INTRODUCTION TO MATRIX RIGIDITY

PART 2

Sasha Golovnev
December 14, 2020
OUTLINE

• Reminder of Part 1
Outline

- Reminder of Part 1
  - Explicit Constructions
Outline

• Reminder of Part 1
  • Explicit Constructions
  • Semi-explicit Constructions
Outline

- Reminder of Part 1
  - Explicit Constructions
  - Semi-explicit Constructions

- Limitations
Outline

• Reminder of Part 1
  • Explicit Constructions
  • Semi-implicit Constructions

• Limitations

• Applications
Reminder of Part 1
RIGIDITY. DEFINITION

Definition

\[ \mathcal{R}^F_A(r) := \min_{\text{rank}(A+C) \leq r} \|C\|_0. \]
**Rigidity. Definition**

**Definition**

\[
\mathcal{R}_A^F(r) := \min_{\text{rank}(A+C) \leq r} \|C\|_0.
\]

Let \( F \) be a field, \( A \in F^{n \times n} \) be a matrix, and \( 0 \leq r \leq n \). The **rigidity** of \( A \) over \( F \), denoted by \( \mathcal{R}_A^F(r) \), is the Hamming distance between \( A \) and the set of matrices of rank at most \( r \).
Non-rigid = Sparse + Low-Rank
Rigid $\neq$ Sparse + Low-Rank
EXAMPLE [Mid05]

\[ M_n = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \cdot \left( \frac{n^2}{4R^2} \right) \text{ matrices} \]

\[ R \cdot \frac{n^2}{4R^2} = \bigotimes \left( \frac{n^2}{r^2} \right) \]

\[ R_{M_n}^F (r) = \Omega \left( \frac{n^2}{r} \right) \]
Explicit Bounds on Rigidity

- What we need for circuit lower bounds:
  \[ \mathcal{R}_{M_n}^F (r) = n^{1+\delta} \text{ for } r = \Omega(n). \]
**Explicit Bounds on Rigidity**

- What we need for circuit lower bounds:
  \[ R^F_{M_n}(r) = n^{1+\delta} \text{ for } r = \Omega(n) . \]
- (Even \( R^F_{M_n}(r) = \omega(n) \) for \( r = \Omega(n) \) would give new circuit lower bounds).
Explicit Bounds on Rigidity

• What we need for circuit lower bounds:
  \( \mathcal{R}^F_{M_n}(r) = n^{1+\delta} \) for \( r = \Omega(n) \).

• (Even \( \mathcal{R}^F_{M_n}(r) = \omega(n) \) for \( r = \Omega(n) \) would give new circuit lower bounds).

• A simple explicit matrix of rigidity
  \[ \mathcal{R}^F_{M_n}(r) = \Omega \left( \frac{n^2}{r} \right) \]
  when \( r = \Theta(n) \),
  \( = \mathcal{O}(n) \).
Explicit Bounds on Rigidity

- What we need for circuit lower bounds:
  \[ R_{M_n}^F (r) = n^{1+\delta} \] for \( r = \Omega(n) \).

- Even \( R_{M_n}^F (r) = \omega(n) \) for \( r = \Omega(n) \) would give new circuit lower bounds.

- A simple explicit matrix of rigidity
  \[ R_{M_n}^F (r) = \Omega \left( \frac{n^2}{r} \right) \] .

- The best known explicit bound:
  \[ R_{M_n}^F (r) = \Omega \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right) . \]

- \( E = \text{Time} \left[ 2^{O(n)} \right] \) \( E^{NP} \)

- \( r = \Theta(n) \)
  \( \log \frac{n}{r} = \Theta(1) \)
<table>
<thead>
<tr>
<th>construction</th>
<th>rigidity</th>
<th>run-time</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>construction</th>
<th>rigidity</th>
<th>run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>$R(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>poly($n$)</td>
</tr>
</tbody>
</table>
### Semi-explicit Bounds on Rigidity

<table>
<thead>
<tr>
<th>construction</th>
<th>rigidity</th>
<th>run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>$R(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>poly$(n)$</td>
</tr>
<tr>
<td>brute force</td>
<td>$R(\varepsilon n) \geq n^2$</td>
<td>$2^{n^2}$</td>
</tr>
</tbody>
</table>
## Semi-explicit Bounds on Rigidity

<table>
<thead>
<tr>
<th>construction</th>
<th>rigidity</th>
<th>run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>$\mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>$\text{poly}(n)$</td>
</tr>
<tr>
<td>brute force</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^2$</td>
<td>$2^{n^2}$</td>
</tr>
<tr>
<td>sparse</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^{1+\delta}$</td>
<td>$2^{n^{1+\delta} \log n}$</td>
</tr>
</tbody>
</table>
## Semi-explicit Bounds on Rigidity

<table>
<thead>
<tr>
<th>Construction</th>
<th>Rigidity</th>
<th>Run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>( \mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r} )</td>
<td>( \text{poly}(n) )</td>
</tr>
<tr>
<td>brute force</td>
<td>( \mathcal{R}(\varepsilon n) \geq n^2 )</td>
<td>( 2^{n^2} )</td>
</tr>
<tr>
<td>sparse</td>
<td>( \mathcal{R}(\varepsilon n) \geq n^{1+\delta} )</td>
<td>( 2^{n^{1+\delta} \log n} )</td>
</tr>
<tr>
<td>sub-exponential</td>
<td>( \mathcal{R}(n^{0.5-\varepsilon}) \geq \frac{n^2}{\log n} )</td>
<td>( 2^{n^{1-\varepsilon}} )</td>
</tr>
</tbody>
</table>
## Semi-explicit Bounds on Rigidity

<table>
<thead>
<tr>
<th>construction</th>
<th>rigidity</th>
<th>run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>$\mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>poly($n$)</td>
</tr>
<tr>
<td>brute force</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^2$</td>
<td>$2^{n^2}$</td>
</tr>
<tr>
<td>sparse</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^{1+\delta}$</td>
<td>$2^{n^{1+\delta} \log n}$</td>
</tr>
<tr>
<td>sub-exponential</td>
<td>$\mathcal{R}(n^{0.5-\varepsilon}) \geq \frac{n^2}{\log n}$</td>
<td>$2^{n^{1-\varepsilon}}$</td>
</tr>
<tr>
<td>Vanderm. alg. ind.</td>
<td>$\mathcal{R}(\sqrt{n}) \geq \delta n^2$</td>
<td>NA</td>
</tr>
<tr>
<td>construction</td>
<td>rigidity</td>
<td>run-time</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>explicit</td>
<td>$\mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>$\text{poly}(n)$</td>
</tr>
<tr>
<td>brute force</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^2$</td>
<td>$2^{n^2}$</td>
</tr>
<tr>
<td>sparse</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^{1+\delta}$</td>
<td>$2^{n^{1+\delta} \log n}$</td>
</tr>
<tr>
<td>sub-exponential</td>
<td>$\mathcal{R}(n^{0.5-\varepsilon}) \geq \frac{n^2}{\log n}$</td>
<td>$2^{n^{1-\varepsilon}}$</td>
</tr>
<tr>
<td>Vanderm. alg. ind</td>
<td>$\mathcal{R}(\sqrt{n}) \geq \delta n^2$</td>
<td>NA</td>
</tr>
<tr>
<td>$\sqrt{p_i}$</td>
<td>$\mathcal{R}(\varepsilon n) \geq \delta n^2$</td>
<td>NA</td>
</tr>
</tbody>
</table>

Ben Lee’s talk
<table>
<thead>
<tr>
<th>Construction</th>
<th>Rigidity</th>
<th>Run-time</th>
</tr>
</thead>
<tbody>
<tr>
<td>explicit</td>
<td>$\mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>$\text{poly}(n)$</td>
</tr>
<tr>
<td>brute force</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^2$</td>
<td>$2^{n^2}$</td>
</tr>
<tr>
<td>sparse</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^{1+\delta}$</td>
<td>$2^{n^{1+\delta} \log n}$</td>
</tr>
<tr>
<td>sub-exponential</td>
<td>$\mathcal{R}(n^{0.5-\varepsilon}) \geq \frac{n^2}{\log n}$</td>
<td>$2^{n^{1-\varepsilon}}$</td>
</tr>
<tr>
<td>Vanderm. alg. ind</td>
<td>$\mathcal{R}(\sqrt{n}) \geq \delta n^2$</td>
<td>NA</td>
</tr>
<tr>
<td>$\sqrt{p_i}$</td>
<td>$\mathcal{R}(\varepsilon n) \geq \delta n^2$</td>
<td>NA</td>
</tr>
<tr>
<td>Hankel</td>
<td>$\mathcal{R}(r) \geq \frac{n^3}{r^2 \log n}$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>construction</td>
<td>rigidity</td>
<td>run-time</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>explicit</td>
<td>$\mathcal{R}(r) \geq \frac{n^2}{r} \cdot \log \frac{n}{r}$</td>
<td>poly$(n)$</td>
</tr>
<tr>
<td>brute force</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^2$</td>
<td>$2^{n^2}$</td>
</tr>
<tr>
<td>sparse</td>
<td>$\mathcal{R}(\varepsilon n) \geq n^{1+\delta}$</td>
<td>$2^{n^{1+\delta} \log n}$</td>
</tr>
<tr>
<td>sub-exponential</td>
<td>$\mathcal{R}(n^{0.5-\varepsilon}) \geq \frac{n^2}{\log n}$</td>
<td>$2^{n^{1-\varepsilon}}$</td>
</tr>
<tr>
<td>Vanderm. alg. ind</td>
<td>$\mathcal{R}(\sqrt{n}) \geq \delta n^2$</td>
<td>NA</td>
</tr>
<tr>
<td>$\sqrt{p_i}$</td>
<td>$\mathcal{R}(\varepsilon n) \geq \delta n^2$</td>
<td>NA</td>
</tr>
<tr>
<td>Hankel</td>
<td>$\mathcal{R}(r) \geq \frac{n^3}{r^2 \log n}$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>PCP $^{\text{Amey's talk}}$</td>
<td>$\mathcal{R}(2^{\log n / \log \log n}) \geq \delta n^2$</td>
<td>P$^{\text{NP}}$</td>
</tr>
</tbody>
</table>
LIMITATIONS
<table>
<thead>
<tr>
<th>Conjecture [Lokam 2009]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Many candidate matrices are conjectured to have rigidity as high as in Valiant’s question. Examples include:</td>
</tr>
</tbody>
</table>
Rigidity Candidates

- Walsh-Hadamard matrix
- Generalized Hadamard matrices
- Fourier transform matrices
- Vandermonde matrices
- Cauchy matrices
- Super regular matrices
- Good linear codes
- Hankel matrices
- Incidence matrices of projective planes
- Cayley graphs
Untouched Minor Lower Bound

- Untouched minor
Untouched Minor Lower Bound

- Untouched minor

- Step 1: $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ changes in $n \times n$ matrix leave an $r \times r$ submatrix untouched
Untouched Minor Lower Bound

- **Untouched minor**

- **Step 1:** $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ changes in $n \times n$ matrix leave an $r \times r$ submatrix untouched

- **Step 2:** Take a matrix where each $r \times r$ submatrix is full-rank
Untouched Minor Lower Bound

- Untouched minor

- Step 1: $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ changes in $n \times n$ matrix leave an $r \times r$ submatrix untouched

- Step 2: Take a matrix where each $r \times r$ submatrix is full-rank

- After $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ changes, the rank is $\geq r$
Limitations of Untouched Minor

- This method can give bounds of $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$
LIMITATIONS OF UNTouched MINOR

• This method can give bounds of $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$

• Limitation 1:
There is a set of $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ elements of a matrix that touches every $r \times r$ submatrix [Lok00]
LIMITATIONS OF UNTouched MINOR

• This method can give bounds of $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$

• Limitation 1:
  There is a set of $O\left(\frac{n^2}{r} \cdot \log \frac{n}{r}\right)$ elements of a matrix that touches every $r \times r$ submatrix [Lok00]

• Limitation 2:
  There is a matrix where all submatrices have full rank, yet it is not rigid [Val75]
Rigidity Candidates

Walsh-Hadamard matrix  
Generalized Hadamard matrices  
Fourier transform matrices  
Vandermonde matrices  
Cauchy matrices  
Super regular matrices  
Good linear codes  
Hankel matrices  
Incidence matrices of projective planes  
Cayley graphs
LINEAR CODES

- A **linear code** $C$ is a $k$-dimensional subspace of $\mathbb{F}^n$.
LINEAR CODES

- A **linear code** $C$ is a $k$-dimensional subspace of $\mathbb{F}^n$
- The **distance** of $C$ is

$$d(C) = \min (\|w\|_0 : w \in C, w \neq \mathbf{0})$$
LINEAR CODES

• A linear code $C$ is a $k$-dimensional subspace of $\mathbb{F}^n$

• The distance of $C$ is

$$d(C) = \min (\|w\|_0 : w \in C, w \neq 0)$$

• A generator matrix $G \in \mathbb{F}^{n \times k}$ is a matrix whose columns form a basis of $C$
Explicit Linear Codes

Proposition

For any finite field $\mathbb{F}$, there exists an explicit family of linear error correcting codes over $\mathbb{F}$ of dimension $k = n/4$ and minimum distance $d = \delta n$ for a constant $\delta > 0$. 
Explicit Linear Codes

Proposition
For any finite field $\mathbb{F}$, there exists an explicit family of linear error correcting codes over $\mathbb{F}$ of dimension $k = n/4$ and minimum distance $d = \delta n$ for a constant $\delta > 0$.

Such codes are called good.
RIGIDITY OF CODES

- [Fri93], [PR94], [SSS97]: every generator matrix $G$ of a good code has rigidity

$$R_G^F(r) \geq \Omega \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right).$$
RIGIDITY OF CODES

• [Fri93], [PR94], [SSS97]: every generator matrix $G$ of a good code has rigidity

$$\mathcal{R}_G^F(r) \geq \Omega \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right).$$

• Every good code has a generator matrix $G$

$$\mathcal{R}_G^F(\varepsilon n) \geq \Omega(n^2).$$
RIGIDITY OF CODES

• [Fri93], [PR94], [SSS97]: every generator matrix $G$ of a good code has rigidity

$$\mathcal{R}_G^F(r) \geq \Omega \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right).$$

• Every good code has a generator matrix $G$

$$\mathcal{R}_G^F(\varepsilon n) \geq \Omega(n^2).$$

• [Dvi16] Some good codes have a generator matrix $G$

$$\mathcal{R}_G^F(r) \leq O \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right).$$
RIGIDITY OF CODES

- [Fri93], [PR94], [SSS97]: every generator matrix \( G \) of a good code has rigidity

\[
\mathcal{R}_G^\mathbb{F}(r) \geq \Omega \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right).
\]

- Every good code has a generator matrix \( G \)

\[
\mathcal{R}_G^\mathbb{F}(\varepsilon n) \geq \Omega(n^2).
\]

- [Dvi16] Some good codes have a generator matrix \( G \)

\[
\mathcal{R}_G^\mathbb{F}(r) \leq O \left( \frac{n^2}{r} \cdot \log \frac{n}{r} \right).
\]

- Thus, we cannot improve the known explicit bound for all generator matrices of good codes
<table>
<thead>
<tr>
<th>Rigidity Candidates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walsh-Hadamard matrix</td>
</tr>
<tr>
<td>Generalized Hadamard matrices</td>
</tr>
<tr>
<td>Fourier transform matrices</td>
</tr>
<tr>
<td>Vandermonde matrices</td>
</tr>
<tr>
<td>Cauchy matrices</td>
</tr>
<tr>
<td>Super regular matrices</td>
</tr>
<tr>
<td>Good linear codes</td>
</tr>
<tr>
<td>Hankel matrices</td>
</tr>
<tr>
<td>Incidence matrices of projective planes</td>
</tr>
<tr>
<td>Cayley graphs</td>
</tr>
</tbody>
</table>
Hadamard Matrix

\[ H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \]

\[ H_N = \begin{pmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & -H_{N/2} \end{pmatrix} \text{ for } N = 2^n > 2. \]
<table>
<thead>
<tr>
<th>Known Bounds for Hadamard</th>
<th>Rigidity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{n^2}{r^4 \log^2 r}$</td>
<td>Pudlák and Savický, 88</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{r^3 \log r}$</td>
<td>Razborov, 88</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{r^2}$</td>
<td>Alon, 90</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{256r}$</td>
<td>Kashin and Razborov, 98</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{4r}$</td>
<td>Midrijānis, 05; de Wolf, 06</td>
<td></td>
</tr>
<tr>
<td>rigidity</td>
<td>reference</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{r^4 \log^2 r}$</td>
<td>Pudlák and Savický, 88</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{r^3 \log r}$</td>
<td>Razborov, 88</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{r^2}$</td>
<td>Alon, 90</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{r^2}$</td>
<td>Lokam, 95</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{256r}$</td>
<td>Kashin and Razborov, 98</td>
<td></td>
</tr>
<tr>
<td>$\frac{n^2}{4r}$</td>
<td>Midrijānis, 05; de Wolf, 06</td>
<td></td>
</tr>
</tbody>
</table>

[AW17]: $H$ is not rigid for any $r = O(n)$.

Josh's talk
Rigidity Candidates

Walsh-Hadamard matrix
Generalized Hadamard matrices
Fourier transform matrices
Vandermonde matrices
Cauchy matrices
Super regular matrices
Good linear codes
Hankel matrices
Incidence matrices of projective planes
Cayley graphs
Rigidity Candidates [DE17]

Walsh-Hadamard matrix \(\times\)
Generalized Hadamard matrices \(\_\_\_\_\_\_\_\)
Fourier transform matrices \(\_\_\_\_\_\_\_\)
Vandermonde matrices \(\_\_\_\_\_\_\)
Cauchy matrices \(\_\_\_\_\_\_\)
Super regular matrices \(\times\)
Good linear codes \(\times\)
Hankel matrices \(\_\_\_\_\_\_\)
Rigidity Candidates
[DL19]

Walsh-Hadamard matrix
Generalized Hadamard matrices
Fourier transform matrices
Vandermonde matrices
Cauchy matrices
Super regular matrices
Good linear codes
Hankel matrices
Incidence matrices of projective planes
Cayley graphs
<table>
<thead>
<tr>
<th>Rigidity Candidates [DL19]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walsh-Hadamard matrix</td>
</tr>
<tr>
<td>Generalized Hadamard matrixes</td>
</tr>
<tr>
<td>Fourier transform matrices</td>
</tr>
<tr>
<td>Vandermonde matrices</td>
</tr>
<tr>
<td>Cauchy matrices</td>
</tr>
<tr>
<td>Super regular matrices</td>
</tr>
<tr>
<td>Good linear codes</td>
</tr>
<tr>
<td>Hankel matrices</td>
</tr>
<tr>
<td>Incidence matrices of projective planes</td>
</tr>
<tr>
<td>Cayley graphs</td>
</tr>
<tr>
<td>Rigidity Candidates [DL19]</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Walsh-Hadamard matrix</td>
</tr>
<tr>
<td>Generalized Hadamard matrices</td>
</tr>
<tr>
<td>Fourier transform matrices</td>
</tr>
<tr>
<td>Vandermonde matrices</td>
</tr>
<tr>
<td>Cauchy matrices</td>
</tr>
<tr>
<td>Super regular matrices</td>
</tr>
<tr>
<td>Good linear codes</td>
</tr>
<tr>
<td>Hankel matrices</td>
</tr>
<tr>
<td>Incidence matrices of projective planes</td>
</tr>
<tr>
<td>Cayley graphs</td>
</tr>
</tbody>
</table>
Rigidity Candidates
[DL19]

- Walsh-Hadamard matrix
  - Question mark
- Generalized Hadamard matrices
  - Question mark
- Fourier transform matrices
  - Question mark
- Vandermonde matrices
  - Question mark
- Cauchy matrices
  - Question mark
- Super regular matrices
  - Question mark
- Good linear codes
  - Question mark
- Hankel matrices
  - Question mark
- Incidence matrices of projective planes
  - Question mark
- Cayley graphs
  - Question mark
Rigidity Candidates
[DL19]

Walsh-Hadamard matrix  
Generalized Hadamard matrices  
Fourier transform matrices  
Vandermonde matrices  
Cauchy matrices  
Super regular matrices  
Good linear codes  
Hankel matrices  
Incidence matrices of projective planes  
Cayley graphs  

Josh’s talk
APPLICATIONS
APPLICATIONS OF RIGIDITY

- Communication Complexity
- Circuit Complexity
- Data Structures
- Error Correcting Codes
RIGIDITY AND COMMUNICATION COMPLEXITY

Theorem (Raz89)

If \( M \in \mathbb{F}_2^{n \times n} \) has rigidity

\[
\mathcal{R}_M(r) \geq \frac{n^2}{2^{\log r^{o(1)}}} \quad \text{for } r \geq 2^{\log \log n^{\omega(1)}}
\]

then \( M \notin \text{PH}^{cc} \).
RIGIDITY AND COMMUNICATION COMPLEXITY

Theorem (Raz89)

If $M \in \mathbb{F}_2^{n \times n}$ has rigidity

$$R_M^F(r) \geq \frac{n^2}{2^\log r^{o(1)}} \text{ for } r \geq 2^{\log \log n^{o(1)}}$$

then $M \not\in \text{PH}^{cc}$.

Theorem (AC19, BHPT20)

$E^{NP} \not\subseteq \text{PH}^{cc}$.

\[ \text{Time}[2^{2^{(\log \log n)^n}}]^{NP} \not\subseteq \text{PH}^{cc} \]
CIRCUITS AND RIGIDITY
**Boolean Circuits**

\[ f: \{0, 1\}^n \rightarrow \{0, 1\}^n \]

\[
\begin{align*}
g_1 &= x_1 \oplus x_2 \\
g_2 &= x_2 \land x_3 \\
g_3 &= g_1 \lor g_2 \\
g_4 &= g_2 \lor 1 \\
g_5 &= g_3 \equiv g_4 \\
\end{align*}
\]

Circuit

- **Inputs:** \( x_1, \ldots, x_n, 0, 1 \)
- **Gates:** binary functions
- **Fan-out:** unbounded
**EXPO**

**NENTIAL BOUNDS**

**Lower Bound [Sha1949]**

Counting shows that almost all functions of $n$ variables have circuit size at least

$$2^n.$$
# Exponential Bounds

<table>
<thead>
<tr>
<th>Lower Bound [Sha1949]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counting shows that almost all functions of $n$ variables have circuit size at least</td>
</tr>
<tr>
<td>$2^n$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Upper Bound [Lup1958]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every function can be computed by a circuit of size</td>
</tr>
<tr>
<td>$2^n$.</td>
</tr>
</tbody>
</table>
EXPLICIT BOUNDS

Most functions have exponential circuit complexity
Explicit Bounds

Most functions have exponential circuit complexity

$P \neq NP$ We want to prove super-polynomial lower bounds
Explicit Bounds

Most functions have exponential circuit complexity

\[ P \neq NP \]

We want to prove super-polynomial lower bounds (for a function from NP)
**Explicit Bounds**

Most functions have *exponential* circuit complexity.

\[ P \neq NP \]

We want to prove *super-polynomial* lower bounds (for a function from \( NP \)).

We can prove only \( \approx 3n \) lower bounds.
**Explicit Bounds**

Most functions have *exponential* circuit complexity.

\[ P \neq NP \]

We want to prove *super-polynomial* lower bounds (for a function from \( \text{NP} \)).

We can prove only \( \approx 3n \) lower bounds (even for a function from \( E^{\text{NP}} \)).
Super-linear Circuit Lower Bounds?

- Two $n$-bit integers can be multiplied by a circuit of size $O(n \log n)$ [SS71,F07,HH19]
Super-linear Circuit Lower Bounds?

- Two $n$-bit integers can be multiplied by a circuit of size $O(n \log n)$ [SS71,F07,HH19]
- Discrete Fourier Transform of a sequence of length $n$ can be computed by a circuit of size $O(n \log n)$
**Super-linear Circuit Lower Bounds?**

- Two $n$-bit integers can be multiplied by a circuit of size $O(n \log n)$ [SS71,F07,HH19]
- Discrete Fourier Transform of a sequence of length $n$ can be computed by a circuit of size $O(n \log n)$
- Shifts, Permutations
Super-linear Circuit Lower Bounds?

- Two $n$-bit integers can be multiplied by a circuit of size $O(n \log n)$ [SS71,F07,HH19]
- Discrete Fourier Transform of a sequence of length $n$ can be computed by a circuit of size $O(n \log n)$
- Shifts, Permutations
- **NP**-hard problems
Super-linear Circuit Lower Bounds?

- Two $n$-bit integers can be multiplied by a circuit of size $O(n \log n)$ [SS71,F07,HH19]
- Discrete Fourier Transform of a sequence of length $n$ can be computed by a circuit of size $O(n \log n)$
- Shifts, Permutations
- **NP**-hard problems
- ...
WHAT WE CAN PROVE
WHAT WE CAN PROVE

- Depth 2: CNF/DNF. Even $\oplus_n$ requires circuits of size $\Omega(2^n)$.
WHAT WE CAN PROVE

• Depth 2: CNF/DNF. Even $\oplus_n$ requires circuits of size $\Omega(2^n)$.

• Constant depth $d$. Lower bounds $2^{n^{1/(d-1)}}$. 
WHAT WE CAN PROVE

• Depth 2: CNF/DNF. Even $\oplus_n$ requires circuits of size $\Omega(2^n)$.

• Constant depth $d$. Lower bounds $2^n^{1/(d-1)}$.

• Depth $1.9 \log n$. Know functions that cannot be computed.
WHAT WE CAN PROVE

• Depth 2: CNF/DNF. Even $\oplus_n$ requires circuits of size $\Omega(2^n)$.

• Constant depth $d$. Lower bounds $2^{n^{1/(d-1)}}$.

• Depth $1.9 \log n$. Know functions that cannot be computed.

• Depth $2 \log n$. Nothing better than $\approx 3n$. 
<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prove a lower bound of $10n$ against circuits of depth $10 \log n$.</td>
</tr>
</tbody>
</table>
**Problem**

Prove a lower bound of $10n$ against circuits of depth $10 \log n$.

More generally, a lower bound of $\omega(n)$ against circuits of depth $O(\log n)$. 
## Problem on the Frontier

### Problem

Prove a lower bound of $10n$ against circuits of depth $10 \log n$.

More generally, a lower bound of $\omega(n)$ against circuits of depth $O(\log n)$.

Valiant [Val77] gives us an amazing tool to study such circuits.
Another Problem on the Frontier

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Prove a lower bound of</em> $\omega(n)$ <em>against linear circuits of depth</em> $O(\log n)$.</td>
</tr>
</tbody>
</table>
### Another Problem on the Frontier

**Problem**

Prove a lower bound of $\omega(n)$ against linear circuits of depth $O(\log n)$.

- Incomparable to the previous problem (bounds against non-linear circuits):
## Another Problem on the Frontier

<table>
<thead>
<tr>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prove a lower bound of $\omega(n)$ against linear circuits of depth $O(\log n)$.</td>
</tr>
</tbody>
</table>

- Incomparable to the previous problem (bounds against non-linear circuits):
  - Weaker computational model
ANOTHER PROBLEM ON THE FRONTIER

Problem

Prove a lower bound of $\omega(n)$ against linear circuits of depth $O(\log n)$.

- Incomparable to the previous problem (bounds against non-linear circuits):
  - Weaker computational model
  - But fewer problems to prove lower bounds for.
Rigidity Implies Circuit Lower Bounds

Theorem (Val77)

If $\mathcal{R}_A^F(\epsilon n) > n^{1+\delta}$ for constant $\epsilon, \delta > 0$, then any $O(\log n)$-depth linear circuit computing $x \rightarrow Ax$ must be of size $\omega(n)$.
**Rigidity Implies Circuit Lower Bounds**

**Theorem (Val77)**

If $R_A^F(\varepsilon n) > n^{1+\delta}$ for constant $\varepsilon, \delta > 0$, then any $O(\log n)$-depth linear circuit computing $x \rightarrow Ax$ must be of size $\omega(n)$.
RIGIDITY IMPLIES CIRCUIT LOWER BOUNDS

Theorem (Val77)

If $\mathcal{R}_A^F(\varepsilon n) > n^{1+\delta}$ for constant $\varepsilon, \delta > 0$, then any $O(\log n)$-depth linear circuit computing $x \rightarrow Ax$ must be of size $\omega(n)$. 

\begin{tikzpicture}
  \node (p1) at (0,0) {$p_1$};
  \node (p2) at (1,0) {$p_2$};
  \node (ps) at (2,0) {$p_{s-1}$};
  \node (pen) at (3,0) {$p_{\varepsilon n}$};
  \node (x1) at (0,-1) {$x_1$};
  \node (x2) at (1,-1) {$x_2$};
  \node (xns) at (2,-1) {$x_{n-1}$};
  \node (xn) at (3,-1) {$x_n$};
  \node (q1) at (0,3) {$q_1$};
  \node (q2) at (1,3) {$q_2$};
  \node (qns) at (2,3) {$q_{n-1}$};
  \node (qn) at (3,3) {$q_n$};
  \draw (p1) -- (p2) -- (ps) -- (pen) -- (p1);
  \draw (x1) -- (x2) -- (xns) -- (xn) -- (x1);
  \draw (q1) -- (p1) -- (x1) -- (q1);
  \draw (q2) -- (p2) -- (x2) -- (q2);
  \draw (qn) -- (pen) -- (xn) -- (qn);
  \draw (qns) -- (ps) -- (xns) -- (qns);
  \node at (1.5,-0.5) {unbounded};
\end{tikzpicture}
Rigidity for rank $n/100$ and sparsity $n^{1.01}$ implies super-linear log-depth circuit lower bounds
DEPTH REDUCTIONS

- The proof reduces the depth of a circuit from $O(\log n)$ to 2 (and the latter is equivalent to rigidity)
DEPTH REDUCTIONS

- The proof reduces the depth of a circuit from $O(\log n)$ to 2 (and the latter is equivalent to rigidity)
- The proof is graph-theoretic, and graph-theoretic proofs cannot go beyond $O(\log n)$ depth [Sch82, Sch83, Kla94]
Depth Reductions

- The proof reduces the depth of a circuit from $O(\log n)$ to 2 (and the latter is equivalent to rigidity)

- The proof is graph-theoretic, and graph-theoretic proofs cannot go beyond $O(\log n)$ depth [Sch82, Sch83, Kla94]

- A non-graph-theoretic proof [GKW21] works for unbounded-depth circuits, but alas only for size $< 4n$
Theorem (GKW21)

Let $\mathbb{F}$ be a field, and $A \in \mathbb{F}^{n \times n}$ be a family of matrices for $n \in \mathbb{N}$.

If $\mathcal{R}_{A}^{\mathbb{F}}(\varepsilon n) > 16n$, then any linear circuit computing $x \rightarrow Ax$ must be of size $\geq 4\varepsilon n$. 
Rigidity for rank $0.99n$ and sparsity $16n$ implies circuit lower bound of $3.9n$ unbounded depth
**General Boolean**
(non-linear gates)

\[
\begin{align*}
\land &\lor \land \quad 5n-o(n) \\
(\text{all binary } \land \lor \land) &\equiv 3.01n \\
\end{align*}
\]

\[
\{0,1\}^n \Rightarrow \{0,1\}^n
\]

(upper bounds \(6n-o(n)\))

4.01n

**Linear Boolean circuits**

\[
\begin{align*}
\{0,1\}^n &\Rightarrow \{0,1\}^n \\
(3n-o(n)) \\
\{0,1\}^n &\Rightarrow \{0,1\}^{\log n} \\
(2n-o(n))
\end{align*}
\]
DATA STRUCTURES AND RIGIDITY

Siva's talk
Data Structures

Stack, Queue, List, Heap

Search Trees

Hash Tables

hash(unsigned x) {
    x ^= x >> (w-m);
    return (a*x) >> (w-m);
}
Static Data Structures. Examples

- **Graph Distances**: Preprocess a road network in order to efficiently compute distances between cities (Google Maps)
Static Data Structures. Examples

- **Graph Distances**: Preprocess a road network in order to efficiently compute distances between cities (Google Maps)
- **Nearest Neighbors**: Preprocess a set of points in order to efficiently find closest point to a query point (Netflix recommendations)
Static Data Structures. Examples

- **Graph Distances**: Preprocess a road network in order to efficiently compute distances between cities (Google Maps)
- **Nearest Neighbors**: Preprocess a set of points in order to efficiently find closest point to a query point (Netflix recommendations)
- **Range Counting**: Preprocess a set of points in order to efficiently compute the number of points in a given rectangle (Amazon market size estimation)
Static Data Structures

Preprocessing
STATIC DATA STRUCTURES

Queries

Preprocessing
STATIC DATA STRUCTURES

Queries

Dabolim — Washington
STATIC DATA STRUCTURES

Queries

Dabolim — Washington

Preprocessing
STATIC DATA STRUCTURES

Queries

Dabolim — Washington

Bangalore — New York

Preprocessing
Static Data Structures. Definition

\[ x_1 \quad x_2 \quad \cdots \quad x_{n-1} \quad x_n \]
Static Data Structures. Definition

\[ p_1 \ p_2 \ \cdots \ p_{s/2} \ \cdots \ p_{s-1} \ p_s \]

unbounded

\[ x_1 \ x_2 \ \cdots \ x_{n-1} \ x_n \]
Static Data Structures. Definition

$q_1 \quad q_2 \quad \cdots \quad q_{m/2} \quad \cdots \quad q_{m-1} \quad q_m$

$p_1 \quad p_2 \quad \cdots \quad p_{s/2} \quad \cdots \quad p_{s-1} \quad p_s$

unbounded

$x_1 \quad x_2 \quad \cdots \quad x_{n-1} \quad x_n$
Static Data Structures. Definition

Efficient DS:
\[ s = n \text{ poly log } n \]
\[ t = \text{poly log } n \]
COMPARISON

\[ q_1 \; q_2 \; \cdots \; q_{m/2} \; \cdots \; q_{m-1} \; q_m \]
\[ p_1 \; p_2 \; \cdots \; p_{s/2} \; \cdots \; p_{s-1} \; p_s \]

\[ x_1 \; x_2 \; \cdots \; x_{n-1} \; x_n \]

unbounded

DS

Circuits
Linear Circuits

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$
where $M \in \mathbb{F}^{m \times n}$
LINEAR CIRCUITS

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,
LINEAR CIRCUITS

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,
Linear Circuits

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

$$M = A + C \cdot D$$
**LINEAR CIRCUITS**

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$.
- For a circuit of size $O(n)$ and depth $O(\log n)$,

\[
M = A + C \cdot D
\]

outputs on inputs
**Linear Circuits**

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,
**LINEAR CIRCUITS**

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,

$$M = A + C \cdot D = A + B$$

- low-rank
**Linear Circuits**

- A linear circuit computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
- For a circuit of size $O(n)$ and depth $O(\log n)$,
  
  \[
  M = A + C \cdot D = A + B \\
  \text{sparse} \quad \text{sparse} \quad \text{low-rank}
  \]

- $M \in \mathbb{F}^{m \times n}$ is $(\varepsilon n, t)$-rigid iff
  
  \[
  M \neq A + B \\
  t\text{-sparse} \quad \text{rk} \leq \varepsilon n
  \]
Linear Data Structures

- A linear DS computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
A linear DS computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$
A linear DS computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$

$M = A \cdot B$
A linear DS computes $Mx$ for input $x \in \mathbb{F}^n$ where $M \in \mathbb{F}^{m \times n}$, with $m \times n \mapsto m \times s \mapsto s \times n$.

$$M = A \cdot B$$
**LINEAR DATA STRUCTURES**

\[ m = n^2, n^{10}, n^{100} \]

- A linear DS computes \( Mx \) for input \( x \in \mathbb{F}^n \) where \( M \in \mathbb{F}^{m \times n} \)

\[ M = A \cdot B \]

- \( m \times n \) \quad \( m \times s \) \quad \( s \times n \)
- \( t \)-sparse \quad \text{small}

\[ M \in \mathbb{F}^{n^{100} \times n} \]

\[ B \in \mathbb{F}^{n \log_2 n \times n} \]
Small circuit / Non-rigid

\[
M = A + B \\
t\text{-sparse} \quad \text{rk} \leq \varepsilon n
\]
**Comparison**

Small circuit / Non-rigid

\[ M = A + B \]
\[ \text{t-sparse} \quad \text{rk} \leq \varepsilon n \]

Efficient Data Structure

\[ M = A \cdot B \]
\[ \text{t-sparse} \quad \text{small} \]
I’m looking for prospective PhD students who are interested in theory.

alex.golovnev@gmail.com

Thank you for your attention!