Unique Games hardness of Quantum Max-Cut, and a conjectured vector-valued Borell’s inequality

Yeongwoo Hwang
Harvard

Joe Neeman
UT Austin

Ojas Parekh
Kevin Thompson
Sandia

John Wright
UC Berkeley
(Classical) Max-Cut

\[ G = (V, E) \]

7 edges cut
(Classical) Max-Cut

$G = (V, E)$

8 edges cut (the max cut)
(Classical) Max-Cut

Goal: find partition \( f: V \rightarrow \{+1, -1\} \) maximizing

\[
\sum_{(u,v) \in E} \left( \frac{1 - f(u) \cdot f(v)}{2} \right)
\]

NP-hard to solve exactly!

So instead look for approximation algorithms.

An algorithm achieves an \( \alpha \)-approximation if it outputs a partition \( f \) such that

\[
\text{# of edges } f \text{ cuts} \geq \alpha \cdot \text{Max-Cut}(G).
\]
Approximating Max-Cut

[Goemans-Williamson]: Poly-time algorithm achieving a $0.878 \ldots$ approximation

Based on a technique known as **semidefinite programming** (SDP)

Powerful & versatile framework for designing algorithms

Will describe later!

[KKMO, MOO]: **GW** is the **optimal** poly-time algorithm.
(assuming **Unique Games Conjecture (UGC)**)
Constraint satisfaction problems

Max-Cut is e.g. of a constraint satisfaction problem (CSP)

**CSP**: set of variables with constraints on them

**Examples**: Max-3Sat, Max-3Coloring, Max-3XOR

For every CSP, there is a canonical SDP aka the basic SDP

[Raghavendra]: The basic SDP is the optimal poly-time alg (assuming UGC)

**Complete** understanding of approximability of CSPs. (modulo the UGC)

Beautiful theory!
The local Hamiltonian problem

Like quantum version of CSPs

Input: a physical system that looks like

Output: the “ground state” of the system
or the “ground state energy” of the system

Very important problem in physics!
The local Hamiltonian problem

Like **quantum** version of CSPs

**Input:**
1. A graph $G = (V, E)$ on $n$ vertices
2. A Hamiltonian $H$ on $n$ qubits:

   $h_{uv} \in \mathbb{R}$ for $u \neq v$, $h_{uu} = 0$

$H = \sum_{(u,v) \in E} h_{uv} \otimes I_{V \setminus \{u,v\}}$

**Output:**

- a quantum state $|\psi\rangle$ on $n$ qubits with the **maximum** energy $\lambda_{\text{max}}(H)$

  $\text{energy} = \langle \psi | H | \psi \rangle$
Problems with the LHP

Much less understood than classical CSPs!

We do not have:

• a theory of optimal algorithms
• a good understanding of the power of SDPs

like classical CSPs, there is a basic SDP

how well does it do?

is it also the optimal algorithm?

• a quantum version of the PCP theorem

(quantum PCP conjecture is still a conjecture)
Problems with the LHP part 2: the ansatz

Supposed to output quantum state $|\psi\rangle$. What does that mean?

If algorithm is classical, should output classical description of $|\psi\rangle$.

$|\psi\rangle$ must be efficiently describable!

E.g. product states $|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$

Def: An ansatz is a family of quantum states which is efficiently describable.

Question: What is the optimal ansatz for approximation algorithms?
Quantum Max-Cut

Special case of 2-local Hamiltonian:

\[ H_G = \sum_{(u,v) \in E} \frac{1}{4} \cdot (I - X_uX_v - Y_uY_v - Z_uZ_v) \]

Goal: Output the **maximum energy state** of \( H_G \)

Note: max energy state of \( H_G \) = **min** energy state of \( \sum_{(u,v) \in E} (X_uX_v + Y_uY_v + Z_uZ_v) \)

(antiferromagnetic) **Heisenberg model**

Dates back to [Heisenberg 1928]

Well-studied class of Hamiltonians
Quantum Max-Cut

Special case of 2-local Hamiltonian:

\[ H_G = \sum_{(u,v) \in E} \frac{1}{4} \cdot (I - X_u X_v - Y_u Y_v - Z_u Z_v) \]

Intuition

Term 1: Does nothing

Term 2: Measure in X basis
- \(-1\) if same (+ + or − −)
- \(+1\) if different (+ − or − +)

want both different!

\[ |\psi\rangle (n \text{ qubits}) \]
Quantum Max-Cut

Special case of 2-local Hamiltonian:

\[
H_G = \sum_{(u,v) \in E} \frac{1}{4} \cdot (I - X_u X_v - Y_u Y_v - Z_u Z_v)
\]

**Intuition**

**Term 1:** Does nothing

**Term 2:** Should be different in \(X\) basis

**Term 3:** Should be different in \(Y\) basis

**Term 4:** Should be different in \(Z\) basis

Like *(classical)* Max-Cut in \(X\), \(Y\), and \(Z\) bases!
Approximating Quantum Max-Cut

[CM16,PM17]: Given a Quantum Max-Cut instance $H_G$, estimating $\lambda_{\text{max}}(H_G)$ to error $\pm 1/poly(n)$ is QMA-hard.

[Gharibian-Parekh 2019]: $0.498$-approximation

**Ansatz:** product states $|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$

[Anshu-Gosset-Morenz 2020]: $0.531$-approximation

**Ansatz:** tensor products of one- and two-qubit states

$|\psi\rangle = |\psi_{12}\rangle \otimes |\psi_3\rangle \otimes \cdots \otimes |\psi_n\rangle$

[Parekh-Thompson 2020]: $0.533$-approximation

**Ansatz:** same as [AGM20]
Approximating Quantum Max-Cut

[CM16,PM17]: Given a Quantum Max-Cut instance $H_G$, estimating $\lambda_{\text{max}}(H_G)$ to error $\pm 1/\text{poly}(n)$ is QMA-hard.

[Gharibian-Parekh 2019]: 0.498-approximation

[Anshu-Gosset-Morenz 2020]: 0.531-approximation

[Parekh-Thompson 2020]: 0.533-approximation

[Lee 2022]: 0.562-approximation

[King 2022]: 0.582-approximation*

Ansatz: something more complicated

*only works for triangle-free graphs
Approximating Quantum Max-Cut

[CM16,PM17]: Given a Quantum Max-Cut instance $H_G$, estimating $\lambda_{\text{max}}(H_G)$ to error $\pm 1/\text{poly}(n)$ is QMA-hard.

[Gharibian-Parekh 2019]: 0.498-approximation uses the basic SDP for Quantum Max-Cut

[Anshu-Gosset-Morenz 2020]: 0.531-approximation

[Parekh-Thompson 2020]: 0.533-approximation

[Lee 2022]: 0.562-approximation

[King 2022]: 0.582-approximation*

do not use the basic SDP
Our motivation

[GP] uses **basic SDP** for Quantum Max-Cut

But [AGM], [PT], [L], and [K] outperform it

Classically, the basic SDP is **always** optimal

Originally, I figured we could just design a **better** SDP algorithm

Maybe product states are a **bad** ansatz?

So we spent some time working on it…
Our results

**Thm:** The Gharibian Parekh algorithm is the **optimal** algorithm using the basic SDP.

Any alg using basic SDP has approx ratio $\leq 0.498 \ldots$

*(show this via an integrality gap)*

**Thm:** It is **NP**-hard to achieve a $0.956 + \epsilon$ approximation for Quantum Max-Cut.

*(assuming UGC)*

Far from the best algorithm (0.562)

Seems to be the limit of current techniques

Can we improve either bound?
Our results

**Thm:** The Gharibian Parekh algorithm is the **optimal** algorithm using the basic SDP. *

**Aka:** Any alg using basic SDP has approx ratio \( \leq 0.498 \ldots \)

(\textit{show this via an integrality gap})

**Thm:** It is **NP**-hard to achieve a \( 0.956 + \epsilon \) \hspace{1cm} (assuming UGC)

approximation for Quantum Max-Cut. *

* Assuming a conjecture in Gaussian geometry
The “**vector-valued Borell’s conjecture**”
We don’t know how to prove it!
Will explain later...
Product states for QMax-Cut

States of the form $|\psi\rangle = \bigotimes_{u \in V} |\psi_u\rangle$

$n$ qubits: 〇 〇 〇 〇 〇 〇 〇 〇 〇 〇

$|\psi_u\rangle$ $|\psi_v\rangle$

Product states possess no entanglement

But they can often be close to the ground state!

[Brandao Harrow 2016]: The ground state is close to product if $G$ is high degree.
Product states for QMax-Cut

States of the form \( |\psi\rangle = \bigotimes_{u \in V} |\psi_u\rangle \)

\( f: \)

\( n \) qubits: \( \bigotimes_{u \in V} |\psi_u\rangle \)

Useful to look at \textbf{Bloch sphere} representation:

\[ |\psi_u\rangle\langle\psi_u| = \frac{1}{2} \cdot (I + c_X \cdot X + c_Y \cdot Y + c_Z \cdot Z) \]

\( |\psi_u\rangle \) is pure \( \Rightarrow c_X^2 + c_Y^2 + c_Z^2 = 1 \)

Set \( f(u) = (c_X, c_Y, c_Z) \). Then \( f: V \to S^2 \).
Product states for QMax-Cut

States of the form $|\psi\rangle = \bigotimes_{u \in V} |\psi_u\rangle$

**Bloch sphere** representation: $f: V \rightarrow S^2$.

Product state energy formula:

$$
\langle \psi | H_G | \psi \rangle = \sum_{(u,v) \in E} \left( \frac{1 - \langle f(u), f(v) \rangle}{4} \right)
$$

"Want" neighboring $f(u)$ and $f(v)$ to point in opposite directions.

Like *(classical)* Max-Cut! There, $f: V \rightarrow \{\pm 1\} = S^0$. 
Product state results

Also consider the problem of finding the best product state

[BOV]: Algorithm outputting a product state $|\psi\rangle$ with energy $\geq 0.956 \times$ (best product state energy).

(uses basic SDP)

[Us]: 1. This is the optimal alg using the basic SDP. *

2. This is the optimal poly-time alg. (assuming UGC)

Aka it NP-hard to achieve a $0.956 + \epsilon$ approx. *

Same NP-hardness as for general QMax-Cut!

* Assuming the same conjecture as before
Outline for rest of talk

**Goal:** explain this result:

Any alg for Quantum Max-Cut using basic SDP has approx ratio \( \leq 0.498 \ldots \)

**Outline:**

1. Goemans-Williamson algorithm
2. Gharibian-Parekh algorithm
3. Why GW is optimal alg for basic SDP  
   (uses Borell’s isoperimetric inequality)
4. Why GP is optimal alg for basic SDP  
   (assuming vector-valued Borell’s conjecture)
Goemans-Williamson algorithm
Goemans-Williamson algorithm

**Goal:** maximize

\[
\sum_{(u,v) \in E} \left( \frac{1 - f(u) \cdot f(v)}{2} \right)
\]

where \( f(u) \in \{\pm 1\} \)

Can’t optimize over this!

*(NP-hard)*
Goemans-Williamson algorithm

**Goal:** maximize

\[
\sum_{(u,v) \in E} \left( \frac{1 - \langle f(u), f(v) \rangle}{2} \right)
\]

where \( f(u) \in (\pm 1, 0, \ldots, 0) \)

Still can’t optimize over this!
**Goemans-Williamson algorithm**

**Goal:** maximize

\[
\sum_{(u,v) \in E} \left( \frac{1 - \langle f(u), f(v) \rangle}{2} \right)
\]

where \( f(u) \) unit vector \( \in \mathbb{R}^n \)

**Can** optimize over this!

Known as the Max-Cut **semidefinite program** (SDP)

Let \( \text{SDP}_{\text{MC}}(G) \) be the value of the best vectors

**Note:** \( \text{SDP}_{\text{MC}}(G) \geq \text{Max-Cut}(G) \)
Goemans-Williamson algorithm

Step 1: Compute the optimal SDP vectors

\[ f(u), f(v), f(w) \]

Step 2: “Round” \( f: V \to S^{n-1} \) into a partition \( g: V \to \{\pm 1\} \)
Goemans-Williamson algorithm

Step 1: Compute the optimal SDP vectors

\( \mathbf{f}(\mathbf{u}) \)  
\( \mathbf{f}(\mathbf{v}) \)  
\( \mathbf{f}(\mathbf{w}) \)

(a) Pick a random vector \( \mathbf{z} \in \mathbb{R}^n \)

(b) Set \( g(\mathbf{u}) = \text{sign}(\langle \mathbf{z}, \mathbf{f}(\mathbf{u}) \rangle) \), for all \( \mathbf{u} \in V \)

[GW]: Achieves a \( 0.878 \ldots \) approximation
Analyzing the GW algorithm

**Step 1:** Compute the optimal SDP vectors

**Step 2:** “Round” $f: V \rightarrow S^{n-1}$ into a partition $g: V \rightarrow \{\pm 1\}$

**Goal:** $\mathbb{E}[\text{# edges cut by } g] \geq 0.878 \cdot (\text{SDP value of } f)$

(recall) $0.878 \cdot \text{Max-Cut}(G)$
Analyzing the GW algorithm

**Step 1:** Compute the optimal SDP vectors

**Step 2:** “Round” $f: V \rightarrow S^{n-1}$ into a partition $g: V \rightarrow \{\pm 1\}$

**Goal:** $E[\text{# edges cut by } g] \geq 0.878 \cdot \text{(SDP value of } f\text{)}$

\[
\sum_{(u,v) \in E} \left( \frac{1 - g(u) \cdot g(v)}{2} \right)
\]

\[
\sum_{(u,v) \in E} \left( \frac{1 - \langle f(u), f(v) \rangle}{2} \right)
\]

Goal inequality is true for each edge!

**Key lemma:**

\[
\forall u, v: E[1 - g(u) \cdot g(v)] \geq 0.878 \cdot (1 - \langle f(u), f(v) \rangle)
\]
Analyzing the GW algorithm

Key lemma:
\[ \forall u, v: \mathbb{E}[1 - g(u) \cdot g(v)] \geq 0.878 \cdot (1 - \rho_{uv}) \]

**Pf:** Set \( \rho_{uv} = \langle f(u), f(v) \rangle \).

LHS is function only of \( \rho_{uv} \) too.

Want to compute

\[
\min_{-1 \leq \rho_{uv} \leq 1} \mathbb{E}[1 - g(u) \cdot g(v)] \\
(1 - \rho_{uv})
\]

Equal to 0.878 (when \( \rho_{uv} = -0.69 \ldots \)). \( \square \)

**Note:** \( g \)'s value on edge \((u, v)\)

is only 0.878 \( \times \) (\( f \)'s value)

if \( \rho_{uv} \) is the “bad angle” \(-0.69 \ldots \)
Gharibian-Parekh algorithm
Back to **Quantum Max-Cut**

**Goal:** Gharibian-Parekh *0.498*-approximation algorithm

**Recall:**
1. Based on **semidefinite programming**
2. Outputs product state $|\psi\rangle = |\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle$
   (aka a Bloch sphere assignment $g: V \to S^2$)
Back to **Quantum Max-Cut**

**Goal:** Gharibian-Parekh 0.498-approximation algorithm

**Recall:** 1. Based on *semidefinite programming*

For every optimization problem, there is a *canonical* SDP

a) Gives an upper bound to the true optimum
b) Can be efficiently solved
c) Probably involves some vectors

For Quantum Max-Cut, a bit complicated

But you can simplify it...
Back to **Quantum Max-Cut**

**Goal:** Gharibian-Parekh 0.498-approximation algorithm

**Recall:** 1. Based on **semidefinite programming**

Given a graph $G$ and Hamiltonian $H_G$, the SDP is:

$$\text{SDP}_{\text{QMC}}(G) = \max \sum_{(u,v) \in E} \left( \frac{1 - 3 \cdot \langle f(u), f(v) \rangle}{4} \right)$$

s. t. $f: V \rightarrow S^{n-1}$

Very similar to Max-Cut SDP!

Optimizing $f: V \rightarrow S^{n-1}$ is same in both!
Back to **Quantum Max-Cut**

**Goal:** Gharibian-Parekh 0.498-approximation algorithm

**Recall:**
1. Based on *semidefinite programming*

Given a graph $G$ and Hamiltonian $H_G$, the SDP is:

$$\text{SDP}_{\text{QMC}}(G) = \max \sum_{(u,v) \in E} \left( \frac{1 - 3 \cdot \langle f(u), f(v) \rangle}{4} \right)$$

s. t. $f: V \to S^{n-1}$

Gives an upper bound to the true optimum:

$$\text{QMax-Cut}(G) \leq \text{SDP}_{\text{QMC}}(G)$$

Next step: rounding this into a Bloch sphere assignment $g: V \to S^2$.

How to do this???
Step 1: Compute the optimal SDP vectors $\mathbf{f}(\mathbf{u}), \mathbf{f}(\mathbf{v})$.

(a) Pick random 3-dimensional projector $\Pi \in \mathbb{R}^{3 \times n}$ random, orthonormal

$$\Pi = \begin{bmatrix} x & y & z \end{bmatrix},$$

where $x, y, z \in \mathbb{R}^n$ are random.

Step 2: "Round" $f$ into a Bloch sphere assignment $g: V \rightarrow S^2$.
Gharibian-Parekh algorithm

Step 1: Compute the optimal SDP vectors

\[ f(u) \]
\[ f(v) \]

(a) Pick random 3-dimensional projector \( \Pi \in \mathbb{R}^{3 \times n} \)

(b) For all \( u \in V \), set \( g(u) = \frac{\Pi \cdot f(u)}{\|\Pi \cdot f(u)\|_2} \).

Step 2: “Round” \( f \) into a Bloch sphere assignment \( g : V \to S^2 \)
**Gharibian-Parekh algorithm**

**Step 1:** Compute the optimal SDP vectors

(a) Pick random 3-dimensional projector \( \Pi \in \mathbb{R}^{3 \times n} \)

(b) For all \( u \in V \), set \( g(u) = \Pi \cdot f(u)/\|\Pi \cdot f(u)\|_2 \).

[GP]: Achieves a **0.498** ... approximation

**Step 2:** “Round” \( f \) into a Bloch sphere assignment \( g: V \to S^2 \)

(a) Pick random 3-dimensional projector \( \Pi \in \mathbb{R}^{3 \times n} \)

(b) For all \( u \in V \), set \( g(u) = \Pi \cdot f(u)/\|\Pi \cdot f(u)\|_2 \).
Analyzing the GP algorithm

Similar to the GW analysis

Boils down to a per edge inequality:

Key lemma:
\[ \forall u, v: \mathbb{E}[1 - \langle g(u), g(v) \rangle] \geq 0.498 \cdot (1 - 3 \cdot \rho_{uv}) \]

inequality is tight if \( \rho_{uv} = -0.97 \ldots \)

this is the “bad angle” for GP algorithm

for these edges, \( g \)’s value is only \( 0.498 \times (f \)’s value)
Why GW is optimal alg for the basic SDP
GW analysis

Wanted to show: \( \alpha \cdot \text{Max-Cut}(G) \leq E[\# \text{ edges cut by } g] \)
\[\alpha = 0.878 \ldots\]

Actually showed: \( \alpha \cdot \text{SDP}_{MC}(G) \leq E[\# \text{ edges cut by } g] \leq \text{Max-Cut}(G) \)

Stronger statement!

"Standard analysis": comparing \( g \) to SDP value

Note: \( \alpha \leq \min_G \left\{ \frac{\text{Max-Cut}(G)}{\text{SDP}_{MC}(G)} \right\} \)

"Integrality gap" of SDP

Integrality gap provides limitation on any algorithm analyzed via "standard analysis"
Integrality gap for Max-Cut

[FS]: The basic SDP for Max-Cut has integrality gap $0.878 \ldots$

∴ Any algorithm which rounds basic SDP has approx. ratio $\leq 0.878 \ldots$

(according to standard analysis)

[Raghavendra]: For any CSP, the optimal poly-time approx. ratio is equal to the integrality gap of basic SDP.

(assuming UGC)

Can’t beat the standard analysis!
Integrality gap for Max-Cut

[FS]: The basic SDP for Max-Cut has integrality gap $0.878 \ldots$

Pf: Want a graph $G = (V, E)$ satisfying the following:

1. $\text{Max-Cut}(G) = 0.878 \cdot \text{SDP}_{\text{MC}}(G)$
2. $\text{Max-Cut}(G) = E[\# \text{ edges cut by } g]$

(Since $E[\# \text{ edges cut by } g] \geq 0.878 \cdot \text{SDP}_{\text{MC}}(G)$)
Integrality gap for Max-Cut

[FS]: The basic SDP for Max-Cut has integrality gap $0.878 \ldots$

Pf: Want a graph $G = (V, E)$ satisfying the following:

1. $\text{Max-Cut}(G) = 0.878 \cdot \text{SDP}_{MC}(G)$
2. $\text{Max-Cut}(G) = \mathbb{E}[\# \text{edges cut by } g]$  
   GW algorithm always outputs optimal cut!
3. For each edge $(u, v)$,  
   $$\rho_{uv} = \langle f(u), f(v) \rangle$$
   equals the "bad angle" $\rho_{uv} = -0.69$

(Necessary for GW to lose factor of $0.878 \ldots$ on each edge)
Gaussian graph

Vertex set:

\[ V = \mathbb{R}^n \]

Random edge:

\((u, v)\), where \( u \sim \rho v \) are \( \rho \)-correlated Gaussians

\(\rho = -0.69 \ldots\)
**Gaussian graph**

**Vertex set:**

$$\mathcal{V} = \mathbb{R}^n$$

**Random edge:**

$$(u, v), \text{ where } u \sim \rho v \text{ are } \rho\text{-correlated Gaussians}$$

$$(\rho = -0.69 \ldots)$$

**SDP solution** $f : \mathcal{V} \rightarrow \mathbb{R}^n$:

1. $f(u) = u / \|u\|_2$
2. $\langle f(u), f(v) \rangle \approx \rho$

- GW loses $0.878 \ldots$ on each edge
- Need to show GW is **optimal**
Optimality of GW

Value of solution: $E_{u \sim \rho v} \left( \frac{1 - g(u) \cdot g(v)}{2} \right)$
Optimality of GW

Value of solution: \( E_{u \sim \rho v} \left[ \left( \frac{1 - \text{sign}(\langle u, r \rangle) \cdot \text{sign}(\langle v, r \rangle)}{2} \right) \right] \)

• Rotationally symmetric
• Can pick \( r = (1, 0, ..., 0) \) WOLOG
Optimality of GW

Value of solution: \[ E_{u \sim \rho \nu} \left[ \left( \frac{1 - \text{sign}(u_1) \cdot \text{sign}(\nu_1)}{2} \right) \right] \]

- Rotationally symmetric
- Can pick \( r = (1, 0, \ldots, 0) \) WOLOG
Optimality of GW

Value of solution: $E_{u \sim \rho, v} \left[ \frac{1 - \text{sign}(u_1) \cdot \text{sign}(v_1)}{2} \right]$

Want this to be optimal partition.

Should be at least: $E_{u \sim \rho, v} \left[ \frac{1 - h(u) \cdot h(v)}{2} \right]$, for all partitions $h: V \rightarrow \{\pm 1\}$
Borell’s theorem

First proven in [Borell 1985]

Basic result in Gaussian geometry
Why GP is optimal alg for the basic SDP
Integrality gap for Quantum Max-Cut

[Us]: The basic SDP for Quantum Max-Cut has integrality gap 0.498 ...

Same graph $G$: $\rho$-correlated Gaussian graph,

$\rho$ is the “bad angle” $\rho = -0.97$ ...

$\Rightarrow$ GP algorithm loses factor of 0.498

Goal: GP algorithm produces optimal solution
Optimality of GP

**GP** picks random 3-dimensional projector $\Pi$

Sets $g(u) = \Pi \cdot u / \|\Pi \cdot u\|_2$ for each $u \in \mathbb{R}^n$

Value of solution:

$$E_{z \sim p(y)}\left[ \left( 1 - \frac{\langle g(u), g(v) \rangle}{4} \right) \right]$$

- Rotationally symmetric
- Can pick $\Pi =$ projector onto $(u_1, u_2, u_3)$

$$\Rightarrow g(u) = (u_1, u_2, u_3) / \|(u_1, u_2, u_3)\|_2$$
Optimality of GP

**GP** picks random 3-dimensional projector $\Pi$

Sets $g(u) = \Pi \cdot u / \|\Pi \cdot u\|_2$ for each $u \in \mathbb{R}^n$

Value of solution:

$$E_{u \sim \rho v} \left[ \left( \frac{1 - \langle g(u), g(v) \rangle}{4} \right) \right]$$

Want this to be optimal quantum state.

This is a product state (Bloch sphere rep.).

Need to compare against all quantum states

However, $G$ is high degree $\Rightarrow$ ground state is **product**

[Brandao Harrow 2016]

Only need to compare to product states $h: \mathbb{R}^n \rightarrow S^2$
A vector-valued Borell’s conjecture

For all \( k \geq 1 \), the quantity

\[
E_{u \sim \nu} \left[ \langle h(u), h(v) \rangle \right]
\]

is minimized by \( h(u) = (u_1, ... , u_k)/\|u_1, ... , u_k\|_2 \)

We \textbf{thought} we had proven this!

(but Steve Heilman found a \textbf{bug} in our proof)

What we \textbf{can} show:

1. Show that \( h \) is “essentially \( k \) dimensional” (\textbf{positive} \( \rho \))
   (via calculus of variations)

2. Prove it in the case of \( n = k \) (\textbf{negative} \( \rho \))
   (via Fourier analysis)
Conclusion
Open problems

1. Prove the vector-valued Borell’s conjecture!
2. Improve our UG-hardness past 0.956
3. Improve the best approx. ratio past 0.562 [Lee]
4. The [PT] algorithm uses degree-4 Sum of Squares (SoS)
   (more powerful SDP than the basic SDP)
   Can we understand the power of SoS?
5. Algorithmic gap for the Gharibian-Parekh algorithm:
   Graph $G = (V, E)$ s.t.
   • $\text{QMax-Cut}(G) = \nu$
   • GP alg produces state with energy $0.498 \cdot \nu$
   (Or maybe “standard analysis” is not tight?)
Thanks!