Differential Privacy from Locally Adjustable Graph Algorithms:

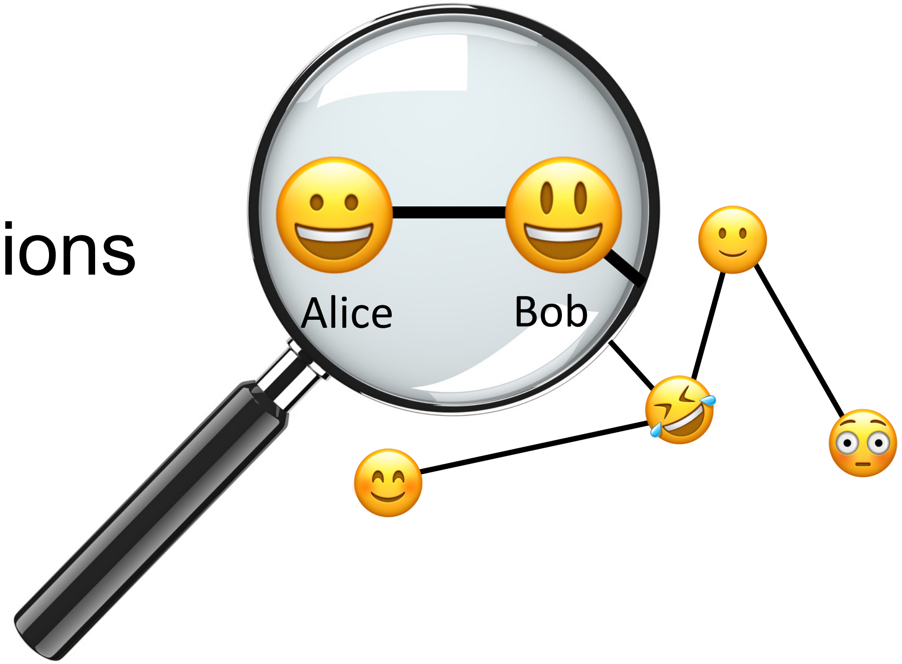
k-Core Decomposition, Low Out-Degree Ordering, and Densest Subgraphs

Laxman Dhulipala, **Quanquan C. Liu**, Sofya Raskhodnikova, Jessica Shi, Julian Shun, Shangdi Yu

Northwestern
MIT CSAIL
BOSTON UNIVERSITY
UNIVERSITY OF MARYLAND

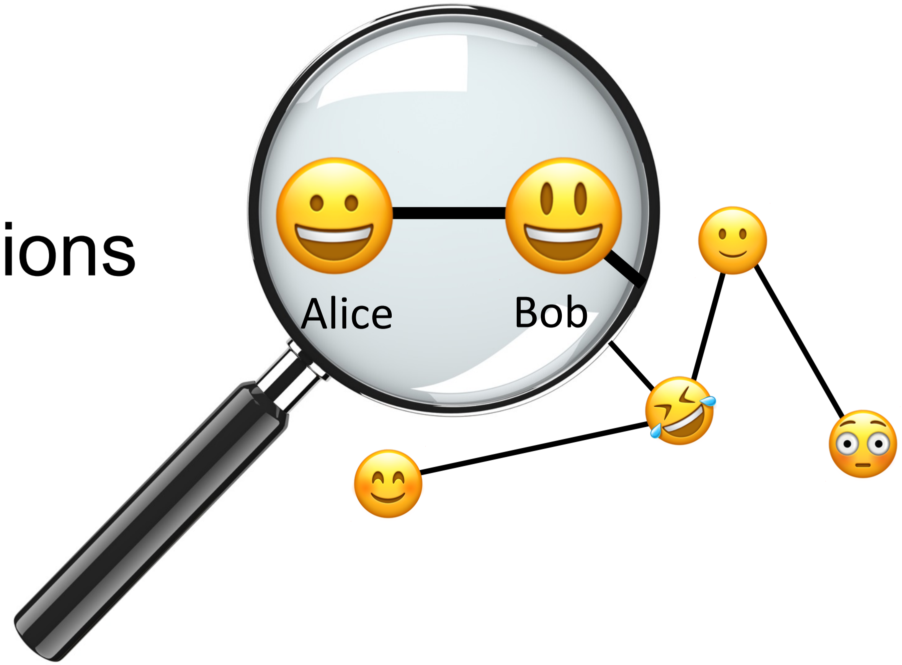
Publishing Sensitive Graph Information

- 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

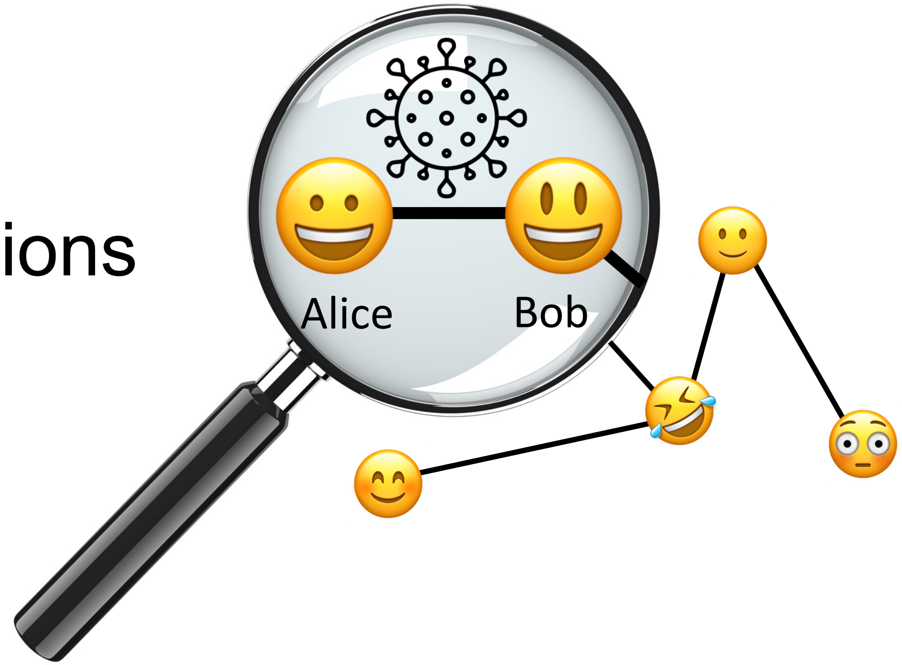
- Potentially **sensitive connections between individuals** published as graphs
  - Financial transactions
  - Relationship information
  - Email and cell phone communications
  - Search data
  - Disease network data



Anonymization ≠ Privacy!

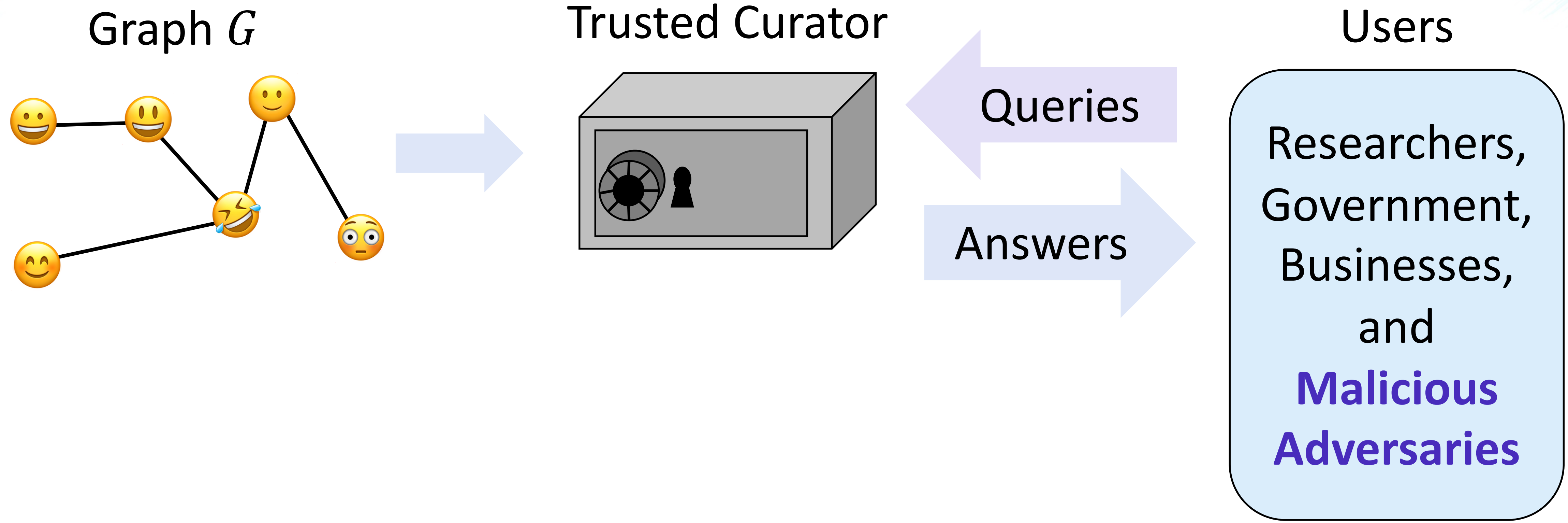
Publishing Sensitive Graph Information

- 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

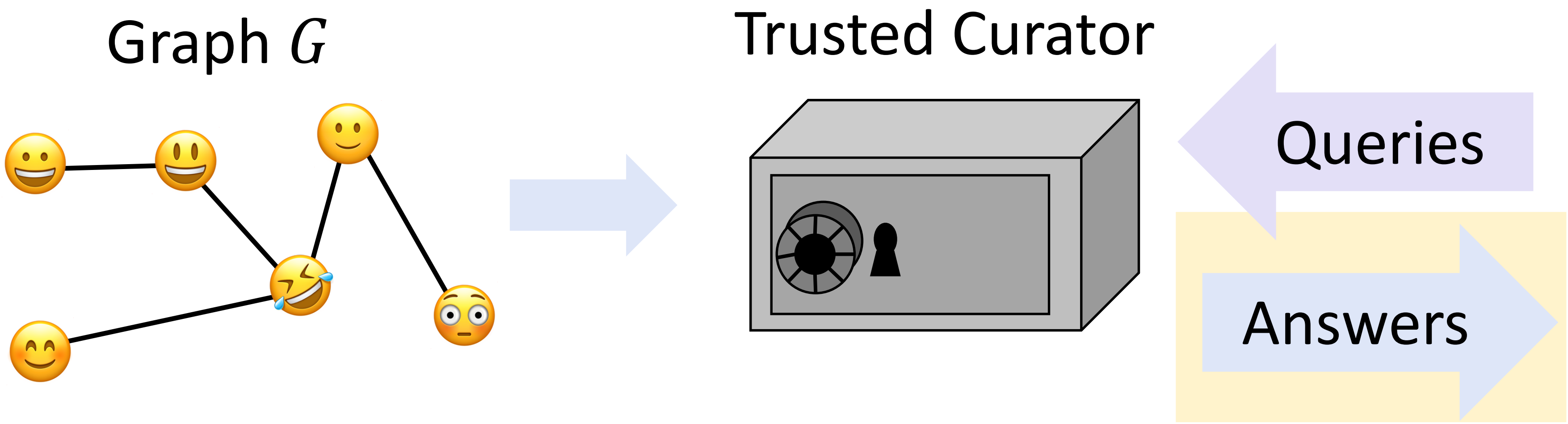


Anonymization ≠ Privacy!

Private Analysis of Graph Data



Private Analysis of Graph Data

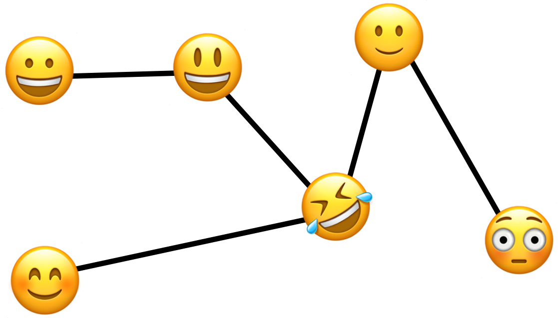


Two **conflicting** goals:

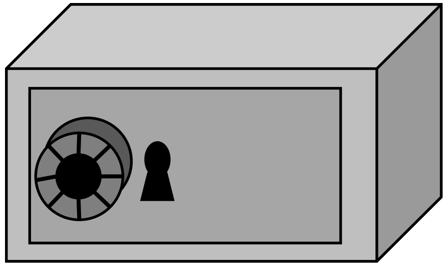
- **Accurate** outputs
- **Data privacy**

Private Analysis of Graph Data

Graph G



Trusted Curator



Queries

Answers

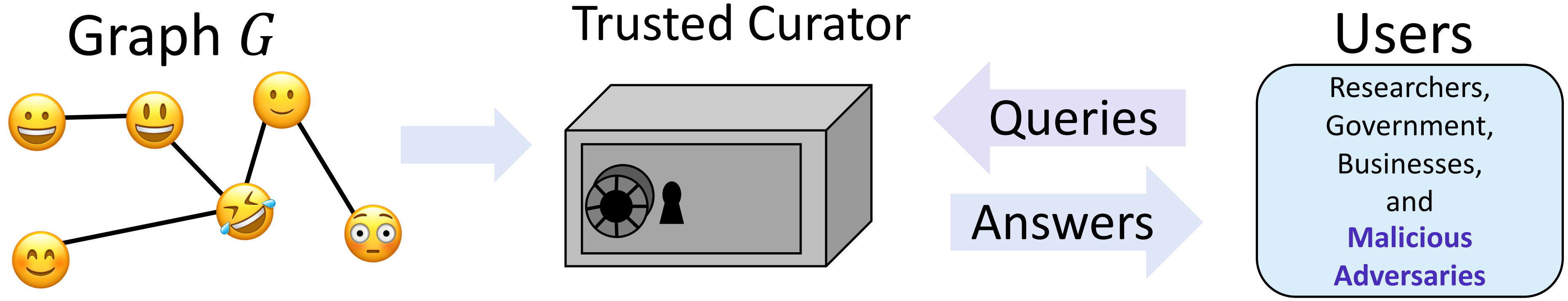
Users

Researchers, Government, Businesses, and **Malicious Adversaries**

Two **conflicting** goals:

- **Accurate** outputs
- **Data privacy**

(Central Model of) Differential Privacy



- **Neighboring** inputs differ in some information we'd like to hide

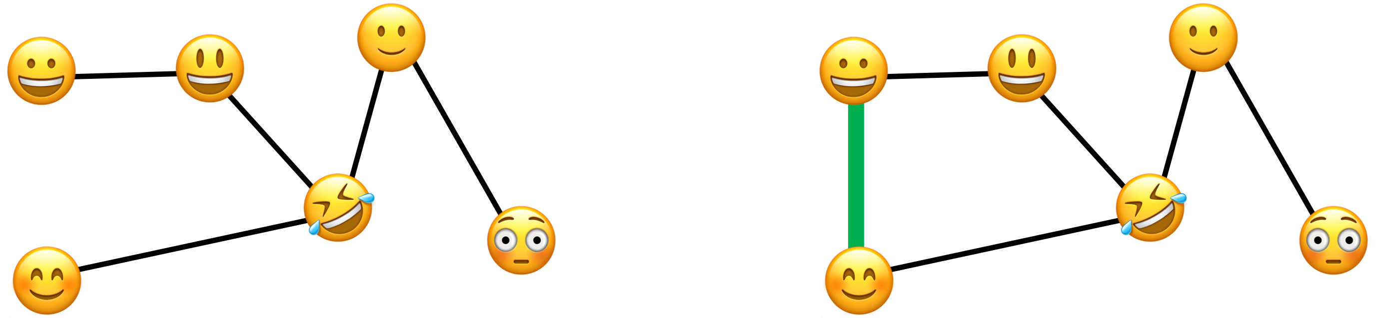
<u>Differential Privacy [Dwork-McSherry-Nissim-Smith '06]</u>

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

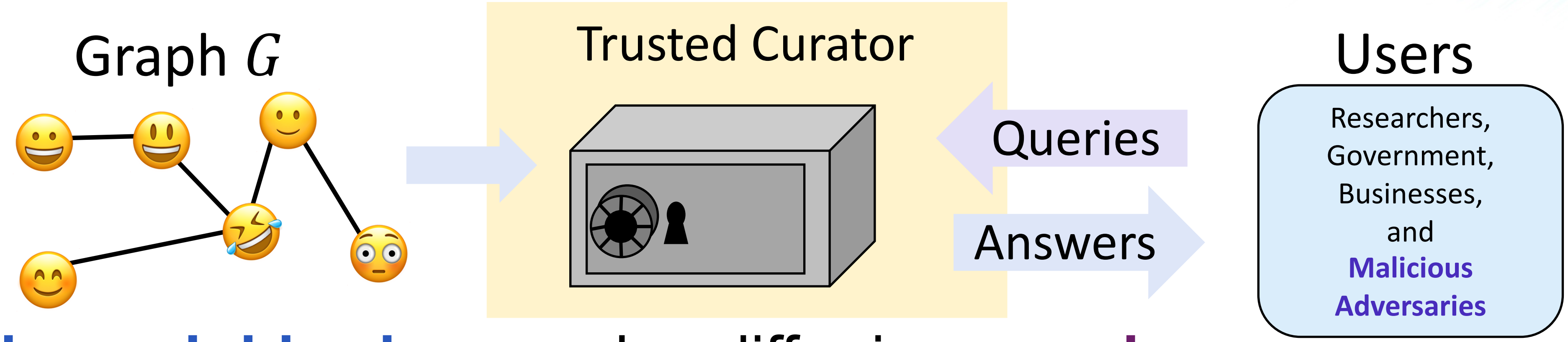
$$\Pr[\mathcal{A}(G) \in S] \leq e^{\varepsilon} \cdot \Pr[\mathcal{A}(G') \in S].$$

Edge-Neighboring Graphs

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



(Central Model of) Differential Privacy



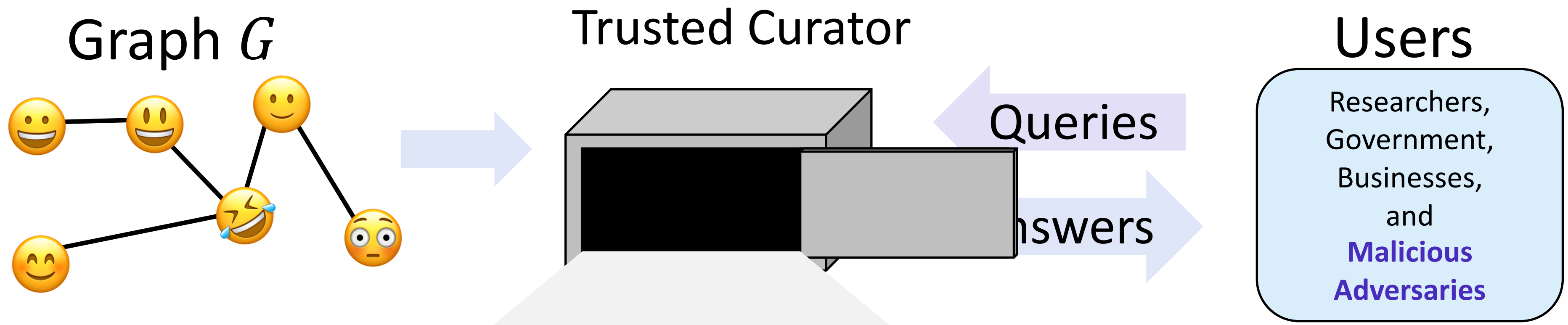
- **Edge-neighboring** graphs: differ in **one edge**

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

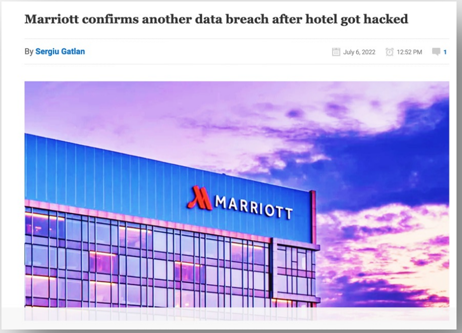
An algorithm $\mathcal{A}$ is **ε-differentially private** if for all pairs of neighbors G and G' and all sets of possible outputs S:

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

(Central Model of) Differential Privacy



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

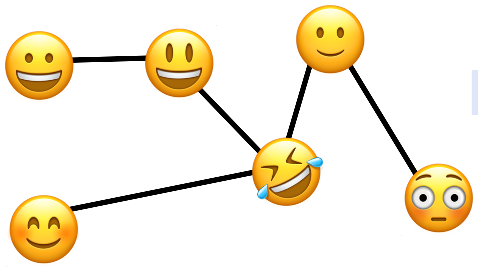


https://www.bleepingcomputer.com/news/security/marriott-confirms-another-data-breach-after-hotel-got-hacked/

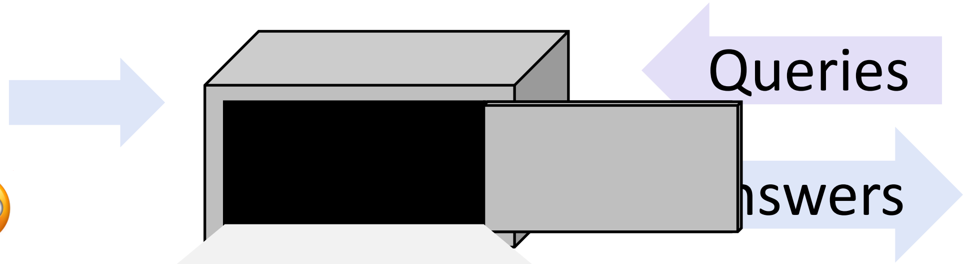
https://www.malwarebytes.com/blog/news/2021/06/second-colossal-linkedin-breach-in-3-months-almost-all-users-affected

(Central Model of) Differential Privacy

Graph G



Trusted Curator



Users

Researchers,
Government,
Businesses,
and
Malicious Adversaries

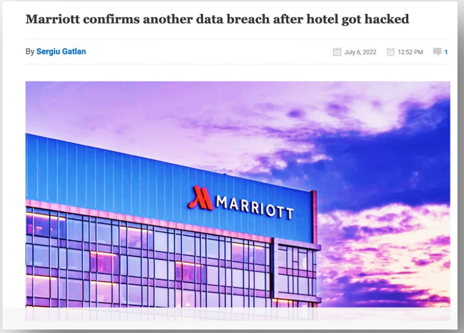

TECHNOLOGY

After Data Breach Exposes 530 Million, Facebook Says It Will Not Notify Users

April 9, 2021 · 11:58 PM ET

EMMA BOWMAN

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


Marriott confirms another data breach after hotel got hacked

By Sergiu Gatlan

July 6, 2022 12:52 PM 1

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

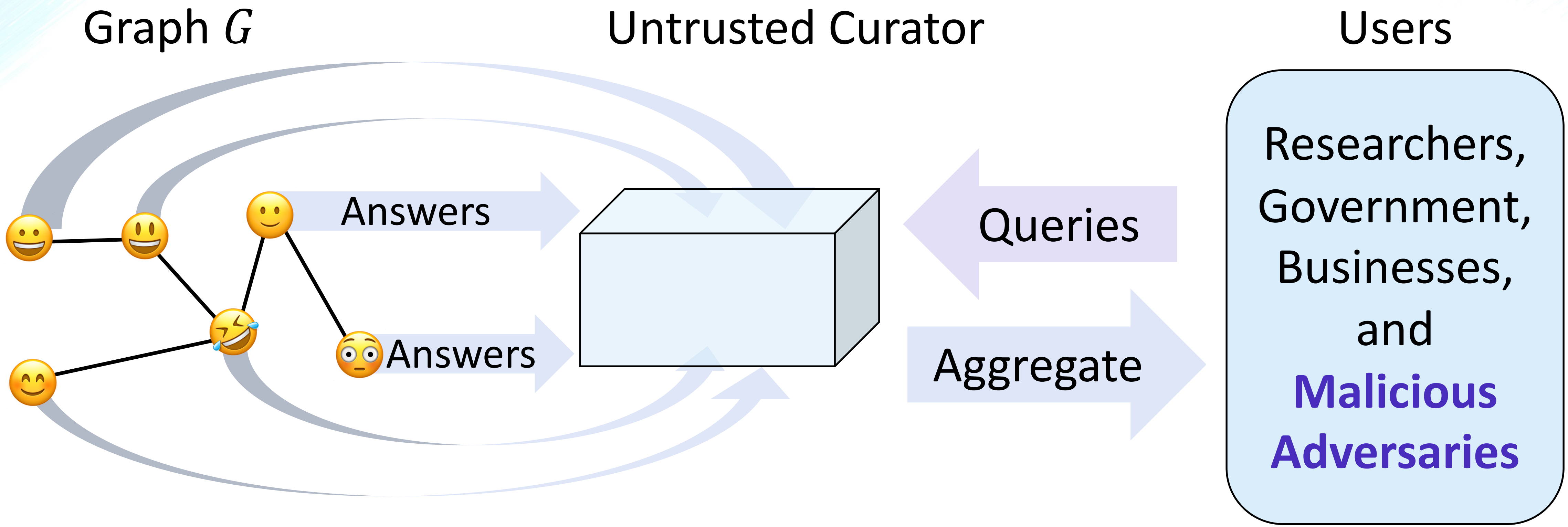

AWARENESS | NEWS

Second colossal LinkedIn "breach" in 3 months, almost all users affected

https://www.malwarebytes.com/blog/news/2021/06/second-colossal-linkedin-breach-in-3-months-almost-all-users-affected

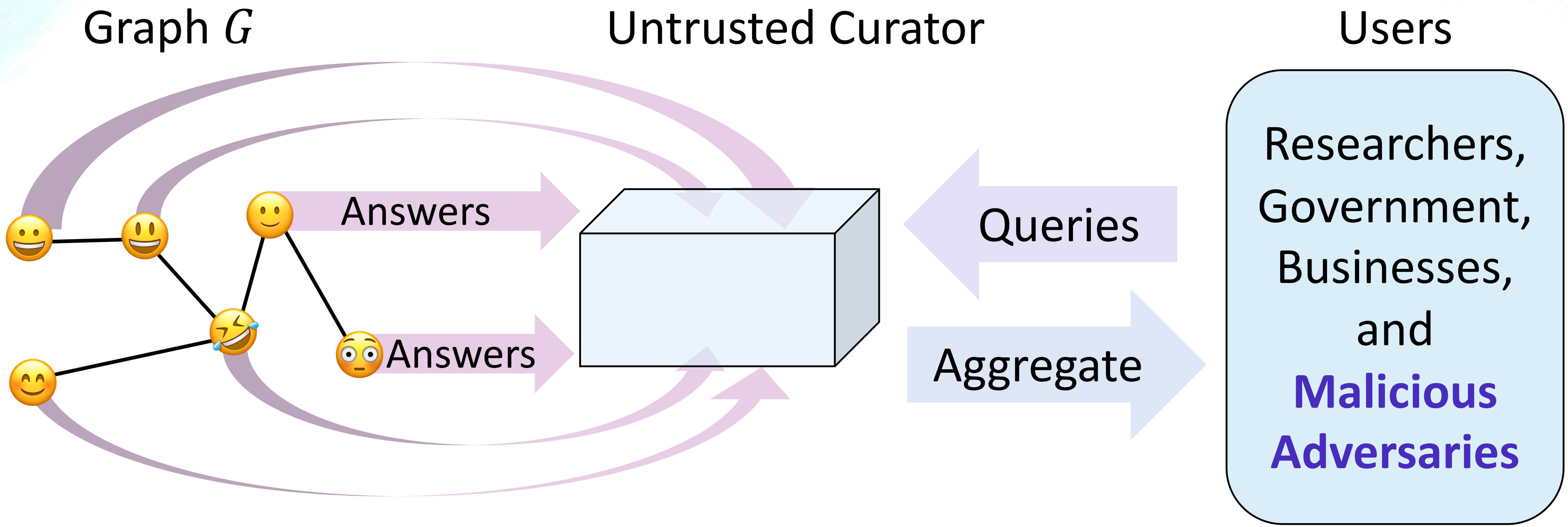
Unrealistic trust in **trusted curator**

Weaker Notion of Trust: **Local Model**



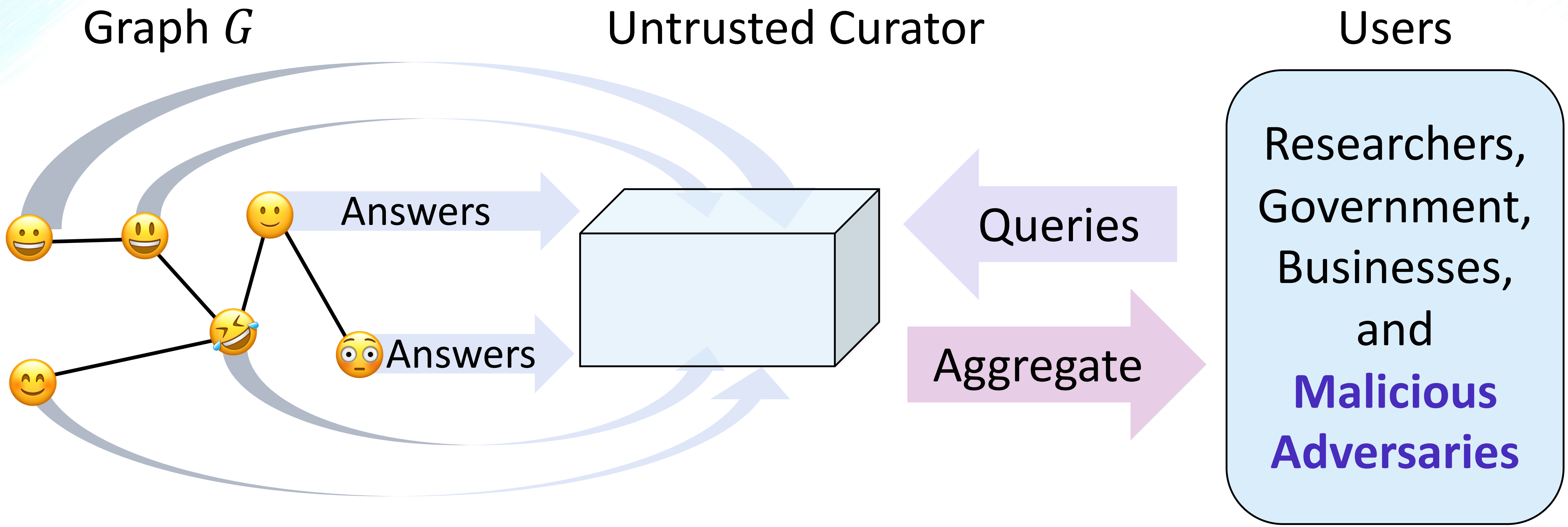
- **Each node publishes privatized** output
- Curator computes aggregated statistics using outputs

Weaker Notion of Trust: **Local Model**



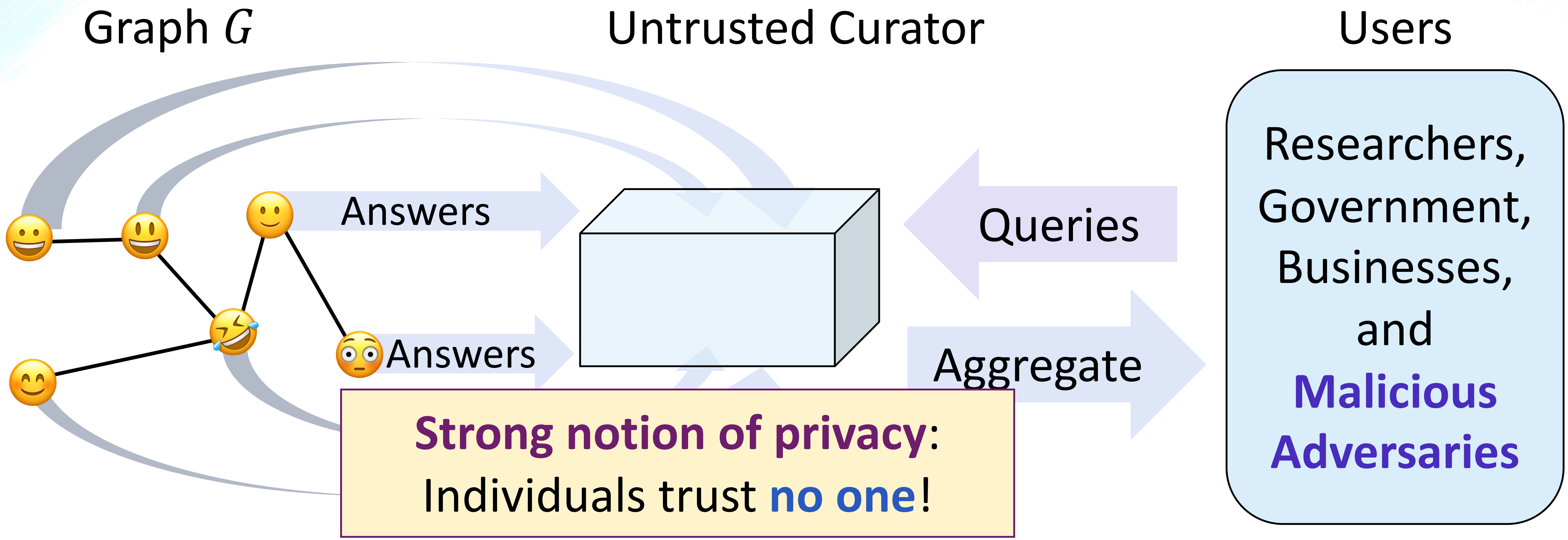
- **Each node publishes privatized** output
- Curator computes aggregated statistics using outputs

Weaker Notion of Trust: **Local Model**



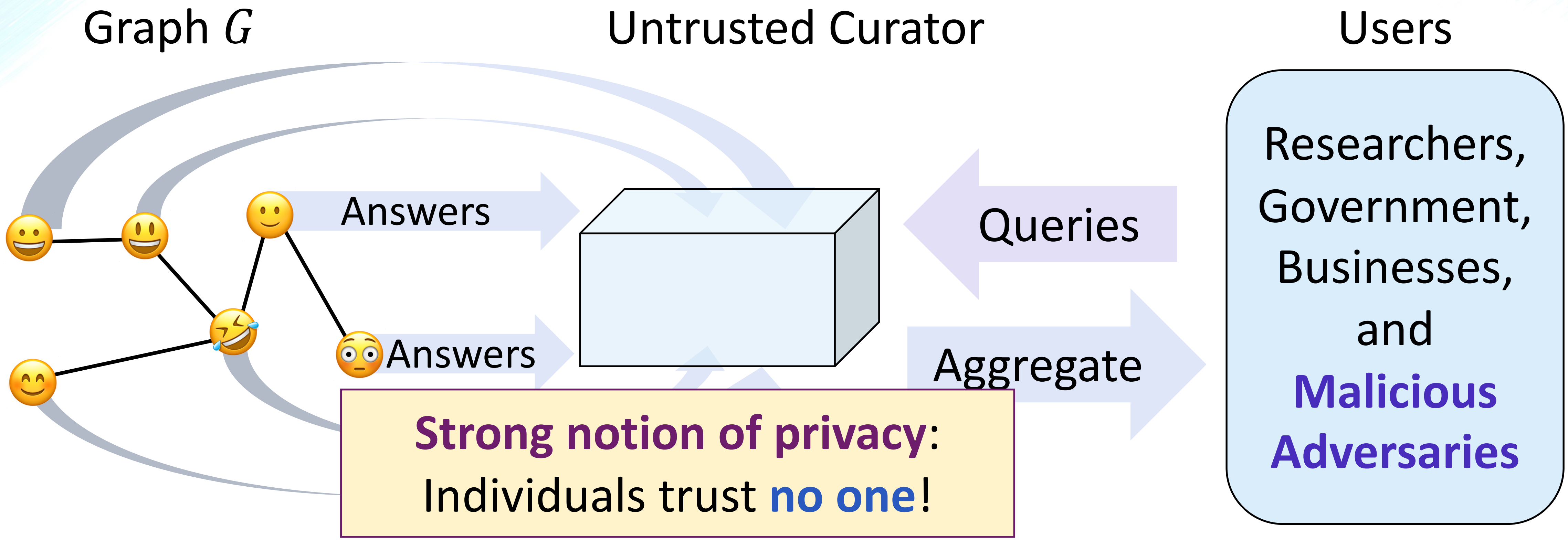
- **Each node publishes privatized** output
- Curator computes aggregated statistics using outputs

Weaker Notion of Trust: **Local Model**



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

Weaker Notion of Trust: **Local Model**



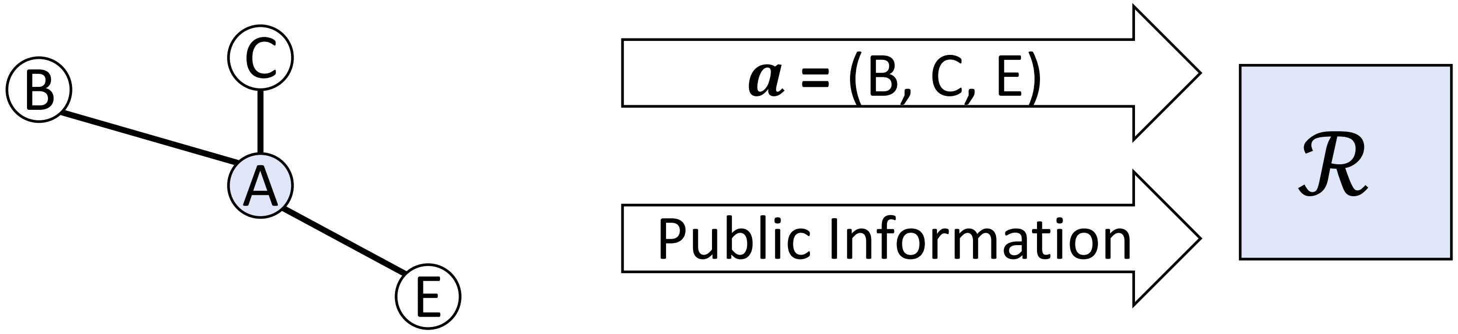
- **Each node publishes privatized** output
- Curator computes aggregated statistics using outputs

Local Edge Differential Privacy (LEDP)

Local Randomizer

[Adapted from Kasiviswanathan-Lee-Nissim-Raskhodnikova-Smith '11]

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

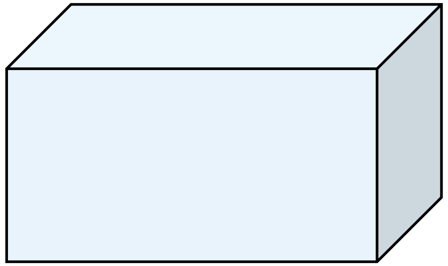


Local Edge Differential Privacy (LEDP)

Untrusted Curator

Users

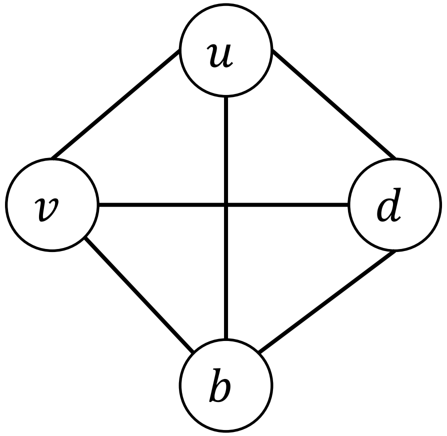
Algorithm proceeds in **rounds** in **distributed graph** using local randomizers



Queries

Aggregate

Researchers, Government, Businesses, and **Malicious Adversaries**



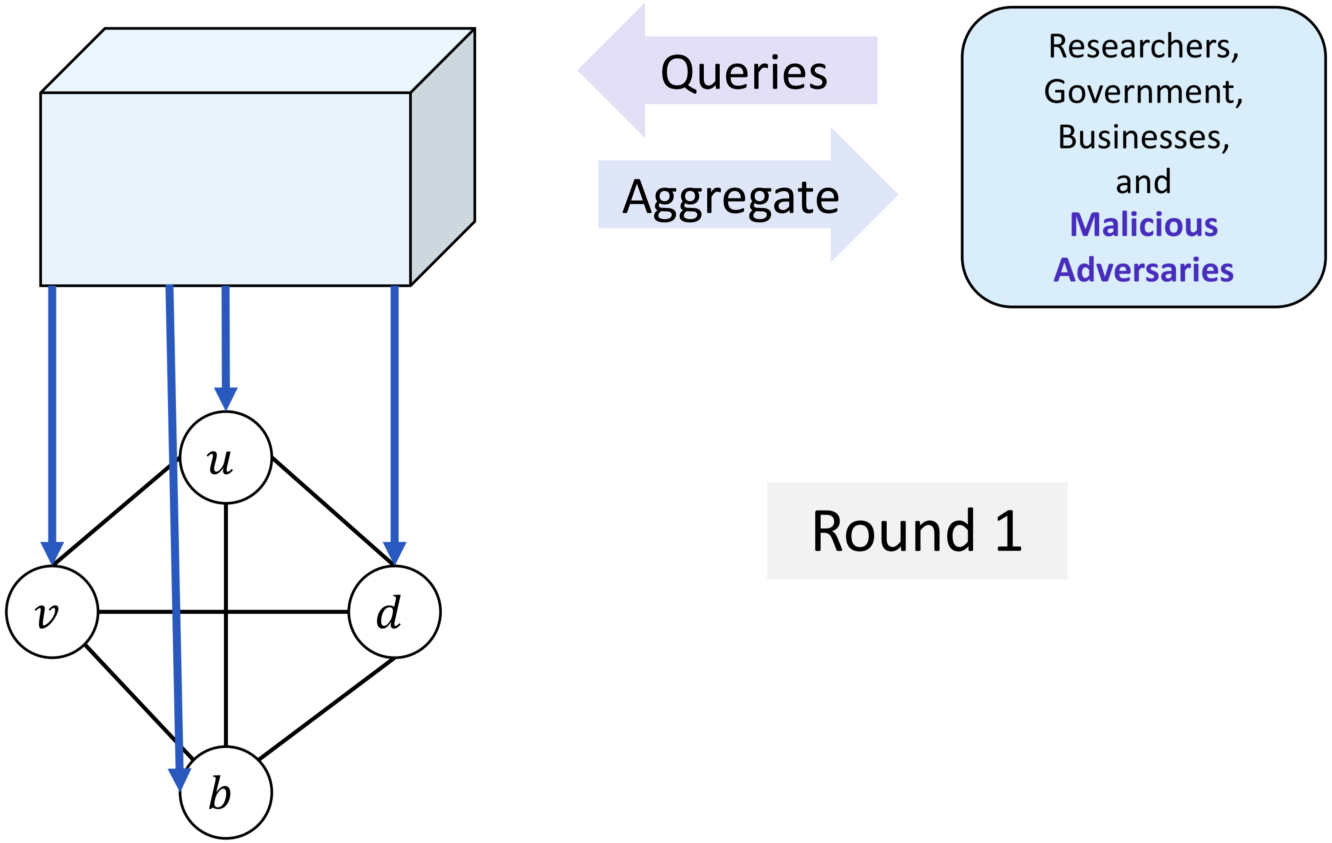
Round 1

Local Edge Differential Privacy (LEDP)

Untrusted Curator

Users

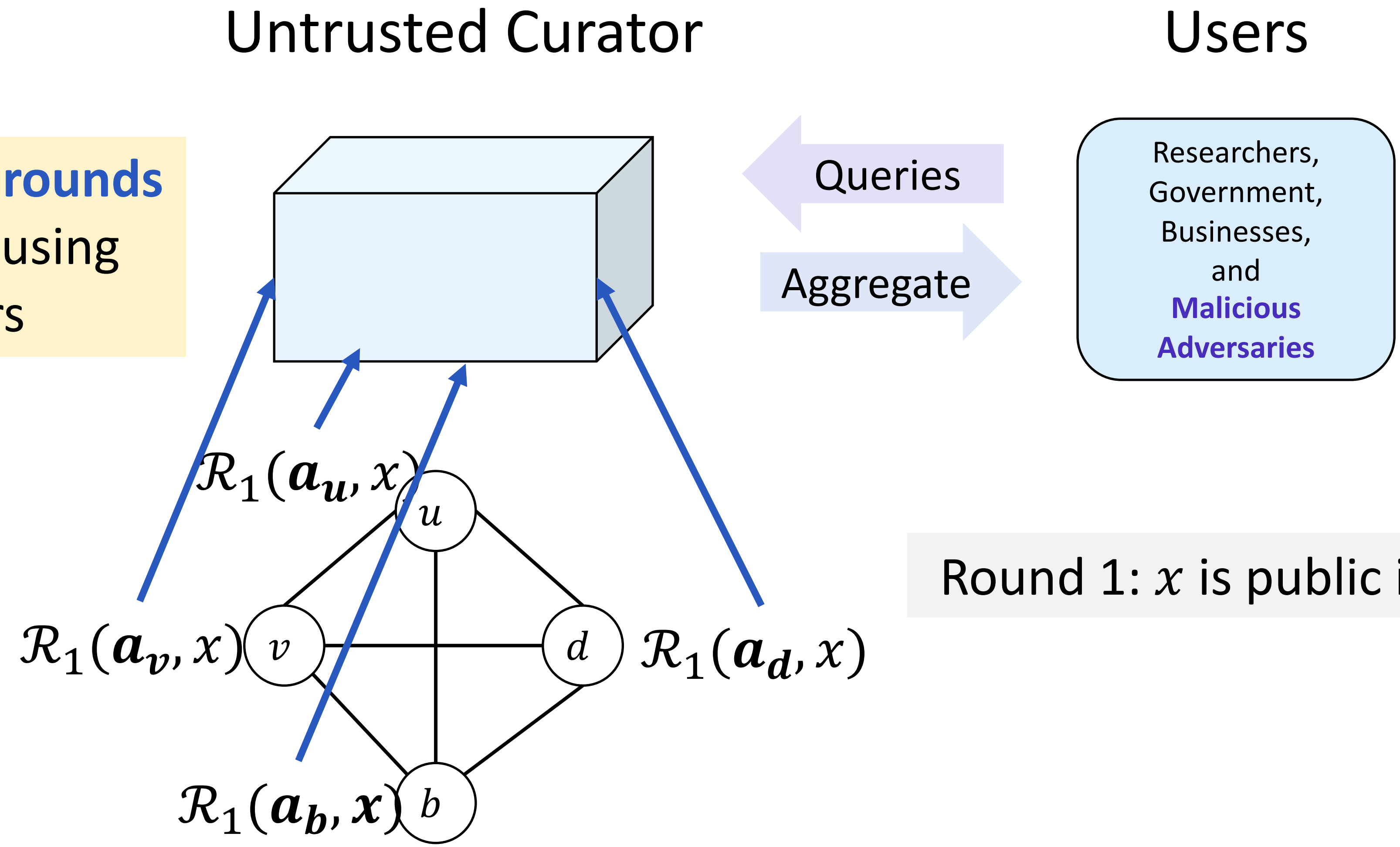
Algorithm proceeds in **rounds** in **distributed graph** using local randomizers



Round 1

Local Edge Differential Privacy (LEDP)

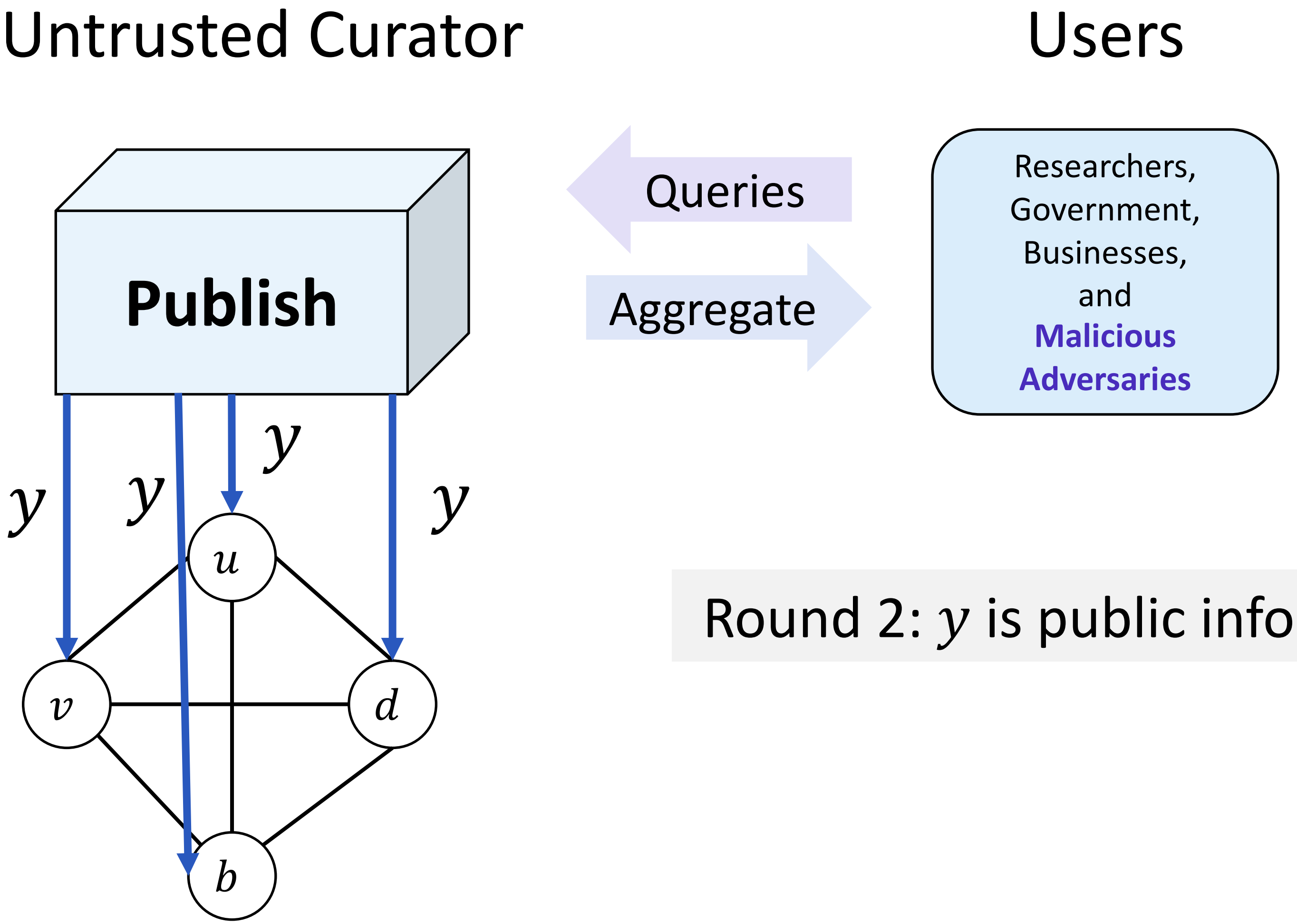
Algorithm proceeds in **rounds** in **distributed graph** using local randomizers



Round 1: x is public info

Local Edge Differential Privacy (LEDP)

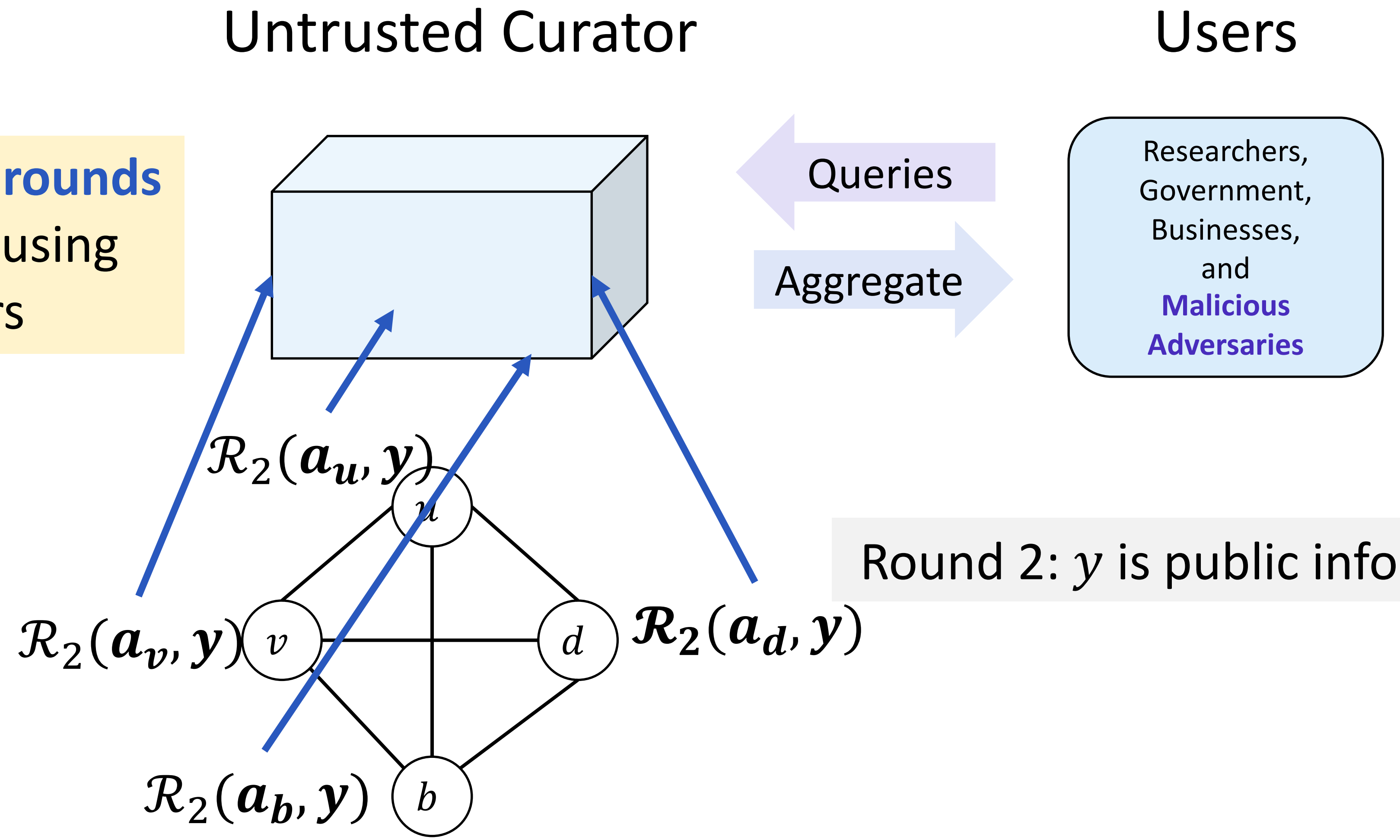
Algorithm proceeds in **rounds** in **distributed graph** using local randomizers



Round 2: y is public info

Local Edge Differential Privacy (LEDP)

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



Round 2: y is public info

Local Edge Differential Privacy (LEDP)

Untrusted Curator

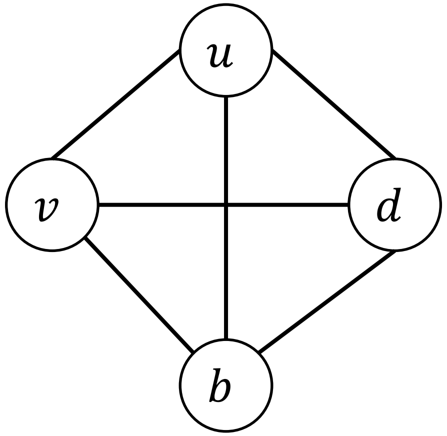
Users

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

Queries

Aggregate

Researchers,
Government,
Businesses,
and
Malicious Adversaries



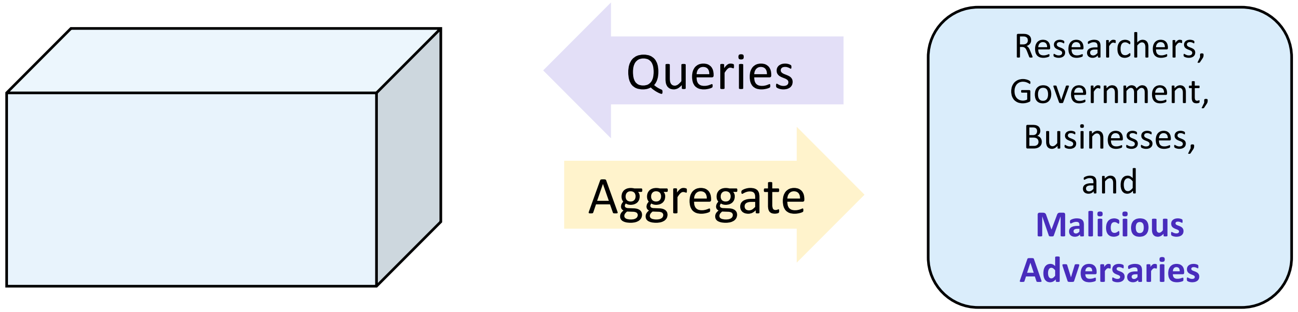
Round 2

Local Edge Differential Privacy (LEDP)

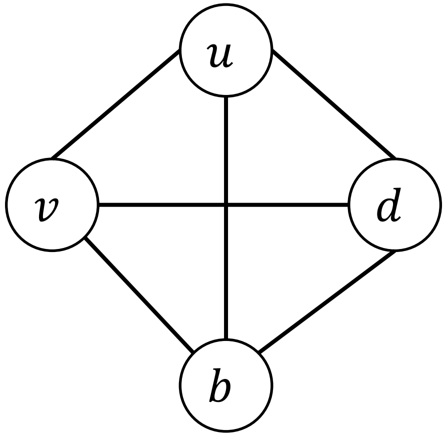
Untrusted Curator

Users

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



Relevant Complexity Measure:
Number of Rounds of Communication



Round 2

Local Edge Differential Privacy (LEDP)

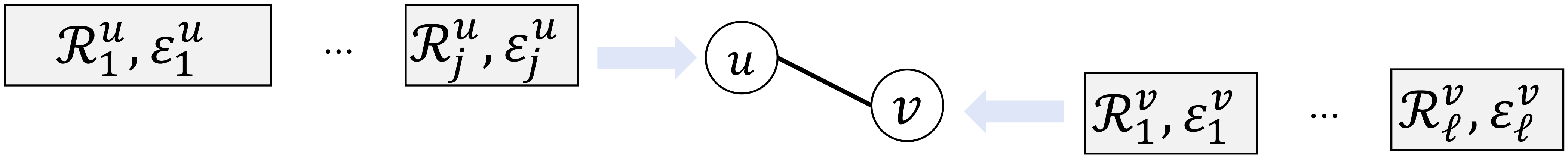
Local Edge Differential Privacy

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

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

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

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



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

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

Central DP vs. LEDP

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

Related Work

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

Related Work

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

Yes!

Our contributions and previous work

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

Our contributions and previous work

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

Our contributions and previous work

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

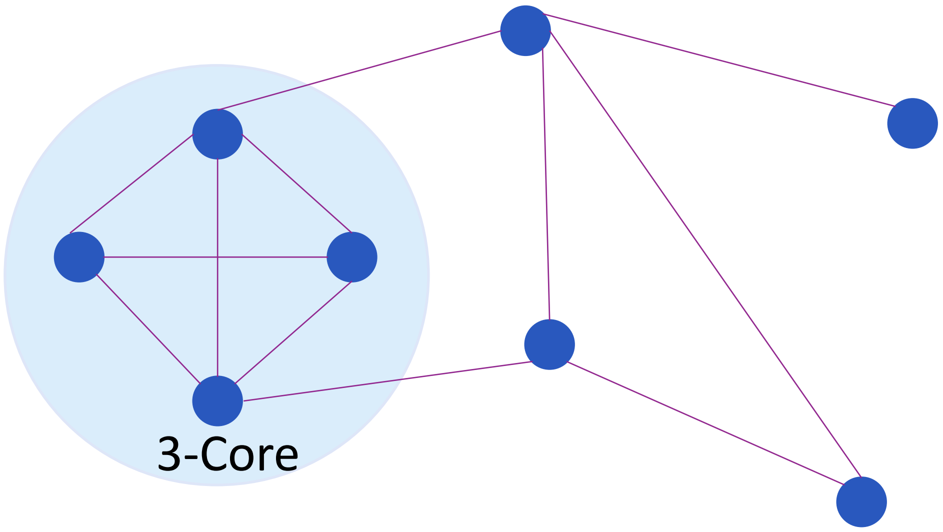
Our contributions and previous work

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

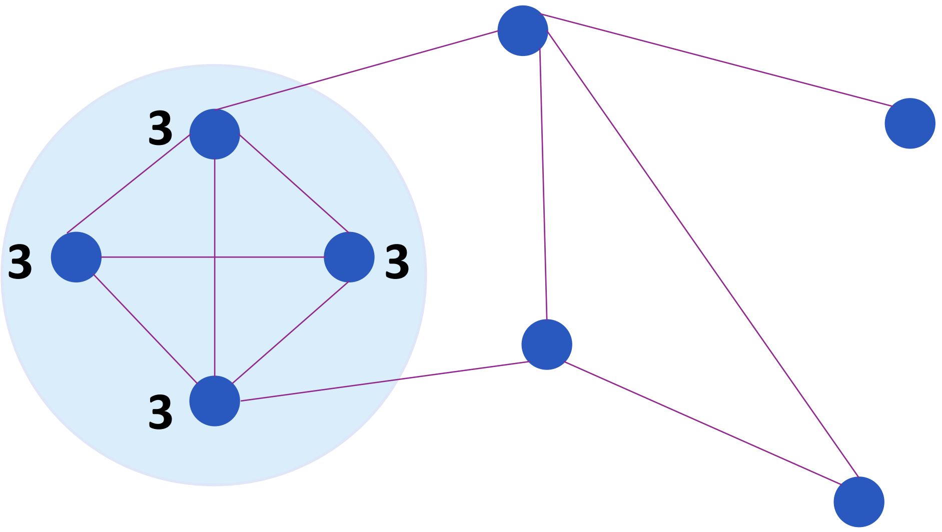
OPEN: Framework Approximation Guarantee

k-Core



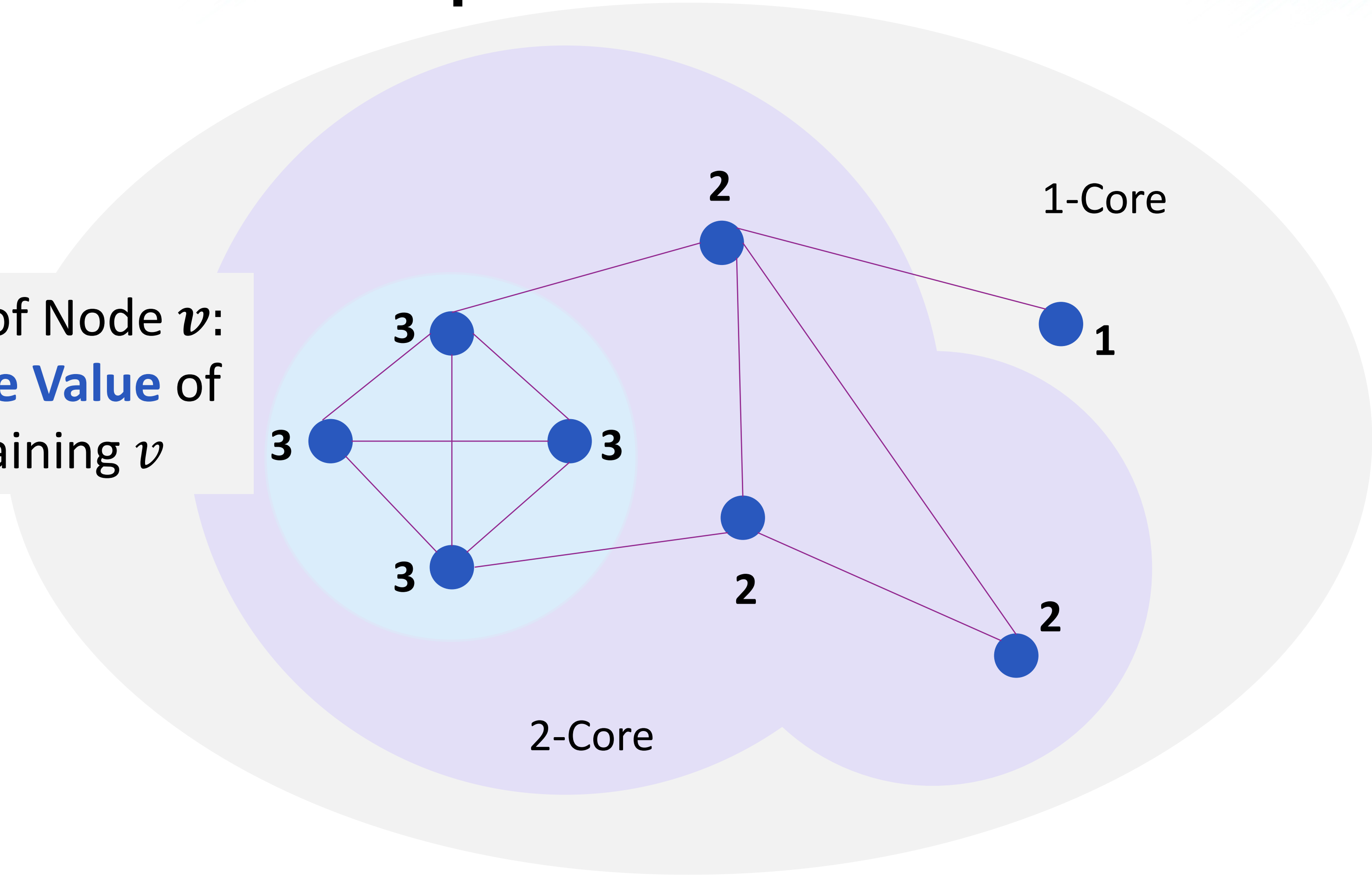
k-Core Decomposition

Core Number of Node v:
Maximum Core Value of
a Core Containing v

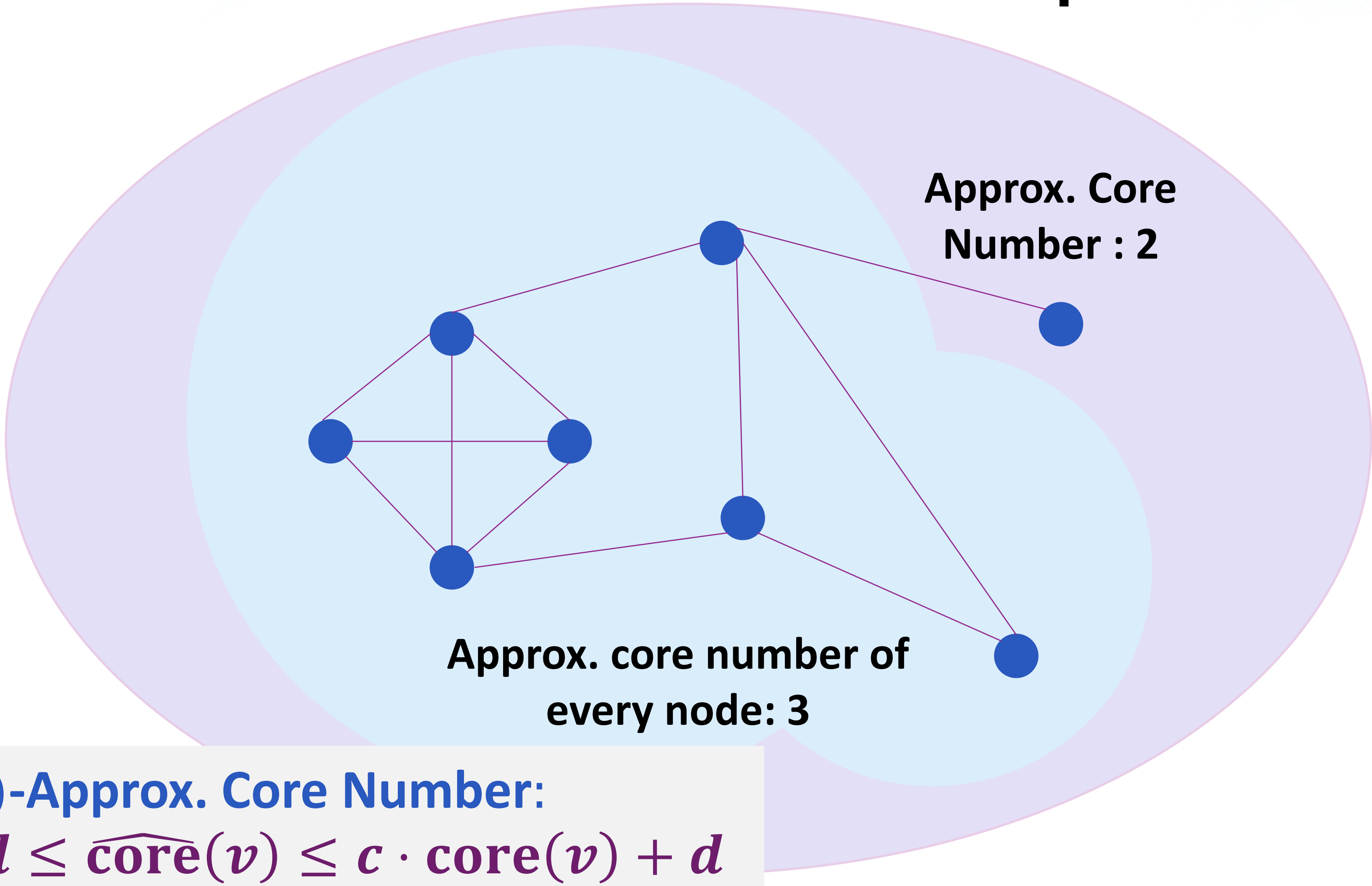


k-Core Decomposition

Core Number of Node v:
Maximum Core Value of a Core Containing v



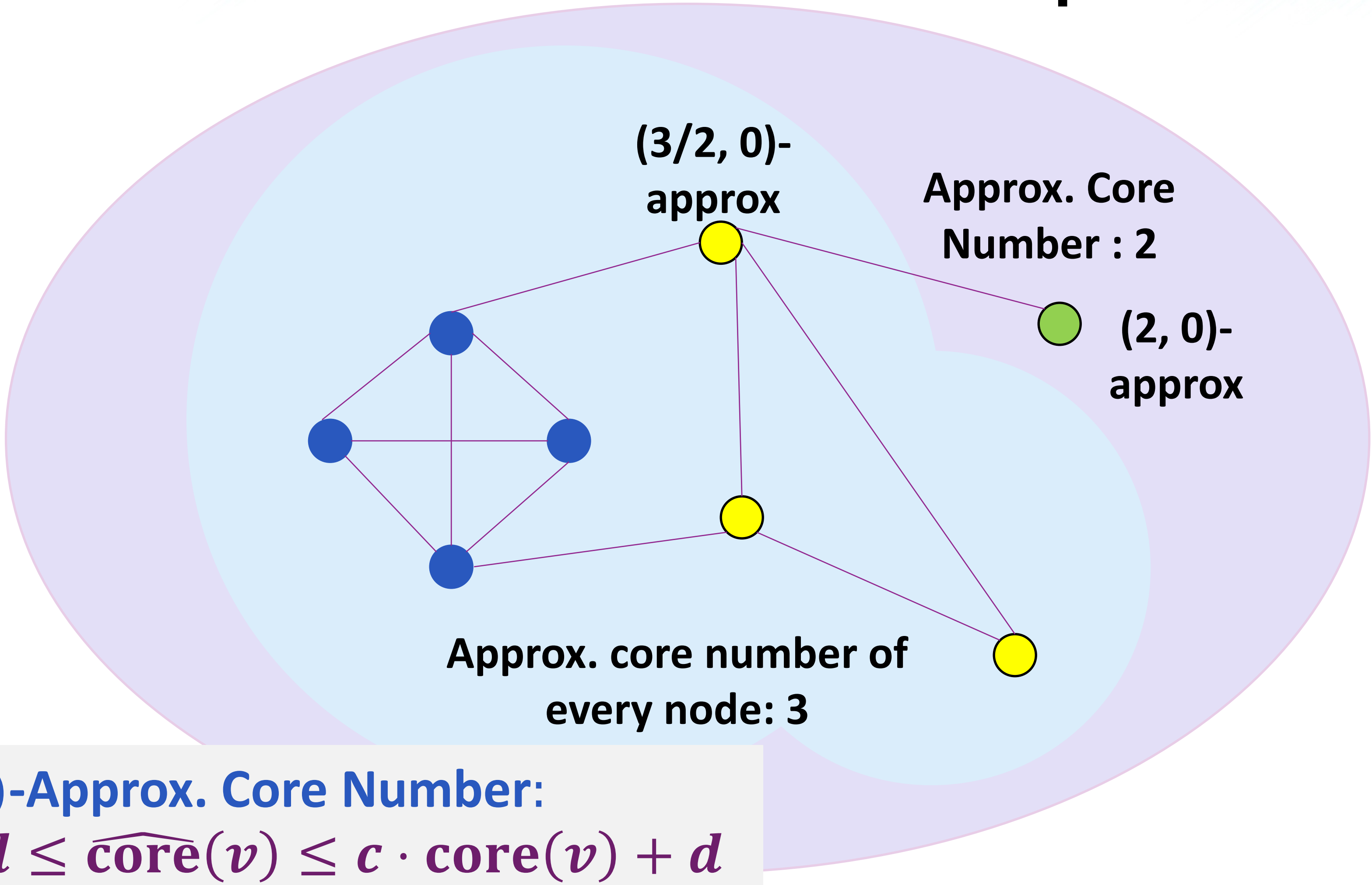
Approximate *k*-Core Decomposition



(c, d)-Approx. Core Number:

$$\mathrm{core}(v) - d \leq \widehat{\mathrm{core}}(v) \leq c \cdot \mathrm{core}(v) + d$$

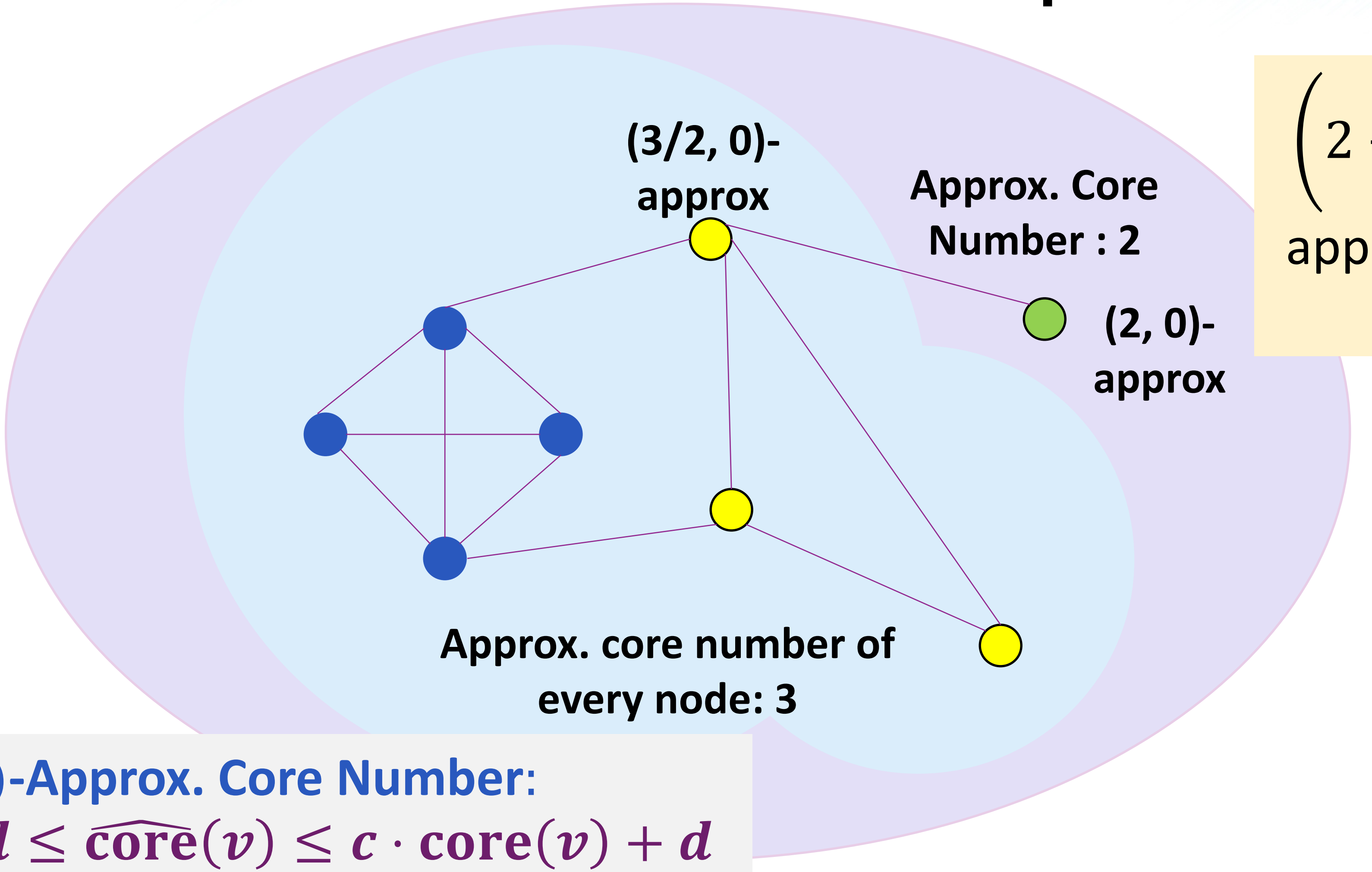
Approximate k-Core Decomposition



(c, d)-Approx. Core Number:

$$\mathbf{core}(v) - d \leq \widehat{\mathbf{core}}(v) \leq c \cdot \mathbf{core}(v) + d$$

Approximate k-Core Decomposition



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

(c, d)-Approx. Core Number:

$$\mathrm{core}(v) - d \leq \widehat{\mathrm{core}}(v) \leq c \cdot \mathrm{core}(v) + d$$

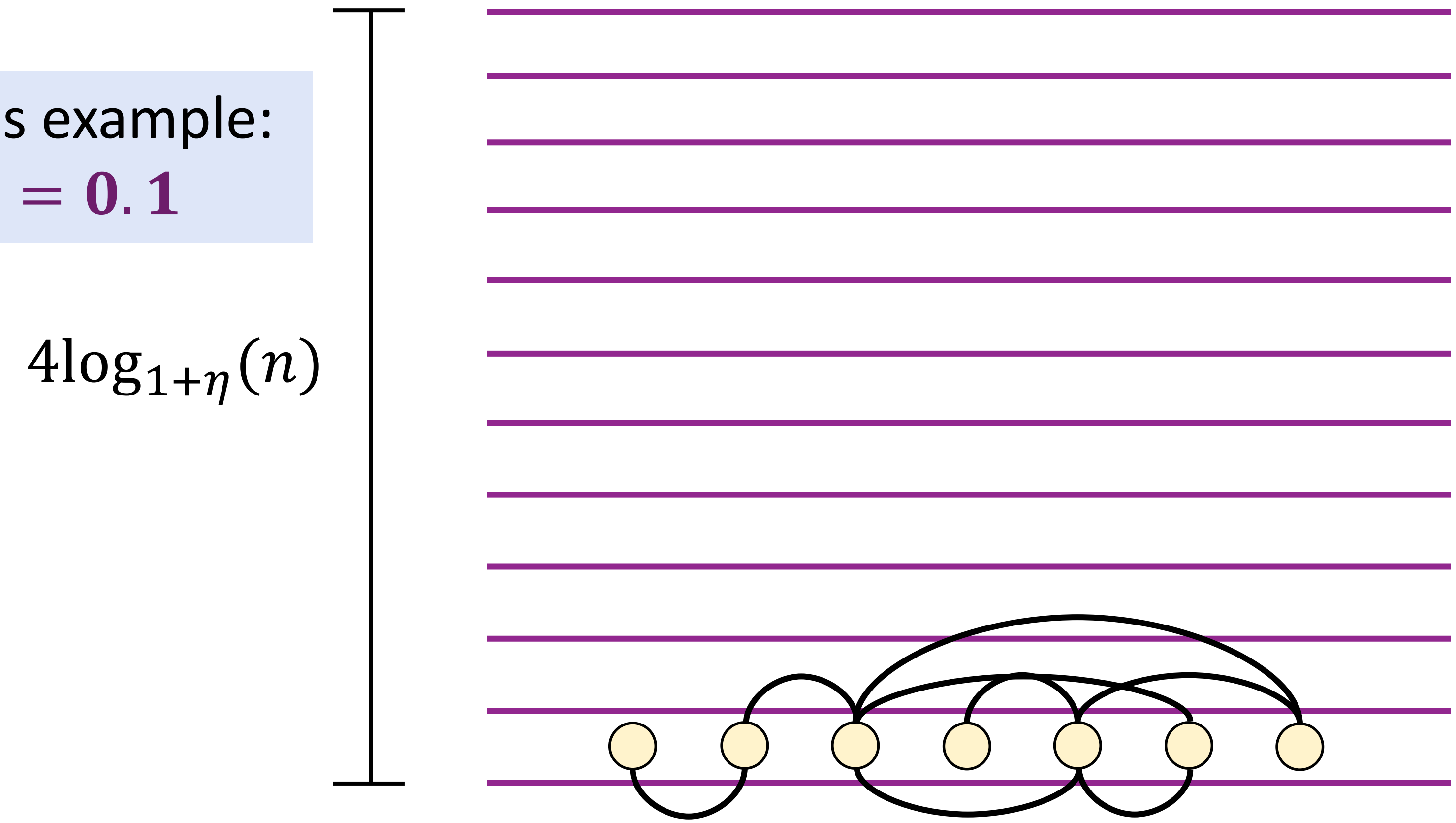
Level Data Structure and Core Numbers

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:
$\boldsymbol{\eta = 0.1}$



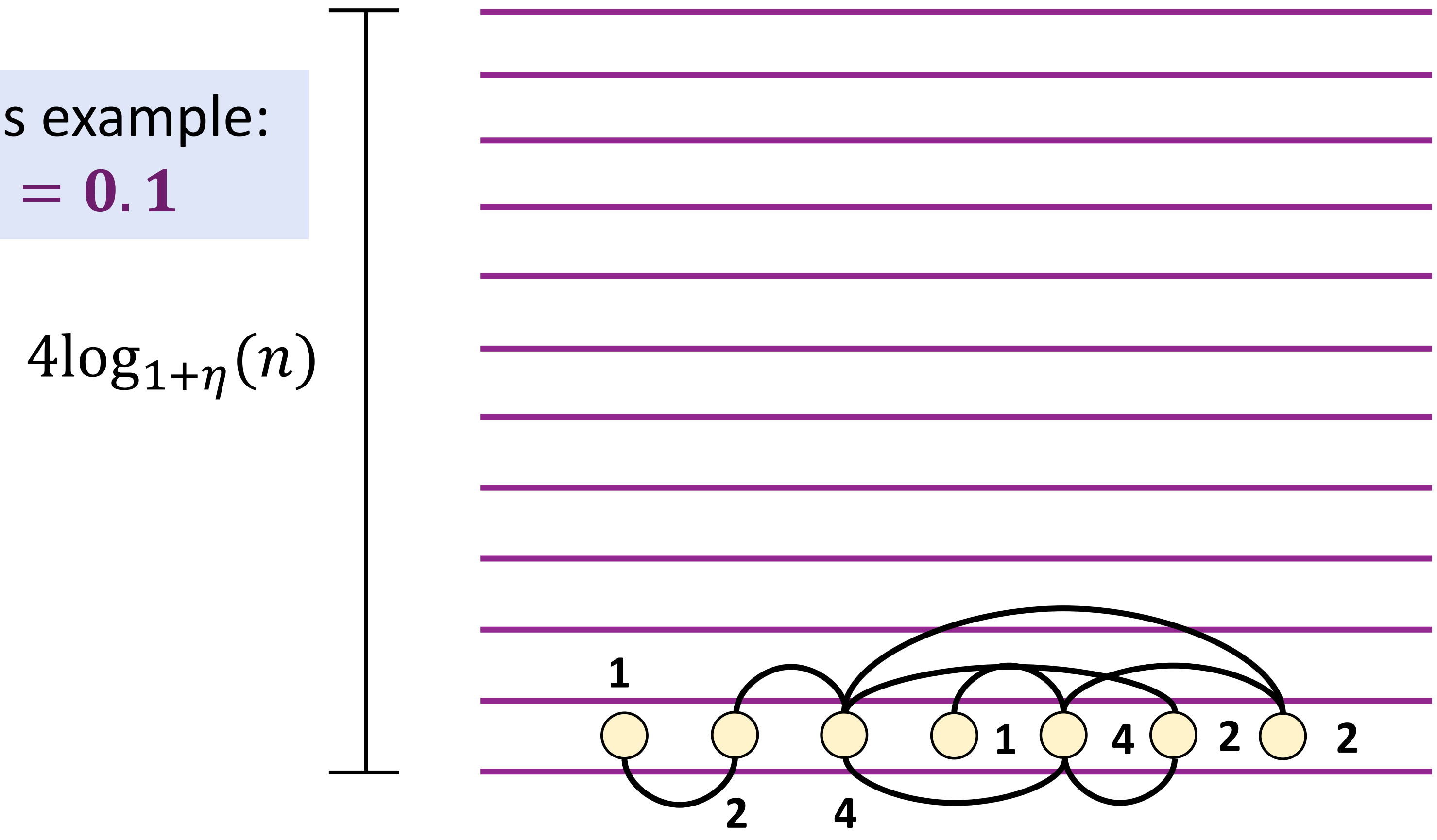
Move up if induced degree in **active** vertices $> (1 + \eta)$

Initially all vertices are active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]

Level Data Structure and Core Numbers

In this example:
$\boldsymbol{\eta = 0.1}$



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

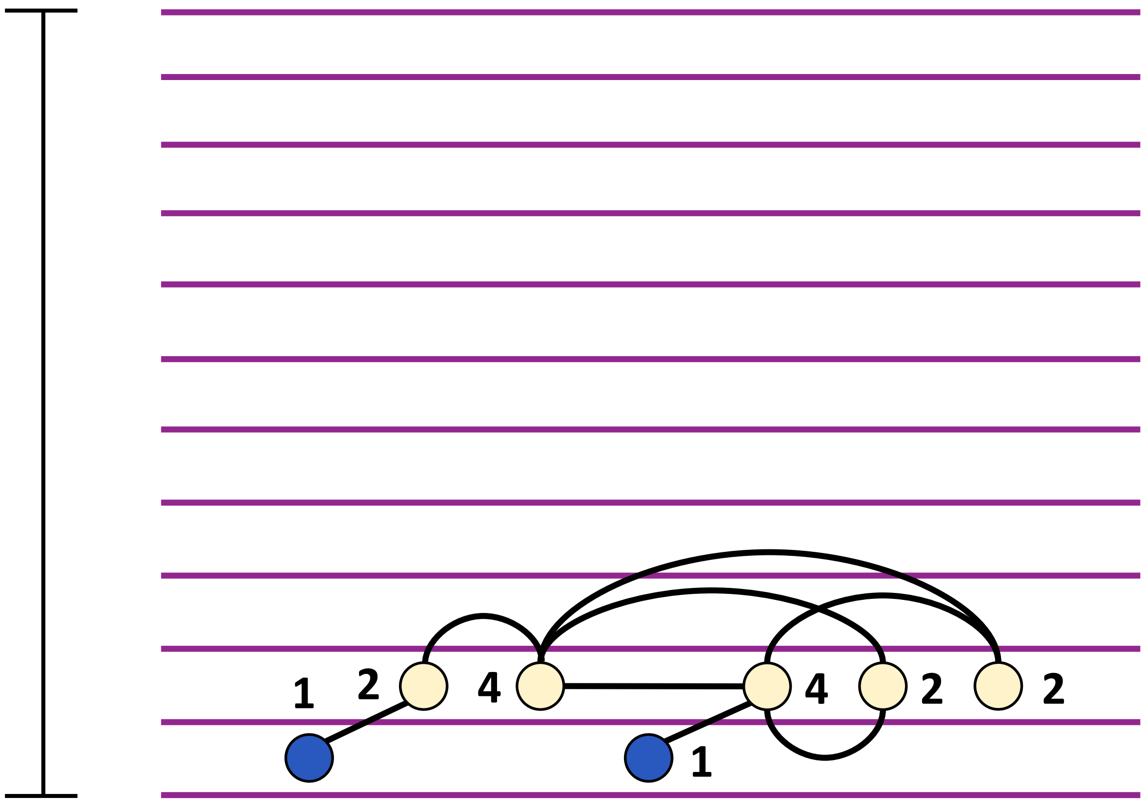
Initially all vertices are active

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]

Level Data Structure and Core Numbers

In this example:
$\boldsymbol{\eta = 0.1}$

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



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

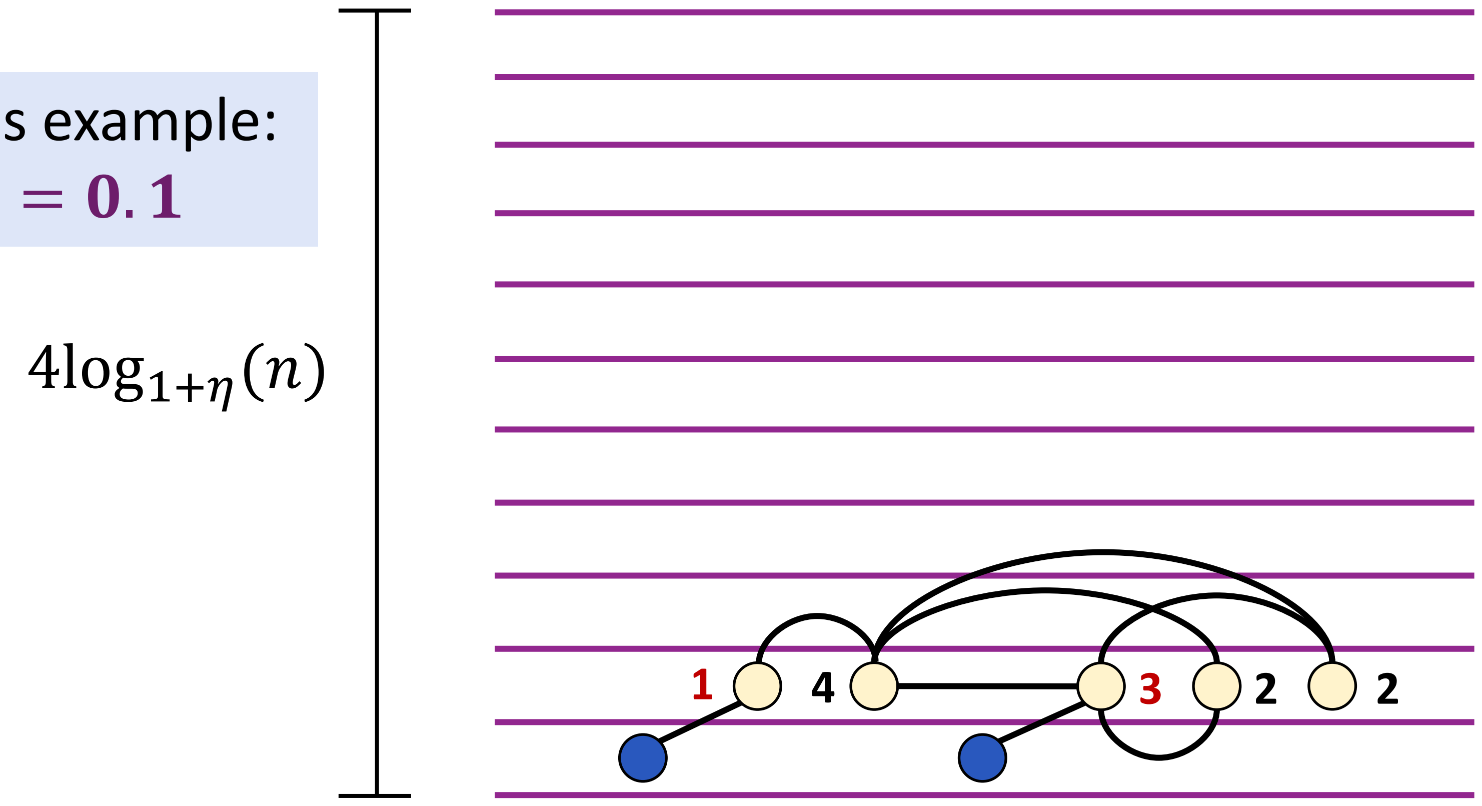
Initially all vertices are active

Vertices which **moved** remain **active**

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]

Level Data Structure and Core Numbers

In this example:
$\eta = 0.1$



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

Initially all vertices are active

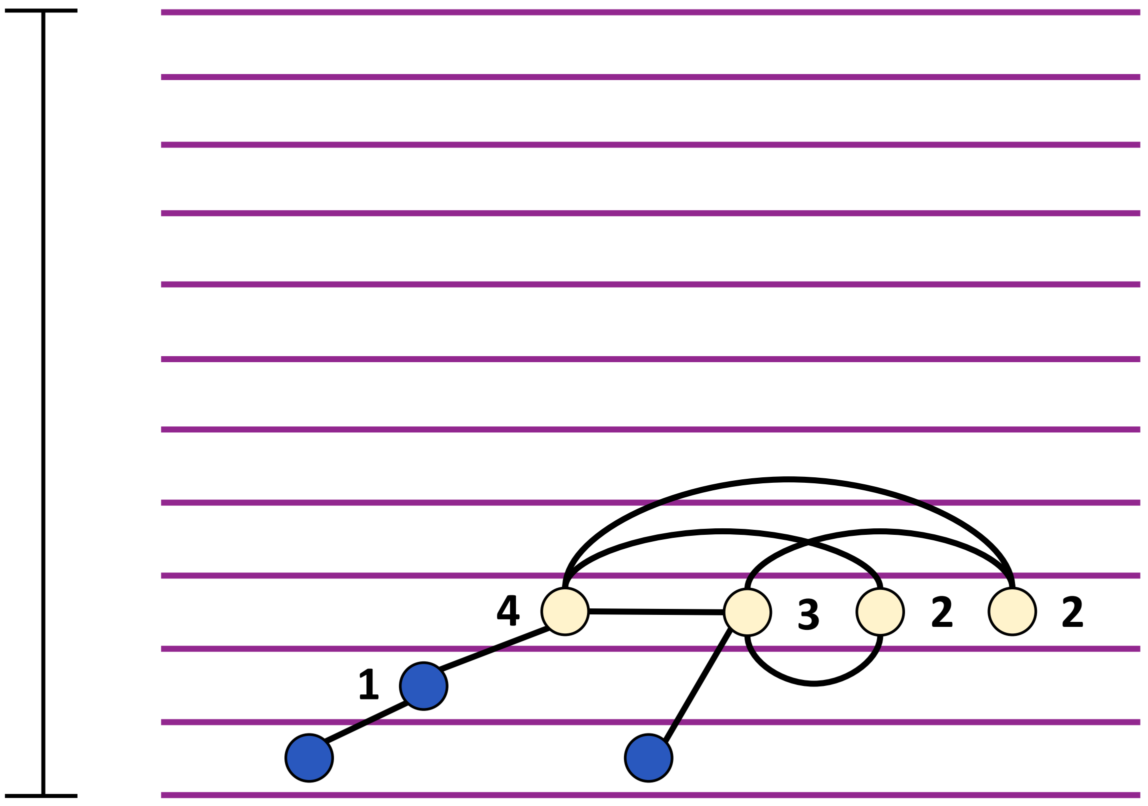
Vertices which **moved** remain **active**

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]

Level Data Structure and Core Numbers

In this example:
$\boldsymbol{\eta = 0.1}$

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



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

Initially all vertices are active

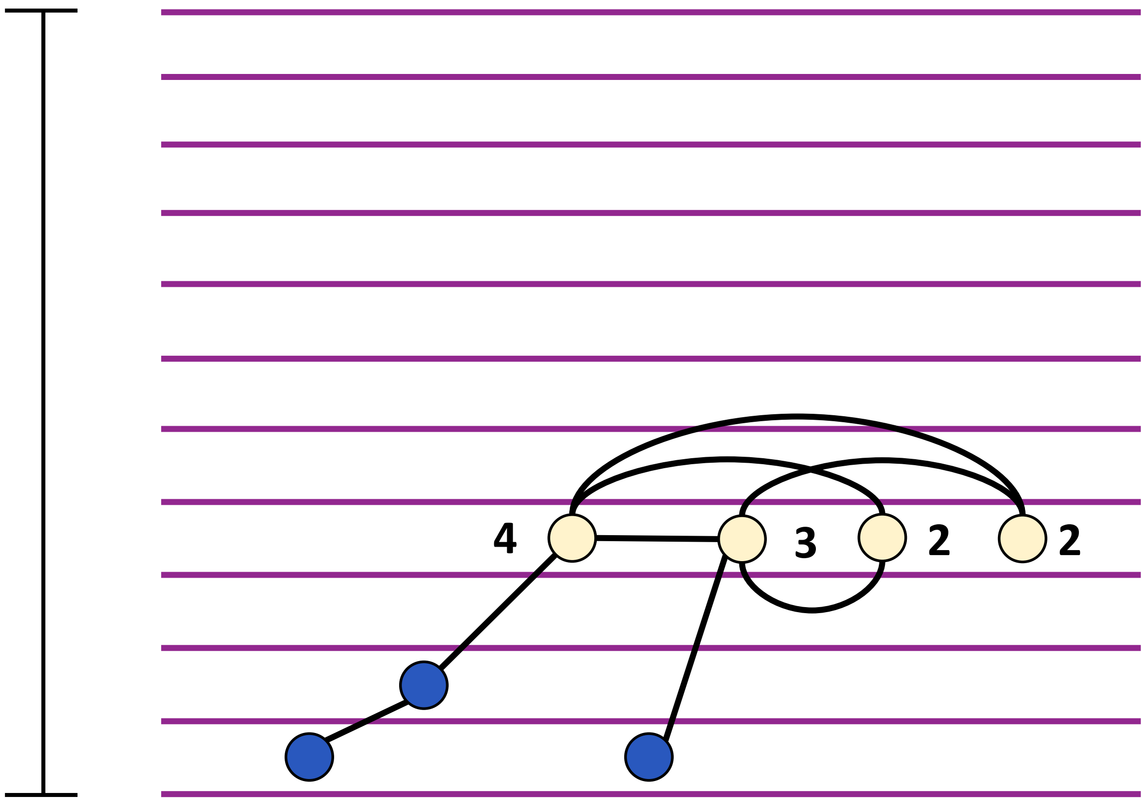
Vertices which **moved** remain **active**

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis '15, Henzinger-Neumann-Wiese '20, Liu-Shi-Yu-Dhulipala-Shun '22]

Level Data Structure and Core Numbers

In this example:
$\boldsymbol{\eta = 0.1}$

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



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

Initially all vertices are active

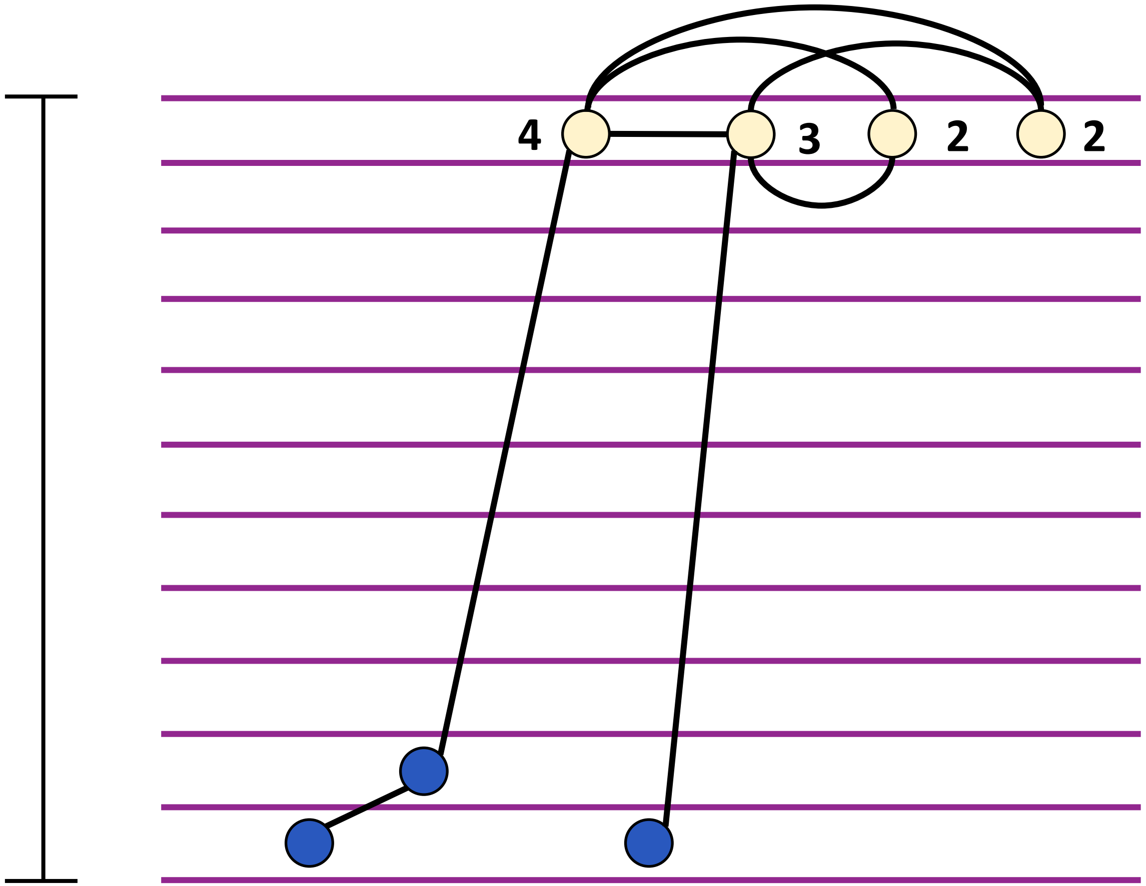
Vertices which **moved** remain **active**

[Bhattacharya-Henzinger-Nanongkai-Tsourakakis ‘15, Henzinger-Neumann-Wiese ‘20, Liu-Shi-Yu-Dhulipala-Shun ‘22]

Level Data Structure and Core Numbers

In this example:
$\boldsymbol{\eta = 0.1}$

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



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

$\boldsymbol{\eta = 0.1}$

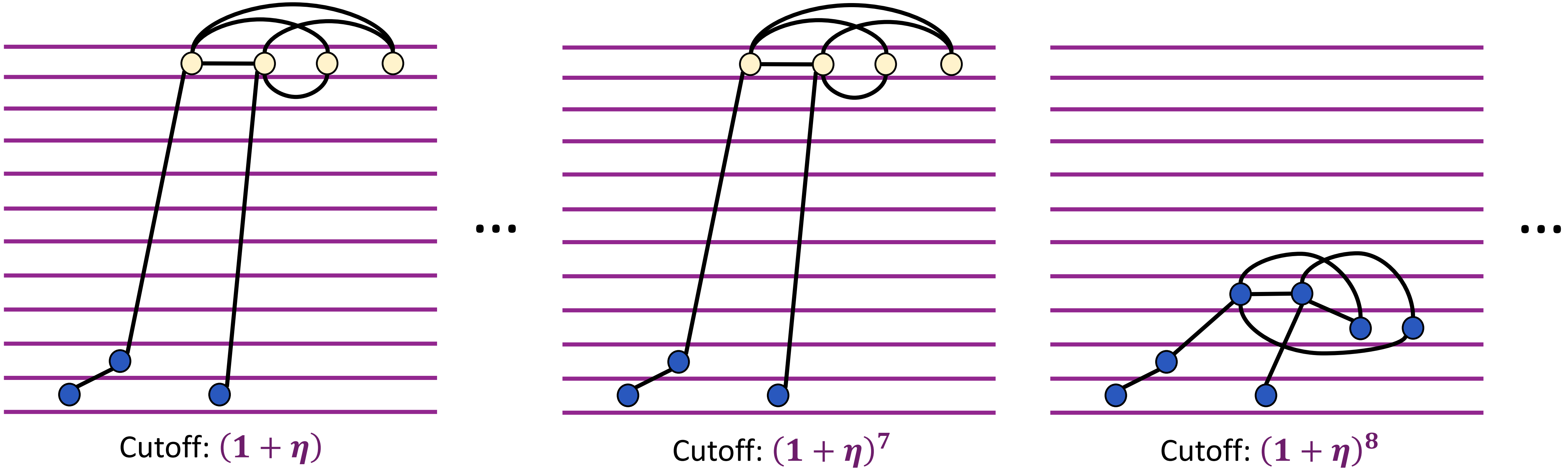
Set cutoffs $\boldsymbol{(1+\eta)^i}$ for all $\boldsymbol{i} \in [\boldsymbol{\log_{1+\eta}(n)}]$

[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 $i \in [\log_{1+\eta}(n)]$

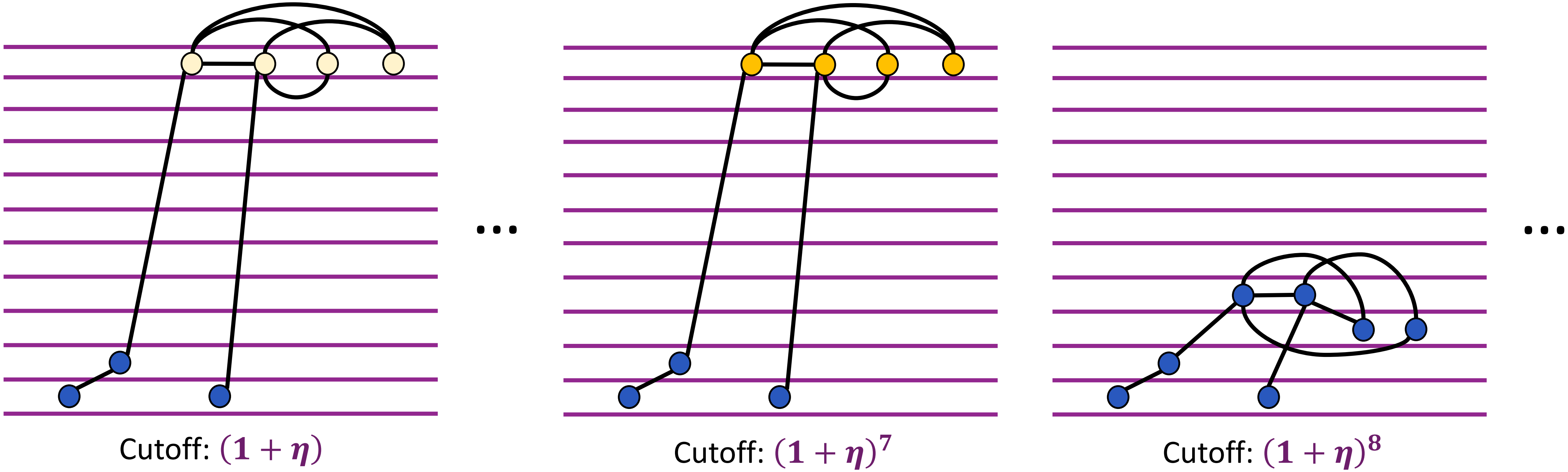


Give approx core number $2 \cdot (1+\eta)^i$
using **largest cutoff** where node is on the **topmost level**

Non-Private Core Number Approximation

$\eta = 0.1$

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

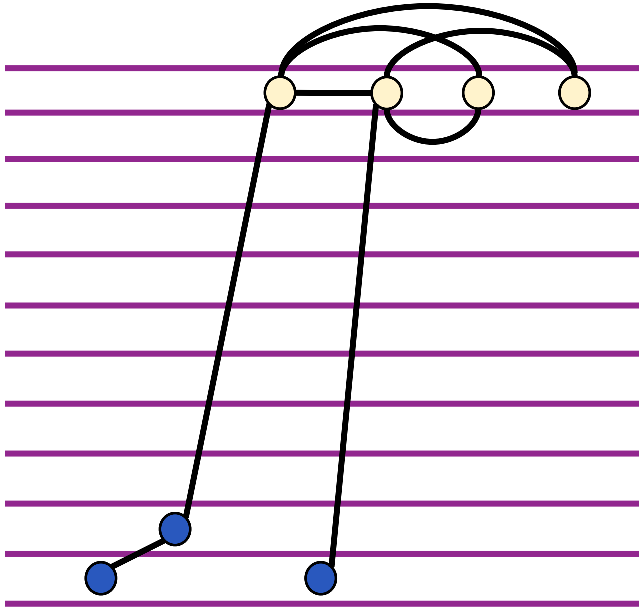


Approximation: $2 \cdot (1+\eta)^7 = 2 \cdot 1.1^7 = 2 \cdot 1.95 = 3.9$

Non-Private Core Number Approximation

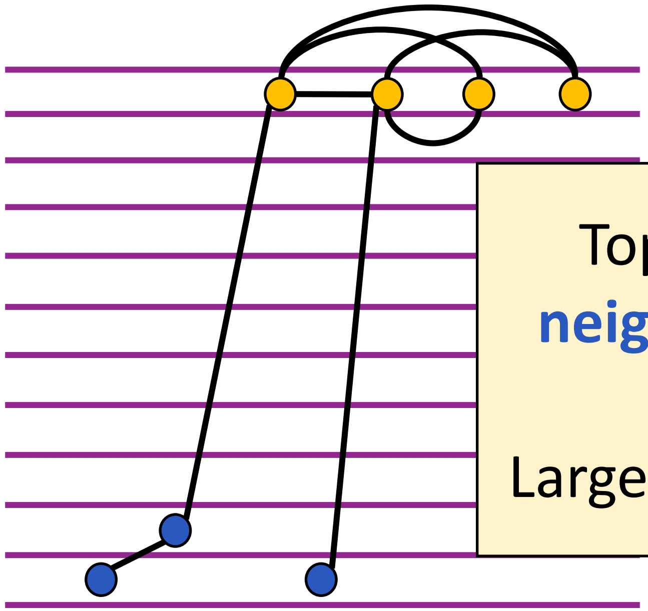
$\boldsymbol{\eta = 0.1}$

Set cutoffs $(\mathbf{1}+\boldsymbol{\eta})^{\boldsymbol{i}}$ for all $\boldsymbol{i} \in [\mathbf{log}_{\mathbf{1}+\boldsymbol{\eta}}(\boldsymbol{n})]$



Cutoff: $(\mathbf{1}+\boldsymbol{\eta})$

…



Cutoff: $(\mathbf{1}+\boldsymbol{\eta})^{\mathbf{7}}$

Top level means adjacent to **many neighbors of sufficiently high degree**

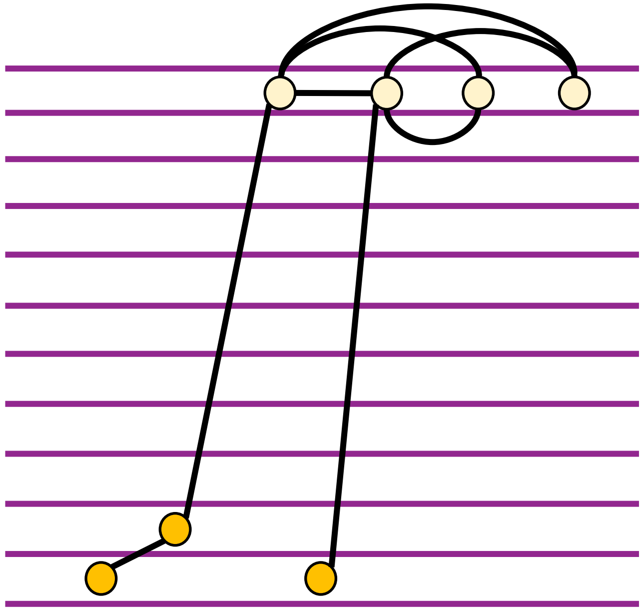
Largest cutoff gives **largest such degree**

Approximation: $2 \cdot (1+\eta)^7 = 2 \cdot 1.1^7 = 2 \cdot 1.95 = 3.9$

Non-Private Core Number Approximation

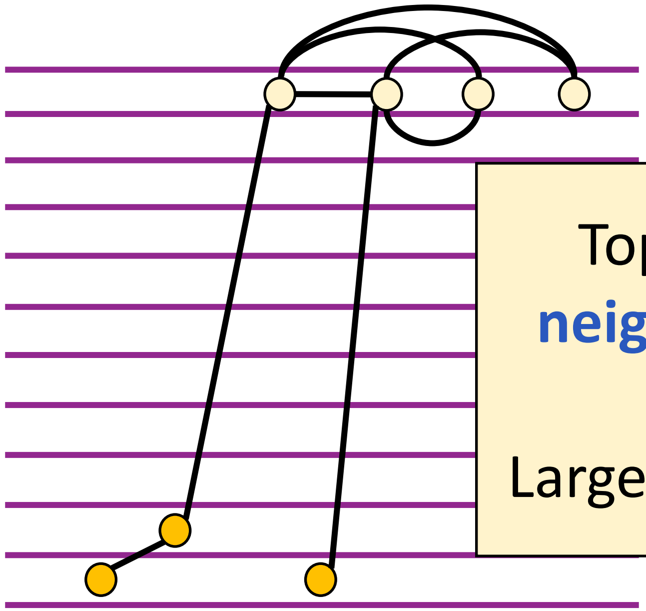
$\eta = 0.1$

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



Cutoff: $(1+\eta)$

...



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

Top level means adjacent to **many neighbors of sufficiently high degree**

Largest cutoff gives **largest such degree**

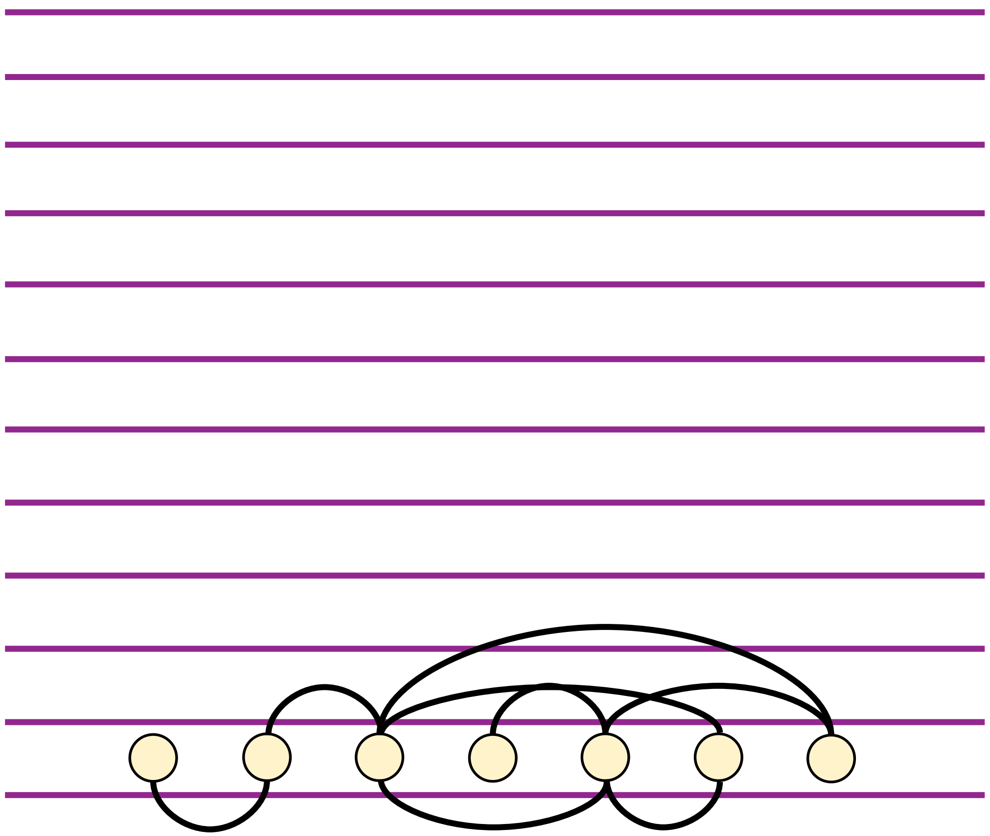
Approximation: $\mathbf{1}$

ε-LEDP Core Numbers

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

Distribution $\boldsymbol{Geom(b)}$

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



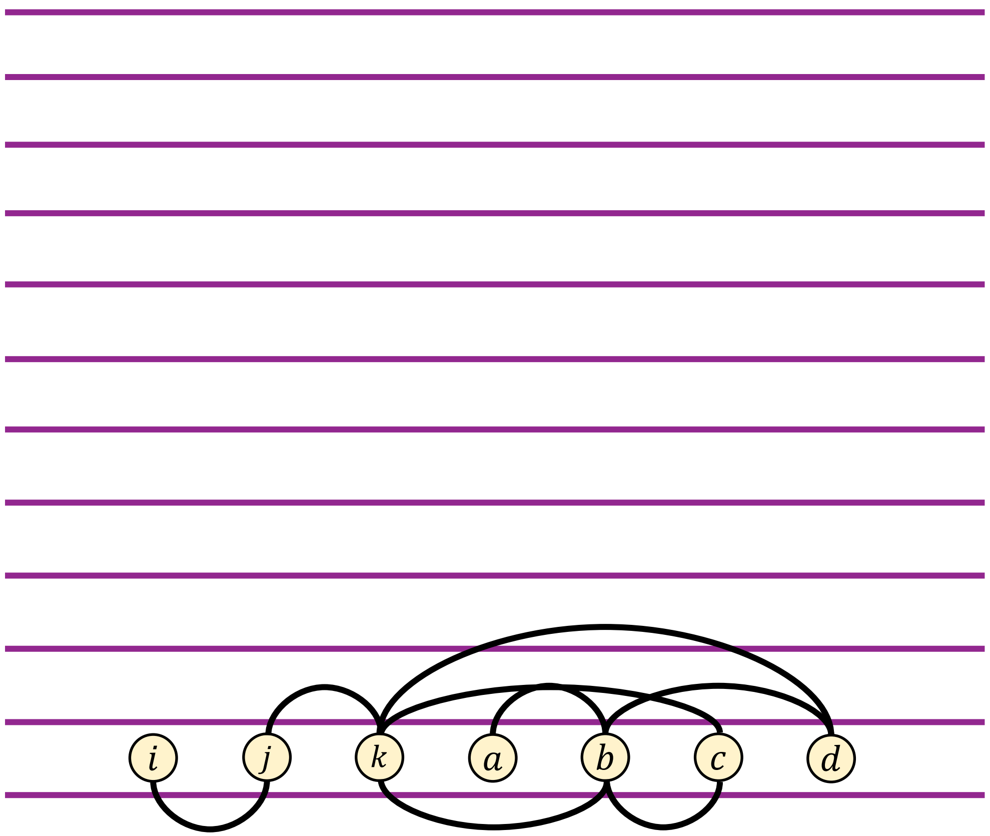
Release and move up degree **+ noise** in **active** vertices $> (1+\eta)$

ε-LEDP Core Numbers

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

Distribution $\boldsymbol{Geom(b)}$

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



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

ε-LEDP Core Numbers

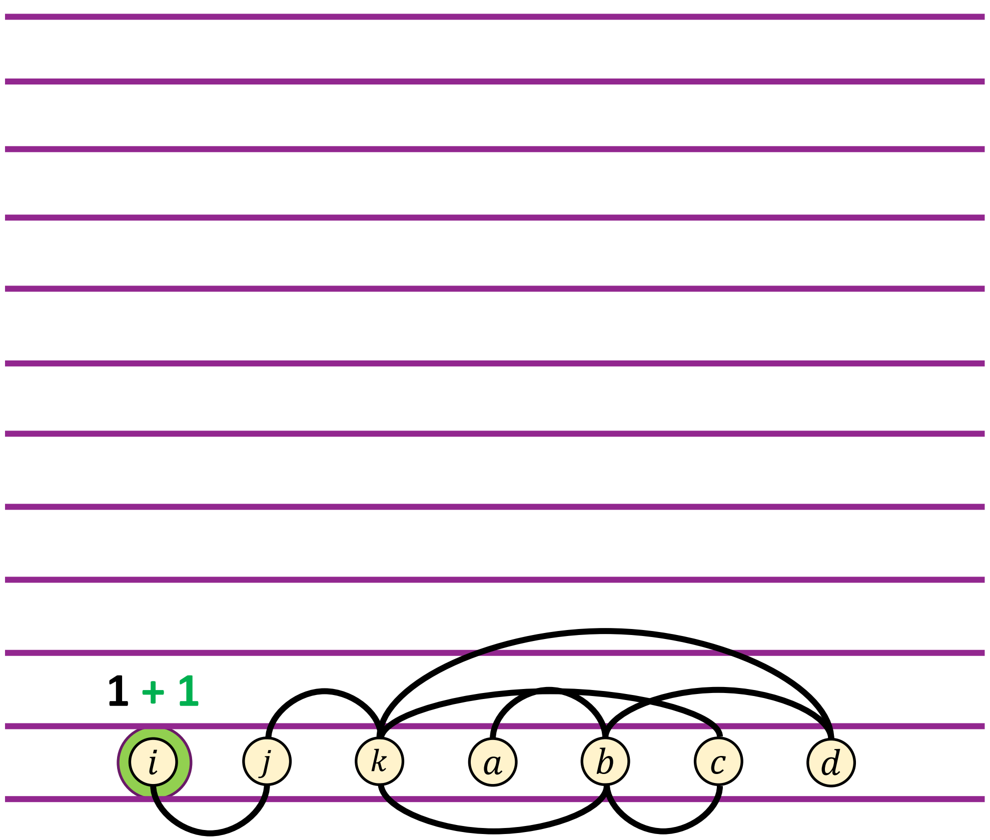
Each **active vertex draws i.i.d. noise** from **symmetric geometric distribution**

Distribution $\boldsymbol{Geom(b)}$

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

If $\mathbf{deg}(\boldsymbol{i}) + \boldsymbol{N_i} > (\mathbf{1} + \boldsymbol{\eta})$, **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 $> (1 + \eta)$

In this example: $\boldsymbol{\eta = 0.1}$

ε-LEDP Core Numbers

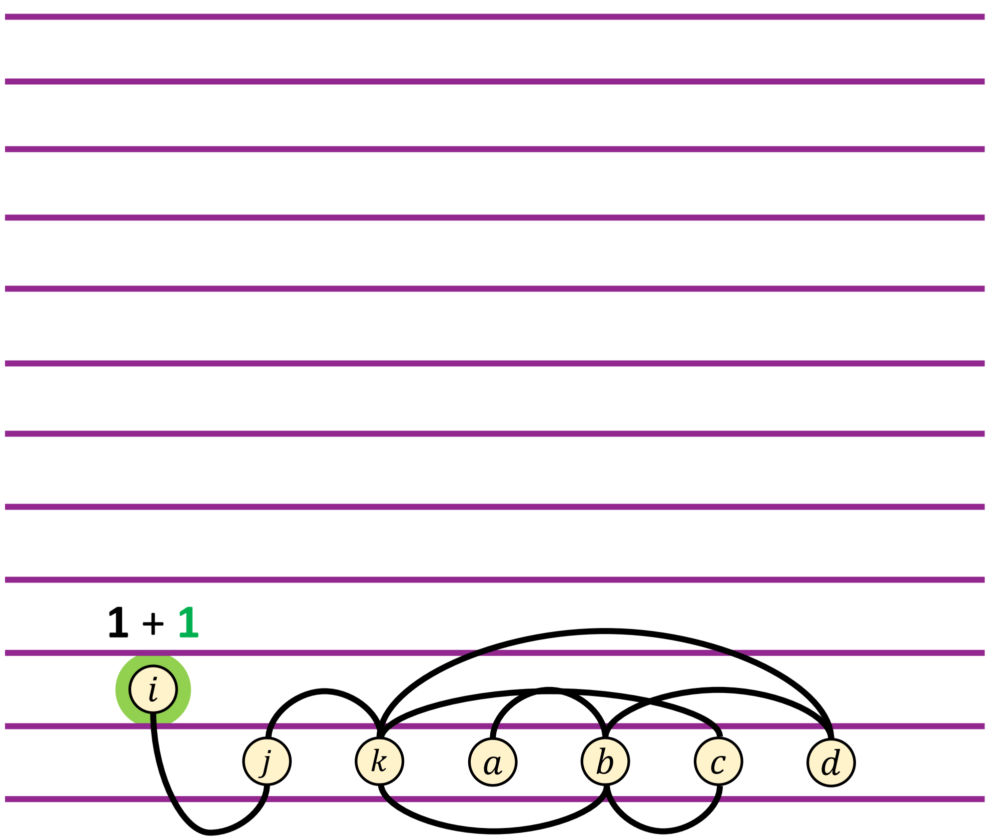
Each **active vertex draws i.i.d. noise** from **symmetric geometric distribution**

Distribution $\boldsymbol{Geom(b)}$

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

If $\mathbf{deg}(\boldsymbol{i}) + \boldsymbol{N_i} > (\mathbf{1} + \boldsymbol{\eta})$, **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 $> (1+\eta)$

ε-LEDP Core Numbers

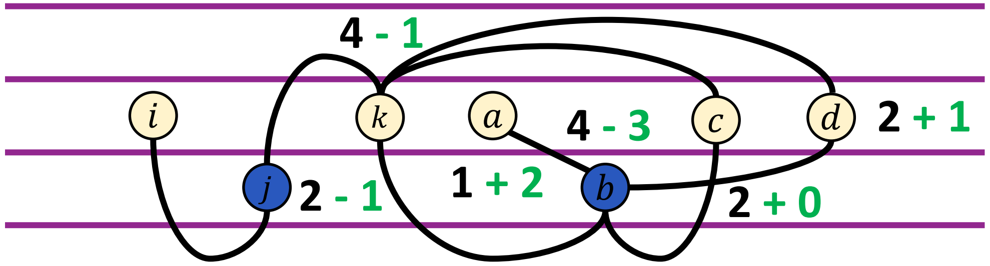
Each **active vertex draws i.i.d. noise** from **symmetric geometric distribution**

Distribution $\boldsymbol{Geom(b)}$

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

If $\mathbf{deg(i) + N_i > (1+\eta)}$, **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

ε-LEDP Core Numbers

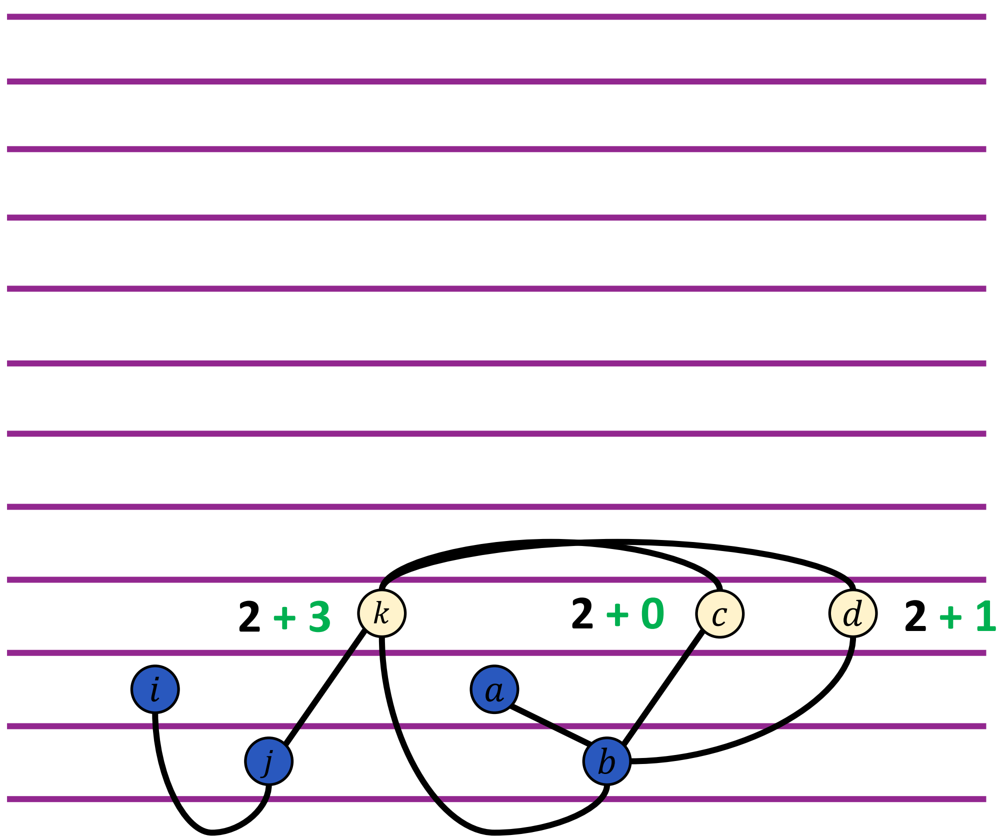
Each **active vertex draws i.i.d. noise** from **symmetric geometric distribution**

Distribution $\boldsymbol{Geom(b)}$

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

If $\mathbf{deg}(\boldsymbol{i}) + \boldsymbol{N_i} > (\mathbf{1} + \boldsymbol{\eta})$, **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

Approx. as before $\mathbf{2(1 + \boldsymbol{\eta})^{\boldsymbol{i}}}$ using topmost level

ε-LEDP Core Numbers

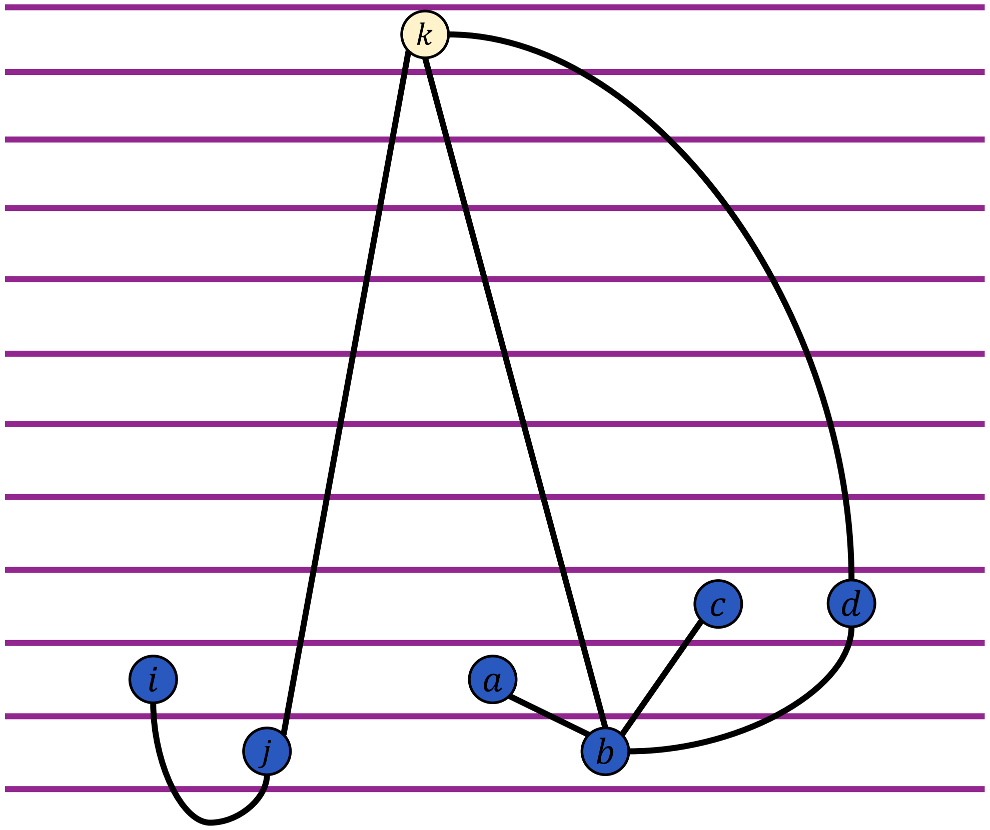
Each **active vertex draws i.i.d. noise** from **symmetric geometric distribution**

Distribution $\boldsymbol{Geom(b)}$

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

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

Where $N_k \sim Geom\left(\frac{\varepsilon}{8\log^2_{1+\eta}(n)}\right)$



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

Redraw new noise each time vertex remains active

Approx. as before $\mathbf{2(1+\eta)^i}$ using topmost level

ε-LEDP Core Numbers

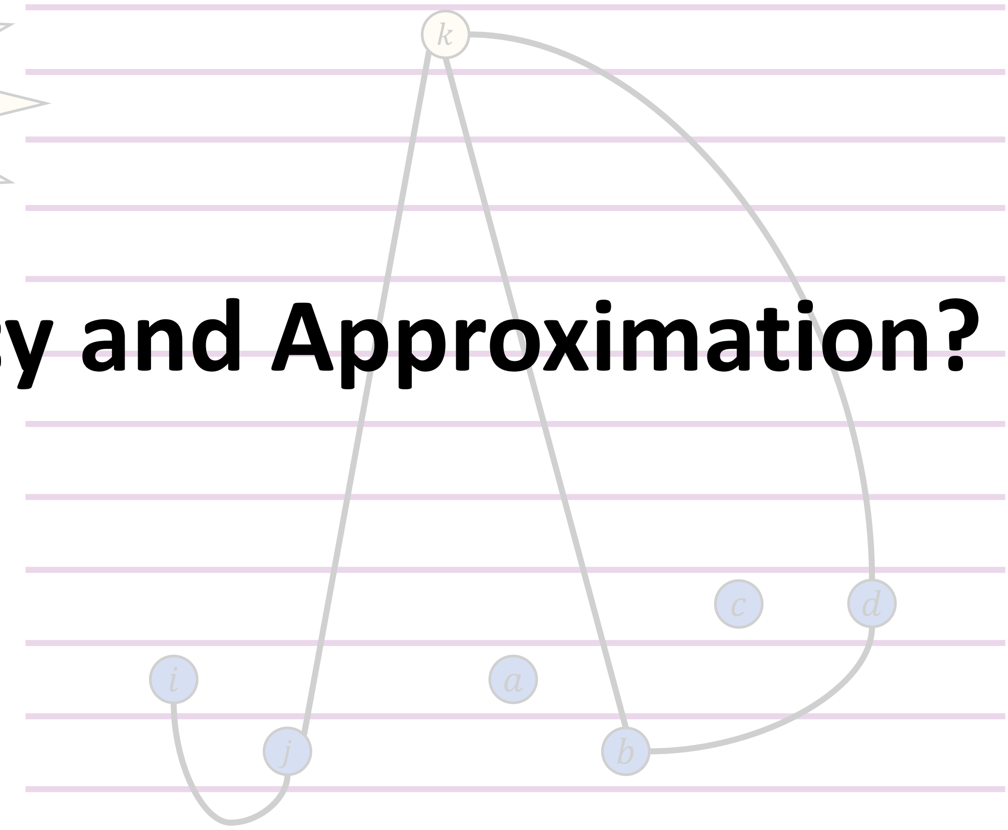
Each **active vertex** **draws i.i.d. noise** from **symmetric geometric distribution**

Distribution $\boldsymbol{Geom(b)}$

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

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

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



Privacy and Approximation?

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 $\mathbf{2(1+\eta)^i}$ where i largest that vertex is on the topmost level

Privacy Proof

- Can be implemented via **local randomizers $\boldsymbol{R}$**

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

Privacy Proof

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

Privacy Proof

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

Global Sensitivity:

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

$$f(\boldsymbol{a}, A) = |\boldsymbol{a} \cap A|$$

Privacy Proof

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

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

Geometric Mechanism:

[Chan-Shi-Song '11; Balcer-Vadhan '18]

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

$\boldsymbol{M}$ **is** $\boldsymbol{\varepsilon}$**-DP**

Privacy Proof

- Can be implemented via **local randomizers $\boldsymbol{R}$**
- R takes as **input $\boldsymbol{a}$ (adjacency list)** and **public set of active vertices A**
  - R computes size of intersection $|\boldsymbol{a} \cap \boldsymbol{A}|$ **Sensitivity of 1**
  - Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log_{1+\eta}^2(n)}\right)$
- **$\boldsymbol{R}$ is $\boldsymbol{\frac{\varepsilon}{8\log_{1+\eta}^2(n)}}$ - LR** by privacy of **Geometric Mechanism** [Chan-Shi-Song '11; Balcer-Vadhan '18]

Privacy Proof

- Can be implemented via **local randomizers $\boldsymbol{R}$**
- R takes as **input $\boldsymbol{a}$ (adjacency list)** and **public set of active vertices $\boldsymbol{A}$**
  - R computes size of intersection $|\boldsymbol{a} \cap \boldsymbol{A}|$ **Sensitivity of 1**
  - Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log_{1+\eta}^{2}(n)}\right)$
- **$\boldsymbol{R}$ is $\boldsymbol{\frac{\varepsilon}{8\log_{1+\eta}^{2}(n)}}$ - LR** by privacy of **Geometric Mechanism** [Chan-Shi-Song ‘11; Balcer-Vadhan ‘18]
- Same LR called for all vertices $\boldsymbol{4\log_{1+\eta}^{2}(n)}$ times

Privacy Proof

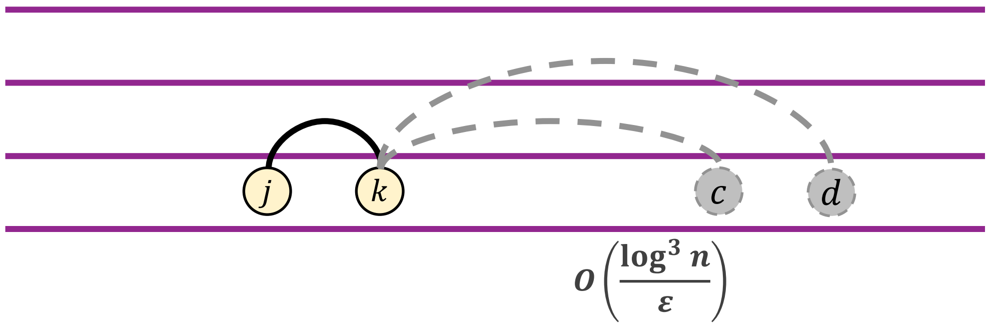
- Can be implemented via **local randomizers $\boldsymbol{R}$**
- R takes as **input $\boldsymbol{a}$ (adjacency list)** and **public set of active vertices $\boldsymbol{A}$**
  - R computes size of intersection $|\boldsymbol{a} \cap \boldsymbol{A}|$ **Sensitivity of 1**
  - Then, add symmetric geometric noise $X \sim Geom\left(\frac{\varepsilon}{8\log_{1+\eta}^2(n)}\right)$
- **$\boldsymbol{R}$ is $\frac{\boldsymbol{\varepsilon}}{\mathbf{8}\log_{\mathbf{1}+\boldsymbol{\eta}}^{\mathbf{2}}(\boldsymbol{n})}$ - LR** by privacy of **Geometric Mechanism** [Chan-Shi-Song '11; Balcer-Vadhan '18]
- Same LR called for all vertices $\mathbf{4}\log_{\mathbf{1}+\boldsymbol{\eta}}^{\mathbf{2}}(\boldsymbol{n})$ times
- For each edge, called $\mathbf{8}\log_{\mathbf{1}+\boldsymbol{\eta}}^{\mathbf{2}}(\boldsymbol{n})$; then, $\mathbf{8}\log_{\mathbf{1}+\boldsymbol{\eta}}^{\mathbf{2}}(\boldsymbol{n}) \cdot \frac{\boldsymbol{\varepsilon}}{\mathbf{8}\log_{\mathbf{1}+\boldsymbol{\eta}}^{\mathbf{2}}(\boldsymbol{n})} = \varepsilon$ and so ε-LEDP

Approximation Proof Sketch

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

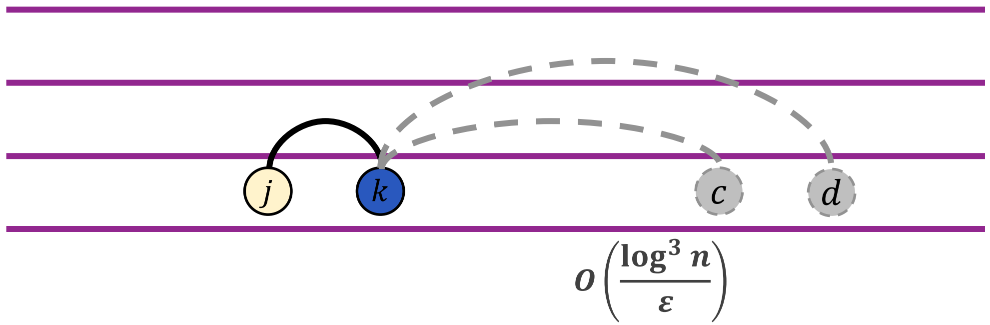
Approximation Proof Sketch

- 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 $\geq i$**



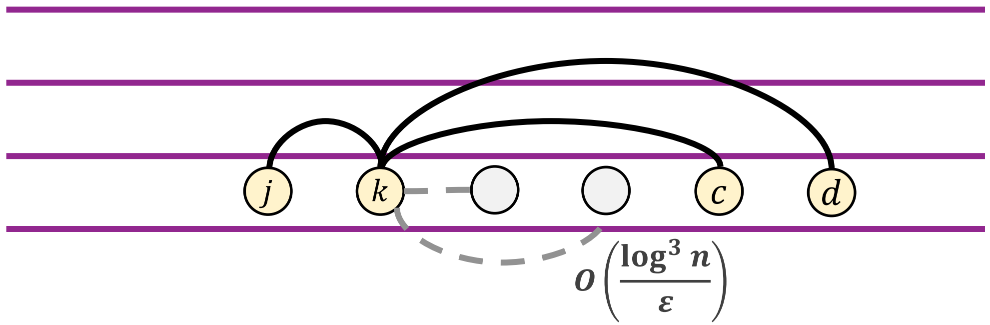
Approximation Proof Sketch

- 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 $\geq i$**



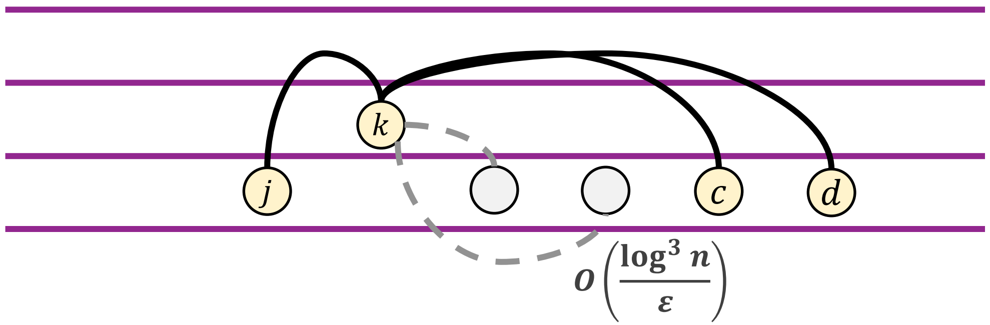
Approximation Proof Sketch

- 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 $\geq i - 1$**



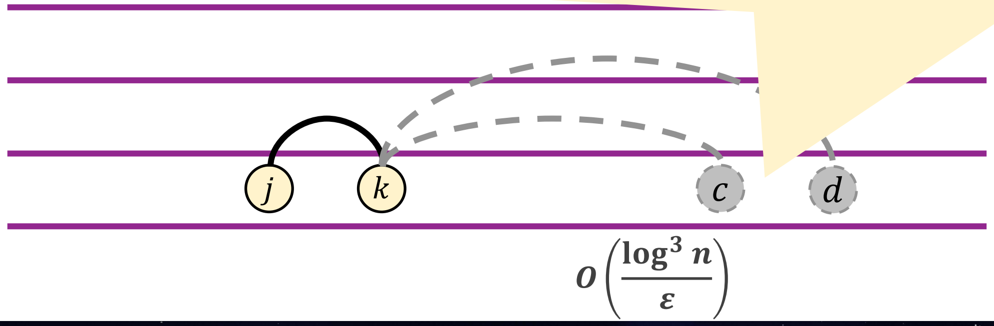
Approximation Proof Sketch

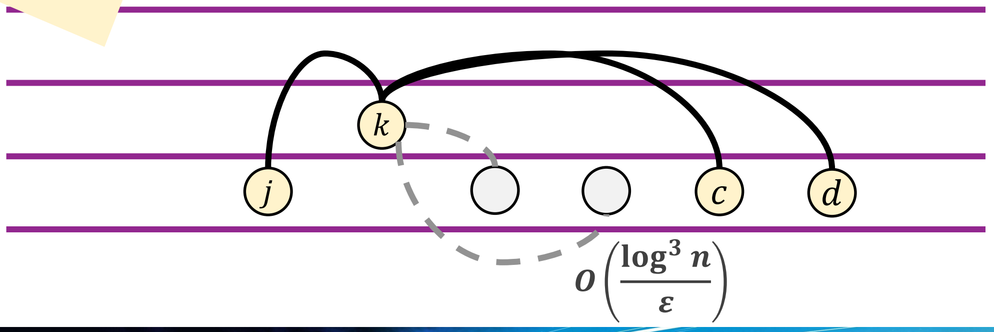
- With high probability, magnitude of each drawn noise is **upper bounded by** $\boldsymbol{O\left(\frac{\log^3 n}{\varepsilon}\right)}$
- **Degree Upper Bound:** If a vertex v is on level $i < 4\log_{1+\eta}(n)$ at end of algorithm, then it has **at most** $\boldsymbol{(1+\eta)^i + O\left(\frac{\log^3 n}{\varepsilon}\right)}$ **neighbors on levels** $\boldsymbol{\geq i}$
- **Degree Lower Bound:** If a vertex v is on level $i > 0$ at end of algorithm, then it has **at least** $\boldsymbol{(1+\eta)^i - O\left(\frac{\log^3 n}{\varepsilon}\right)}$ **neighbors on levels** $\boldsymbol{\geq i - 1}$



Approximation Proof Sketch

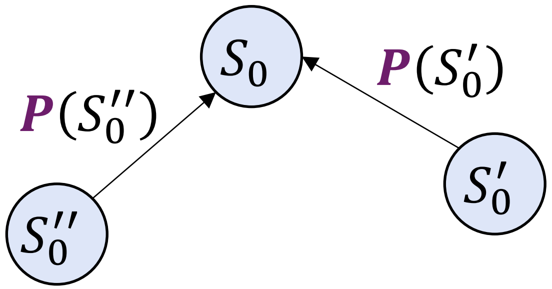
Key: Largest cutoff increases/decreases by **additive $O\left(\frac{\log^3 n}{\varepsilon}\right)$**





Locally Adjustable Graph Algorithms

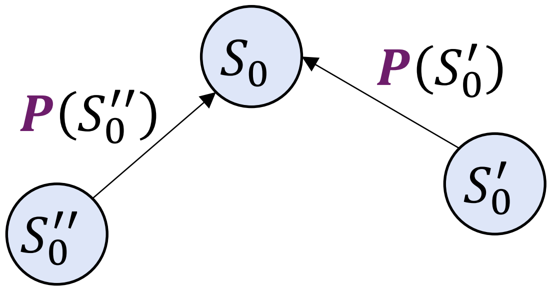
- **Intuition:** Each node's current state depends on **number of neighbors** whose **previous state satisfies predicate $\boldsymbol{P}$**



Locally Adjustable Graph Algorithms

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

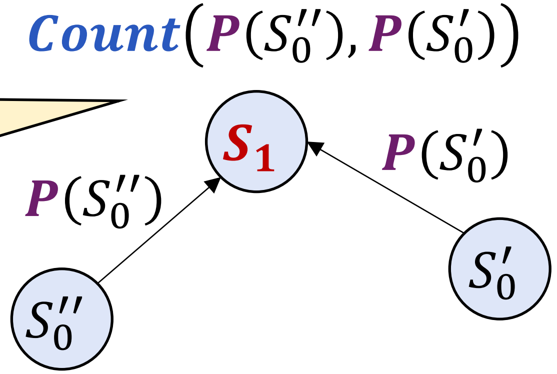
$$Count\left(P(S_0''), P(S_0')\right)$$



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

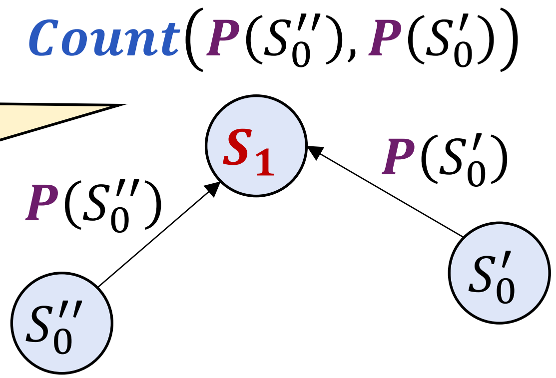
Sensitivity of **Count** of number of neighbors satisfying P is **1**



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

Sensitivity of **Count** of number of neighbors satisfying P is **1**

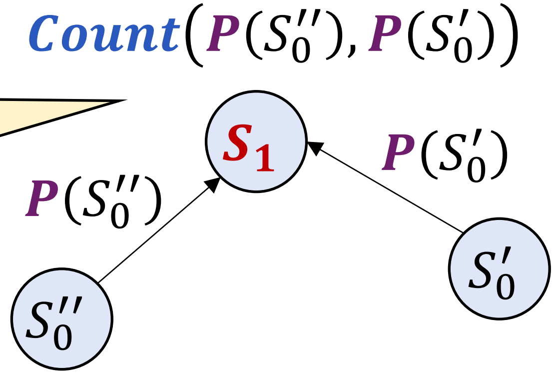


Then, **just add geometric noise** to the counts!

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

Many **distributed/parallel graph algorithms** use **small rounds/depth** and may fall under framework

Sensitivity of **Count** of number of neighbors satisfying P is **1**



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

Sample from $\boldsymbol{Geom}\left(\frac{\varepsilon}{2 \cdot \boldsymbol{rounds}}\right)$ where $\boldsymbol{rounds}$ 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)

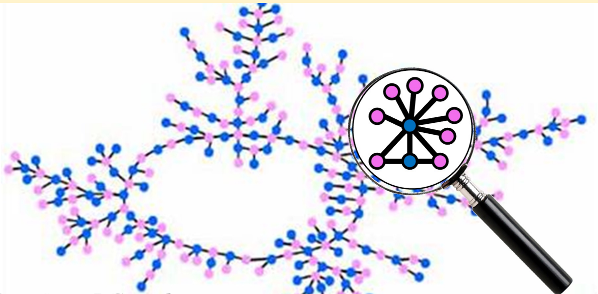
Additional Open Questions

- 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)

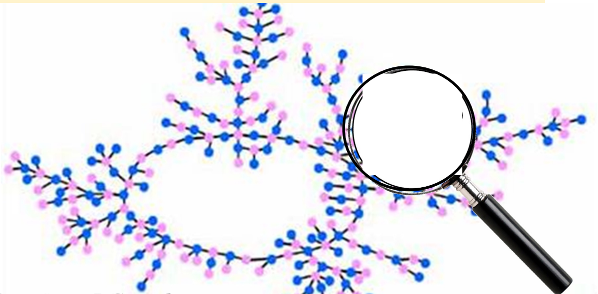
Additional Open Questions

- 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**



American J. Sociology, Bearman, Moody, Stovel



American J. Sociology, Bearman, Moody, Stovel