Differential Privacy from Locally Adjustable Graph Algorithms:

k-Core Decomposition, Low Out-Degree Ordering, and Densest Subgraphs

Laxman Dhulipala, Quanquan C. Liu, Sofya Raskhodnikova, Jessica Shi, Julian Shun, Shangdi Yu
Publishing Sensitive Graph Information

• Potentially sensitive connections between individuals published as graphs
  • Financial transactions
  • Relationship information
  • Email and cell phone communications
  • Search data
  • Disease network data
Publishing Sensitive Graph Information

- Potentially **sensitive connections between individuals** published as graphs
  - Financial transactions
  - Relationship information
  - Email and cell phone communications
  - Search data
  - Disease network data

**Anonymization ≠ Privacy!**
Publishing Sensitive Graph Information

- Potentially **sensitive connections between individuals** published as graphs
  - Financial transactions
  - Relationship information
  - Email and cell phone communications
  - Search data
  - Disease network data
  - COVID transmission data

Anonymization ≠ Privacy!
Private Analysis of Graph Data

Graph $G$ → Trusted Curator → Users

Graphs, Trusted Curator, and Users

Graph $G$

Trusted Curator

Queries → Answers

Users: Researchers, Government, Businesses, and Malicious Adversaries
Private Analysis of Graph Data

Graph $G$ $\rightarrow$ Trusted Curator $\rightarrow$ Users

Two conflicting goals:
- **Accurate** outputs
- **Data privacy**

Users:
- Researchers,
- Government,
- Businesses,
- and **Malicious Adversaries**
Private Analysis of Graph Data

Two conflicting goals:
• Accurate outputs
• Data privacy
(Central Model of) Differential Privacy

• **Neighboring** inputs differ in some information we’d like to hide

**Differential Privacy [Dwork-McSherry-Nissim-Smith ‘06]**

An algorithm $\mathcal{A}$ is $\varepsilon$-differentially private if for all pairs of neighbors $G$ and $G'$ and all sets of possible outputs $S$:

$$\Pr[\mathcal{A}(G) \in S] \leq e^\varepsilon \cdot \Pr[\mathcal{A}(G') \in S].$$
Edge-Neighboring Graphs

- **Edge-neighboring** graphs: differ in **one edge**
(Central Model of) Differential Privacy

- **Edge-neighboring** graphs: differ in **one edge**

Differential Privacy [Dwork-McSherry-Nissim-Smith ‘06]

An algorithm $\mathcal{A}$ is **$\epsilon$-differentially private** if for all pairs of neighbors $G$ and $G'$ and all sets of possible outputs $S$:

$$\Pr[\mathcal{A}(G) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(G') \in S].$$
(Central Model of) Differential Privacy

Graph $G$ → Trusted Curator → Users

Researchers, Government, Businesses, and Malicious Adversaries

Graph $G$

Users

https://www.npr.org/2021/04/09/986005820/after-data-breach-exposes-530-million-facebook-says-it-will-not-notify-users

https://www.bleepingcomputer.com/news/security/marriott-confirms-another-data-breach-after-hotel-got-hacked/

https://www.malwarebytes.com/blog/news/2021/06/second-colossal-linkedin-breach-in-3-months-almost-all-users-affected
(Central Model of) Differential Privacy

Graph $G$

Trusted Curator

Users

Researchers, Government, Businesses, and Malicious Adversaries

Queries

Answers

Unrealistic trust in trusted curator

https://www.npr.org/2021/04/09/986005820/after-data-breach-exposes-530-million-facebook-says-it-will-not-notify-users

https://www.bleepingcomputer.com/news/security/marriott-confirms-another-data-breach-after-hotel-got-hacked/

https://www.malwarebytes.com/blog/news/2021/06/second-colossal-linkedin-breach-in-3-months-almost-all-users-affected
Weaker Notion of Trust: **Local Model**

Graph $G$  
Untrusted Curator  
Users

- Researchers, Government, Businesses, and Malicious Adversaries

• Each node publishes privatized output
• Curator computes aggregated statistics using outputs
Weaker Notion of Trust: **Local Model**

- Each node publishes privatized output
- Curator computes aggregated statistics using outputs
Weaker Notion of Trust: **Local Model**

- Each node publishes privatized output
- Curator computes aggregated statistics using outputs
Weaker Notion of Trust: **Local Model**

- **Graph** $G$
- **Untrusted Curator**
- **Users**
  - Researchers, Government, Businesses, and Malicious Adversaries

**Strong notion of privacy:** Individuals trust **no one**!

- Each node publishes privatized output
- Curator computes aggregated statistics using outputs
Weaker Notion of Trust: **Local Model**

Graph $G$  Untrusted Curator  Users

- Each node publishes privatized output
- Curator computes aggregated statistics using outputs

**Strong notion of privacy:** Individuals trust **no one**!
Local Edge Differential Privacy (LED DP)

**Local Randomizer**
[Adapted from Kasiviswanathan-Lee-Nissim-Raskhodnikova-Smith ‘11]

An $\varepsilon$-local randomizer $\mathcal{R}$ is an $\varepsilon$-differentially private algorithm that takes as input an adjacency list $\mathbf{a}$ and public information.

An example adjacency list $\mathbf{a} = \{(B, C, E)\}$ is processed by the local randomizer $\mathcal{R}$.
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers

Untrusted Curator

Queries

Aggregate

Users

Researchers, Government, Businesses, and Malicious Adversaries

Round 1

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers

Untrusted Curator

Researchers, Government, Businesses, and Malicious Adversaries

Users

Queries

Aggregate

Round 1
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers

**Untrusted Curator**

- $\mathcal{R}_1(a_u, x)$
- $\mathcal{R}_1(a_v, x)$
- $\mathcal{R}_1(a_b, x)$
- $\mathcal{R}_1(a_d, x)$

**Users**

- Queries
- Aggregate

Researchers, Government, Businesses, and Malicious Adversaries

**Round 1:** $x$ is public info
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in rounds in distributed graph using local randomizers.

Untrusted Curator

Publish

Users

Queries

Aggregate

Researchers, Government, Businesses, and Malicious Adversaries

Round 2: $y$ is public info
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in rounds in distributed graph using local randomizers

Untrusted Curator

Users

Researchers, Government, Businesses, and Malicious Adversaries

Algorithm proceeds in rounds in distributed graph using local randomizers

Round 2: \( y \) is public info

\[
R_2(a_v, y) \\
R_2(a_w, y) \\
R_2(a_b, y) \\
R_2(a_d, y)
\]
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers.

Untrusted Curator

Users

Researchers, Government, Businesses, and Malicious Adversaries

Aggregate

Queries

Round 2

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers.
Local Edge Differential Privacy (LEDP)

Algorithm proceeds in **rounds** in **distributed graph** using local randomizers.

**Relevant Complexity Measure:**
**Number of Rounds of Communication**

Untrusted Curator

Researchers, Government, Businesses, and Malicious Adversaries

Users

Queries ➔ Aggregate ➔

Round 2
Local Edge Differential Privacy (LEDP)

**Local Edge Differential Privacy**
[Adapted from Kasiviswanathan-Lee-Nissim-Raskhodnikova-Smith '11]

Let algorithm $\mathcal{A}$ use (potentially different) local randomizers $\mathcal{R}_1^u, \ldots, \mathcal{R}_j^u$ and $\mathcal{R}_1^v, \ldots, \mathcal{R}_\ell^v$ on nodes $u, v$ with privacy parameters $\varepsilon_1^u, \ldots, \varepsilon_j^u$ and $\varepsilon_1^v, \ldots, \varepsilon_\ell^v$.

$\mathcal{A}$ is $\varepsilon$-local edge differentially private ($\varepsilon$-LEDP) if for every edge $\{u, v\}$,

$$\varepsilon_1^u + \cdots + \varepsilon_j^u + \varepsilon_1^v \cdots + \varepsilon_\ell^v \leq \varepsilon.$$
Related Work

• **Local edge differentially private algorithms:**
  • Relatively **new direction**
  • **Triangle and other subgraph counting:** [Imola-Murakami-Chaudhuri ‘21, ’22; Eden-Liu-Raskhodnikova-Smith ‘22]
  • **Other graph problems** in empirical settings in “decentralized” privacy models [Sun-Xiao-Khalil-Yang-Qin-Wang-Yu ‘19; Qin-Yu-Yang-Khalil-Xiao-Ren ‘17; Gao-Li-Chen-Zou ‘18; Ye-Hu-Au-Meng-Xiao ‘20]
Related Work

Central DP vs. LEDP
## Related Work

| Triangle Counting | Central DP vs. LEDP |
|-------------------|---------------------|
|                   | DP Upper Bound      | LEDP Lower Bound   |
|                   | $O \left( \frac{n}{\varepsilon} \right)$ additive error | $\Omega \left( \frac{n^{3/2}}{\varepsilon} \right)$ additive error |
|                   | (trivial)           | (multiple rounds)  |
|                   |                     | $\Omega \left( \frac{n^2}{\varepsilon} \right)$ additive error |
|                   |                     | (one round)        |
|                   |                     | [Eden-Liu-Raskhodnikova-Smith] |

[Eden-Liu-Raskhodnikova-Smith]
Related Work

**Natural Question:** Does there exist $\varepsilon$-LEDP algorithms where *multiplicative error matches best distributed algorithm* and with $\frac{\text{polylog}(n)}{\varepsilon}$ additive error?
Related Work

Natural Question: Does there exist $\varepsilon$-LEDP algorithms where multiplicative error matches best distributed algorithm and with $\frac{\text{polylog}(n)}{\varepsilon}$ additive error?

Yes!
## Our contributions and previous work

| LEDP | Our Results | Best Previous Non-Private Results | Best Previous Private Results |
|------|-------------|-----------------------------------|-------------------------------|
|      | All \(\text{polylog}(n)/\varepsilon\) additive | \((2 + \eta)\)-mult. \(O(\log n)\) rounds |NONE|
|      | \(k\)-core decomposition: \((2 + \eta)\)-mult. \(O(\log n)\) rounds | [Chan-Sozio-Sun ‘21]|
|      | Low out-degree ordering: Same as above | | |
Our contributions and previous work

| LEDP | Our Results | Best Previous Non-Private Results | Best Previous Private Results |
|------|-------------|----------------------------------|------------------------------|
|      | All $\text{polylog}(n)/\varepsilon$ additive $k$-core decomposition: $\left(2 + \eta\right)$-mult. $O(\log n)$ rounds | $\left(2 + \eta\right)$-mult. $O(\log n)$ rounds [Chan-Sozio-Sun ‘21] | NONE |
|      | Low out-degree ordering: Same as above | | NONE |
|      | Densest subgraph: $\left(4 + \eta\right)$-mult. | $\left(1 + \eta\right)$-multiplicative [Bahmani-Goel-Munagala ‘14] [Ghaffari-Lattanzi-Mitrović ‘19] [Su-Vu ‘20] | NONE |
## Our contributions and previous work

|                | Our Results                                      | Best Previous Non-Private Results                        | Best Previous Private Results |
|----------------|--------------------------------------------------|----------------------------------------------------------|-------------------------------|
| **LEDP**       | All $\text{polylog}(n)/\epsilon$ additive       | $(2 + \eta)$-mult.                                        | NONE                          |
|                | $k$-core decomposition:                          | $O(\log n)$ rounds                                       |                               |
|                | $(2 + \eta)$-mult.                               | [Chan-Sozio-Sun '21]                                      |                               |
|                | $O(\log n)$ rounds                              |                                                          |                               |
|                | Low out-degree ordering:                         |                                                          |                               |
|                | Same as above                                    |                                                          |                               |
| **Densest subgraph:** | $\text{polylog}(n)/\epsilon$ additive | $(1 + \eta)$-multiplicative                              | NONE                          |
|                | $k$-core decomposition:                          |                                                          |                               |
|                | $(4 + \eta)$-mult.                               | [Bahmani-Goel-Munagala ‘14]                              |                               |
|                | $O(\log n)$ rounds                              | [Ghaffari-Lattanzi-Mitrović ‘19]                         |                               |
|                | Low out-degree ordering:                         | [Su-Vu ‘20]                                              |                               |
|                | Same as above                                    |                                                          |                               |

| **DP**         | Densest subgraph:                                | $\text{polylog}(n)/\epsilon$-additive                   |                               |
|                | $\text{(1 + \eta)}$-multiplicative               |                                                          |                               |
|                | $(1 + \eta)$-multiplicative                      | [Bahmani-Goel-Munagala ‘14]                              |                               |
|                | $\text{polylog}(n)/\epsilon$-additive           | [Chekuri-Quanrud-Torres ‘22]                             |                               |
|                | Densest subgraph:                                |                                                          |                               |
|                | $(1 + \eta)$-mult.                               | [Nguyen-Vullikanti ‘21]                                  |                               |
|                | $\text{(2 + \eta)}$-mult.                        | [Farhadi-Hajiaghayi-Shi ‘22]                             |                               |
### Our contributions and previous work

| LEDP | Our Results | Best Previous Non-Private Results | Best Previous Private Results |
|------|-------------|-----------------------------------|-------------------------------|
|      | All $\text{polylog}(n) / \varepsilon$ additive $k$-core decomposition: $(2 + \eta)$-mult. $O(\log n)$ rounds | $(2 + \eta)$-mult. $O(\log n)$ rounds [Chan-Sozio-Sun ‘21] | NONE |
|      | Low out-degree ordering: Same as above | | |
|      | Densest subgraph: $(4 + \eta)$-mult. | | |

| DP   | Densest subgraph: $(1 + \eta)$-mult. | $(1 + \eta)$-multiplicative $\text{poly}(\log n) / \varepsilon$-additive [Bahmani-Goel-Munagala ‘14] [Chekuri-Quanrud-Torres ‘22] | $(2 + \eta)$-mult. $\text{poly}(\log n) / \varepsilon$-additive [Nguyen-Vullikanti ‘21] [Farhadi-Hajiaghayi-Shi ‘22] |

**Privacy Framework**

**OPEN:** Framework Approximation Guarantee

- [Ghaffari-Lattanzi-Mitrović ‘19]
- [Su-Vu ‘20]
### Our contributions and previous work

| LEDP | **Our Results** | **Best Previous Non-Private Results** | **Best Previous Private Results** |
|------|----------------|--------------------------------------|----------------------------------|
|      | All $\text{polylog}(n)/\varepsilon$ additive | $(2 + \eta)$-mult. $O(\log n)$ rounds | NONE |
|      | $k$-core decomposition: $(2 + \eta)$-mult. $O(\log n)$ rounds | [Chan-Sozio-Sun ‘21] |
|      | Low out-degree ordering: Same as above | | |
|      | Densest subgraph: $(4 + \eta)$-mult. | [Ghaffari-Lattanzi-Mitrovic ‘19] [Su-Vu ‘20] | |
|      | | | |
| **DP** | Densest subgraph: $(1 + \eta)$-mult. | $(1 + \eta)$-multiplicative $\text{poly}(\log n)/\varepsilon$-additive | (2 + $\eta$)-mult. |
|      | | [Bahmani-Goel-Munagala ‘14] [Chekuri-Quanrud-Torres ‘22] | [Nguyen-Vullikanti ‘21] [Farhadi-Hajiaghayi-Shi ‘22] |

**Privacy Framework**

- OPEN: Framework Approximation Guarantee
- NONE
$k$-Core

3-Core
$k$-Core Decomposition

**Core Number** of Node $v$: Maximum Core Value of a Core Containing $v$
Core Number of Node $v$: Maximum Core Value of a Core Containing $v$
Approximate $k$-Core Decomposition

Approx. Core Number : 2

Approx. core number of every node: 3

$(c, d)$-Approx. Core Number:
$\text{core}(v) - d \leq \overline{\text{core}}(v) \leq c \cdot \text{core}(v) + d$
Approximate $k$-Core Decomposition

Approx. core number of every node: 3

$(3/2, 0)$-approx

$(2, 0)$-approx

Approx. Core Number: 2

$(c, d)$-Approx. Core Number:
\[
\text{core}(v) - d \leq \overline{\text{core}}(v) \leq c \cdot \text{core}(v) + d
\]
Approximate $k$-Core Decomposition

Approx. core number of every node: 3

Approx. Core Number: 2

(2, 0)-approx

(3/2, 0)-approx

$\left(2 + \eta, O\left(\frac{\log^3(n)}{\varepsilon}\right)\right)$-approximations in this paper

$(c, d)$-Approx. Core Number:
core$(v) - d \leq \overline{\text{core}(v)} \leq c \cdot \text{core}(v) + d$
Level Data Structure and Core Numbers

Non-private sequential and parallel level data structures for dynamic problem:

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15;
Henzinger-Neumann-Wiese ‘20;
Liu-Shi-Yu-Dhulipala-Shun ‘22]
Level Data Structure and Core Numbers

In this example: \( \eta = 0.1 \)

\[ 4 \log_{1+\eta}(n) \]

Move up if induced degree in active vertices > \((1 + \eta)\)

Initially all vertices are active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
In this example: \( \eta = 0.1 \)

\[ 4\log_{1+\eta}(n) \]

Move up if induced degree in **active** vertices > \((1 + \eta)\)

Initially all vertices are active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
In this example: $\eta = 0.1$

$4\log_{1+\eta}(n)$

Move up if induced degree in active vertices > $(1 + \eta)$

Initially all vertices are active

Vertices which moved remain active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis '15, Henzinger-Neumann-Wiese '20, Liu-Shi-Yu-Dhulipala-Shun '22]
In this example: \( \eta = 0.1 \)

\( 4 \log_{1+\eta}(n) \)

Move up if induced degree in active vertices > \((1 + \eta)\)

Initially all vertices are active

Vertices which moved remain active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
In this example: $\eta = 0.1$

$4\log_{1+\eta}(n)$

Move up if induced degree in active vertices $> (1 + \eta)$

Initially all vertices are active

Vertices which moved remain active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
In this example: \( \eta = 0.1 \)

\[ 4 \log_{1+\eta}(n) \]

Move up if induced degree in active vertices > \((1 + \eta)\)

Initially all vertices are active

Vertices which moved remain active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
In this example: $\eta = 0.1$

$4\log_{1+\eta}(n)$

Move up if induced degree in active vertices $> (1 + \eta)$

Initially all vertices are active

Vertices which moved remain active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
Non-Private Core Number Approximation

\( \eta = 0.1 \)

Set cutoffs \((1 + \eta)^i\) for all \(i \in [\log_{1+\eta}(n)]\)

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]
Non-Private Core Number Approximation

\[ \eta = 0.1 \]

Set cutoffs \((1 + \eta)^i\) for all \(i \in [\log_{1+\eta}(n)]\)

Give approx core number \(2 \cdot (1 + \eta)^i\) using largest cutoff where node is on the topmost level
Non-Private Core Number Approximation

\[ \eta = 0.1 \]

Set cutoffs \((1 + \eta)^i\) for all \(i \in \left[ \log_{1+\eta}(n) \right] \)

Approximation: \(2 \cdot (1 + \eta)^7 = 2 \cdot 1.1^7 = 2 \cdot 1.95 = 3.9\)
Non-Private Core Number Approximation

\[ \eta = 0.1 \]

Set cutoffs \((1 + \eta)^i\) for all \(i \in [\log_{1+\eta}(n)]\)

Top level means adjacent to many neighbors of sufficiently high degree

Largest cutoff gives largest such degree

Approximation: \(2 \cdot (1 + \eta)^7 = 2 \cdot 1.1^7 \approx 2 \cdot 1.95 = 3.9\)
Non-Private Core Number Approximation

\[ \eta = 0.1 \]

Set cutoffs \((1 + \eta)^i\) for all \(i \in [\log_{1+\eta}(n)]\)

Cutoff: \((1 + \eta)\)

... 

Top level means adjacent to many neighbors of sufficiently high degree

Cutoff: \((1 + \eta)^7\)

Largest cutoff gives largest such degree

Approximation: 1
$\varepsilon$-LEDLP Core Numbers

Each **active vertex** draws i.i.d. **noise** from symmetric **geometric distribution**

Distribution $\textbf{Geom}(b)$

PMF: $\frac{e^b - 1}{e^b + 1} \cdot e^{-|x| \cdot b}$

Release and move up degree $+ \textbf{noise}$ in **active** vertices $> (1 + \eta)$
\( \varepsilon \)-LEDP Core Numbers

Each active vertex draws i.i.d. noise from symmetric geometric distribution

Distribution \( \text{Geom}(b) \)
PMF: \( \frac{e^b - 1}{e^b + 1} \cdot e^{-|x| \cdot b} \)

Release and move up degree + noise in active vertices > (1 + \eta)
\( \varepsilon \)-LEDG Core Numbers

Each active vertex draws i.i.d. noise from symmetric geometric distribution

Distribution \( \text{Geom}(b) \)

PMF: \( \frac{e^b - 1}{e^b + 1} \cdot e^{-|x| \cdot b} \)

If \( \text{deg}(i) + N_i > (1 + \eta) \), move up

Where \( N_i \sim \text{Geom}\left(\frac{\varepsilon}{8\log_2^{1+\eta}(n)}\right) \)

Release and move up degree + noise in active vertices > (1 + \( \eta \))

In this example: \( \eta = 0.1 \)

\[ 1 + 1 \]

\( i \) \( j \) \( k \) \( a \) \( b \) \( c \) \( d \)
\( \varepsilon \)-LEDPA Core Numbers

Each active vertex draws i.i.d. noise from symmetric geometric distribution

Distribution \( Geom(b) \)
PMF: \( \frac{e^b - 1}{e^b + 1} \cdot e^{-|X| \cdot b} \)

If \( \text{deg}(i) + N_i > (1 + \eta) \), move up

Where \( N_i \sim Geom\left(\frac{\varepsilon}{8\log_{1+\eta}(n)}\right) \)

Release and move up degree + noise in active vertices > \((1 + \eta)\)
**ε-LEDP Core Numbers**

Each active vertex draws i.i.d. noise from symmetric geometric distribution

Distribution $Geom(b)$

PMF: $\frac{e^b - 1}{e^b + 1} \cdot e^{-|X| \cdot b}$

If $\text{deg}(i) + N_i > (1 + \eta)$, move up

Where $N_i \sim Geom\left(\frac{\varepsilon}{8\log_2(1+\eta(n))}\right)$

Release and move up degree + noise in active vertices $> (1 + \eta)$

Redraw new noise each time vertex remains active

If $\text{deg}(i) + N_i > (1 + \eta)$, move up
**ε-LEDП Core Numbers**

Each active vertex draws i.i.d. noise from symmetric geometric distribution.

Distribution $\text{Geom}(b)$

PMF: $\frac{e^{b-1}}{e^b + 1} \cdot e^{-|x| \cdot b}$

If $\text{deg}(i) + N_i > (1 + \eta)$, move up

Where $N_i \sim \text{Geom}\left(\frac{\varepsilon}{8\log_{1+\eta}(n)}\right)$

- Release and move up degree + noise in active vertices $> (1 + \eta)$
- Redraw new noise each time vertex remains active
- Approx. as before $2(1 + \eta)^i$ using topmost level
\( \varepsilon \)-LEDG Core Numbers

Each active vertex draws i.i.d. noise from symmetric geometric distribution.

Distribution \( \text{Geom}(b) \)

PMF: \( \frac{e^b - 1}{e^b + 1} \cdot e^{-|X| \cdot b} \)

If \( \text{deg}(k) + N_k > (1 + \eta) \), move up

Where \( N_k \sim \text{Geom} \left( \frac{\varepsilon}{8\log_2^2 n + \eta(n)} \right) \)

Release and move up degree + noise in active vertices > (1 + \eta)

Redraw new noise each time vertex remains active

Approx. as before \( 2(1 + \eta)^i \) using topmost level
\(\varepsilon\)-LEDП Core Numbers

Each active vertex draws i.i.d. noise from symmetric geometric distribution.

Distribution \(\text{Geom}(p)\) with PMF:

\[
\frac{e^b - 1}{e^b + 1} \cdot e^{-|X|b}
\]

If \(\text{deg}(k) + N_k > (1 + \eta)\), move up.

Where \(N_k \sim \text{Geom}\left(\frac{\varepsilon}{8\log_{1+\eta}(n)}\right)\)

Move up if induced degree + noise in active vertices > \((1 + \eta)\).

Redraw new noise each time vertex remains active and determines whether move up.

Approx. as before \(2(1 + \eta)^i\) where \(i\) largest that vertex is on the topmost level.

Privacy and Approximation?
Privacy Proof

• Can be implemented via local randomizers $R$
Privacy Proof

• Can be implemented via local randomizers $R$
• $R$ takes as input a (adjacency list) and public set of active vertices $A$
Privacy Proof

• Can be implemented via local randomizers $R$
• $R$ takes as input $a$ (adjacency list) and public set of active vertices $A$
  • $R$ computes size of intersection $|a \cap A|$
Privacy Proof

• Can be implemented via local randomizers $R$

• $R$ takes as input $a$ (adjacency list) and public set of active vertices $A$
  
  • $R$ computes size of intersection $|a \cap A|$

• Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log^2(1+\eta(n))}\right)$
Privacy Proof

- Can be implemented via **local randomizers** $R$
- $R$ takes as **input** $a$ (adjacency list) and **public set of active vertices** $A$
  - $R$ computes size of intersection $|a \cap A|$ **Sensitivity of 1**

- Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log^2(1+\eta(n))}\right)$

**Global Sensitivity:**

$$\Delta f = \max_{\text{edge-neighbors } G \text{ and } G'} |f(G) - f(G')|$$

$$f(a, A) = |a \cap A|$$
Privacy Proof

• Can be implemented via local randomizers $R$
• $R$ takes as input $a$ (adjacency list) and public set of active vertices $A$
  • $R$ computes size of intersection $|a \cap A|$ Sensitivity of 1
  • Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log^2{n} + \eta(n)}\right)$

**Geometric Mechanism:**
[Chan-Shi-Song ‘11; Balcer-Vadhan ‘18]

$$M(a, A) = f(a, A) + Geom\left(\frac{\varepsilon}{\Delta f}\right)$$

$M$ is $\varepsilon$-DP
Privacy Proof

• Can be implemented via local randomizers $R$
• $R$ takes as input $a$ (adjacency list) and public set of active vertices $A$
  • $R$ computes size of intersection $|a \cap A|$ Sensitivity of 1

• Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log_{1+\eta}(n)}\right)$
• $R$ is $\frac{\varepsilon}{8\log_{1+\eta}(n)}$ - LR by privacy of Geometric Mechanism [Chan-Shi-Song ‘11; Balcer-Vadhan ‘18]
Privacy Proof

• Can be implemented via local randomizers $R$

• $R$ takes as input $a$ (adjacency list) and public set of active vertices $A$
  
  - $R$ computes size of intersection $|a \cap A|$ **Sensitivity of 1**

  - Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log^2_{1+\eta}(n)}\right)$

• $R$ is $\frac{\varepsilon}{8\log^2_{1+\eta}(n)}$-LR by privacy of Geometric Mechanism [Chan-Shi-Song ‘11; Balcer-Vadhan ‘18]

• Same LR called for all vertices $4\log^2_{1+\eta}(n)$ times
Privacy Proof

• Can be implemented via local randomizers $R$
• $R$ takes as input $a$ (adjacency list) and public set of active vertices $A$
  - $R$ computes size of intersection $|a \cap A|$ Sensitivity of 1
• Then, add symmetric geometric noise $X \sim \text{Geom} \left( \frac{\varepsilon}{8 \log_{1+\eta}(n)} \right)$
• $R$ is $\frac{\varepsilon}{8 \log_{1+\eta}(n)}$ - LR by privacy of Geometric Mechanism [Chan-Shi-Song ‘11; Balcer-Vadhan ‘18]
• Same LR called for all vertices $4 \log_{1+\eta}(n)$ times
• For each edge, called $8 \log_{1+\eta}(n)$; then, $8 \log_{1+\eta}(n) \cdot \frac{\varepsilon}{8 \log_{1+\eta}(n)} = \varepsilon$ and so $\varepsilon$-LEDP
Approximation Proof Sketch

• With high probability, magnitude of each drawn noise is upper bounded by $O\left(\frac{\log^3 n}{\varepsilon}\right)$
Approximation Proof Sketch

• With high probability, magnitude of each drawn noise is upper bounded by $O\left(\frac{\log^3 n}{\varepsilon}\right)$

• **Degree Upper Bound:** If a vertex $v$ is on level $i < 4\log_{1+\eta}(n)$ at end of algorithm, then it has at most $(1 + \eta)^i + O\left(\frac{\log^3 n}{\varepsilon}\right)$ neighbors on levels $\geq i$. 
Approximation Proof Sketch

• With high probability, magnitude of each drawn noise is upper bounded by $O\left(\frac{\log^3 n}{\varepsilon}\right)$

• Degree Upper Bound: If a vertex $v$ is on level $i < 4\log_{1+\eta}(n)$ at end of algorithm, then it has at most $(1 + \eta)^i + O\left(\frac{\log^3 n}{\varepsilon}\right)$ neighbors on levels $\geq i$
Approximation Proof Sketch

• With high probability, magnitude of each drawn noise is upper bounded by \( O \left( \frac{\log^3 n}{\varepsilon} \right) \)

• **Degree Upper Bound:** If a vertex \( v \) is on level \( i < 4 \log_{1+\eta}(n) \) at end of algorithm, then it has at most \((1 + \eta)^i + O \left( \frac{\log^3 n}{\varepsilon} \right)\) neighbors on levels \( \geq i \)

• **Degree Lower Bound:** If a vertex \( v \) is on level \( i > 0 \) at end of algorithm, then it has at least \( (1 + \eta)^i - O \left( \frac{\log^3 n}{\varepsilon} \right) \) neighbors on levels \( \geq i - 1 \)
Approximation Proof Sketch

• With high probability, magnitude of each drawn noise is upper bounded by $O\left(\frac{\log^3 n}{\varepsilon}\right)$

• **Degree Upper Bound:** If a vertex $v$ is on level $i < 4\log_{1+\eta}(n)$ at end of algorithm, then it has at most $(1 + \eta)^i + O\left(\frac{\log^3 n}{\varepsilon}\right)$ neighbors on levels $\geq i$

• **Degree Lower Bound:** If a vertex $v$ is on level $i > 0$ at end of algorithm, then it has at least $(1 + \eta)^i - O\left(\frac{\log^3 n}{\varepsilon}\right)$ neighbors on levels $\geq i - 1$
Approximation Proof Sketch

**Key:** Largest cutoff increases/decreases by additive $O\left(\frac{\log^3 n}{\varepsilon}\right)$
Locally Adjustable Graph Algorithms

• **Intuition:** Each node’s current state depends on **number of neighbors** whose **previous state satisfies** predicate $P$
Locally Adjustable Graph Algorithms

- **Intuition:** Each node’s current state depends on the number of neighbors whose previous state satisfies predicate $P$.

\[
\text{Count}(P(S_0''), P(S_0'))
\]

Diagram:
- $S_0''$ to $S_0$
- $P(S_0'')$ to $P(S_0')$
- $P(S_0')$ to $S_0'$

 FOCS 2022
Locally Adjustable Graph Algorithms

• **Intuition:** Each node’s current state depends on **number of neighbors** whose **previous state satisfies predicate** $P$
  • **Update new state** based on this count

Sensitivity of Count of number of neighbors satisfying $P$ is 1
Locally Adjustable Graph Algorithms

**Intuition:** Each node’s current state depends on **number of neighbors** whose *previous state satisfies predicate* $P$

**Update new state** based on this count

\[ \text{Count}(P(S_0''), P(S_0')) \]

Then, **just add geometric noise** to the counts!

Sample from $\text{Geom}\left(\frac{\varepsilon}{2 \cdot \text{rounds}}\right)$ where *rounds* is # rounds of algorithm.

Sensitivity of *Count* of number of neighbors satisfying $P$ is 1
Many distributed/parallel graph algorithms use small rounds/depth and may fall under framework.

Sensitivity of \text{Count} of number of neighbors satisfying $P$ is $1$.

Then, just add geometric noise to the counts!

Sample from $\text{Geom} \left( \frac{\varepsilon}{2 \cdot \text{rounds}} \right)$ where \textit{rounds} is # rounds of algorithm.
Many distributed/parallel graph algorithms use small rounds/depth and may fall under framework

Use small rounds/depth to get small noise

OPEN: approximation bounds for framework
Additional Open Questions

- Better multiplicative approximation for LEDP densest subgraph (currently $4 + \eta$ for LEDP compared to $1 + \eta$ for DP)
Additional Open Questions

• Better **multiplicative approximation** for **LEDP densest subgraph** (currently $4 + \eta$ for LEDP compared to $1 + \eta$ for DP)

• **Better upper bounds or tight lower bounds** for the additive noise (current best lower bound of [Farhadi-Hajiaghayi-Shi ’22] is sub-logarithmic)
Additional Open Questions

• Better **multiplicative approximation** for LEDP densest subgraph (currently $4 + \eta$ for LEDP compared to $1 + \eta$ for DP)

• **Better upper bounds** or **tight lower bounds** for the additive noise (current best lower bound of [Farhadi-Hajiaghayi-Shi ’22] is sub-logarithmic)

• **Node privacy** for $k$-core decomposition (deleting one node changes the core number of any node by at most 1)

**Node-neighboring** graphs: differ in **one node and adjacent edges**