Random Walks as a Stable Analogue of Eigenvectors
(with Applications to Nearly-Linear-Time Graph Partitioning)

Lorenzo Orecchia, MIT Math

Based on joint works with Michael Mahoney (Stanford), Sushant Sachdeva (Yale) and Nisheeth Vishnoi (MSR India).
Why Spectral Algorithms for Graph Problems ...

... in practice?
- Simple to implement
- Can exploit very efficient linear algebra routines
- Perform well in practice for many problems

... in theory?
- Connections between spectral and combinatorial objects
- Connections to Markov Chains and Probability Theory
- Intuitive geometric viewpoint

RECENT ADVANCES:
Fast algorithms for fundamental combinatorial problems rely on spectral and optimization ideas
Spectral Algorithms for Graph Partitioning

Spectral algorithms are widely used in many graph-partitioning applications: clustering, image segmentation, community-detection, etc.

CLASSICAL VIEW:

- Based on Cheeger’s Inequality
- Eigenvectors sweep-cuts reveal sparse cuts in the graph
Spectral Algorithms for Graph Partitioning

Spectral algorithms are widely used in many graph-partitioning applications: clustering, image segmentation, community-detection, etc.

CLASSICAL VIEW:
- Based on Cheeger’s Inequality
- Eigenvectors sweep-cuts reveal sparse cuts in the graph

NEW TREND:
- Random walk vectors replace eigenvectors:
  • Fast Algorithms for Graph Partitioning
  • Local Graph Partitioning
  • Real Network Analysis
- Different random walks: PageRank, Heat-Kernel, etc.
Advantages of Random Walks:

1) Quick **approximation to eigenvector** in massive graphs

\[ A = \text{adjacency matrix} \quad D = \text{diagonal degree matrix} \]

\[ W = AD^{-1} = \text{natural random walk matrix} \quad L = D - A = \text{Laplacian matrix} \]

**Second Eigenvector of the Laplacian** can be computed by iterating \( W \) :

For random \( y_0 \) s.t. \( y_0^T D^{-1} 1 = 0 \), compute

\[ D^{-1} W^t y_0 \]
Why Random Walks? A Practitioner’s View

Advantages of Random Walks:

1) Quick **approximation to eigenvector** in massive graphs

- $A =$ adjacency matrix
- $D =$ diagonal degree matrix
- $W = AD^{-1} =$ natural random walk matrix
- $L = D - A =$ Laplacian matrix

**Second Eigenvector of the Laplacian** can be computed by iterating $W$:

For random $y_0$ s.t. $y_0^T D^{-1} 1 = 0$, compute

$$D^{-1} W^t y_0$$

In the limit, $x_2(L) = \lim_{t \to \infty} \frac{D^{-1} W^t y_0}{||W^t y_0||_{D^{-1}}}$. 
Why Random Walks? A Practitioner’s View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

\[ A = \text{adjacency matrix} \quad \quad D = \text{diagonal degree matrix} \]

\[ W = AD^{-1} = \text{natural random walk matrix} \quad L = D - A = \text{Laplacian matrix} \]

Second Eigenvector of the Laplacian can be computed by iterating \( W \) :

For random \( y_0 \) s.t. \( y_0^T D^{-1} 1 = 0 \) , compute

\[ D^{-1} W^t y_0 \]

In the limit, \( x_2(L) = \lim_{t \to \infty} \frac{D^{-1} W^t y_0}{\| W^t y_0 \|_{D^{-1}}} \).

Heuristic: For massive graphs, pick \( t \) as large as computationally affordable.
Why Random Walks? A Practitioner’s View

Advantages of Random Walks:

1) Quick approximation to eigenvector in massive graphs

2) Statistical robustness

Real-world graphs are noisy

GROUND TRUTH GRAPH
Why Random Walks? A Practitioner’s View

Advantages of Random Walks:
1) Quick approximation to eigenvector in massive graphs
2) Statistical robustness

Real-world graphs are noisy

**GOAL**: estimate eigenvector of ground-truth graph.
Why Random Walks? A Practitioner’s View

Advantages of Random Walks:
1) Quick approximation to eigenvector in massive graphs
2) Statistical robustness

GROUND-TRUTH GRAPH

**GOAL:** estimate eigenvector of ground-truth graph.

INPUT GRAPH

**OBSERVATION:** eigenvector of input graph can have very large variance, as it can be very sensitive to noise

RANDOM-WALK VECTORS provide better, more stable estimates.
This Talk

**QUESTION:**
Why random-walk vectors in the *design of fast algorithms*?
QUESTION:
Why random-walk vectors in the design of fast algorithms?

ANSWER: Stable, regularized version of the eigenvector
This Talk

**QUESTION:** Why random-walk vectors in the design of fast algorithms?

**ANSWER:** Stable, regularized version of the eigenvector

**GOALS OF THIS TALK:**
- Show optimization perspective on why random walks arise
- Application to nearly-linear-time balanced graph partitioning
Random Walks as Regularized Eigenvectors
What is Regularization?

Regularization is a fundamental technique in **optimization**

**OPTIMIZATION PROBLEM** → **WELL-BEHAVED OPTIMIZATION PROBLEM**

• Stable optimum
• Unique optimal solution
• Smoothness conditions
  ...

What is Regularization?

Regularization is a fundamental technique in optimization.

\[
\min_{x \in H} L(x) \quad \rightarrow \quad \min_{x \in H} L(x) + \lambda \cdot F(x)
\]

**Benefits of Regularization in Learning and Statistics:**
- Increases stability
- Decreases sensitivity to random noise
- Prevents overfitting
Instability of Eigenvector

EXPANDER
Instability of Eigenvector

Current eigenvector

1

\(-\epsilon\)

\(-\epsilon\)

\(-\epsilon\)

EXPANDER
Instability of Eigenvector

Current eigenvector

Eigenvector Changes Completely!
The Laplacian Eigenvalue Problem

Quadratic Formulation

\[
\frac{1}{d} \min x^T L x
\]

s.t. \( \|x\|_2 = 1 \)

\( x^T 1 = 0 \)

For simplicity, take G to be d-regular.
The Laplacian Eigenvalue Problem

Quadratic Formulation

\[
\frac{1}{d} \min x^T L x \\
\text{s.t. } \|x\|_2 = 1 \\
x^T 1 = 0
\]

SDP Formulation

\[
\frac{1}{d} \min L \bullet X \\
\text{s.t. } I \bullet X = 1 \\
11^T \bullet X = 0 \\
X \succeq 0
\]
The Laplacian Eigenvalue Problem

Quadratic Formulation

\[
\frac{1}{d} \min x^T L x \quad \text{s.t.} \quad \|x\|_2 = 1, \quad x^T 1 = 0
\]

SDP Formulation

\[
\frac{1}{d} \min L \cdot X \quad \text{s.t.} \quad I \cdot X = 1, \quad 11^T \cdot X = 0, \quad X \succeq 0
\]

Programs have the same optimum. Take optimal solution

\[
X^* = x^* (x^*)^T
\]
Instability of Linear Optimization

Consider a convex set $S \subseteq \mathbb{R}^n$ and a linear optimization problem:

$$f(c) = \arg \min_{x \in S} c^T x$$

The optimal solution $f(c)$ may be very unstable under perturbation of $c$:

$$\|c' - c\| \leq \delta \quad \text{and} \quad \|f(c') - f(c)\| \gg \delta$$
Consider a convex $S \subset \mathbb{R}^n$ and a **regularized** linear optimization problem

$$f(c) = \arg\min_{x \in S} c^T x + F(x)$$

where $F$ is $\sigma$-strongly convex.

Then: \[\|c' - c\| \leq \delta\] implies \[\|f(c) - f(c')\| \leq \frac{\delta}{\sigma}\]
Consider a convex set \( \mathcal{S} \subset \mathbb{R}^n \) and a regularized linear optimization problem

\[
f(c) = \arg \min_{x \in \mathcal{S}} \ c^T x + F(x)
\]

where \( F \) is \( \sigma \)-strongly convex.

Then:

\[
\| c' - c \| \leq \delta \quad \text{implies} \quad \| f(c) - f(c') \| \leq \frac{\delta}{\sigma}
\]

slope \( \leq \delta \)
Regularized Spectral Optimization

SDP Formulation

\[
\begin{align*}
\frac{1}{d} \min & \quad L \bullet X \\
\text{s.t.} & \quad I \bullet X = 1 \\
& \quad 11^T \bullet X = 0 \\
& \quad X \succeq 0
\end{align*}
\]

Density Matrix

Eigenvector decomposition of \( X \):

\[
X = \sum p_i v_i v_i^T
\]

\[
\begin{align*}
\forall i, p_i & \geq 0, \\
\sum p_i & = 1, \\
\forall i, v_i^T 1 & = 0.
\end{align*}
\]

Eigenvalues of \( X \) define probability distribution
Regularized Spectral Optimization

SDP Formulation

\[
\frac{1}{d} \min \ L \cdot X \\
\text{s.t.} \quad I \cdot X = 1 \\
J \cdot X = 0 \\
X \succeq 0
\]

Density Matrix

Eigenvalues of \( X \) define \textit{probability distribution}

\[
X^* = x^* (x^*)^T
\]

TRIVIAL DISTRIBUTION
Regularized Spectral Optimization

\[
\frac{1}{d} \min L \cdot X + \eta \cdot F(X) \quad \text{Regularizer } F \\
\text{Parameter } \eta \\
\text{s.t.} \quad I \cdot X = 1 \\
11^T \cdot X = 0 \\
X \succeq 0
\]

The regularizer \( F \) forces the distribution of eigenvalues of \( X \) to be non-trivial.

\[ X^* = x^* (x^*)^T \]

\[ X^* = \sum p_i v_i v_i^T \]
Regularizers

Regularizers are SDP-versions of common regularizers

- von Neumann Entropy
  \[ F_H(X) = \text{Tr}(X \log X) = \sum p_i \log p_i \]

- p-Norm, \( p > 1 \)
  \[ F_p(X) = \frac{1}{p} \|X\|_p^p = \frac{1}{p} \text{Tr}(X^p) = \frac{1}{p} \sum p_i^p \]

- And more, e.g. log-determinant.
Our Main Result

Regularized SDP

\[
\frac{1}{d} \min L \cdot X + \eta \cdot F(X)
\]

s.t.

\[
I \cdot X = 1 \\
J \cdot X = 0 \\
X \succeq 0
\]

**RESULT:** Explicit correspondence between regularizers and random walks

<table>
<thead>
<tr>
<th>REGULARIZER</th>
<th>OPTIMAL SOLUTION OF REGULARIZED PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(F = F_H)</td>
<td>(X^* \propto H_G^t) where (t) depends on (\eta)</td>
</tr>
<tr>
<td>(F = F_p)</td>
<td>(X^* \propto (qI + (1 - q)W)^{\frac{1}{p-1}}) where (q) depends on (\eta)</td>
</tr>
</tbody>
</table>
Our Main Result

\[
\frac{1}{d} \min \ L \cdot X + \eta \cdot F(X)
\]
\[
\text{s.t.} \quad I \cdot X = 1 \\
J \cdot X = 0 \\
X \succeq 0
\]

RESULT:
Explicit correspondence between regularizers and random walks

<table>
<thead>
<tr>
<th>REGULARIZER</th>
<th>OPTIMAL SOLUTION OF REGULARIZED PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F = F_H ) Entropy</td>
<td>( X^* \propto H^t_G ) where ( t ) depends on ( \eta )</td>
</tr>
<tr>
<td>( F = F_p ) ( p )-Norm</td>
<td>( X^* \propto (qI + (1 - q)W)^{\frac{1}{p-1}} ) where ( q ) depends on ( \eta )</td>
</tr>
</tbody>
</table>
Background: Heat-Kernel Random Walk

For simplicity, take $G$ to be \textbf{d-regular}.

- The Heat-Kernel Random Walk is a \textbf{Continuous-Time Markov Chain} over $V$, modeling the \textbf{diffusion of heat} along the edges of $G$.

- Transitions take place in \textbf{continuous time} $t$, with an \textbf{exponential} distribution.

\[
\frac{\partial p(t)}{\partial t} = -L \frac{p(t)}{d}
\]

\[
p(t) = e^{-\frac{t}{d}L}p(0)
\]

- The Heat Kernel can be interpreted as \textbf{Poisson distribution} over number of steps of the natural random walk $W=AD^{-1}$:

\[
e^{-\frac{t}{d}L} = e^{-t} \sum_{k=1}^{\infty} \frac{t^k}{k!}W^k
\]
Background: Heat-Kernel Random Walk

For simplicity, take $G$ to be \textbf{d-regular}.

• The Heat-Kernel Random Walk is a Continuous-Time Markov Chain over $V$, modeling the diffusion of heat along the edges of $G$.

• Transitions take place in continuous time $t$, with an exponential distribution.

\[
\frac{\partial p(t)}{\partial t} = -L \frac{p(t)}{d} \quad \text{Notation}
\]

\[
p(t) = e^{-\frac{t}{d}L} p(0) =: H_{G}^{t} p(0)
\]

• The Heat Kernel can be interpreted as \textbf{Poisson distribution} over number of steps of the natural random walk $W=AD^{-1}$:

\[
e^{-\frac{t}{d}L} = e^{-t} \sum_{k=1}^{\infty} \frac{t^{k}}{k!} W^{k}
\]
Heat Kernel Walk: Stability Analysis

Consider a convex set $\mathcal{S} \subset \mathbb{R}^n$ and a regularized linear optimization problem

$$f(c) = \arg \min_{x \in \mathcal{S}} c^T x + F(x)$$

where $F$ is $\sigma$-strongly convex.

Then: $\|c' - c\| \leq \delta$ implies $\|f(c) - f(c')\| \leq \frac{\delta}{\sigma}$
Heat Kernel Walk: Stability Analysis

Consider a convex set $S \subset \mathbb{R}^n$ and a regularized linear optimization problem

$$f(c) = \arg \min_{x \in S} c^T x + F(x)$$

where $F$ is $\sigma$-strongly convex.

Then: $\|c' - c\| \leq \delta$ implies $\|f(c) - f(c')\| \leq \frac{\delta}{\sigma}$

Analogous statement for Heat Kernel:

$$\|G' - G\|_\infty \leq \delta \quad \text{implies} \quad \left\| \frac{H_{G'}}{I \cdot H_{G'}} - \frac{H_G^T}{I \cdot H_G^T} \right\|_1 \leq \tau \cdot \delta$$
Applications to Graph Partitioning: Nearly-Linear-Time Balanced Cut
Partitioning Graphs - Conductance

Undirected unweighted $G = (V, E), |V| = n, |E| = m$

Conductance of $S \subseteq V$

$$\phi(S) = \frac{|E(S, \overline{S})|}{\min\{\text{vol}(S), \text{vol}(\overline{S})\}}$$
Partitioning Graphs – Balanced Cut

NP-HARD DECISION PROBLEM

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

\[ \phi(S) < \gamma \]
\[ \frac{1}{2} > \frac{\text{vol}(S)}{\text{vol}(V)} > b \]
NP-HARD DECISION PROBLEM
Does $G$ have a $b$-balanced cut of conductance $< \gamma$?

- Important primitive for many recursive algorithms.
- Applications to clustering and graph decomposition.
**Spectral Approximation Algorithms**

Does $G$ have a $b$-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>$\tilde{O}(mn)$</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>Local Random Walks</td>
<td>$O\left(\sqrt{\gamma \log n}\right)$</td>
<td>$\tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right)$</td>
</tr>
</tbody>
</table>
### Spectral Approximation Algorithms

**Does G have a \( b \)-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}(mn) )</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>Local Random Walks</td>
<td>( O\left(\sqrt{\gamma \log n}\right) )</td>
<td>( \tilde{O}\left(\frac{m}{\sqrt{\gamma}}\right) )</td>
</tr>
<tr>
<td>[Orecchia, Sachdeva, Vishnoi ’12]</td>
<td>Random Walks</td>
<td>( O(\sqrt{\gamma}) )</td>
<td>( \tilde{O}(m) )</td>
</tr>
</tbody>
</table>
Recursive Eigenvector Algorithm

**INPUT:** $(G, b, \gamma)$  \hspace{1cm} **DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?
Recursive Eigenvector Algorithm

**INPUT:** $(G, b, \gamma)$  
**DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?

- Compute eigenvector of $G$ and corresponding Laplacian eigenvalue $\lambda_2$
Recursive Eigenvector Algorithm

**INPUT:** $(G, b, \gamma)$  **DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?

- Compute eigenvector of $G$ and corresponding Laplacian eigenvalue $\lambda_2$
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find $S_1$ such that

$$\phi(S_1) \leq O(\sqrt{\gamma})$$
Recursive Eigenvector Algorithm

**INPUT:** \((G, b, \gamma)\)    **DECISION:** does there exists \(b\)-balanced \(S\) with \(\phi(S) < \gamma\)?

- Compute eigenvector of \(G\) and corresponding Laplacian eigenvalue \(\lambda_2\).
- If \(\lambda_2 \geq \gamma\), output **NO**. Otherwise, sweep eigenvector to find \(S_1\) such that \(\phi(S_1) \leq O(\sqrt{\gamma})\).
- If \(S_1\) is \((b/2)\)-balanced, Output \(S_1\). Otherwise, consider the graph \(G_1\) induced by \(G\) on \(V-S_1\) with self-loops replacing the edges going to \(S_1\).
Recursive Eigenvector Algorithm

**INPUT:** $(G, b, \gamma)$  \hspace{1cm} **DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?

- Compute eigenvector of $G$ and corresponding Laplacian eigenvalue $\lambda_2$
- If $\lambda_2 \geq \gamma$, output NO. Otherwise, sweep eigenvector to find $S_1$ such that
  \[ \phi(S_1) \leq O(\sqrt{\gamma}) \]
- If $S_1$ is $(b/2)$-balanced, Output $S_1$. Otherwise, consider the graph $G_1$ induced by $G$ on $V-S_1$ with self-loops replacing the edges going to $S_1$.
- Recurse on $G_1$. 

![Diagram of Recursive Eigenvector Algorithm](image)
Recursive Eigenvector Algorithm

**INPUT:** \((G, b, \gamma)\)  \hspace{1cm} **DECISION:** does there exists \(b\)-balanced \(S\) with \(\phi(S) < \gamma\)?

- Compute eigenvector of \(G\) and corresponding Laplacian eigenvalue \(\lambda_2\).
- If \(\lambda_2 \geq \gamma\), output **NO**. Otherwise, sweep eigenvector to find \(S_1\) such that
  \[
  \phi(S_1) \leq O(\sqrt{\gamma})
  \]
- If \(S_1\) is \((b/2)\)-balanced, Output \(S_1\). Otherwise, consider the graph \(G_1\) induced by \(G\) on \(V-S_1\) with self-loops replacing the edges going to \(S_1\).
- Recurse on \(G_1\).
Recursive Eigenvector Algorithm

**INPUT:** $(G, b, \gamma)$  **DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?

- Compute eigenvector of $G$ and corresponding Laplacian eigenvalue $\lambda_2$
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find $S_1$ such that
  \[ \phi(S_1) \leq O(\sqrt{\gamma}) \]
- If $S_1$ is $(b/2)$-balanced, output $S_1$. Otherwise, consider the graph $G_1$ induced by $G$ on $V-S_1$ with self-loops replacing the edges going to $S_1$.
- Recurse on $G_1$. 
Recursive Eigenvector Algorithm

**INPUT:** $(G, b, \gamma)$  **DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?

- Compute eigenvector of $G$ and corresponding Laplacian eigenvalue $\lambda_2$
- If $\lambda_2 \geq \gamma$, output NO. Otherwise, sweep eigenvector to find $S_1$ such that 
  \[ \phi(S_1) \leq O(\sqrt{\gamma}) \]
- If $S_1$ is $(b/2)$-balanced, Output $S_1$. Otherwise, consider the graph $G_1$ induced by $G$ on $V-S_1$ with self-loops replacing the edges going to $S_1$.
- Recurse on $G_1$.

LARGE INDUCED EXPANDER = NO-CERTIFICATE
**Recursive Eigenvector Algorithm**

**INPUT:** $(G, b, \gamma)$  
**DECISION:** does there exists $b$-balanced $S$ with $\phi(S) < \gamma$ ?

- Compute eigenvector of $G$ and corresponding Laplacian eigenvalue $\lambda_2$
- If $\lambda_2 \geq \gamma$, output **NO**. Otherwise, sweep eigenvector to find $S_1$ such that 
  $$\phi(S_1) \leq O(\sqrt{\gamma})$$
- If $S_1$ is $(b/2)$-balanced. Output $S_1$. Otherwise, consider the graph $G_1$ induced by $G$ on $V-S_1$ with self-loops replacing the edges going to $S_1$.
- Recurse on $G_1$.

**RUNNING TIME:** $\tilde{O}(m)$ per iteration, $O(n)$ iterations. Total: $\tilde{O}(mn)$
Recursive Eigenvector: The Worst Case

\[ \Omega(n) \] nearly-disconnected components

EXPANDER

Varying conductance
Recursive Eigenvector: The Worst Case

NB: Recursive Eigenvector eliminates one component per iteration. \(\Omega(n)\) iterations are necessary. Each iteration requires \(\Omega(m)\) time.
Recursive Eigenvector: The Worst Case

Note: Recursive Eigenvector eliminates one component per iteration. $\Omega(n)$ iterations are necessary. Each iteration requires $\Omega(mn)$ time.

Goal: Eliminate unbalanced low-conductance cuts faster.
Recursive Eigenvector: The Worst Case

STABILITY VIEW:

• Ideally, we would like to enforce progress: $\lambda_2(G_{t+1}) \gg \lambda_2(G_t)$

• Eigenvector may change completely at every iteration. Impossible to enforce any non-trivial relation between $\lambda_2(G_{t+1})$ and $\lambda_2(G_t)$
## Our Algorithm: Contributions

<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>$\tilde{O}(mn)$</td>
</tr>
<tr>
<td>OUR ALGORITHM</td>
<td>Random Walks</td>
<td>$O(\sqrt{\gamma})$</td>
<td>$\tilde{O}(m)$</td>
</tr>
</tbody>
</table>

**MAIN FEATURES:**
- Compute $O(\log n)$ global heat-kernel random-walk vectors at each iteration
- Unbalanced cuts are removed in $O(\log n)$ iterations
- Method to compute heat-kernel vectors in nearly-linear time

**TECHNICAL COMPONENTS:**
1) New iterative algorithm with a simple random walk interpretation
2) Novel analysis of Lanczos methods for computing heat-kernel vectors
Eliminating Unbalanced Cuts

• The graph eigenvector may be correlated with only one sparse unbalanced cut.
Eliminating Unbalanced Cuts

- The graph eigenvector may be correlated with only one sparse unbalanced cut.

- Consider the Heat-Kernel random walk-matrix $H_G^\tau$ for $\tau = \log n/\gamma$. 

  Probability vector for random walk started at vertex $i$ 

  Long vectors are slow-mixing random walks
Eliminating Unbalanced Cuts

• The graph eigenvector may be correlated with only one sparse unbalanced cut.

• Consider the Heat-Kernel random walk-matrix $H_G^\tau$ for $\tau = \log n/\gamma$. 

Unbalanced cuts of conductance $< \sqrt{\gamma}$
The graph eigenvector may be correlated with only one sparse unbalanced cut.

Consider the Heat-Kernel random walk-matrix $H_G^\tau$ for $\tau = \log n/\gamma$. 

Unbalanced cuts of conductance $< \sqrt{\gamma}$
Eliminating Unbalanced Cuts

• The graph eigenvector may be correlated with only one sparse unbalanced cut.

SINGLE VECTOR
SINGLE CUT

AFTER CUT REMOVAL ...

... eigenvector can change completely

• Consider the Heat-Kernel random walk-matrix $H^\tau_G$ for $\tau = \log n/\gamma$.

VECTOR EMBEDDING
MULTIPLE CUTS

... vectors do not change a lot
Our Algorithm for Balanced Cut

IDEA BEHIND OUR ALGORITHM:
Replace eigenvector in recursive eigenvector algorithm with
Heat-Kernel random walk $H_G^\tau$ for $\tau = \log n/\gamma$

Consider the embedding $\{v_i\}$ given by $H_G^\tau$:

$v_i = H_G^\tau e_i$
Our Algorithm for Balanced Cut

IDEA BEHIND OUR ALGORITHM:
Replace eigenvector in recursive eigenvector algorithm with Heat-Kernel random walk $H_G^\tau$ for $\tau = \log n / \gamma$.

Consider the embedding $\{v_i\}$ given by $H_G^\tau$:

$$v_i = H_G^\tau e_i$$

Chosen to emphasize cuts of conductance $\approx \gamma$.

Stationary distribution is uniform as $G$ is regular.
Our Algorithm for Balanced Cut

IDEA BEHIND OUR ALGORITHM:
Replace eigenvector in recursive eigenvector algorithm with Heat-Kernel random walk $H_G^\tau$ for $\tau = \log n/\gamma$

Consider the embedding $\{v_i\}$ given by $H_G^\tau$:

$$v_i = H_G^\tau e_i$$

MIXING:
Define the total deviation from stationary for a set $S \subseteq V$ for walk

$$\Psi(H_G^\tau, S) = \sum_{i \in S} ||v_i - \bar{1}/n||^2$$

FUNDAMENTAL QUANTITY TO UNDERSTAND CUTS IN $G$
Our Algorithm: Case Analysis

Recall:

\[
\tau = \log n / \gamma \quad \Psi(H_G^\tau, S) = \sum_{i \in S} \|H_G^\tau e_i - \vec{1}/n\|^2
\]

**CASE 1:** Random walks have **mixed**

\[
v_i = H_G^\tau e_i
\]

**ALL VECTORS ARE SHORT**

\[
\Psi(H_G^\tau, V) \leq \frac{1}{\text{poly}(n)}
\]
Our Algorithm: Case Analysis

Recall:

\[ \tau = \log n / \gamma \quad \Psi(\mathbf{H}_G^\tau, S) = \sum_{i \in S} \| \mathbf{H}_G^\tau e_i - \mathbf{1}/n \|^2 \]

**CASE 1:** Random walks have **mixed**

\[ v_i = \mathbf{H}_G^\tau e_i \]

**ALL VECTORS ARE SHORT**

\[ \Psi(\mathbf{H}_G^\tau, V) \leq \frac{1}{\text{poly}(n)} \]

\[ \lambda_2 \geq \Omega(\gamma) \]

\[ \phi_G \geq \Omega(\gamma) \]

By definition of \( \tau \)
Our Algorithm

\[ \tau = \log \frac{n}{\gamma} \]

\[ \Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \bar{1}/n||^2 \]

\[ v_i = H_G^\tau e_i \]

**CASE 2:** Random walks have **not** mixed

\[ \Psi(H_G^\tau, V) > \frac{1}{\text{poly}(n)} \]

We can either find an \( \Omega(b) \)-balanced cut with conductance \( O(\sqrt{\gamma}) \)
Our Algorithm

\[ \tau = \log \frac{n}{\gamma} \]

\[ \Psi(H_G^\tau, S) = \sum_{i \in S} \|H_G^\tau e_i - \bar{1}/n\|^2 \]

\[ v_i = H_G^\tau e_i \]

**CASE 2**: Random walks have not mixed

\[ \Psi(H_G^\tau, V) > \frac{1}{\text{poly}(n)} \]

We can either find an \( \Omega(b) \)-balanced cut with conductance \( O(\sqrt{\gamma}) \)
Our Algorithm

\[ \tau = \log n / \gamma \]

\[ \Psi(H_G^\tau, S) = \sum_{i \in S} \|H_G^\tau e_i - \bar{1}/n\|^2 \]

CASE 2: Random walks have not mixed

\[ \Psi(H_G^\tau, V) > \frac{1}{\text{poly}(n)} \]

We can either find an \( \Omega(b) \)-balanced cut with conductance \( O(\sqrt{\gamma}) \)

OR a ball cut yields \( S_1 \) such that \( \phi(S_1) \leq O(\sqrt{\gamma}) \) and

\[ \Psi(H_G^\tau, S_1) \geq \frac{1}{2} \Psi(H_G^\tau, V). \]
Our Algorithm: Iteration

\[ \tau = \log n / \gamma \]

\[ \Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \bar{1}/n||^2 \]

CASE 2: We found an unbalanced cut \( S_1 \) with \( \phi(S_1) \leq O(\sqrt{\gamma}) \) and

\[ \Psi(H_G^\tau, S_1) \geq \frac{1}{2} \Psi(H_G^\tau, V). \]

Modify \( G = G^{(1)} \) by adding edges across \( (S_1, \bar{S}_1) \) to construct \( G^{(2)} \).

Analogous to removing unbalanced cut \( S_1 \)
in Recursive Eigenvector algorithm.
Our Algorithm: Modifying G

CASE 2: We found an unbalanced cut $S_1$ with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\Psi(H_G^T, S_1) \geq \frac{1}{2} \Psi(H_G^T, V).$$

Modify $G = G^{(1)}$ by adding edges across $(S_1, \bar{S}_1)$ to construct $G^{(2)}$. 
Our Algorithm: Modifying G

**CASE 2:** We found an unbalanced cut $S_1$ with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\Psi(H_G^r, S_1) \geq \frac{1}{2} \Psi(H_G^r, V).$$

Modify $G = G^{(1)}$ by **adding edges** across $(S_1, \bar{S}_1)$ to construct $G^{(2)}$.

$$G^{(t+1)} = G^{(t)} + \gamma \sum_{i \in S_t} \text{Star}_i$$

where Star$_i$ is the **star graph** rooted at vertex $i$. 
**Our Algorithm: Modifying G**

**CASE 2:** We found an unbalanced cut $S_1$ with $\phi(S_1) \leq O(\sqrt{\gamma})$ and

$$\Psi(H_G^T, S_1) \geq \frac{1}{2} \Psi(H_G^T, V).$$

Modify $G = G^{(1)}$ by **adding edges** across $(S_1, \bar{S}_1)$ to construct $G^{(2)}$.

\[
G^{(t+1)} = G^{(t)} + \gamma \sum_{i \in S_t} \text{Star}_i
\]

where $\text{Star}_i$ is the **star graph** rooted at vertex $i$.

The random walk can now escape $S_1$ more easily.
Our Algorithm: Iteration

\[ \tau = \log n / \gamma \]

\[ \Psi(H_G^{\tau}, S') = \sum_{i \in S} ||H_G^{\tau} e_i - \bar{1}/n||^2 \]

**CASE 2:** We found an unbalanced cut \( S_1 \) with \( \phi(S_1) \leq O(\sqrt{\gamma}) \) and

\[ \Psi(H_G^{\tau}, S_1) \geq \frac{1}{2} \Psi(H_G^{\tau}, V). \]

Modify \( G = G^{(1)} \) by **adding edges** across \( (S_1, \bar{S}_1) \) to construct \( G^{(2)} \).

**POTENTIAL REDUCTION:**

\[ \Psi(H_G^{\tau(t+1)}, V) \leq \Psi(H_G^{\tau(t)}, V) - \frac{1}{2} \Psi(H_G^{\tau(t)}, S_t) \leq \frac{3}{4} \Psi(H_G^{\tau(t)}, V) \]
Our Algorithm: Iteration

\[ \tau = \log n / \gamma \]

\[ \Psi(H_G^\tau, S) = \sum_{i \in S} ||H_G^\tau e_i - \bar{1}/n||^2 \]

**CASE 2:** We found an unbalanced cut \( S_1 \) with \( \phi(S_1) \leq O(\sqrt{\gamma}) \) and

\[ \Psi(H_G^\tau, S_1) \geq \frac{1}{2} \Psi(H_G^\tau, V). \]

Modify \( G = G^{(1)} \) by adding edges across \( (S_1, \bar{S}_1) \) to construct \( G^{(2)} \).

**POTENTIAL REDUCTION:**

\[ \Psi(H_G^{(t+1)}, V) \leq \Psi(H_G^{(t)}, V) - \frac{1}{2} \Psi(H_G^{(t)}, S_t) \leq \frac{3}{4} \Psi(H_G^{(t)}, V) \]

**CRUCIAL APPLICATION OF STABILITY OF RANDOM WALK**
Summary and Potential Analysis

IN SUMMARY:
At every step $t$ of the recursion, we either
1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
Potential Reduction

IN SUMMARY:
At every step $t$ of the recursion, we either
1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
2. Find that

$$\Psi(H_{G(t)}^\tau, V) \leq \frac{1}{\text{poly}(n)}, \text{ OR}$$
Potential Reduction

IN SUMMARY:

At every step $t$ of the recursion, we either

1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
2. Find that

$$\Psi\left(H_{G(t)}^\tau, V\right) \leq \frac{1}{\text{poly}(n)}, \text{ OR}$$

3. Find an unbalanced cut $S_t$ of the required conductance, such that for the graph $G^{(t+1)}$, modified to have increased edges from $S_t$,

$$\Psi\left(H_{G(t+1)}^\tau, V\right) \leq \frac{3}{4} \Psi\left(H_{G(t)}^\tau, V\right)$$
Potential Reduction

IN SUMMARY:
At every step $t-1$ of the recursion, we either

1. Produce a $\Omega(b)$-balanced cut of the required conductance, OR
2. Find that
   \[ \Psi(H^\tau_{G(t)}, V) \leq \frac{1}{\text{poly}(n)}, \text{OR} \]
3. Find an unbalanced cut $S_t$ of the required conductance, such that for the process $P^{(t+1)}$, modified to have increased transitions from $S_t$,
   \[ \Psi(H^\tau_{G(t+1)}, V) \leq \frac{3}{4} \Psi(H^\tau_{G(t)}, V) \]

After $T=O(\log n)$ iterations, if no balanced cut is found:

\[ \Psi(H^\tau_{G(T)}, V) \leq \frac{1}{\text{poly}(n)} \]

From this guarantee, using the definition of $G^{(T)}$, we derive an SDP-certificate that no $b$-balanced cut of conductance $O(\gamma)$ exists in $G$.

NB: Only $O(\log n)$ iterations to remove unbalanced cuts.
Heat-Kernel and Certificates

• If no balanced cut of conductance is found, our potential analysis yields:

\[
\Psi(H^\tau_{G(x), V}) \leq \frac{1}{\text{poly}(n)} \rightarrow L + \gamma \sum_{j=1}^{T-1} \sum_{i \in S_j} L(\text{Star}_i) \geq \gamma L(K_V)
\]

Modified graph has \( \lambda_2 \geq \gamma \)

**CLAIM:** This is a certificate that no balanced cut of conductance \(< \gamma \) existed in \( G \).
Heat-Kernel and Certificates

• If no balanced cut of conductance is found, our potential analysis yields:

\[ \Psi(H^\tau_{G(T)}, V) \leq \frac{1}{\text{poly}(n)} \quad \rightarrow \quad L + \gamma \sum_{j=1}^{T-1} \sum_{i \in S_j} L(\text{Star}_i) \geq \gamma L(K_V) \]

Modified graph has \( \lambda_2 \geq \gamma \)

CLAIM: This is a certificate that no balanced cut of conductance \(< \gamma \) existed in G.

\[ \phi(T) \geq \gamma - \gamma \frac{\left| \bigcup S_j \right|}{|T|} \]
Heat-Kernel and Certificates

• If no balanced cut of conductance is found, our potential analysis yields:

$$\Psi(H^T_{G(T)}, V) \leq \frac{1}{\text{poly}(n)} \rightarrow L + \gamma \sum_{j=1}^{T-1} \sum_{i \in S_j} L(\text{Star}_i) \geq \gamma L(K_V)$$

Modified graph has $\lambda_2 \geq \gamma$

**CLAIM:** This is a certificate that no balanced cut of conductance $< \gamma$ existed in $G$.

$$\phi(T) \geq \gamma - \gamma \frac{|\cup S_j|}{|T|} \geq \gamma - \gamma \frac{b/2}{b} \geq \gamma / 2$$
Comparison with Recursive Eigenvector

**RECURSIVE EIGENVECTOR:**
We can only bound number of iterations by volume of graph removed. \( \Omega(n) \) iterations possible.

**OUR ALGORITHM:**
Use variance of random walk as potential. Only \( O(\log n) \) iterations necessary.

\[
\Psi(P, V) = \sum_{i \in V} ||P e_i - \bar{I}/n||^2 \\
\text{STABLE SPECTRAL NOTION OF POTENTIAL}
\]
Running Time

• Our Algorithm runs in $O(\log n)$ iterations.

• In one iteration, we perform some simple computation (projection, sweep cut) on the vector embedding $H^\tau_{G(t)}$. This takes time $\tilde{O}(md)$, where $d$ is the dimension of the embedding.

• Can use Johnson-Lindenstrauss to obtain $d = O(\log n)$.

• Hence, we only need to compute $O(\log^2 n)$ matrix-vector products

$$H^\tau_{G(t)} u$$

• We show how to perform one such product in time $\tilde{O}(m)$ for all $\tau$.

• OBSTACLE:

$\tau$, the mean number of steps in the Heat-Kernel random walk, is $\Omega (n^2)$ for path.
Conclusion

NOVEL ALGORITHMIC CONTRIBUTIONS

• Balanced-Cut Algorithm using Random Walks in time \( \tilde{O}(m) \)

MAIN IDEA
Random walks provide a very useful
stable analogue of the graph eigenvector
via regularization

OPEN QUESTION
More applications of this idea?
Applications beyond design of fast algorithms?
A Different Interpretation

THEOREM:
Suppose eigenvector $x$ yields an unbalanced cut $S$ of low conductance and no balanced cut of the required conductance. Then, $S$

$$\sum d_i x_i = 0$$

Then,

$$\sum_{i \in S} d_i x_i^2 \geq \frac{1}{2} \sum_{i \in V} d_i x_i^2.$$ 

In words, $S$ contains most of the variance of the eigenvector.
A Different Interpretation

THEOREM:
Suppose eigenvector $x$ yields an unbalanced cut $S$ of low conductance and no balanced cut of the required conductance.

Then,

$$\sum_{i \in S} d_i x_i^2 \geq \frac{1}{2} \sum_{i \in V} d_i x_i^2.$$

In words, $S$ contains most of the variance of the eigenvector.

QUESTION: Does this mean the graph induced by $G$ on $V - S$ is much closer to have conductance at least $\gamma$?
A Different Interpretation

THEOREM:
Suppose eigenvector $x$ yields an unbalanced cut $S$ of low conductance and no balanced cut of the required conductance.

Then,

$$\sum_{i \in S} d_i x_i^2 \geq \frac{1}{2} \sum_{i \in V} d_i x_i^2.$$ 

QUESTION: Does this mean the graph induced by $G$ on $V - S$ is much closer to have conductance at least $\gamma$?

ANSWER: NO. $x$ may contain little or no information about $G$ on $V - S$. Next eigenvector may be only infinitesimally larger.

CONCLUSION: To make significant progress, we need an analogue of the eigenvector that captures sparse
THEOREM 1: (WALKS HAVE NOT MIXED)

\[ \Psi(P^{(t)}, V) > \frac{1}{\text{poly}(n)} \]

Can find cut of conductance \( O(\sqrt{\gamma}) \)
Theorems for Our Algorithm

**THEOREM 1: (WALKS HAVE NOT MIXED)**

\[ \Psi(P^{(t)}, V) > \frac{1}{\text{poly}(n)} \]

Can find cut of conductance \( O(\sqrt{\gamma}) \)

**Proof:** Recall that

\[ P^{(t)} = e^{-\tau Q^{(t)}} \]
\[ \tau = \log n / \gamma \]
\[ \Psi(P, V) = \sum_{i \in V} ||P e_i - \vec{1}/n||^2 \]

Use the definition of \( \tau \). The spectrum of \( P^{(t)} \) implies that

\[ \sum_{i,j \in E} ||P^{(t)} e_i - P^{(t)} e_j||^2 \cdot O(\gamma) \cdot \Psi(P^{(t)}, V) \]

\[ \cdot \text{EDGE LENGTH} \]
\[ \cdot \text{TOTAL VARIANCE} \]
Theorems for Our Algorithm

**THEOREM 1: (WALKS HAVE NOT MIXED)**

\[ \Psi(P^{(t)}, V) > \frac{1}{\text{poly}(n)} \]

Can find cut of conductance \( O(\sqrt{\gamma}) \)

**Proof:** Recall that

\[ P^{(t)} = e^{-\tau Q^{(t)}} \quad \tau = \log n / \gamma \]

\[ \Psi(P, V) = \sum_{i \in V} \|Pe_i - \vec{1}/n\|^2 \]

Use the definition of \( \tau \). The spectrum of \( P^{(t)} \) implies that

\[ \sum_{i,j \in E} \|P^{(t)}e_i - P^{(t)}e_j\|^2 \cdot O(\gamma) \cdot \Psi(P^{(t)}, V) \]

Hence, by a random projection of the embedding \( \{P_e_i\} \), followed by a sweep cut, we can recover the required cut.

SDP ROUNDING TECHNIQUE
THEOREM 2: (WALKS HAVE MIXED)

$$
\Psi(P(t), V) \cdot \frac{1}{\text{poly}(n)} \rightarrow \text{No } \Omega(b)-\text{balanced cut of conductance } O(\gamma)
$$
Theorems for Our Algorithm

THEOREM 2: (WALKS HAVE MIXED)

\[ \Psi(P^{(t)}, V) \cdot \frac{1}{\text{poly}(n)} \Rightarrow \text{No } \Omega(b)\text{-balanced cut of conductance } O(\gamma) \]

Proof: Consider \( S = \bigcup S_i \). Notice that \( S \) is unbalanced.

Assumption is equivalent to

\[ L(K_V) \cdot e^{-\tau L - O(\log n)} \sum_{i \in S} L(S_i) \cdot \frac{1}{\text{poly}(n)}. \]
Theorems for Our Algorithm

THEOREM 2: (WALKS HAVE MIXED)

\[ \Psi(P^{(t)}, V) \cdot \frac{1}{\text{poly}(n)} \rightarrow \text{No } \Omega(b)\text{-balanced cut of conductance } O(\gamma) \]

Proof: Consider \( S = \bigcup S_i \). Notice that \( S \) is unbalanced.
Assumption is equivalent to

\[ L(K_V) \cdot e^{-\tau L - O(\log n)} \sum_{i \in S} L(S_i) \cdot \frac{1}{\text{poly}(n)}. \]

By taking logs,

\[ L + O(\gamma) \sum_{i \in S} L(S_i) \geq \Omega(\gamma)L(K_V). \]

SDP DUAL CERTIFICATE
Theorems for Our Algorithm

**THEOREM 2: (WALKS HAVE MIXED)**

\[ \Psi(P(t), V) \cdot \frac{1}{\text{poly}(n)} \rightarrow \text{No } \Omega(b)-\text{balanced cut of conductance } O(\gamma) \]

**Proof:** Consider \( S = \cup S_i \). Notice that \( S \) is unbalanced.

Assumption is equivalent to

\[ L(K_V) \cdot e^{-\tau L - O(\log n)} \sum_{i \in S} L(S_i) \cdot \frac{1}{\text{poly}(n)}. \]

By taking logs,

\[ L + O(\gamma) \sum_{i \in S} L(S_i) \geq \Omega(\gamma)L(K_V). \]

This is a certificate that no \( \Omega(1) \)-balanced cut of conductance \( O(\gamma) \) exists, as evaluating the quadratic form for a vector representing a balanced cut \( U \) yields

\[ \phi(U) \geq \Omega(\gamma) - \frac{\text{vol}(S)}{\text{vol}(U)}O(\gamma) \geq \Omega(\gamma) \]

as long as \( S \) is sufficiently unbalanced.
SDP Interpretation

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

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

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

SHORT EDGES

FIXED VARIANCE

LENGTH OF STAR EDGES
SDP Interpretation

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

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

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

- SHORT EDGES
- FIXED VARIANCE
- LENGTH OF STAR EDGES
- SHORT RADIUS
Background: Heat-Kernel Random Walk

For simplicity, take $G$ to be d-regular.

• The Heat-Kernel Random Walk is a Continuous-Time Markov Chain over $V$, modeling the diffusion of heat along the edges of $G$.

• Transitions take place in continuous time $t$, with an exponential distribution.

$$\frac{\partial p(t)}{\partial t} = -L \frac{p(t)}{d}$$

$$p(t) = e^{-\frac{t}{d}L}p(0) =: H_{G}^{t} p(0)$$

• The Heat Kernel can be interpreted as Poisson distribution over number of steps of the natural random walk $W = A D^{-1}$.

$$e^{-\frac{t}{d}L} = e^{-t} \sum_{k=1}^{\infty} \frac{t^{k}}{k!} W^{k}$$

• In practice, can replace Heat-Kernel with natural random walk $W^{t}$