

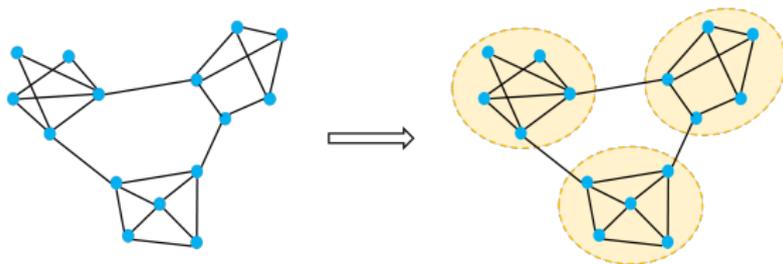
# A Sublinear-Time Spectral Clustering Oracle with Improved Preprocessing Time

Ranran Shen Pan Peng

University of Science and Technology of China

# Graph Clustering

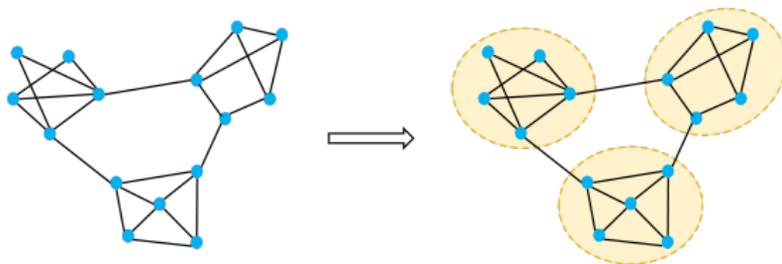
- ▶ **Input:**  $G = (V, E)$  and  $k$  ( $k \geq 2$ )
- ▶ **Goal:** partition  $V$  into  $k$  disjoint clusters  $C_1, \dots, C_k$ , such that each cluster exhibits
  - ▶ tight connections inside
  - ▶ loose connections outside



Example ( $k = 3$ )

# Graph Clustering

- ▶ **Input:**  $G = (V, E)$  and  $k$  ( $k \geq 2$ )
- ▶ **Goal:** partition  $V$  into  $k$  disjoint clusters  $C_1, \dots, C_k$ , such that each cluster exhibits
  - ▶ tight connections inside
  - ▶ loose connections outside

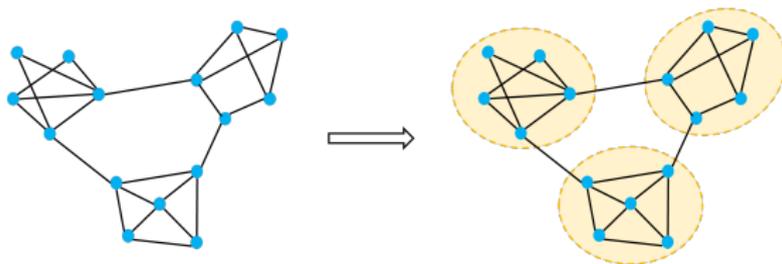


Example ( $k = 3$ )

Global algorithms run in  $\text{poly}(n)$  time:  $n$  increases  $\Rightarrow$  impractical ( $n = |V|$ )

# Graph Clustering

- ▶ **Input:**  $G = (V, E)$  and  $k$  ( $k \geq 2$ )
- ▶ **Goal:** partition  $V$  into  $k$  disjoint clusters  $C_1, \dots, C_k$ , such that each cluster exhibits
  - ▶ tight connections inside
  - ▶ loose connections outside



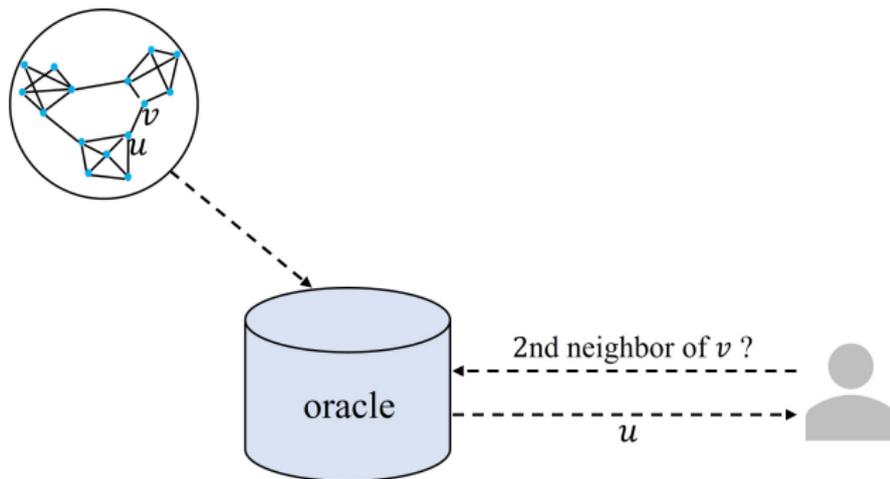
Example ( $k = 3$ )

Global algorithms run in  $\text{poly}(n)$  time:  $n$  increases  $\Rightarrow$  impractical ( $n = |V|$ )

We focus on **sublinear-time spectral clustering oracles**.

# Query Access

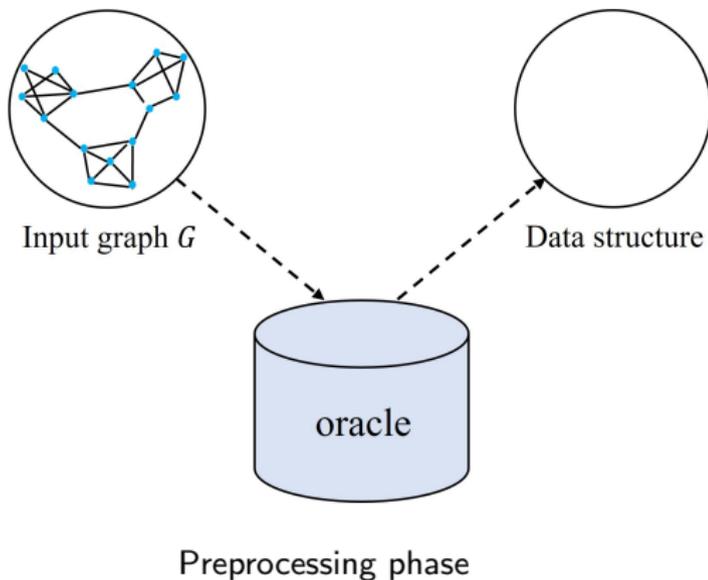
- ▶ Has **query access** to the adjacency list of the input graph  $G$



Neighbor query

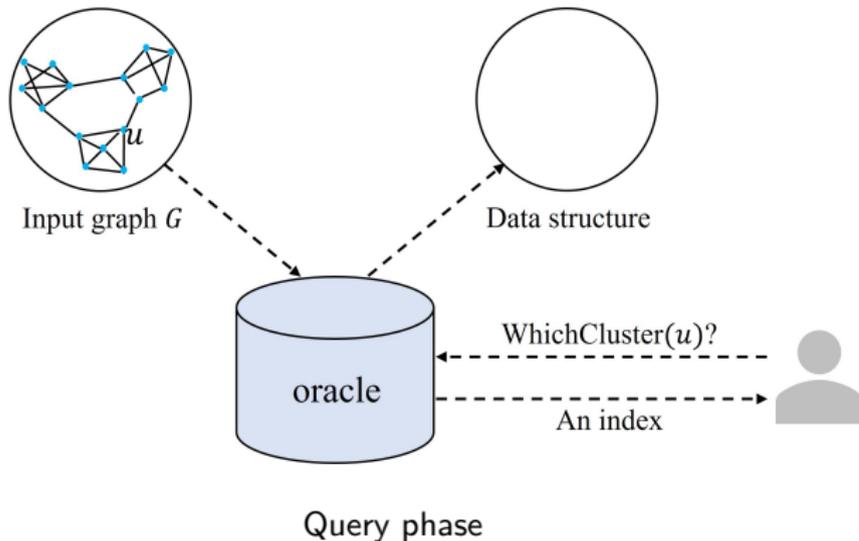
# Two Phases

- ▶ **Preprocessing phase** (sublinear-time)
  - ▶ build a data structure



# Two Phases

- ▶ **Preprocessing phase** (sublinear-time)
  - ▶ build a data structure
- ▶ **Query phase** (sublinear-time)
  - ▶ answer  $\text{WHICHCLUSTER}(v)$  queries



# Requirements

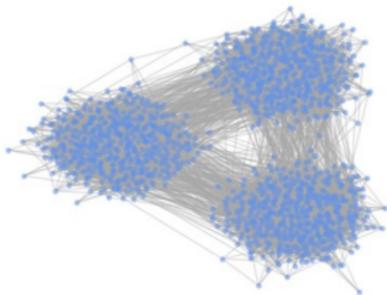
- ▶ Consistent
- ▶ Close to the ground-truth clustering

# More Formally About Input Graph $G$

- ▶  $d$ -bounded graphs: maximum degree  $\leq d$

# More Formally About Input Graph $G$

- ▶  **$d$ -bounded** graphs: maximum degree  $\leq d$

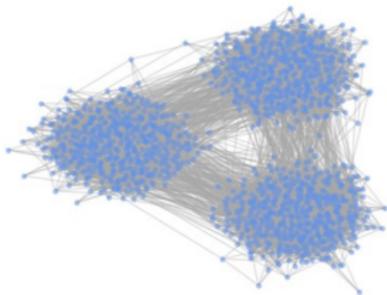


Example: a clusterable graph

- ▶  **$(k, \varphi, \varepsilon)$ -clusterable** graphs ( $\varepsilon \ll \varphi$ )

# More Formally About Input Graph $G$

- ▶  **$d$ -bounded** graphs: maximum degree  $\leq d$

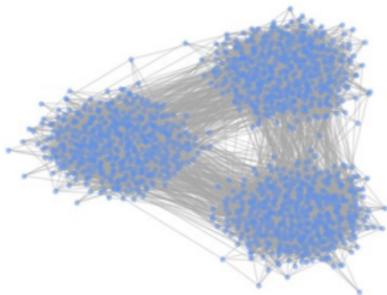


Example: a clusterable graph

- ▶  **$(k, \varphi, \varepsilon)$ -clusterable** graphs ( $\varepsilon \ll \varphi$ )
  - ▶ has a  **$k$ -partition** of  $V$ , denoted by  $C_1, \dots, C_k$ ,  $\frac{|C_i|}{|C_j|} \in O(1)$

# More Formally About Input Graph $G$

- ▶  **$d$ -bounded** graphs: maximum degree  $\leq d$

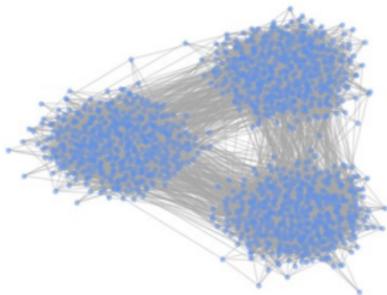


Example: a clusterable graph

- ▶  **$(k, \varphi, \varepsilon)$ -clusterable** graphs ( $\varepsilon \ll \varphi$ )
  - ▶ has a  **$k$ -partition** of  $V$ , denoted by  $C_1, \dots, C_k$ ,  $\frac{|C_i|}{|C_j|} \in O(1)$
  - ▶ tight connections inside: inner conductance  $\phi_{\text{in}}(C_i) \geq \varphi$

# More Formally About Input Graph $G$

- ▶  **$d$ -bounded** graphs: maximum degree  $\leq d$



Example: a clusterable graph

- ▶  **$(k, \varphi, \varepsilon)$ -clusterable** graphs ( $\varepsilon \ll \varphi$ )
  - ▶ has a  **$k$ -partition** of  $V$ , denoted by  $C_1, \dots, C_k$ ,  $\frac{|C_i|}{|C_j|} \in O(1)$
  - ▶ tight connections inside: inner conductance  $\phi_{\text{in}}(C_i) \geq \varphi$
  - ▶ loose connections outside: outer conductance  $\phi_{\text{out}}(C_i, V) \leq \varepsilon$

# Previous Work

	[Pen20]	[GKL+21]
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k) \cdot \log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$
preprocessing time	$O\left(\sqrt{n} \cdot \text{poly}\left(\frac{k \log n}{\varepsilon}\right)\right)$	$2^{\text{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)} \cdot \text{poly}(\log n)$
query time	$O\left(\sqrt{n} \cdot \text{poly}\left(\frac{k \log n}{\varepsilon}\right)\right)$	$n^{1/2+O(\varepsilon)} \cdot \text{poly}\left(\frac{k \log n}{\varepsilon}\right)$
misclassification error (fraction)	$O(k\sqrt{\varepsilon})$	$O(\log k \cdot \varepsilon)$ per cluster

# Motivation

	[Pen20]	[GKL+21]
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k) \cdot \log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$
preprocessing time	$O\left(\sqrt{n} \cdot \text{poly}\left(\frac{k \log n}{\varepsilon}\right)\right)$	$2^{\text{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)} \cdot \text{poly}(\log n)$
query time	$O\left(\sqrt{n} \cdot \text{poly}\left(\frac{k \log n}{\varepsilon}\right)\right)$	$n^{1/2+O(\varepsilon)} \cdot \text{poly}\left(\frac{k \log n}{\varepsilon}\right)$
misclassification error (fraction)	$O(k\sqrt{\varepsilon})$	$O(\log k \cdot \varepsilon)$ per cluster

Can we get a spectral clustering oracle with

- ▶ better conductance gap than [Pen20] and
- ▶ better preprocessing time than [GKL+21]?

# Our Results

	[Pen20]	[GKL+21]	this work
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k) \cdot \log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k)}$
preprocessing time	$\tilde{O}(\sqrt{n} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(2^{\text{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)})$	$\tilde{O}(\text{poly}(k) \cdot n^{1/2+O(\varepsilon)})$
query time	$\tilde{O}(\sqrt{n} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(n^{1/2+O(\varepsilon)} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(n^{1/2+O(\varepsilon)} \cdot \text{poly}(k))$
misclassification error (fraction)	$O(k\sqrt{\varepsilon})$	$O(\log k \cdot \varepsilon)$ per cluster	$O(\text{poly}(k) \cdot \varepsilon^{1/3})$ per cluster

# Our Results

	[Pen20]	[GKL+21]	this work
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k) \cdot \log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k)}$
preprocessing time	$\tilde{O}(\sqrt{n} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(2^{\text{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)})$	$\tilde{O}(\text{poly}(k) \cdot n^{1/2+O(\varepsilon)})$
query time	$\tilde{O}(\sqrt{n} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(n^{1/2+O(\varepsilon)} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(n^{1/2+O(\varepsilon)} \cdot \text{poly}(k))$
misclassification error (fraction)	$O(k\sqrt{\varepsilon})$	$O(\log k \cdot \varepsilon)$ per cluster	$O(\text{poly}(k) \cdot \varepsilon^{1/3})$ per cluster

- ▶ Conductance gap:  $\text{poly}(k)$ 
  - ▶ better than  $\text{poly}(k) \cdot \log n$  in [Pen20]
  - ▶ a slightly worse than  $\log k$  in [GKL+21]

# Our Results

	[Pen20]	[GKL+21]	this work
conductance gap	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k) \cdot \log n}$	$\varepsilon \ll \frac{\varphi^3}{\log k}$	$\varepsilon \ll \frac{\varphi^2}{\text{poly}(k)}$
preprocessing time	$\tilde{O}(\sqrt{n} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(2^{\text{poly}(\frac{k}{\varepsilon})} \cdot n^{1/2+O(\varepsilon)})$	$\tilde{O}(\text{poly}(k) \cdot n^{1/2+O(\varepsilon)})$
query time	$\tilde{O}(\sqrt{n} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(n^{1/2+O(\varepsilon)} \cdot \text{poly}(\frac{k}{\varepsilon}))$	$\tilde{O}(n^{1/2+O(\varepsilon)} \cdot \text{poly}(k))$
misclassification error (fraction)	$O(k\sqrt{\varepsilon})$	$O(\log k \cdot \varepsilon)$ per cluster	$O(\text{poly}(k) \cdot \varepsilon^{1/3})$ per cluster

- ▶ Conductance gap: **poly(k)**
  - ▶ better than  $\text{poly}(k) \cdot \log n$  in [Pen20]
  - ▶ a slightly worse than  $\log k$  in [GKL+21]
- ▶ Preprocessing time: **polynomial in k**, better than exponential in [GKL+21]

# Our Results

Our oracle is robust against a few edge deletions.

## Theorem (Robust; Informal)

Let  $G_0 = (V, E)$  be a  $(k, \varphi, \varepsilon)$ -clusterable graph, where  $\frac{\varepsilon}{\varphi^4} \ll \frac{1}{\text{poly}(k)}$ .

- ▶  $G$  is obtained from  $G_0$  by deleting at most  $O(d\varphi^2)$  edges in each cluster, or
- ▶  $G$  is obtained from  $G_0$  by randomly deleting at most  $O\left(\frac{kd^2}{d+\log k}\right)$  edges in  $G_0$

Then w.h.p., there exists a clustering oracle for  $G$  with the same guarantees as presented in above table.

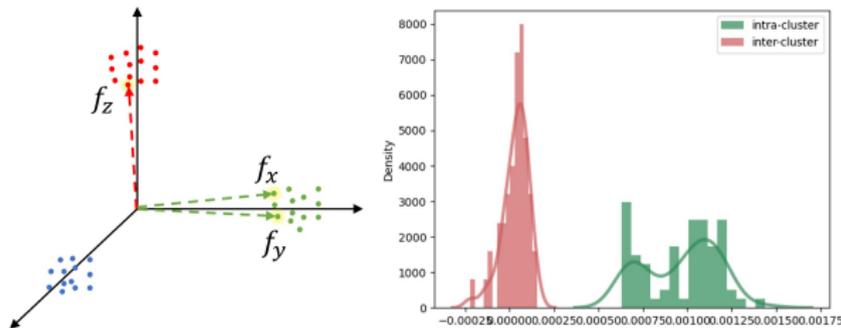
# Our Technique: A Nice Gap

$f_x$ : spectral embedding of  $x \in V$ .  $\langle f_x, f_y \rangle$ : dot product of  $f_x$  and  $f_y$ .

## Lemma 1 (Informal)

For most vertices in a  $(k, \varphi, \varepsilon)$ -clusterable graph,

- ▶ if  $x$  and  $y$  are in the **same** cluster, then  $\langle f_x, f_y \rangle$  is close to  $O(\frac{k}{n})$
- ▶ if  $x$  and  $z$  are in the **different** clusters, then  $\langle f_x, f_z \rangle$  is close to **0**.



Example: dot product gap

# Experiments

Input graph: generated by SBM

- ▶ can handle graphs with a **smaller conductance gap** than [CPS15]

$p$	0.02	0.025	0.03	0.035	0.04	0.05	0.06	0.07
error([CPS15])	-	0.6208	0.4970	0.1996	0.0829	0.0168	0.0030	0.0003
error(this work)	0.3887	0.0030	0.0004	0	0	0	0	0

- ▶ **robust** against a few adversarial edge deletions

delNum	0	25	32	40	45	50	55	60	65
error	0.0	0.00007	0.00007	0.00013	0.00047	0.00080	0.00080	0.00080	0.00087

# Thanks!

## References:

[CPS15] Czumaj A, Peng P, Sohler C. Testing cluster structure of graphs. STOC 2015.

[Pen20] Peng P. Robust clustering oracle and local reconstructor of cluster structure of graphs. SODA 2020.

[GKL<sup>+</sup>21] Gluch G, Kapralov M, Lattanzi S, Mousavifar A and Sohler C. Spectral clustering oracles in sublinear time. SODA 2021.