Explanatory Research Notes on

SoS Lower Bounds, SS-HDX and NLTS

Sanchayan Dutta
dutta@ucdavis.edu

Contents

Disclaimer

These notes are intended to facilitate my personal understanding and clarification, based primarily on my readings of the papers: NLTS Hamiltonians from good quantum codes and Explicit Lower Bounds Against $\Omega(n)$-Rounds of Sum-of-Squares.

I've generated most of these notes through my interactions with ChatGPT-4, an AI language model, which assisted me in quickly parsing the dense content of the papers. Any conceptual or typographical errors present in these notes can be attributed to either my oversight or potential limitations of the ChatGPT-4 model.

The following video talks/lectures were referenced during the preparation of these notes. Often screenshots have been directly included where relevant.

1. Anthony Leverrier's introduction to quantum LDPC codes from left-right Cayley complexes at Technion.

2. Chinmay Nirkhe's presentations on the NLTS theorem at QIP 2023 and at Simons Institute.

3. Max Hopkins' discussion on high-dimensional expanders and hardness of approximation at IISc Bangalore.

1 The NLTS Theorem / Conjecture

1.1 The NLTS and quantum PCP

According to the No Low-Energy Trivial State (NLTS) conjecture, originally put forth by Freedman and Hastings, there exist families of Hamiltonians (which describe the total energy of quantum systems) such that all their low-energy states have non-trivial complexity. This complexity is measured by the quantum circuit depth needed to prepare the state.

[ABN] proves this conjecture by showing that certain families of quantum low-density parity-check (LDPC) codes correspond to NLTS local Hamiltonians. This means that these quantum codes map to Hamiltonians that satisfy the conditions laid out in the NLTS conjecture.

The introduction also relates the NLTS conjecture to the quantum PCP conjecture, one of the most important open questions in quantum complexity theory. This conjecture asserts that local Hamiltonians with a constant fraction promise gap remain QMA-complete, which is the quantum analog of NP-complete problems.

The paper suggests that proving the NLTS conjecture could shed light on the validity of the quantum PCP conjecture. However, proving the NLTS conjecture itself has been challenging in the most general case.

1.2 NLTS from quantum LDPC codes

[ABN] introduces the main result, that there indeed exist such NLTS local Hamiltonians. The Hamiltonians in question are associated with quantum LDPC error-correcting codes that have an additional property related to the clustering of approximate codewords of the underlying classical codes.

Finally, the introduction lists a series of open questions. These are related to whether the property of clustering approximate codewords holds for all constant-rate and linear-distance quantum codes, the relationship between this property and the small-set boundary and co-boundary expansion, and whether the proof techniques can be generalized for non-commuting Hamiltonians.

In simple terms, it has shown that the NLTS conjecture is true. This conjecture states that for certain families of quantum systems (described by Hamiltonians), the lower-energy states have high complexity, meaning they need complex quantum circuits to be prepared.

The breakthrough was proving this conjecture by connecting it to quantum error-correcting codes. More specifically, they found that families of quantum low-density parity-check (QLDPC) codes that have constant rates and linear distances correspond to these Hamiltonians.

Quantum low-density parity-check (QLDPC) codes are a type of quantum error-correcting code, which help protect quantum information from errors due to decoherence and other quantum noise. The fact that these codes correspond to the Hamiltonians of the NLTS conjecture is a significant result, as it potentially provides a new way of studying and understanding these complex quantum systems.

It may also have implications for quantum computing, as understanding the complexity of low energy states could be important for quantum algorithm design and error correction. Their work could thus represent a substantial contribution to the field of quantum information and computation.

1.3 QMA-complete local Hamiltonian problem and quantum PCP

The QMA-complete local Hamiltonian problem is presented as a quantum analogue of the NP-complete constraint satisfaction problem (CSP). In simpler terms, the challenge of finding the lowest energy state of a quantum system (the local Hamiltonian problem) mirrors the difficulty of solving certain classical problems (the constraint satisfaction problem).

The quantum PCP (Probabilistically Checkable Proofs) conjecture is also highlighted as one of the most important open questions in quantum complexity theory. It essentially posits that certain problems remain "hard" (QMA-complete) even when a bit of approximation or "promise gap" is allowed. This is analogous to the classical PCP theorem which established that certain problems remained NP-complete even when approximations were permitted.

1.4 The relation between NLTS and quantum PCP

The NLTS conjecture proposes that for a given family of local Hamiltonians (which describe systems of n qubits), any low-energy state (with energy less than a certain fixed fraction of the total number of qubits, ϵn) cannot be prepared by a simple (constant depth) quantum circuit. This essentially means that these low-energy states are complex and not easily producible, hence the name "No Low-Energy Trivial States".

This conjecture is seen as a direct consequence of the quantum PCP conjecture. This is because if the NLTS conjecture were false, it would imply that there is a simple quantum solution to a problem that is expected to be QMA-complete, essentially contradicting the quantum PCP conjecture. The NLTS conjecture can thus be viewed as addressing the issue of how much quantum states of local Hamiltonians can be approximated using classical resources.

1.5 Wait, what is the quantum PCP conjecture?

The classical PCP theorem, a cornerstone of theoretical computer science, says that for every decision problem solved by a nondeterministic Turing machine, there is a "proof" that can be checked probabilistically by examining a constant number of random positions.

The Quantum PCP Conjecture states that the problem of approximating the ground state energy of a local Hamiltonian is QMA-complete. Here, a local Hamiltonian is a simple model for the energy of a quantum system, where the Hamiltonian (energy operator) is a sum of terms, each of which involves only a constant number of particles.

QMA (Quantum Merlin-Arthur) is the class of problems for which a "yes" answer can be proven to a quantum verifier by a quantum proof, whereas if the answer is "no" then no quantum proof can convince the verifier otherwise with high probability. QMA-completeness is an indicator that the problem is one of the hardest problems in the QMA complexity class, in the sense that any problem in QMA can be efficiently reduced to it.

Despite evidence both supporting and contradicting the quantum PCP conjecture, its validity remains undetermined, signifying a major open problem in quantum information theory. [ABN] contributes to this ongoing dialogue in the quantum computing and complexity theory community.

1.6 The Key Theorem

Theorem 1 (No low-energy trivial states) [ABN]

There exists a fixed constant $\epsilon > 0$ and an explicit family of $O(1)$-local frustration-free commuting Hamiltonians $\left\{\mathbf{H}^{(n)}\right\}_{n=1}^{\infty}$ where $\mathbf{H}^{(n)} = \sum_{i=1}^{m} h_i^{(n)}$ acts on n particles and consists of $m = \Theta(n)$ local terms such that for any family of states $\{\psi_n\}$ satisfying $\operatorname{tr}\left(\mathbf{H}^{(n)}\psi\right) < \epsilon n$, the circuit complexity of the state ψ_n is at least $\Omega(\log n)$.

This theorem provides a significant advancement in understanding the complexity properties of low-energy states in quantum many-body systems. It is essentially saying that there is a specific family of Hamiltonians (i.e., quantum mechanical operators representing the energy of the system), which are local and frustration-free, such that any low-energy state of these Hamiltonians requires a quantum circuit of a nontrivial size (measured by the circuit depth) to be generated.

Let's break down some key terms:

- **Local Hamiltonian**: These are physical systems where each particle (or qubit in the case of a quantum computer) interacts only with its nearby neighbors. Mathematically, these Hamiltonians are sums of terms, each of which acts nontrivially only on a small number of particles. The "$O(1)$-local" here means that the number of particles that each term acts on is a constant (does not grow with the system size).

- **Frustration-free**: A system is said to be frustration-free if there is a global ground state (a state of minimal energy) where each local term in the Hamiltonian is minimized. In other words, all local interactions can be simultaneously satisfied.

- **Commuting Hamiltonian**: This means that all the local terms in the Hamiltonian commute with each other, i.e., the order in which they are applied does not matter. This is a special class of Hamiltonians, as not all quantum systems have this property.

- **Circuit complexity**: This is a measure of the size of the smallest quantum circuit (a sequence of quantum gates) that can prepare a given state from some simple initial state (like all particles in the state 0).

The theorem states that if a state has energy less than ϵn (where $\epsilon > 0$ is some fixed constant and n is the number of particles), then the complexity of the state is at least $\Omega(\log n)$. Here, $\Omega(\log n)$ means that the complexity grows at least logarithmically with the system size.

In essence, this theorem asserts the nontriviality of low-energy states in certain quantum systems, as evidenced by their circuit complexity. Such states cannot be easily prepared, which is an important consideration in various fields, including condensed matter physics and quantum computing. The complexity here is typically measured by the quantum circuit depth necessary to prepare the state. Quantum circuit depth is a measure of the computational resources required to implement a quantum computation: the deeper the circuit, the more complex the computation.

In this context, a "trivial" state would be one that could be prepared with a quantum circuit of shallow (i.e., constant) depth, no matter how large the system is. So, the NLTS conjecture asserts that for the systems it concerns, all low-energy states require quantum circuits of more than constant depth – they require "super-constant" depth, which increases with the size of the system.

1.7 Quantum LDPC codes

A quantum Low-Density Parity-Check (LDPC) code is a type of quantum error correction code that shares some of the favorable properties of classical LDPC codes. Quantum codes are used to protect quantum information from errors due to decoherence and other quantum noise.

LDPC codes, in the classical setting, are a type of error correcting code characterized by a sparse parity-check matrix. This sparsity leads to efficient algorithms for error correction. Classical LDPC codes have been widely used in communication systems due to their capacity-achieving performance and efficient decoding algorithms.

In the quantum setting, a quantum LDPC code is a kind of stabilizer code where the stabilizer generators involve only a few qubits (they are "low-density"). These codes are particularly interesting because of their potential for fault-tolerant quantum computation.

Quantum LDPC codes are not as well-understood as some other types of quantum error-correcting codes, like the surface code. Nevertheless, there has been significant interest in them because of their potential for high error thresholds and efficient decoding, which are important properties for practical quantum error correction. However, designing quantum LDPC codes that are both high-rate and have good minimum distance is a challenging open problem.

[ABN] introduces a significant connection between quantum error-correcting codes, specifically Quantum Low-Density Parity-Check (QLDPC) codes, and the NLTS (No Low-Energy Trivial States) conjecture.

The robust circuit-lower bounds, which verify the NLTS conjecture, apply to local Hamiltonians associated with certain quantum codes. Specifically, the codes in question are constant-rate and linear-distance QLDPC codes, which are known for their scalability and error-correction capabilities. They mention that these codes possess an additional property related to the clustering of approximate codewords in the underlying classical codes.

1.8 The special case of quantum Tanner codes

The specific construction where they have confirmed this property exists is the Quantum Tanner code, introduced by Leverrier and Zémor in 2022. While they hypothesize that the property might also hold for other constructions of constant-rate and linear-distance QLDPC codes, they have not directly proven this.

The fact that this property of clustering of approximate codewords is sufficient to confirm the NLTS conjecture is a significant result. It opens up a new question, namely, whether this property is inherently satisfied by all constant-rate and linear-distance QLDPC codes. This could potentially mean that a wide class of quantum codes have a deep connection with the computational complexity of preparing low-energy states of local Hamiltonians, and further research is needed to explore this intriguing prospect.

1.9 A quick review of CSS codes

We describe a formalization of a CSS (Calderbank–Shor–Steane) quantum error-correcting code with parameters $[[n, k, d]]$. Here's a breakdown:

- The CSS code is built from two classical binary error-correcting codes C_{x} and C_{z}, with C_{z} containing the dual $C_{\mathrm{x}}^{\perp}$ of the other.

- Each of these classical codes can be defined as the kernel (null space) of a sparse binary matrix. C_{z} corresponds to the matrix H_{z} with dimensions $m_{\mathrm{z}} \times n$ and C_{x} corresponds to the matrix H_{x} with dimensions $m_{\mathrm{x}} \times n$.

- The rank of H_{z} is denoted as r_{z} and the rank of H_{x} is denoted as r_{x}. These ranks represent the number of linearly independent rows in the corresponding matrices.

- The parameter n in the quantum code corresponds to the total number of physical qubits, which is the sum of the logical information k, and the ranks r_{x} and r_{z}. This can be written as $n = k + r_{\mathrm{x}} + r_{\mathrm{z}}$.

- In a constant-rate, linear-distance code, the logical information k, distance d, and ranks r_{x} and r_{z} are all proportional to the total number of qubits, n. This means they scale linearly with the size of the code. This is expressed as $k, d, r_{\mathrm{x}}, r_{\mathrm{z}} = \Omega(n)$.

- For the specific codes considered in their work, they also have the number of rows in the parity check matrices, m_{z} and m_{x}, scaling linearly with n. This is expressed as $m_{\mathrm{z}}, m_{\mathrm{x}} = \Omega(n)$.

  Overall, we've outlined how a CSS code is constructed and characterized, and defined the parameters and conditions specific to our study, namely constant-rate, linear-distance codes.

1.10 Distance Measure and Approximate Codewords

We define some important terms related to the error detection capabilities of CSS quantum codes:

Distance Measure ($|\cdot|_S$): For any subset $S \subset \{0,1\}^n$, a distance measure $|\cdot|_S$ is defined as $|y|_S = \min_{s\in S}|y+s|$, where $|\cdot|$ denotes the Hamming weight. The Hamming weight is a measure of the number of 1's in a binary vector, and $|y+s|$ denotes the Hamming weight of the sum (performed bitwise modulo 2) of the binary vectors y and s. The distance measure $|y|_S$ therefore represents the minimum Hamming weight (i.e., the minimum number of 1's) among all the vectors that can be obtained by adding y to an element s of the set S.

Approximate Codewords (G_{z}^{δ} and G_{x}^{δ}): These are the sets of vectors which violate at most a δ-fraction of checks from the classical codes C_{z} and C_{x} respectively.

This represents the vectors that are "close" to the code C_{z} in terms of the fraction of parity checks that they fail. The set G_{x}^{δ} is defined similarly for the code C_{x}.

The paper describes the concept of approximate codewords in the context of classical codes C_{z} and C_{x}. Here's a breakdown of the key elements:

G_{z}^{δ} represents the set of binary vectors that violate at most a δ-fraction of checks from the classical code C_{z}. In other words, it consists of vectors y that satisfy the condition $|H_{\mathrm{z}}y| \leq \delta m_{\mathrm{z}}$, where H_{z} is the matrix defining the code C_{z}, and m_{z} is the number of rows in H_{z}. The matrix H_{z} is typically a parity-check matrix associated with C_{z}. The Hamming weight of $H_{\mathrm{z}}y$ refers to the number of nonzero elements in the vector resulting from the matrix-vector multiplication $H_{\mathrm{z}}y$.

The set G_{x}^{δ} is defined similarly to G_{z}^{δ} but corresponds to the classical code C_{x}. It consists of binary vectors that violate at most a δ-fraction of checks from C_{x}. The condition $|H_{\mathrm{x}}y| \leq \delta m_{\mathrm{x}}$ is satisfied, where H_{x} is the matrix defining C_{x}, and m_{x} is the number of rows in H_{x}.

In summary, the sets G_{z}^{δ} and G_{x}^{δ} represent the approximate codewords for the classical codes C_{z} and C_{x}, respectively. These sets consist of binary vectors that violate at most a specified fraction (δ) of the parity checks associated with the respective codes. The concept of approximate codewords is useful for evaluating the closeness or proximity of a given vector to a particular code based on the fraction of failed parity checks.

To put it in a condensed matter physics context, this creates a measure of "distance" between a state and a set of states and then defines sets of states that are "close" to our chosen classical codes C_{z} and C_{x}.

1.11 Clustering of Approximate Codewords

This property, known as the Clustering of Approximate Codewords, sets a crucial requirement for a CSS code to be considered for proving the No Low-Energy Trivial States (NLTS) conjecture.

1. The first part of the property pertains to vectors y that are close to the classical code C_{z} (i.e., $y \in G_{\mathrm{z}}^{\delta}$). It states that such vectors y either have small distance to the orthogonal complement of the code C_{x} ($|y|_{C_{\mathrm{x}}^{\perp}} \leq c_1\delta n$), or they have large distance to it ($|y|_{C_{\mathrm{x}}^{\perp}} \geq c_2 n$). In other words, the vectors that are close to C_{z} are either also close to $C_{\mathrm{x}}^{\perp}$, or far from it, without any intermediate distances. This shows a kind of dichotomy or 'clustering' of these vectors with respect to their distance to $C_{\mathrm{x}}^{\perp}$.

2. The second part of the property mirrors the first part, but it swaps the roles of the codes C_z and C_x. It pertains to vectors y that are close to C_x (i.e., $y \in G_\mathrm{x}^\delta$), and states that such vectors are either close to $C_\mathrm{z}^\perp$, or far from it, without any intermediate distances.

In sum, the Clustering of Approximate Codewords property states that for a CSS quantum code, vectors that are close to one of the classical codes (C_z or C_x) must be either close to or far from the orthogonal complement of the other classical code, with no in-between cases.

1.12 Tanner codes with spectral expansion

The reference they make to the "classical Tanner codes with spectral expansion" refers to a particular type of error-correcting code. Tanner codes are named after their inventor, Michael Tanner. They are constructed from smaller "component" codes using a bipartite graph called a Tanner graph. When these Tanner codes exhibit spectral expansion (i.e., the Tanner graph has good expansion properties), they have certain beneficial properties in terms of their decoding performance and error-correcting capabilities.

In the context of the Clustering of Approximate Codewords property, it seems that these Tanner codes with spectral expansion fulfill this property, as indicated in the cited theorem from the work of [AB22]. The use of these codes helped to prove the combinatorial version of the No Low-Energy Trivial States (NLTS) conjecture.

As per the reference to "Lemma 9 in the Appendix", it appears that a more generalized class of classical codes, those with small-set expanding interaction graphs, also satisfy Property 1. However, instead of using the distance $|\cdot|_{C_\mathrm{x}^\perp}$, the standard Hamming weight $|\cdot|$ is used.

Finally, they mentioned that the quantum analog of this property, which is probably related to the construction of quantum error-correcting codes based on these classical codes, is sufficient for proving the full NLTS conjecture. This suggests that these specific properties of the classical codes are crucial in extending the results to the quantum domain and thus proving the NLTS conjecture.

What is spectral expansion?

In graph theory, the expansion of a graph is a measure of how well connected the graph is. Roughly speaking, a graph with good expansion is one where every subset of vertices is adjacent to a large number of vertices outside the subset.

Spectral expansion refers to a property of Tanner codes where the associated Tanner graph exhibits good expansion characteristics. The Tanner graph is a bipartite graph representing the connectivity between the component codes in the Tanner code construction. Spectral expansion is related to the eigenvalues of the adjacency matrix of the Tanner graph. A Tanner code with spectral expansion has a Tanner graph with eigenvalues that are sufficiently spread out, leading to improved decoding performance and error-correcting capabilities.

For Tanner codes, the expansion properties of the Tanner graph impact the error correcting capabilities of the code. When the Tanner graph has good expansion properties (often quantified by a property called the "spectral gap"), the Tanner code has strong error-correcting performance. This is essentially because good expansion ensures that errors on different vertices (which correspond to bits in the code) are likely to be "visible" to a large number of check nodes, enabling the errors to be detected and corrected.

The spectral expansion property is desirable because it indicates that the Tanner graph has good connectivity and low density of short cycles, which can enhance the ability of the code to correct errors. This property is important in the decoding process and plays a role in the proof of the combinatorial NLTS conjecture.

The terminology behind "spectral"

Yes, "spectral" in this context does indeed refer to eigenvalues. The terminology comes from the field of spectral graph theory, which studies the properties of a graph in relation to the characteristic polynomial, eigenvalues, and eigenvectors of matrices associated with the graph, such as its adjacency matrix or Laplacian matrix.

The spectral gap of a graph is the difference between the largest and second largest eigenvalue of its adjacency matrix or, in some contexts, its Laplacian matrix. This quantity turns out to be closely related to the connectivity and expansion properties of the graph. In particular, graphs with a large spectral gap are well-connected and have good expansion, which is desirable in the context of error-correcting codes, as it helps with error detection and correction.

"Spectral expansion," on the other hand, is a measure of how well a graph expands, i.e., how well-connected it is, as seen through the spectrum (eigenvalues) of its associated matrices. It's often quantified using something called the "Cheeger constant" or "isoperimetric number," which measures how well-separated the graph is. A graph with high spectral expansion is one where every subset of nodes has a large number of connections to the rest of the graph, which is again desirable for the construction of good error-correcting codes.

1.13 CSS codes and local Hamiltonians

Local Hamiltonians of CSS codes

The described local Hamiltonian is naturally associated with the aforementioned quantum error-correcting codes and is based on the CSS (Calderbank–Shor–Steane) construction.

For each row w_z in H_z, which corresponds to a stabilizer term Z^{w_z} in the quantum error-correcting code, a Hamiltonian term $\frac{1}{2}(\mathbb{I} - Z^{w_z})$ is defined. Summing up these terms over all rows of H_z, the Hamiltonian $\mathbf{H}_{\mathrm{z}}$ is obtained.

An analogous process is performed for H_{x}, resulting in the Hamiltonian $\mathbf{H}_{\mathrm{x}}$. The complete Hamiltonian $\mathbf{H}$ is then obtained by adding $\mathbf{H}_{\mathrm{x}}$ and $\mathbf{H}_{\mathrm{z}}$.

The local terms in the Hamiltonian correspond to the checks of the classical codes, thus the number of local terms is $m_{\mathrm{x}} + m_{\mathrm{z}}$, which scales linearly with n, the length of the quantum code.

The ground state energy of $\mathbf{H}$ is zero, which means that the ground state is a valid code state in the associated quantum error-correcting code. This is a typical feature of quantum error-correcting codes, where the ground state of a Hamiltonian encodes the logical quantum information, and the excited states correspond to the presence of errors.

NB. The notation Z^{w_z} stands for applying the Pauli Z operator to those qubits for which the corresponding entry in the vector w_z is 1.

A brief review of stabilizer codes

A stabilizer group of a quantum code is a group of tensor products of Pauli matrices (I, X, Y, and Z). Each element of this group is called a stabilizer. A quantum state that is stabilized by all elements of this group is a codeword (or a code state) of the quantum code.

In the context of a Hamiltonian, each term corresponds to an energy level, and the total energy of a state is the sum of the energies corresponding to each term in the Hamiltonian. Now, in a stabilizer Hamiltonian, we associate each term of the Hamiltonian with a stabilizer of the quantum code.

Consider a specific stabilizer, say S. We would then have a corresponding Hamiltonian term H_S, which is designed to "penalize" states that are not stabilized by S. A common way to define this term is as $H_S = (I - S)/2$. We can verify that this operator has eigenvalues of 0 for states stabilized by S (since $S\psi = \psi$ for these states) and 1 for states not stabilized by S (since $S\psi = -\psi$ for these states).

So, the energy contribution of the H_S term for a state ψ is 0 if ψ is stabilized by S, and it's 1 if ψ is not stabilized by S.

When you sum over all these terms for all stabilizers in the stabilizer group, the resulting Hamiltonian has its lowest energy (often set to zero) for states that are stabilized by all the stabilizers, i.e., the codewords of the quantum code. All other states have a higher energy because they violate one or more stabilizers and thus get "penalized" with a higher energy.

This way, we create a Hamiltonian whose ground state corresponds to the code space of the quantum code, and whose excited states correspond to erroneous states. This is very useful in quantum error correction and quantum computation as it translates the problem of finding error-free states into a ground state problem, which is a central problem in quantum mechanics.

1.14 Open Problems in the NLTS paper [ABN]

Question 1: Clustering of Approximate Codewords (CoAC)

Does CoAC "morally" hold for all constant-rate and linear-distance quantum codes? This question relates to the generality of Property 1, which is tied to the clustering of approximate code-words in a CSS quantum code. It is interesting to explore if this property could be a characteristic of a broader class of quantum codes, specifically those with constant-rate and linear-distance.

Question 2: Connection between CoAC and small-set (co-)boundary expansion

Is there a connection between CoAC and small-set boundary and co-boundary expansion? This question hints at a potential bridge between quantum and classical complexity theory. The referenced work [HL22] involves the construction of classical Hamiltonians that are challenging to approximate. It would be intriguing to discover if a classical analogue to the NLTS property exists, and whether it has any implications on the quantum PCP conjecture. It also raises the interesting point of the relationship between local testability and the NLTS property.

Problem 3: Non-commuting Hamiltonians

Can the proof techniques be generalized to prove non-trivial lower bounds for non-commuting Hamiltonians? The present proof revolves around commuting Hamiltonians, i.e., Hamiltonians whose terms pairwise commute. Commuting Hamiltonians have unique mathematical properties and have been extensively used in the context of quantum error correction and topological quantum computing. However, in general, quantum systems are described by non-commuting Hamiltonians, and therefore it would be of great interest to generalize these techniques to such Hamiltonians. This could potentially lead to new insights in the context of quantum complexity theory and many-body quantum physics.

Exploring these questions could potentially lead to advancements in the understanding of the complexity of quantum systems and the applicability of quantum error-correcting codes.

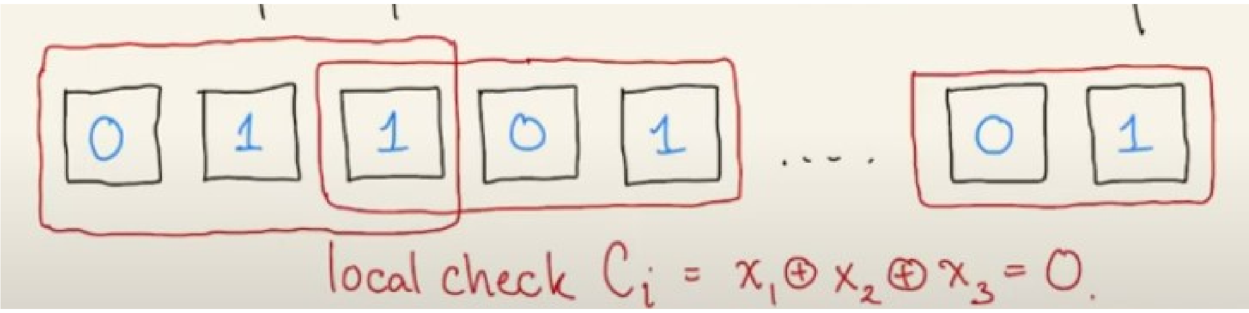
2 NLTS Hamiltonian from Good Quantum Codes

2.1 Understanding classical proofs

NP = the class of efficiently (poly(n) time) checkable proofs.

NP has complete problems such as Constraint Satisfaction Problems (CSPs).

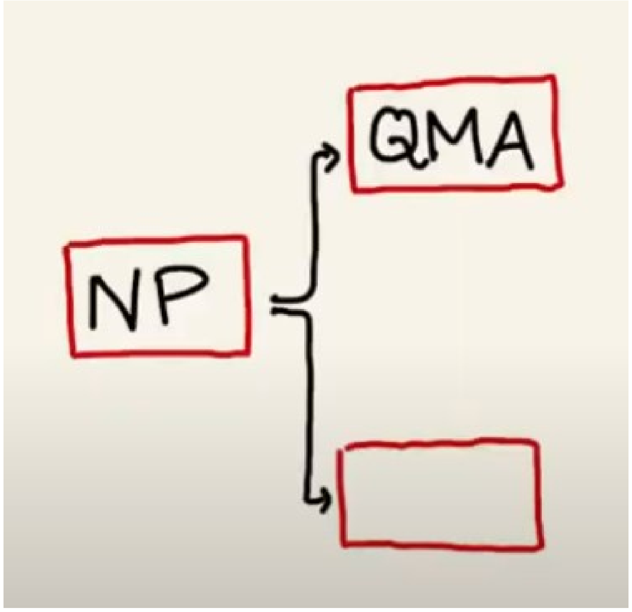
C_i is not necessarily geometrically local.



Let C_i:$0, 1^3 \rightarrow [0, 1]$ and define C: $\{0, 1\}^n \rightarrow [0, m]$ by $C(x) = \sum_{i=1}^{m} C_i(x)$. It is NP-complete to decide if

1. $\exists x$, $C(x) = 0$
2. $\forall x$, $C(x) \geq 1$

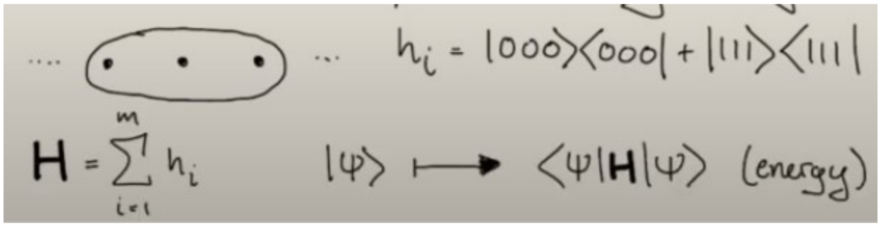
2.2 Two extensions of the notion of proof



Quantum proofs will necessitate quantum verifiers (BQP).

Kitaev showed that calculating the ground energy of local Hamiltonians serves as a complete problem for the class QMA.

h_i = local linear operator calculating energy.


$h_i = |000\rangle\langle 000| + |111\rangle\langle 111|$
$\mathbf{H} = \sum_{i=1}^{m} h_i$
$|\psi\rangle \longmapsto \langle\psi|\mathbf{H}|\psi\rangle$ (energy)

This notation is common in quantum mechanics, specifically in the context of quantum many-body systems.

Here, h_i represents a local operator that acts on part of the system to calculate the energy associated with that part. These local operators could act on individual particles (if the system is a collection of particles) or on individual sites (if the system is a lattice of sites), for instance.

$H = \sum_i h_i$ is the total Hamiltonian of the system, which is the sum of all the local Hamiltonians. The Hamiltonian is an operator that encodes the total energy of the system, including both kinetic and potential energy. It governs the time evolution of the system according to the Schrödinger equation.

$|\psi\rangle$ represents a state of the quantum system, and it could be a simple state associated with a single particle or a complicated, entangled state associated with many particles.

$\langle\psi|H|\psi\rangle$ is the expectation value of the energy of the system when it is in the state $|\psi\rangle$. This quantity gives you the average energy you would expect to measure if you prepared the system in the state $|\psi\rangle$ and then measured the energy. The way to calculate it is to take the inner product of the state $|\psi\rangle$ with the state obtained by acting with H on $|\psi\rangle$.

Frustration: In the context of physics, and in particular in the study of spin systems in quantum mechanics, the term "frustration" refers to a situation where it is impossible to simultaneously satisfy all the interactions in the system.

In simple terms, consider a system with many parts (like a system of spins) where each part interacts with its neighbors. Each interaction has a preferred configuration that would minimize its energy. The system as a whole is said to be frustrated if, due to the geometry of the interactions or the nature of the interacting parts, there is no global configuration where all the interactions achieve their individually preferred configurations simultaneously. This leads to competition between interactions, preventing the system from reaching a unique ground state (state of minimum energy) that would satisfy all the interactions simultaneously.

In the context of combinatorial optimization and constraint satisfaction problems (CSPs), a problem or system is said to be frustrated if there is no assignment of values to variables that satisfies all constraints simultaneously.

The concept of frustration is central to many areas of physics, including spin glasses, magnetism, and superconductivity, as well as computer science and mathematics, and plays a crucial role in understanding the complexity of these systems.

The **ground energy** is defined as $\lambda_{\min}(H) = \min_{|\psi\rangle}\langle\psi|\mathbf{H}|\psi\rangle$.

It is QMA-hard to decide for $b - a = 1/\text{poly}(m)$,

1. $\lambda_{\min}(\mathbf{H}) \leq a \iff \exists|\psi\rangle, \langle\psi|\mathbf{H}|\psi\rangle \leq a$
2. $\lambda_{\min}(\mathbf{H}) \geq b \iff \forall|\psi\rangle, \langle\psi|\mathbf{H}|\psi\rangle \geq b$

This excerpt introduces the concept of "ground energy" of a quantum system. The ground energy, denoted as $\lambda_{\min}(H)$, is the minimum possible energy that the system can have. In terms of the system's Hamiltonian H and possible states $|\psi\rangle$, it is the minimum expectation value $\langle\psi|H|\psi\rangle$ over all possible states.

The text then states a problem in the context of quantum complexity theory, specifically referring to the complexity class QMA (Quantum Merlin-Arthur), which is the quantum analog of the classical complexity class NP. A problem is QMA-hard if any problem in QMA can be polynomial-time reduced to it, making it among the most difficult problems in the QMA class.

The problem outlined is a decision problem about the ground energy of a quantum system:

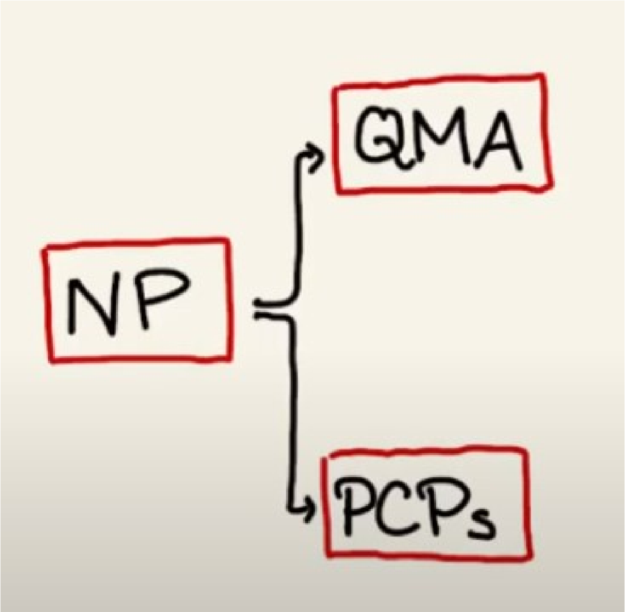
1. The first part is to decide whether the ground energy of the system is less than or equal to a certain value 'a'. In terms of the quantum states, this means there exists a state $|\psi\rangle$ such that the expectation value $\langle\psi|H|\psi\rangle$ is less than or equal to 'a'.

2. The second part is to decide whether the ground energy of the system is greater than or equal to a certain value 'b'. This means that for all states $|\psi\rangle$, the expectation value $\langle\psi|H|\psi\rangle$ is greater than or equal to 'b'.

The problem is hard in the sense that there is a small gap (1/poly(m), i.e., inverse polynomial in 'm') between 'a' and 'b'. This makes the problem of deciding whether the ground energy falls into this range a QMA-hard problem. This is often the case in quantum complexity theory, where the difficulty arises from having to decide something about a quantum system based on a small energy difference.

In NP, all proofs are morally like CSPs. In QMA, all proofs are morally like groundstates of local Hamiltonian problems. That is, ground states of local Hamiltonians are a "canonical" form for all quantum proofs.

It is now that we recall the general assumption NP $\neq$ QMA*. Therefore, not all groundstates of local Hamiltonians can be classically described in an efficiently verifiable manner.*



PCP Theorem: We usually think of proofs as step-by-step checking. The PCP theorem, however, breaks down this intuition and says that every NP problem (i.e., every proof) can be converted into a form s.t. only $O(1)$ bits need to be read to be 99% confident in its validity.

An alternate way of looking at this is, it's NP-hard to decide if

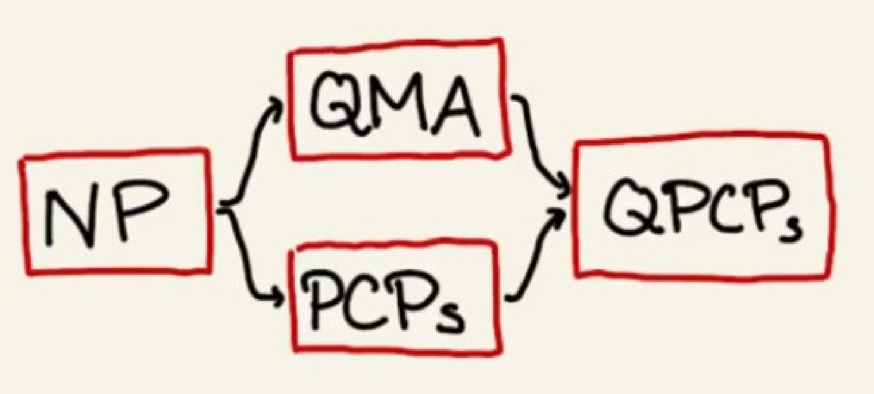
1. $\exists x$, $C(x) = 0$

2. $\forall x$, $C(x) \geq \frac{m}{2}$ (prev. 1)

where $C(x)$ is analogue of $\langle\psi|\mathbf{H}|\psi\rangle$.

Hint: How is this related to the previous formulation of PCP? Well, just think of it in terms of world 1 and world 2. If you're blindfolded and placed in one of these worlds, it would take you only a constant number of coin flips or checks to determine which world you're in.

Important consequence: A notion of noisy proofs suffices! Any x s.t. $C(x) < \frac{m}{4}$ can be probably verified with $O(1)$ queries.



Conjecture: Every QMA-problem (i.e., quantum proof) can be converted into a form s.t. only $O(1)$ qubits need to be measured.

Alternate form: For $\varepsilon > 0$, it's QMA-hard to decide

1. $\exists|\psi\rangle$ s.t. $\langle\psi|\mathbf{H}|\psi\rangle = 0$ (morally)
2. $\forall|\psi\rangle$, $\langle\psi|\mathbf{H}|\psi\rangle \geq \varepsilon m$

Similar to PCP theorem, (if quantum PCP were true) every state of energy $\leq \frac{\varepsilon}{2}m$ is a valid proof for a QPCP local Hamiltonians. Set of proofs is much larger.

2.3 An important consequence of QPCPs

A. (If NP $\neq$ QMA) quantum proofs cannot be classically described, in any efficiently checkable manner.

B. Low-energy states of QPCP local Hamiltonians are also valid proofs (since they are noisy proofs).

A and B together imply that there exist local Hamiltonians with no succinct classical descriptions for any low-energy state.

No succinct classical description is sort of a vague idea. Maybe the above statement is a bit too hard to tackle right now. Let's take a more concrete version of the above problem.

Constant depth quantum circuits are classically checkable proofs for output state.

Combining the two bold statements above, we get:

No low energy trivial states. There exist local Hamiltonians s.t. no low-energy state is the output of a constant depth quantum circuit. [Freedman-Hastings' 14]

In fact, there was a lot of evidence to point that NLTS might have been false. For instance, there was a landmark result by Brandao and Harrow which provided very good product state approximations for a large family of local state Hamiltonians. So it was suspected that maybe product state approximations, or constant depth approximations, of all local Hamiltonians existed and maybe NLTS was false.

- If it was false, then quantum PCP would have been trivially false.

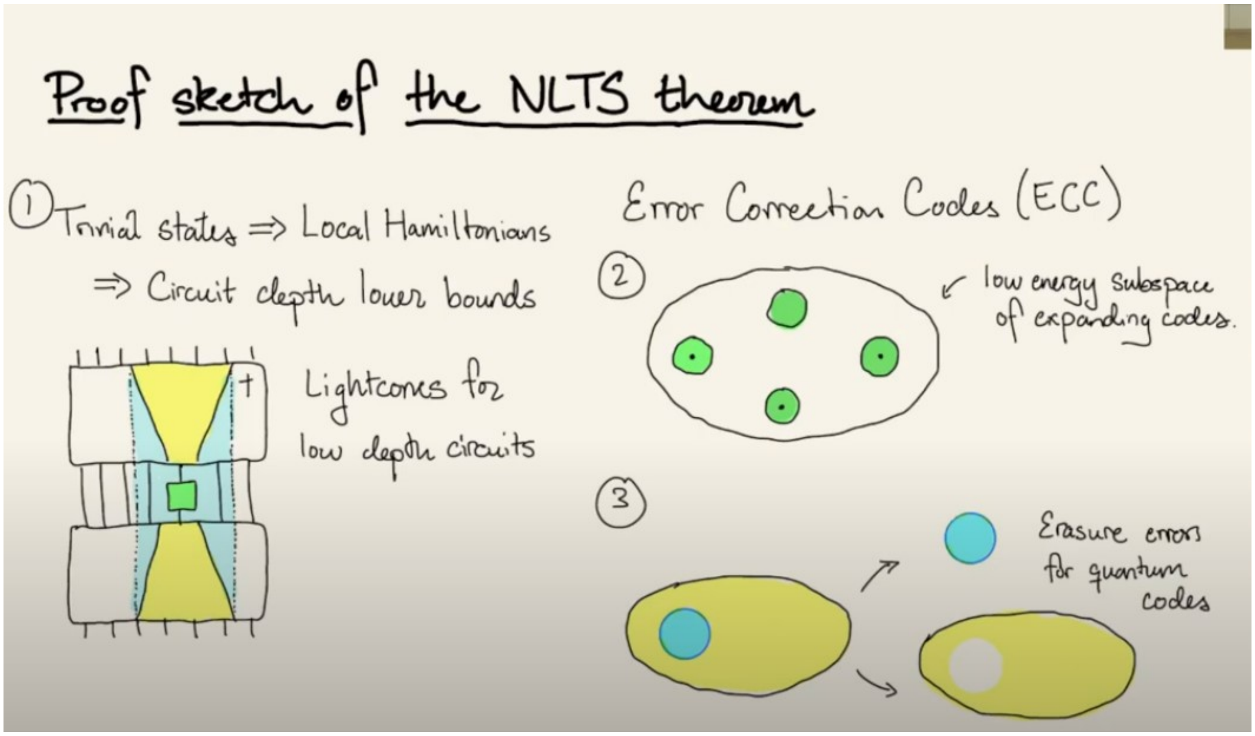
- Makes a statement about physically realizable robust entanglement. Because constant-depth quantum circuits are classically describable states. That is, what we're saying is that the low-energy subspaces of some Hamiltonians have no classical description. Or that, at constant temperatures, some constant of the total energy, the state seems to be always entangled, no matter which low-energy state you're in.

Theorem [Anurag Anshu, Niko Breuckmann, Chinmay Nirkhe'22]

Local Hamiltonians corresponding to most* linear-rate and -distance QLDPC error-correcting codes are NLTS Hamiltonians. (This includes the Leverrier-Zemor construction.)

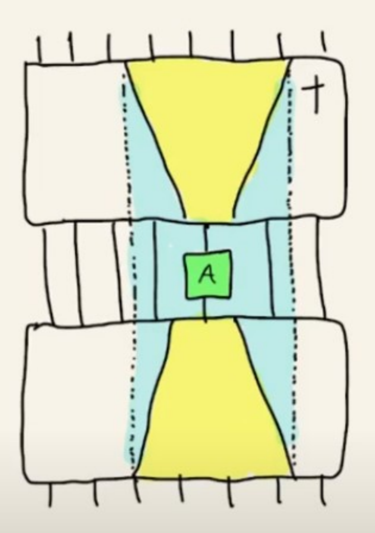
The main result they showed is that $\exists \varepsilon > 0$, and a Hamiltonian family $\mathbf{H}$, s.t. every state ψ of energy $\leq \varepsilon n$, the minimum circuit depth to generate ψ is $\Omega(\log n)$.

2.4 Proof Sketch of the NLTS Theorem

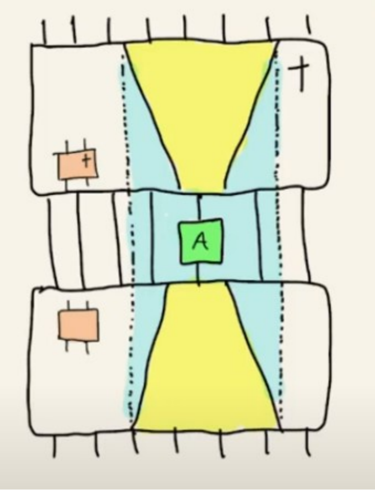

Proof sketch of the NLTS theorem
① Trivial states ⇒ Local Hamiltonians
⇒ Circuit depth lower bounds
†
Lightcones for
low depth circuits
Error Correction Codes (ECC)
②
low energy subspace
of expanding codes.
③
Erasure errors
for quantum
codes

Lightcones and quantum circuits

Low-depth states are classical witnesses for energy. We claimed earlier that this is a reasonable ansatz for energy. We will make this statement rigorous.



If A is a local operator, and $\mathcal{U}$ is a quantum circuit of depth t, then $\mathcal{U}^T A \mathcal{U}$ is a $\leq 2^t|A|$ local operator.



Notice that if there's a small orange operator outside the lightcone, then it cancels with its conjugate.

Now we want to show that if any state can be written as the output of a short circuit, then I can always calculate the energy very quickly.

Given a local Hamiltonian $\mathbf{H} = \sum_i^m h_i$ and a state $|\psi\rangle = \mathcal{U}|0^{n'}\rangle$, we can evaluate $\langle\psi|\mathbf{H}|\psi\rangle$ in classical time $2^{2^t} \cdot \text{poly}(n) = \text{poly}(n)$ when $t = O(1)$.

Let's see how we can do that.

If A is a local operator and $\mathcal{U}$ is a quantum circuit of depth t, then $\mathcal{U}A\mathcal{U}$ is a $\leq 2^t|A|$ local operator.

Given a local Hamiltonian $\mathbf{H} = \sum_i^m h_i$ and a state $|\psi\rangle = \mathcal{U}|0^{n'}\rangle$, we can evacuate $\langle\psi|\mathbf{H}|\psi\rangle$ in classical time $2^{2^t} \cdot \text{poly}(n) = \text{poly}(n)$ when $t = O(1)$.

$$\langle\psi|\mathbf{H}|\psi\rangle = \sum_i^m \langle\psi|h_i|\psi\rangle = \sum_i^m \langle 0^{n'}|\mathcal{U}^\dagger h_i \mathcal{U}|0^{n'}\rangle$$

This is a computation on $O(2^t)$ qubits. So it in fact suffices to have a circuit of depth $\log(\log n)$ – which can be witnesses.

"**Low depth circuits are classical witnesses for energy**"

Trivial states $\implies$ local Hamiltonians

The state $|0^{n'}\rangle$ is the unique solution to a very simple local Hamiltonian.

$\mathbf{H}_O = \sum_{i=1}^{n'} |1\rangle\langle 1|_i$ – qubit-wise projectors enforcing qubits equal $|0\rangle$.

$\mathbf{H}_0$ is commuting and has a spectrum of $0, 1, 2, \ldots, n'$ with eigenvectors $|x\rangle$ of eigenvalue $|x|$.

Let $\mathbf{H}_{\mathcal{U}} = \mathcal{U}^\dagger \mathbf{H} \mathcal{U}$ for depth t circuit $\mathcal{U}$.

$\mathbf{H}_{\mathcal{U}}$ is commuting and has a spectrum of $0, 1, 2, \ldots, n'$ with eigenvectors $\mathcal{U}|x\rangle$ of eigenvalue $|x|$.

And $\mathbf{H}_{\mathcal{U}}$ is a 2^t-local Hamiltonian.

Local indistinguishability

Two states $|\psi\rangle$ and $|\psi'\rangle$ are d-locally indistinguishable if for every region S of size $\leq d$,

$$\psi_{-S} = \psi'_{-S}$$

Ex. The states $|\pm\rangle = \frac{|0^n\rangle \pm |1^n\rangle}{\sqrt{2}}$ are locally $(n-1)$ indistinguishable.

Any strict reduced density matrix equals

$$(\pm)_S = \frac{|0\rangle\langle 0|^{n-|s|} + |1\rangle\langle 1|^{n-|s|}}{2}$$

Local indistinguishability $\implies$ Ckt depth lower bounds

Two states $|\psi\rangle$ and $|\psi'\rangle$ are d-locally indistinguishable if for every region S of size $\leq d$,

$$\psi_{-S} = \psi'_{-S}$$

Lemma. If $|\psi\rangle$ and $|\psi'\rangle$ are d-locally indistinguishable, then if $|\psi\rangle = \mathcal{U}|0^n\rangle$ for $\mathcal{U}$ of depth t, then $2^t \geq d \implies t \geq \log d$.

Proof. If $|\psi\rangle$ and $|\psi'\rangle$ are d-locally indistinguishable, then if $|\psi\rangle = \mathcal{U}|0^n\rangle$ for $\mathcal{U}$ of depth t, then $2^t \geq d \implies t \geq \log d$.

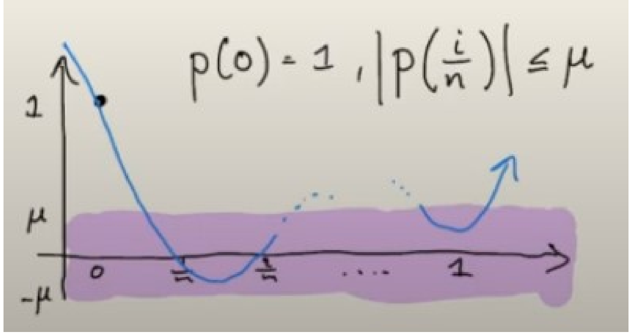
$$\langle \psi' | \mathbf{H}_{\mathcal{U}} | \psi' \rangle = \sum_i \langle \psi' | h_i | \psi' \rangle = \langle \psi | h_i | \psi \rangle = \langle \psi | \mathbf{H}_{\mathcal{U}} | \psi \rangle = 0$$

But groundstate $|\psi\rangle$ is unique!, i.e., $|\psi\rangle = |\psi'\rangle$, is a contradiction!

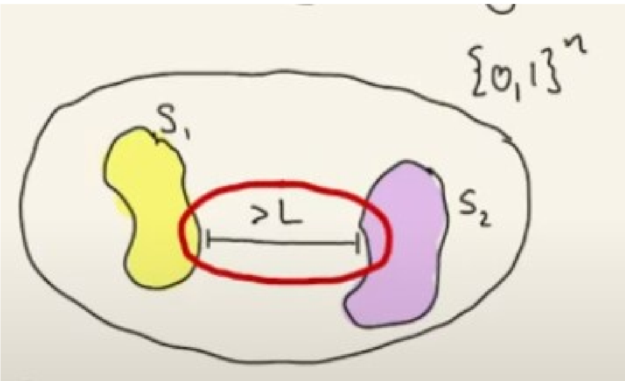
Caution. Since, spectral gap of $\mathbf{H}_{\mathcal{U}}$ is 1, this argument is only robust to perurbations of $O(\frac{1}{n})$.

Using mathematics from Chebychev polynomials, we can make l.b. robust.

Robust local indistinguishability



$\Pi := \mathbb{I} - \mathbf{H}_{\mathcal{U}} \implies ||\Pi - |\psi\rangle\langle|\psi\rangle||_\infty \leq 1 - \frac{1}{n}$ which is a weak approximate projector.
$\exists p$: $\mathbb{R} \to \mathbb{R}$ of deg$O_\mu(\sqrt{n})$ s.t. $||p(\mathbf{H}_{\mathcal{U}}) - |\psi\rangle\langle\psi||| \leq \mu$.
$1 - p$ is the Chebyshev polynomial approximation of the OR function.
$p(\mathbf{H}_{\mathcal{U}})$ is a local Hamiltonian of locality $L = O(2^t.\sqrt{n})$.



Now, let D be a distribution on $\{0,1\}^n$ formed by measuring $|\psi\rangle$.

Assume $D(S_1) > \mu$ and $D(S_2) > \mu$. Let Π_{S_1} and Π_{S_2} be the projector onto the sets S_1 and S_2 respectively.

1.

$$||\Pi_{S_1}|\psi\rangle\langle\Pi_{S_2}||_\infty > \mu$$

Assume that the Hamming distance between S_1 and S_2 is $> L$. Then we get the following consequence.

2.

$$||\Pi_{S_1}|p(\mathbf{H}_{\mathcal{U}})\Pi_{S_2}||_\infty = 0$$

due to the locality of $\mathbf{H}_{\mathcal{U}}$ is small.

Thm. Any distribution D s.t. $D(S_1), D(S_2) > \mu$ cannot be generated by a quantum circuit of depth $\leq \Omega(\log \frac{L^2\mu}{n})$.

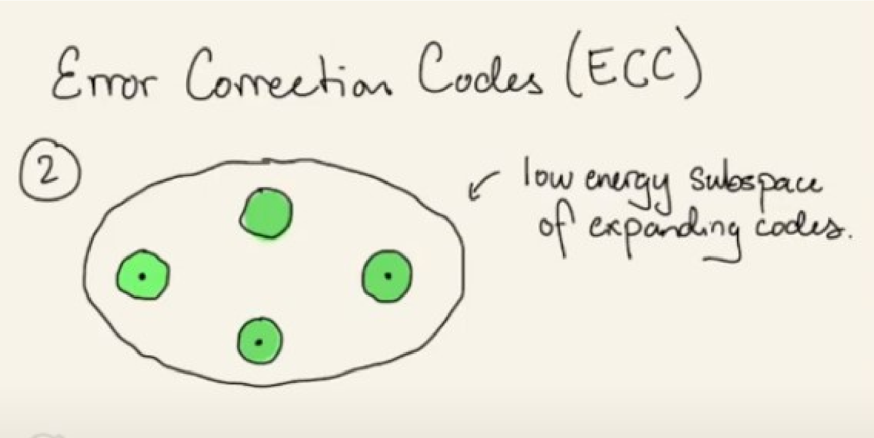
If we look at equations (1), (2) and $||p(\mathbf{H}_{\mathcal{U}}) - |\psi\rangle\langle\psi||| \leq \mu$ we realize that we have introduced a contradiction somewhere. The only possible point of contradiction could have been that $L = O(2^t\sqrt{n})$ must be greater than L.

Thus, **Thm**. Any distribution D s.t. $D(S_1), D(S_2) > \mu$ cannot be generated by a quantum circuit of depth $\leq \Omega(\log(L^2\mu/n))$.

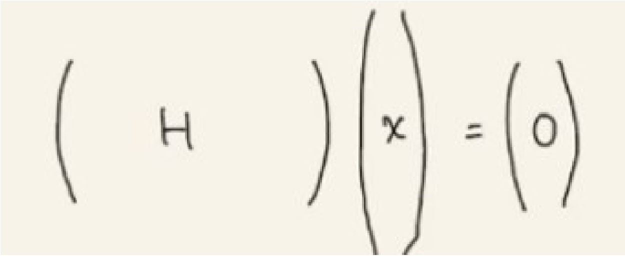
Cor. Any state $|\psi\rangle$ whose measurement distribution is D also has the same lower bound.

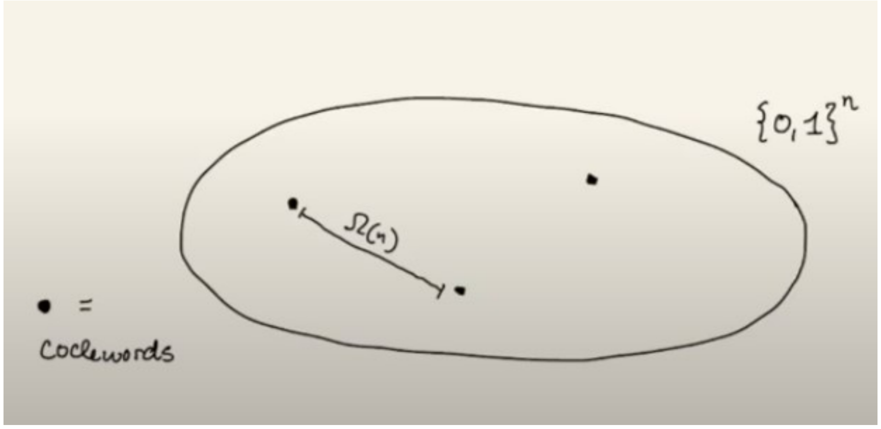
We notice that L needs to be at least $\sqrt{n}$ for this lower bound to be non-trivial. If $L \geq \omega(\sqrt{n})$ and $\mu \geq \Omega(1)$, call D a "well-spread" distribution. Well-spread distribution is a signature of quantum depth.

Error Correcting Codes


Error Correction Codes (ECC)
2
low energy subspace of expanding codes.

Expanding codes and Tanner codes
A linear code $\subseteq \{0,1\}^n$ can be expressed as $\ker H$ for $H \in \mathbb{F}_2^{m \times n}$.

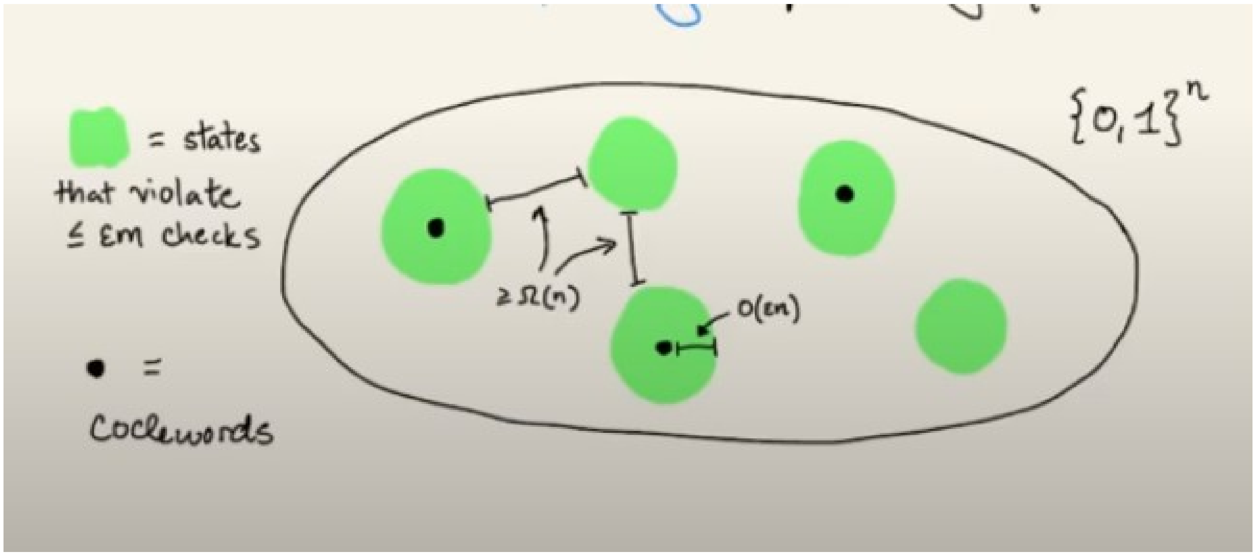




when H is the adjacency matrix of a small-set expanding bipartite graph. The distance between the codewords is kind of large.

Now what happens when I start to plot out the states that violate only a small fraction of the terms?

Starting from the codewords, I should be able to bubble out a little bit – if I violate only a few checks. This code is LDPC, so changing a few bits, does not drastically change the number of checks you violate. So you should definitely expect some amount of bubbling around the codewords.



What's sort of surprising is that you'll see some phantom clusters appear where's there no bubbling out at all.

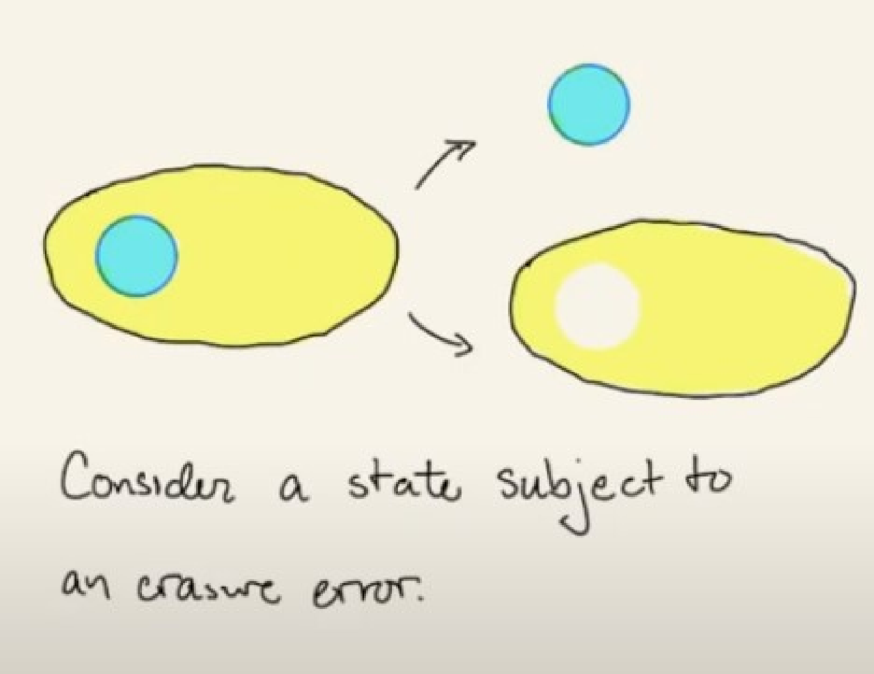
Say, you take the matrix H and deleted a row. If H was a good code, then deleting a row shouldn't change the distance too much. But what it will do – you delete a row from H, it will double the number of solutions. Those will form the centers of these phantom bubble clusters that appear. If I continue this, and delete some ϵ fraction, I will still get this clustering to appear. You can make all this rigorous by looking at small-set expansion slightly.

But let's just pause and realize that these clusters sort of look like well-spread distributions. Yeah, in a well-spread distribution we drew last time, there were only two regions S_1 and S_2. But if you take the union of some of these clusters and call them S_1 and the union of the rest of the clusters and call them S_2, you seem to have a well-spaced distribution. That is, the low energy space of a code is a great support for a code that we hope to prove is well-spread.

You might ask this point: Why aren't you just considering the classical code? Well, that's a great idea. If you consider a classical code and a Hamiltonian that corresponds to it, indeed it will be supported on one of these kinds of distributions. But you wouldn't get the key property required for well-preparedness, i.e., constant mass on both halves. If you consider a classical Hamiltonians, there will be classical solutions which are just distributions supported on singletons.

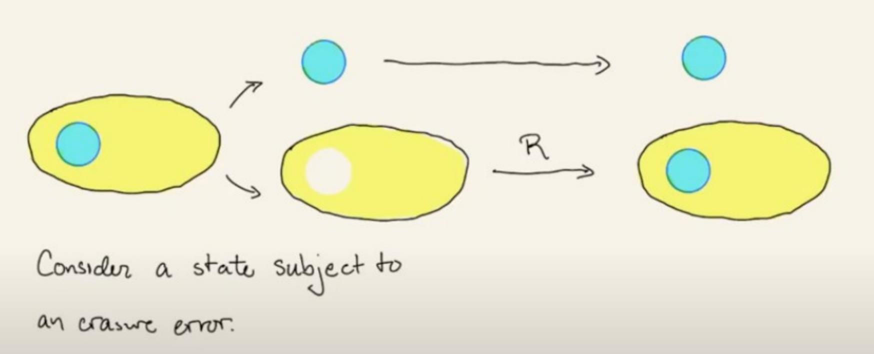
For the required property, we will now go to the third property of erasure errors.

Erasure Errors



Consider a state subject to an erasure error.

If you have quantum error correcting code and you have an erasure, this naturally induces local indistinguishably.



Consider a state subject to an erasure error.

Though experiment. Imagine that you're walking down the street with your error-correcting code and some of your qubits fall out. You say, that sucks – you apply your recovery map and reconstruct it again. But say someone were to come around and pickup those missing qubits – now your seemingly have a problem. But then you cloned the state of those missing qubits.

Is there a simple resolution to this conundrum? Whatever was encoded in that blue region was an invariant of the code – it contained no information of about the original state, this violates the no-cloning theorem.

Another way of saying this is, no matter what code you started off with, the reduced density matrix on that small blue region is an invariant. In other words:

Erasure error correction imply local indistinguishability for codes.

This immediately implies that *exact codewords of distance d require circuits of depth $\geq \Omega(\log d)$ to generate.* We immediately have that exact codewords have some $\log n$ circuit lower bounds because we know of codes of polynomial distance.

Furthermore, error-correcting codes that are LDPC naturally have a local Hamiltonian, one that applies every local check.

Real Question: How do we prove circuit depth lower bounds for the low-energy subspace of these code Hamiltonians?

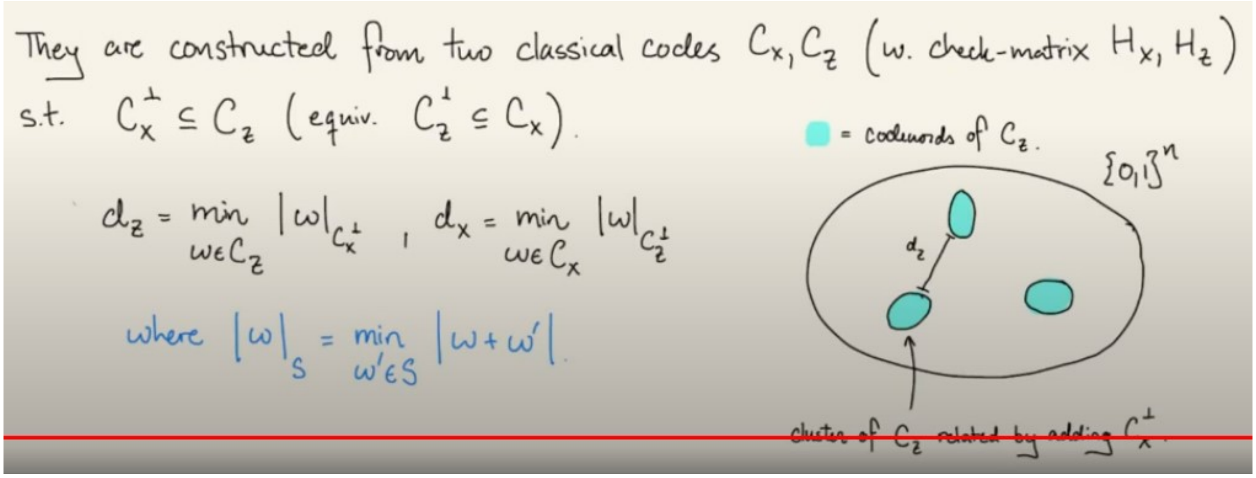
We will need to think about something more specific now.

Optimal-parameter CSS codes

There is a class of quantum codes called Calderbank-Shor-Steane codes that correct for X-type (bit-flip) and Z-type (phase-flip) errors separately.

They are constructed from two classical codes C_X, C_Z (with check matrix H_X, H_Z) s.t. $C_X^\perp \subseteq C_Z$ (equiv. $C_Z^\perp \subseteq C_X$).

What really matters to us is that CSS codes have a picture that looks very similar to how classical codes look.



They are constructed from two classical codes C_X, C_Z (w. check-matrix H_X, H_Z) s.t. $C_X^\perp \subseteq C_Z$ (equiv. $C_Z^\perp \subseteq C_X$).

$d_Z = \min_{w \in C_Z} |w|_{C_X^\perp}$, $d_X = \min_{w \in C_X} |w|_{C_Z^\perp}$

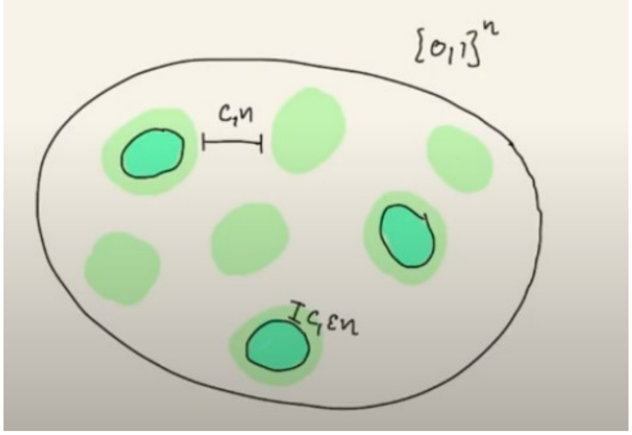
where $|w|_S = \min_{w' \in S} |w + w'|$.

There's always two "dual" pictures, one for X and one for Z.

Expanding CSS Codes

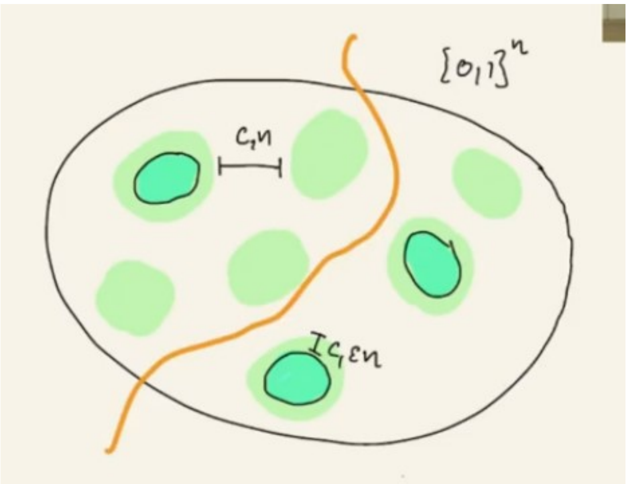
Similar to the classical example, we consider codes that have the property that if $|H_Z y| \leq \varepsilon m$ then either

1. $|y|_{C_X^\perp} \leq C_1 \varepsilon n$ or
2. $|y|_{C_X^\perp} \geq C_2 n$



We start off with the original bubbles and the bubbling scales like ε like in the bottom of the picture. But we will also have these phantom bubbles with phantom clusters that appear.

So we get that if we have an expanding CSS code, and, if we consider a $\varepsilon/200$ low-energy state of the code's local Hamiltonian, measuring in the Z basis yields a distribution 99% supported on the green patches. Likewise, if I measure in the X basis I will be well supported on the bubbles in the Z-picture. This follows because CSS codes have roughly 50% X-checks and 50% Z-checks.



All that's left to prove is that the distribution when I measure has mass on two separate regions. There must be some way of partitioning the X-picture or the Z-picture so that I have some constant mass on both sides.

All we need to argue is that the distribution is not 99% concentrated on any 1 cluster $\implies$ the distribution is well-spread ($\mu = \frac{1}{400}$) $\implies$ circuit depth lower bound.

What we're going to be able to show is that the distribution cannot be simultaneously clustered in both the X-picture and the Z-picture.

Uncertainity principle. For sets $S, T \subseteq \{0,1\}^n$, any state ψ with distributions D_X, D_Z

$$D_X(T) \leq 2\sqrt{1 - D_Z(S)} + \sqrt{\frac{|S|.|T|}{2^n}}$$

Assume D_Z is 99% concentrated on some Z-cluster S. Then for any X-cluster T, $D_X(T) \leq 0.99$ $\implies$ Either D_X or D_Z is well-spread.

The way we do this is just to compute the sizes of these clusters. Remember that these clusters are formed by starting with a codeword and bubbling out a radius of ϵ. So that's exactly what we're gonna get. The size $|S|$ is bounded by this combinatorial term that scales in ϵ.

$$|S| \leq \binom{n}{O(\varepsilon n)}.2^{r_x} \leq 2^{r_x + O(\sqrt{epsilon}n)}$$

The solution term $\binom{n}{O(\varepsilon n)}$ represents violate checks and 2^{r_x} comes from the defintion of $C_X^{\perp}$. Similarly,

$$|T| \leq 2^{r_z + O(\varepsilon n)}$$

Putting this in the uncertainty principle,

$$D_X(T) \leq 2\sqrt{\frac{1}{100}} + 2^{r_x+r_z+O(\sqrt{\varepsilon})-n} = \frac{1}{S} + 2^{-k+O(\sqrt{\varepsilon}n)}$$

The code rate appears in this. Because when the code is degenerate, there are many local indistinguishabilty arguments that you're applying in tandem.

As long as, $\varepsilon < O(k^2/n^2)$, then $D_X(T) < 0.99$. So we're not too supported on any one cluster.

Conclusion of the proof

CSS codes of linear-distance and linear-rate which are expanding are NLTS.

The [Leverrier-Zemor'21] construction can be shown by small modification of the distance bound proof to satisfy these conditions.

In progress: All linear-rate and linear-distance codes are NLTS.

What's next after NLTS?

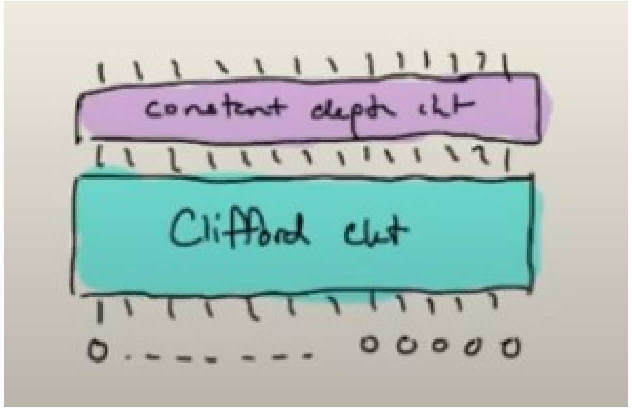
First, NLTS is a necessary consequence of QPCP that isolated the problem of robust entanglement from the computational question.

Next step: Introduce computation, find NLTS Hamiltonians that capture NP (or MA) computations.

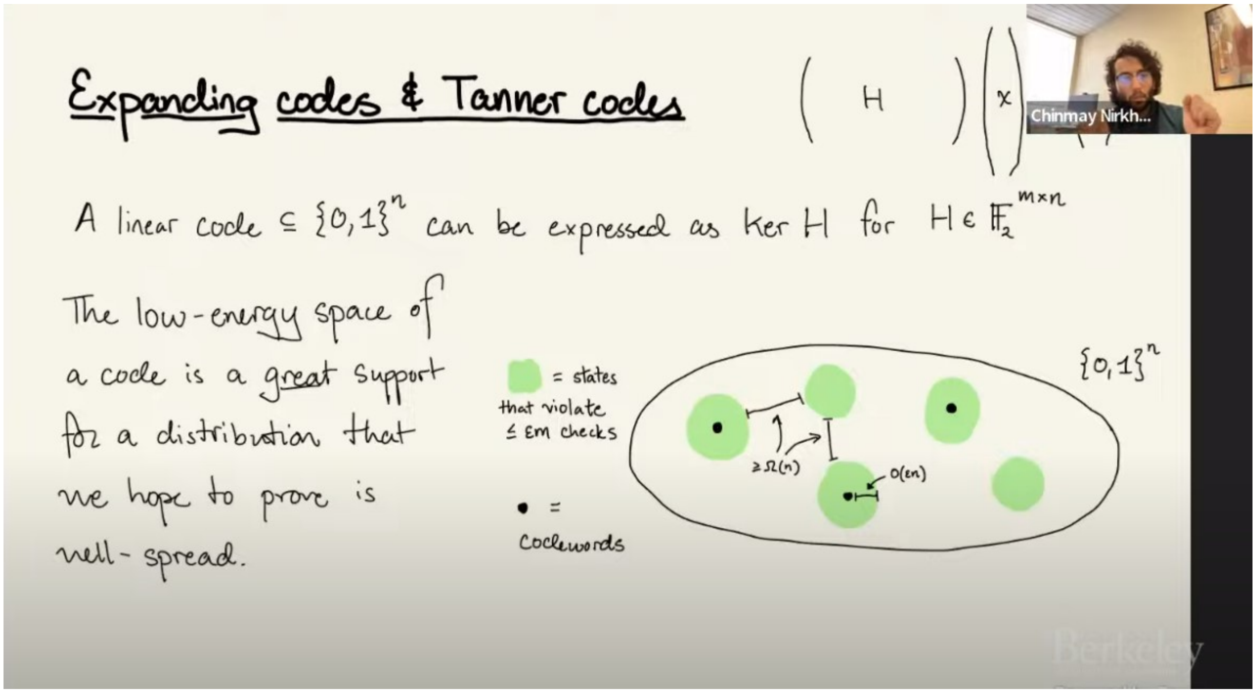
Secondly, constant depth quantum circuits are just one of many possible NP proofs of the ground-energy. (Really, quantum PCP describes that all such classical descriptions are insufficient. Constant depth quantum circuits are just one kind.)

Other examples include stabilizer circuits, some efficiently contractible tensors, etc. or samplable-queryable states ([Gharibian-Le Gall '21] MA witness). We should be able to prove lower bounds against all of these descriptions.

For one, we need to prove lower bounds for the following ansatz:



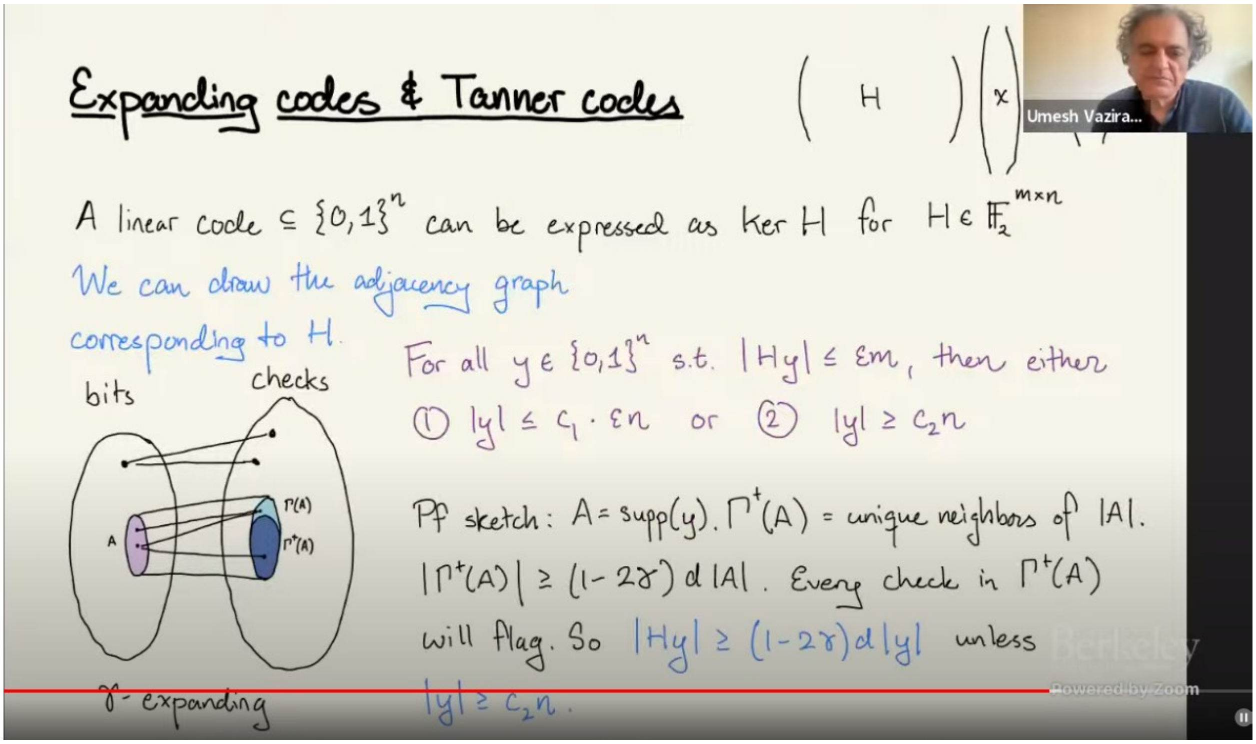
2.5 Simons Institute Notes on Expanding and Tanner Codes (Nirkhe)



Expanding codes & Tanner codes

A linear code $\subseteq \{0,1\}^n$ can be expressed as $\ker H$ for $H \in \mathbb{F}_2^{m\times n}$

The low-energy space of a code is a great support for a distribution that we hope to prove is well-spread.



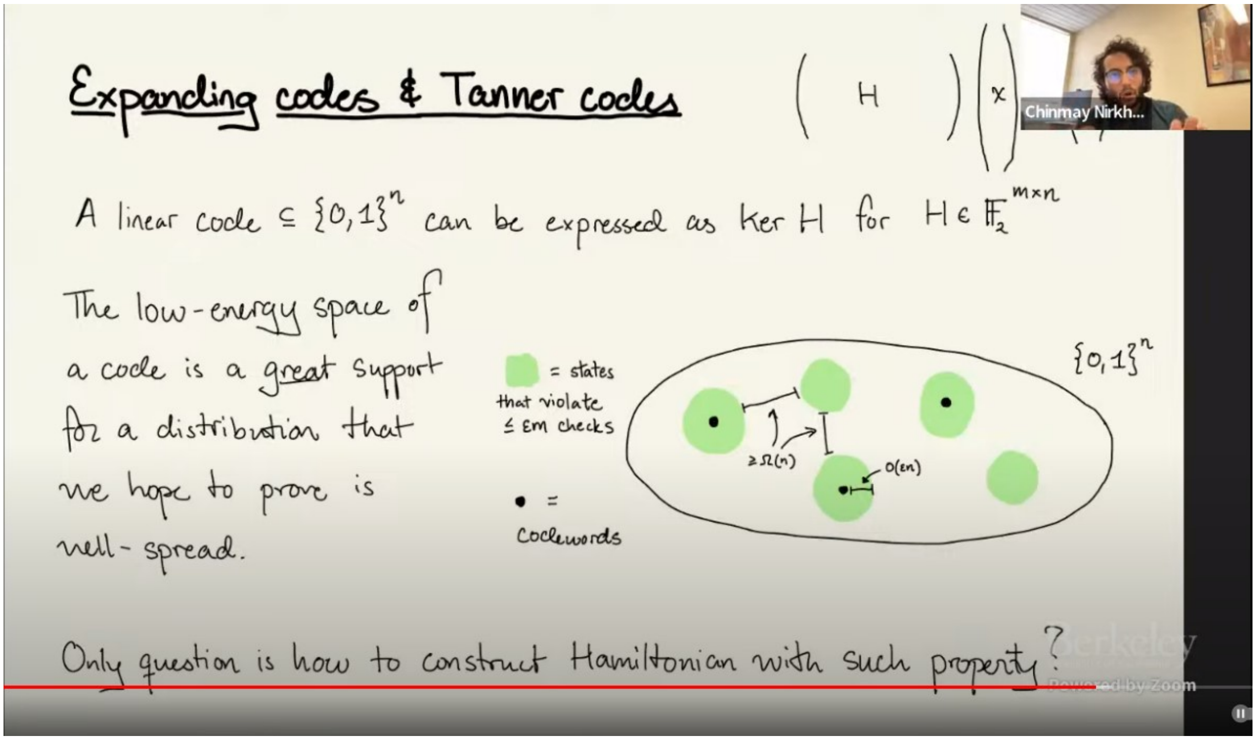
Expanding codes & Tanner codes

A linear code $\subseteq \{0,1\}^n$ can be expressed as $\ker H$ for $H \in \mathbb{F}_2^{m\times n}$

We can draw the adjacency graph corresponding to H.

For all $y \in \{0,1\}^n$ s.t. $|Hy| \leq \varepsilon m$, then either ① $|y| \leq c_1 \cdot \varepsilon n$ or ② $|y| \geq c_2 n$

Pf sketch: $A = \mathrm{supp}(y)$. $\Gamma^+(A)$ = unique neighbors of $|A|$. $|\Gamma^+(A)| \geq (1-2\gamma)\, d\, |A|$. Every check in $\Gamma^+(A)$ will flag. So $|Hy| \geq (1-2\gamma) d |y|$ unless $|y| \geq c_2 n$.



Expanding codes & Tanner codes

A linear code $\subseteq \{0,1\}^n$ can be expressed as $\ker H$ for $H \in \mathbb{F}_2^{m \times n}$

The low-energy space of a code is a great support for a distribution that we hope to prove is well-spread.

Only question is how to construct Hamiltonian with such property?

3 SoS Lower Bounds and the SS-HDX Recipe

3.1 Sum-of-Squares (SoS) and CSPs

Sum-of-Squares (SoS) semi-definite programming (SDP) is a method used to approximate solutions for problems called constraint satisfaction problems (CSPs). These problems require finding a solution that meets a certain number of constraints or conditions. However, it's difficult to determine the structure of problems that are challenging for this SoS method.

[HL] is discussing the breakthrough that there's now an explicit group (or "family") of highly unsatisfiable CSPs that the SoS method cannot solve. The breakthrough is important because before this, the most effective method to find hard instances for SoS was pretty much brute force search, which is a time-consuming and inefficient method.

The main result or theorem (Theorem 1.1) of this paper claims that there are specific values and an infinite group of 3-XOR problems (a type of CSP) that have two key characteristics:

1. No assignment can satisfy more than a certain fraction of the constraints. This fraction is represented by (1 - μ_1) where μ_1 is some constant between 0 and 1.

2. No problem in this group can be refuted by $\mu_2 n$ levels of the SoS SDP relaxation. Here, μ_2 is also a constant between 0 and 1, and n represents the size or complexity of the problem.

In simpler words, this theorem says that there is a set of very difficult 3-XOR problems that can't be solved by the SoS method, no matter how much you increase the complexity or the levels of the method.

Theorem 1.1 also provides the first example of an approximation problem with short witnesses of unsatisfiability that the Sum-of-Squares proof system cannot handle. In other words, it gives a problem which the SoS system fails to solve. This proves that the SoS system isn't complete or perfect in its ability to solve all problems, which is a significant discovery in the field.

3.2 What is the SoS hierarchy?

The Sum-of-Squares (SoS) semi-definite programming (SDP) hierarchy is an advanced computational tool that is often used to approximate solutions for constraint satisfaction problems (CSPs). CSPs are a type of problem in theoretical computer 'science that involve finding a solution that satisfies a series of constraints or conditions.

Despite the SoS SDP hierarchy's power and extensive study, we know very little about the types of CSPs that are difficult for it to handle. While it has been known for a while that random instances of CSPs are often challenging for SoS, there haven't been many significant advances in constructing explicitly hard instances for SoS, with the best methods generally being equivalent to a simple brute force search.

[HL] leverages recent developments in locally testable codes and quantum low-density parity-check (qLDPC) codes. With the help of these tools, they claim to have created the first explicit family (or group) of CSPs that are very difficult to satisfy (unsatisfiable) and cannot be solved by using a large number of rounds of SoS, specifically Omega(n) rounds. In complexity theory, the notation "Omega(n)" usually refers to lower bound on the growth rate of a function, indicating that a large, but unspecified, number of rounds of SoS cannot refute (disprove) these CSPs.

3.3 Theorem 1.1 (Main Result: Explicit 3-XOR Instances Hard for SoS)

Theorem 1.1 introduces a significant result related to the complexity of certain problems in the context of the Sum-of-Squares (SoS) semi-definite programming (SDP) hierarchy. This result concerns 3-XOR instances, which are a type of constraint satisfaction problem that involve equations with three variables, all linked by XOR (exclusive or) operations.

The theorem states that there exist constants (μ_1 and μ_2), both between 0 and 1, and an infinite set of 3-XOR instances that can be built in deterministic polynomial time. The following conditions apply to these instances:

1. No possible assignment of values to the variables in a given problem can satisfy more than a $(1 - \mu_1)$ fraction of the constraints. This means that no matter how you try to solve these problems, you will always leave at least μ_1 fraction of the constraints unsatisfied.

2. No instance can be disproven (refuted) by using $\mu_2 n$ rounds of the corresponding Sum-of-Squares SDP relaxation. Here, "n" refers to the size of the problem (for example, the number of variables or constraints), and "relaxation" is a technique often used in optimization problems where a harder problem is replaced by an easier one that provides an upper or lower bound. This point implies that these instances are challenging for the SoS algorithm, as even a substantial number of rounds of the SoS SDP relaxation fail to refute the instances.

3.4 The Integrality Gap

While Theorem 1.1 reveals an 'integrality gap' - the difference between the optimal value of the integer problem and its relaxation - of 1 versus $(1 - \mu_1)$, this means that the instances can satisfy $(1 - \mu_1)$ of the constraints but they appear fully satisfiable to the Sum-of-Squares (SoS) algorithm. This gap can be amplified to $(1 - \epsilon)$ versus $((1/2) + \epsilon)$ for any $\epsilon > 0$ when combined with standard PCP (Probabilistically Checkable Proof)-like reductions in the SoS hierarchy. This essentially matches the difficulty of random 3-XOR instances, allowing for some degree of imperfection in the solutions.

3.5 Explicit family of 3-XORs

It's important to note that Theorem 1.1 introduces the first explicit family of Constraint Satisfaction Problems (CSPs) that outperform more than $O(\log(n))$ levels of the SoS hierarchy. This can be achieved through either unique neighbor expanders, which are a certain type of graph with special properties, or simply by brute force search, although the latter may come with some lower-order factors.

While there were known examples of explicit constructions that go against $\Omega(n)$ rounds of SoS in the field of proof complexity (e.g., Tseitin formulas, knapsack), these examples do not lead to **inapproximability** because their **satisfiability** is not bounded away from 1, meaning they can be fully or almost fully satisfied. The introduced 3-XOR instances, however, exhibit a bounded away from 1 satisfiability, thus presenting a harder case for the SoS algorithm.

3.6 Inapproximability

In many cases, we want to understand the limits of approximation algorithms, that is, we want to show that it's not possible to approximate the optimal solution beyond a certain ratio in polynomial time (unless P=NP). One of the ways this is done is by showing that a problem is hard to approximate within some ratio for a powerful algorithmic framework like SoS. If we can show that even the SoS hierarchy can't approximate the solution beyond a certain point, it provides evidence that no polynomial time algorithm can (under standard complexity assumptions).

For instance, if you have a problem and you can show that after a certain number of rounds in the SoS hierarchy, you can't find a solution that approximates the optimal solution beyond a certain ratio, then this provides a lower bound on the inapproximability of that problem. This means that there is no polynomial time algorithm that can guarantee a better approximation ratio (unless P=NP).

So in summary, the SoS hierarchy is an algorithmic tool that we use to solve problems, and inapproximability is a concept that describes how well we can solve problems. By using SoS as a benchmark, we can gain insights into the inapproximability of various problems.

3.7 Satisfiability

The satisfiability of a constraint satisfaction problem (CSP) like the 3-XOR problem refers to the fraction of constraints that can be simultaneously satisfied by the best possible assignment of values to the variables.

For random 3-XOR instances, the satisfiability is not known exactly but is understood to be very high under random assignment. A random 3-XOR problem is generated by picking each constraint (a XOR b XOR c = 0 or 1) uniformly at random from among all possible constraints on three variables.

3.8 Satisfiability

For a large random 3-XOR problem, a random assignment of the variables will satisfy, on average, about half the constraints. However, there exist algorithms that can find assignments satisfying significantly more than half the constraints in polynomial time.

The fact that it's challenging to determine the exact satisfiability or find an assignment that satisfies all constraints is part of what makes random 3-XOR a difficult problem and an interesting benchmark for studying the limits of approximation algorithms and the complexity of solving CSPs.

3.9 What is Theorem 1.1 doing for us?

At a high level, Theorem 1.1 provides the first example of an approximation problem with short proofs (or "witnesses") of unsatisfiability that the Sum-of-Squares (SoS) proof system cannot handle. This conclusion negatively settles the question of whether SoS is complete, meaning capable of handling all problems of this nature, in this context.

Additionally, it's important to note that the specific choice of a 3-XOR problem is not particularly special or essential for this result. As pointed out by earlier research (specifically [DFHT20]), which demonstrated a similar outcome for $O(\sqrt{\log(n)})$ levels of SoS, Theorem 1.1's approach can be used to construct hard instances across many types of Constraint Satisfaction Problems (CSPs). This is achievable through standard reduction techniques.

These hard instances can include those with the largest possible difference (or "integrality gaps") between the best possible solutions for the exact and relaxed versions of CSPs. Specifically, this is the case for CSPs with predicates that are resistant to approximations, based on pairwise independent subgroups. These predicates are mathematical expressions that, when true, satisfy the constraints of the CSP.

The "short witnesses of unsatisfiability" mentioned here likely refer to a concise evidence or proof that a given problem instance cannot be fully satisfied. The theorem shows that, even when such short witnesses exist, they cannot always be identified by the SoS proof system. This resolves an open question about the completeness of SoS for problems of this type, showing that SoS is not always able to recognize unsatisfiable instances, even when the proof of unsatisfiability is relatively simple.

The reference to “3-XOR” indicates that this particular constraint satisfaction problem (CSP) served as a specific example for demonstrating this limitation of SoS. However, the implications of the theorem extend beyond just the 3-XOR problem.

As observed in [DFHT20], Theorem 1.1 can be used to construct hard instances of many types of CSPs using standard reduction techniques. This includes instances of CSPs with “approximation-resistant predicates based on pairwise independent subgroups”, which are particularly difficult problems for approximation algorithms.

The “optimal integrality gaps” phrase refers to a measure of the difference between the optimal solutions of the integer programming and its continuous (or ’relaxed’) counterpart. An instance with an “optimal integrality gap” is one where this difference is as large as possible, making it a hard instance for approximation algorithms.

This theorem has far-reaching implications for our understanding of the limits of approximation algorithms and the SoS proof system in particular. It provides both a new insight into the capabilities of SoS and a method for constructing hard instances of a variety of constraint satisfaction problems.

3.10 Approximation-resistant predicates based on pairwise independent subgroups

In this context, “approximation-resistant predicates based on pairwise independent subgroups” refers to a specific type of function or condition used in constraint satisfaction problems (CSPs).

1. A **predicate** in this context refers to a boolean-valued function or condition that is applied to a set of variables in a CSP. For example, in a 3-SAT problem, a predicate could be a clause like (x OR NOT y OR z), which takes the values of x, y, and z and returns either true or false.

2. **Approximation-resistant** means that it is hard to find an approximation to the maximum number of predicates that can be satisfied simultaneously. In other words, even approximation algorithms cannot significantly outperform simply picking a solution at random.

3. In the context of CSPs and predicates, **pairwise independent subgroups** likely means that the set of satisfying assignments for the predicate forms a subgroup (i.e., subfamily) and any two elements picked from this subgroup are independent.

Taken together, “approximation-resistant predicates based on pairwise independent subgroups” likely refers to predicates for which the set of satisfying assignments forms a pairwise independent subgroup, and finding an approximation to the maximum number of these predicates that can be satisfied simultaneously is a hard problem. The specifics of how these predicates are constructed and used would depend on the problem and the details of the underlying mathematical framework.

3.11 Small-set High Dimensional Expanders (SS-HDX)

Theorem 1.1 is based on a newly emergent concept of high dimensional expansion (HDX), a budding field in computer science and mathematics that has already witnessed numerous significant results in areas such as coding theory, approximate sampling, approximation algorithms, analysis of boolean functions, agreement testing, and recently, Sum-of-Squares lower bounds.

Most of these works consider notions of expansion on hypergraphs, which are often called simplicial complexes in this context. However, the authors of this paper draw inspiration from recent advances in Locally Testable Codes (LTCs) and quantum codes and consider expansion on a more general class of mathematical structures known as chain complexes.

Here, the symbol “X” represents a chain complex, which is a sequence of vector spaces or modules connected by homomorphisms. The vector spaces $\mathbb{F}_2^{X(0)}$, $\mathbb{F}_2^{X(1)}$, and $\mathbb{F}_2^{X(2)}$ represent different “levels” of the chain complex, and the arrows δ_0, δ_1, ∂_1, and ∂_2 represent homomorphisms (functions that preserve structure) between these spaces.

In the context of this paper, the chain complex is a mathematical structure that encapsulates the relationships between different “dimensions” of the problem the authors are studying, and studying the “expansion” properties of this chain complex can lead to new insights about the structure of hard instances for the Sum-of-Squares (SoS) semi-definite programming (SDP) hierarchy.

3.12 Some basic notions from homology and cohomology

The symbols δ_0 and δ_1 represent linear maps known as the co-boundary operators, which form the backbone of the mathematical structure of a cochain complex. These operators map each component (or dimension) of the complex to the next.

Similarly, ∂_1 and ∂_2 are the transposes of δ_0 and δ_1, respectively, and are called the boundary operators in the context of a chain complex.

The equalities $\partial_1\partial_2 = 0$ and $\delta_1\delta_0 = 0$ reflect fundamental properties of chain complexes and cochain complexes, respectively. They state that the composition of two consecutive boundary operators (or two consecutive co-boundary operators) is the zero map, which is essential for the concept of homology (or cohomology) that underpins the topological and algebraic study of such complexes.

3.13 Back to SS-HDX

The concept of high-dimensional (co)-boundary expansion, an analogue of edge expansion in graphs, is introduced in the context of chain complexes. Edge expansion in graphs is a property that measures how “quickly” one can escape a subset of vertices by traversing edges. Similarly, high-dimensional (co)-boundary expansion in a chain complex measures the “expansion” from one dimension to the next in the complex.

An important structural feature of chain complexes is highlighted: any function f in the image of δ_0, known as a co-boundary, satisfies $|\delta_1 f| = 0$. In simple terms, this means that applying the co-boundary operator δ_1 to a co-boundary f (i.e., a function in the image of δ_0) results in the zero function. This is analogous to how in a graph, applying the boundary operator to a boundary (an edge) results in the zero function (no vertices).

A complex is considered a ρ-co-boundary expander when the above property is the only reason that $|\delta_1 f|$ isn't larger. This is formalized in the inequality, which states that for all functions f in $\mathbb{F}_2^{X(1)}$, the size of the image of f under δ_1 is greater than or equal to ρ times the distance from f to the image of δ_0.

In this definition, the term “distance” could refer to a measure of how far the function f is from being a co-boundary (a function in the image of δ_0), perhaps in terms of some norm or another mathematical measure.

In this context, $|\delta_1 f|$ refers to the size, or “weight”, of the function f after it has been transformed by the coboundary operator δ_1.

The weight of a function in $\mathbb{F}_2^E$, where E is the set of edges, is typically understood as the number of edges for which the function evaluates to 1. More formally, given a function, the weight $|f|$ is defined as the cardinality of the set $\{e \in E | f(e) = 1\}$.

In the context of the expression $|\delta_1 f|$, f is first transformed by the operator δ_1 into a new function, and then the weight of this new function is calculated.

3.14 Generalizing from graphs to chain complexes

Chain complexes admit a natural analog of boundary (edge) expansion in graphs called high-dimensional (co)-boundary expansion [LM06]. To see this, we first note an important inherent structural property of chain complexes: any function $f \in \operatorname{im}(\delta_0)$ (called a co-boundary) satisfies $|\delta_1 f| = 0$. A complex is called a ρ-co-boundary expander essentially when this is the only obstruction to $|\delta_1 f|$ being large:

$$|\delta_1 f| \geqslant \rho \cdot d\left(f, \operatorname{im}(\delta_0)\right).$$

The concept of co-boundary expansion is a generalization of the notion of edge (or boundary) expansion, which originates from the field of graph theory. It is used in the context of chain complexes, which are higher-dimensional analogs of graphs. The co-boundary expansion of a chain complex is a measure of how well the complex expands in high dimensions.

In simple terms, an edge in a graph separates two sets of vertices. Analogously, a (higher-dimensional) face in a simplicial complex separates two (lower-dimensional) chains. The co-boundary of a chain is the set of all faces that separate the chain from its complement.

The co-boundary expansion of a chain complex is then defined in terms of the size of the co-boundary of every chain. Specifically, the co-boundary expansion is the minimum over all chains of the ratio of the size of the co-boundary to the size of the chain itself. The larger the co-boundary expansion, the better the chain complex is at expanding in high dimensions.

To make this more concrete, consider a function f that assigns a value to every element in a certain dimension of the chain complex (for instance, the vertices in a graph, or the edges in a hypergraph). The co-boundary of this function f is the set of all elements in the next higher dimension that are adjacent to an odd number of elements to which f assigns the value 1.

Then the co-boundary expansion property essentially says that for every such function f, the size of the co-boundary (the number of elements in the co-boundary) is large, unless the function f is itself a co-boundary (which can be thought of as a trivial or uninteresting case). This is a measure of how well the elements in the chain complex are interconnected, which has many important applications, for instance in coding theory and in complexity theory.

The concept of high-dimensional (co)-boundary expansion, an analogue of edge expansion in graphs, is introduced in the context of chain complexes. Edge expansion in graphs is a property that measures how "quickly" one can escape a subset of vertices by traversing edges. Similarly, high-dimensional (co)-boundary expansion in a chain complex measures the "expansion" from one dimension to the next in the complex.

An important structural feature of chain complexes is highlighted: any function f in the image of δ_0, known as a co-boundary, satisfies $|\delta_1 f| = 0$. In simple terms, this means that applying the co-boundary operator δ_1 to a co-boundary f (i.e., a function in the image of δ_0) results in the zero function. This is analogous to how in a graph, applying the boundary operator to a boundary (an edge) results in the zero function (no vertices).

A complex is considered a ρ-co-boundary expander when the above property is the only reason that $|\delta_1 f|$ isn't larger. This is formalized in the inequality, which states that for all functions f in $\mathbb{F}_2^{X(1)}$, the size of the image of f under δ_1 is greater than or equal to ρ times the distance from f to the image of δ_0.

In this definition, the term "distance" could refer to a measure of how far the function f is from being a co-boundary (a function in the image of δ_0), perhaps in terms of some norm or another mathematical measure.

In this context, $|\delta_1 f|$ refers to the size, or "weight", of the function f after it has been transformed by the coboundary operator δ_1.

The weight of a function in $\mathbb{F}_2^E$, where E is the set of edges, is typically understood as the number of edges for which the function evaluates to 1. More formally, given a function $f|E \to \mathbb{F}_2$, the weight $|f|$ is defined as the cardinality of the set $e \in E|f(e) = 1$.

In the context of the expression $|\delta_1 f|$, f is first transformed by the operator δ_1 into a new function, and then the weight of this new function is calculated.

The coboundary operator δ_0 is defined to map a function f defined on vertices to a constant function, i.e., a function defined on the empty set, which effectively represents the entire graph (since any function defined on the empty set is essentially a constant). This is somewhat abstract and is essentially a formalism, but it's useful in setting up the properties of the coboundary operators and the overall cochain complex.

The choice of this specific definition allows us to conveniently formulate certain properties of the graph, like the requirement that $\delta_1 \delta_0 = 0$ which must hold for a cochain complex, and it also leads to the interpretation of co-boundary expansion that is equivalent to the standard definition of boundary expansion in graphs.

Indeed, it might seem unusual that the coboundary operator δ_0 would map a function f defined on the vertices of a graph to a constant function. But keep in mind, this is a mathematical abstraction. The choice is made to meet the requirements of a cochain complex, in which the composition of two successive boundary or coboundary operators is zero. Specifically, in a cochain complex, we have $\delta_i \delta_{i-1} = 0$.

To achieve this in our current setup, where we're working with a graph, we have δ_1 defined on the edges of the graph and δ_0 defined on the vertices. For $\delta_1 \delta_0$ to be zero for all inputs, we must have δ_0 map every vertex function to a constant function. This is because δ_1 is taking an XOR of function values on the vertices. If those function values are all the same (i.e., a constant), then their XOR will always be zero, no matter what edge we're considering.

Notice that in this setup, the only co-boundaries are $\operatorname{im}(\delta_0) = \{\varnothing, V\}$. Furthermore, for any subset $S \subset V$ and any edge $e \in E$, the value of $\delta_1 1_S$ on e is 1 if and only if e crosses the "cut" defined by S. Here, 1_S denotes the indicator function of the set S.

This observation leads to the conclusion that the ratio $\frac{|\delta_1 1_S|}{d\left(1_S, \operatorname{im}(\delta_0)\right)}$ is equivalent to $\frac{E(S, V \setminus S)}{\min\{|S|, |V \setminus S|\}}$, which is simply the standard definition of boundary expansion in graphs.

In other words, high-dimensional co-boundary expansion in chain complexes extends the idea of boundary expansion in graphs, allowing us to study "expansion" properties in more complex, high-dimensional structures.

The two ratios are essentially measures of how well-connected a set of vertices S is to the rest of the graph. They are both forms of "expansion" of a graph or a set within a graph.

Let's break it down:

1. $\frac{|\delta_1 1_S|}{d\left(1_S, \operatorname{im}(\delta_0)\right)}$: This ratio is the number of edges that "cross" the cut defined by S, divided by the size of S. In other words, it's a measure of how many edges are leaving the set S compared to the size of S. If this number is large, then S is very well connected to the rest of the graph.

2. $\frac{E(S, V \setminus S)}{\min\{|S|, |V \setminus S|\}}$: This ratio is the number of edges between S and the complement of S (i.e., the rest of the graph), divided by the smaller of the sizes of S and $V \setminus S$. This is the standard measure

of “boundary expansion” in a graph. If this ratio is large, it means that the set S has many edges connecting it to the rest of the graph compared to its size.

Therefore, the two ratios essentially quantify the same property about the set S – the number of edges connecting S to the rest of the graph relative to the size of S. They both serve as a measure of “expansion” in a graph, where a larger value indicates better connectivity or expansion. The specific definitions and terms used (like δ_1 or d) depend on the mathematical framework or context, but the essential idea remains the same.

In the context of this discussion, $d(1_S, \operatorname{im}(\delta_0))$ refers to the 'distance' between the characteristic function of the set S (denoted as 1_S) and the image of the co-boundary operator δ_0. The exact nature of this 'distance' might vary depending on the particular mathematical setting, but it's often defined in terms of some norm or metric on the function space that the chain complex lives in.

On the other hand, $\min\{|S|, |V \backslash S|\}$ is simply the smaller of the sizes of the set S and its complement in the vertex set V.

To see why these two quantities might be related, consider what they represent. $d(1_S, \operatorname{im}(\delta_0))$ captures some notion of how 'far' the function 1_S is from being a co-boundary – in other words, how far it is from being a function that could be expressed as the 'boundary' of some higher-dimensional object in the chain complex. When S is a subset of V that's approximately half the size of V, this 'distance' might intuitively be expected to be large, because such a function 1_S won't have much of a higher-dimensional 'structure' to it – it's just splitting the vertices into two roughly equal-sized groups.

Similarly, $\min\{|S|, |V \backslash S|\}$ is a measure of how balanced the cut defined by S is. When this quantity is small, the cut is very imbalanced – one side of the cut has much fewer vertices than the other.

Therefore, both quantities capture, in different ways, some measure of how 'imbalanced' or 'structureless' the cut defined by S is. They are not equivalent, and their relationship could be complex and depend on the specifics of the chain complex and the operators δ_0 and δ_1, but they both serve to quantify certain aspects of the 'quality' of the cut defined by S in the graph.

3.15 The notion of small-set boundary expander

Unfortunately, while standard boundary expansion on (random) graphs has been quite useful for proving SoS lower bounds in the past [BSW99, Gri01b, Sch08], high dimensional co-boundary expansion seems to be too strong a notion for this setting: good (co)-boundary expanders are not known to exist (even probabilistically), and their structure is prohibitively restrictive in other senses as well 2 We avoid these issues by introducing a simple relaxation of boundary expansion to small-sets:

Definition 1.2 (Small-set (Co)-Boundary Expansion). We call X a (ρ_1, ρ_2)-small-set boundary expander if the weight of any 'small' function $f \in \mathbb{F}_2^{X(1)}$ satisfying $|f| \leqslant \rho_1 |X(1)|$ expands:

$$|\partial_1 f| \geqslant \rho_2 \cdot d\left(f, \operatorname{im}\left(\partial_2\right)\right)$$

This passage introduces a relaxation of the (co)-boundary expansion property for a chain complex X. This relaxation is designed to focus on “small” functions, overcoming the challenges that arise from the fact that good (co)-boundary expanders seem to be hard to find and their structures tend to be overly restrictive.

In particular, Definition 1.2 is given for a Small-set (Co)-Boundary Expander:

We say that the chain complex X is a (ρ_1, ρ_2)-small-set boundary expander if the following property holds: for any 'small' function $f \in \mathbb{F}_2^{X(1)}$, where 'small' means that the function satisfies $|f| \leqslant \rho_1 |X(1)|$, the weight of the function expands under the boundary operator ∂_1.

Mathematically, this property can be expressed as:

$$|\partial_1 f| \geqslant \rho_2 \cdot d\left(f, \operatorname{im}\left(\partial_2\right)\right)$$

This essentially means that the weight (or "size") of the transformed function $\partial_1 f$ is at least a ρ_2 fraction of the distance from f to the image of the boundary operator ∂_2.

In practical terms, this property is checking how much the weight (or "size") of a small function can be expanded by the action of the boundary operator ∂_1. This concept is useful for designing and analyzing algorithms, especially in contexts such as constraint satisfaction problems where the ability to expand small sets is a valuable property.

When we talk about a "small-set" in this context, we're referring to a function f that assigns a value to a relatively small number of elements in the chain complex. In other words, f is non-zero on a small number of elements.

Now, the "co-boundary" of such a function f is the set of all elements in the next higher dimension that are adjacent to an odd number of elements for which f is non-zero.

The concept of "expansion" then refers to the size of the co-boundary of f. If the co-boundary is large (i.e., there are many higher-dimensional elements adjacent to an odd number of elements where f is non-zero), then we say that f expands.

So, the property of small-set co-boundary expansion essentially means that for every function f that is non-zero on a small number of elements, the co-boundary of f is large, unless f is a co-boundary itself.

In simpler terms, it's a measure of how interconnected or "expanded" the elements in a complex are, even when we're only looking at a small subset of those elements. It's a particularly useful concept in the study of the efficiency of certain algorithms, and it has applications in fields like coding theory and computational complexity theory.

3.16 Small-set coboundary expander

Similarly, X is a (ρ_1, ρ_2)-small-set co-boundary expander if all $f \in \mathbb{F}_2^{X(1)}$ s.t. $|f| \leqslant \rho_1 |X(1)|$ satisfy:

$$|\delta_1 f| \geqslant \rho_2 \cdot d\left(f, \operatorname{im}\left(\delta_0\right)\right)$$

We call X a (ρ_1, ρ_2)-small-set HDX $(SS - HDX)$ if it satisfies both the above conditions.

This passage defines the concept of a Small-set Co-Boundary Expander and a Small-Set High Dimensional Expander (SS-HDX).

A chain complex X is referred to as a (ρ_1, ρ_2)-small-set co-boundary expander if it fulfills the following condition: For all functions $f \in \mathbb{F}_2^{X(1)}$ with $|f| \leqslant \rho_1 |X(1)|$ (i.e., the function f is small), the size of the function f expands under the action of the co-boundary operator δ_1:

$$|\delta_1 f| \geqslant \rho_2 \cdot d\left(f, \operatorname{im}\left(\delta_0\right)\right)$$

This condition means that the weight of the transformed function $\delta_1 f$ is at least a ρ_2 fraction of the distance from f to the image of the co-boundary operator δ_0.

3.17 The exact definition of SS-HDX

Then, the chain complex X is called a (ρ_1, ρ_2)-small-set high dimensional expander (SS-HDX) if it satisfies both of the above conditions, meaning it is both a small-set boundary expander and a small-set co-boundary expander. In other words, a SS-HDX has the property that all small sets are expanded when acted on by both the boundary and the co-boundary operators. This generalization of the expansion property to high-dimensional settings provides a powerful tool in the study of theoretical computer science problems.

3.18 Constructing an infinite family of SS-HDX

We saw small-set (co)-boundary expansion on high dimensional expanders (SS-HDX) is a generalization of small-set expansion in graphs. The concept of small-set expansion in graphs is critical to several problems in the hardness of approximation, especially in relation to Khot's unique games conjecture. The paper demonstrates in the next section how SS-HDX can naturally lead to hard instances of XOR for the Sum-of-Squares hierarchy, providing the first link between the hardness of approximation and high dimensional small-set expanders.

The main result, Theorem 1.1, therefore focuses on constructing an infinite family of SS-HDXs with a growing number of vertices that can be constructed in deterministic polynomial time. While this may seem overly ambitious, this has recently been achieved in some form in the breakthrough constructions of quantum Low-Density Parity-Check (qLDPC) codes. More specifically, the authors claim that the recent qLDPC codes proposed by Leverrier and Zémor already demonstrate the properties of small-set HDX, indicating that it may be possible to achieve the requirements laid out in Theorem 1.1.

Following that, Theorem 1.3 is introduced, which states that there exist constants $\rho_1, \rho_2 \in (0, 1)$ and an explicit (constructable in polynomial time) infinite family of bounded-degree (3-term) chain complexes $\{X_i\}$. These complexes satisfy two conditions:

1. X_i has non-trivial 'co-homology', that is, the image of δ_0 is not equal to the kernel of δ_1.
2. X_i is a (ρ_1, ρ_2)-SS-HDX, i.e., a small-set high-dimensional expander with parameters ρ_1 and ρ_2.

This theorem appears to provide the key to constructing the required instances mentioned in Theorem 1.1.

3.19 Connection to quantum locally testable codes

The paper indicates that the conditions given in Theorem 1.3 are stronger than those initially established by Leverrier and Zémor [LZ22]. This theorem demonstrates the most potent known form of bidirectional high-dimensional expansion to this day.

Moreover, the expansion is so robust that if one could discard the small-set requirement or demonstrate similar bounds for a 5-term chain complex, it would solve the qLTC (quantum Locally Testable Code) conjecture, a significant open question in the field of quantum computation. This conjecture [KKL14, EH17, LH22a] is about the existence of quantum error-correcting codes that are locally testable, which is a critical issue in developing robust quantum computers.

4 From SS-HDX to Hardness

4.1 Introduction

A chain complex, particularly a Small-Set High Dimensional Expander (SS-HDX), can be transformed into a hard instance of the 3-XOR problem. The 3-XOR problem is a constraint satisfaction problem (CSP) where the goal is to find a solution that satisfies a maximum number of 3-variable XOR (exclusive or) constraints.

The construction starts by creating a bipartite graph B with left vertex set L and right vertex set R. Elements in L correspond to variables and elements in R correspond to the set of constraints. A variable assignment is then fixed to the constraints.

An XOR instance associated with this graph ensures that the sum (mod 2) across neighbors of each right vertex in R equals the variable assignment for that vertex. This is the XOR constraint. If this sum equals the variable assignment, the constraint is satisfied.

In previous constructions for proving hardness, the bipartite graph B was typically picked at random to satisfy strong expansion properties (which make it hard for approximation algorithms to find a solution), and the variable assignment was also typically chosen randomly to ensure unsatisfiability of the XOR instance (i.e., there is no assignment of the variables that can satisfy all the constraints).

However, a significant point of interest in this discussion is the de-randomization of the variable assignment (or the constraints). Unlike the graph B, whose choice can sometimes be de-randomized and still retain good inapproximability guarantees, no efficient method was known to de-randomize the variable assignment, until this work. [HL] state that prior to this, the best method was brute force search over log(n)-size instances, which is computationally intensive.

In previous work, B was typically chosen randomly in order to achieve strong expansion properties, and β was chosen randomly to ensure unsatisfiability of the CSP. While it's sometimes possible to de-randomize the choice of B and still achieve good inapproximability guarantees, de-randomizing the choice of β has traditionally required brute force search over instances of size $\log(n)$. However, the text seems to suggest that new methods for de-randomizing β have been developed in this context.

4.2 From chain complex to XOR instance

The authors outline the conversion of a chain complex into an instance of XOR problem. This conversion is an important step because it allows us to relate structural properties of the chain complex to the hardness of the resulting XOR instance, thus revealing the interplay between algebraic topology and computational complexity.

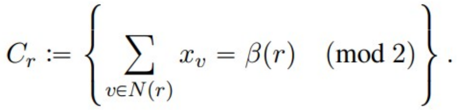
A chain complex is a sequence of abelian groups (or modules, or vector spaces), connected by homomorphisms. In this case, the chain complex is represented over a finite field of order 2, $\mathbb{F}_2$, and it's a sequence of three spaces:

$$X : \mathbb{F}_2^{X(0)} \underset{\partial_1}{\overset{\delta_0}{\rightleftarrows}} \mathbb{F}_2^{X(1)} \underset{\partial_2}{\overset{\delta_1}{\rightleftarrows}} \mathbb{F}_2^{X(2)},$$

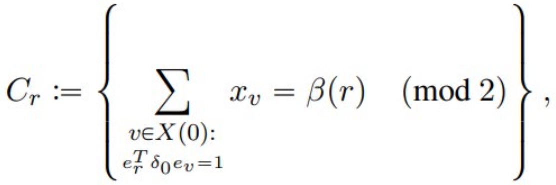
Here, $X(0)$, $X(1)$, and $X(2)$ are the 0-, 1-, and 2-dimensional parts of the complex respectively, and δ_0 and δ_1 are the co-boundary operators between these parts.

To create an XOR instance from this chain complex, the authors propose a method to transform the linear map δ_0 into a graph. They note that any linear operator from $\mathbb{F}_2^{X(0)}$ to $\mathbb{F}_2^{X(1)}$ can be represented as a matrix over $\mathbb{F}_2$ of size $(|X(1)| \times |X(0)|)$. This matrix can be considered as the bipartite adjacency matrix of a graph with left vertices $L = X(0)$ and right vertices $R = X(1)$.

Given a function $\beta \in \mathbb{F}_2^{X(1)}$, an XOR instance, denoted as $\mathcal{I}_{X,\beta}$, is constructed by adding a constraint C_r for each $r \in X(1)$:



$$C_r := \left\{ \sum_{v \in N(r)} x_v = \beta(r) \pmod 2 \right\}.$$



$$C_r := \left\{ \sum_{\substack{v \in X(0): \\ e_r^T \delta_0 e_v = 1}} x_v = \beta(r) \pmod 2 \right\},$$

Here, e_v and e_r are the standard basis vectors corresponding to $v \in X(0)$ and $r \in X(1)$. The term $e_r^T \delta_0$ gives the list of neighbors of r, thus making this construction an instantiation of the standard bipartite framework.

The authors point out that their approach generalizes the one presented by [DFHT20] where XOR instances were constructed using 3-dimensional simplicial complexes (4-uniform hypergraphs) by letting triangles correspond to constraints, and edges correspond to variables. This approach is essentially the result of applying their construction to the natural chain complex associated with a 3-dimensional simplicial complex.

4.3 Breakdown of the basic terminology

- An XOR instance is a problem in which we try to assign binary (0 or 1) values to a set of variables such that certain constraints are satisfied, where each constraint is an XOR of a subset of the variables.

- $\mathbb{F}_2^{X(1)}$ denotes the set of all binary functions defined on $X(1)$. Each such function maps each element of $X(1)$ to a binary value.

- $\beta \in \mathbb{F}_2^{X(1)}$ is a specific function from the set $X(1)$ to the binary numbers $\{0, 1\}$.

- C_r is a constraint that corresponds to an element r in the set $X(1)$.

- The $\sum x_v$ notation refers to the sum over all $v \in X(0)$ for which the edge (v, r) exists in the graph (that is, r and v are connected). Here, x_v represents a variable corresponding to $v \in X(0)$.

- e_v and e_r denote the standard basis vectors corresponding to $v \in X(0)$ and $r \in X(1)$. These are vectors that have a single '1' in the position corresponding to the specific element (either v or r) and '0' everywhere else.

- The notation $e_r^T \delta_0 e_v = 1$ is a condition to determine whether r and v are connected. This is equivalent to asking if there is an edge from v to r in the graph. This relationship is defined by the linear operator δ_0.
- The expression $\beta(r) \pmod{2}$ means the output of the function β at r, modulo 2.

So, the constraint C_r for a given r states that the sum (modulo 2) of the variables x_v over all v connected to r should equal the value of the function β at r. This forms a system of XOR equations which constitute the XOR instance $\mathcal{I}_{X,\beta}$.

4.4 The choice of β – the XOR instance is satisfiable iff β is a coboundary!

Remember that for any instance of XOR derived from a chain complex X and a choice of function β, the instance is satisfiable if and only if β is a coboundary. Following a framework presented in prior work [DFHT20], the authors choose β to be a cocycle but not a coboundary. That means, β satisfies the condition of being closed (its boundary is zero), but it is not exact (it's not the boundary of another form).

When working with a sufficiently expanding complex, this particular choice of β imparts a global structure on the XOR instance that cannot be captured by local views of the complex. The global structure induced by β is non-trivial and cannot be discerned when looking only at small pieces of the complex, where the homology and cohomology seem trivial. This is particularly relevant when considering the Sum-of-Squares (SoS) method, which operates over local views of the instance.

This leads to **Theorem 2.1**, which essentially asserts that if you start with a chain complex X that has non-trivial cohomology and high-dimensional expansion, you can derive XOR instances with certain "hardness" properties. Specifically, if β is a cocycle but not a coboundary, the resulting XOR instance will have two characteristics:

1. **Soundness:** The XOR instance is at most $(1-\mu_1)$-satisfiable, meaning it can't be completely satisfied. This reflects the infeasibility of the problem.
2. **Completeness:** The XOR instance cannot be refuted using up to $\mu_2|X(0)|$ levels of the SoS hierarchy. This reflects the computational hardness or intractability of the problem.

The exact values of μ_1 and μ_2 are not specified here, but they are within the interval $(0,1)$. This indicates that there's some degree of flexibility in the degree of soundness and completeness.

4.5 How does small-set expansion contribute to soundness and completeness?

Soundness: The soundness comes intuitively from the small-set coboundary expansion property. If an element in $Z^1 \setminus B^1$ is far from the coboundary, by definition of the XOR instance construction, the instance is satisfiable exactly when β in $\mathbb{F}_2^{X(1)}$ is a coboundary. So, intuitively, the farther the function is from being a coboundary, the less likely it is that the XOR instance is satisfiable. This is a property of small-set coboundary expanders, and it means that instances created from them will be far from satisfiable.

Completeness: Completeness, on the other hand, requires the full power of small-set boundary expansion. It comes from the global structure of the (co)homology that cannot be detected through local views of the complex. The authors note that this can be reformulated as an isoperimetric inequality: "small, minimal functions have large boundaries." A minimal function here is one where adding any boundary can only increase its size (Hamming weight).

The authors then outline how to use this property to prove the completeness of their instances. They propose to combine the isoperimetric inequality with classical arguments from previous work to show that the width of any refutation of an XOR instance in the plus-resolution proof system is large.

A plus-resolution proof system is a system that uses a set of rules to deduce all the possible consequences of a given set of logical statements.

Since it was previously shown that any bound on the width of a refutation transfers to a completeness lower bound for the Sum-of-Squares method, this argument completes the proof of completeness for the XOR instances.

4.6 Construction of refutation in the $\oplus$-resolution proof system and conversion to 3-XOR instances

The authors further elaborate on the construction of a refutation in the $\oplus$-resolution proof system, explaining how it corresponds to a directed acyclic graph (DAG). The leaves of this DAG correspond to the original XOR constraints, internal nodes correspond to the XOR of their parent nodes, and the root derives a contradiction.

Every element $s \in X(1)$ corresponds to a constraint in the XOR instance. A function $h_v \in \mathbb{F}_2^{X(1)}$ is assigned to each node v in the DAG to keep track of which XOR constraints are being used. The boundary of this function, $\partial_1 h_v \in \mathbb{F}_2^{X(0)}$, corresponds to the set of variables in the equation corresponding to node v. Lower bounding the width of the refutation is equivalent to finding a node with a large boundary.

Small-set boundary expansion, specifically the isoperimetric inequality, becomes relevant here. It states that to lower bound the width of the refutation, it's enough to find a node of 'medium' weight. This weight is small enough to apply the inequality, but large enough to result in a large boundary.

The authors also mention that the instances of CSPs provided by Equation (2) and Theorem 2.1 are usually instances of MAX-k-XOR and not 3-XOR, where k is the maximum degree of the complex. However, this is not a significant issue, as the SS-HDX constructed are of bounded degree. This means every constraint in the XOR has a constant number of variables, and every variable appears in a constant number of constraints. This observation allows them to convert to hard instances of 3-XOR using standard NP-reduction arguments within the SoS hierarchy while only losing constant factors in the soundness and levels of hardness for SoS.

4.7 Application of small-set expansion for soundness and completeness

This passage explains the application of small-set expansion in ensuring soundness and completeness for XOR instances created using simplicial complexes.

Soundness is easier to prove. The small-set co-boundary expansion guarantees that any element in $Z^1 \backslash B^1$ is far from being a co-boundary. This implies that the XOR instance, $\mathcal{I}_{X,\beta}$, is satisfiable only when $\beta \in \mathbb{F}_2^{X(1)}$ is a co-boundary. Therefore, functions that are far from being a co-boundary are also far from being satisfiable.

Completeness is trickier to prove and needs the full power of small-set boundary expansion. The argument revolves around the inability of local views of the complex to detect the global structure of (co)-homology. It's observed that small-set boundary expansion can be equivalently restated as an isoperimetric inequality: small, minimal functions have large boundaries. The concept of co-systolic distance is important here.

The subsequent arguments are then leveraged to show that the width of any refutation of $\mathcal{I}_{X,\beta}$ in the $\oplus$-resolution proof system is large. This transfers into a lower bound for completeness for Sum-of-Squares (SoS).

This proof is supplemented by detailed examination of refutations in the $\oplus$-resolution system. The proof culminates with the creation of a potential function that demonstrates the existence of an interior node with medium potential, thereby proving the desired bound.

Lastly, the author notes that the CSPs are generally instances of MAX-k-XOR, not 3-XOR. However, the complexes constructed are of bounded degree, allowing the reduction to hard instances of 3-XOR while only losing constant factors in the soundness and levels of hardness for SoS.

4.8 A step-by-step proof of completeness

The proof of completeness relies on some rather sophisticated concepts from graph theory, computer science, and homological algebra. Here is a step-by-step breakdown of the proof:

Establishing the problem

The proof involves an XOR instance $\mathcal{I}_{X,\beta}$ created using a simplicial complex with small-set boundary expansion properties. The instance is satisfiable only when $\beta \in \mathbb{F}_2^{X(1)}$ is a co-boundary. The objective is to show that the width of any refutation of $\mathcal{I}_{X,\beta}$ in the $\oplus$-resolution proof system is large.

The "width of a refutation" in a proof system, in particular in the context of the $\oplus$-resolution proof system, refers to the size of the largest clause used in the refutation. The $\oplus$-resolution proof system is a specific proof system that is used for reasoning about XOR-constraints, i.e., equations of the form $x_1 \oplus x_2 \oplus \cdots \oplus x_n = b$, where x_i are Boolean variables, $\oplus$ denotes addition modulo 2 (XOR operation), and b is a Boolean constant.

In the $\oplus$-resolution proof system, the resolution rule allows us to take two constraints $x_1 \oplus \cdots \oplus x_n = b$ and $x_1 \oplus \cdots \oplus x_n \oplus x_{n+1} = b'$ and derive the new constraint $x_{n+1} = b \oplus b'$. The "width" of a constraint in this context is the number of variables that appear in the constraint.

In the context of this completeness proof, the aim is to show that any refutation of the given XOR instance must involve a clause with a large number of variables (i.e., it has a large width), indicating that the proof must be complex. This is typically interpreted as demonstrating the computational hardness of the underlying problem.

In other words, "width of refutation" in the $\oplus$-resolution proof system measures the complexity of the refutation (proof that there is no satisfying assignment) in terms of the maximum number of variables involved in any step of the refutation. A large width implies that the problem is hard to solve for the $\oplus$-resolution proof system.

Let's consider a set of XOR constraints as follows:

1. $x \oplus y = 0$
2. $y \oplus z = 1$
3. $x \oplus z = 0$

In this case, the XOR constraints are unsatisfiable. The refutation could proceed as follows:

4. $x \oplus y \oplus y \oplus z = 0 \oplus 1$ (from constraints 1) and 2)
5. $x \oplus z = 1$ (using the property that $y \oplus y = 0$)
6. $x \oplus z = 1$ (from 5) and constraint 3), derive $x \oplus z = 1$ and $x \oplus z = 0$, leading to $1 = 0$!

In this case, the width of the refutation is 3, which is the maximum number of literals in any of the clauses.

Even though we arrived at a contradiction, which proves the unsatisfiability of the set of constraints, the contradiction clause itself ($1 = 0$) doesn't contribute to the width since it has no literals. The width is still determined by the maximum number of literals in any clause, which in this case is two (x and z in the clause $x \oplus z = 1$).

Using an isoperimetric inequality

The small-set boundary expansion can be equivalently restated as an isoperimetric inequality: small, minimal functions have large boundaries. This is a key property used in this proof.

The concept of small-set boundary expansion in the context of a simplicial complex has a close relationship with an isoperimetric inequality. This relationship can be best understood by breaking it down.

1. Small-Set Boundary Expansion: This property of a graph or complex refers to the idea that small subsets have relatively large "boundaries". In the context of a simplicial complex, a "boundary" is a sort of edge or face that is adjacent to a given subset, but not contained within it. The "small-set boundary expansion" property quantifies the idea that small subsets have disproportionately large boundaries. In other words, for a small subset of the vertices, the set of all vertices that can be reached by crossing a single edge (i.e., the "boundary") is large.

2. Isoperimetric Inequality: An isoperimetric inequality, in the simplest terms, is a mathematical statement relating the volume (or size) of a set to the size of its boundary. These inequalities arise in various areas of mathematics, including geometry, probability, and graph theory.

In the context of this proof, the isoperimetric inequality says that for any function which is "small" and "minimal" (in the sense that adding any boundary can only increase its size), the boundary of the function (i.e., the set of variables appearing in the equation corresponding to the node for which the function is defined) is large.

The equivalence between small-set boundary expansion and this isoperimetric inequality comes from the shared focus on the relationship between the size of a set and the size of its boundary. Essentially, the idea is that for small sets, the boundary is large, which is analogous to the idea that for small, minimal functions, the boundary is large.

This property of small-set boundary expansion being equivalent to the isoperimetric inequality is crucial in the proof of the completeness, as it is used to find a node in the resolution graph with a large boundary, which in turn is used to lower bound the width of the resolution refutation.

Representing the $\oplus$-resolution system

Any refutation in the $\oplus$-resolution system is represented as a Directed Acyclic Graph (DAG), where leaves are the original XOR constraints, internal nodes are the XOR of their two parents, and the root is the contradiction $0 = 1$.

A $\oplus$-resolution system is a method for finding a contradiction within a set of logical statements. This is used in computer science to reason about problems and find solutions.

We can visualize this process using a Directed Acyclic Graph (DAG), which is a graph with nodes and edges, where the edges have a direction (from one node to another), and there are no loops (a path that starts and ends at the same node).

Here's how the DAG works in this case:

1. Leaves (the nodes with no incoming edges): These represent the original constraints in the problem, which are statements involving XOR operations.

2. Internal nodes (nodes that have both incoming and outgoing edges): These represent new statements that we derive from combining the statements of their 'parent' nodes. In this case, we combine the parent statements using the XOR operation. For example, if we have two parent nodes with statements '$A \oplus B = 0$' and '$B \oplus C = 1$', we can combine these to get a new statement '$A \oplus C = 1$' at the internal node.

3. Root (the node with no outgoing edges): This is the final statement that we derive. In a successful $\oplus$-resolution refutation, the root node contains a contradiction, which in this case is '$0 = 1$'. This contradiction demonstrates that the original constraints cannot all be true together.

In summary, this graph is a visual way to keep track of how we derive new statements from the original constraints by XOR-ing them together, with the aim of finding a contradiction.

Assigning functions

Each node v in the DAG is assigned a function $h_v \in \mathbb{F}_2^{X(1)}$ that tracks which XOR constraints are being used at that node. The boundary of this function, $\partial_1 h_v \in \mathbb{F}_2^{X(0)}$, is the set of variables appearing in the equation corresponding to node v.

To illustrate this concept, let's consider a simple example using the XOR resolution system.

Assume we have the following three XOR constraints as our original problem:

1. $a \oplus b = 0$ 2. $b \oplus c = 0$ 3. $c \oplus d = 0$

We'll represent these constraints in our Directed Acyclic Graph (DAG) as the leaves (the starting points).

Now, let's perform an XOR resolution operation on constraints 1 and 2. This gives us a new constraint, $a \oplus c = 0$. We represent this as a new node in our DAG, which becomes a parent of nodes 1 and 2.

We could denote the function h_v for this new node v as a binary vector where each position corresponds to one of the original constraints (1, 2, 3), and the value at each position is 1 if the constraint contributes to the node and 0 otherwise. For our new node, the function would be $h_v = (1, 1, 0)$, because the new node is the result of XORing constraints 1 and 2.

The boundary $\partial_1 h_v$ of this function is the set of variables in the equation at node v. For our new node with the equation $a \oplus c = 0$, the boundary is $\partial_1 h_v = \{a, c\}$.

So, in this example, the function h_v helps us keep track of the original constraints that contribute to the derived constraint at node v, and the boundary $\partial_1 h_v$ tells us which variables appear in the derived constraint.

Finding a node with large boundary

To lower bound the width of the refutation, a node with a large boundary needs to be found.

In simple terms, finding a node with a large boundary means we are trying to identify a node in the resolution graph where a significant number of variables come into play.

Recall that the boundary of a node represents the set of variables involved in the equation corresponding to that node. Therefore, the larger the boundary, the more variables are involved in the equation at that node.

Why is this important? Well, it's related to the complexity of the resolution process. In our context, the width of the refutation, which we're trying to lower bound, is essentially the size of the largest set of variables involved at any step in the proof. A node with a large boundary contributes more to this width.

In other words, the larger the boundary, the more complicated (or 'wider') our refutation becomes, and therefore it becomes more difficult to refute the given instance in the $\oplus$-resolution system. This, in turn, is connected to the hardness of the problem: if refutations have large width, the problem is difficult to solve, and hence it is more likely to be a hard instance for certain algorithms like the Sum-of-Squares (SoS) hierarchy.

Using small-set boundary expansion

This is where the isoperimetric inequality (or small-set boundary expansion) is crucial. The inequality suggests that to achieve a large boundary, it suffices to find a node v of 'medium' weight that is small enough to apply the inequality, but large enough to result in a large boundary.

Small-set boundary expansion is a concept that relates the size of a set with the size of its boundary (the items it touches or is adjacent to).

In our graph, remember that each node has an assigned function and a corresponding boundary that consists of the variables used in that function.

This small-set boundary expansion property - or isoperimetric inequality - tells us something interesting: if we have a node (or, thinking in terms of the function, a set of constraints) that is 'medium' in size, its boundary (the variables it touches) is going to be large.

Why is this 'medium' size important? Well, it has to do with the nature of the isoperimetric inequality itself. If our set of constraints (node) is too small, it might not touch enough variables to make a large boundary. But, if it's too large, we could be going beyond the conditions where the isoperimetric inequality applies.

So we're looking for a 'Goldilocks' node - not too small, not too large - a medium one. This will help us ensure that we've got a large boundary (which, as we've discussed before, increases the width of the refutation, making the problem harder to solve).

Establishing a potential function

Standard potential arguments are used to establish a potential function that tracks this weight throughout the DAG. The proof argues that the leaves have small potential, the root has large potential, and that potential is sub-additive.

A potential function is like a scorekeeper. It watches what's going on in the game (in our case, as we navigate through the graph) and assigns a value - the "potential" - to each state of the game. It's a tool used in proofs to help keep track of the properties of the system we're interested in.

Here's how it works in this context:

- The potential function keeps track of the 'weight' or size of each node's function (remember, these represent sets of constraints).

- We then argue that the leaves (nodes at the end of the graph with no children) have a small potential. That's because they are just single constraints - they can't combine with other constraints to create more complex ones, hence their potential (or ability to increase the width of a refutation) is small.

- On the other hand, the root node (the one that represents our contradiction) has a large potential. This is the point where all our constraints have been combined together to form a contradiction, so it's the most complex part of our system.

- The 'sub-additive' part means that if we take two nodes and combine them, their combined potential is less than or equal to the sum of their individual potentials before they were combined. This is a common property in potential function arguments and it ensures that the potential does not blow up as we traverse through the graph.

- The use of the potential function combined with our knowledge about small-set boundary expansion allows us to argue that somewhere in our graph, there must be a node with a medium potential and a large boundary. As we discussed earlier, this implies a large width of refutation, completing our proof.

4.9 Completing the proof

This argumentation implies the existence of an interior node with medium potential, hence achieving a node with large boundary and demonstrating a lower bound for completeness for SoS.

The key idea is that through the potential function, we can track how much 'weight' (or size) is being passed along as we move through our graph from the leaves (the original constraints) towards the root (the contradiction). This 'weight' corresponds to the amount of XOR constraints being utilized.

The potential function is set up so that the leaves of the graph have a small potential (since they are just single constraints) and the root has a large potential (since it contains all the XOR constraints combined into a contradiction).

Now, because the potential function is sub-additive (meaning, when you combine two nodes, their combined potential is less than or equal to the sum of their potentials before combining), there must be a point moving from the leaves to the root where the potential is neither too small (like the leaves) nor too large (like the root), but somewhere in the middle - a node with 'medium' potential.

But here's the important part: due to the small-set boundary expansion property, a node with a 'medium' weight has a large boundary. And since the boundary represents the number of variables in the XOR equation at that node, this implies a large width of refutation.

In essence, because of the properties of the simplicial complex (small-set boundary expansion) and the way the potential function is set up, we can argue that there must exist a refutation in the $\oplus$-resolution system that has a large width. This implies the lower bound for completeness for the Sum-of-Squares proof system, thereby completing the proof.

So, to put it simply: the potential function helps us navigate through the graph from simpler to more complex parts, and due to the properties of the graph, we are guaranteed to find a point where the complexity (width of refutation) is sufficiently large. This proves the robustness of the system (completeness) for dealing with complex instances.

5 Constructing SS-HDX

5.1 Construction of Small-Set High-Dimensional Expanders (SS-HDX)

This section of the paper discusses the construction of Small-Set High-Dimensional Expanders (SS-HDX), a class of chain complexes that are useful for creating hard instances of the 3-XOR problem. The construction of SS-HDX relies on recent advances in the field of locally testable codes (LTCs) and quantum low-density parity-check (qLDPC) codes.

Here is a breakdown of the key points discussed in this section:

1. **Quantum LDPC Codes and Expanding Chain Complexes**: The authors first discuss the relationship between quantum LDPC codes and expanding chain complexes. Quantum LDPC codes are error-correcting codes used in quantum computing, characterized by the property that each check (or constraint) involves only a small number of bits. These codes can be represented as chain complexes, where each dimension of the complex corresponds to a type of constraint in the code. The "expansion" property of the code relates to its ability to spread out errors, enabling efficient error detection and correction.

2. **The qLDPC Construction of Leverrier and Zémor**: The authors review a specific construction of qLDPC codes proposed by Leverrier and Zémor in 2022. This construction is of particular interest as it provides a way to construct chain complexes with desired properties.

3. **Proof of Small-Set (Co)-Boundary Expansion**: The authors present their own proof demonstrating that the chain complexes resulting from the qLDPC construction exhibit "small-set (co)-boundary expansion". This property is crucial because it is connected to the computational complexity of solving the corresponding XOR instances, as discussed earlier in the paper.

5.2 Connection between quantum LDPC codes and chain complexes

In this section, the paper introduces the connection between quantum LDPC (Low-Density Parity-Check) codes and chain complexes, which are both tools used in the study of error correction and data encoding.

A classical error-correcting code is a method used to encode a string of k bits into a longer string of $n > k$ bits, allowing for the recovery of the original string even if some bits in the encoded string are flipped or corrupted. One common type of such codes is linear codes, which are defined by linear operators.

A linear operator $M : \mathbb{F}_2^n \to \mathbb{F}_2^{n-k}$, also known as a parity check matrix, is used to define the code $\mathcal{C}$, which is the kernel of M. Here, $\mathbb{F}_2^n$ represents all bit strings of length n, and $\mathbb{F}_2^{n-k}$ represents all bit strings of length $n-k$.

In the context of error-correcting codes, the "kernel" of M refers to the set of bit strings that are mapped to the zero string by M. These are the valid codewords in the code, representing the bit strings that can be transmitted without any error.

Regarding the footnotes:

[9]: The width of a refutation is a measure of complexity in proof systems. It is defined as the maximum number of variables appearing in any equation during the proof.

[10]: This footnote explains the rules of the $\oplus$-resolution proof system, a type of proof system used for reasoning about XOR constraints. In this system, a "refutation" is a sequence of steps leading to a contradiction, demonstrating that the XOR instance is unsatisfiable.

[11]: Here, the term "weight" is used in a slightly different sense than the standard Hamming weight (which counts the number of 1s in a bit string). It also takes into account the distance from

the boundary of a region in the chain complex.

5.3 Quantum Error-Correcting Codes and CSS Codes

In this section, the authors delve into the topic of quantum error-correcting codes. While classical error-correcting codes are used to protect classical bits against corruption, quantum error-correcting codes are designed to protect quantum bits, or qubits, from errors. These errors can be of two types: X-type errors, analogous to bit flips in classical codes, and Z-type errors, which correspond to phase flips.

A class of quantum error-correcting codes that is particularly useful is the CSS (Calderbank-Shor-Steane) codes. These codes have the advantage of being describable using classical terms. A CSS code is defined by two classical linear codes, denoted $\mathcal{C}_0$ and $\mathcal{C}_1$, which are respectively the kernels of two different parity check matrices, denoted M_0 and M_1.

These two codes are related in such a way that the orthogonal complement of $\mathcal{C}_0$, denoted $\mathcal{C}_0^{\perp}$, is a subset of $\mathcal{C}_1$. This implies that the product of M_1 and the transpose of M_0 is a zero matrix.

The dimension k of the CSS code is given by the difference between the dimensions of $\mathcal{C}_0$ and $\mathcal{C}_0^{\perp}$, while the distance d of the code measures its error-correcting capacity. The distance is defined as the minimum of d_x and d_z, where d_x and d_z represent the minimum Hamming weight (number of non-zero bits) among the non-zero vectors in $\mathcal{C}_0 \setminus \mathcal{C}_1^{\perp}$ and $\mathcal{C}_1 \setminus \mathcal{C}_0^{\perp}$, respectively.

Low-Density Parity-Check (LDPC) codes are a type of error-correcting code characterized by a parity check matrix that has a small number of ones in each row and column, hence the name 'low-density'. The quantum LDPC conjecture, which was recently resolved, states that there exists a family of quantum CSS codes with linear dimension and distance, where M_0 and M_1 are LDPC matrices. This means that there are quantum codes that can simultaneously achieve a large encoding rate ($k = \Theta(n)$) and strong error-correction capacity ($d = \Theta(n)$), while also being efficiently encodable and decodable due to the low density of the parity check matrices.

Parity check matrices are used for encoding and decoding error-correcting codes. Specifically, a parity-check matrix is a binary matrix that represents the linear equations that the codewords of an error-correcting code must satisfy. The sparsity, or low-density, of these matrices can greatly improve the efficiency of both encoding and decoding.

1. **Efficient Encodability:** Encoding a message into a codeword involves matrix multiplication. When the parity-check matrix is sparse (with mostly zeros), the number of operations required for this multiplication is significantly reduced. While dense matrices typically have a complexity of $O(n^2)$ or $O(n^3)$ (depending on the algorithm used), sparse matrices can have a complexity as low as $O(n)$, where n is the number of bits in the message or codeword.

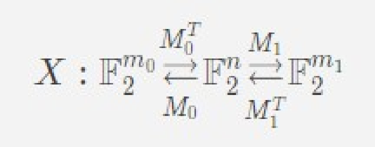
2. **Efficient Decodability:** Similarly, decoding received data back into a message involves solving a system of linear equations. This task can be computationally expensive for dense matrices. However, if the parity-check matrix is sparse, iterative decoding algorithms such as belief propagation (used for LDPC codes) can be employed. These algorithms leverage the sparsity of the matrix to achieve lower computational complexity and faster decoding speed.

In summary, the low density of parity-check matrices can significantly reduce the computational complexity of encoding and decoding, making these processes more efficient. This property makes LDPC codes particularly well-suited for applications requiring fast and efficient encoding/decoding, such as high-speed data transmission and storage systems.

5.4 Connection between Quantum CSS Codes and Chain Complexes

The key idea in this section is the connection between quantum CSS codes and chain complexes. A quantum CSS code naturally induces a chain complex, which is a sequence of vector spaces (or groups) connected by linear maps over the field with two elements, $\mathbb{F}_2$, and vice versa.

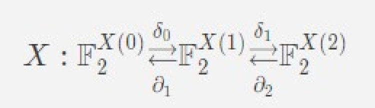
Given a quantum CSS code defined by the matrices M_0 and M_1, the induced chain complex is represented as:



$$\begin{tikzcd}X : \mathbb{F}_2^{m_0} \arrow[r, shift left, "M_0^T"] & \mathbb{F}_2^{n} \arrow[l, shift left, "M_0"] \arrow[r, shift left, "M_1"] & \mathbb{F}_2^{m_1} \arrow[l, shift left, "M_1^T"]\end{tikzcd}$$

Here, the linear maps between the vector spaces are given by the matrices defining the CSS code, and m_i denotes the dimension of the image of the map M_i, or equivalently, the rank of the matrix M_i.

Similarly, given a chain complex represented as:



$$\begin{tikzcd}X : \mathbb{F}_2^{X(0)} \arrow[r, shift left, "\delta_0"] & \mathbb{F}_2^{X(1)} \arrow[l, shift left, "\partial_1"] \arrow[r, shift left, "\delta_1"] & \mathbb{F}_2^{X(2)} \arrow[l, shift left, "\partial_2"]\end{tikzcd}$$

one can obtain a quantum CSS code by setting $M_0 : = \partial_1$ and $M_1 : \delta_1$. This means that the linear maps between the vector spaces in the chain complex can be treated as the parity-check matrices for a CSS code.

This connection allows for the translation of properties of the quantum code into topological properties of the corresponding chain complex, and vice versa. It is a key idea for the approach presented in the paper to construct hard instances of the 3-XOR problem.

5.5 Translation between Quantum CSS Codes and Chain Complexes

This part delves deeper into the connection between quantum CSS codes and chain complexes, demonstrating that many properties of the quantum codes can be translated into a homological language related to the study of algebraic topology.

Here are the mappings of these properties:

1. **Cycles and Co-cycles**: The classical codes $\mathcal{C}_0$ and $\mathcal{C}_1$ in the quantum CSS codes are analogs of cycles and co-cycles in the chain complex. A cycle is an element of the kernel of a boundary map, indicating that it is mapped to zero by the boundary map. A co-cycle is an element of the kernel of a co-boundary map.

2. **Co-boundaries and Boundaries**: The dual codes $\mathcal{C}_0^\perp$ and $\mathcal{C}_1^\perp$ in the quantum CSS codes correspond to co-boundaries and boundaries in the chain complex. A boundary is an image of an element under a boundary map, and a co-boundary is an image of an element under a co-boundary map.

3. **Dimension**: The dimension of the quantum CSS code, denoted as k, corresponds to the dimension of the cohomology of the chain complex. Cohomology measures the extent to which the boundary of a boundary fails to be zero.

4. **Maximum Degree**: The maximum degree of the chain complex corresponds to the maximum density of the parity check codes. Thus, the conditions of having bounded degree and being LDPC (low-density parity-check) are equivalent.

5. **(Co)-systolic Distance**: The X-distance and Z-distance in the quantum CSS codes correspond to the (co)-systolic distance of the chain complex. This measures the minimum weight of any (co)-cycle that is not a (co)-boundary. In other words, it represents the smallest size of a "hole" in the complex.

The equivalences presented above establish a bridge between the world of quantum CSS codes and algebraic topology. They illustrate that problems and properties in one domain can be translated and studied using the language of the other. This connection is fundamental to the methods employed in the paper to construct hard instances for the 3-XOR problem.

5.6 Establishing Conditions for Constructing Hard Instances

The papers [PK21a] and [LZ22] have presented explicit constructions of good quantum Low-Density Parity-Check (LDPC) codes. These codes map to bounded-degree chain complexes with non-trivial cohomology and linear co-systolic distance. These properties partially satisfy the requirements for constructing hard instances of the 3-XOR problem, as they ensure the soundness of the XOR construction.

However, in order to complete the proof of Theorem 1.1, the authors must further demonstrate that these complexes also satisfy a stronger condition known as "small-set (co)boundary expansion." This property, when combined with the previously mentioned conditions, guarantees not only the soundness but also the completeness of the XOR problem. It serves as the final piece needed to establish the hardness of the constructed instances.

Two footnotes are provided for clarification:

1. The authors explain that the parity check matrix is traditionally denoted by 'H', but they have chosen to use 'M' to avoid conflicts with the existing notation for homology.

2. The term $\mathcal{C}_0^{\perp}$ denotes the dual code consisting of all elements orthogonal to $\mathcal{C}_0$. This code is generated by the transpose of the parity check matrix M_0^T.

5.7 Leverrier and Zémor's qLDPC Codes

The authors now introduce the work of Leverrier and Zémor ([LZ22]), which focuses on quantum Low-Density Parity-Check (qLDPC) codes. These codes play a crucial role in the authors' construction of hard instances for the 3-XOR problem. A more detailed exposition of this work can be found in sections 7 and 8 of the paper.

Leverrier and Zémor's qLDPC codes are based on Tanner codes, a classical concept in coding theory. A Tanner code is derived from a regular graph $\mathcal{G} = (V, E)$ and a linear code C of length n_0. Here, n_0 denotes the degree of the vertices in the graph, indicating that each vertex is connected to n_0 other vertices. The Tanner code $T(\mathcal{G}, C)$ is defined as the set of all vectors c in $\mathbb{F}_2^E$ such that for each vertex v in the graph, the values on the edges connected to v form a vector in the linear code C.

The breakthrough contribution of [LZ22] lies in the observation that a quantum CSS code can be constructed from two Tanner codes derived from a higher-dimensional object called the left-right Cayley complex. This complex was recently employed in [DEL+21] to construct c3-Local Testability Codes (LTCs).

5.8 Left-Right Cayley Complex and qLDPC Construction

The construction proposed by Leverrier and Zémor involves the utilization of a mathematical object called the left-right Cayley complex, which relies on a group G and two sets of group generators $A = A^{-1}$ and $B = B^{-1}$. Here is a more detailed explanation:

- The vertex set V of the Cayley complex is identified with the group G.

- The edges are determined by two distinct Cayley graphs, namely $C(G, A)$ and $C(G, B)$, which are derived using the sets of generators A and B, respectively.

- Higher-dimensional "squares" are formed by the set $\{g, ag, gb, agb\}$, where g is an element of G, and a and b are generators from A and B, respectively.

To construct their qLDPC codes, Leverrier and Zémor consider a double cover of this complex. They define a vertex set $V = V_0 \cup V_1$, where $V_0 = G \times \{0\}$ and $V_1 = G \times \{1\}$.

The "A-edges" and "B-edges" are given by:

$$E_A = \{\{(g, 0), (ag, 1)\} | g \in G, a \in A\}, \quad E_B = \{\{(g, 0), (gb, 1)\} | g \in G, b \in B\}.$$

Finally, the squares are defined by the set:

$$F = \{\{(g, 0), (ag, 1), (gb, 1), (agb, 0)\} | g \in G, a \in A, b \in B\}.$$

This complex structure serves as the foundational framework for their construction of quantum LDPC codes.

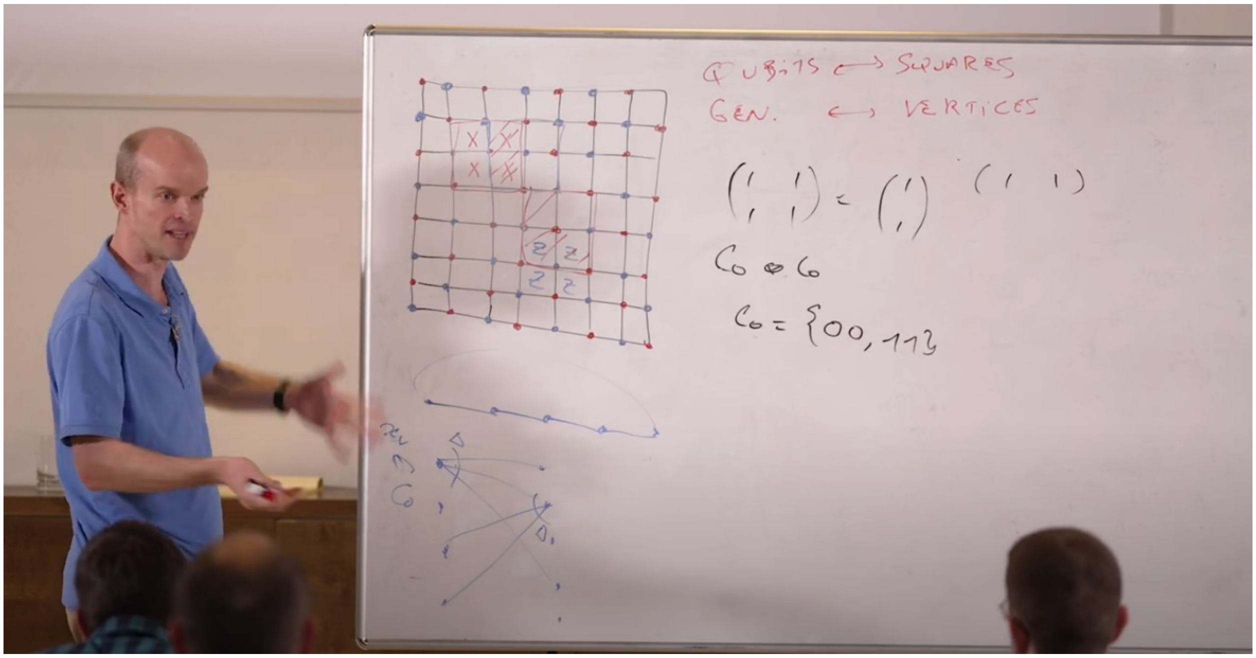
5.9 Construction of qLDPC Codes from Square Objects

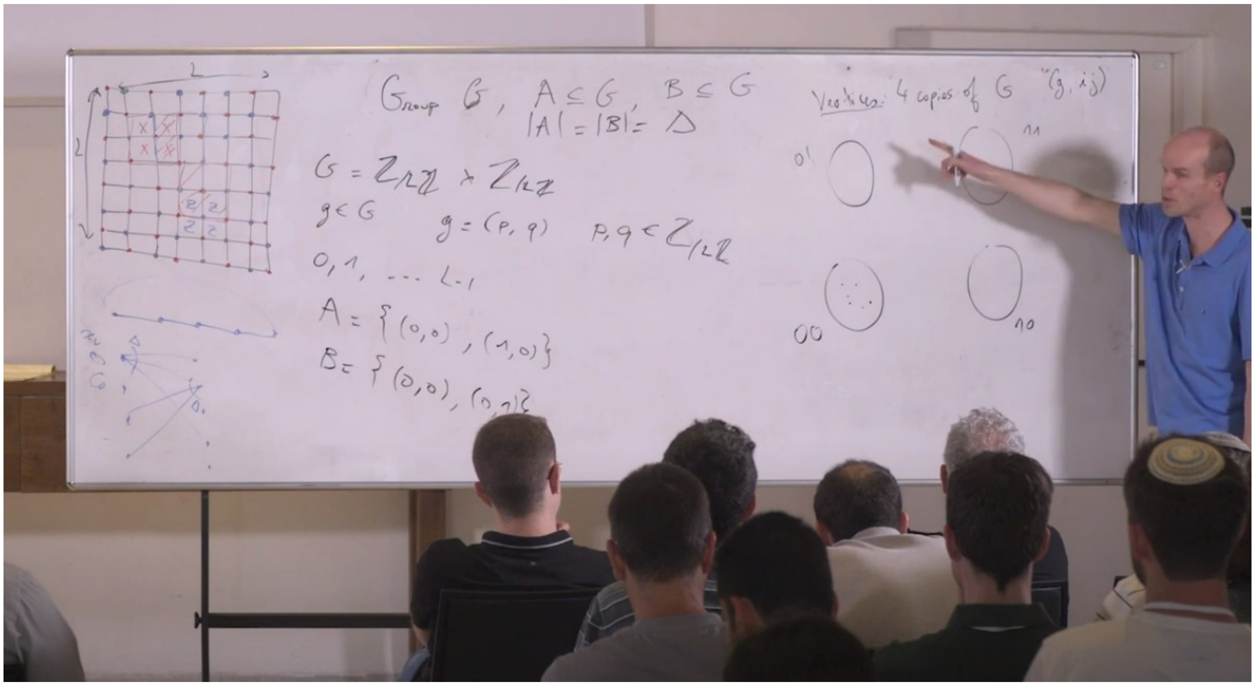
Leverrier and Zémor's construction of qLDPC codes involves the utilization of square objects in the double cover of the Cayley complex. These squares enable the definition of two graphs, denoted as $\mathcal{G}_0^\square$ and $\mathcal{G}_1^\square$, with vertices in V_0 and V_1, respectively. Each square is regarded as an edge connecting two vertices within either V_0 or V_1.

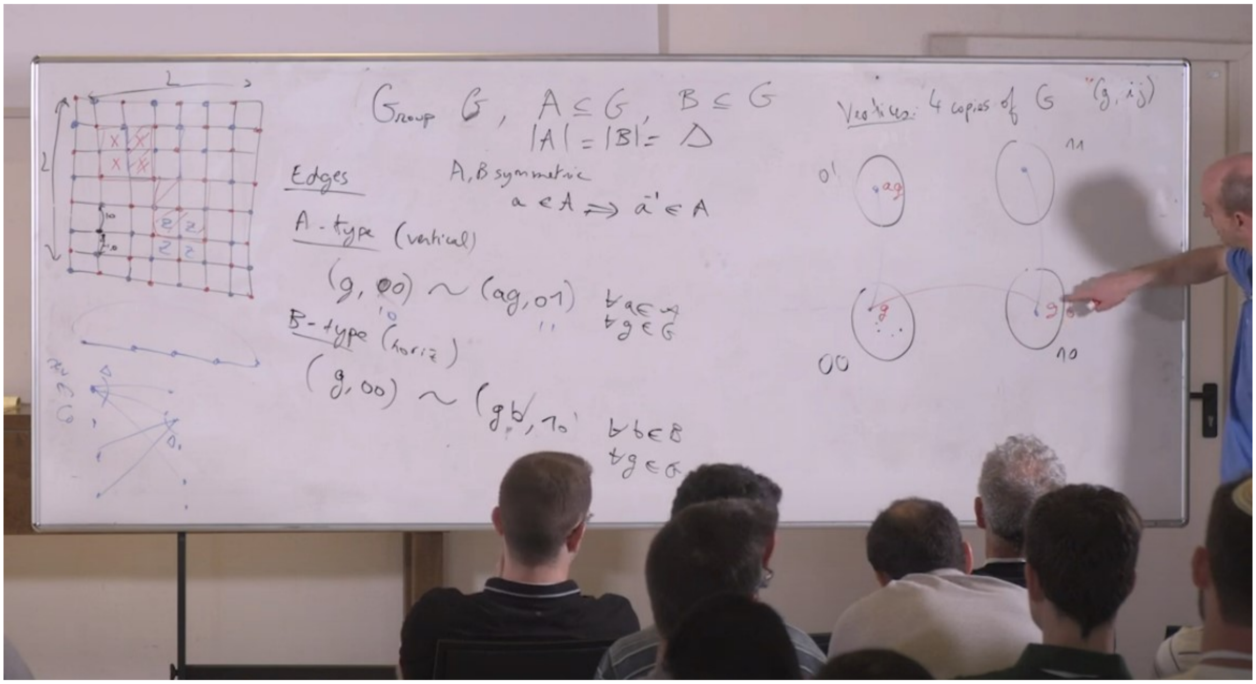
To capture the local views around each vertex (g, i) in these graphs, squares are represented by $\{(g, i), (ag, 1 - i), (gb, 1 - i), (agb, i)\}$, where $a \in A$ and $b \in B$. These local views can be envisioned as square matrices, with rows indexed by A and columns indexed by B, assuming $|A| = |B| = \Delta$.

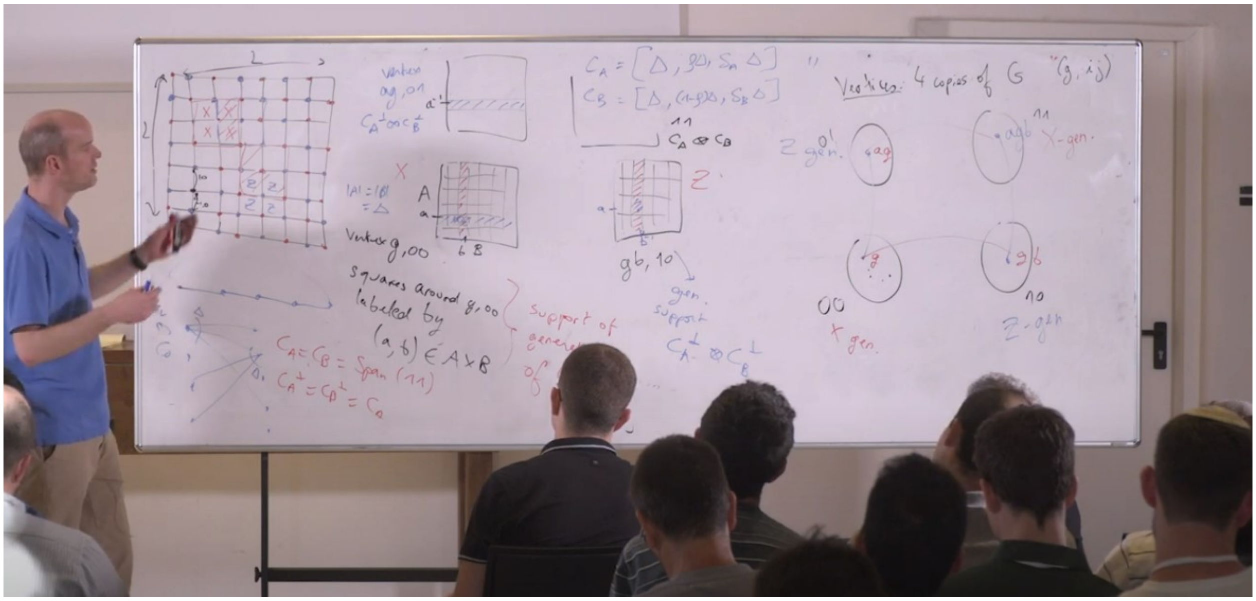
Leverrier and Zémor propose the use of Tanner codes, specifically $\mathcal{C}_0 = T\left(\mathcal{G}_0^\square, C_0^\perp\right)$ and $\mathcal{C}_1 = T\left(\mathcal{G}_1^\square, C_1^\perp\right)$, to construct a quantum CSS code satisfying $\mathcal{C}_0^\perp \subset \mathcal{C}_1$. They observed that this inclusion holds when local codes $C_0 = C_A \otimes C_B$ and $C_1 = C_A^\perp \otimes C_B^\perp$ are tensor products of linear codes $C_A \subseteq \mathbb{F}_2^A$ and $C_B \subseteq \mathbb{F}_2^B$.

Furthermore, they demonstrated that if codes $C_A, C_B, C_A^\perp, C_B^\perp$ possess linear distance and codes $C_1^\perp$ and $C_0^\perp$ meet certain robustness properties, the resulting quantum code exhibits linear distance. By employing random base codes C_A, C_B that satisfy these properties with high probability, they successfully complete the construction. Since the base codes have constant size, the construction process can be explicitly brute-forced.


QUBITS ⟷ SQUARES
GEN. ⟷ VERTICES
X X
X X
Z Z
Z Z
C₀ ⊕ C₀
C₀ = {00, 11}
Δ
Δ₁


Group G, A ⊆ G, B ⊆ G
|A| = |B| = Δ
G = Z/LZ × Z/LZ
g ∈ G g = (p, q) p, q ∈ Z/LZ
0, 1, ... L-1
A = {(0,0), (1,0)}
B = {(0,0), (0,1)}
Vertices: 4 copies of G (g, ij)
01
11
00
10
L
L

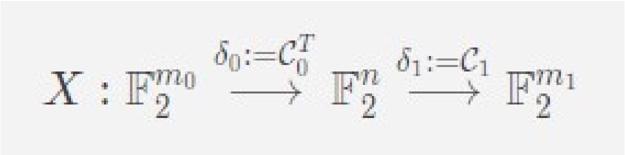

Group G, A ≤ G, B ≤ G
|A| = |B| = Δ
A, B symmetric
a ∈ A ⟹ a' ∈ A
Edges
A-type (vertical)
(g, 00) ~ (ag, 01) ∀a ∈ A ∀g ∈ G
B-type (horiz)
(g, 00) ~ (gb, 10) ∀b ∈ B ∀g ∈ G
Vertices: 4 copies of G (g, i, j)
01
11
00
10
ag
g


C_A = [Δ, gΔ, S_A Δ]
C_B = [Δ, (A-g)Δ, S_B Δ]
C_A ⊗ C_B
Vertices: 4 copies of G (g, i, j)
Z gen.
X-gen.
X gen.
Z-gen
|A| = |B| = Δ
Vertex g, 00
gb, 10
Squares around g, 00 labeled by (a, b) ∈ A × B
support of generators
gen. support C_A^⊥ ⊗ C_B^⊥
C_A = C_B = Span (11)
C_A^⊥ = C_B^⊥ = C_0
ag
agb
g
gb

5.10 Small-Set (Co)-Boundary Expansion

The authors seek to prove the property of small-set (co)-boundary expansion for the quantum CSS code constructed by Leverrier and Zémor. This property is associated with the isoperimetric inequality for small, minimal functions.

To this end, they consider a chain complex:



$$X : \mathbb{F}_2^{m_0} \xrightarrow{\delta_0 := \mathcal{C}_0^T} \mathbb{F}_2^{n} \xrightarrow{\delta_1 := \mathcal{C}_1} \mathbb{F}_2^{m_1}$$

To establish small-set co-boundary expansion for this chain complex, they aim to demonstrate the existence of constants $\rho_1, \rho_2 \in (0,1)$ such that for any minimal $x \in \mathbb{F}_2^n$ with weight $|x| \leqslant \rho_1 n$, the size of the boundary is significant: $|\delta_1 x| \geqslant \rho_2 |x|$.

To accomplish this, they employ a proof by contradiction. Assuming $|\delta_1 x| < \rho_2 |x|$, they aim to prove that x cannot be minimal by finding $y \in B^1$ (a set of 1-boundaries) such that $|x + y| < |x|$, which contradicts the assumption of minimality. In other words, they intend to show that if x has a smaller-than-expected boundary, then it cannot be a minimal element, thus contradicting the original assumption that x is minimal. This establishes the small-set (co)-boundary expansion property for the complex.

5.11 Extension to Arbitrary Functions

In this paragraph, the authors differentiate their approach from that of [LZ22] in proving co-systolic distance properties. The key distinction lies in the consideration of arbitrary functions, as opposed to solely focusing on co-cycles (elements of the cohomology group), as done in [LZ22].

Within this construction, a co-cycle corresponds to a codeword in the Tanner code $T\left(\mathcal{G}_1^{\square}, C_1^{\perp}\right)$. In a more intuitive sense, co-cycles can be seen as functions defined on the edges of the graph, such that the values of the function around any vertex form a codeword of $C_1^{\perp}$.

However, since the present proof considers arbitrary functions, these functions do not necessarily exhibit this structured property. Consequently, the authors need to account for "violations," i.e., the local views around vertices that do not correspond to codewords of $C_1^{\perp}$. These violations occur where the function x fails to be a co-cycle, or equivalently, where the boundary operator $\delta_1 x$ is non-zero.

5.12 Partitioning of Vertices based on Local Views

In this paragraph, the authors describe the partitioning of vertices in the set S, which represents the vertices incident to any square in function x. The partitioning is done based on the "local view" of x around each vertex.

The set S is divided into three categories: violated vertices S_v, normal vertices S_n, and exceptional vertices S_e:

- A vertex is considered violated and belongs to S_v if the local view of x around that vertex does not form a codeword in the code $C_1^{\perp}$.

- If the local view forms a codeword, the vertex is classified as normal and falls into S_n if the weight of the codeword is less than $w = \Delta^{3/2-\varepsilon}$. Otherwise, it is deemed exceptional and is placed in S_e.

This categorization based on weight stems from the robustness condition of the local tensor code, which ensures that codewords with weight less than w are predominantly composed of zeros, except for a small number of positions.

Furthermore, the authors note that in the local view of a normal vertex, each column is at most $O\left(\Delta^{1/2-\varepsilon}\right)$ away from a codeword in C_A, and similarly for each row and codewords in C_B. This observation relates to the Hamming distance, suggesting that any column or row in the local view of a normal vertex is close to being a codeword in the respective code.

5.13 Finding a Vertex with Large Intersection

This paragraph describes the subsequent step in the proof, which involves finding a vertex $v \in V_0$ that shares a substantial number of columns or rows with the set of normal vertices S_n (specifically $\Omega(\Delta)$ columns or rows).

Under the assumption that the sets of exceptional vertices S_e and violated vertices S_v are not significantly larger than S_n, the robustness condition of the code leads to the conclusion that the local view of vertex v is in close proximity (in terms of Hamming distance) to a codeword $c \in C_A \otimes C_B$, while also possessing a high total weight (precisely $\Omega\left(\Delta^2\right)$).

Given this, the authors construct a vector $y \in B^1$ such that y matches c on the local view of v and is zero elsewhere. Since $x + y$ and x differ only in the local view of v, and because the weight of x in that view exceeds the weight of $x + y$ (as $x + y$ incorporates the lower-weight codeword c), it follows that the weight of $x + y$ is less than the weight of x. This fulfills the desired property for the proof by contradiction, demonstrating that x was not minimal and contradicting the initial assumption.

5.14 Finding Heavy Edges and Completing the Proof

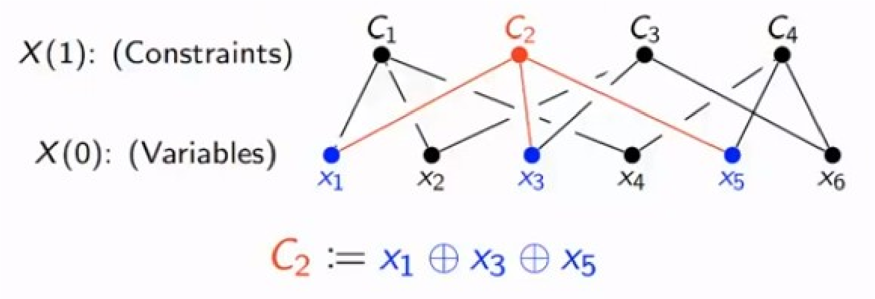
This paragraph discusses the main technical aspect of the proof, which involves identifying the vertex $v \in V_0$ that shares numerous columns or rows with the set of normal vertices S_n. A subset $T \subset V_0$ is defined as the vertices that share at least one "heavy" column or row with a normal vertex. Here, a "heavy" column or row contains a significant number of 1s. Alternatively, a "heavy" edge is one that is present in multiple squares in x.

The objective is to demonstrate the existence of many such "heavy" edges that connect S (the set of vertices incident to any square in x) and T. By utilizing the expansion property of the underlying graph and the assumption $|\delta_1 x| < \rho_2 |x|$, it can be proven that the subsets T, S_e (the exceptional vertices), and S_v (the violated vertices) are small in comparison to S_n.

Consequently, a typical vertex in T possesses not just one, but $\Omega(\Delta)$ heavy edges connecting to S_n. This, in turn, corresponds to sharing $\Omega(\Delta)$ rows and columns with normal vertices. This completes the proof of small-set co-boundary expansion.

Note:

[15] As an aside, it is mentioned that the codes C_A and C_B can be chosen to have linear distance, ensuring that the local view of vertex v exhibits a high total weight.



6 Max Hopkins' Talk on Explicit SoS Lower Bounds from HDX

6.1 Sum of Squares Hierarchy

It's a powerful algorithmic paradigm for algorithmic optimization.

1. Hierarchy of SDP relaxations roughly looking at t-local assignments. (Allowed to look at t variables at once, along with some local consistency checks.)

2. Best-known algorithm for CSP approximation (including unique games).

3. Optimal under UGC! [Rag08]

A CSP consists of variables $\{x_i\}_{i=1}^{n}$ and constraints $\{C_j\}_{j=1}^{m}$.

- Value of CSP is maximum number of satisfiable constraints. - Examples: XOR, SAT, UG...

We say a CSP (family) is 'hard to approximate' for SoS if:

1. (Soundness): CSPs are far from satisfiable (Value ¡¡ 1)

2. (Completeness): SoS 'thinks' they are satisfiable (SDP Value $\approx$ 1)

Motivating Question: What type of structure is hard for SoS?

What CSPs are hard for SoS?

Classical answer: Random CSPs are hard! [Gri03, Sch08, Tul09]

Consider XOR instance defined by bipartite graph:

And an assignment $\beta \in \mathbb{F}_2^{X(1)}$

$$\mathrm{XOR} - \mathrm{instance}(X, \beta) | \{C_i = \beta_i\}$$

If underlying graph is an expander, random β is 'hard' for SoS [Gri03]. - For any assignment, 1/2 constraints violated w.h.p. - But due to expansion, still looks satisfiable *locally*!

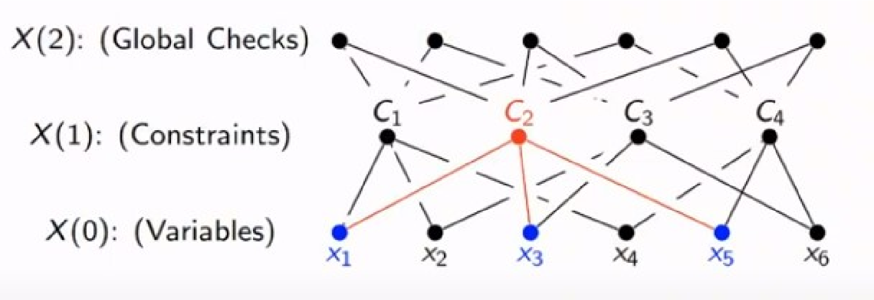
Grigoriev proved that if the underlying bipartite graph is a good expander, and you pick the assignments of β randomly, then this is hard for the SoS paradigm.

You have to go all the way upto linear levels of SoS to be able to find a contradiction. Roughly speaking, why is this the case: Because I'm picking β randomly, if I have a fixed assignment to variables, the probability for each individual constraint, that I'm going to pick the right bit is $\frac{1}{2}$. So by Chernoff bound and union bound you can argue that with a very high probability always about half of the constraints are going to be violated. So this a CSP that is very far from being satisfiable.

Yet, due to expansion, that is the graph doesn't turn in on itself in a way, it's very hard to find contradictions, even though only half of the things are satisfiable.

This doesn't tell us much about the structure of hard instances. Can we find explicit examples?

Idea: Replace randomness with high-dimensional structure [DFHT'21]



We're going to look at stacked bipartite graphs.

Pick $\beta \in \mathbf{F}_2^{X(1)}$ s.t. i. β satisfies all global checks. ii. (X, β) is unsatisfiable.

Hope. If X is an HDX, instance will be hard!

i. [DFHT21] Hardness vs. $O(\sqrt{\log(n)})$-levels via simplicial HDX.

ii. [HL22] Hardness vs. $\Omega(n)$-levels via expanding chain complexes.

Chain Complexes (Math Formalism)

Chain complexes are a generalization of Simplicial Complexes. i. Let $X(0)$, $X(1)$ and $X(2)$ be sets. ii. Let $\delta_0 : \mathbb{F}_2^{X(0)} \to \mathbb{F}_2^{X(1)}$ and δ_1: $\mathbb{F}_2^{X(1)} \to \mathbb{F}_2^{X(2)}$ be linear maps.

Chain Complex. The sequence $X : \mathbb{F}_2^{X(0)} \xrightarrow{\delta_0} \mathbb{F}_2^{X(1)} \xrightarrow{\delta_1} \mathbb{F}_2^{X(2)}$ is called a (3-term) chain complex if $\delta_1\delta_0 = 0$.

$\{\delta_1, \delta_0\}$ are called the co-boundary operators.

B^1: $= \mathrm{Im}(\delta_0)$ are called the 'co-boundaries'.

$Z^1 := \mathrm{Ker}(\delta_1) \supset \mathrm{Im}(\delta_0)$ are called co-cycles.

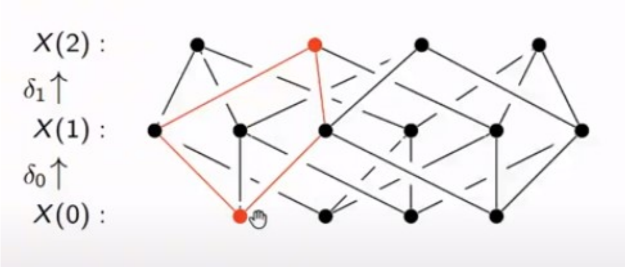
$H^1 := \frac{Z^i}{B^i}$ is called the 'co-homology'.

A set $S \subset X(1)$ 'expands' if $|\partial_1 1_S|$ is large.

The Graphical Interpretation

Chain complexes also have a 'graphical' interpretation.

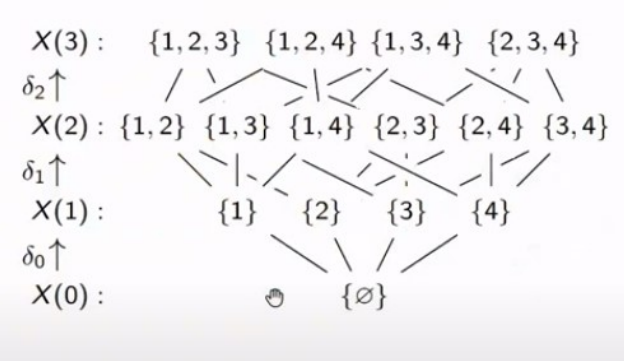
Let $(X(0), X(1), \delta_0)$ and $(X(1), X(2), \delta_1)$ be bipartite graphs.



Graphs form a chain complex if $\delta_1\delta_0 = 0 \pmod 2$. This has a natural combinatorial interpretation:

Number of paths between any $x \in X(0)$ and $y \in X(2)$ is even.

Simplicial complexes are chain complexes.



$X(3)$: $\{1,2,3\}$ $\{1,2,4\}$ $\{1,3,4\}$ $\{2,3,4\}$

$\delta_2\uparrow$

$X(2)$: $\{1,2\}$ $\{1,3\}$ $\{1,4\}$ $\{2,3\}$ $\{2,4\}$ $\{3,4\}$

$\delta_1\uparrow$

$X(1)$: $\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$

$\delta_0\uparrow$

$X(0)$: $\{\varnothing\}$

Number of ways to get from $\tau \in X(i)$ to $\sigma \in X(i+2)$.
- If $\tau \not\subset \sigma$: 0 ways - If $\tau \subset \sigma$: 2 ways

6.2 Boundary expansion in high dimensions

Expansion in complexes

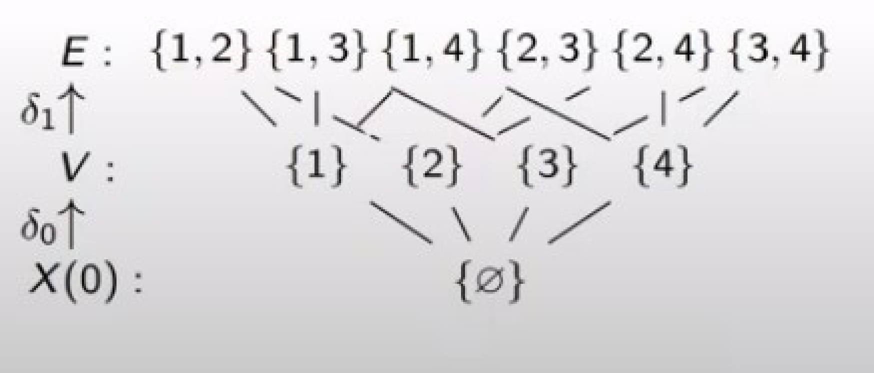
The 'Cheeger constant' of a regular graph G is:

$$\rho_G \propto \min\nolimits_{S\subset V} \frac{|E(S,\overline{S})|}{\min(S,\overline{S})}$$

Another viewpoint: Edge-boundary scales with distance from $\{\varnothing, V\}$:

$$|E(S,\overline{S})| \geq \rho_G \cdot \mathrm{dist}(S, \{\varnothing, V\})$$

Let's view G as a complex again,



E : $\{1,2\}$ $\{1,3\}$ $\{1,4\}$ $\{2,3\}$ $\{2,4\}$ $\{3,4\}$

$\delta_1\uparrow$

V : $\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$

$\delta_0\uparrow$

$X(0)$: $\{\varnothing\}$

i. $|E(S,\overline{S})| = |\delta_1 S|$ (as $\delta_1 1_S(v,w) = 1_S(v) \oplus 1_S(w)$) ii. $\mathrm{dist}(S, \{\varnothing, V\} = \mathrm{dist}(1_S, \mathrm{Im}(\delta_0)))$ as $X(0) = \varnothing$.

ii. $\mathrm{dist}(S, \{\varnothing, V\}) = \mathrm{dist}(1_S, \mathrm{Im}(\delta_0))$ (as $X(0) = \varnothing$).

This suggests the following generalization of Cheeger.

Definition (Coboundary Expansion).

$$X|\mathbb{F}_2^{X(0)} \overset{\delta_0}{\to} \mathbb{F}_2^{X(1)} \overset{\delta_1}{\to} \mathbb{F}_2^{X(2)}$$

is a ρ-coboundary expander if $\forall S \in X(1)$:

$$|\delta_1 1_S| \geq \rho \cdot \mathrm{dist}(1_S, \mathrm{Im}(\partial_0))$$

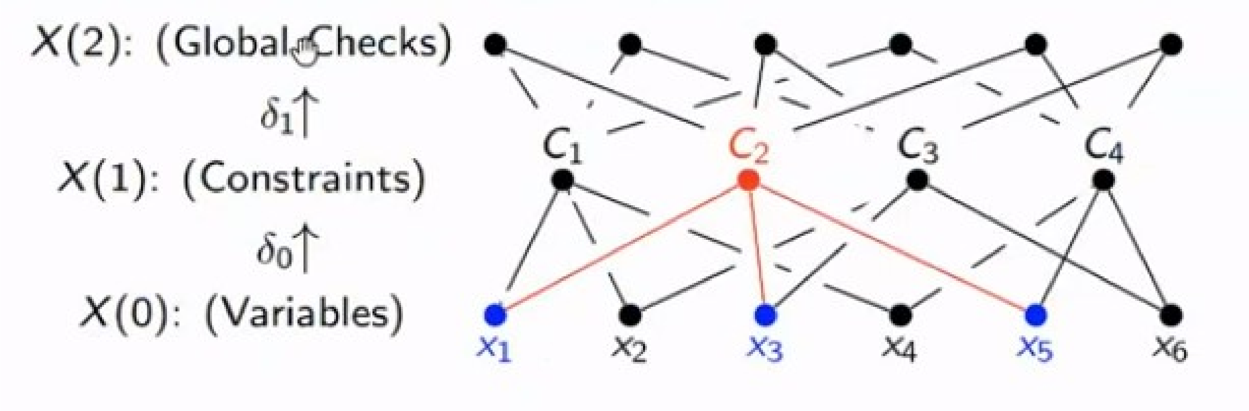
We can also talk about expansion in reverse direction.

$$X|\mathbb{F}_2^{X(0)} \overset{\delta_0^T}{\leftarrow} \mathbb{F}_2^{X(1)} \overset{\delta_1^T}{\leftarrow} \mathbb{F}_2^{X(2)}$$

Complex is a ρ-boundary expander if $\forall S \in X(1)$:
$|\partial_0^T 1_S| \geq \rho \cdot \mathrm{dist}(1_S, \mathrm{Im}(\delta_1^T))$

Revewing the construction

Recall the XOR construction (now in chain complex form).



How do we pick β given $\{\delta_0, \delta_1\}$?
- Satisfiable assignments are exactly $\mathrm{Im}(\delta_0)$
- 'Global structure' given by $\mathrm{Ker}(\delta_1) \supset \mathrm{Im}(\delta_0)$

Choose $\beta \in \mathrm{Ker}(\delta_1) \setminus \mathrm{Im}(\delta_0)$. β can be found efficiently by Gaussian elimination. Co-cycle but not co-boundary. β can be found by Gaussian elimination.

Problem

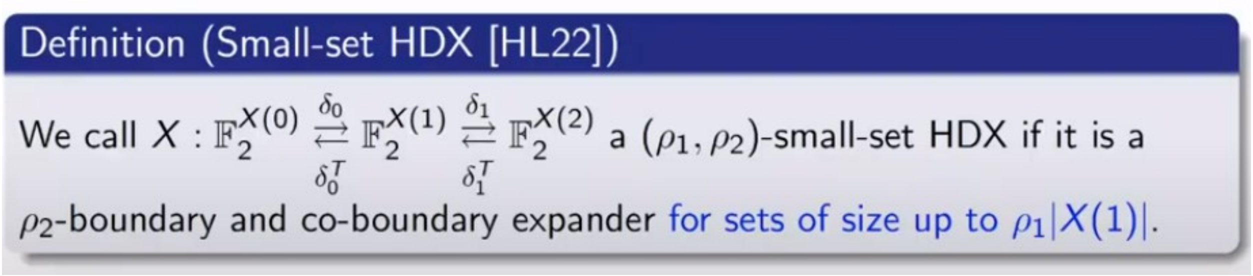
No such β exists! Coboundary expansion is a very very strong property that results in the vanishing of cohomology. That is, the kernel of δ_1 is equal to the image of δ_0. That is, there is no way to pick a function, that is global in some sense, without it being satisfied. This is the major problem with this approach.

Furthermore, we don't know any sparse constructions...

We circumvent these issues by relaxing to small sets.

6.3 Small-set HDX are hard for SoS

Definition (Small-set HDX).

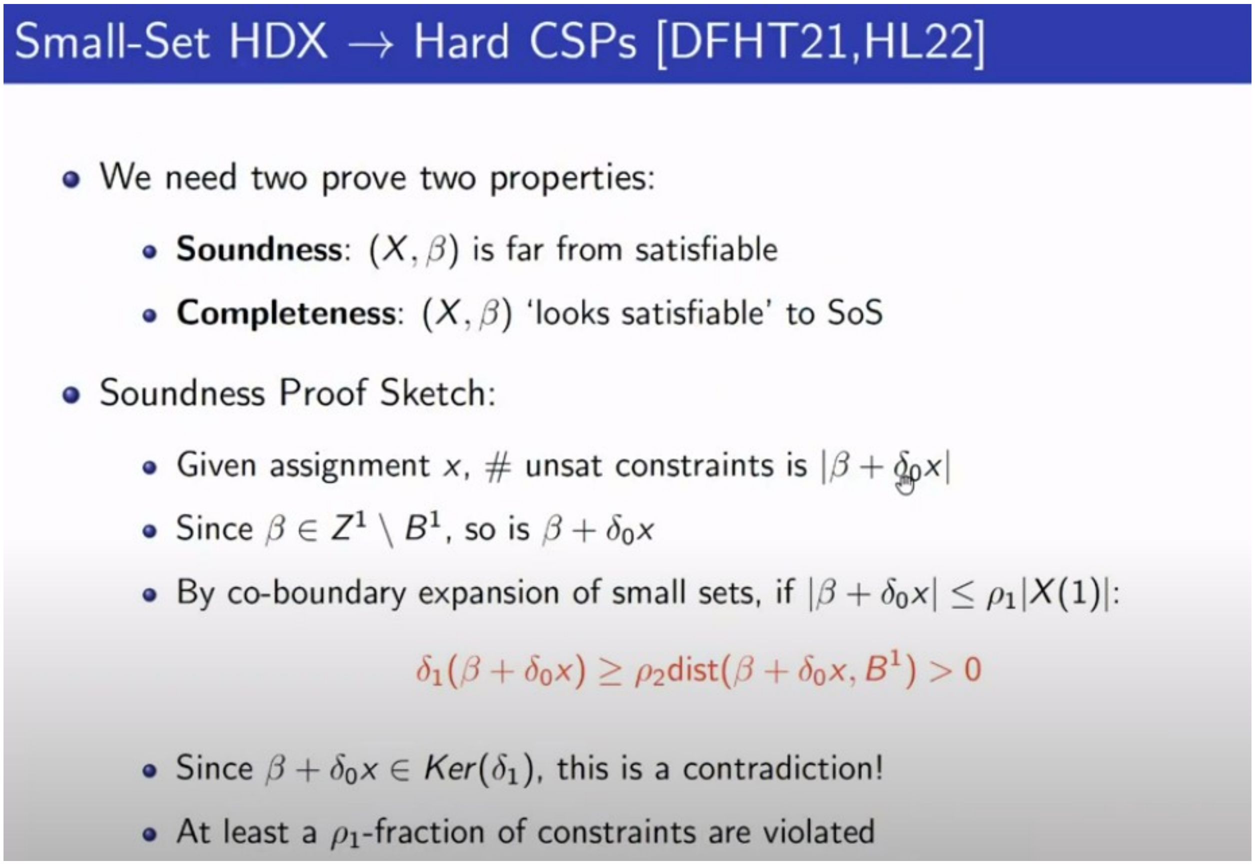


Definition (Small-set HDX [HL22])

We call $X : \mathbb{F}_2^{X(0)} \underset{\delta_0^T}{\overset{\delta_0}{\rightleftarrows}} \mathbb{F}_2^{X(1)} \underset{\delta_1^T}{\overset{\delta_1}{\rightleftarrows}} \mathbb{F}_2^{X(2)}$ a (ρ_1, ρ_2)-small-set HDX if it is a ρ_2-boundary and co-boundary expander for sets of size up to $\rho_1|X(1)|$.

Theorem. Explicit (bounded-degree) small-set HDX exist. [HL22]
- Construction a bit complicated, but come from qLDPC codes [LZ22].
- Worth noting, this implies the NLTS conjecture!

Soundness Proof Sketch



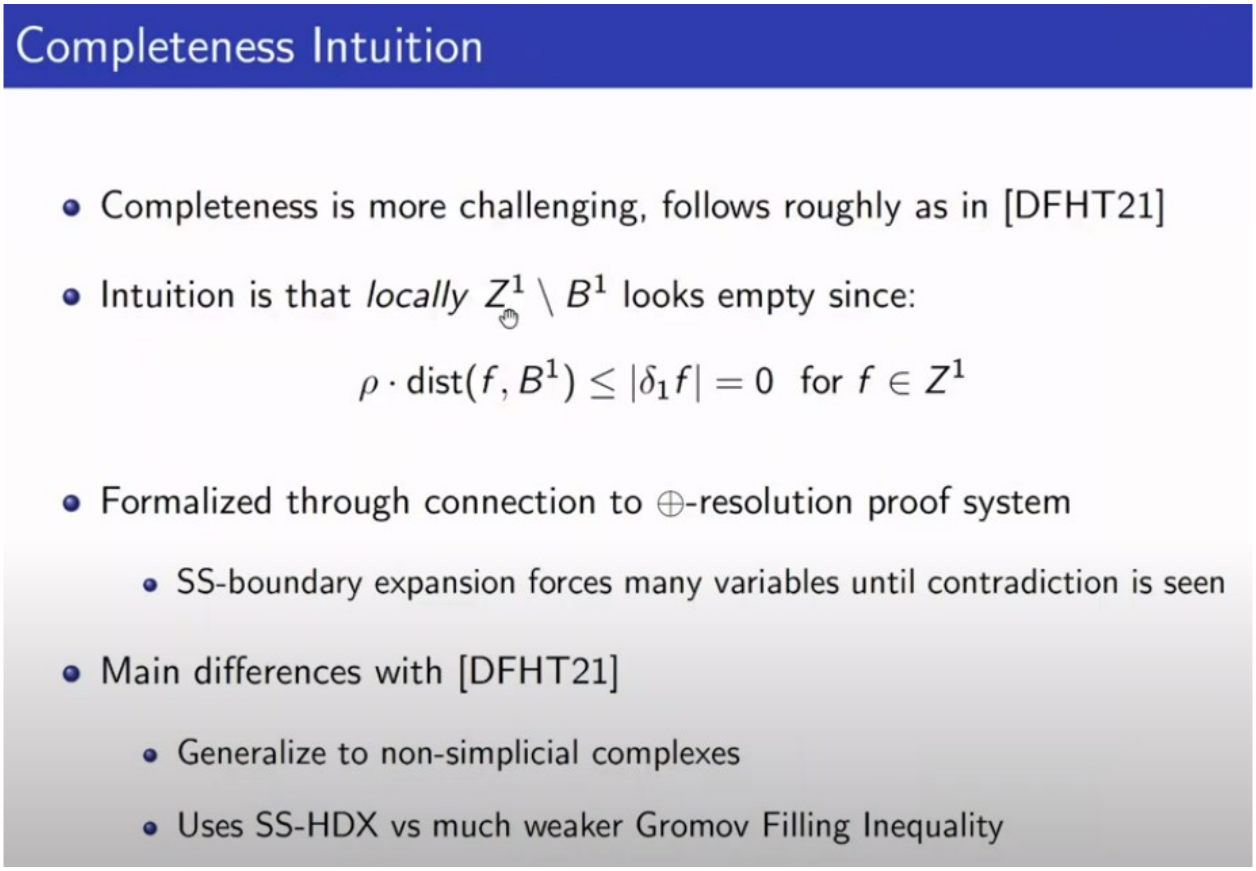
Small-Set HDX → Hard CSPs [DFHT21,HL22]

- We need two prove two properties:
  - **Soundness**: (X, β) is far from satisfiable
  - **Completeness**: (X, β) 'looks satisfiable' to SoS
- Soundness Proof Sketch:
  - Given assignment x, # unsat constraints is $|\beta + \delta_0 x|$
  - Since $\beta \in Z^1 \setminus B^1$, so is $\beta + \delta_0 x$
  - By co-boundary expansion of small sets, if $|\beta + \delta_0 x| \leq \rho_1 |X(1)|$:

$$\delta_1(\beta + \delta_0 x) \geq \rho_2 \mathrm{dist}(\beta + \delta_0 x, B^1) > 0$$

  - Since $\beta + \delta_0 x \in Ker(\delta_1)$, this is a contradiction!
  - At least a ρ_1-fraction of constraints are violated

Completeness Proof Sketch



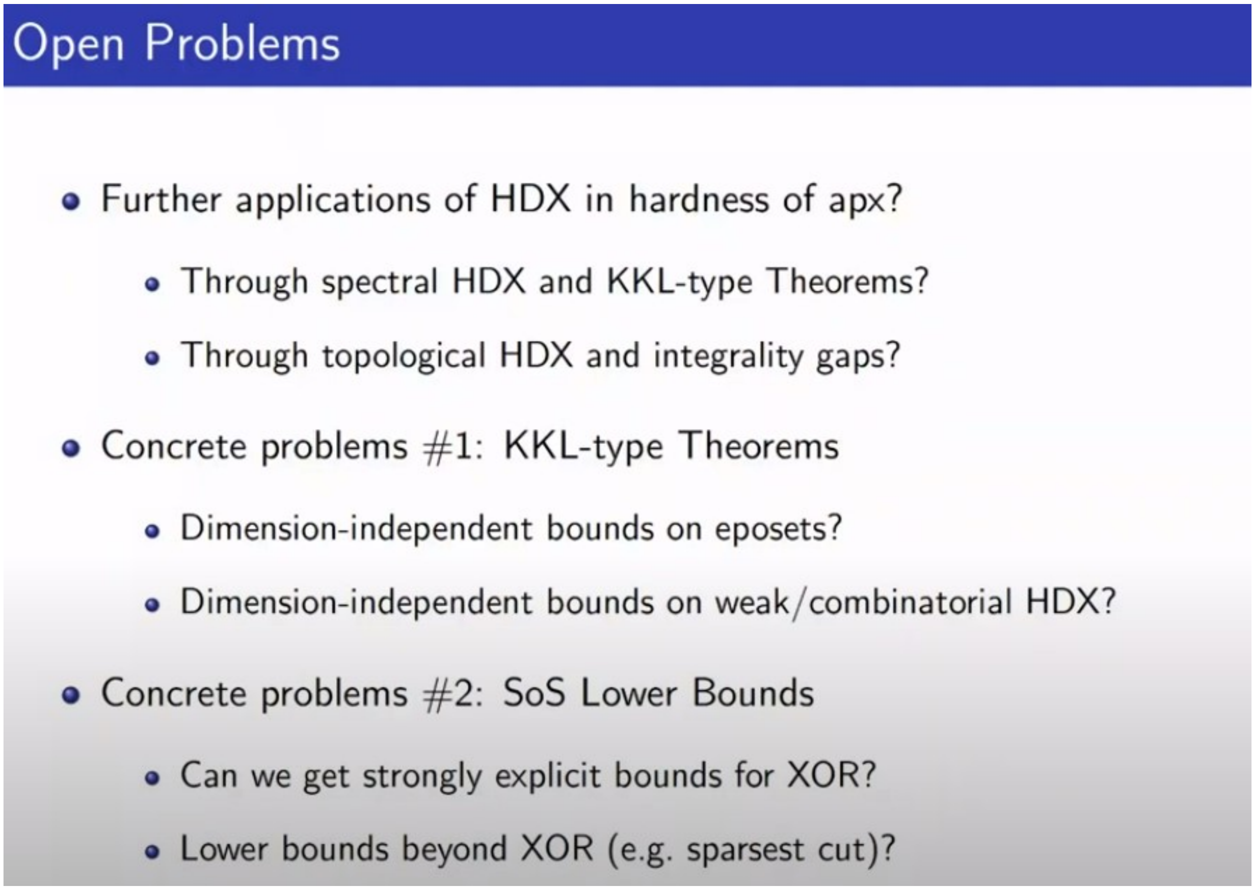
Completeness Intuition

- Completeness is more challenging, follows roughly as in [DFHT21]
- Intuition is that *locally* $Z^1 \setminus B^1$ looks empty since:

$$\rho \cdot \mathrm{dist}(f, B^1) \leq |\delta_1 f| = 0 \quad \text{for } f \in Z^1$$

- Formalized through connection to $\oplus$-resolution proof system
  - SS-boundary expansion forces many variables until contradiction is seen
- Main differences with [DFHT21]
  - Generalize to non-simplicial complexes
  - Uses SS-HDX vs much weaker Gromov Filling Inequality

Open Problems



Open Problems

- Further applications of HDX in hardness of apx?
  - Through spectral HDX and KKL-type Theorems?
  - Through topological HDX and integrality gaps?
- Concrete problems #1: KKL-type Theorems
  - Dimension-independent bounds on eposets?
  - Dimension-independent bounds on weak/combinatorial HDX?
- Concrete problems #2: SoS Lower Bounds
  - Can we get strongly explicit bounds for XOR?
  - Lower bounds beyond XOR (e.g. sparsest cut)?