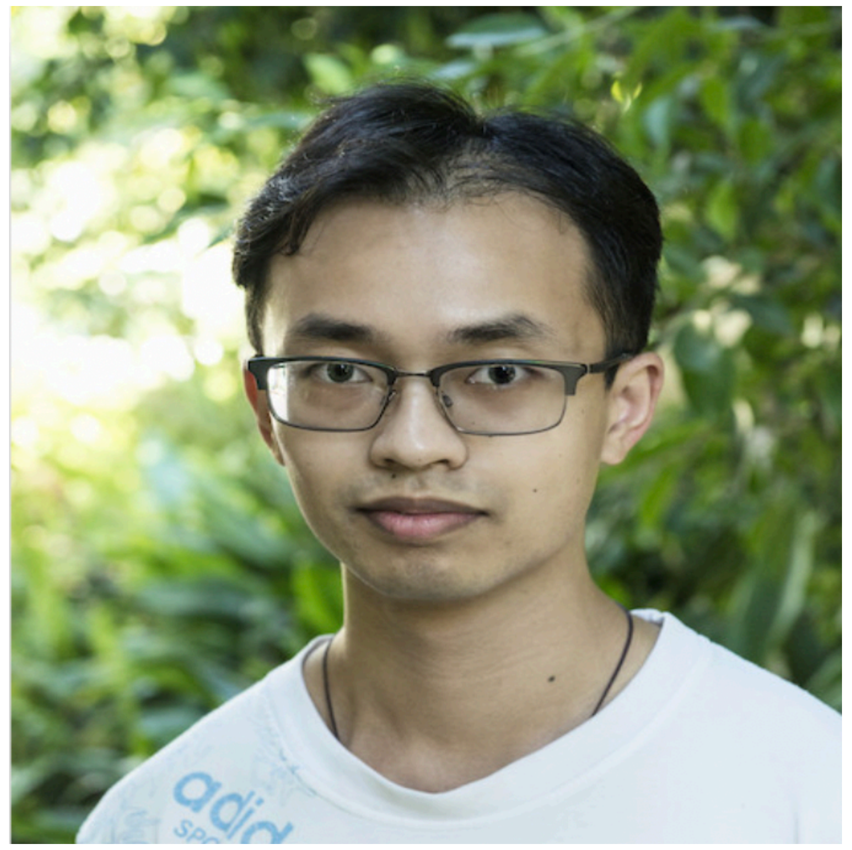
Almost linear time private release of synthetic graphs

Zongrui Zou
Nanjing University

Joint work with

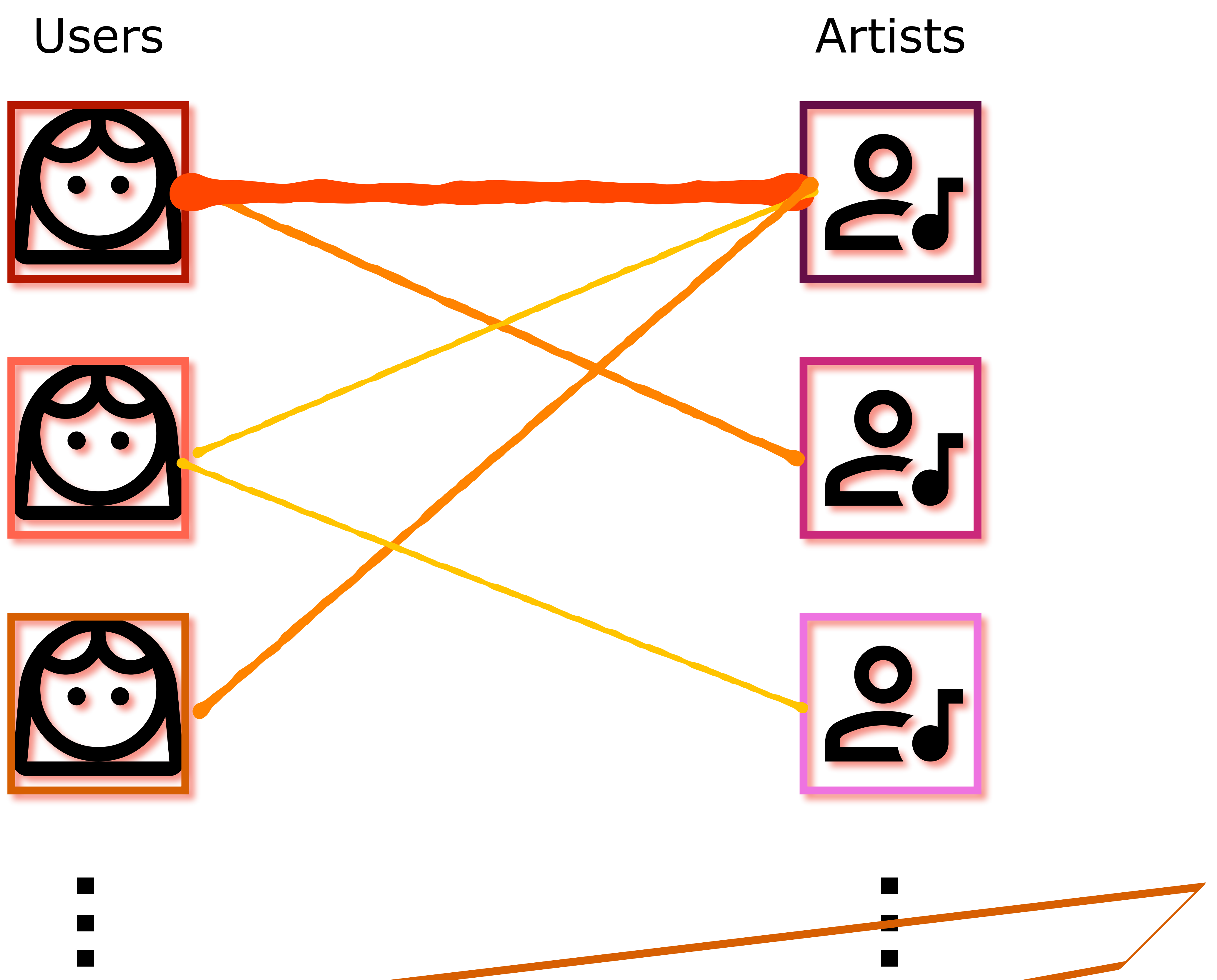


Jingcheng Liu
(Nanjing University)

Jalaj Upadhyay
(Rutgers)

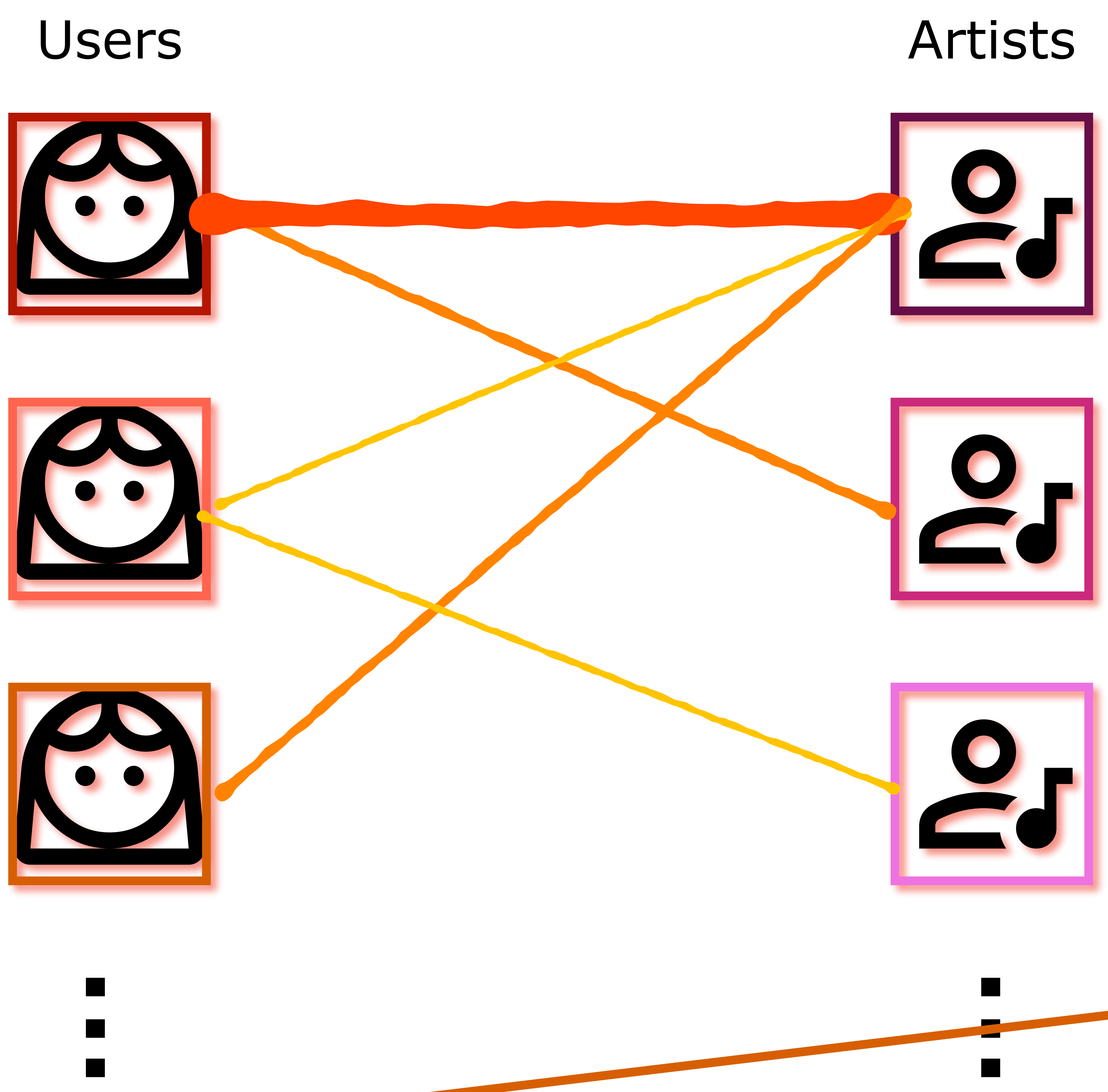
Artificial Intelligence and Statistics 2025

Sensitive data encoded by weighted graphs

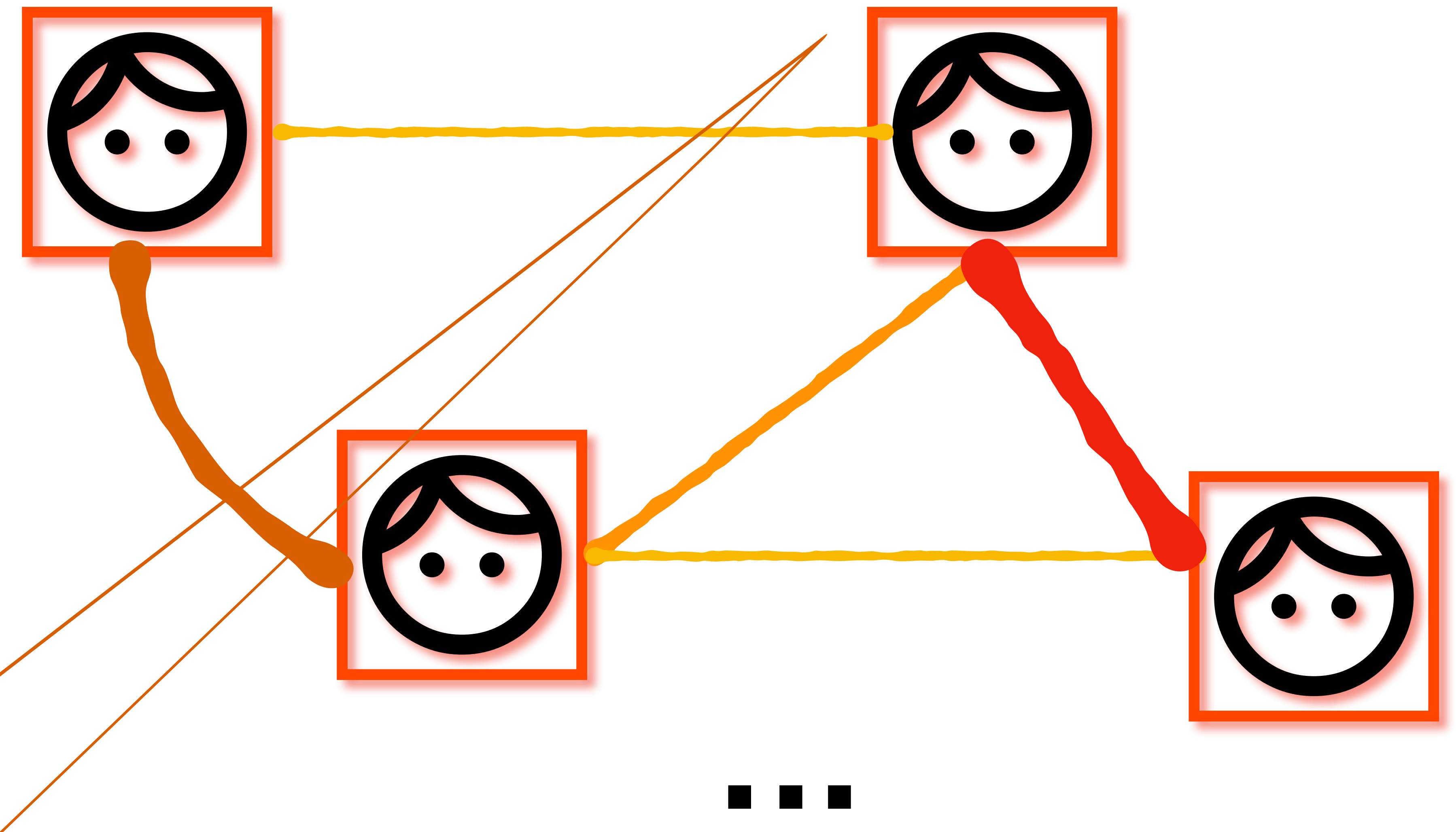


The bipartite graph for user preferences in some music app
(weighted edges represent "preference")

Sensitive data encoded by **weighted** graphs

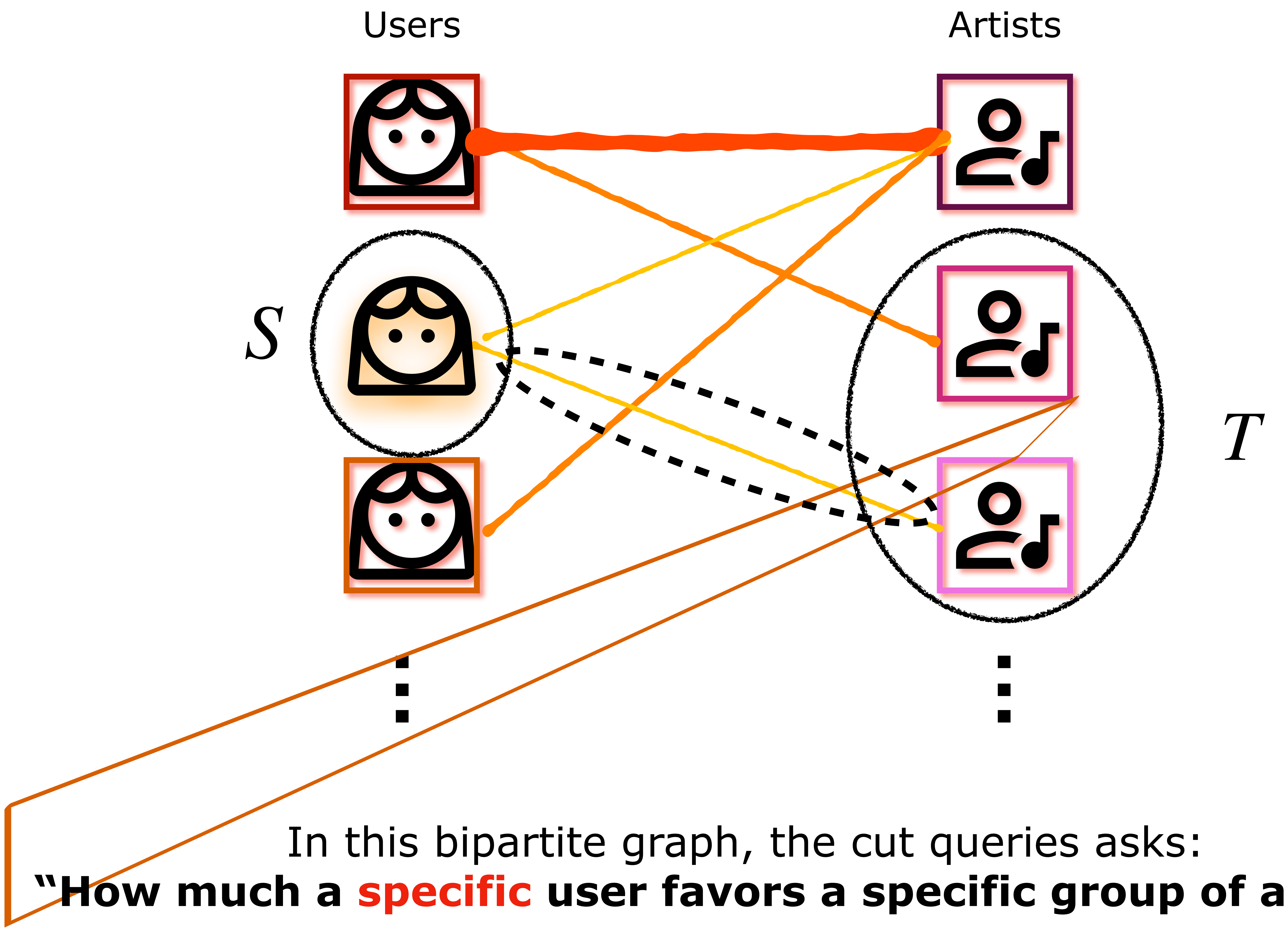


The bipartite graph for user preferences in some music APP
(weighted edges represent “preference”)



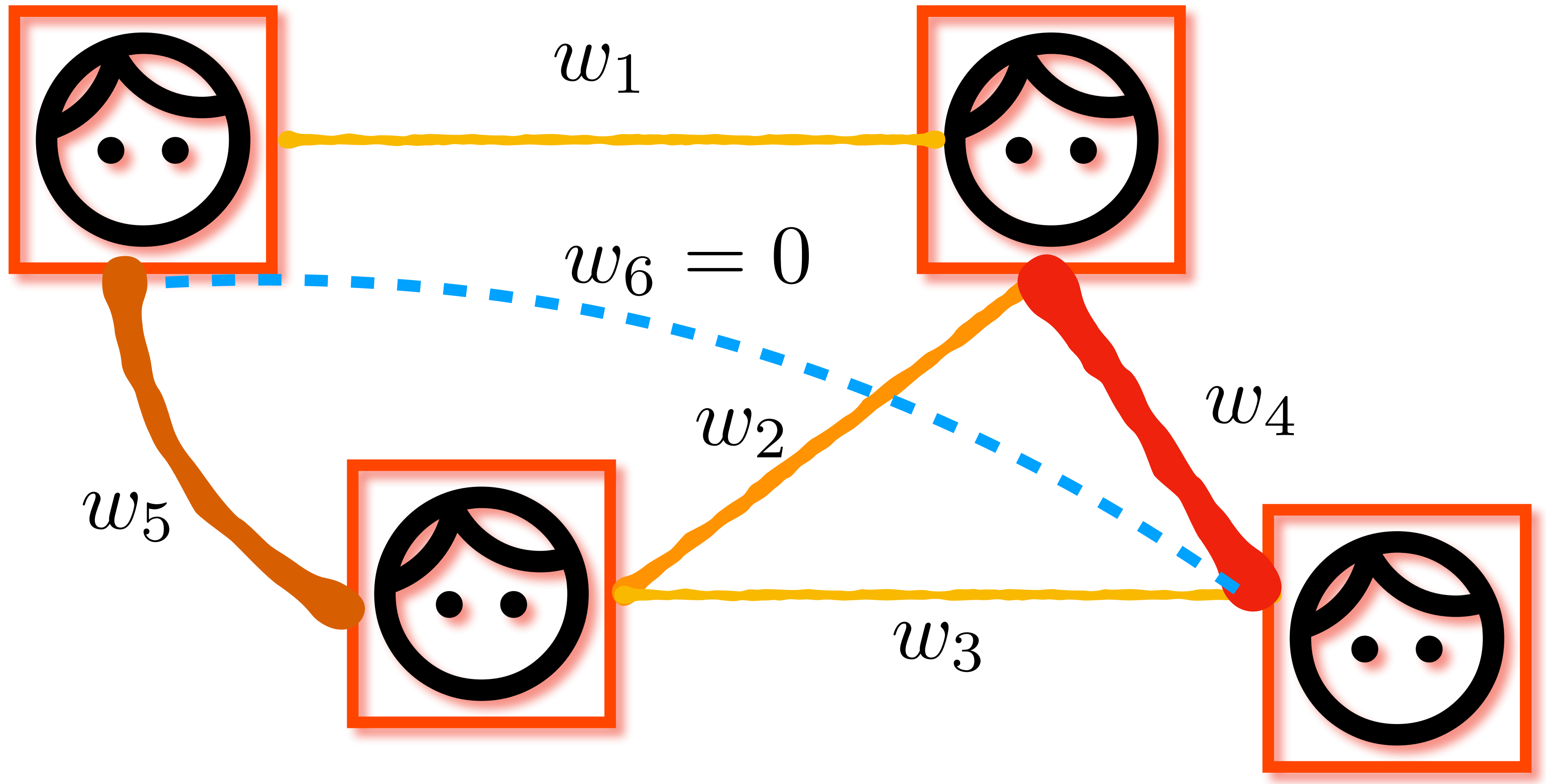
Typical social networks
(weighted edges represent “interaction frequency”)

Cut queries in graph data



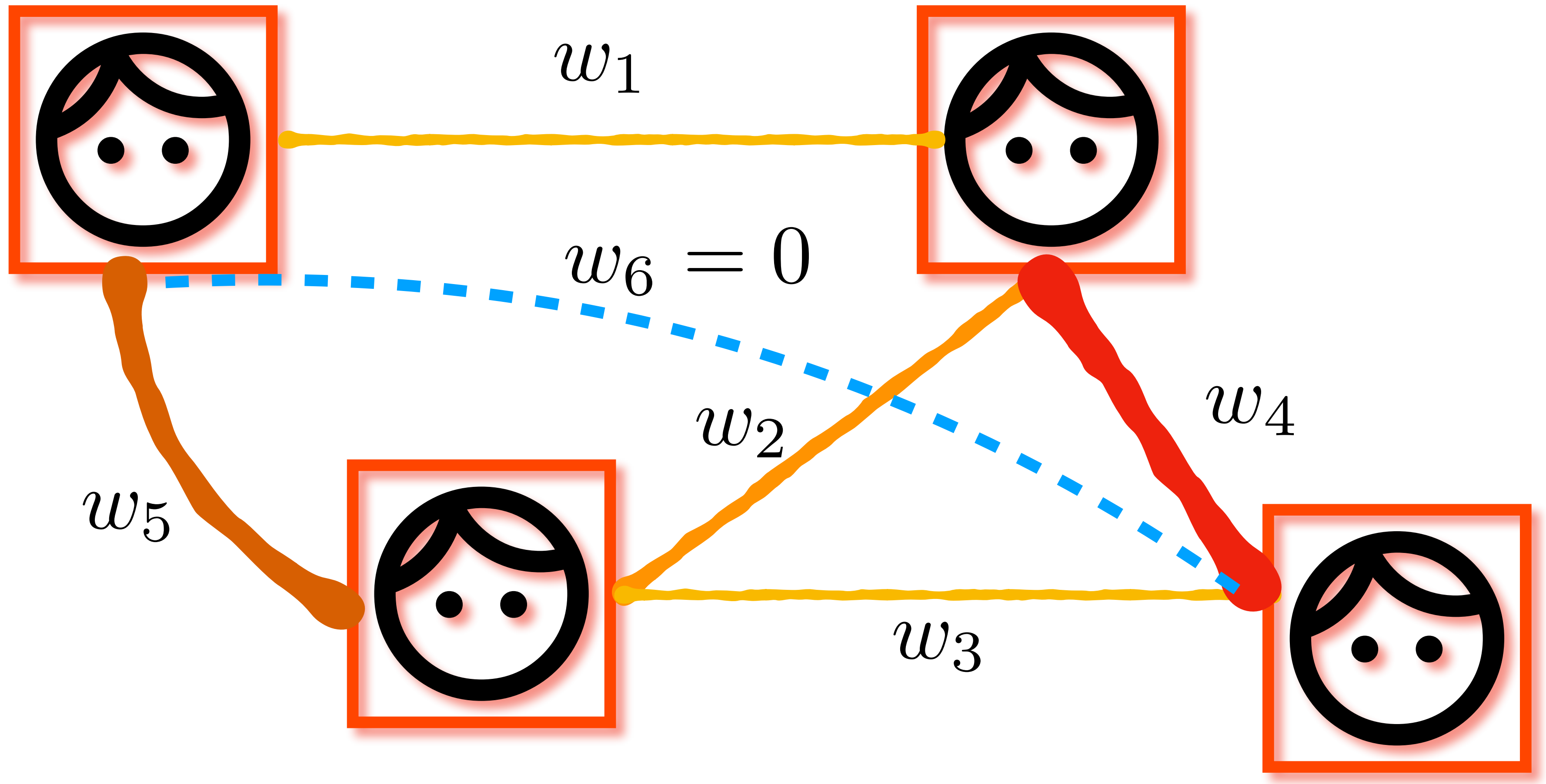
In this bipartite graph, the cut queries asks:
"How much a specific user favors a specific group of artists?"

Private synthetic graph

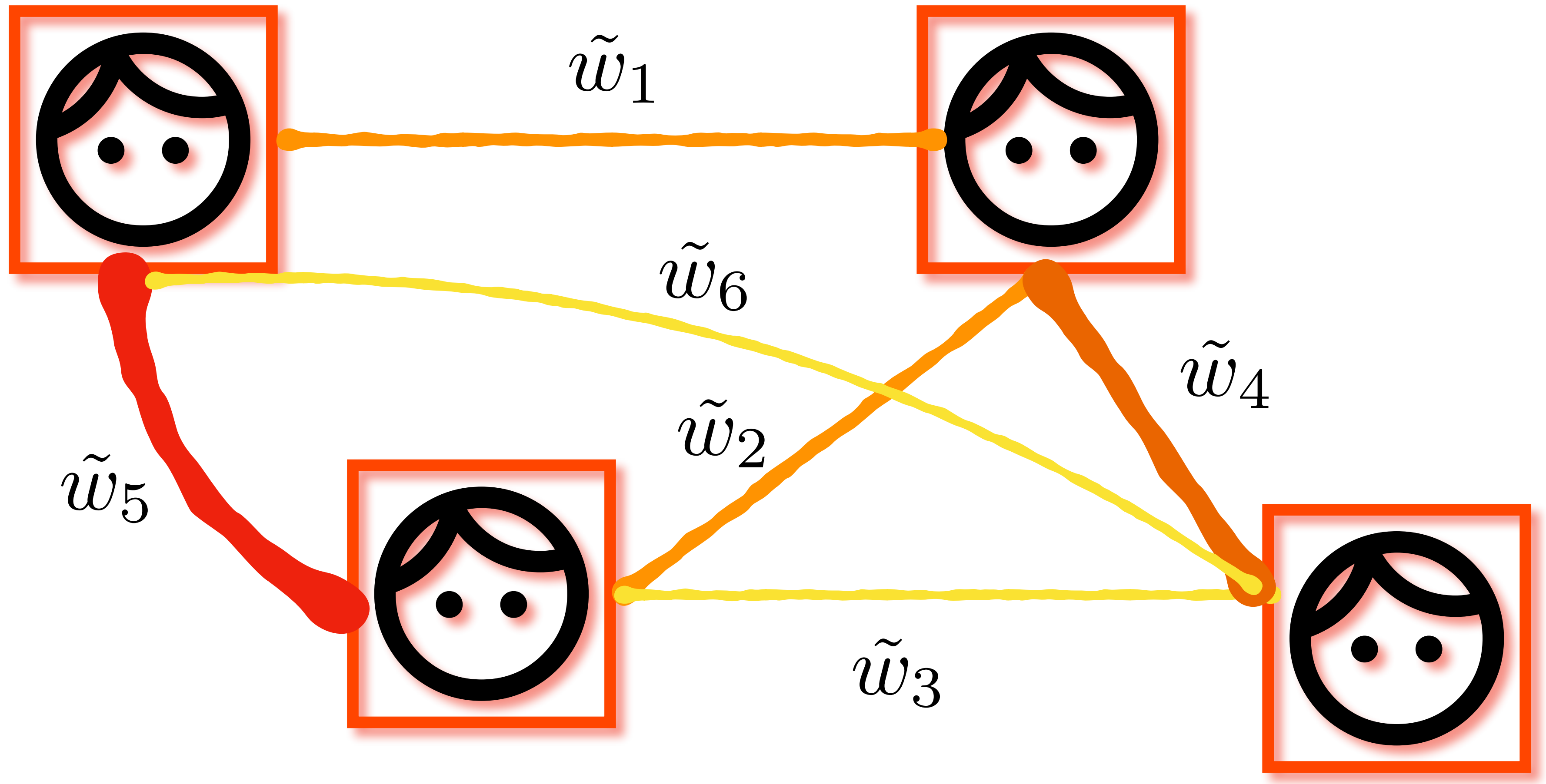


Undirected graph $G = (V, E, w)$
n vertices, m edges
Unweighted maximum degree Δ

Private synthetic graph



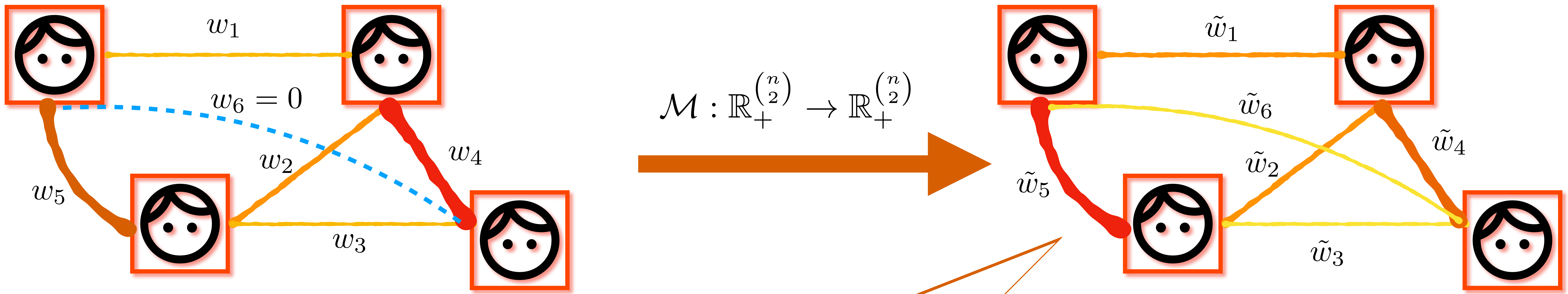
$\mathcal{M} : \mathbb{R}_+^{\binom{n}{2}} \to \mathbb{R}_+^{\binom{n}{2}}$



Undirected graph $G = (V, E, w)$
n vertices, m edges
Unweighted maximum degree Δ

A synthetic graph $G' = (V, E', w')$
(does not necessarily have same topology)

Private synthetic graph



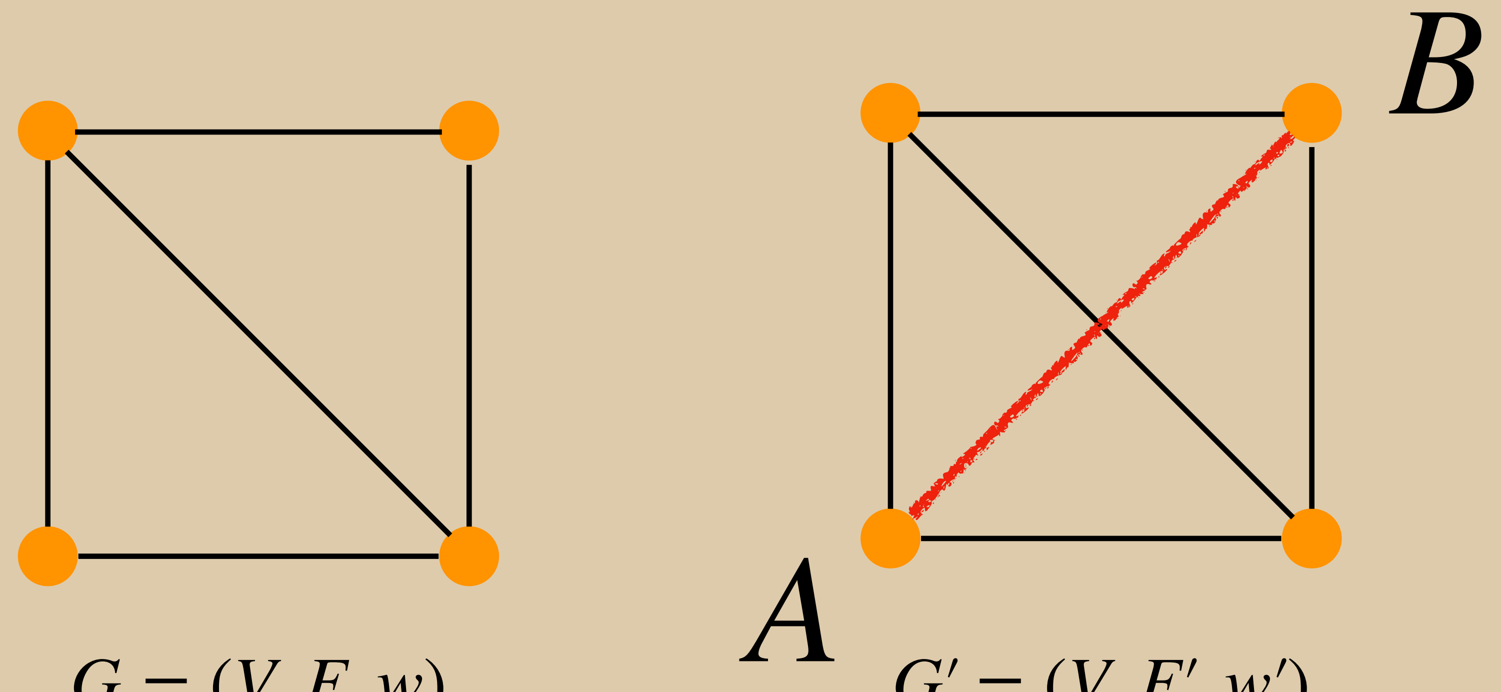
Undirected graph $G = (V, E, w)$
n vertices, m edges
Unweighted maximum degree Δ

A synthetic graph $G' = (V, E', w')$
(does not necessarily have same topology)

- **Privacy**: $\mathcal{M}$ should be differentially private.
- **Utility**: G' maintains certain algebraic (i.e., spectrum) and combinatorial properties (i.e., cut function) of G.

Graph differential privacy

Neighboring graphs



$G = (V, E, w)$ $\quad$ $G' = (V, E', w')$

G and G' are neighboring if and only if
$\|w - w'\|_0 \leq 1$ and $\|w - w'\|_\infty \leq 1$.

- $w \in \mathbb{R}^{\binom{n}{2}}$ encodes the edge weights.

The goal is to make any pair of neighboring datasets **indistinguishable** from reading their private copies.

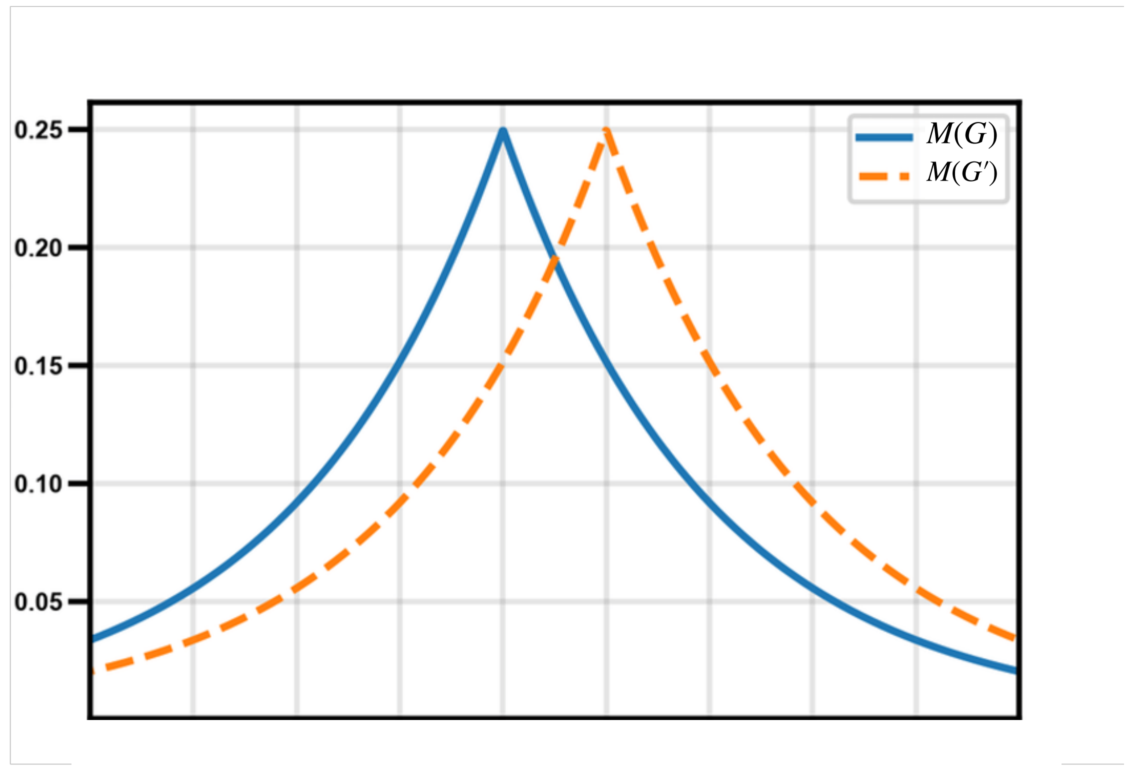
Differential privacy

A randomized mechanism M outputting a synthetic graph is (ε, δ)-differentially private if for any pair of neighboring graphs G, G' and any subset $S \subseteq \mathbb{R}^{\binom{n}{2}}$,

$$\Pr[M(G) \in S] \leq e^{\varepsilon} \cdot \Pr[M(G') \in S] + \delta.$$

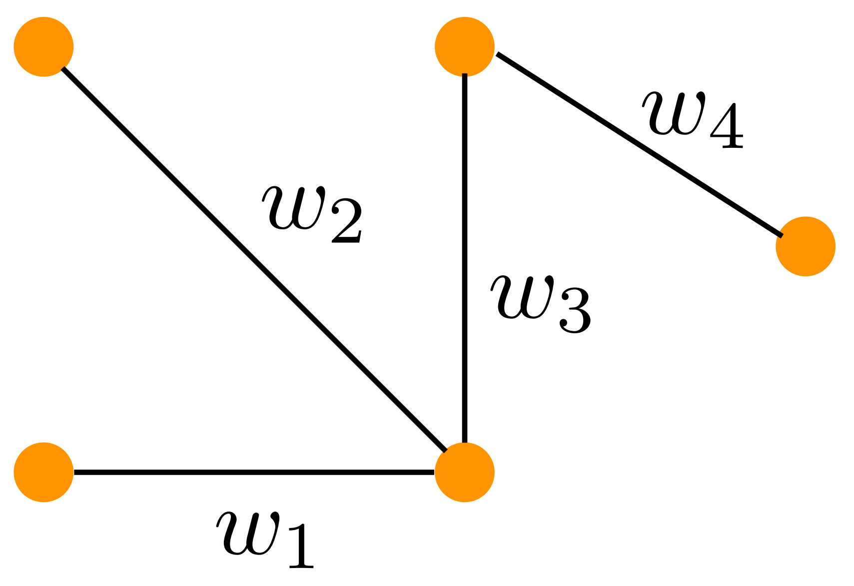
- Here, $\varepsilon > 0$ and $0 \leq \delta \leq 1$;
- If $\delta = 0$, the mechanism preserves **pure** differential privacy.
- Unless specified, we set $\varepsilon = O(1)$ and $\delta = 1/n^c$.

PDF



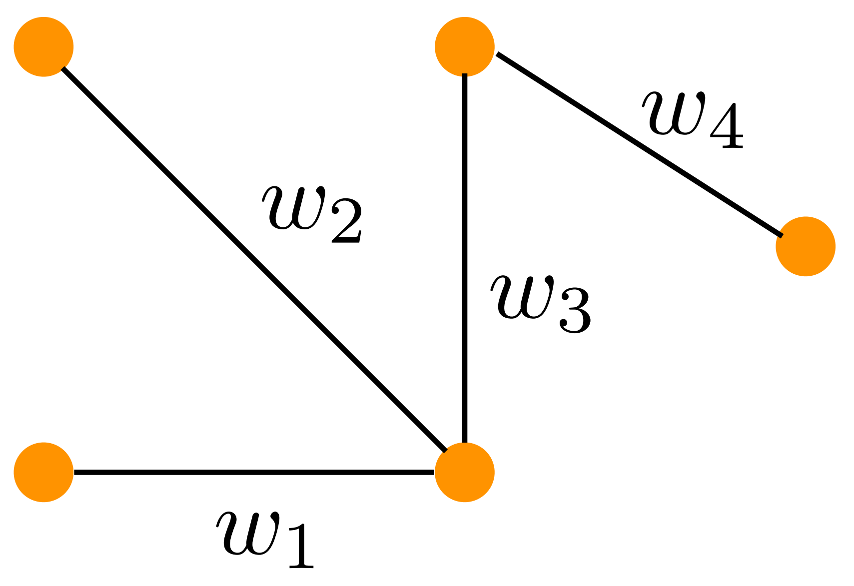
Possible outputs

The classical approach

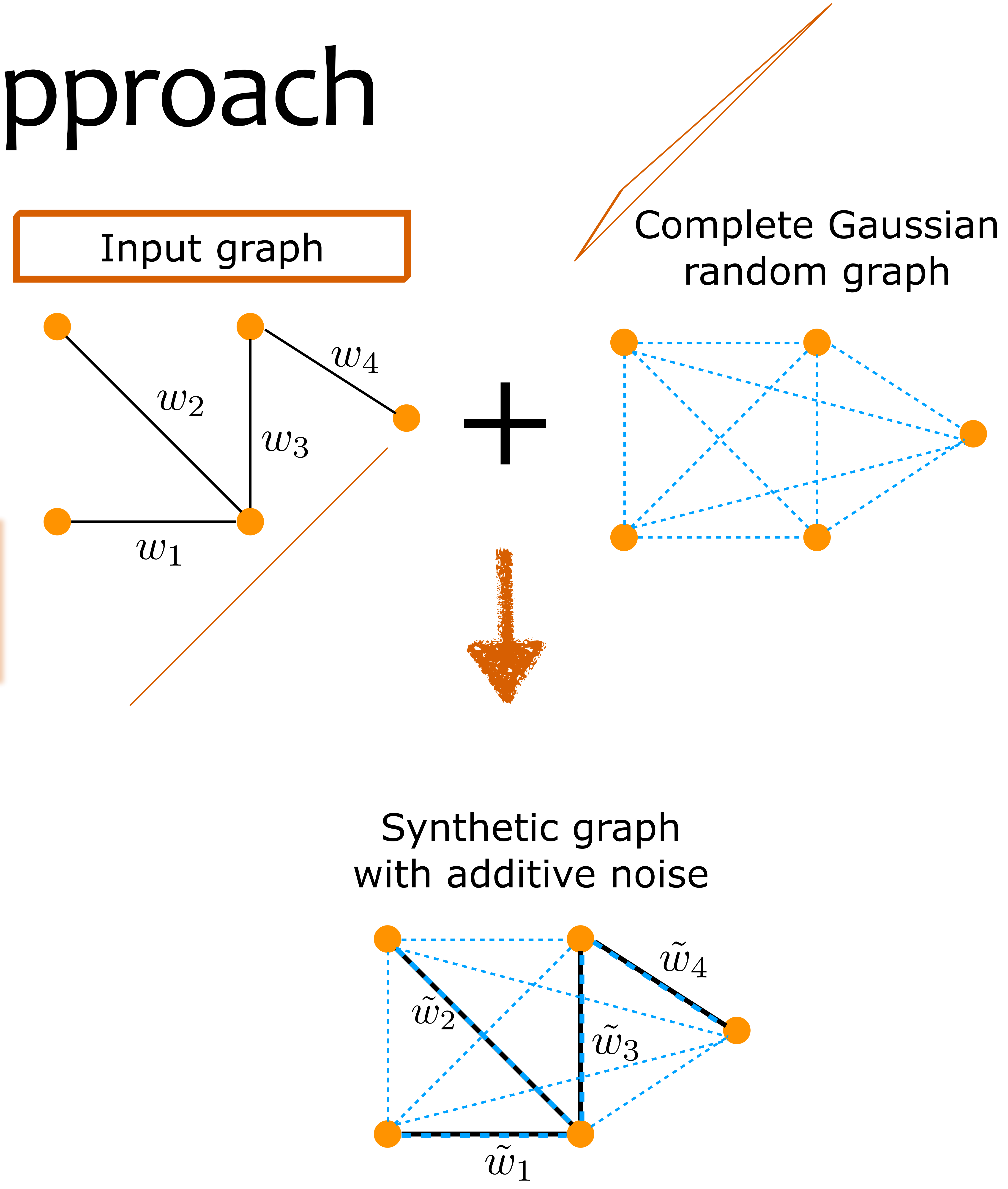


For an undirected graph $G = ([n], E, w)$, a neighboring graph could differ in any of the $\binom{n}{2}$ pair of vertices.

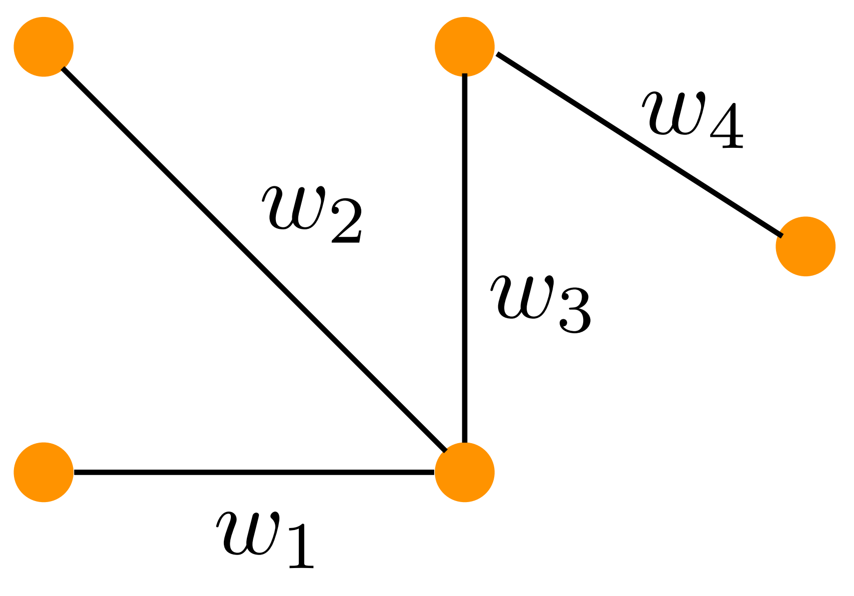
The classical approach



For an undirected graph $G = ([n], E, w)$, a neighboring graph could differ in any of the $\binom{n}{2}$ pair of vertices.

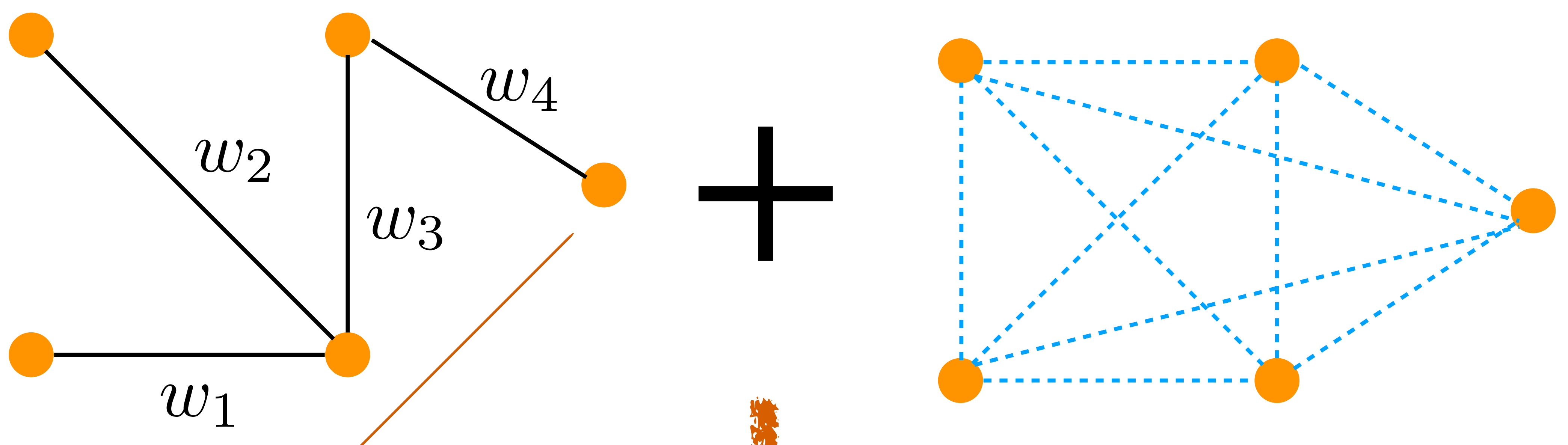


The classical approach



Input graph

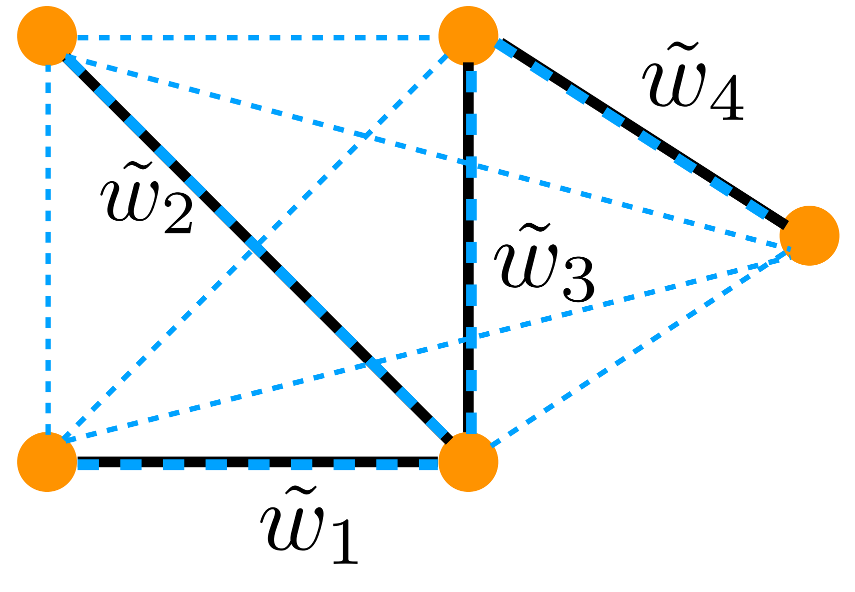
Complete Gaussian random graph



For an undirected graph $G = ([n], E, w)$, a neighboring graph could differ in any of the $\binom{n}{2}$ pair of vertices.

- $\tilde{O}(n^{1.5})$ error for cut is **inevitable** ([Eliáš, Kapralov, Kulkarni and Lee 2020]) even for **sparse graphs**.
- Running time: $O(n^2)$
- The output is **dense** no matter the sparsity of the input.

Synthetic graph with additive noise



[Liu, Upadhyay, **Zou** 2024]

Private Topology Selection

[EKKL20]

Mirror descent

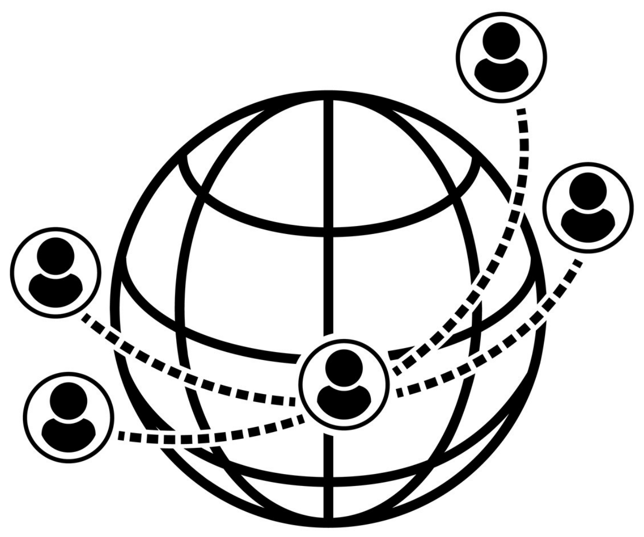
$O(n^7)$ running time

Instance optimal error:
$\tilde{\Theta}\left(\sqrt{mn}\right)$

The real-world graphs are usually sparse

◆ **Facebook messenger** is another undirected graph with about 3×10^9 **users** in 2022 and a total about 5×10^{12} **messages exchanged** in the year 2022.

◆ **Chase Bank** has approximately **18 million accounts** and 16,000 ATMs, while the total number of ATM transactions done in 2021 is about **600 million**.



◆ **Internet Activity Graph** currently has **4.3 billion** active IP address, which is a typical sparse graph because the number of connections between nodes (websites, servers, etc.) is much smaller than the number of possible connections.

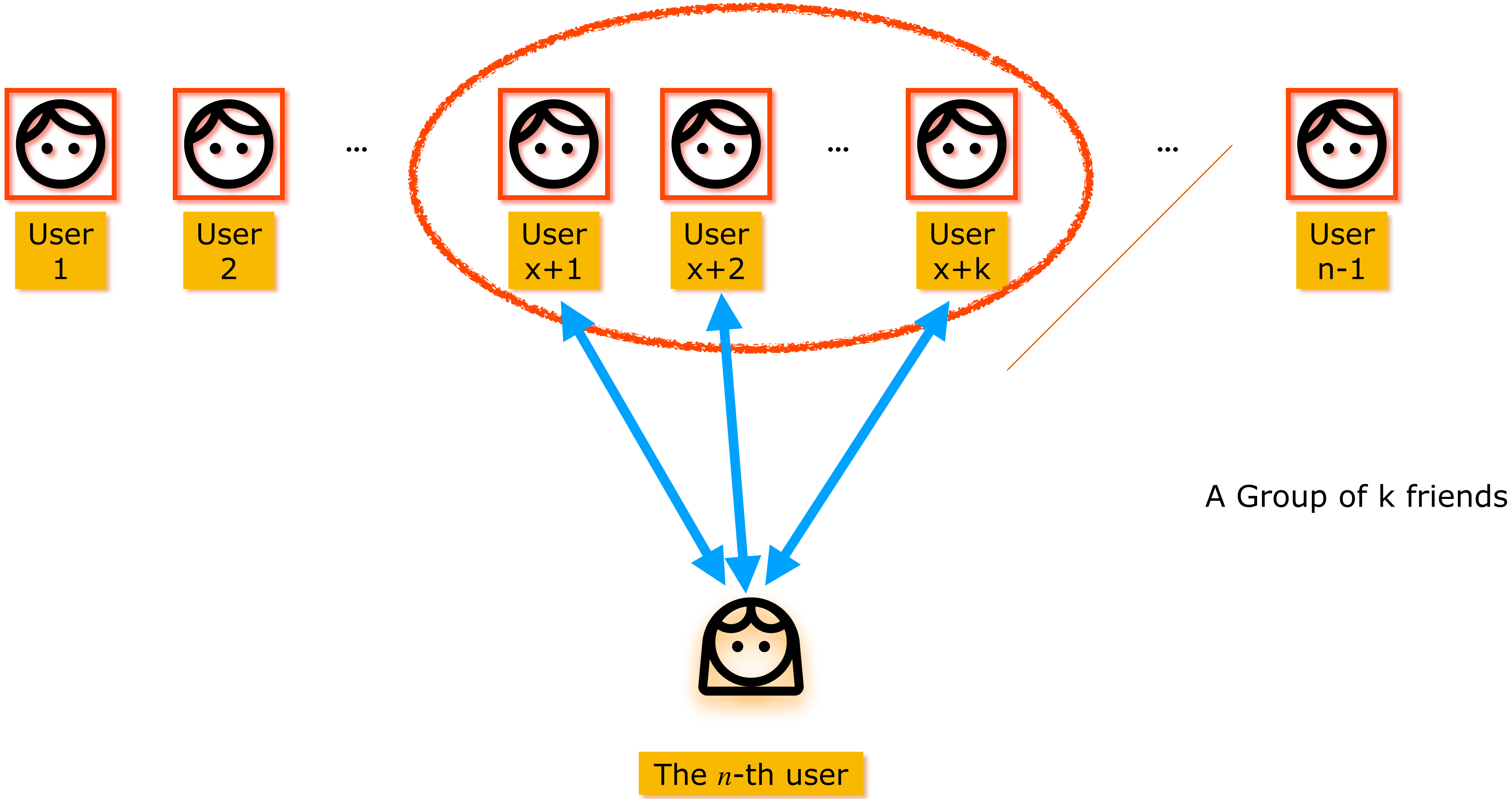
Efficient and private graph release?

Question: Is it possible to release a **useful** and **private** graph such that:

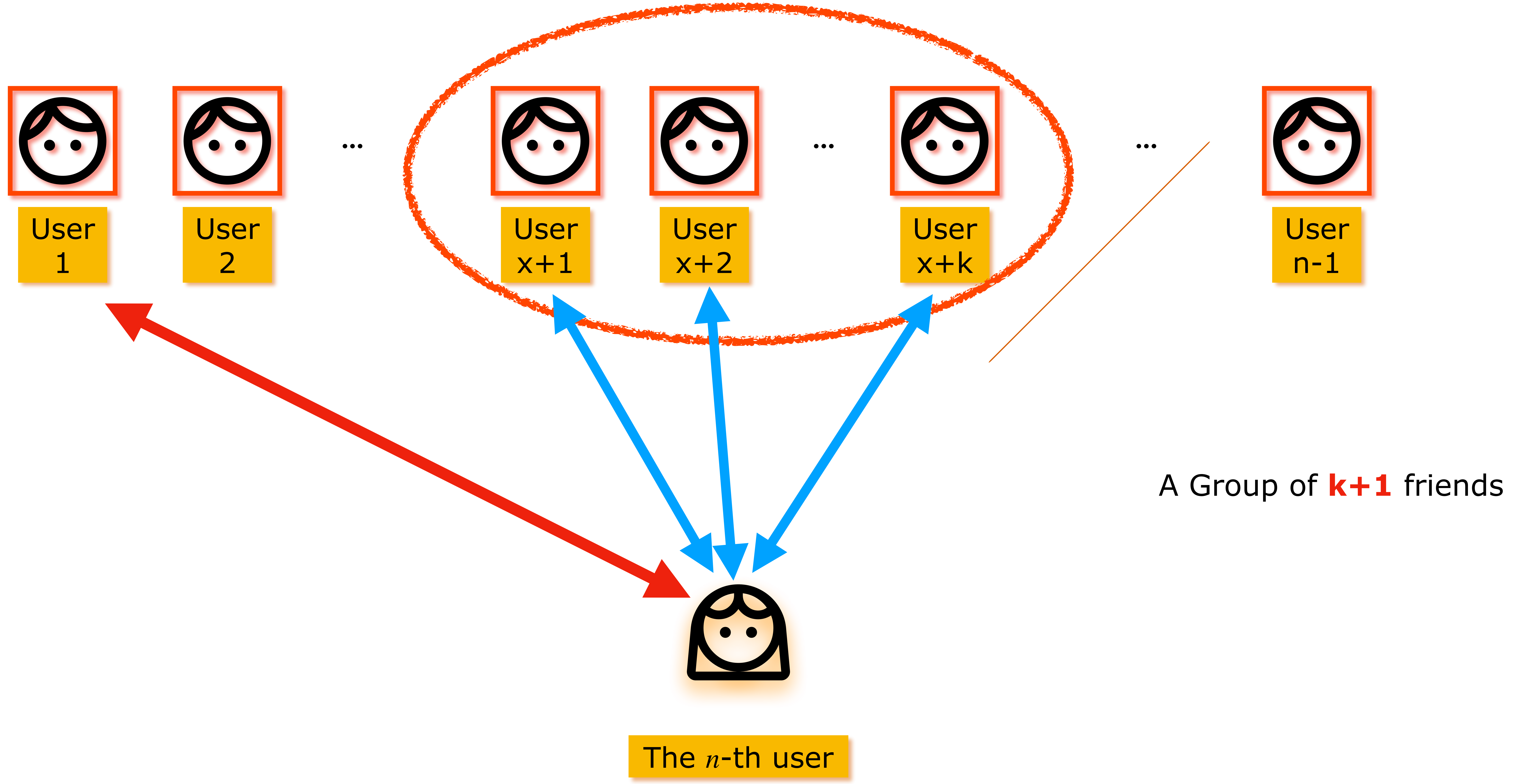
(i) The computation time & space required is comparable to the non-private setting;

(ii) The output graph is still sparse if the input graph is sparse.

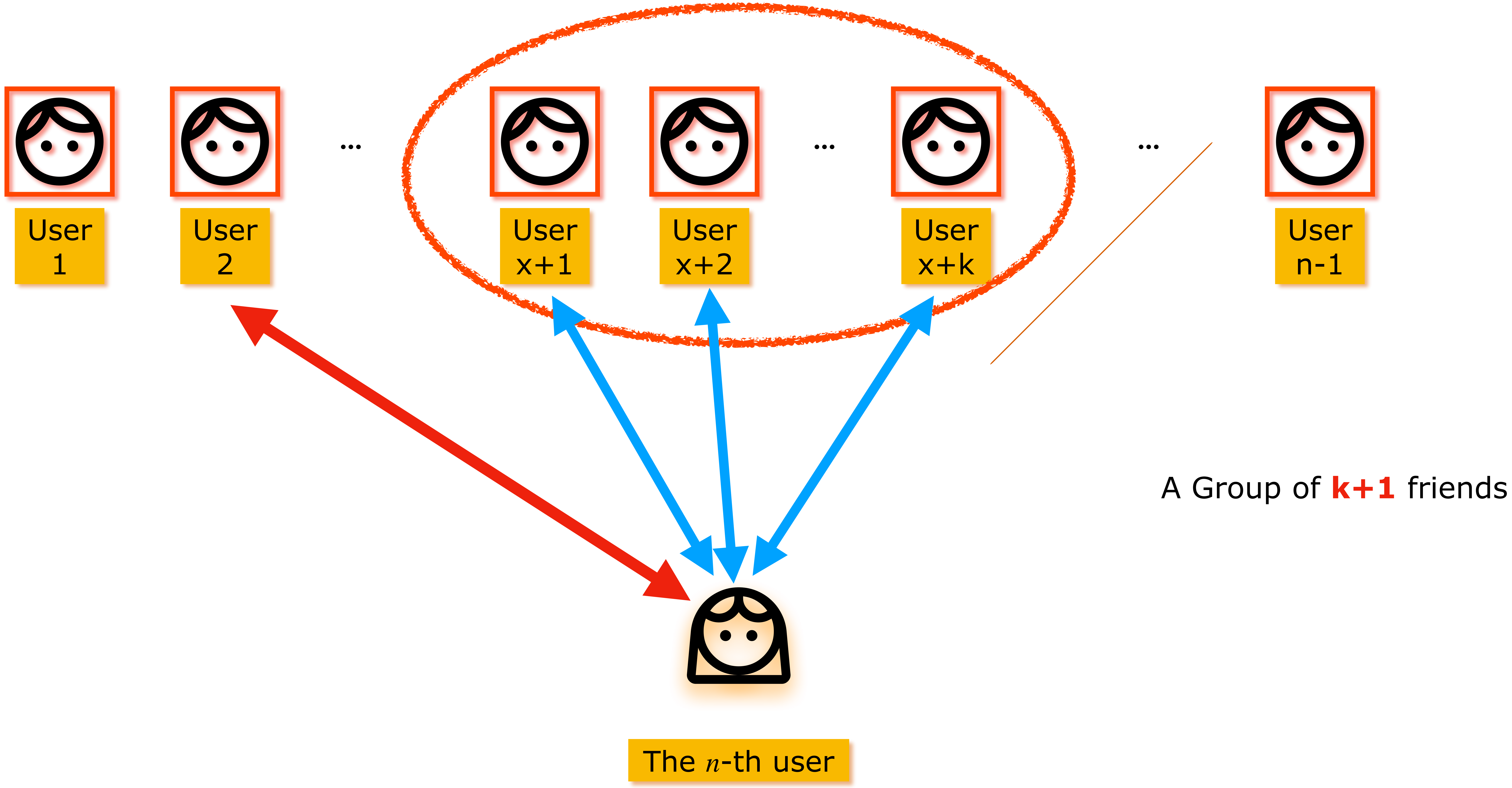
Beyond additive noise mechanism?



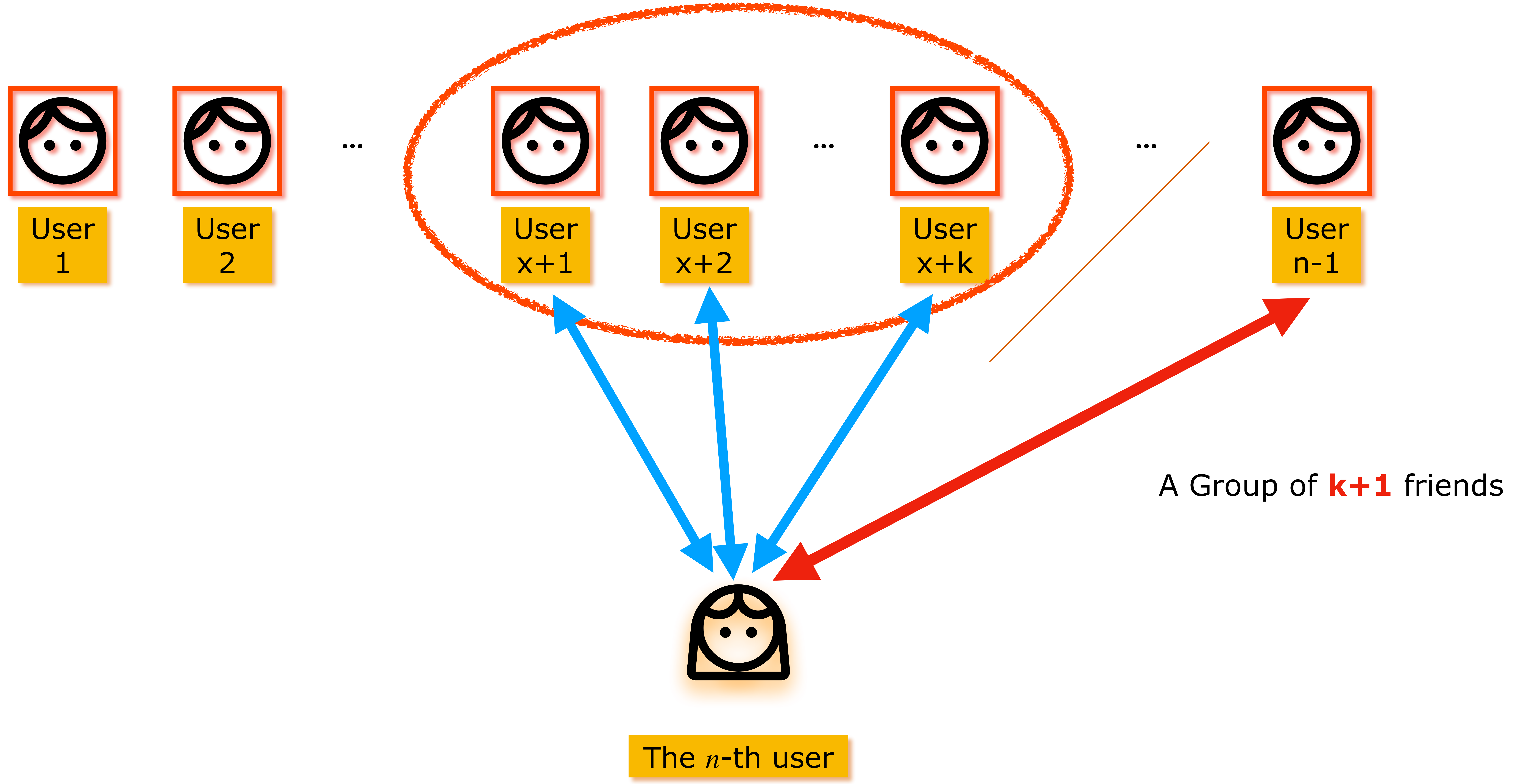
Beyond additive noise mechanism?



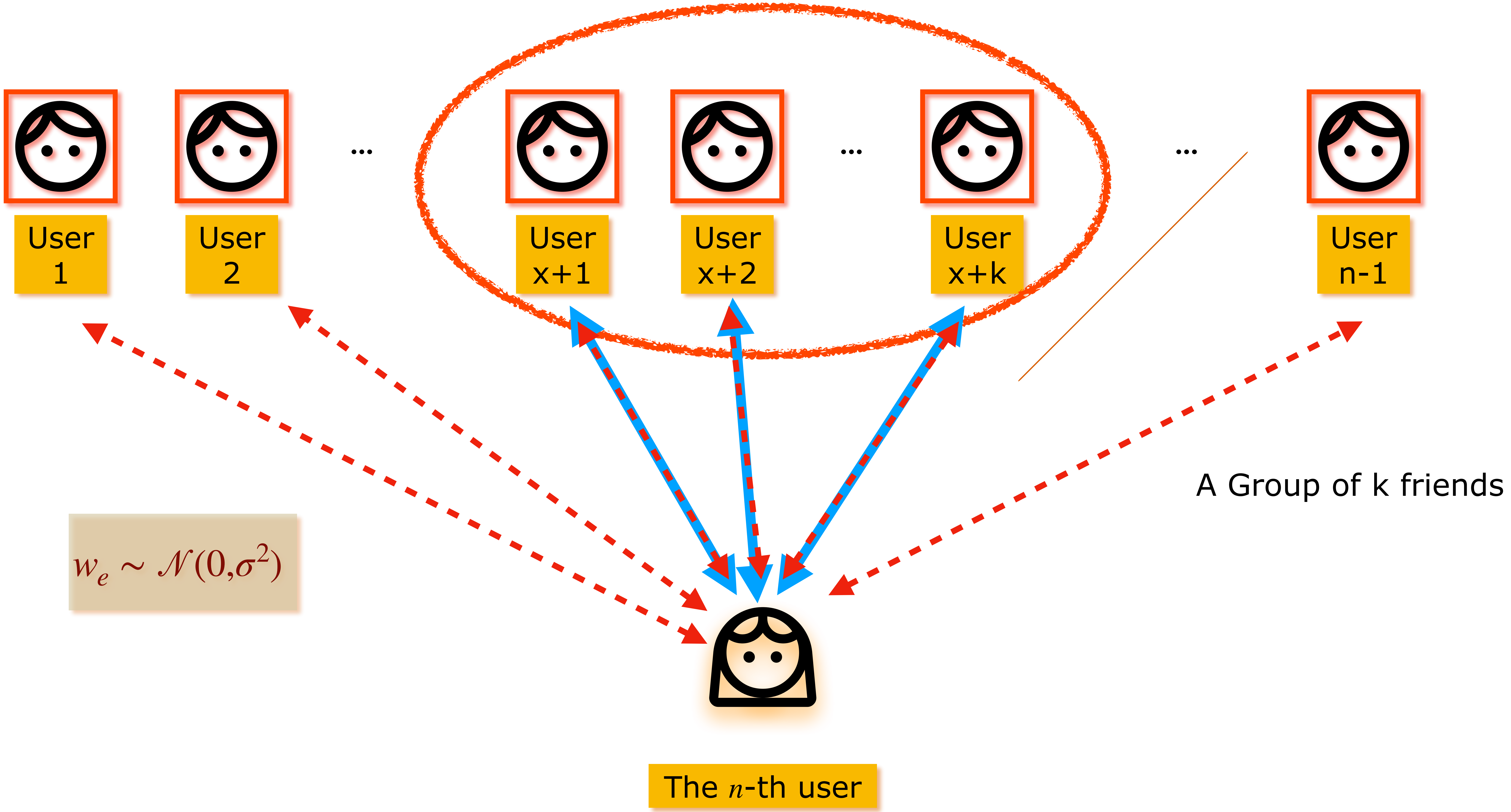
Beyond additive noise mechanism?



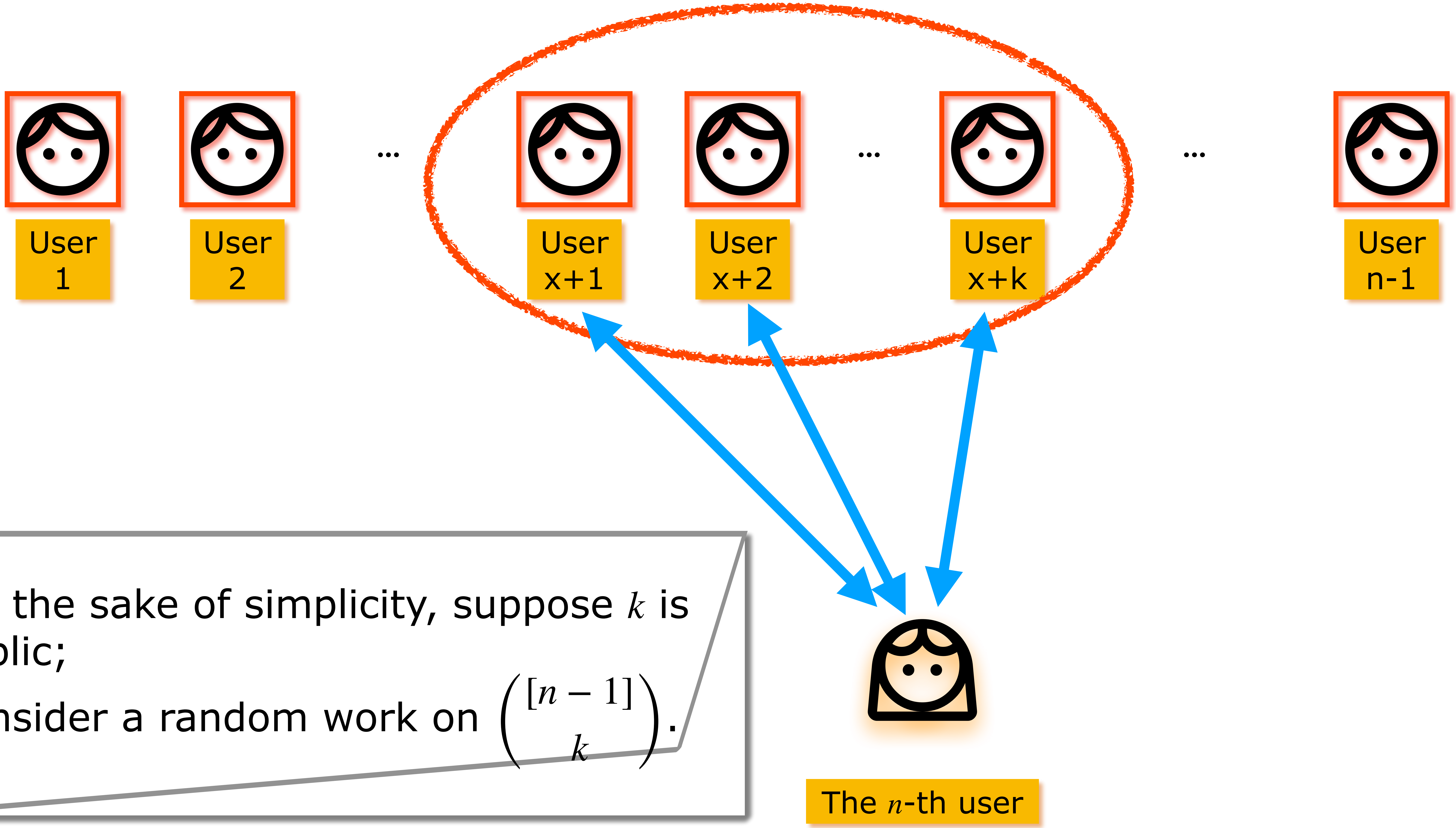
Beyond additive noise mechanism?



Beyond additive noise mechanism?

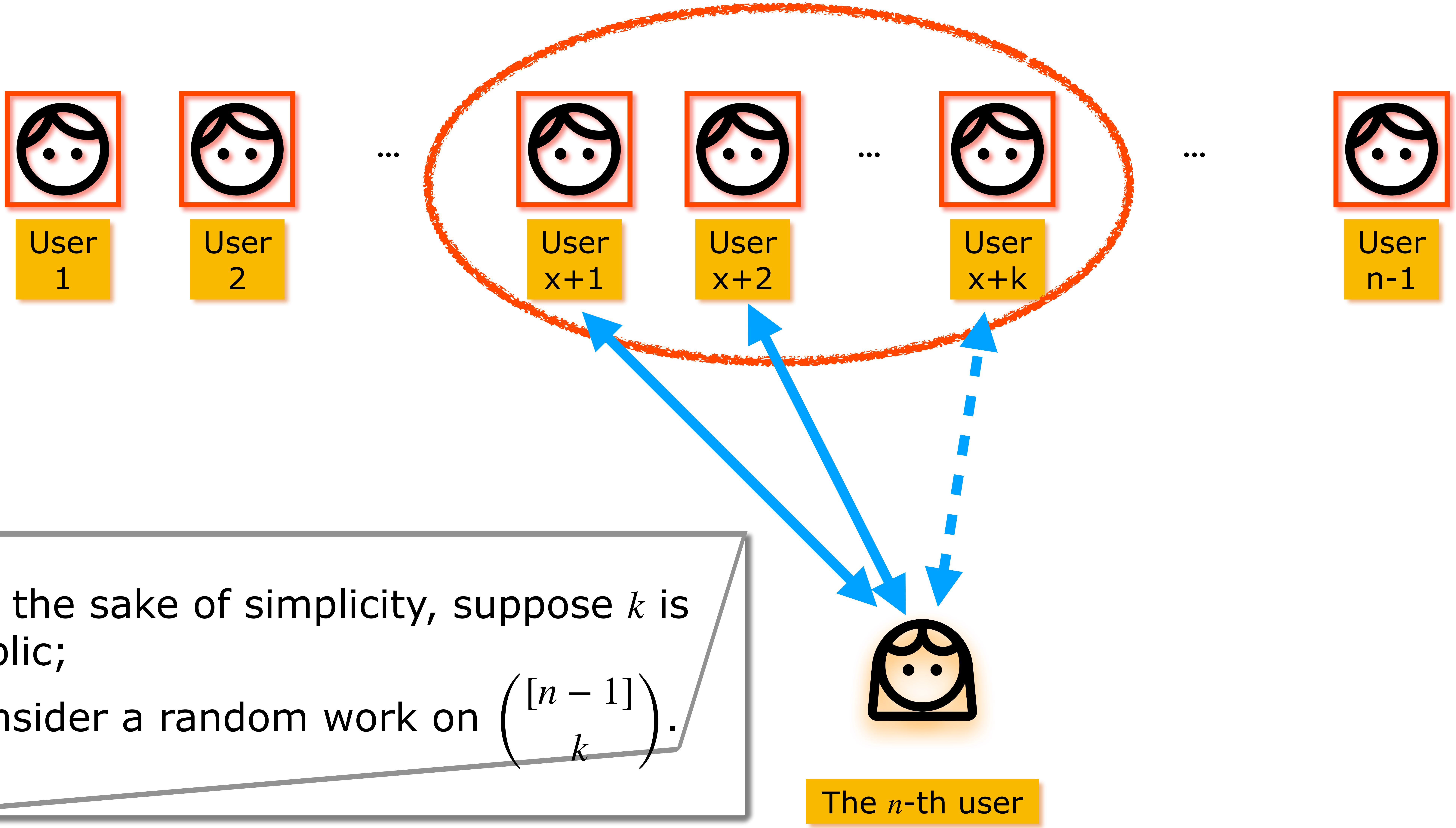


Beyond additive noise mechanism — perturbing edges by random walk



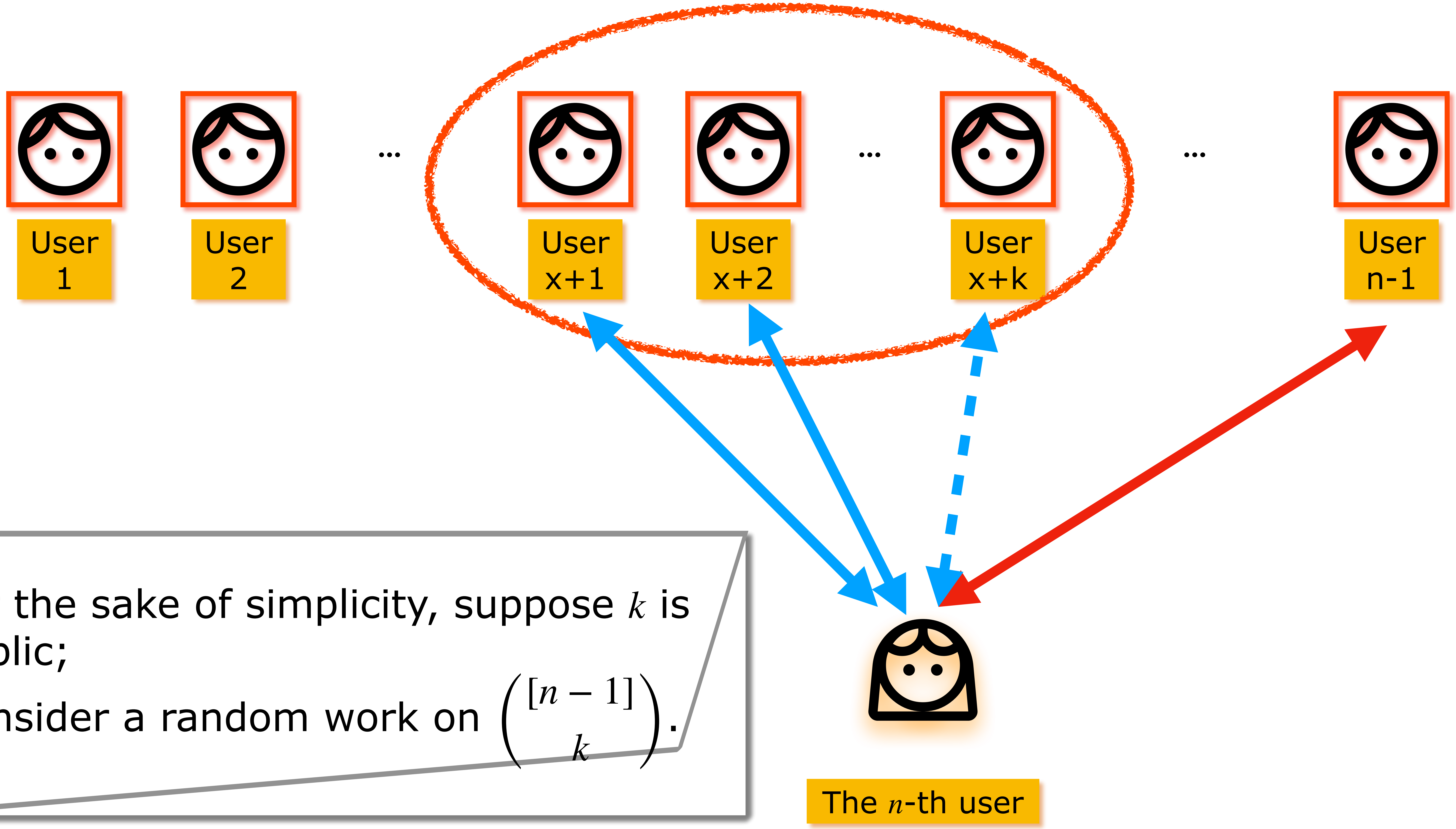
- For the sake of simplicity, suppose k is public;
- Consider a random work on $\binom{[n-1]}{k}$.

Beyond additive noise mechanism — perturbing edges by random walk



- For the sake of simplicity, suppose k is public;
- Consider a random work on $\binom{[n-1]}{k}$.

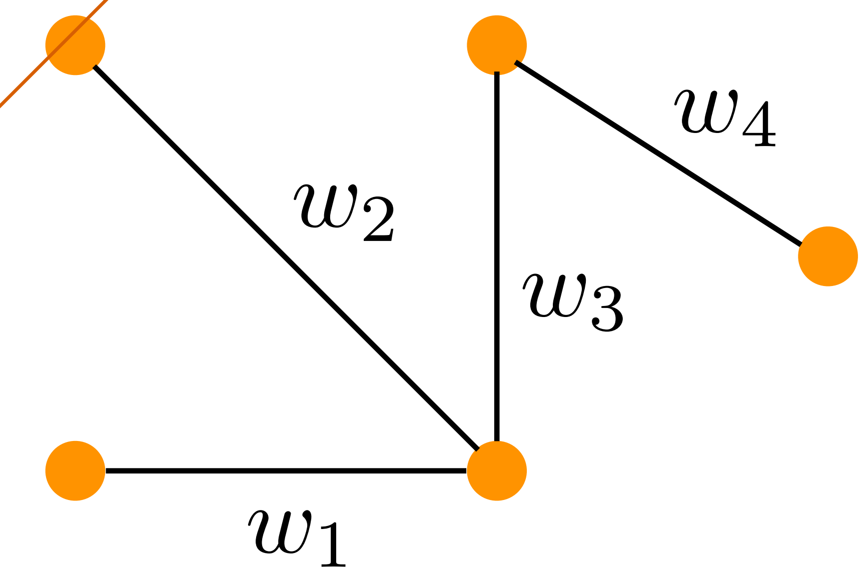
Beyond additive noise mechanism — perturbing edges by random walk



- For the sake of simplicity, suppose k is public;
- Consider a random work on $\binom{[n-1]}{k}$.

A Target distribution on graph topology

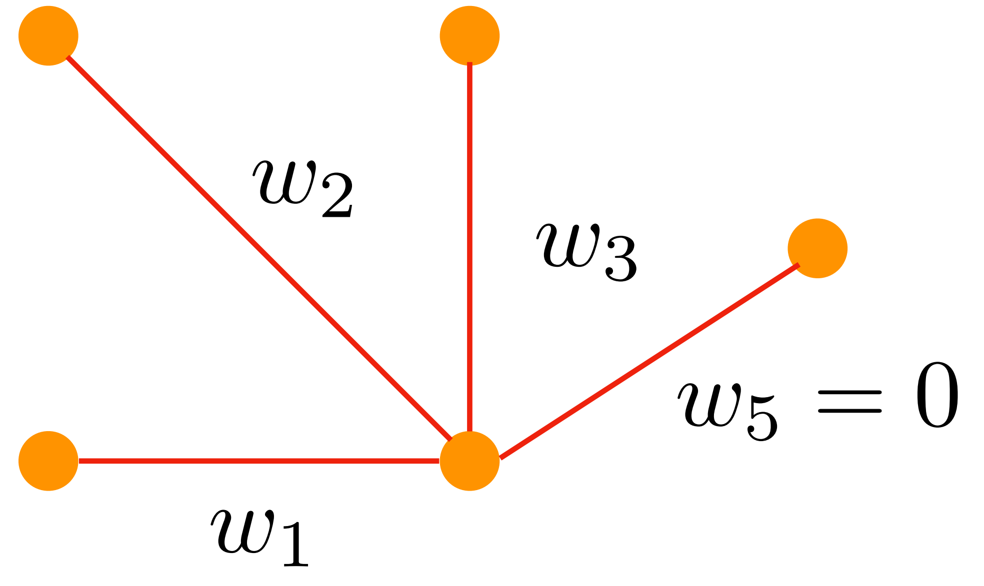
Input graph G



Given number of edges m, we sample from all undirected graphs with k edges **approximately** according to:

$$\forall S \in \{0,1\}^{\binom{n}{2}} \wedge |S| = k, \Pr[S] \propto \prod_{e \in S} \exp(\varepsilon \cdot w_e)$$

$G|S$, the restriction of G on S

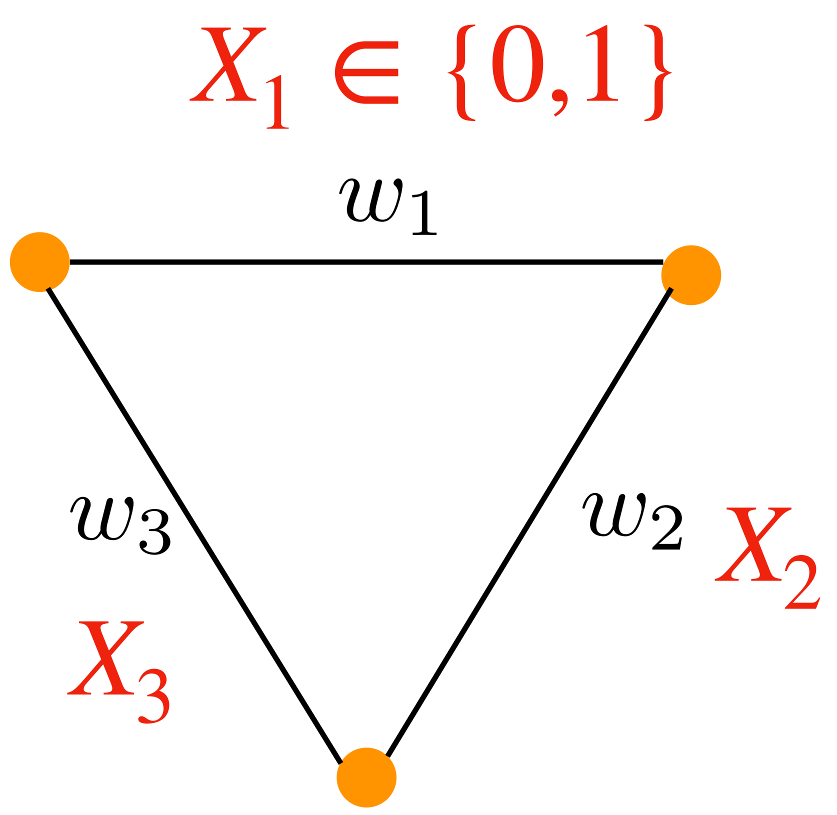


One can verify that this is equivalent to running exponential mechanism on such set of topologies with the utility function

$$f(G,S) = \|G - G|S\|_1.$$

$$\sum_{\substack{e \in G \\ e \notin S}} w_e$$

Sampling from the target distribution



$$\Pr[X_e = 1] = \frac{\exp(\varepsilon w_e)}{1 + \exp(\varepsilon w_e)}$$

A distribution on $\{0,1\}^n$

$$\forall X \in \{0,1\}^{\binom{n}{2}} \wedge |X| = k, \Pr[X] \propto \prod_{e \in X} \exp(\varepsilon \cdot w_e)$$

Fact: There exists an $O(n^2 m)$ time algorithm for exact sample by dynamic programming.

Question: If allow approximate sampling, could we do faster than $O(n^2 m)$ or $O(n^2)$?

Yes! We can do it in almost linear time $\tilde{O}(m)$.

Sampling faster: a MCMC way

Markov Chain Monte Carlo

Photoed in Piscataway, NJ

Sampling faster: a MCMC way

Markov Chain Monte Carlo

Photoed in Piscataway, NJ

$\{X_t \mid t \in T\}$ $X_t \in \Omega$

time t state space $\boxed{\Omega}$ state $x \in \Omega$

$\{X = \{0,1\}^{\binom{n}{2}} \mid \mathrm{nnz}(X) = k\}$ → All size k subsets of $\binom{n}{2}$

Sampling faster: a MCMC way

Markov Chain Monte Carlo

$\{X_t \mid t \in T\}$ $X_t \in \Omega$

time t state space $\boxed{\Omega}$ state $x \in \Omega$

$\{X = \{0,1\}^{\binom{n}{2}} \mid \mathrm{nnz}(X) = k\}$ → All size k subsets of $\binom{n}{2}$

Stochastic process over size k subsets: $X_0 \leftarrow E, X_1, \cdots, X_t$

X_{i+1} only depends on X_i

The real edge set E

Edge set E_t in time t

Sampling faster: a MCMC way

Markov Chain Monte Carlo

$\{X_t \mid t \in T\}$ $X_t \in \Omega$

time t state space $\boxed{\Omega}$ state $x \in \Omega$

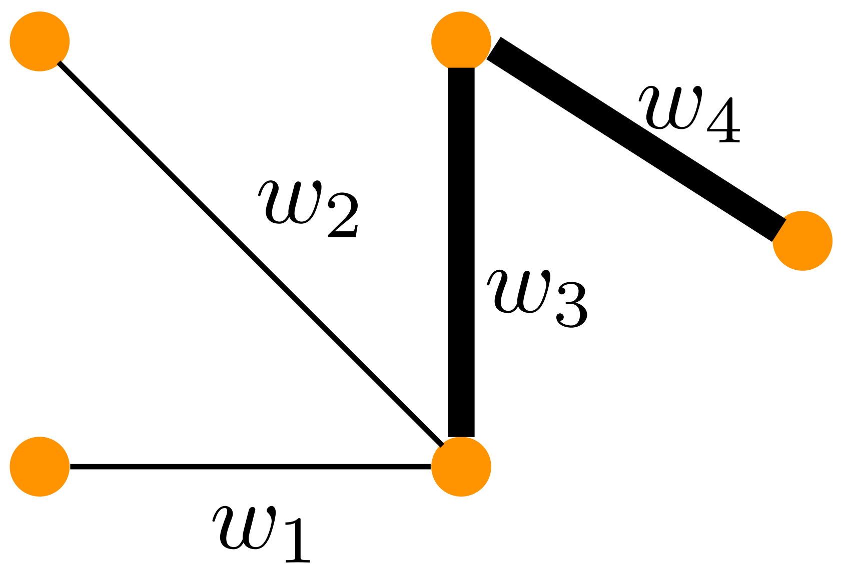
$\{X = \{0,1\}^{\binom{n}{2}} \mid \text{nnz}(X) = k\}$ → All size k subsets of $\binom{n}{2}$

Stochastic process over size k subsets: $X_0 \leftarrow E, X_1, \cdots, X_t$

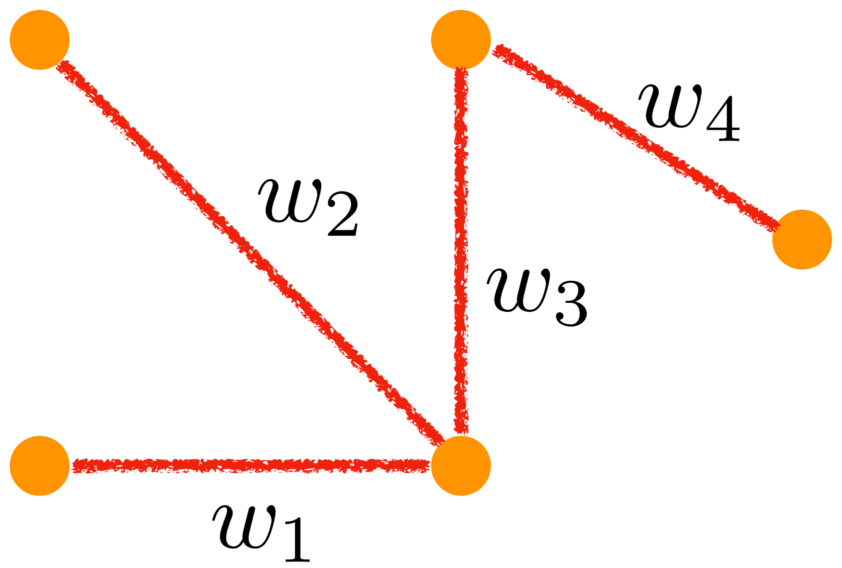
X_{i+1} only depends on X_i

$$\Pr[X_t] \propto \prod_{e \in X_t} \exp(\varepsilon \cdot w_e)$$

Basis-exchange sampling

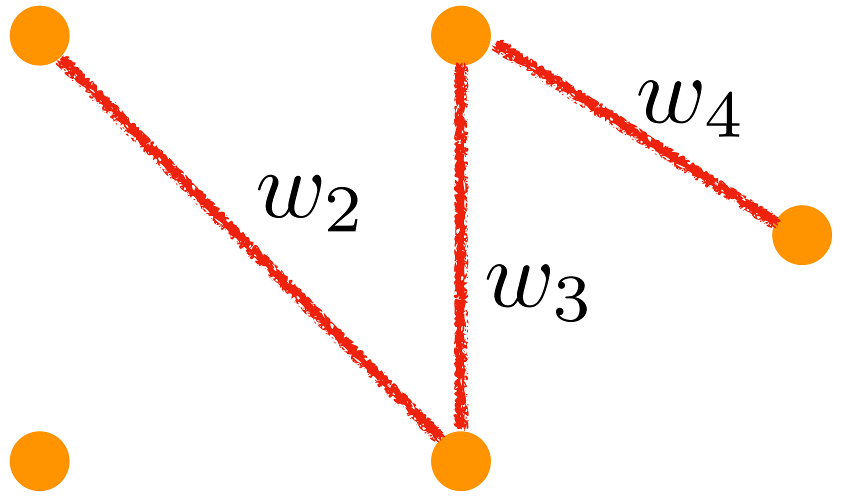


Input graph



X_0

Removing e

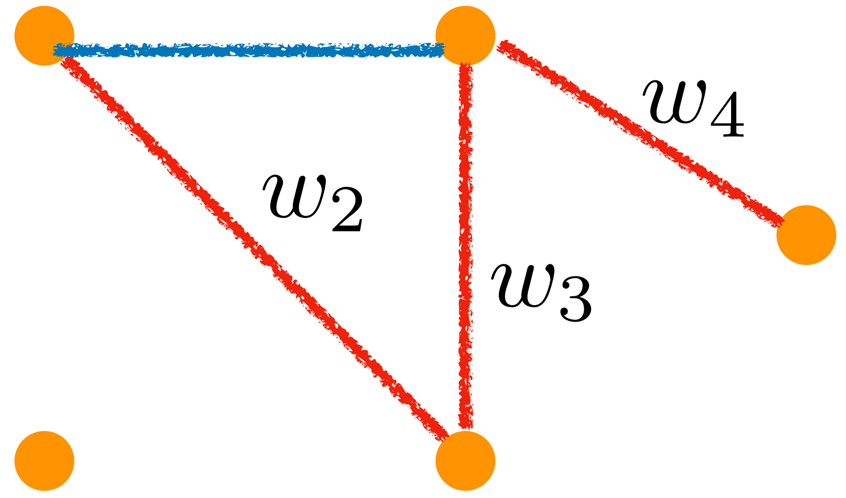


X_1'

Adding e'

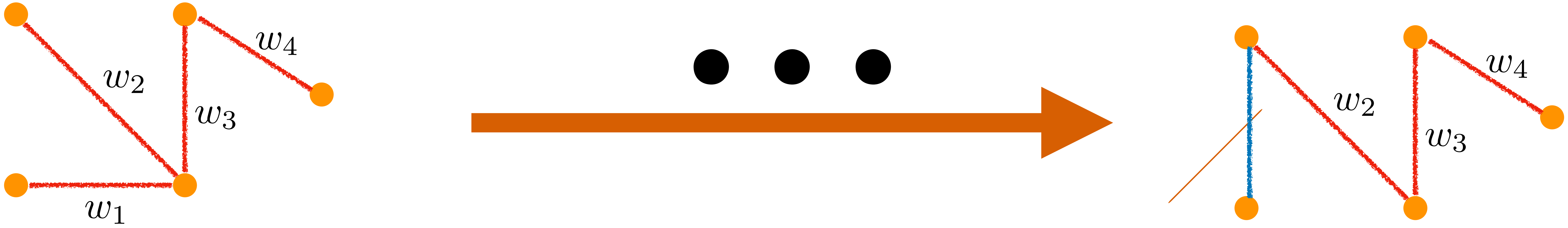
$\propto \pi(X_1' \cup \{e'\})$

Pr$[e'] \propto e^{\varepsilon w_{e'}}$



X_1

Basis-exchange sampling

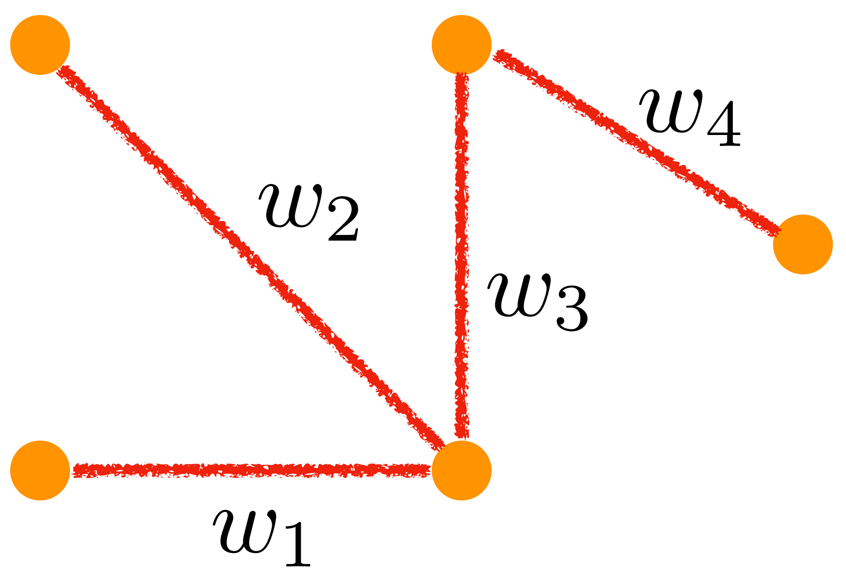


X_0 X_t

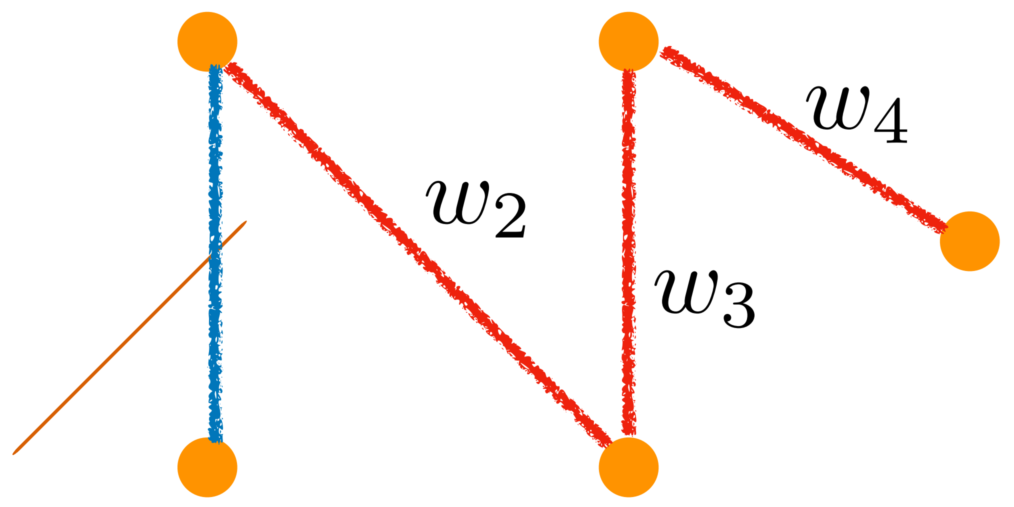
Facts 1: When $t = \infty$, $X_t \sim \pi$.

Facts 2: (Rapid mixing) When $t = \Omega(k \ln(1/\delta))$, X_t and X_∞ has total variation distance $\leq \delta$ (derived from strong log-concavity).

Basis-exchange sampling



X_0



X_t

◆ **Theorem (almost linear time sampler)**

For any $\varepsilon > 0$, there is a sampler $\mathrm{TS}_\varepsilon : \mathbb{N} \times \mathbb{R}_+^N \to 2^{[N]}$ such that for any given integer $k \leq N$ and an undirected graph G on n vertices, it outputs an edge set of size k in time $\tilde{O}(k)$ approximately according to distribution

$$\forall S \in \{0,1\}^{\binom{n}{2}} \wedge |S| = k, \pi(S) \propto \prod_{e \in S} \exp(\varepsilon \cdot w_e)$$

Almost linear time Approximate DP algorithm !

Our results on private cut & spectral approximation

Method	Additive error	Preserve sparsity?	Purely additive error?	Run-time
JL transformation [BBDS12]	$O\left(\frac{\sqrt{n}\log(n/\delta)}{\varepsilon}\right)$	No	No	$O(n^3)$
Analyze Gauss [DR14]	$O\left(\frac{\sqrt{n}\log(n/\delta)}{\varepsilon}\right)$	No	Yes	$O(n^2)$
Topology Sampler [LUZ24]	$O\left(\frac{\Delta\log^2(n)}{\varepsilon}\right)$	Yes	Yes	$O(n^2\|E\|\Delta)$
This paper	$O\left(\frac{\Delta\log(n/\delta)}{\varepsilon}\right)$	Yes	Yes	$\tilde{O}(\|E\|)$

Private Graph Spectrum Approximation (Δ: maximum unweighted degree)

Our results on private cut & spectral approximation

Method	Additive error	Preserve sparsity?	Purely additive error?	Run-time
Exponential mechanism	$O\left(\frac{n \log n}{\varepsilon}\right)$	Yes	No	Intractable
JL transformation [BBDS12]	$O\left(\frac{n^{1.5} \cdot \text{polylog}(n)}{\varepsilon}\right)$	No	No	$O(n^3)$
Analyze Gauss [DR14]	$O\left(\frac{n^{1.5} \cdot \text{polylog}(n)}{\varepsilon}\right)$	No	Yes	$O(n^2)$
Mirror Descent [EKKL20]	$O\left(\frac{n\sqrt{W} \cdot \text{polylog}(n)}{\varepsilon}\right)$	No	Yes	$\tilde{O}(n^7)$
Topology Sampler [LUZ24]	$O\left(\frac{n \cdot \text{polylog}(n)}{\varepsilon}\right)$	Yes	Yes	$\tilde{O}(n^7)$
This paper	$O\left(\frac{n \cdot \text{polylog}(n)}{\varepsilon}\right)$	Yes	Yes	$\tilde{O}(n)$

Private Graph Cut Approximation on sparse graphs ($m \sim n \cdot \text{polylog}(n)$)

Some further questions

- For private cut approximation, is it possible to achieve the instance **optimal** error bound $O(\sqrt{mn})$ with linear time algorithms?

- Is there other applications of MCMC method in the designation of **efficient** differentially private algorithms?