

*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

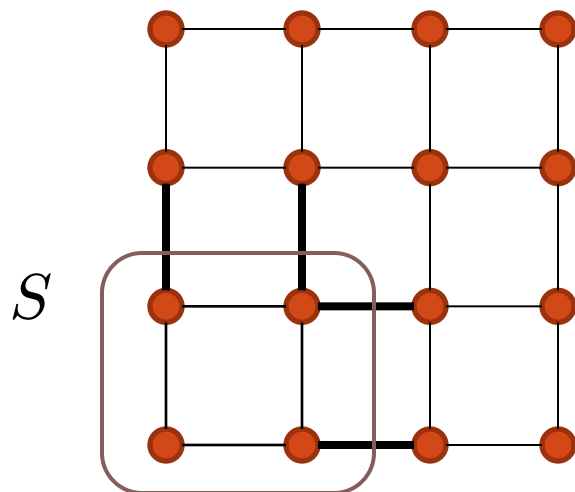
SODA 2011

GRAPH PARTITIONING

Undirected weighted graph $G = (V, E, w)$

$$|V| = n$$

$$|E| = m$$



$$\text{vol}(S) = w(S, V) = 8$$

$$w(S, \bar{S}) = 4$$

$$\phi(S) = \frac{1}{2}$$

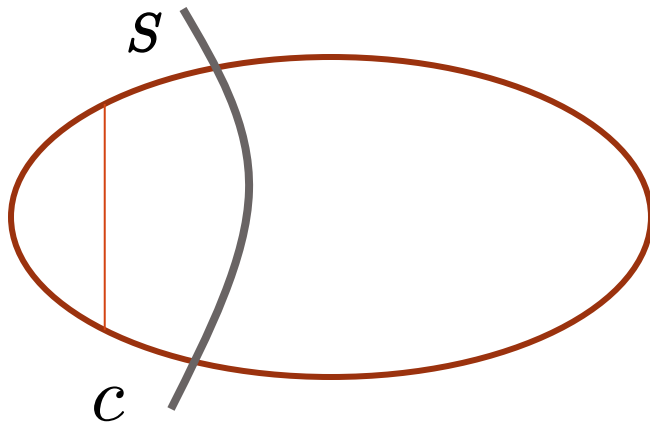
Conductance of $S \subseteq V$

$$\phi(S) = \frac{w(S, \bar{S})}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

GRAPH PARTITIONING

DECISION PROBLEM

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

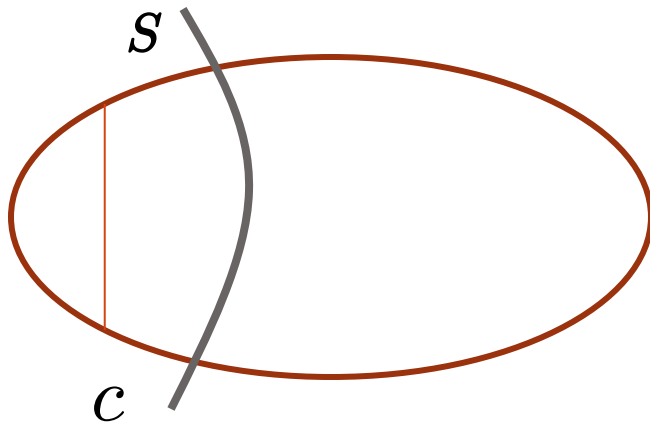


$$\frac{1}{2} > \frac{\text{vol}(S)}{\text{vol}(V)} > c$$

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 $< \gamma$?

Algorithm	Method	Distinguishes $\geq \gamma$ and	Running Time
Recursive Eigenvector	Spectral	$O(\sqrt{\gamma})$	$\tilde{O}(n^2)$
[Leighton, Rao '88]	Flow	$O(\gamma \log n)$	$\tilde{O}(n^{\frac{3}{2}})$ [AK'07, OSVV'08]
[Arora, Rao, Vazirani '04]	SDP (Flow + Spectral)	$O(\gamma \sqrt{\log n})$	$\tilde{O}(n^{\frac{3}{2}})$ [Sherman '09]

APPROXIMATION ALGORITHMS

DECISION PROBLEM

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

Algorithm	Method	Distinguishes $\geq \gamma$ and	Running Time
Recursive Eigenvector	Spectral	$O(\sqrt{\gamma})$	$\tilde{O}(n^2)$
[Leighton, Rao '88]	Flow	$O(\gamma \log n)$	$\tilde{O}(n^{\frac{3}{2}})$ [AK'07, OSVV'08]
[Arora, Rao, Vazirani '04]	SDP (Flow + Spectral)	$O(\gamma \sqrt{\log n})$	$\tilde{O}(n^{\frac{3}{2}})$ [Sherman '09]

GOAL: NEARLY –LINEAR TIME ALGORITHMS

APPROXIMATION ALGORITHMS

DECISION PROBLEM

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

Algorithm	Method	Distinguishes $\geq \gamma$ and	Running Time
Recursive Eigenvector	Spectral	$O(\sqrt{\gamma})$	$\tilde{O}(n^2)$
[Leighton, Rao '88]	Flow	$O(\gamma \log n)$	$\tilde{O}(n^{\frac{3}{2}})$ [AK'07, OSVV'08]
[Arora, Rao, Vazirani '04]	SDP (Flow + Spectral)	$O(\gamma \sqrt{\log n})$	$\tilde{O}(n^{\frac{3}{2}})$ [Sherman '09]

FOCUS ON SPECTRAL METHODS

SPECTRAL ALGORITHMS

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

SPECTRAL ALGORITHMS

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

UNIQUE GAMES IMPLICATIONS

SPECTRAL ALGORITHMS

Algorithm	Method	Distinguishes $\geq \gamma$ and	Time
Recursive Eigenvector	Eigenvector	$O(\sqrt{\gamma})$	$\tilde{O}(n^2)$
[Spielman, Teng '04]	Local Random Walks	$O\left(\sqrt{\gamma \log^3 n}\right)$	$\tilde{O}\left(\frac{m}{\gamma^2}\right)$
[Andersen, Chung, Lang '07]	Local Random Walks	$O\left(\sqrt{\gamma \log n}\right)$	$\tilde{O}\left(\frac{m}{\gamma}\right)$
[Andersen, Peres '09]	Evolving Sets	$O\left(\sqrt{\gamma \log n}\right)$	$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$
[Orecchia, Vishnoi '11]	SDP-based	$O(\sqrt{\gamma})$	$\tilde{O}\left(\frac{m}{\gamma}\right)$

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

[Orecchia, Vishnoi '11]

SDP-based

$O(\sqrt{\gamma})$

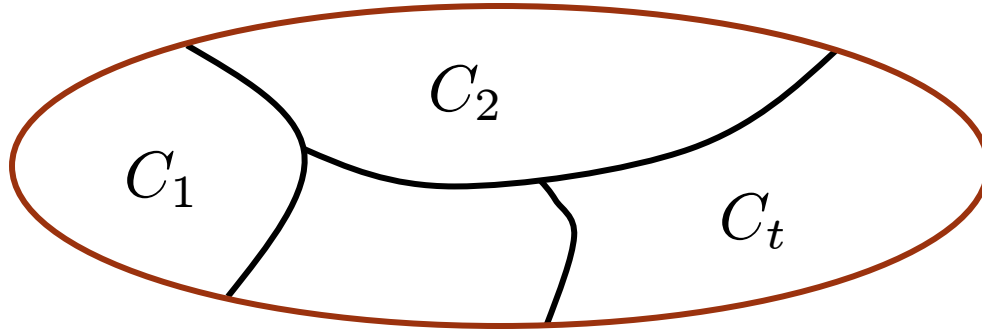
$\tilde{O}\left(\frac{m}{\gamma}\right)$

- 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**.

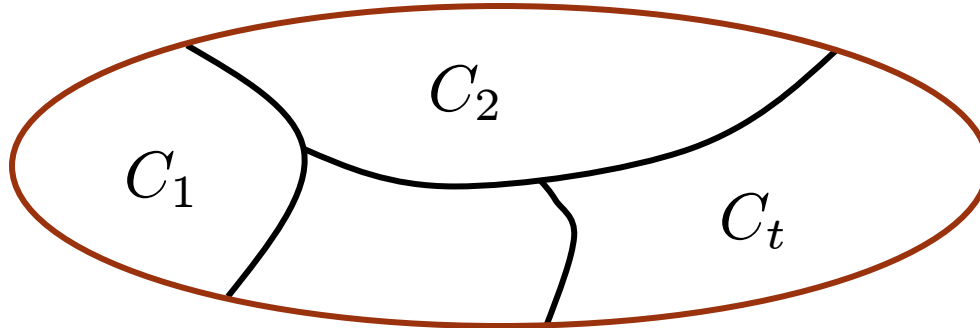
GRAPH DECOMPOSITION



(α, ϵ) - decomposition

Partition V into subsets $C_1, C_2, C_3, \dots, C_t$ such that

GRAPH DECOMPOSITION



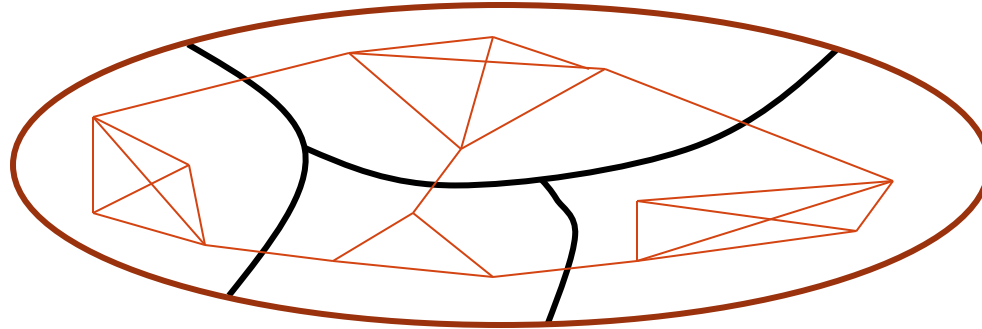
(α, ϵ) - decomposition

Partition V into subsets $C_1, C_2, C_3, \dots, C_t$ such that

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

WELL-CONNECTED CLUSTERS

GRAPH DECOMPOSITION



(α, ϵ) - decomposition

Partition V into subsets $C_1, C_2, C_3, \dots, C_t$ such that

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

WELL-CONNECTED CLUSTERS

- $\sum_{i < j} w(C_i, C_j) \leq \epsilon \cdot \text{vol}(V)$

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.

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.

**SPECIAL CERTIFICATE ENABLES CONSTRUCTION
OF GRAPH DECOMPOSITIONS**

DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition(G, γ, c) either outputs

- an $\Omega(c)$ -balanced cut T with

$$\phi(T) \geq f(\gamma), \text{ or}$$

- a certificate that for all c -balanced $S \subset V$,

$$\phi(S) \geq \gamma.$$

DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition(G, γ, c) either outputs

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

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

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

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

CERTIFICATE

DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition(G, γ, c) either outputs

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

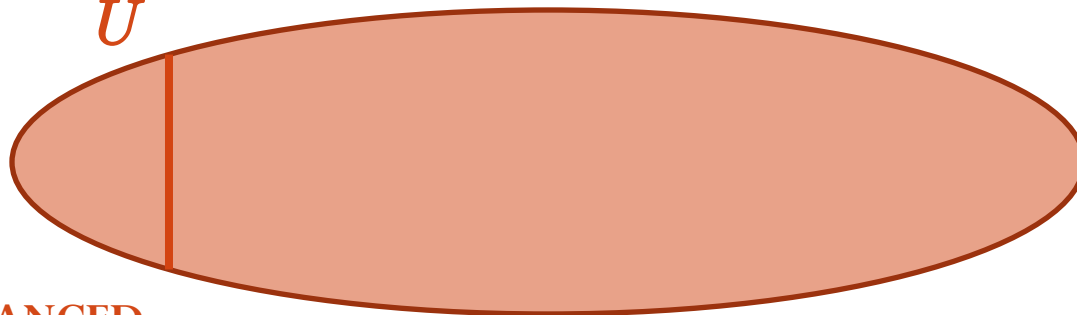
○ a set U , $\text{vol}(U) \leq c/2$, such that

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

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

CERTIFICATE

U



UNBALANCED

DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition(G, γ, c) either outputs

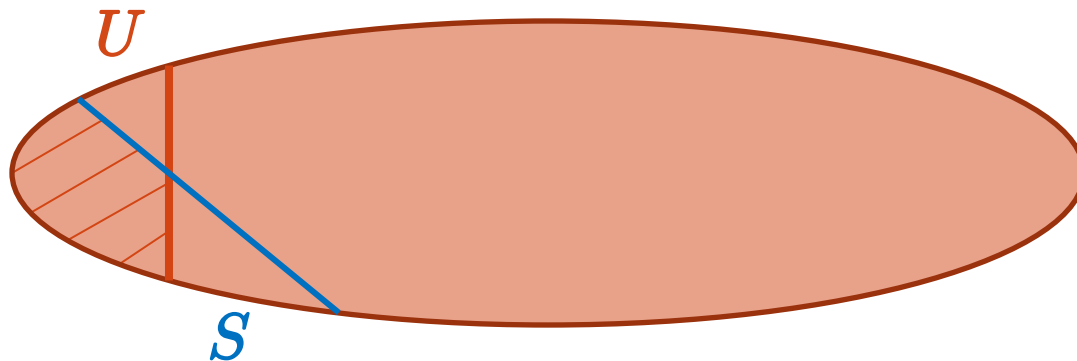
○ an $\Omega(c)$ -balanced cut T with $\phi(T) \leq f(\gamma)$, or

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

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

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

CERTIFICATE



DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition(G, γ, c) either outputs

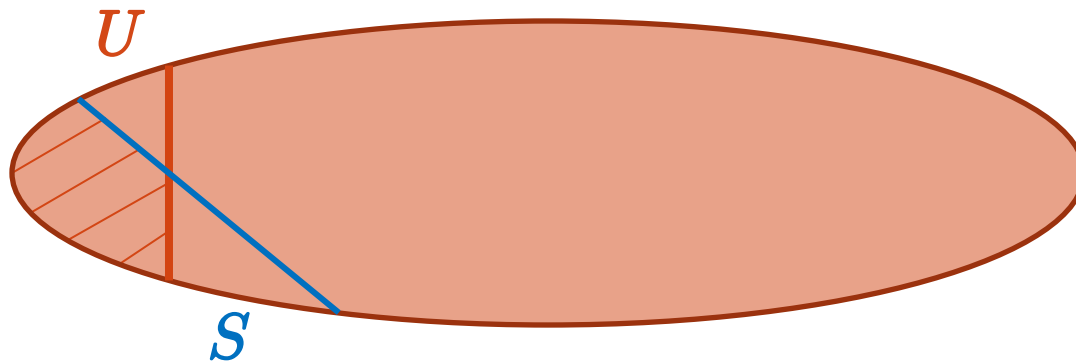
○ an $\Omega(c)$ -balanced cut T with $\phi(T) \leq f(\gamma)$, or

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

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

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

CERTIFICATE



DECOMPOSITION AND PARTITIONING

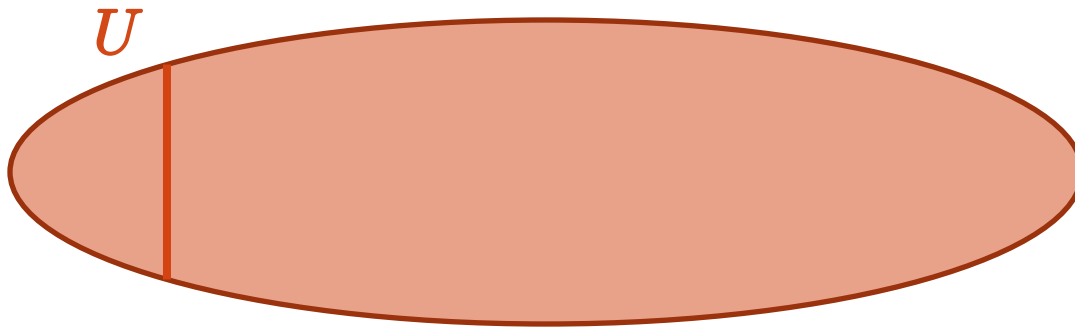
PARTITIONING ALGORITHM

Partition(G, γ, 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 $\phi(S) \leq \gamma$

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

SPECIAL
CERTIFICATE



SPARSE UNBALANCED CUT CORRELATED WITH ALL SPARSE UNBALANCED CUTS

DECOMPOSITION AND PARTITIONING

PARTITIONING ALGORITHM

Partition(G, γ, 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 $\phi(S) \leq \gamma$

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

DECOMPOSITION

Nearly-linear time
Partition(G, γ, c)

APPLY
→
RECURSIVELY

Can construct
 $(\alpha, f(\alpha) \log^2 n)$ -decomposition
in nearly-linear time

[Spielman, Teng '04]

OUR ALGORITHM

BALCUT(G, γ, 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 **special** certificate that for all c -balanced $S \subset V$,

$$\phi(S) \geq \gamma.$$

THE ALGORITHM

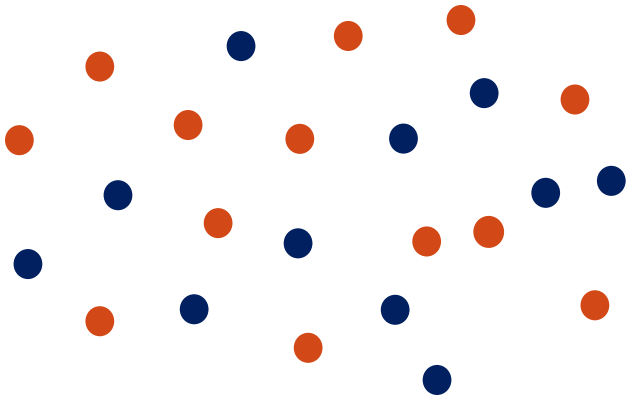
- For each t , keep graph G_t and a rate η_t . $G_0 = G$ and $\eta_0 = 1$.

● = +1 charge

● = -1 charge

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

- Consider G_t .
- Pick random assignment of ± 1 charge to the vertices.



THE ALGORITHM

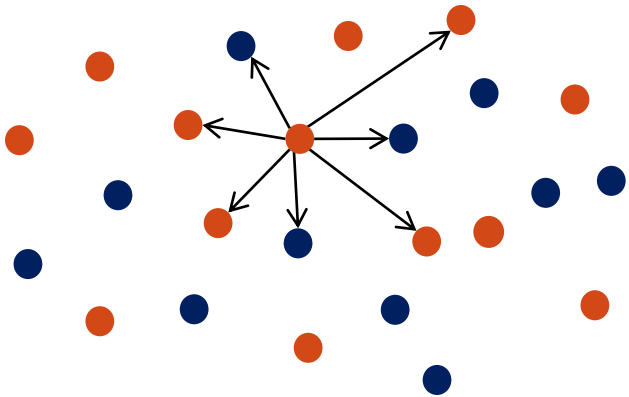
- For each t , keep graph G_t and a rate η_t . $G_0 = G$ and $\eta_0 = 1$.

● = +1 charge

● = -1 charge

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

- Consider G_t .
- Pick random assignment of ± 1 charge to the vertices.
- Mix charge along edges of G_t using **heat kernel with rate η_t** .



THE ALGORITHM

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

● = +1 charge



● = -1 charge

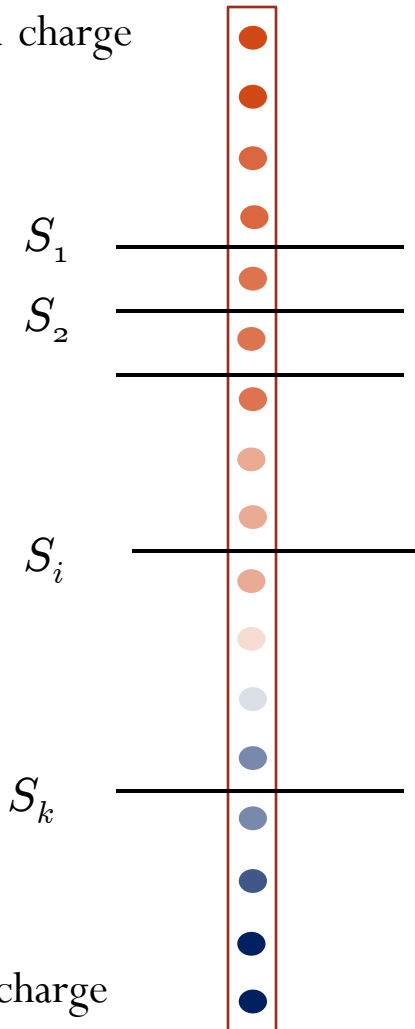
At iteration t , repeat $O(\log n)$ times:

- Consider G_t .
- Pick random assignment of ± 1 charge to the vertices.
- Mix charge along edges with rate η_t .
- Sort **final distribution** by charge.

THE ALGORITHM

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

● = +1 charge



● = -1 charge

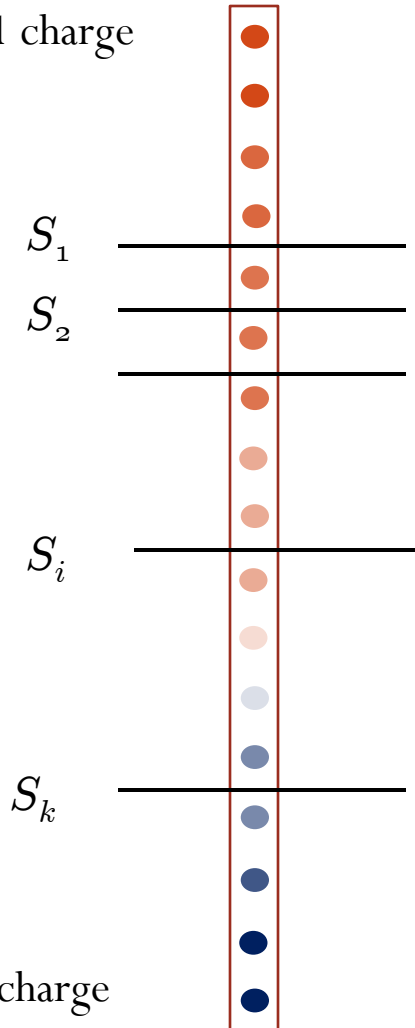
At iteration t , repeat $O(\log n)$ times:

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

THE ALGORITHM

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

● = +1 charge



● = -1 charge

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

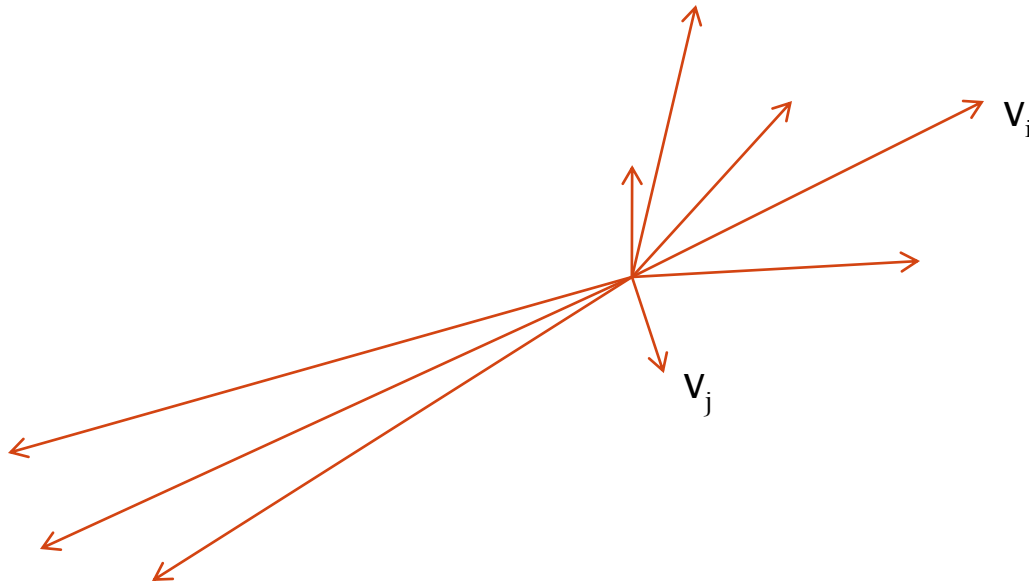
- Consider G_t .
- Pick random assignment of ± 1 charge to the vertices.
- Mix along edges with rate η_t .
- Consider final distribution, sorted by charge.
- Check all $\Omega(b)$ -balanced sweep cuts S_1, \dots, S_k for $\phi(S_i) \leq O(\gamma^{1/2})$ in G .

What if no sparse balanced cut is found?

THE ALGORITHM

- What if no sparse balanced cut is found?

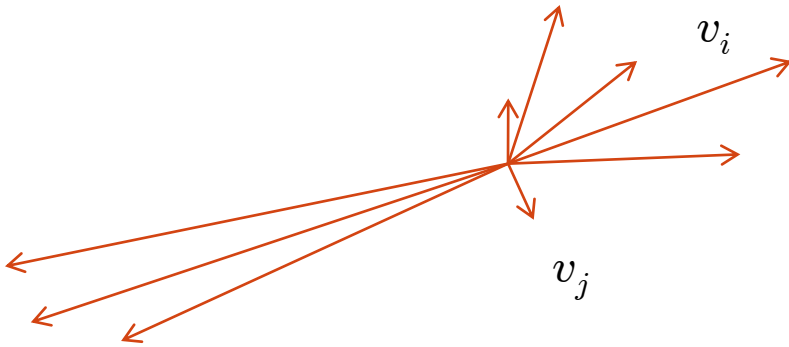
Consider the $O(\log n)$ -dimensional vector embedding of vertices given by the final distributions.



THE ALGORITHM

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.

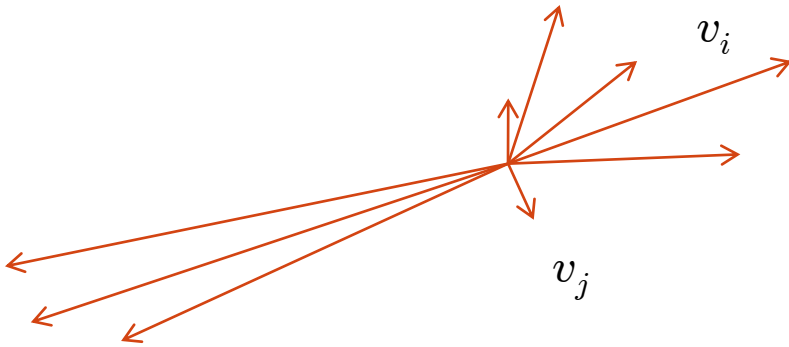


How to fix it?

Increase rate. $\eta_{t+1} = \eta_t + 1$

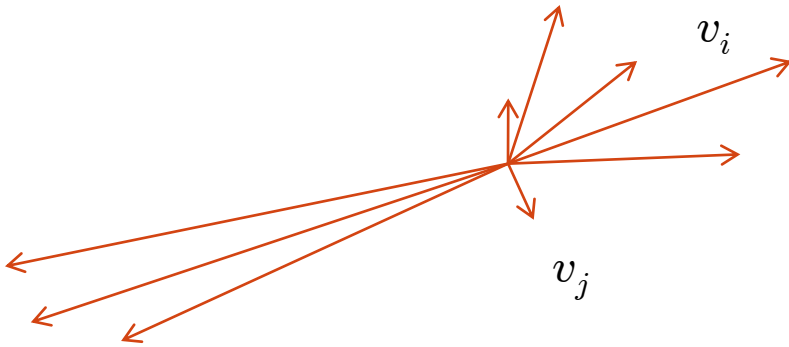
THE ALGORITHM

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

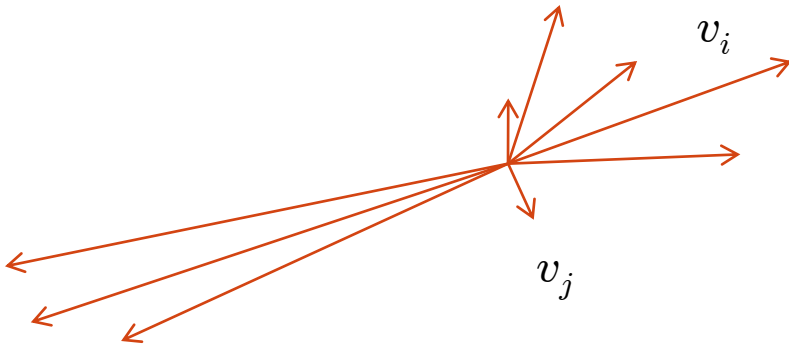


GOAL: find U such that

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

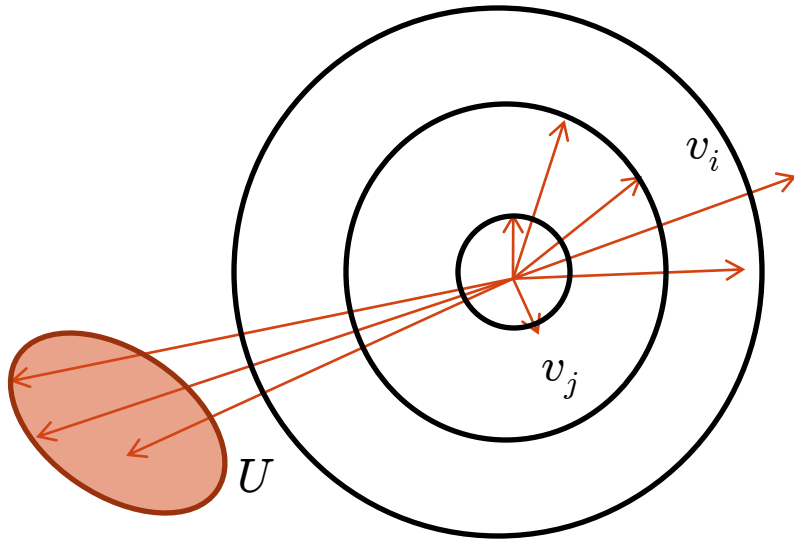
- $\phi(U) \cdot O(\sqrt{\gamma})$,

- U captures large fraction of the variance of the embedding.

$$\sum_{i \in U} \|v_i\|^2 \geq \Omega(1) \cdot \sum_{i \in V} \|v_i\|^2$$

THE ALGORITHM

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



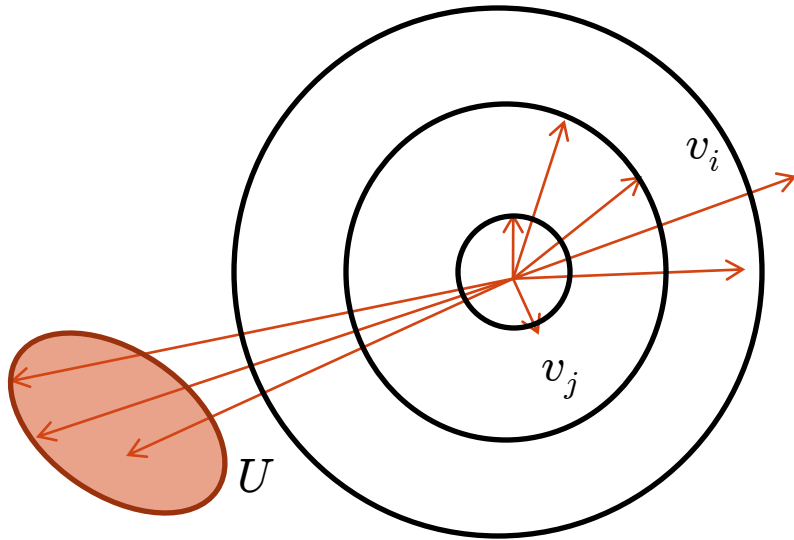
GOAL: find U such that

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

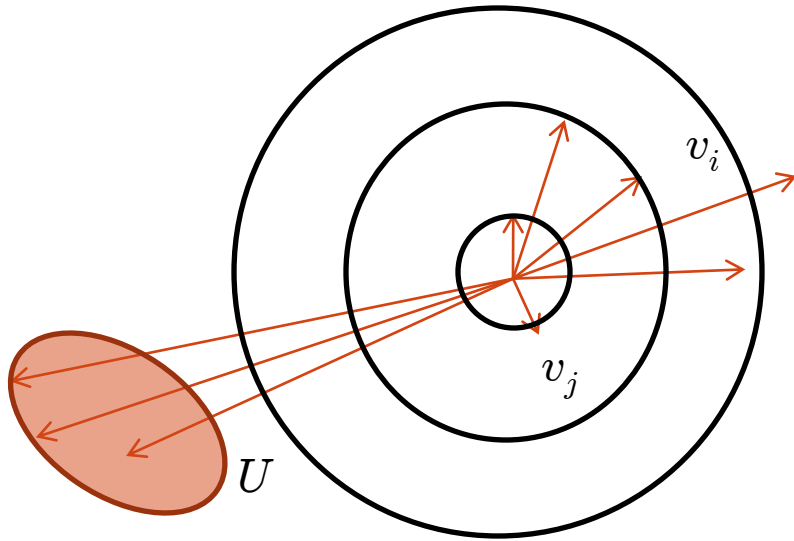
SOLUTION: Check all ball cuts centered around origin.

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

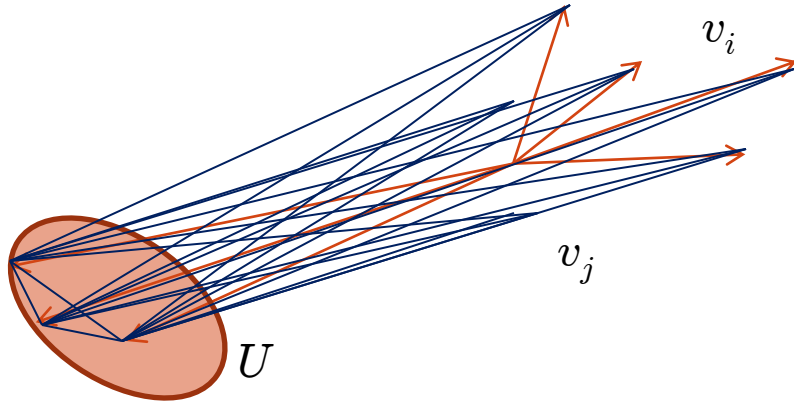
- $\phi(U) \cdot O(\sqrt{\gamma})$,
- U captures large fraction of the variance of the embedding.

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 γ .

- Running Time

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

time per
iteration

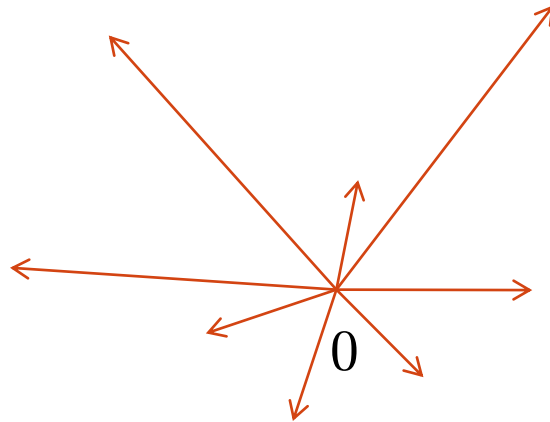
iterations

SDP FORMULATION

$$\mathbb{E}_{\{i,j\} \in E_G} \|v_i - v_j\|^2 \leq \gamma,$$

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

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

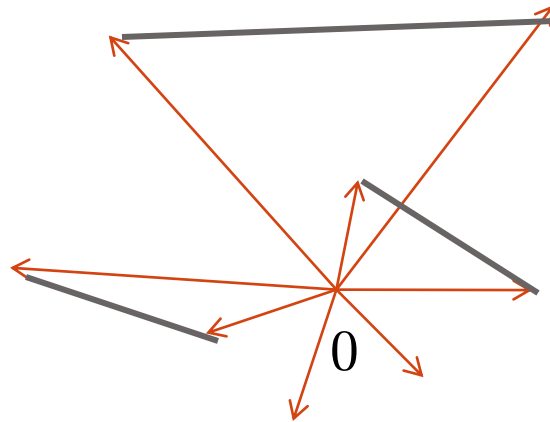


SDP FORMULATION

$$\mathbb{E}_{\{i,j\} \in E_G} \|v_i - v_j\|^2 \leq \gamma, \quad \text{SHORT EDGES}$$

$$\mathbb{E}_{\{i,j\} \in V \times V} \|v_i - v_j\|^2 = \frac{1}{2m}, \quad \text{FIXED VARIANCE}$$

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



SDP FORMULATION

$$\mathbb{E}_{\{i,j\} \in E_G} \|v_i - v_j\|^2 \leq \gamma,$$

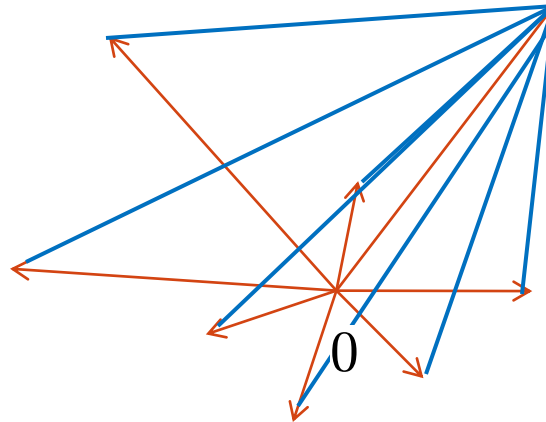
SHORT EDGES

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

FIXED VARIANCE

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

LENGTH OF
STAR EDGES



SHORT RADIUS

SEPARATION ORACLE

BROKEN FIRST CONSTRAINT

1. If $\sum_{\{i,j\} \in E} \|v_i - v_j\|^2 \geq \gamma \cdot \sum_{i \in V} \|v_i\|^2$

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

BROKEN STAR CONSTRAINTS

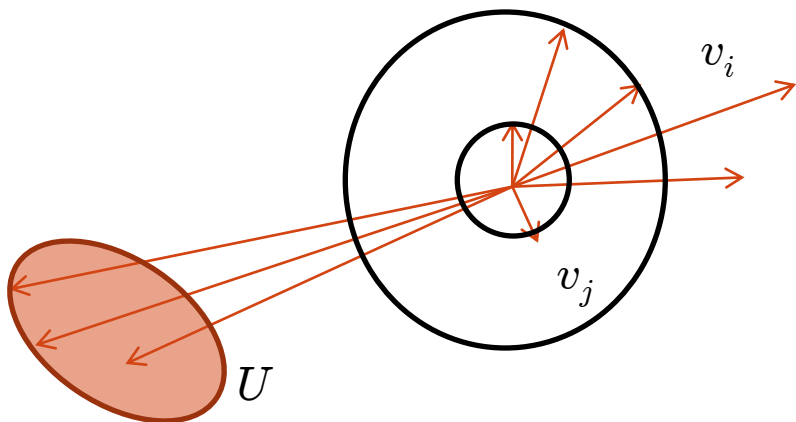
2. Otherwise, grow ball to find an unbalanced cut U such that

- U contains most of the variance of the embedding,

- $\phi(U) \leq O(\gamma^{1/2})$.

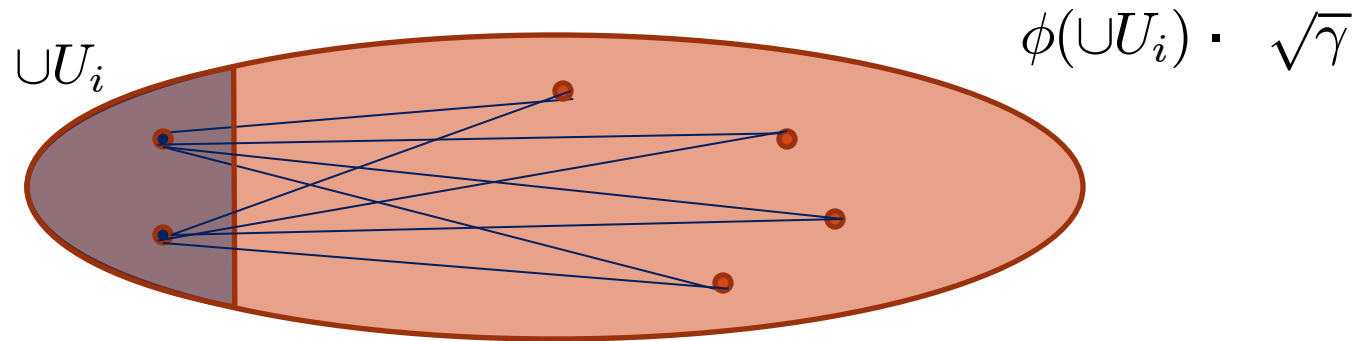
Set:

$$G_{t+1} = G_t + \frac{\gamma}{n} \sum_{i \in U} \text{Star}_i$$



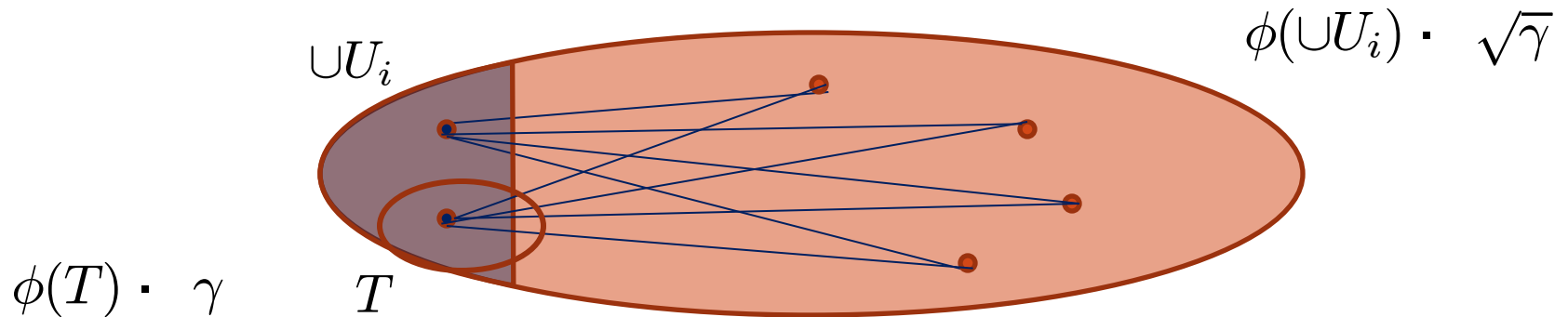
CERTIFICATE

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



CERTIFICATE

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



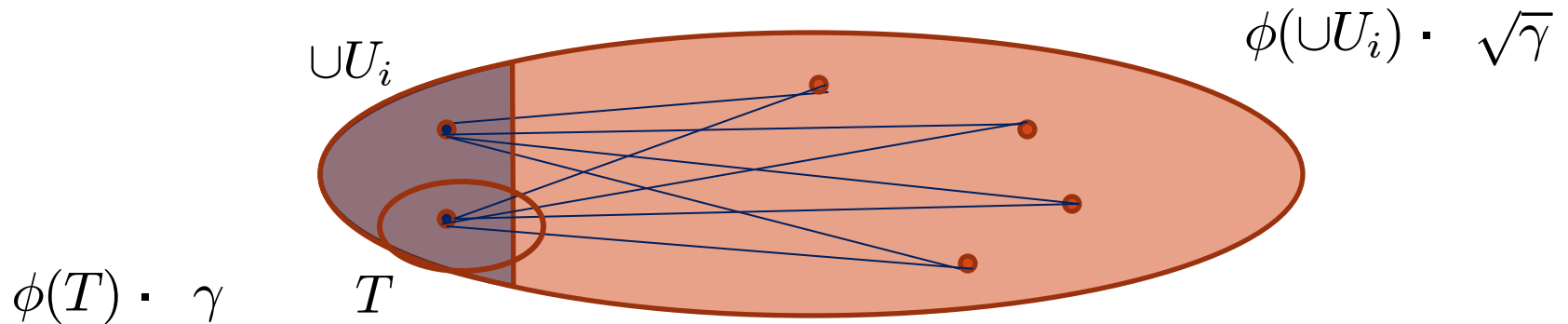
Any **sparse unbalanced cut** T must contain the root of many stars.



T is well-correlated with $\cup U_i$

CERTIFICATE

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



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



T is well-correlated with $\cup U_i$

THANK YOU