SoS Lower Bounds, SS-HDX and NLTS

Sanchayan Dutta

UC Davis

June 26, 2023

Table of Contents

NLTS from quantum LDPC codes

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.

NLTS and quantum PCP

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.

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.

NLTS from quantum LDPC codes

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.

NLTS from quantum LDPC codes

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.

QMA-complete local Hamiltonian problem and QPCP

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.

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.

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.

Wait, what is the quantum PCP conjecture?

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.

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 $\mathrm{tr}\left(\mathbf{H}^{(n)}\psi\right) < \epsilon n$, the circuit complexity of the state ψ_n is at least $\Omega(\log n)$.

The Key Theorem

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.

The Key Theorem

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.

The Key Theorem

- **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.

The Key Theorem

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.

What are 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.

What are quantum LDPC codes?

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.

NLTS and quantum LDPC

[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.

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.

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.

A quick review of CSS codes

- 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.

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.

Distance Measure and Approximate Codewords

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. In other words, $G_z^\delta = \{y : |H_z y| \leq \delta m_z\}$ is the set of binary vectors y such that the Hamming weight of $H_z y$ is at most δm_z, where H_z is the matrix defining the code C_z and m_z is the number of rows in H_z. 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.

Distance Measure and Approximate Codewords

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 $|H_z y| \leq \delta m_z$, 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 $H_z y$ refers to the number of nonzero elements in the vector resulting from the matrix-vector multiplication $H_z y$.

Distance Measure and Approximate Codewords

- 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 $|H_x y| \leq \delta m_x$ 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.

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., $y \in G_z^\delta$). It states that such vectors y either have small distance to the orthogonal complement of the code C_x ($|y|_{C_x^\perp} \leq c_1 \delta n$), or they have large distance to it ($|y|_{C_x^\perp} \geq c_2 n$). In other words, the vectors that are close to C_z are either also close to $C_x^\perp$, 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 $C_x^\perp$.

Clustering of Approximate Codewords

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., $y \in G_x^\delta$), and states that such vectors are either close to $C_z^\perp$, 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.

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.

Tanner codes with spectral expansion

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 $|\cdot|_{C_x^\perp}$, 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.

What is spectral expansion?

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.

The terminology behind "spectral"?

"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.

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 $\mathbf{H}_{\mathrm{z}}$.

Local Hamiltonians of CSS codes

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.

A brief review of stabilizer codes

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.

A brief review of stabilizer codes

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.

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.

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.

Sum-of-Squares (SoS) and CSPs

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.

Sum-of-Squares (SoS) and CSPs

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.

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.

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.

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

2. No instance can be disproven (refuted) by using $\mu_{@}n$ 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.

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.

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.

Explicit family of 3-XORs

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.

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).

Inapproximability

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.

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.

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.

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.

What is Theorem 1.1 doing for us?

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.

What is Theorem 1.1 doing for us?

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.

What is Theorem 1.1 doing for us?

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.

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.

Approximation-resistant predicates based on pairwise independent subgroups

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.

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.

Small-set High Dimensional Expanders (SS-HDX)

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.

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.

Small-set High Dimensional Expanders (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).

Small-set High Dimensional Expanders (SS-HDX)

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.

Small-set High Dimensional Expanders (SS-HDX)

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.

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 $f \in \mathrm{im}(\delta_0)$ (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:

$$\forall f \in \mathbb{F}_2^{X(1)} : |\delta_1 f| \geqslant \rho \cdot d(f, \mathrm{im}(\delta_0)).$$

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

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).

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

$$X : \mathbb{F}_2^{\varnothing} \underset{\partial_1}{\overset{\delta_0}{\rightleftarrows}} \mathbb{F}_2^{V} \underset{\partial_2}{\overset{\delta_1}{\rightleftarrows}} \mathbb{F}_2^{E}$$

For intuition, it is worth briefly discussing why this generalizes boundary expansion on graphs. Any graph $G = (V, E)$ (or indeed hypergraph, see Section 4.2) can be written as a chain complex:

$$X : \mathbb{F}_2^{\varnothing} \underset{\partial_1}{\overset{\delta_0}{\rightleftarrows}} \mathbb{F}_2^{V} \underset{\partial_2}{\overset{\delta_1}{\rightleftarrows}} \mathbb{F}_2^{E}$$

Generalizing from graphs to chain complexes

where $\delta_0 f(v) = f(\varnothing), \delta_1 f((u,v)) = f(u) \oplus f(v)$, *and it is easily checked that* $\delta_1 \delta_0 = 0$. *Notice that in this setting the only co-boundaries are* $\mathrm{im}(\delta_0) = \{\varnothing, V\}$, *and furthermore that for any* $S \subset V$ *and* $e \in E$, *the value of* $\delta_1 1_S$ *on* e *is 1 iff* e *crosses the cut defined by* S. *This implies the ratio* $\frac{|\delta_1 1_S|}{d(1_S, \mathrm{im}(\delta_0))} = \frac{E(S, V \setminus S)}{\min\{|S|, |V|S|\}}$, *which is just the standard boundary expansion of* G!

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

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.

Generalizing from graphs to chain complexes

Notice that in this setup, the only co-boundaries are $\mathrm{im}(\delta_0) = \{\varnothing, V\}$. 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 $\frac{|\delta_1 1_S|}{d(1_S, \mathrm{im}(\delta_0))}$ 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.

Generalizing from graphs to chain complexes

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. $\frac{|\delta_1 1_S|}{d(1_S, \mathrm{im}(\delta_0))}$: 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.

Generalizing from graphs to chain complexes

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, $d(1_S, \mathrm{im}(\delta_0))$ 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.

Generalizing from graphs to chain complexes

To see why these two quantities might be related, consider what they represent. $d(1_S, \mathrm{im}(\delta_0))$ 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.

Generalizing from graphs to chain complexes

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.

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:

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

The notion of small-set boundary expander

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.

The notion of small-set boundary expander

Mathematically, this property can be expressed as:

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

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.

The notion of small-set boundary expander

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.

The notion of small-set boundary expander

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.

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:

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

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

Small-set coboundary expander

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:

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

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.

The 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.

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.

Constructing an infinite family of SS-HDX

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.

Constructing an infinite family of SS-HDX

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.

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.