TOWARDS AN SDP-BASED APPROACH TO SPECTRAL METHODS:

A NEARLY-LINEAR-TIME ALGORITHM FOR GRAPH PARTITIONING AND DECOMPOSITION

Lorenzo Orecchia, UC Berkeley
Nisheeth K. Vishnoi, MSR Bangalore

Speaker: Lorenzo Orecchia

SODA 2011
Undirected weighted graph \( G = (V, E, w) \)

\[ |V| = n \]
\[ |E| = m \]

\[ \text{Conductance of } S \subseteq V \]
\[ \phi(S) = \frac{w(S, \bar{S})}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}} \]

\[ \text{vol}(S) = w(S, V) = 8 \]
\[ w(S, \bar{S}) = 4 \]
\[ \phi(S) = \frac{1}{2} \]
GRAPH PARTITIONING

DECISION PROBLEM

Does $G$ have a $c$-balanced cut of conductance $< \gamma$?

\[ \frac{1}{2} > \frac{\text{vol}(S)}{\text{vol}(V)} > c \]
GRAPH PARTITIONING

DECISION PROBLEM

Does G have a $c$-balanced cut of conductance $<\gamma$? 

\[ \frac{1}{2} > \frac{\text{vol}(S)}{\text{vol}(V)} > c \]

NP-HARD
# APPROXIMATION ALGORITHMS

## DECISION PROBLEM

Does $G$ have a $c$-balanced cut of conductance $< \gamma$?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Distinguishes $\geq \gamma$ and</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Eigenvector</td>
<td>Spectral</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(n^2)$</td>
</tr>
<tr>
<td>[Leighton, Rao ‘88]</td>
<td>Flow</td>
<td>$O(\gamma \log n)$</td>
<td>$\tilde{O}(n^{\frac{3}{2}})$ [AK’07, OSVV’08]</td>
</tr>
<tr>
<td>[Arora, Rao, Vazirani ‘04]</td>
<td>SDP (Flow + Spectral)</td>
<td>$O(\gamma \sqrt{\log n})$</td>
<td>$\tilde{O}(n^{\frac{3}{2}})$ [Sherman ‘09]</td>
</tr>
</tbody>
</table>
## APPROXIMATION ALGORITHMS

### DECISION PROBLEM

Does $G$ have a $c$-balanced cut of conductance $< \gamma$?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Distinguishes $\geq \gamma$ and</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Eigenvector</td>
<td>Spectral</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(n^2)$</td>
</tr>
<tr>
<td>[Leighton, Rao ‘88]</td>
<td>Flow</td>
<td>$O(\gamma \log n)$</td>
<td>$\tilde{O}(n^{3\over 2})$ [AK’07, OSVV’08]</td>
</tr>
<tr>
<td>[Arora, Rao, Vazirani ‘04]</td>
<td>SDP (Flow + Spectral)</td>
<td>$O(\gamma \sqrt{\log n})$</td>
<td>$\tilde{O}(n^{3\over 2})$ [Sherman ‘09]</td>
</tr>
</tbody>
</table>

**GOAL:** NEARLY –LINEAR TIME ALGORITHMS
### APPROXIMATION ALGORITHMS

#### DECISION PROBLEM

Does $G$ have a $c$-balanced cut of conductance $\gamma$?

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Distinguishes $\geq \gamma$ and</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Eigenvector</td>
<td>Spectral</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(n^2)$</td>
</tr>
<tr>
<td>[Leighton, Rao ‘88]</td>
<td>Flow</td>
<td>$O(\gamma \log n)$</td>
<td>$\tilde{O}(n^{3/2})$</td>
</tr>
<tr>
<td>[Arora, Rao, Vazirani ‘04]</td>
<td>SDP (Flow + Spectral)</td>
<td>$O(\gamma \sqrt{\log n})$</td>
<td>$\tilde{O}(n^{3/2})$</td>
</tr>
</tbody>
</table>

**FOCUS ON SPECTRAL METHODS**
# SPECTRAL ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Distinguishes $\geq \gamma$ and</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Eigenvector</td>
<td>Eigenvector</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(n^2)$</td>
</tr>
<tr>
<td>[Spielman, Teng ‘04]</td>
<td>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log^3 n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma^2}\right)$</td>
</tr>
<tr>
<td>[Andersen, Chung, Lang ‘07]</td>
<td>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma}\right)$</td>
</tr>
<tr>
<td>[Andersen, Peres ‘09]</td>
<td>Evolving Sets</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$</td>
</tr>
</tbody>
</table>
### SPECTRAL ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Distinguishes $\geq \gamma$ and</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Eigenvector</td>
<td>Eigenvector</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(n^2)$</td>
</tr>
<tr>
<td>[Spielman, Teng ‘04]</td>
<td>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log^3 n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma^2}\right)$</td>
</tr>
<tr>
<td>[Andersen, Chung, Lang ‘07]</td>
<td>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma}\right)$</td>
</tr>
<tr>
<td>[Andersen, Peres ‘09]</td>
<td>Evolving Sets</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$</td>
</tr>
</tbody>
</table>

### UNIQUE GAMES IMPLICATIONS
## SPECTRAL ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Method</th>
<th>Distinguishes $\geq \gamma$ and</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Eigenvector</td>
<td>Eigenvector</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(n^2)$</td>
</tr>
<tr>
<td>[Spielman, Teng ‘04]</td>
<td>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log^3 n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma^2}\right)$</td>
</tr>
<tr>
<td>[Andersen, Chung, Lang ‘07]</td>
<td>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma}\right)$</td>
</tr>
<tr>
<td>[Andersen, Peres ‘09]</td>
<td>Evolving Sets</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$</td>
</tr>
<tr>
<td>[Orecchia, Vishnoi ‘11]</td>
<td>SDP-based</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}\left(\frac{m}{\gamma}\right)$</td>
</tr>
</tbody>
</table>
OUR RESULT

| [Orecchia, Vishnoi ‘11] | SDP-based | $O(\sqrt{\gamma})$ | $\tilde{O}\left(\frac{m}{\gamma}\right)$ |

• First spectral nearly-linear time algorithm that **matches optimal approximation guarantee**.

• Outputs certificates of special form.
  - Allows application to constructing **graph decompositions**.

• Uses **SDP formulation** to obtain fast spectral algorithm.
  - Arora-Kale framework.

• Algorithm has natural **random walk interpretation**.
## OUR RESULT

<table>
<thead>
<tr>
<th>[Orecchia, Vishnoi ‘11]</th>
<th>SDP-based</th>
<th>$O(\sqrt{\gamma})$</th>
<th>$\tilde{O}\left(\frac{m}{\gamma}\right)$</th>
</tr>
</thead>
</table>

- First spectral nearly-linear time algorithm that **matches optimal approximation guarantee**.
  - Outputs certificates of special form.
    - Allows application to constructing **graph decompositions**.
  - Uses **SDP formulation** to obtain fast spectral algorithm.
    - Arora-Kale framework.
- Algorithm has natural **random walk interpretation**.
(\(\alpha, \varepsilon\)) - decomposition
Partition \(V\) into subsets \(C_1, C_2, C_3, \ldots, C_t\) such that
(α, ε) - decomposition
Partition $V$ into subsets $C_1, C_2, C_3, \ldots, C_t$ such that

- $\forall i, \lambda_{G|C_i} \geq \alpha$

WELL-CONNECTED CLUSTERS
(\(\alpha, \epsilon\)) - decomposition
Partition \(V\) into subsets \(C_1, C_2, C_3, \ldots, C_t\) such that

- \(\forall i, \lambda_G|C_i \geq \alpha\) • WELL-CONNECTED CLUSTERS
- \(\sum_{i<j} w(C_i, C_j) \cdot \epsilon \cdot \text{vol}(V)\) • FEW INTERCLUSTER EDGES
GRAPH DECOMPOSITION

Applications:

- Clustering [Kannan, Vempala, Vetta ‘00]
- Sparsification [Spielman, Teng ‘04]
- Preconditioning [Spielman, Teng ‘04], [Koutis, Miller’08]
- Fast Graph Algorithms and Heuristics

NB: must be computed in nearly-linear time.
Applications:

- Clustering [Kannan, Vempala, Vetta ‘00]
- Sparsification [Spielman, Teng ‘04]
- Preconditioning [Spielman, Teng ‘04], [Koutis, Miller’08]
- Fast Graph Algorithms and Heuristics

**NB**: must be computed in nearly-linear time.

**SPECIAL CERTIFICATE ENABLES CONSTRUCTION OF GRAPH DECOMPOSITIONS**
Partitioning Algorithm

Partition($G$, $\gamma$, $c$) either outputs

- an $\Omega(c)$-balanced cut $T$ with
  \[ \phi(T) \cdot f(\gamma), \text{ or} \]
- a certificate that for all $c$-balanced $S \subset V$,
  \[ \phi(S) \geq \gamma. \]
Partitioning Algorithm

Partition($G, \gamma, c$) either outputs

- an $\Omega(c)$-balanced cut $T$ with $\phi(T) \cdot f(\gamma)$, or
- a set $U$, $\text{vol}(U) \cdot c/2$, such that

  for all $S$ with $\text{vol}(S) \cdot \text{vol}(V)/2$ and $\phi(S) \leq \gamma$

\[ \text{vol}(S' \cup U) \geq \frac{1}{2} \text{vol}(S') \]
DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition($G$, $\gamma$, $c$) either outputs

- an $\Omega(c)$-balanced cut $T$ with $\phi(T) \cdot f(\gamma)$, or

- a set $U$, $\text{vol}(U) \cdot c/2$, such that

  for all $S$ with $\text{vol}(S) \cdot \text{vol}(V)/2$ and $\phi(S) \leq \gamma$

  $$\text{vol}(S \cup U) \geq \frac{1}{2} \text{vol}(S)$$
Partitioning Algorithm

Partition($G$, $\gamma$, $c$) either outputs

- an $\Omega(c)$-balanced cut $T$ with $\phi(T) \cdot f(\gamma)$, or
- a set $U$, $\text{vol}(U) \cdot c/2$, such that
  for all $S$ with $\text{vol}(S) \cdot \text{vol}(V)/2$ and $\phi(S) \leq \gamma$

$$\text{vol}(S' \cup U) \geq \frac{1}{2} \text{vol}(S')$$
Partitioning Algorithm

Partition($G, \gamma, c$) either outputs

- an $\Omega(c)$-balanced cut $T$ with $\phi(T) \cdot f(\gamma)$, or
- a set $U$, $\text{vol}(U) \cdot c/2$, such that for all $S$ with $\text{vol}(S) \cdot \text{vol}(V)/2$ and $\phi(S) \leq \gamma$

$$\text{vol}(S \cup U) \geq \frac{1}{2} \text{vol}(S)$$
DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition($G$, $\gamma$, $c$) either outputs

- an $\Omega(c)$-balanced cut $T$ with $\phi(T) \cdot f(\gamma)$, or
- a set $U$, $\text{vol}(U) \cdot c/2$, such that $\phi(U) \cdot f(\gamma)$, for all $S$ with $\text{vol}(S) \cdot \text{vol}(V)/2$ and $\phi(S) \leq \gamma$

$$\text{vol}(S \cup U) \geq \frac{1}{2} \text{vol}(S)$$

SPARSE UNBALANCED CUT CORRELATED WITH ALL SPARSE UNBALANCED CUTS
Partitioning Algorithm

Partition(G, γ, c) either outputs

- an Ω(c)-balanced cut T with \( \phi(T) \cdot f(\gamma) \), or
- a set \( U \), \( \text{vol}(U) \cdot c/2 \), such that \( \phi(U) \cdot f(\gamma) \),

for all \( S \) with \( \text{vol}(S) \cdot \text{vol}(V)/2 \) and \( \phi(S) \leq \gamma \)

\[ \text{vol}(S' \cup U) \geq \frac{1}{2} \text{vol}(S') \]

Decomposition

Nearly-linear time Partition(G, γ, c) APPLY RECursively Can construct (\( \alpha, f(\alpha) \log^2 n \))-decomposition in nearly-linear time

[Spiegelman, Teng ’04]
OUR ALGORITHM

\textsc{BalCut}(G, \gamma, c) runs in time $\tilde{O} \left( \frac{m}{\gamma} \right)$ and outputs either

- an $\Omega(c)$-balanced cut $T$ with
  \[ \phi(T) \cdot O(\sqrt{\gamma}), \text{ or} \]
- a \textit{special} certificate that for all $c$-balanced $S \subset V$,
  \[ \phi(S) \geq \gamma. \]
THE ALGORITHM

• For each $t$, keep graph $G_t$ and a rate $\eta_t$. $G_0 = G$ and $\eta_0 = 1$.

At iteration $t$, repeat $O(\log n)$ times:

• Consider $G_t$.
• Pick random assignment of $\pm 1$ charge to the vertices.
The Algorithm

- For each $t$, keep graph $G_t$ and a rate $\eta_t$. $G_0 = G$ and $\eta_0 = 1$.

At iteration $t$, repeat $O(\log n)$ times:

- Consider $G_t$.
- Pick random assignment of $\pm 1$ charge to the vertices.
- Mix charge along edges of $G_t$ using heat kernel with rate $\eta_t$. 

Diagram: 

- $\bullet$ = $+1$ charge
- $\bullet$ = $-1$ charge
THE ALGORITHM

• For each $t$, keep graph $G_t$ and a rate $\eta_t$. $G_0 = G$ and $\eta_0 = 1$.

At iteration $t$, repeat $O(\log n)$ times:

• Consider $G_t$.
• Pick random assignment of $\pm 1$ charge to the vertices.
• Mix charge along edges with rate $\eta_t$.
• Sort final distribution by charge.
THE ALGORITHM

- For each $t$, keep graph $G_t$ and a rate $\eta_t$. $G_0 = G$ and $\eta_0 = 1$.

At iteration $t$, repeat $O(\log n)$ times:
- Consider $G_t$.
- Pick random assignment of $\pm 1$ charge to the vertices.
- Mix along edges with rate $\eta_t$.
- Sort final distribution by charge.
- Check all $\Omega(c)$-balanced sweep cuts $S_1, \cdots, S_k$ for $\phi(S_i) \leq O(\gamma^{1/2})$ in $G$.

$\bullet = +1$ charge

$\bullet = -1$ charge

For each $t$, keep graph $G_t$ and a rate $\eta_t$. $G_0 = G$ and $\eta_0 = 1$.
THE ALGORITHM

- For each $t$, keep graph $G_t$ and a rate $\eta_t$. $G_0 = G$ and $\eta_0 = 1$.

At iteration $t$, repeat $O(\log n)$ times:

- Consider $G_t$.
- Pick random assignment of $\pm 1$ charge to the vertices.
- Mix along edges with rate $\eta_t$.
- Consider final distribution, sorted by charge.
- Check all $\Omega(b)$-balanced sweep cuts $S_1, \cdots, S_k$ for $\phi(S_i) \leq O(\gamma^{1/2})$ in $G$.

What if no sparse balanced cut is found?

• $\bullet = +1$ charge

• $\bullet = -1$ charge
THE ALGORITHM

• What if no sparse balanced cut is found?

Consider the $O(\log n)$-dimensional vector embedding of vertices given by the final distributions.
CASE 1: \[ \sum_{\{i,j\} \in E} ||v_i - v_j||^2 \geq \gamma \cdot \sum_{i \in V} ||v_i||^2 \]

Random walks have not mixed enough.

How to fix it?

Increase rate. \( \eta_{t+1} = \eta_t + 1 \)
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \cdot \gamma \cdot \sum_{i \in V} \|v_i\|^2 \]
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \cdot \gamma \cdot \sum_{i \in V} \|v_i\|^2 \]

**GOAL:** find $U$ such that

- $\phi(U) \cdot O(\sqrt{\gamma})$,
- $U$ captures most of the variance of the embedding.
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \cdot \gamma \cdot \sum_{i \in V} \|v_i\|^2 \]

GOAL: find \(U\) such that

- \(\phi(U) \cdot O(\sqrt{\gamma})\),

- \(U\) captures large fraction of the variance of the embedding.

\[ \sum_{i \in U} \|v_i\|^2 \geq \Box(1) \cdot \sum_{i \in V} \|v_i\|^2 \]
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \cdot \gamma \cdot \sum_{i \in V} \|v_i\|^2 \]

**GOAL:** find \( U \) such that

- \( \phi(U) \cdot O(\sqrt{\gamma}) \),
- \( U \) captures large fraction of the variance of the embedding.

**SOLUTION:** Check all ball cuts centered around origin.
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \cdot \gamma \cdot \sum_{i \in V} \|v_i\|^2 \]

GOAL: find \( U \) such that

\[ \phi(U) \cdot O(\sqrt{\gamma}), \]

- \( U \) captures large fraction of the variance of the embedding.

SOLUTION: Check all ball cuts centered around origin.

Check all sweep-cuts of radius vector \( r_i = \|v_i\| \)

Long vectors imply random walks got stuck in \( U \)
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \cdot \gamma \cdot \sum_{i \in V} \|v_i\|^2 \]

GOAL: find $U$ such that

- $\phi(U) \cdot O(\sqrt{\gamma})$,
- $U$ captures large fraction of the variance of the embedding.

SOLUTION: Check all ball cuts centered around origin.

NB: constructing sparse unbalanced $U$ crucial in obtaining a special certificate.
THE ALGORITHM

CASE 2: \[ \sum_{\{i,j\} \in E} ||v_i - v_j||^2 \cdot \gamma \cdot \sum_{i \in V} ||v_i||^2 \]

Given \( U \) such that

- \( \phi(U) \cdot O(\sqrt{\gamma}) \),
- \( U \) captures large fraction of the variance of the embedding.

How to fix it?

\[ G_{t+1} = G_t + \frac{\gamma}{n} \sum_{i \in U} \text{Star}_i \]
THE ALGORITHM

• If no sparse balanced cut found in $T$ iterations for
  
  $$T = O \left( \frac{\log n}{\gamma} \right),$$

  we can show that $\cup U_i$ is a special certificate that $G$ has no $c$-balanced cuts of conductance less than $\gamma$.

• Running Time

  $$\tilde{O}(m) \times O\left( \frac{\log n}{\gamma} \right) = \tilde{O}\left( \frac{m}{\gamma} \right)$$

  time per iteration \hspace{1cm} iterations
**SDP FORMULATION**

\[
\mathbb{E}_{\{i,j\} \in E_G} \|v_i - v_j\|^2 \cdot \gamma, \\
\mathbb{E}_{\{i,j\} \in V \times V} \|v_i - v_j\|^2 = \frac{1}{2m}, \\
\forall i \in V \quad \mathbb{E}_{j \in V} \|v_i - v_j\|^2 \cdot \frac{1}{c} \cdot \frac{1}{2m}.
\]
SDP FORMULATION

\[ \mathbb{E}_{\{i,j\} \in E_G} \|v_i - v_j\|^2 \cdot \gamma, \]

\[ \mathbb{E}_{\{i,j\} \in V \times V} \|v_i - v_j\|^2 = \frac{1}{2m}, \]

\[ \forall i \in V \quad \mathbb{E}_{j \in V} \|v_i - v_j\|^2 \cdot \frac{1}{c} \cdot \frac{1}{2m}. \]
SDP FORMULATION

\[ \mathbb{E}_{\{i,j\} \in E_G} \|v_i - v_j\|^2 \cdot \gamma, \]

\[ \mathbb{E}_{\{i,j\} \in V \times V} \|v_i - v_j\|^2 = \frac{1}{2m}, \]

\[ \forall i \in V \quad \mathbb{E}_{j \in V} \|v_i - v_j\|^2 \cdot \frac{1}{c} \cdot \frac{1}{2m}. \]
1. If \[ \sum_{\{i, j\} \in E} ||v_i - v_j||^2 \geq \gamma \cdot \sum_{i \in V} ||v_i||^2 \]

increase rate: \( \eta_{t+1} = \eta_t + 1 \).

2. Otherwise, grow ball to find an unbalanced cut \( U \) such that
   - \( U \) contains most of the variance of the embedding,
   - \( \phi(U) \leq O(\gamma^{1/2}) \).

Set:
\[
G_{t+1} = G_t + \frac{\gamma}{n} \sum_{i \in U} \text{Star}_i
\]
CERTIFICATE

- SDP dual implies $L(G_T) \geq 2\gamma L(K_n)$

\[ \bigcup U_i \quad \phi(\bigcup U_i) \cdot \sqrt{\gamma} \]
CERTIFICATE

• SDP dual implies $L(G_T) \geq 2\gamma L(K_n)$

$\phi(\cup U_i) \cdot \sqrt{\gamma}$

Any **sparse unbalanced cut** $T$ must contain the root of many stars.

$T$ is well-correlated with $\cup U_i$
CERTIFICATE

- SDP dual implies $L(G_T) \geq 2\gamma L(K_n)$

Any \textit{sparse small cut} $T$ must contain the root of many stars.

\[ \phi(T) \cdot \gamma \]

\[ \phi(\cup U_i) \cdot \sqrt{\gamma} \]

$T$ is well-correlated with $\cup U_i$

THANK YOU