(Quantum) Complexity of Testing Signed Graph Clusterability

Kuo-Chin Chen
Joint work with Simon Apers and Min-Hsiu Hsieh

Foxconn Research

Talk in Workshop on Quantum Science and Technology

Outlines

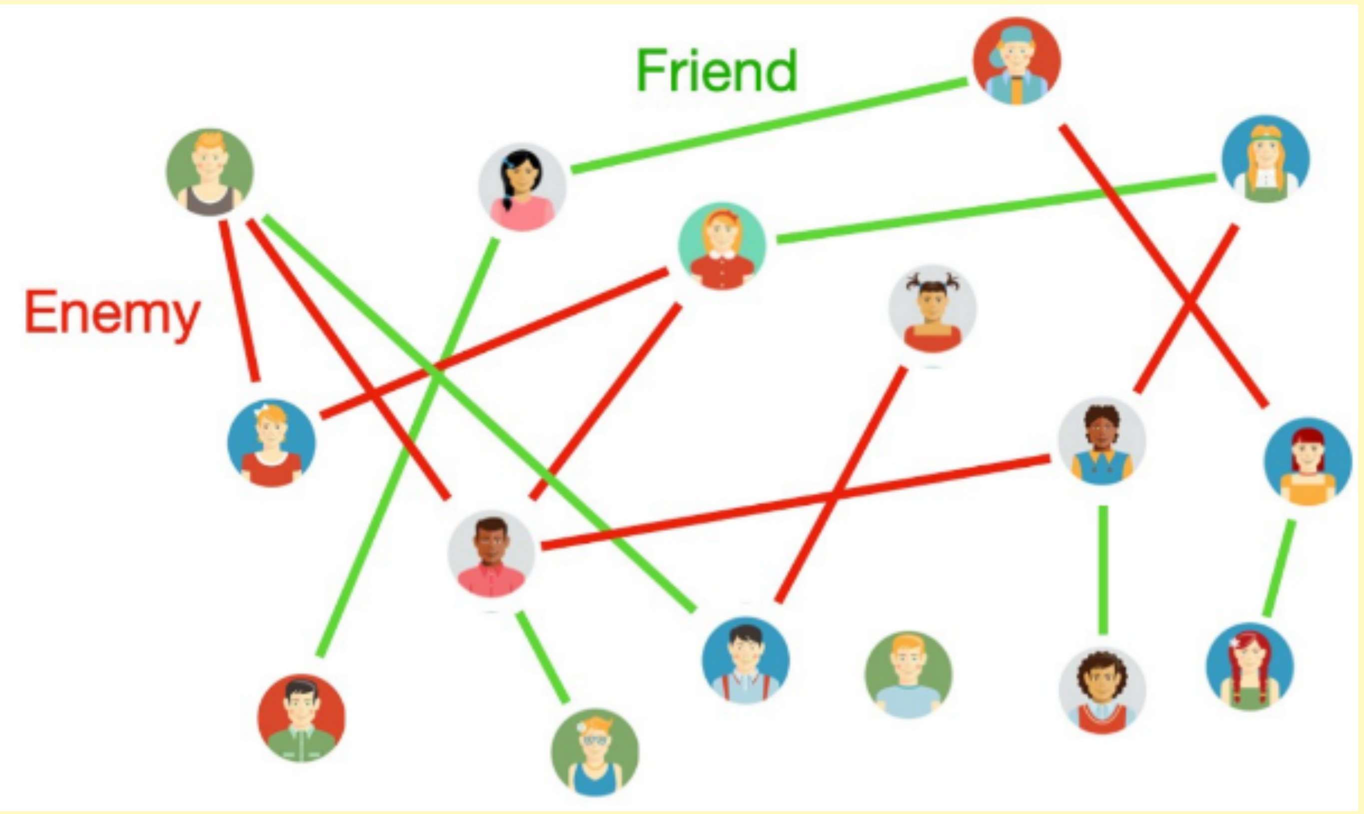
- **Motivation and our main results**
- **Classical clusterability testing query lower bound** $\rightarrow \Omega(\sqrt{N})$
- **Quantum clusterability testing algorithm** $\rightarrow O(N^{1/3})$

Outlines

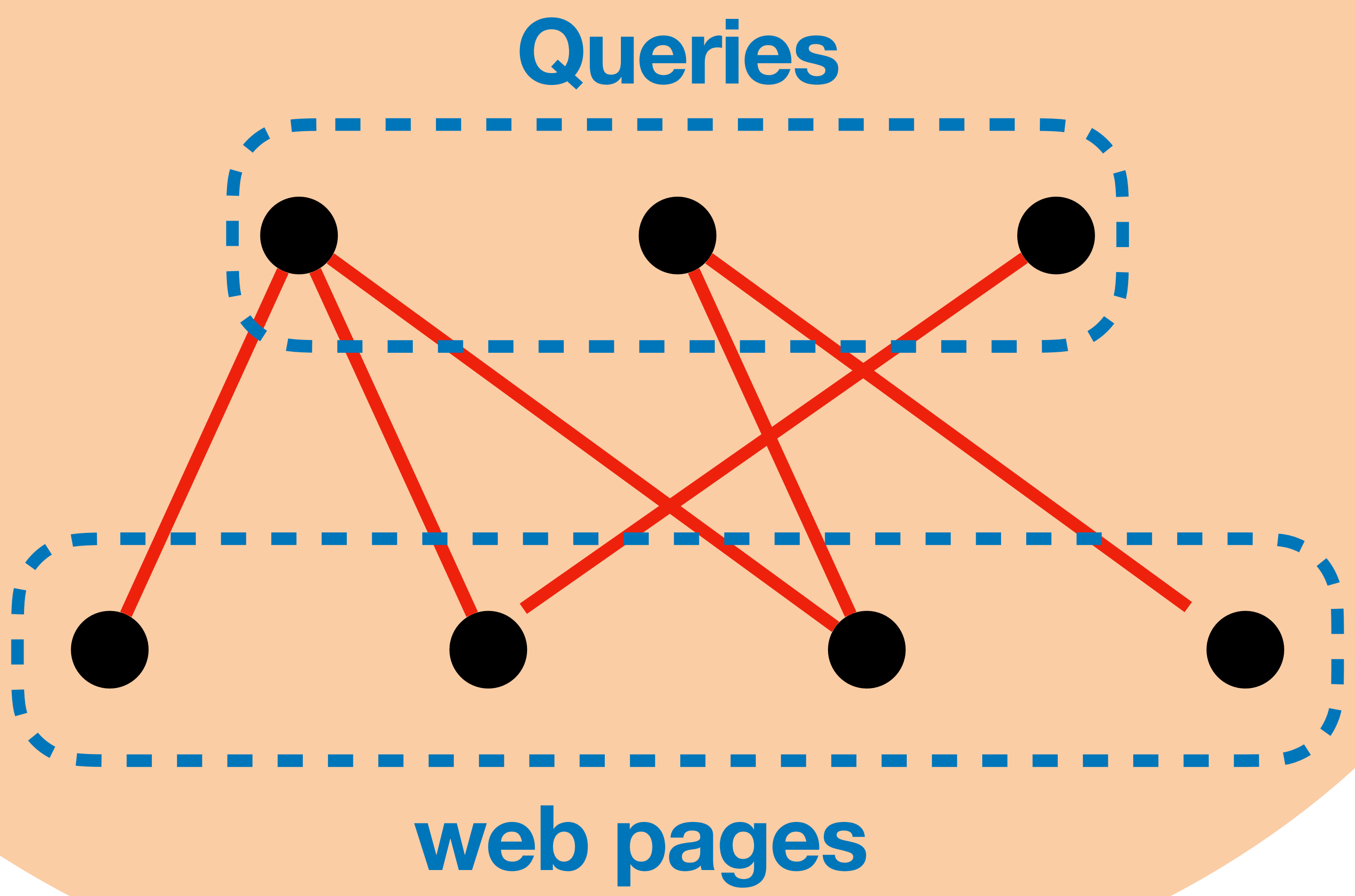
- **Motivation and our main results**
- Classical clusterability testing query lower bound
- Quantum clusterability testing algorithm

Applications of Graph Properties

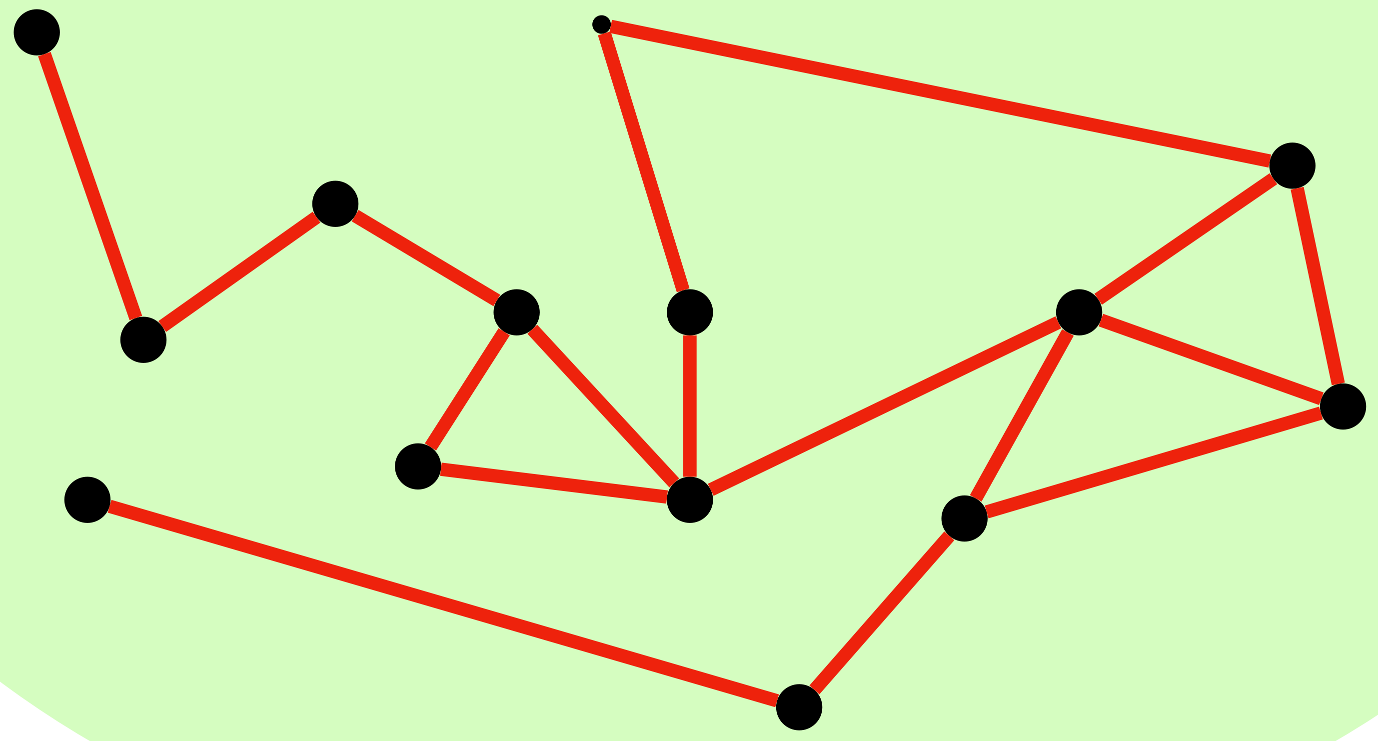
Clusterability
(Social network)



Bipartite
(Web search engines)



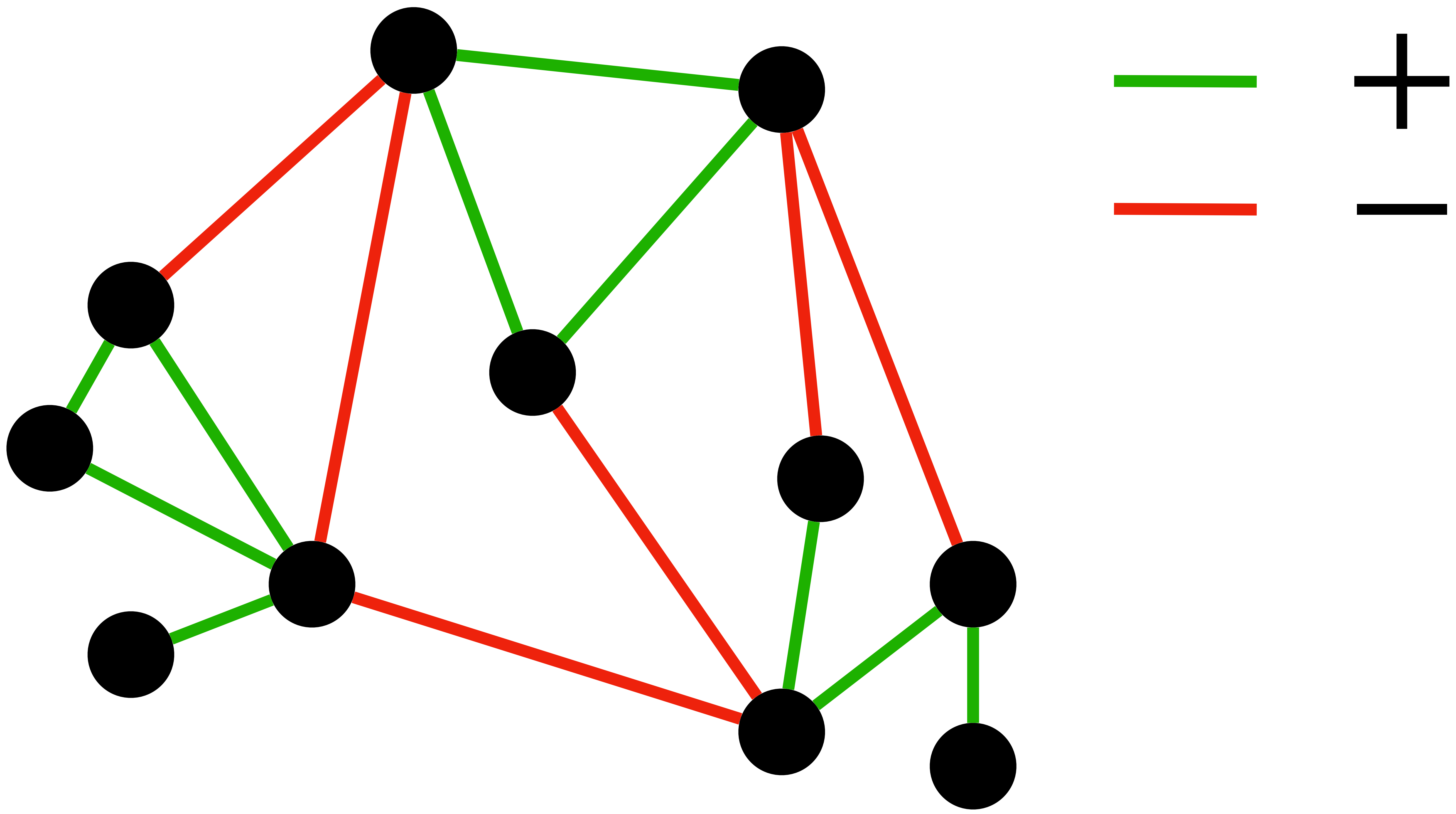
Expander
(MCMC converge rate)



Clusterability for Signed Graphs

Definition 1.2 (Signed graph)

A signed graph $G(V, E, \sigma)$ is a graph whose each edge is assigned by a mapping $\sigma : E \to \{+, -\}$.



Clusterability for Signed Graphs

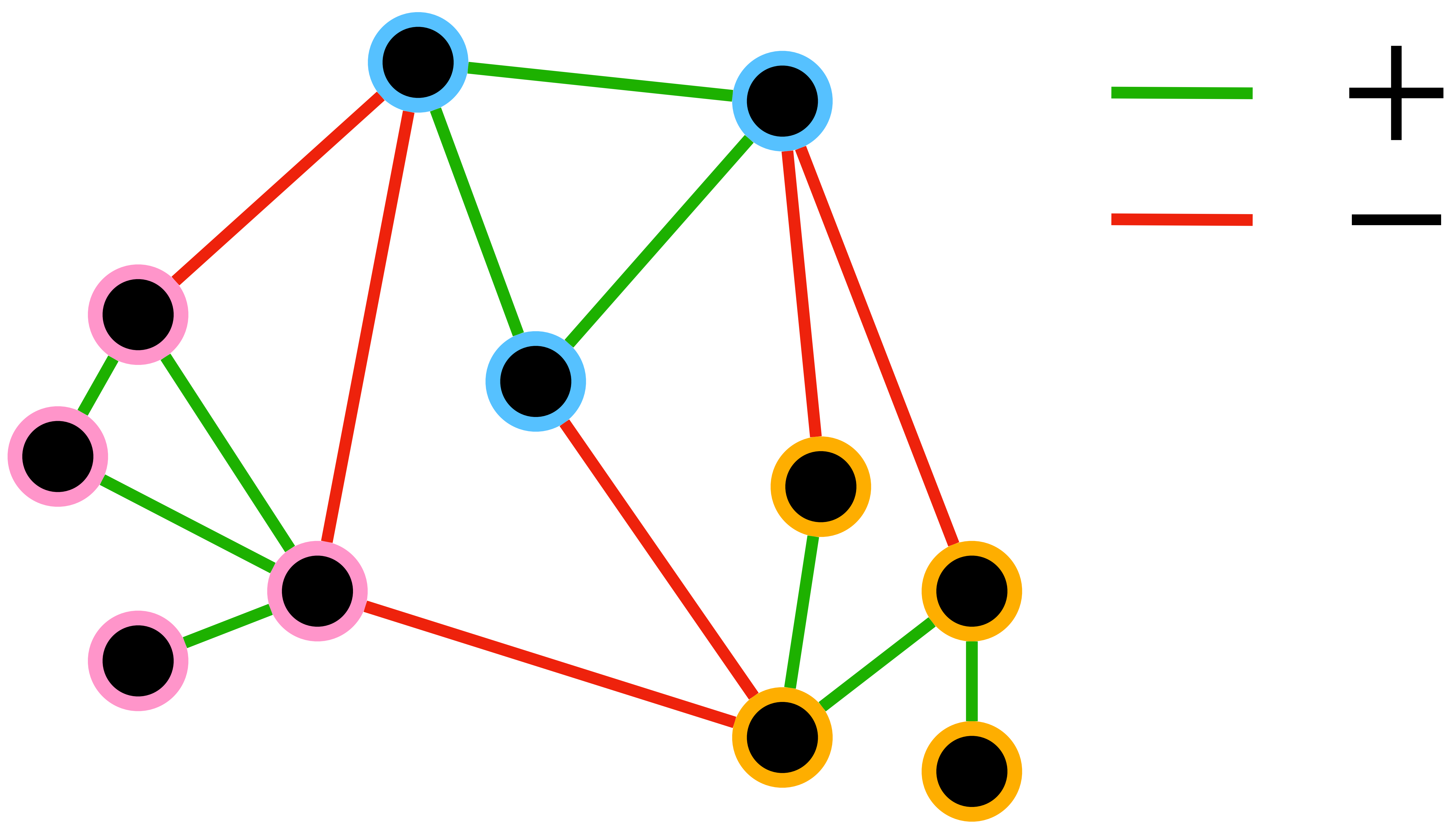
Definition 1.2 (Signed graph)

A signed graph $G(V, E, \sigma)$ is a graph whose each edge is assigned by a mapping $\sigma : E \to \{+, -\}$.

Definition 1.3 (Clusterability)

A signed graph is clusterable if it can be decomposed into several components s.t.

- The edges in each component are $+$
- The edges connecting components are $-$.



Clusterable

Clusterability for Signed Graphs

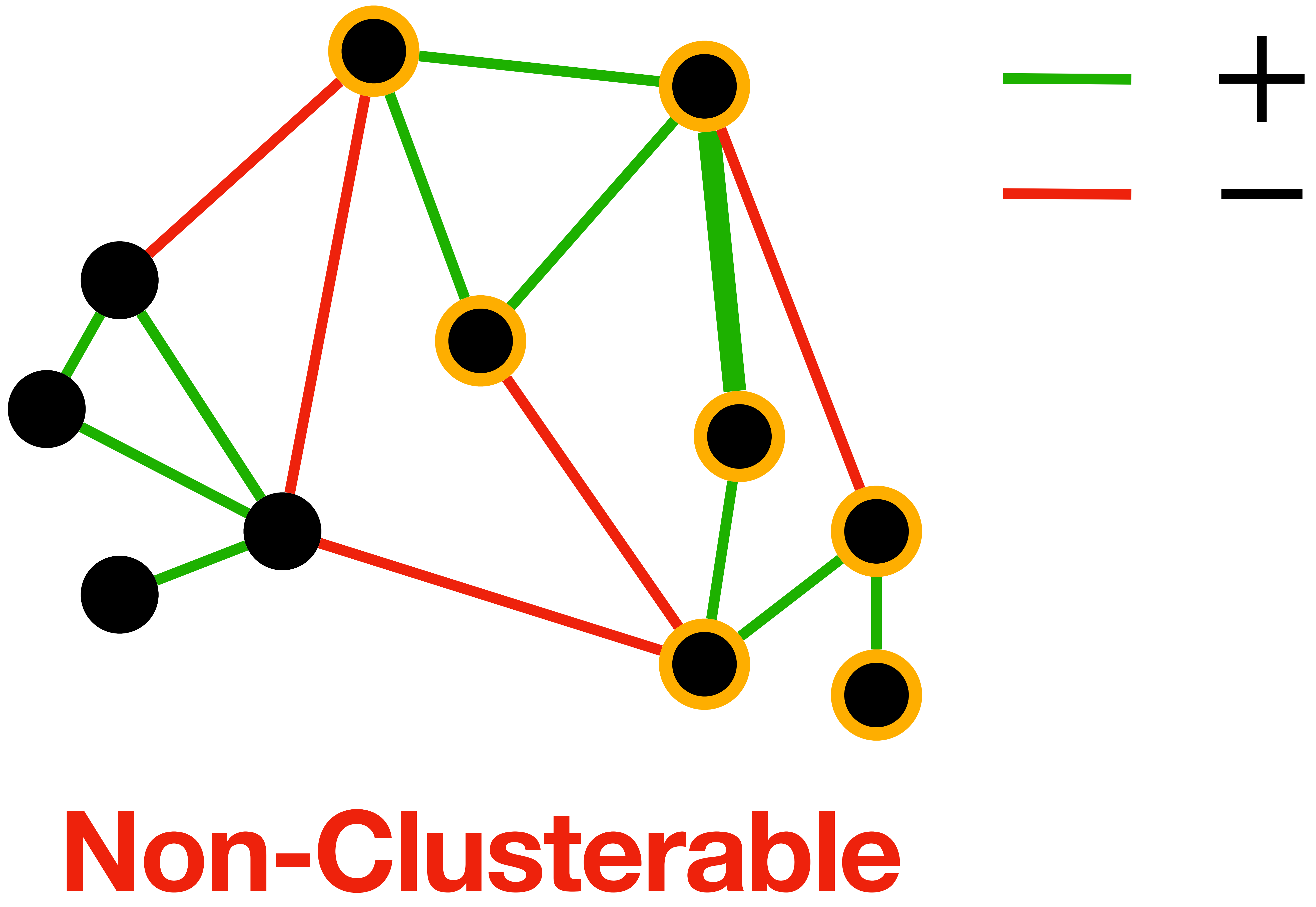
Definition 1.2 (Signed graph)

A signed graph $G(V, E, \sigma)$ is a graph whose each edge is assigned by a mapping $\sigma : E \to \{+, -\}$.

Definition 1.3 (Clusterability)

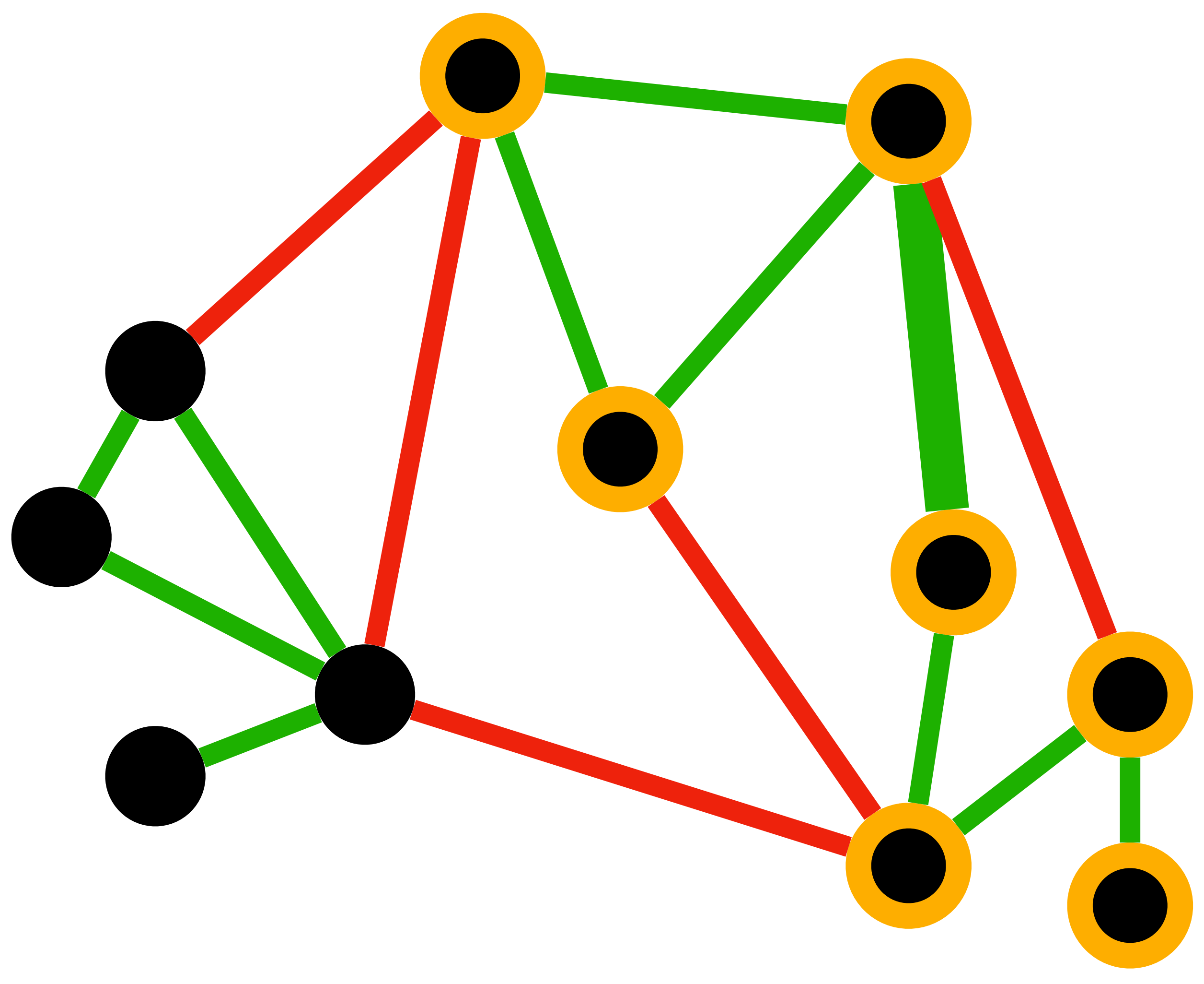
A signed graph is clusterable if it can be decomposed into several components s.t.

- The edges in each component are $+$
- The edges connecting components are $-$.



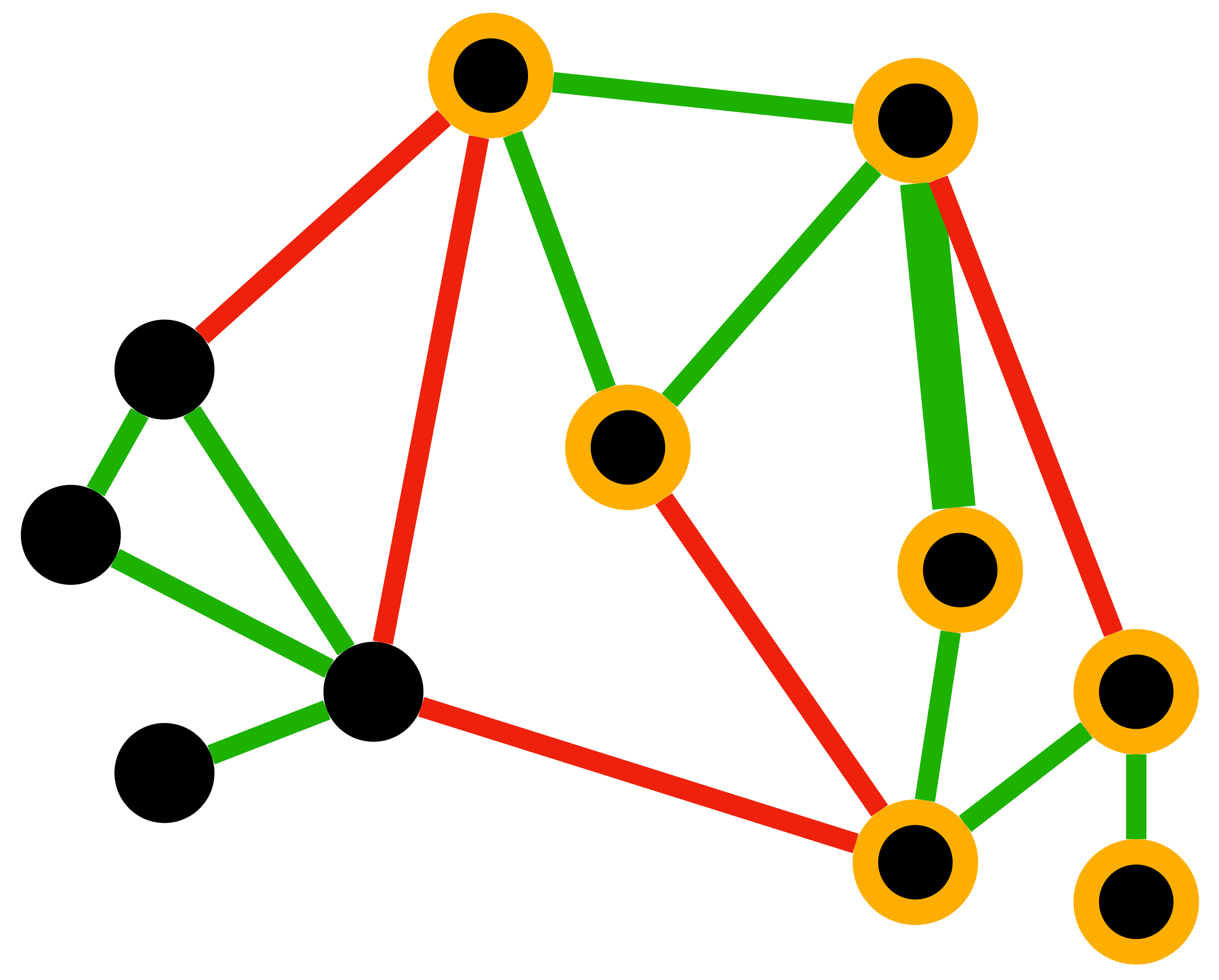
Bounded Degree Graph Query Model

- Given the adjacent list of a graph with degree bound d.
- Query to the list $\rightarrow$ Explore this graph.
- One query $\rightarrow$ one edge



Bounded Degree Graph Query Model

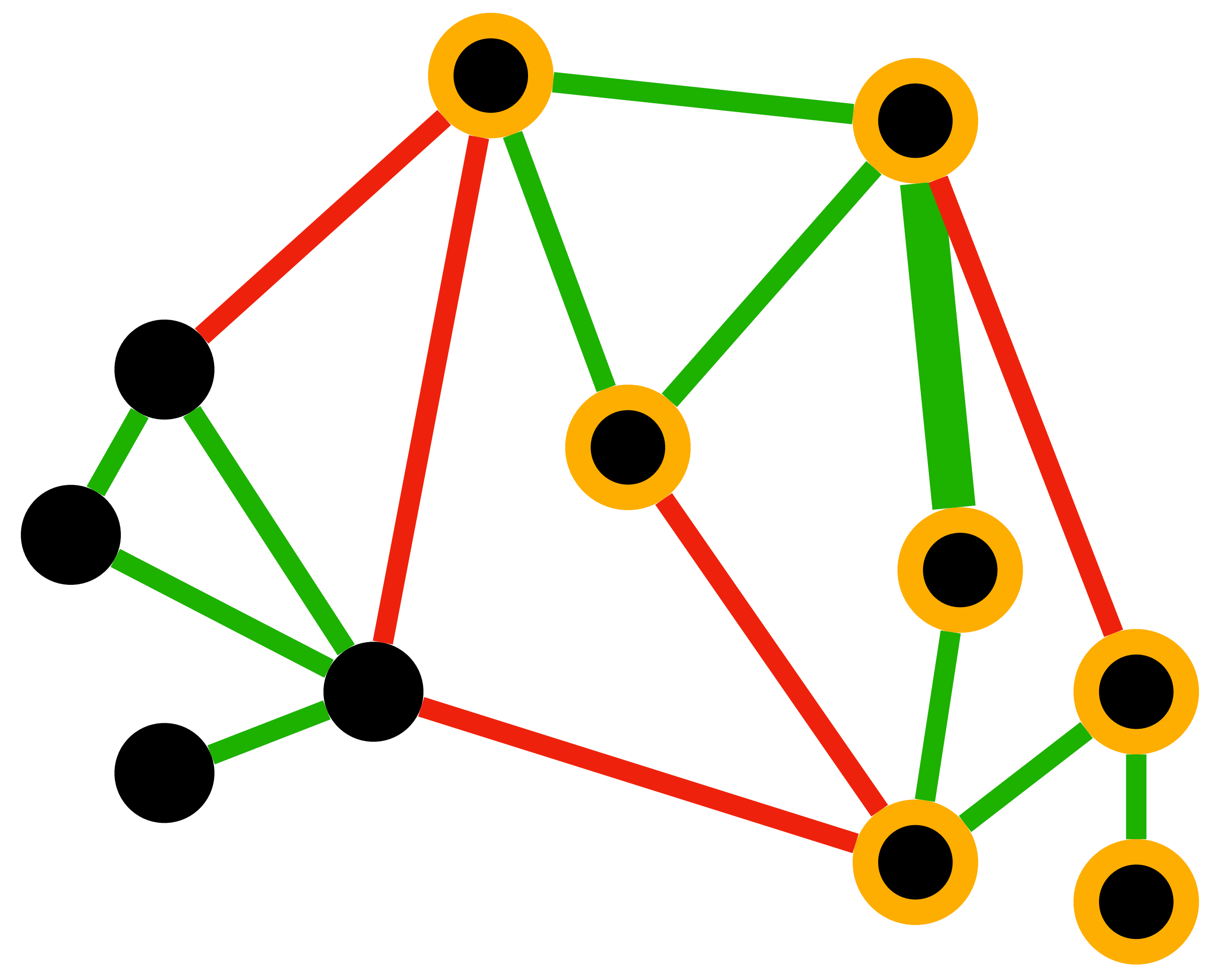
- Given the adjacent list of a graph with degree bound d.
- Query to the list $\rightarrow$ Explore this graph.
- One query $\rightarrow$ one edge



Learn the clusterability without error $\rightarrow$ require $O(N)$ queries.

Bounded Degree Graph Query Model

- Given the adjacent list of a graph with degree bound d.
- Query to the list $\rightarrow$ Explore this graph.
- One query $\rightarrow$ one edge



Learn the clusterability without error $\rightarrow$ require $O(N)$ queries.

Test the clusterability in an approximated manner with less queries?

Graph Property Testing

An approximated algorithm

A graph property $\mathcal{P}$ tester is a randomized algorithm:

Input:

1. Query access to a graph $G(V, E, \sigma)$ with maximum degree d and $|V| = N$.
2. An error parameter ϵ.

Output with probability at least $2/3$:

- Remove or add at least ϵNd edges to make G satisfy $\mathcal{P}$ $\rightarrow$ REJECT (ϵ-far from $\mathcal{P}$)
- Otherwise $\rightarrow$ ACCEPT

Previous Works and Open Problems

	Bipartiteness	Expander	Clusterability
Classical UB	$\tilde{O}(\sqrt{N})$ [1]	$\tilde{O}(\sqrt{N})$ [2]	$\tilde{O}(\sqrt{N})$ [3]
Classical LB	$\tilde{\Omega}(\sqrt{N})$ [4]	$\tilde{\Omega}(\sqrt{N})$ [5]	?
Quantum UB	$\tilde{O}(N^{1/3})$ [6]	$\tilde{O}(N^{1/3})$ [6]	?

[1] Goldreich and Dana, STOC 1998

[2] Goldreich and Dana, ECCC report 2001

[3] Adriaens and Apers, Arxiv 2021

[4] Goldreich and Dana, STOC 1997

[5] Goldreich and Dana, ECCC report 2001

[6] Ambainis, Childs, and Liu, Proc. RANDOM 2011.

Our Contributions

	Bipartiteness	Expander	Clusterability
Classical UB	$\tilde{O}(\sqrt{N})$ [1]	$\tilde{O}(\sqrt{N})$ [2]	$\tilde{O}(\sqrt{N})$ [3]
Classical LB	$\tilde{\Omega}(\sqrt{N})$ [4]	$\tilde{\Omega}(\sqrt{N})$ [5]	$\tilde{\Omega}(\sqrt{N})$
Quantum UB	$\tilde{O}(N^{1/3})$ [6]	$\tilde{O}(N^{1/3})$ [6]	$\tilde{O}(N^{1/3})$

[1] Goldreich and Dana, STOC 1998

[2] Goldreich and Dana, ECCC report 2001

[3] Adriaens and Apers, Arxiv 2021

[4] Goldreich and Dana, STOC 1997

[5] Goldreich and Dana, ECCC report 2001

[6] Ambainis, Childs, and Liu, Proc. RANDOM 2011.

Outlines

- Motivation and our main results
- **Classical clusterability testing query lower bound**
- Quantum clusterability testing algorithm

Classical Query Lower Bound for Testing Clusterability

Theorem 2.1

Any classical clusterability tester requires $\tilde{\Omega}(\sqrt{N})$ queries.

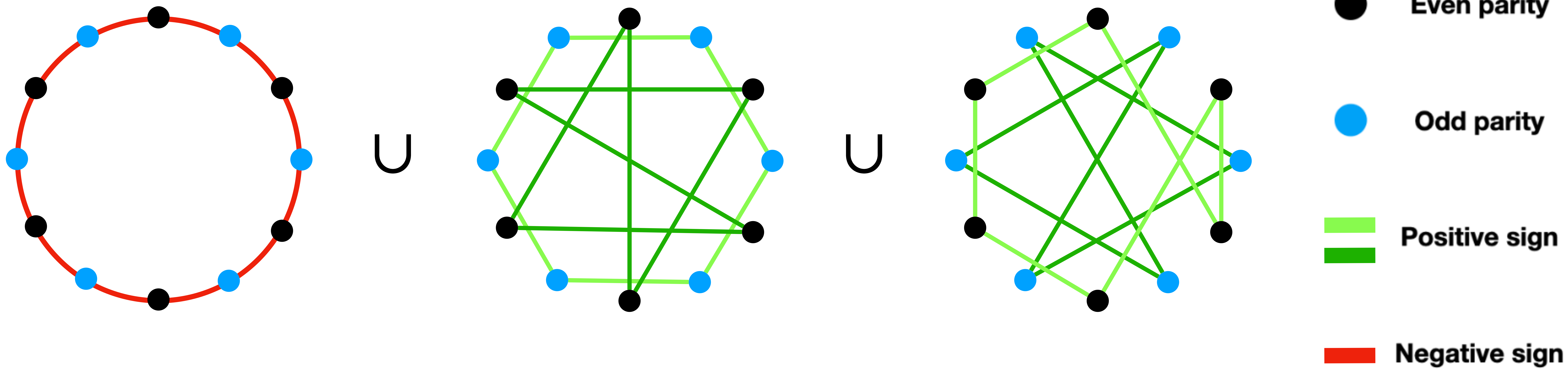
1. To prove the query lower bound → design a **hard** instance.
2. (Lemma 2.2): Construct two sets of graphs $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ s.t.
   - $\mathcal{G}_1^N \to$ clusterable
   - $\mathcal{G}_2^N \to \epsilon$-far from clusterable W.H.P.
3. (Lemma 2.3): $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ can not be distinguished within $\tilde{\Omega}(\sqrt{N})$ queries for any classical algorithm.

Constructing two sets of graphs

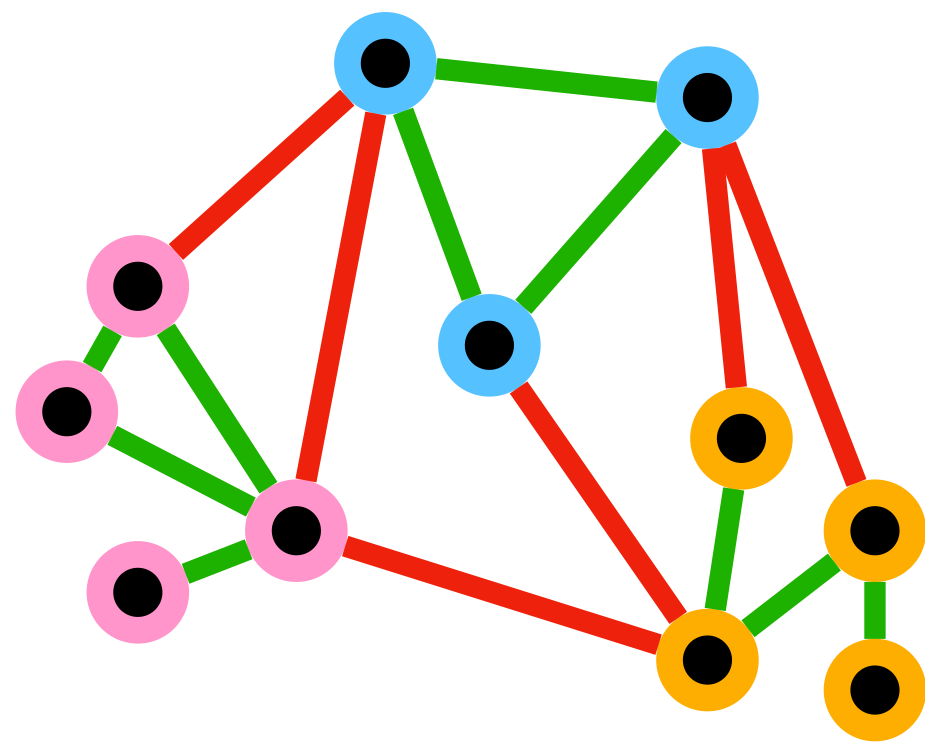
Lemma 2.2

There exist two families of graphs $\mathcal{G}_i^N$ such that the graph in $\mathcal{G}_2^N$ are 0.01-far from clusterable with probability at least $1 - \exp(-\Omega(N))$, and the graph in $\mathcal{G}_1^N$ are all clusterable.

$\mathcal{G}_1^N \rightarrow$



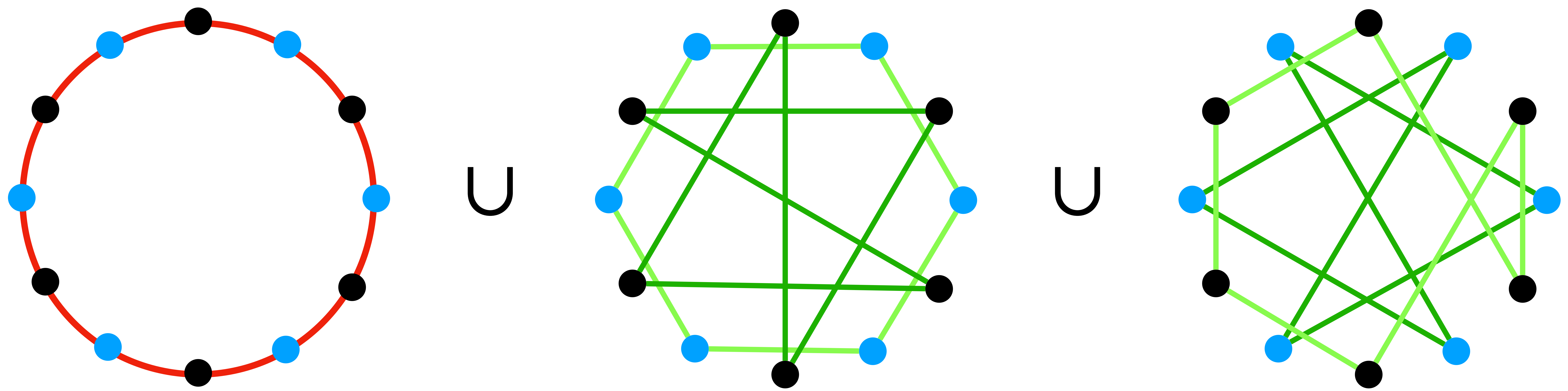
Proof sketch



The graphs in $\mathcal{G}_2^N$ are clusterable obviously since:

1. The positive edges connect the vertices with the same parity.
2. The negative edges connect the vertices with the distinct parity.

$\mathcal{G}_1^N \to$

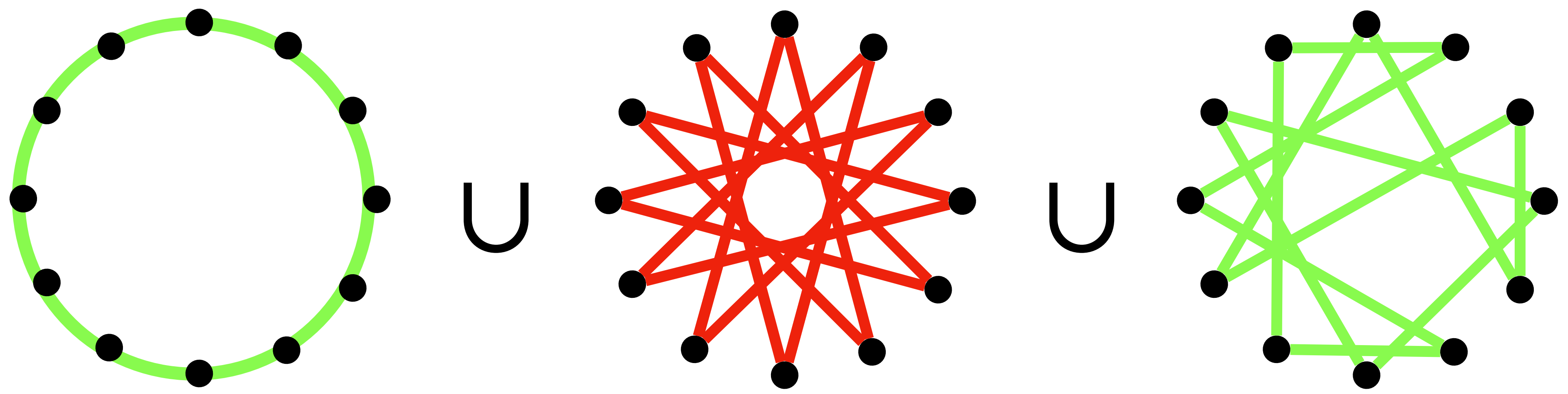


Constructing two sets of graphs

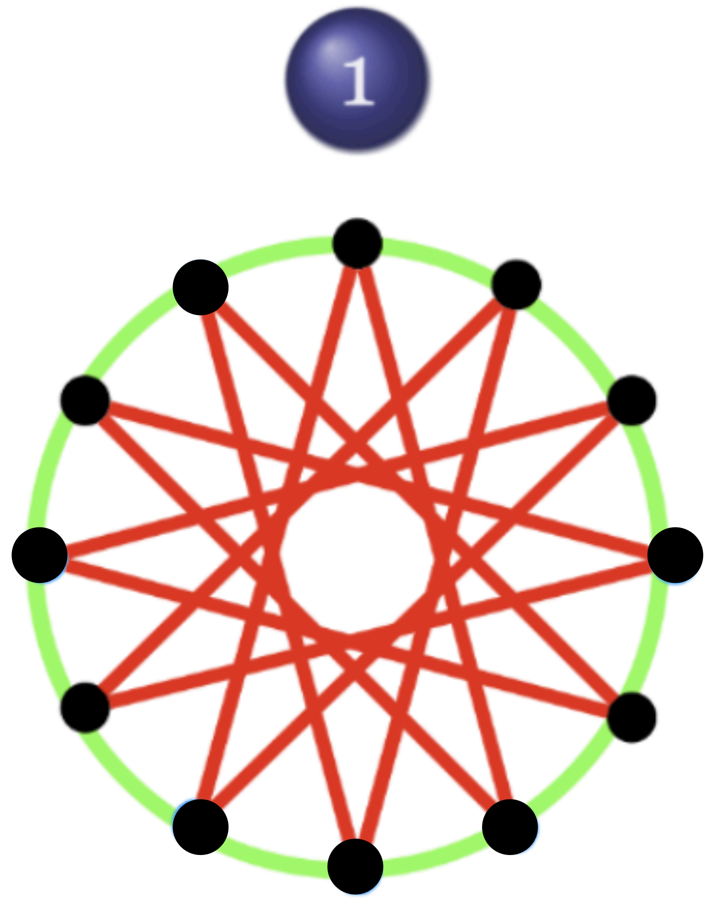
Lemma 2.2

There exist two families of graphs $\mathcal{G}_i^N$ such that the graph in $\mathcal{G}_2^N$ are 0.01-far from clusterable with probability at least $1 - \exp(-\Omega(N))$, and the graph in $\mathcal{G}_1^N$ are all clusterable.

Proof sketch

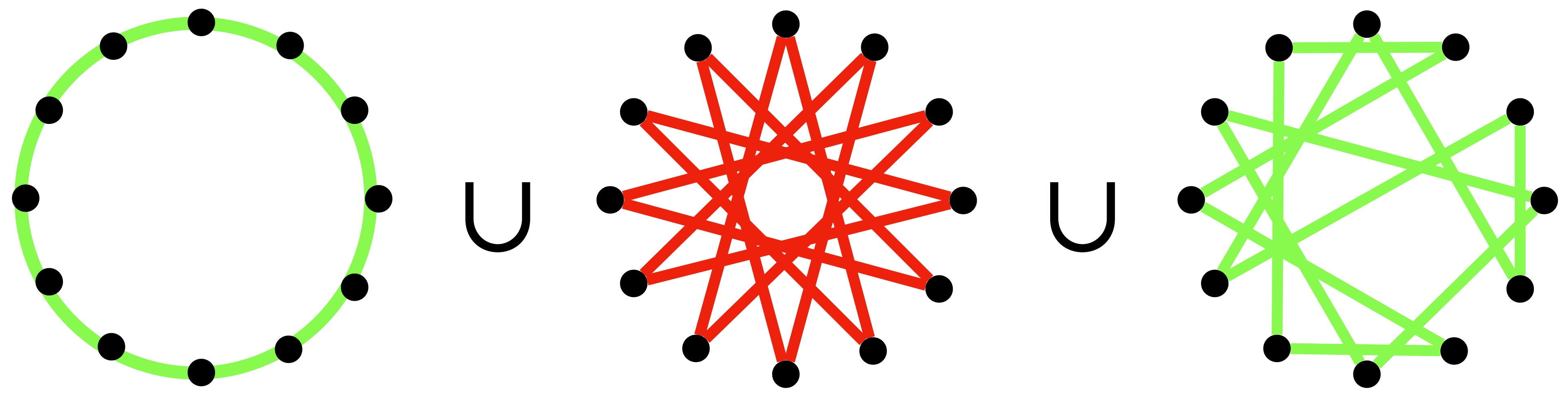
$\mathcal{G}_2^N \rightarrow$ 

1. Every graph in $\mathcal{G}_1^N$ is not clusterable.

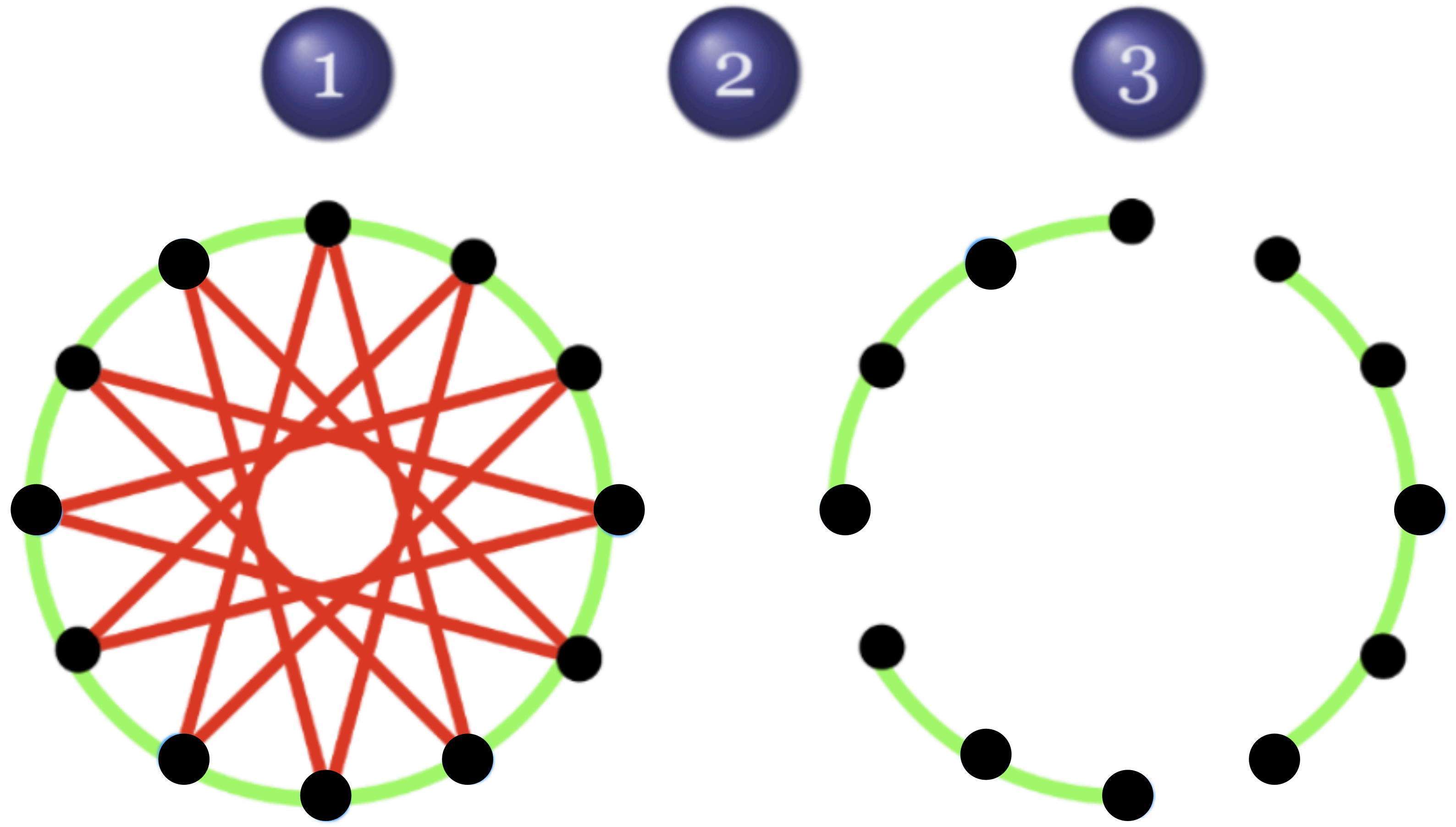


Proof sketch

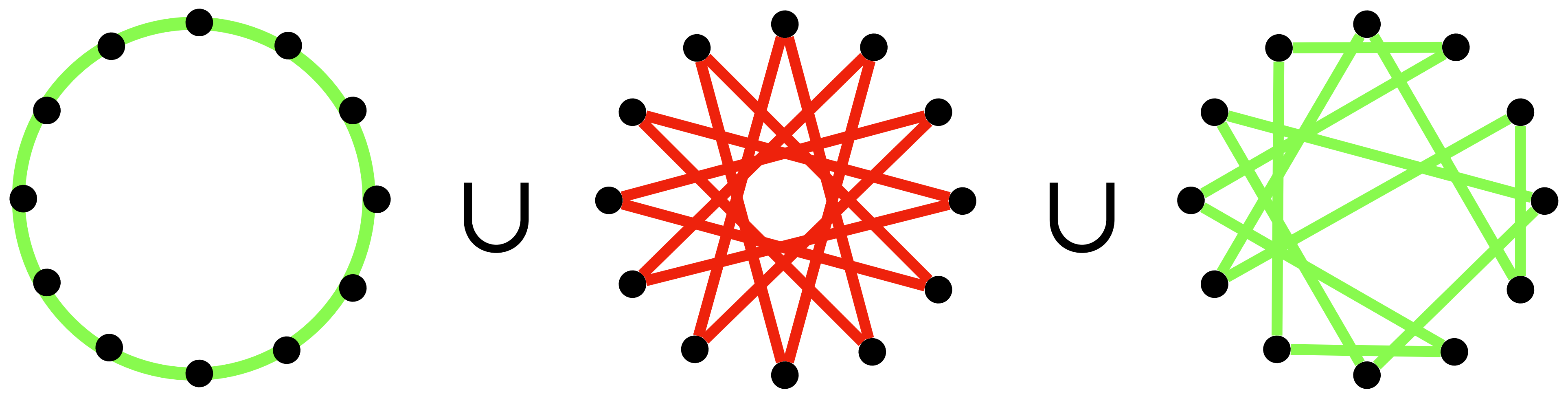
$\mathcal{G}_2^N \rightarrow$



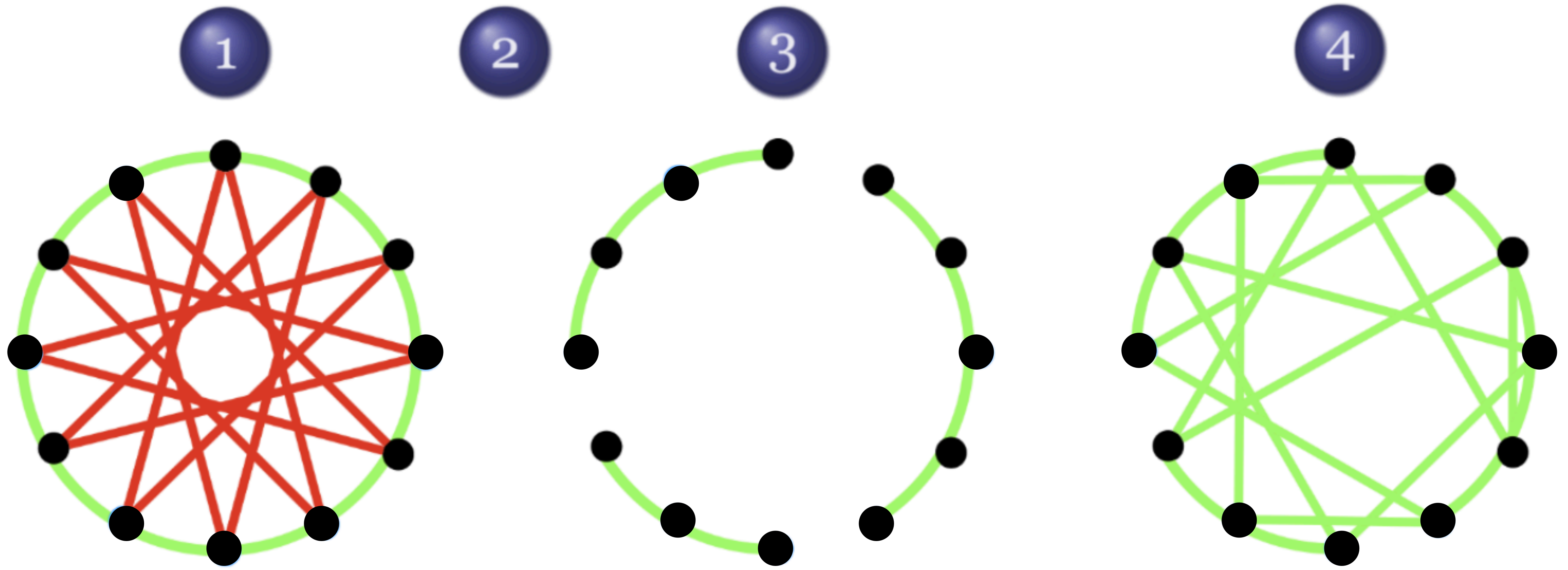
1. Every graph in $\mathcal{G}_1^N$ is not clusterable.
2. Make these graphs clusterable $\rightarrow$ Must remove some cycle edges.
3. Assume $x < \epsilon N d$ cycle edges are removed.
   $\rightarrow$ x cycle components.



Proof sketch

$\mathcal{G}_2^N \to$ 

1. Every graph in $\mathcal{G}_2^N$ is not clusterable.
2. Make these graphs clusterable $\to$ Must remove some cycle edges.
3. Assume $x < \epsilon N d$ cycle edges are removed.
   $\to x$ cycle components.
4. The positive matching edges can connect these cycle components.



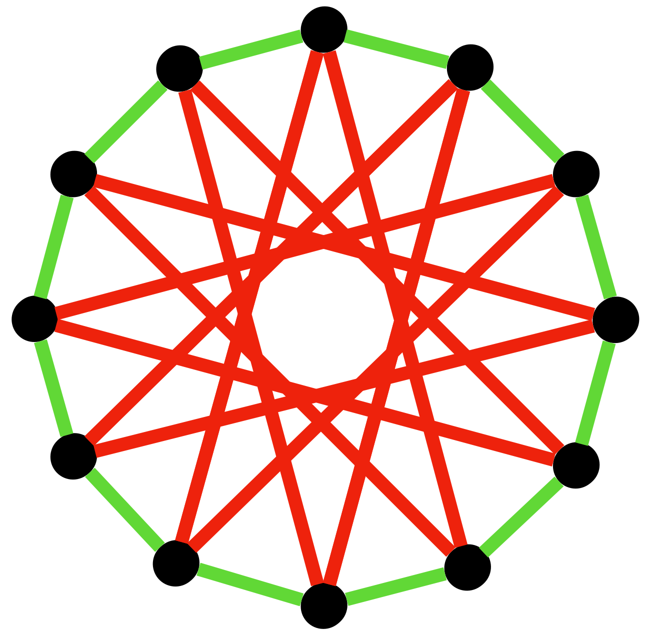
Two sets of graphs are indistinguishable

Proof sketch

Lemma 2.3

No algorithm can distinguish $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ within $\tilde{\Omega}(\sqrt{N})$ queries.

- Sample a graph in $\mathcal{G}_1^N$ or $\mathcal{G}_2^N$



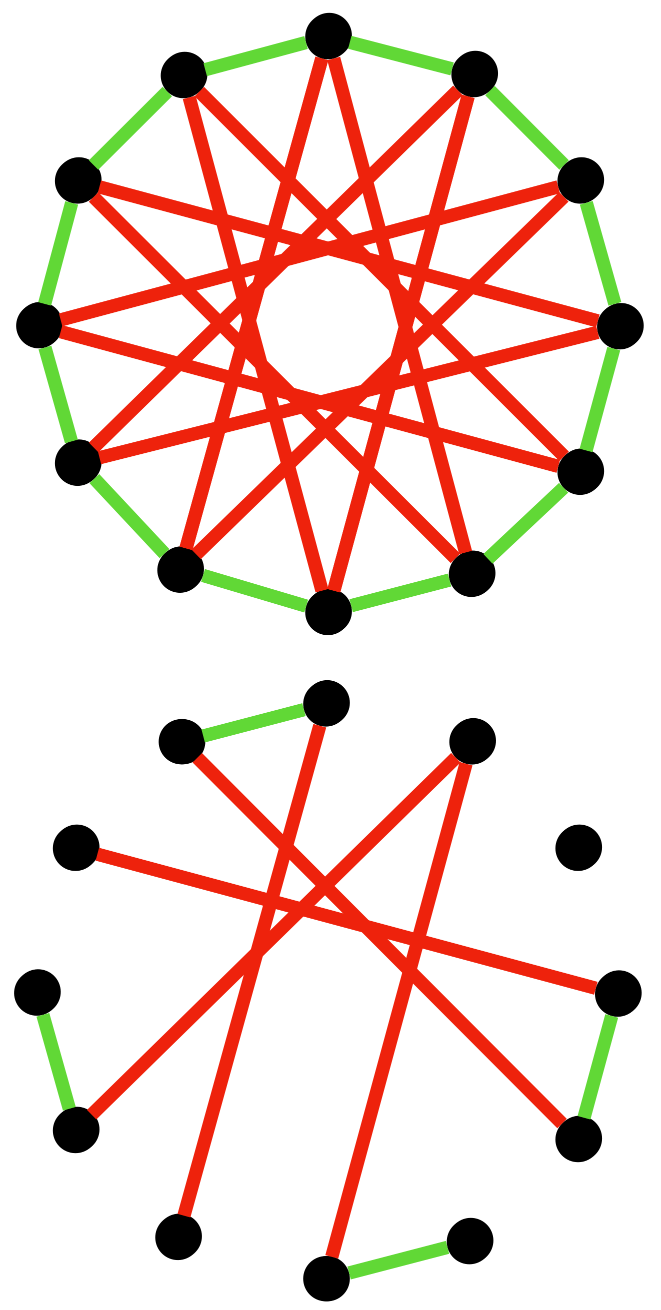
Two sets of graphs are indistinguishable

Proof sketch

Lemma 2.3

No algorithm can distinguish $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ within $\tilde{\Omega}(\sqrt{N})$ queries.

- Sample a graph in $\mathcal{G}_1^N$ or $\mathcal{G}_2^N$
- The algorithm makes $\sqrt{N}$ queries (explores $\sqrt{N}$ edges)



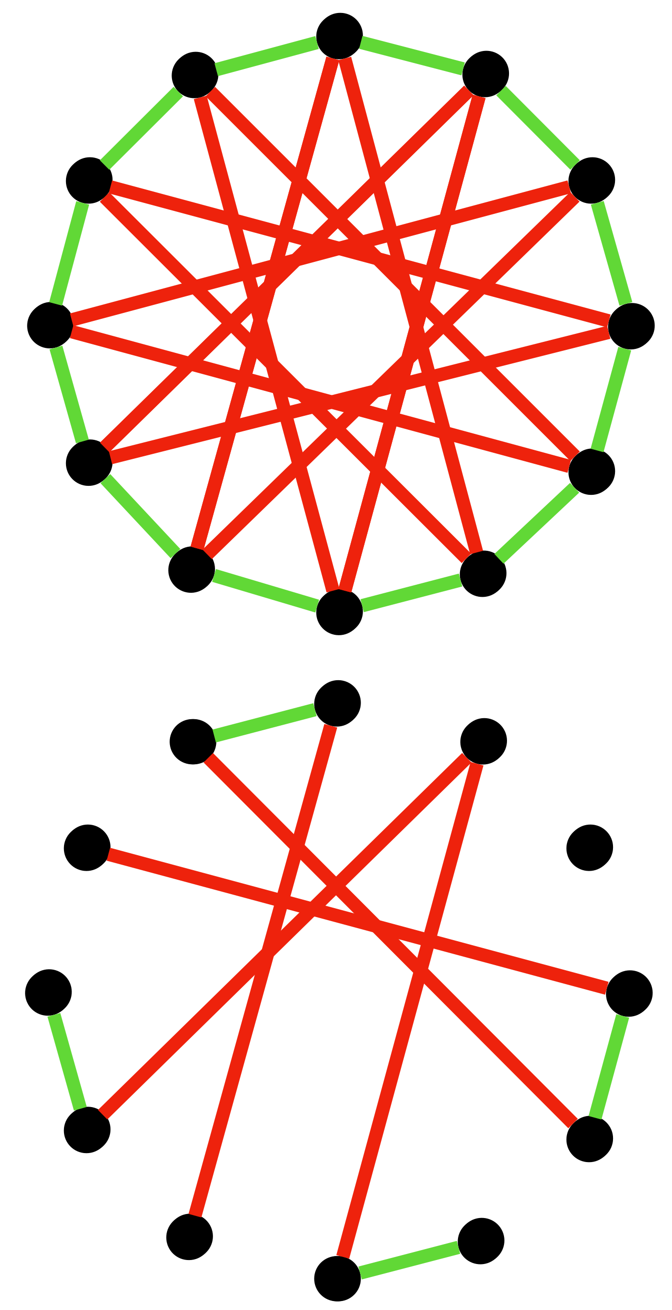
Two sets of graphs are indistinguishable

Proof sketch

Lemma 2.3

No algorithm can distinguish $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ within $\tilde{\Omega}(\sqrt{N})$ queries.

- Sample a graph in $\mathcal{G}_1^N$ or $\mathcal{G}_2^N$
- The algorithm makes $\sqrt{N}$ queries (explores $\sqrt{N}$ edges)
  - Find no cycle: algorithm can not distinguish $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$



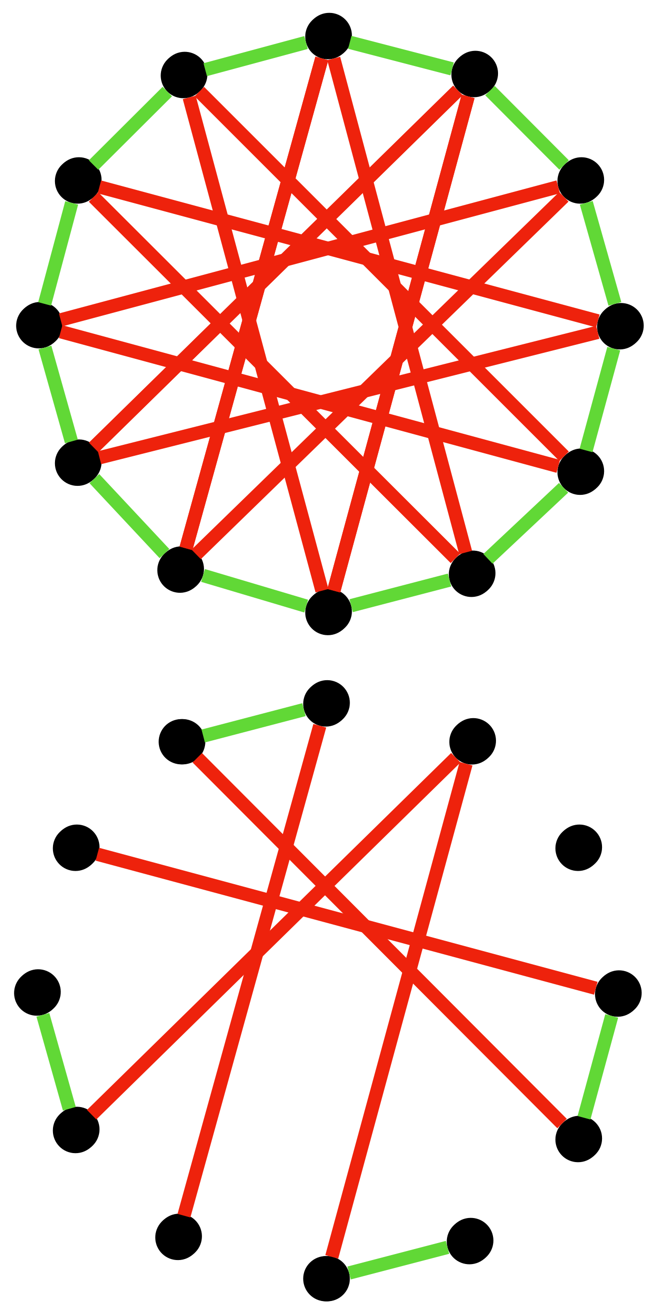
Two sets of graphs are indistinguishable

Proof sketch

Lemma 2.3

No algorithm can distinguish $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ within $\tilde{\Omega}(\sqrt{N})$ queries.

- Sample a graph in $\mathcal{G}_1^N$ or $\mathcal{G}_2^N$
- The algorithm makes $\sqrt{N}$ queries (explores $\sqrt{N}$ edges)
  - Find no cycle: algorithm can not distinguish $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$
  - Find a cycle: with a probability $< 1/10$ by using $\sqrt{N}$ queries



Classical Query Lower Bound for Testing Clusterability

Theorem 2.1

Any classical clusterability tester requires $\tilde{\Omega}(\sqrt{N})$ queries.

1. To prove the query lower bound $\rightarrow$ design a **hard** instance.
2. (Lemma 2.2): Construct two sets of graphs $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ s.t.
   - $\mathcal{G}_1^N \rightarrow$ clusterable
   - $\mathcal{G}_2^N \rightarrow \epsilon$-far from clusterable W.H.P.
3. (Lemma 2.3): $\mathcal{G}_1^N$ and $\mathcal{G}_2^N$ can not be distinguished within $\tilde{\Omega}(\sqrt{N})$ queries for any classical algorithm.

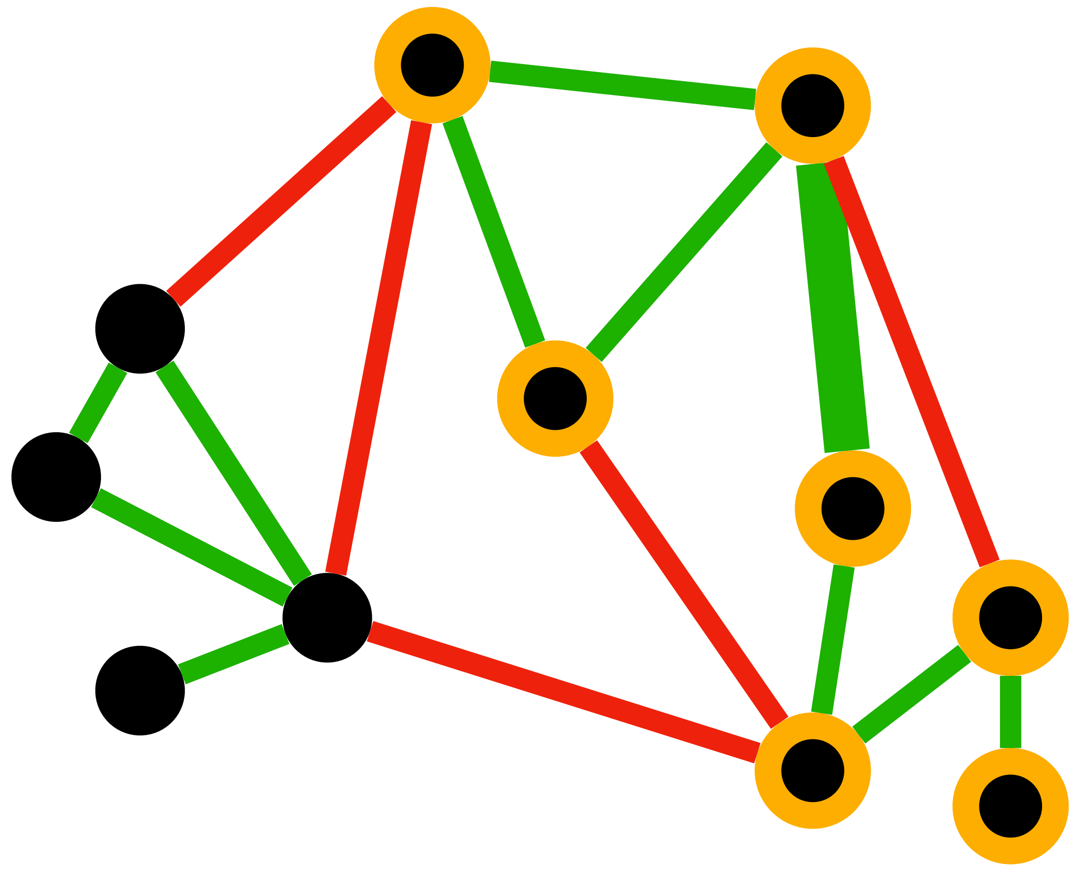
Outlines

- Motivation and our main results
- Classical clusterability testing query lower bound
- **Quantum clusterability testing algorithm**

Quantum Clusterability Testing Algorithm

Theorem 3.1

There exists a quantum algorithm for testing clusterability with $\tilde{O}(N^{1/3})$ queries.



Non-Clusterable ⇔ Finding a cycle containing one negative edge (bad cycle)

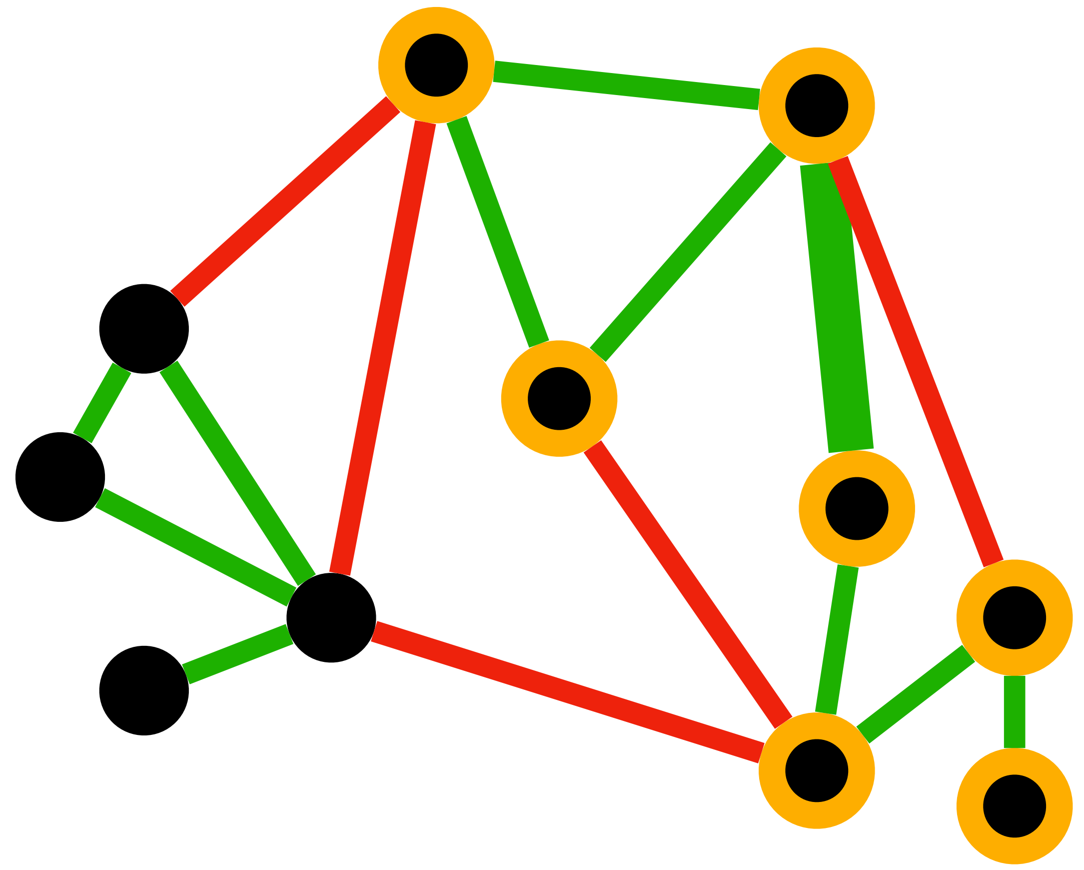
→ **Classical Clusterability Tester**

Ref: Adriaens and Apers (2021)

Quantum Clusterability Testing Algorithm

Theorem 3.1

There exists a quantum algorithm for testing clusterability with $\tilde{O}(N^{1/3})$ queries.



Non-Clusterable ⇔ Finding a cycle containing one negative edge (bad cycle)

→ **Classical Clusterability Tester**

Ref: Adriaens and Apers (2021)

Quantum Collision Finding

Ref: Ambainis, Childs, and Liu (2011)

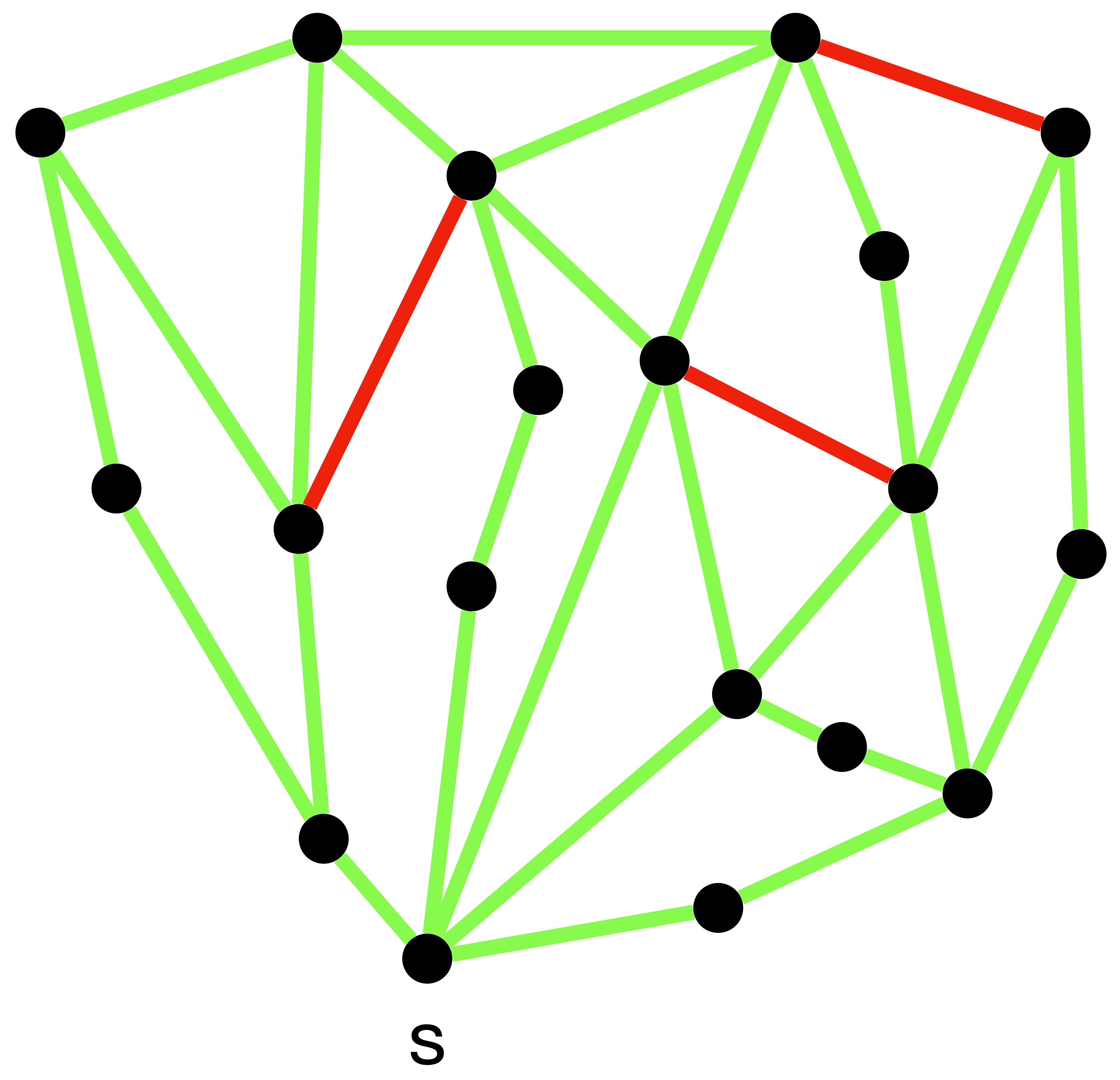
Classical Clusterability Tester

Random walk algorithm

- Select one starting vertex s
- Implement $\square$ random walks with length $\triangle$
- Bad cycle checking

  (This step can be speedup by quantum)

$G(V, E, \sigma)$



Lemma 3.2 (Adriaens and Apers (2021))

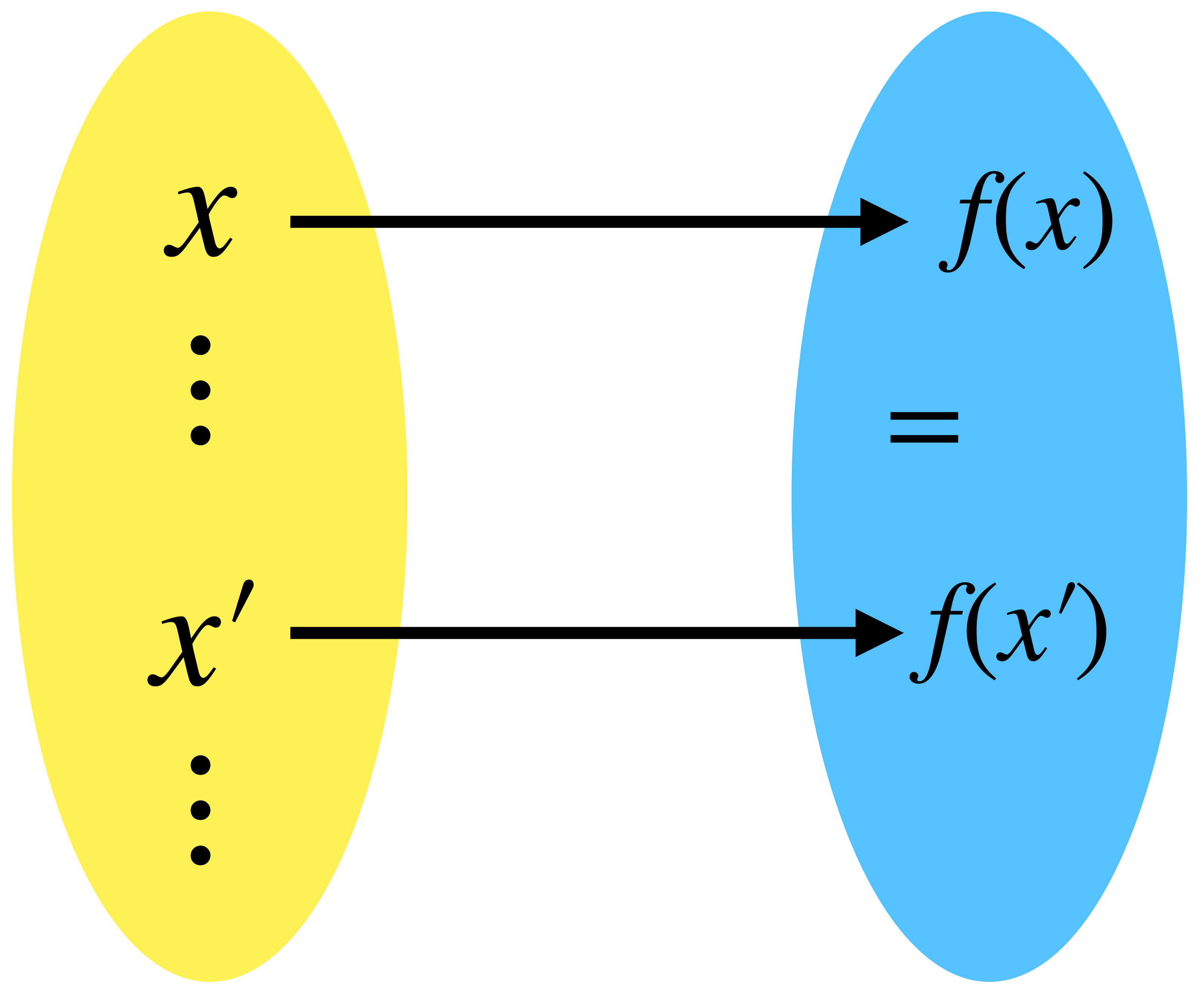
Set $\square = O(\sqrt{N})$ and $\triangle = \mathrm{poly}(\epsilon^{-1}) \rightarrow$

Finding a bad cycle iff ϵ-far from clusterable W.H.P.

Quantum Collision Finding → Quantum Clusterability Tester

Lemma 3.2 (Quantum collision finding)

Given a function $f : X \to Y$, and a symmetric binary relation $R \subseteq Y \times Y$. There exists a quantum algorithm that can find a distinct pair $x, x' \in X$ s.t. $(f(x), f(x')) \in R$ within $O\left(|X|^{2/3}\right)$ queries to f.



- Define a function $f : (i, j) \longmapsto (v, v_{neb})$
- $((v, v_{neb}), (v', v'_{neb})) \in R \Leftrightarrow$ find a bad cycle

Conclusion

We confirmed the quantum advantage on testing clusterability.

	Bipartiteness	Expander	Clusterability
Classical UB	$\tilde{O}(\sqrt{N})$ [1]	$\tilde{O}(\sqrt{N})$ [2]	$\tilde{O}(\sqrt{N})$ [3]
Classical LB	$\tilde{\Omega}(\sqrt{N})$ [4]	$\tilde{\Omega}(\sqrt{N})$ [5]	$\tilde{\Omega}(\sqrt{N})$
Quantum UB	$\tilde{O}(N^{1/3})$ [6]	$\tilde{O}(N^{1/3})$ [6]	$\tilde{O}(N^{1/3})$

Thanks for the listening