforces that under the HGP we get code parameters $[n^2, 2, n]$.

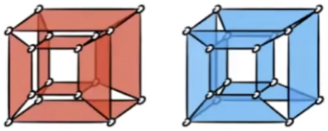
Remark: When we think about the input classical LDPC codes as spin glasses, the excitations are very restricted. When we build a quantum code out of them, we get a mixture of a topological order and a spin glass.

B.5.6 Quantum codes in higher D_c chain complexes

Indeed, we can even go to higher D_c chain complexes, e.g. with the toric code, where $D_c = 4$:

$$\underbrace{V_4}_{\text{local red. } Z \text{ checks}} \overset{\delta_4}{\rightarrow} \underbrace{V_3}_{Z\text{-checks}} \overset{\delta_3 := \delta_Z}{\rightarrow} \underbrace{V_2}_{\text{qubits}} \overset{\delta_2 := \delta_X^T}{\rightarrow} \underbrace{V_1}_{X\text{-checks}} \overset{\delta_1}{\rightarrow} \underbrace{V_0}_{\text{local red. } X \text{ checks}} \tag{B.64}$$

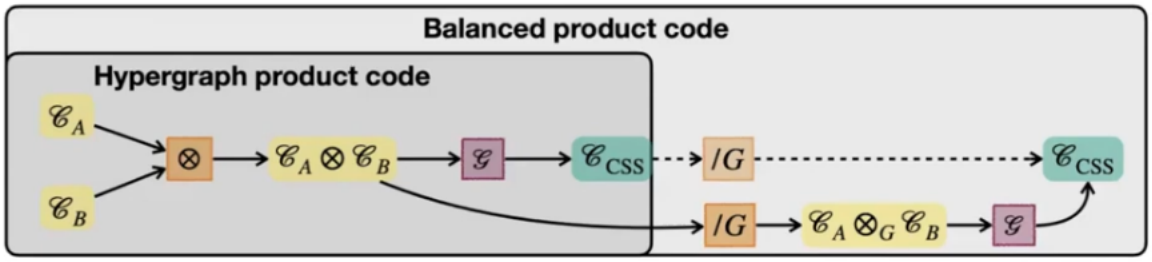
now the X, Z checks have local redundancies, so their excitations become loop like - this yields a thermally stable quantum memory.



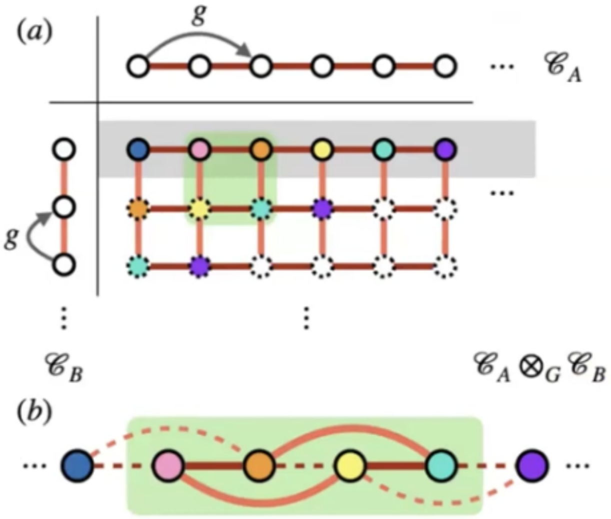
B.5.7 The path to Good qLDPC codes

We saw how taking two repetition codes/1D Ising models with $k = 1, d = n$ (saturating the classical BPT bound) and product + gauging gave the 2D toric code with $k = 2, d \sim \sqrt{n}$ (saturating the quantum BPT bound). If we do this construction with two good classical LDPC codes with $k, d = \Theta(n)$, we get a qLDPC code with $k = \Theta(n), d \sim \sqrt{n}$. This isn't all the way to a good code, but is already quite an improvement.

How do we now boost the $d \sim \sqrt{n}$ to $d \sim n$? We have to dress this product in a clever way. This was first done by Hastings/Haah/O'Donnell in 2009.03921 where they generalize the homological code product to a fiber bundle, which yielded $n^{3/5}$ scaling, with a probabilistic construction. Soon after that, Breukmann and Eberhardt constructed a $k = \Theta(n^{4/5})$ and $d = \Omega(n^{3/5})$ code via a balanced product in 2012.09271 - this requires input codes with some enhanced symmetry (permutation of bits that leave the code invariant - obtained from the Cayley graph of the code). The symmetry is then modded out after taking the product (reducing the number of bits), which boosts the distance (actually the relative distance d/n by reducing n). This balanced product code is pictorially depicted below:



Key point: boost relative distance d/n by reducing n



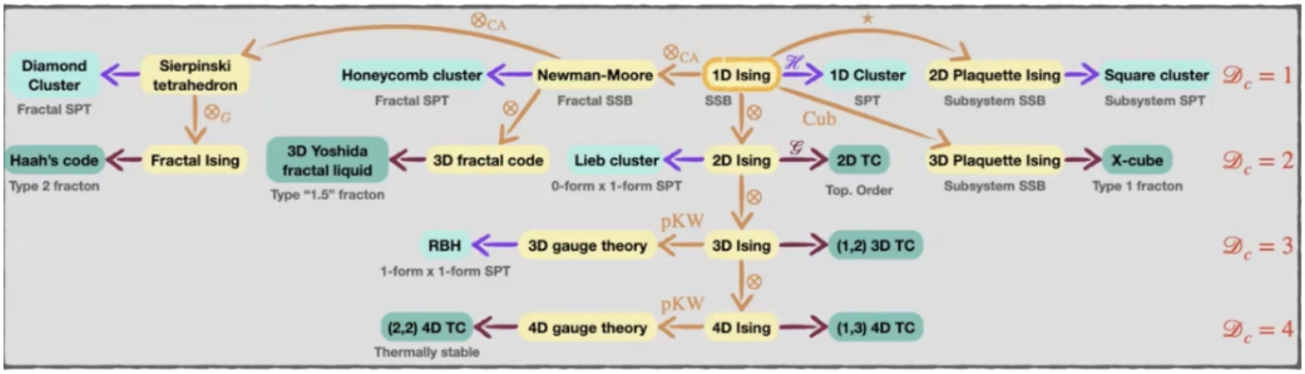
Suppose we had translation symmetry by 2/1 lattice site in 2 1D models. Then, we can identify the 2D system (given its symmetry) as an effectively 1D system with long-range interactions.

This balanced product codes were used by Pantaleev, Kalachev to construct good qLDPC codes in 2111.03654 and were also used to construct good locally testable classical LDPC codes (a property of their energy barriers). It turns out that every good qLDPC code is a gauge dual of a locally testable code.

Remark: Good qLDPC codes also obey the "NLTS" property 2111.03654, which is a very strong form of TO - if we define trivial/non-trivial states by the circuit depth, if we are below some T_c, any state cannot be prepared by a finite depth circuit. This is very different from the toric code - e.g. the 4D toric code is thermally stable but does not have this NLTS property; indeed the eigenstates below T_c are hard to prepare but we can do a sort of "tiling" argument where we prepare small patches of toric codes on tiles of a larger system. The energy penalty scales with the surface/volume ratio, which can be tuned by the size of the tiling, allowing us to create an arbitrarily low energy state (not an eigenstate) efficiently/in finite time. In the good qLDPC setting, because the ratio of surface area to volume is constant, this kind of construction is not possible.

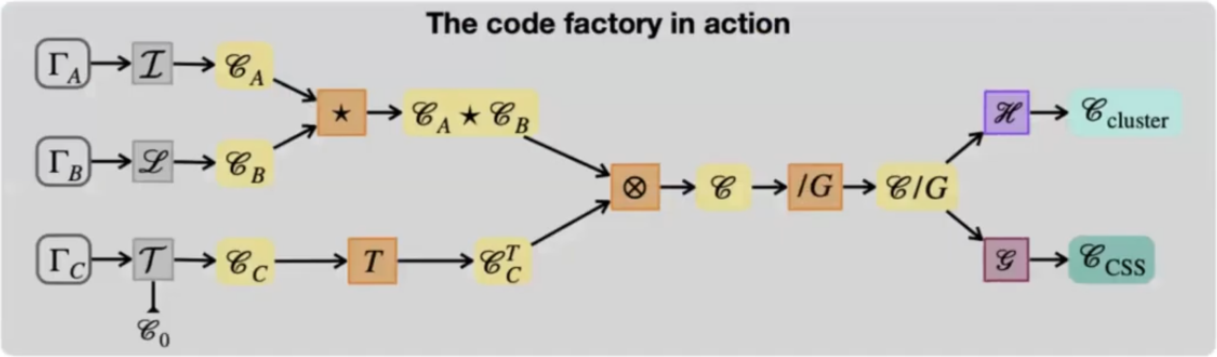
Another comment - ambiguous questions about spin glasses in the Euclidean case resolve much nicely in the expander case when we have this nice scaling of surface area to volume.

This construction gives us a web/zoo of dualities from products and gauging/Higgsing[4], organized in terms of increasing chain complex dimension:



And if we were to replace the 1D Ising model at the root of this tree with a $D_c = 1$ classical expander code, we would get constructions of non-Euclidean analogs of all of these phases/models!

The conceptual upshot of the work is the introduction of a code factory that can take in a graph as input and then produce interesting quantum codes through a physical process:



B.6 Conclusions

- LDPC codes furnish an interesting new classic of stat mech models for many-body physics, leading to novel phenomena in non-Euclidean geometries.
- We can understand qLDPC codes as generalized gauge theories with generalized KW dualities.
- We can recast product constructions in the literature in temrs of physical terms (symmetries, redundancies, excitations). We get a unified descriptiuon of code via homology.
- We can use chain complexes to define SPTs on general graphs, and obtain qLDPC codes by measuring these. (did not get to in detail)
- Tractable classical and quantum spin glasses. (did not get to in detail)

[4] After we have a Tanner graph, we can put a cluster state on that graph, this gives an SPT, and it is exactly the SPT we would need to measure to get a quantum code

C Chao Yin NUS Talk Notes

Based on Chao Yin's talk "Robust many-body localized and error-correcting phases of matter in low-density parity-check codes" here, based on 2405.12279, 2411.01002. Andrew Lucas also gives a talk on these topics which can be found here.

C.1 Introduction - LDPC Codes in Theory and Practice

LDPC codes are a family of error correcting codes that are used everywhere - e.g. in encoding information transmitted from your cellphone to data stations. Originally comes from the computer science/information theory side of things. But as the 2024 Nobel may suggest, it may be fruitful to consider connections to physics!

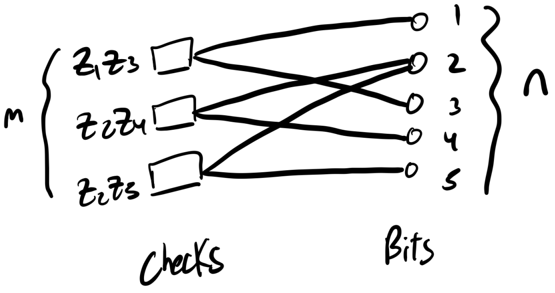
Error correction actually comes up naturally in physics. One needs to look no further than the Ising model and the classical repetition code:

Ising Model	Classical Repetition code
Hamiltonian $H_0 = \sum_i \frac{1-Z_iZ_{i+1}}{2}$	Parity check matrix $H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ \vdots & & & \end{pmatrix}$
Terms are 2-local	Parity checks Z_iZ_{i+1} with weight 2
Ground states $z = 0^n, 1^n$	$2^k = 2$ codewords
Symmetry operator $\prod_i X_i$	Logical operator of the code
SSB	$[n, k, d]$ Error correcting code with large code distance $d = n$

An LDPC code really in some sense is just a generalization of the Ising example. We consider the Hamiltonian:

$$H_0 = \frac{1}{2} \sum_{\text{parity check } C} \left[1 - \prod_{i \in C} Z_i\right] \tag{C.1}$$

with the LDPC condition that $|C| = O(1)$ (the weight of the parity checks is constant) and each bit is in $O(1)$ checks. We can think of the LDPC Hamiltonian as a $O(1)$-degree interaction graph. We can also think about this as a bipartite graph known as a Tanner graph, where the LDPC condition corresponds to the Tanner graph having bounded degree.



A "good" LDPC codes with $d, k = O(n)$ exists on an "expander graph" which is a graph that is locally tree-like + has long loops that make the graph highly connected - cannot be embedded in finite dimension.

Good quantum LDPC codes were discovered in 2022, see 2111.03654. Such codes have interesting physics, and have been used to prove the NLTS (no low-energy trivial state) theorem[5] 2206.13228.

[5] Statement: Existence of (k-local) Hamiltonians such that any low-energy (ϵn for ϵ a constant) state requires a quantum circuit

qLDPC ideas have seen experimental realizations, e.g. Rydberg atoms (Lukin), Trapped ion processors (Quantinuum), Hyperbolic superconducting lattices (Houck), Optical Cavities with all-to-all interactions (Schleier-Smith) which can relax the locality condition native to most physical systems.

This talk will discuss two facets of the physics of LDPC codes. First, we will discuss robust many-body eigenstate localization that arises from a classical LDPC code + a quantum perturbation, discussed in 2405.12279. The argument arises from an extensive energy barrier + detuning. Then, we will discuss the stability of qLDPC codes to quantum perturbations, as discussed in 2411.01002. We will discuss what properties are required for stability beyond being "good" codes.

C.2 Story 1 - Many-Body Localization

C.2.1 ETH and its violations

We consider a quantum many-body system that starts in some pure product state, and evolves under its own Hamiltonian $U(t) = e^{-iHt}$. The folklore is that at long times, if we look at the local density matrices, $\rho \approx \rho_{\text{thermal}}$. The underlying mechanism is the eigenstate thermalization hypothesis, $\rho_{\text{eigenstates}} \approx \rho_{\text{thermal}}$ (for local probes). This is the QM analog of notions of ergodicity/scrambling and gives rise to quantum statistical mechanics. But it is a negative in the sense that it doesn't permit interesting long-time behavior - so counterexamples to this would be interesting! If they existed, perhaps we could use them to store or process information.

Indeed, there are known counterexamples to ETH:

- Integrability, wherein we have many conserved quantitites
- Hilbert space fragmentation into Krylov subspaces that do not interact with each other.
- Anderson localization due to disorder.

But all three of these are fairly fine-tuned examples. E.g. with Hilbert space fragmentation, if we introduce any kind of connection between the subspaces, the eigenstates would mix and we would start to see ETH. Or with integrability if the symmetries are not preserved then we would see ETH etc.

So as a non-finely tuned example, we consider many-body localization. Even if we look at couplings between various states, if we look at the eigenstates, they are still localized in some corner of the Hilbert space. E.g. some exponential suppression around the $|0^n\rangle$ state (where the "weight" is concentrated, and then the weight is supressed as $\sim e^{-n}$ in the number of bitflips). It is hoped that some systems may achieve this, with the most widely studied system being the 1-D disordered spin chain. Dating back to Anderson, we have developed deep physics theories arguing for MBL. But recent numerics 2408.04338 have called into question whether this actually persists in the thermodynamic limit or is a finite size effect. The analytical proofs generally require on non-generic assumptions or the limit of very strong disorder. So, can we come up with a system where we can prove MBL more generically/unambiguously?

C.2.2 Classical LDPC Codes as MBL candidates

We consider good codes with "local testability':

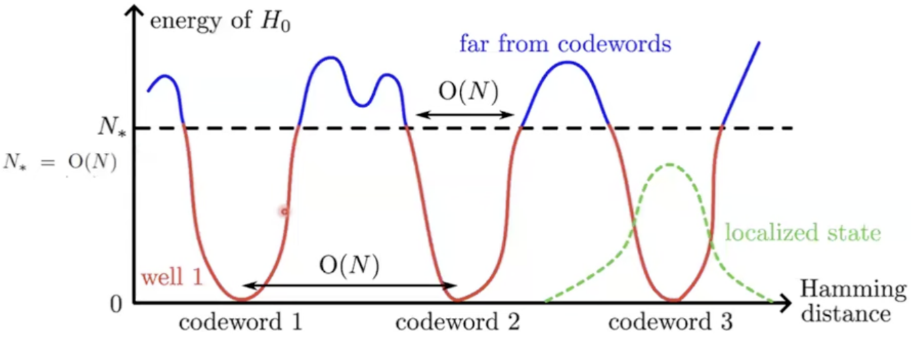
$$|H\mathbf{x}| \geq \alpha \min_{\text{codeword } \mathbf{z}} |\mathbf{x} - \mathbf{z}| \tag{C.2}$$

i.e. that the energy of a codeword should be at least proportional to the minimal distance between codeword. The low energy states of locally testable codes thus have low error. Here $\mathbf{x}$ is a bitstring/computational basis state. Why is this condition called local testability? Suppose I give you a codeword or a codeword

of depth $\Omega(\log n)$ to prepare. Proof overview - constant rate and linear distance quantum codes have expansion properties of their check matrices, which implies their approximate code words are clustered. It can be shown for such codes that the classical distributions of the low-energy states of these Hamiltonians are clustered and also well-spread/not concentrated, which gives rise to the circuit lower bound.

with a finite number of errors. Then, we want to check local parts of the bitstring to check which of the two cases it is - if this condition is obeyed, then local checks are sufficient to verify whether **x** is a codeword or not.

What does local testability buy us? The energy landscape of this "nice" classical code is such that the codewords are far-spaced apart/require the surmounting of an $O(N)$ energy barrier. This is exactly the MBL condition we were looking for. It is thought that spin glass models have similar features, but LDPC setting is a bit simpler/cleaner; we do not require frustration or long-range couplings (? - I am confused by this that somehow the non-local properties of good LDPC codes are not leveraged).



Consider the Hamiltonian:

$$H = H_0 + V + H_L \tag{C.3}$$

with H_0 a (classical) LDPC Hamiltonian,

$$H_L = \frac{\epsilon}{\sqrt{N}} \sum_i h_i Z_i \tag{C.4}$$

and V any $O(1)$-local quantum Hamiltonian with $\|V\| \leq \epsilon N$. h_i is a detuning parameter, and is i.i.d. Gaussian random with 0-mean and 1-variance. So, $V + H_L$ is a generic few-body perturbation.

Then we have the theorem.

Theorem: MBL for LDPC codes

If $\epsilon < C_0$, then with high probability $1 - 2^{-N}$, any eigenstate $|\psi\rangle$ at low energy $E < C_0 N$ is localized in one well $z(\psi)$:

$$\left\| (1 - P_{z(\psi)}) |\psi\rangle \right\| \leq 2^{-N} \tag{C.5}$$

thus we have a robust ETH at low temperature! As a corollary, starting in one well, the state does not escape for any time t.

Proof (sketch).

1. A nice LDPC code exists with exponentially many (required for the localization to be an exponentially small subspace) deep wells. Known from classical coding theory.

2. At low energy, the extensive energy barriers suppress couplings (off-diagonals) among wells (diagonals):

   - $H = H_> + \sum_{\text{wellz}} H_z$ with $\left\| H_> |\text{low } E\rangle \right\| < \epsilon^N$

3. This energy difference is much smaller than the typical energy difference 2^{-N} from detuning. Thus, from a PT argument we can see that the $H_>$ will not mix energy eigenstates.

This proof is somewhat analogous to the proof of Anderson localization.

- Hopping $\epsilon^N \ll$ energy difference 2^{-N}
- $\|V\| \sim \epsilon N \ll$ energy barrier $\sim N$
- High energy states are thermal - mobility edge

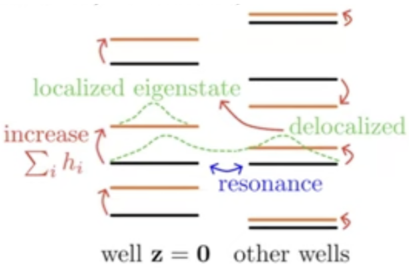
A bit more detail on the third step of the proof (detuning $\implies$ no degeneracy $< \epsilon^N$).

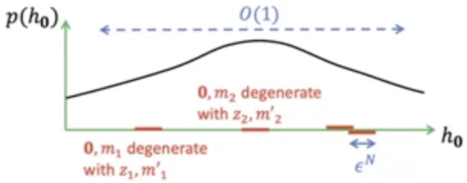
- Let us focus on H_0 vs the other wells H_z.
- i.i.d. Gaussian h_i implies that we have a Gaussian $h_0 := \frac{1}{\sqrt{N}} \sum_i h_i$, with $H_L \propto h_0 \sum_i Z_i$.

Suppose I have an "unlucky" resonance between wells, so that the eigenstates may be delocalized. Then, let us increase h_0 slightly, causing the spectrum to shift:

- $E_{0,m}$ increases with maximal slope $\langle \sum_i Z_i \rangle \approx N$ (Feynman-Hellmann)
- $E_{z,m'}$ of other wells have slope $< N - D/2$

The degeneracy probability is then $\leq 2^{2N} \cdot \epsilon^N < 2^{-N}$ with 2^{2N} the number of energy pairs.





C.3 Story 2 - Stability of quantum LDPC codes

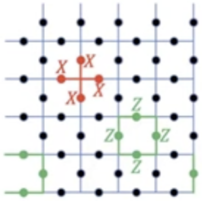
C.3.1 Stability to local and nonlocal perturbations

We will show that the error correcting ground states of qLDPC codes are stable against generic perturbations, and discuss some extra properties required for the stability to hold (beyond being "good" codes).

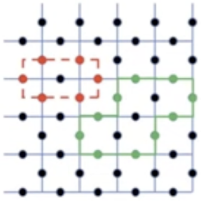
As a warm-up, let us consider the prototypical toric code (Kitaev, quant-ph/9707021):

$$H_{\mathrm{TC}} = -\sum_v A_v - \sum_p B_p \tag{C.6}$$

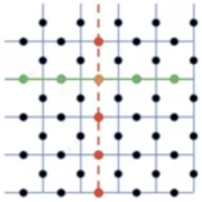
this is a commuting stabilizer Hamiltonian defininied on a torus, and is a QEC with parameters $[[L^2, 2, L]]$.



The stabilizer group is generated by the stars A_v and plaquettes B_p in the Hamiltonian, and consists of all contractible Z, X loops.



The logical operators $\overline{X}_1, \overline{Z}_1, \overline{X}_2, \overline{Z}_2$ are not in the stabilizer group but commute with all stabilizers. They are non-contractible Z/X loops going around the torus.



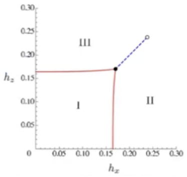
We know that the toric code represents a stable phase of matter; we can consider the perturbed toric code:

$$H = H_{\text{TC}} + V \tag{C.7}$$

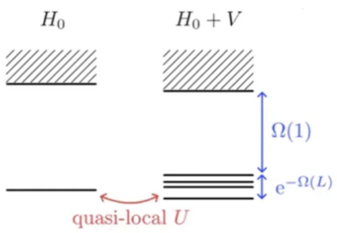
with:

$$V = -h_x \sum_i X_i - h_z \sum_i Z_i \tag{C.8}$$

wherein we obtain the phase diagram (Vidal, 0807.0487):



where we get a stable toric code phase (I) with topological order (vs. magnetically ordered phases II/III, wherein the charges (e)/fluxes (m) condense). Topological order/finite dimensional codes with local topological order is generically known to be stable under generic local perturbations (Bravyi, Hastins, Michalakis 1001.0344):

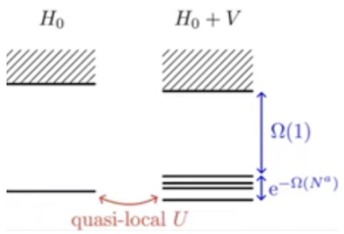


wherein the gap between the encoded and the ground states is $\Omega(1)$, the energy splitting is exponentially small in the system size L, and perturbed and unperturbed ground states are connected via a quasi-local unitary. We may then ask about stability to nonlocal perturbations, as well as the stability of good qLDPC codes (as opposed to the TC which has asymptotically zero rate). This has been explored in prior works, e.g. 2405.19412 (Lavasani, Fullans, Albert, Barkeshli) which shows stability in some cases and 2406.15757 (Li, O'Dea, Khemani) which gives physical arguments for stability. However, whether the family of good qLDPC codes is stable or not was not resolved.

Theorem: Stability of good qLDPC codes

Suppose $O(1)$-local $V = \sum_i V_i$ with $\sum_{i,V_i \text{ touches a given qubit}} \|V_i\| < 0.1$. Then, if a qLDPC (stabilizer) code has "check-soundness", it is stable with $e^{-\Omega(d)}$ splitting (which is exponential so long as $d \sim n^\alpha$)
This is known to apply to:

1. All known good qLDPC codes
2. hypergraph-product expander codes with $d \sim \sqrt{n}$ (Tillich and Zemor 0903.0566)



Implications:

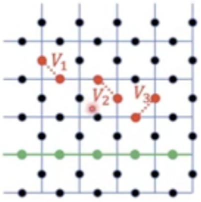
- Robust violation of third law of thermodynamics (as conjectured by 2405.19412) - this is because in the ground state subspace we have robustly many exponential degeneracies/extensive entropy.
- The toric code is not only stable against spatially local perturbations (1001.0344), but also stable against spatially nonlocal perturbations!
- Classical LDPC codes (with check soundness) can be stable against symmetric perturbations - generalizing the known result that the Ising Hamiltonian is stable against symmetric (X) perturbations but not Z perturbations.
- We get stronger locality bounds on the quasi-local unitary U, which is reported on in 2502.02652

C.3.2 Does QEC imply stability?

For the remainder of the talk, we elaborate on the check-soundness condition. To open this discussion, we ask; does QEC imply stability generally? Well, we know that error correction implies that ground states are locally indistinguishable:

$$\rho_\mu^{\text{local}} = \rho_\nu^{\text{local}} \tag{C.9}$$

thus for local $V = \sum_i V_i$ then first order perturbation theory yields $\langle \psi_\mu | V_i | \psi_n \rangle \propto \delta_{\mu\nu}$ or equivalently $PVP \propto P$ - the V_i cannot take us to different parts of the ground state subspace. For higher/k-order perturbation theory, $PV_1 \dots V_k P \propto P$ unless $V_1 \dots V_k$ forms a logical, which requires $k \sim L$, implying an e^{-L} energy splitting. This argument seems to favour the stability of QEC codes.



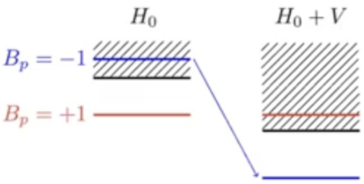
But! Even $V = \sum$stabilizers may destroy the gap. Consider the "Ising" toric code:

$$H_0 = -\sum_v A_v - B_{p_0} - \sum_{\langle p,p' \rangle} B_p B_{p'} \tag{C.10}$$

where the sum runs over neighbouring plaquettes. In this case, we get the same ground states of the toric code. However, the Ising toric code is unstable against:

$$V = \epsilon \sum_p B_p \tag{C.11}$$

Because the $B_p = -1$ becomes more energetically favourable:



so, QEC is not sufficient for stability.

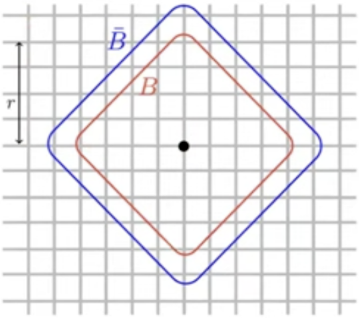
C.3.3 Local Topological Order

What people have considered is the notion of local topological order (1001.0344), which is a stronger version of the error correction condition that is sufficient for stability.

Definition: Local topological order

For any ball region B, let $\overline{B} = B \cup \text{Neigh}(B)$. The LTO condition is then that for any ball with $r \ll L$ (with L, say, the code distance), the local ground states $P_{\overline{B}}$ are indistinguishable, with:

$$P_{\overline{B}} O_B P_{\overline{B}} = c P_{\overline{B}} \tag{C.12}$$

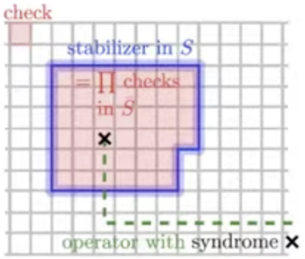


This is a stronger version of the error correction condition, as if we take $\overline{B}$ to be the whole system we recover the error correction condition. This rules out scenarios like the "Ising" TC because the $B_p = \pm 1$ subspaces are distinguishable locally. Note that for stabilizer codes that any stabilizer O_B is a product of checks in $\overline{B}$.

C.3.4 Generalizing LTO - check soundness

How do we generalize LTO to qLDPC codes/expander graph settings? The generalization is not straightforwards. First, its worth noting that these ball regions are not good, because we can only consider balls with $L \leq \log N$ due to expansion properties. But, we cannot control perturbations $V_1 V_2 \dots V_k$ at order $k \geq \log N$. So, we can only get an inverse polynomial energy splitting, rather than exponential.

Other suggestion - could we consdier general regions with bounded volume instead of diameter? No - LTO does not hold for such general regions, and we can even just consider the toric code:



Does this imply that we must fill holes in infinite dimensions? This seems quite daunting. Instead, we consider an algebraic construction.

Definition: Check soundness

A code has check soundness f, if each weight-x ($x < d$) stablizer can be expanded as a product of $\leq f(x)$ checks.

The (2D) toric code has $f(x) \sim x^2$, because the weight x loop stabilizer can be expanded as a product of $\sim x^2$ checks on the interior of the loop. For LTO in d dimensions, we have that $f(x) \sim x^d$.

The best case is a linear check soundness with $f(x) \sim x$ - the LDPC conditions oif constant stabilizer weight and qubit degree ensure that linear is the best we can do.

Interestingly, the linear check soundness is equivalent to the local testability of the corresponding classical LDPC code. Morally, this is because local testability means that low energy states are low error states (as a map from checks (low energy) to bits (low error)). Linear check soundness is the other way around, where a small stabilizer implies a small number of checks (as a map from bits (small stabilizer) to checks (small number of checks)). Check soundness thus becomes a natural figure-of-merit for diagonising qLDPC codes.

Stability holds for better-than-quadratic $f(x) \sim x^{2-\delta}$, or any polynomial $f(x)$ in finite dimensions. Indeed, all known good qLDPC constructions satisfy this.

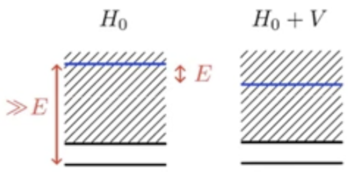
As a non-example, we note that the Ising TC does not have check soundness, because:

$$B_{p_n} = B_{p_0}(B_{p_0}B_{p_1})\dots(B_{p_{n-1}}B_{p_n}) \tag{C.13}$$

If we consider a generic perturbed Hamiltonian:

$$H = H_0 + V = -\sum \text{checks} + \epsilon \sum \text{local stabs} \tag{C.14}$$

check-soundness implies that each local stabilizer is associated to a small number of checks. For a state to gain a lower energy E from violating stabilizers, it needs to violate many checks such that the energy penalty from H_0 is $\gg E$.



Thus the gap is stable.

C.4 Conclusion + Outlook

In the first part of the talk, we showed ergodicity breaking from energy barriers, proving robust localization for classical LDPC codes.

- Can we give a proof for finite-dimensional models with long-range interactions?
- Can we break ergodicity up to long timescales? Quantum metastability (2408.05261)

In the second part, we showed that quantum LDPC codes represent stable error-correcting phase of matter.

- What does the classification of these error correcting phases look like?
- Are excited states similarly stable? This might tell us that qLDPC codes are stable quantum memories.
- NLTS property has been proven for qLDPC codes... can we show that there is true quantumness at finite temperature? This relates to the longstanding open quantum PCP conjecture.