A Tolerant Independent Set Tester

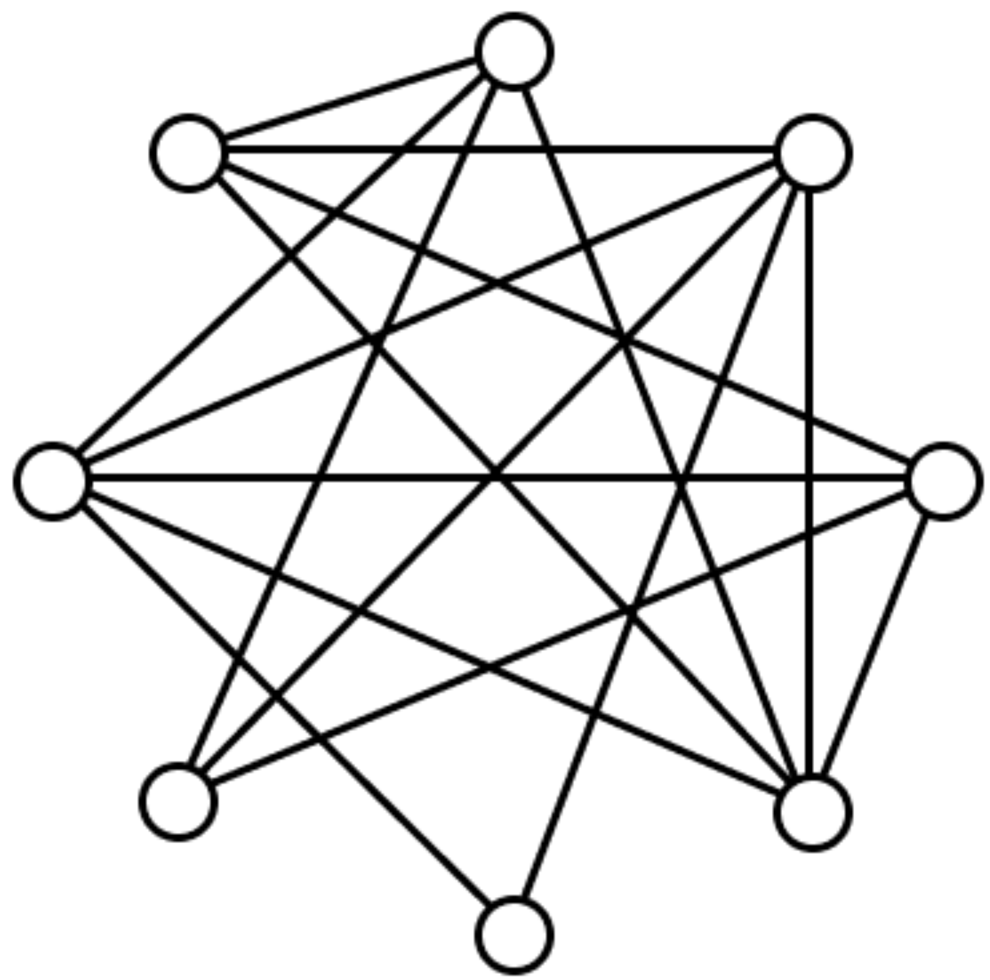
Cameron Seth

STOC 2025

UNIVERSITY OF WATERLOO

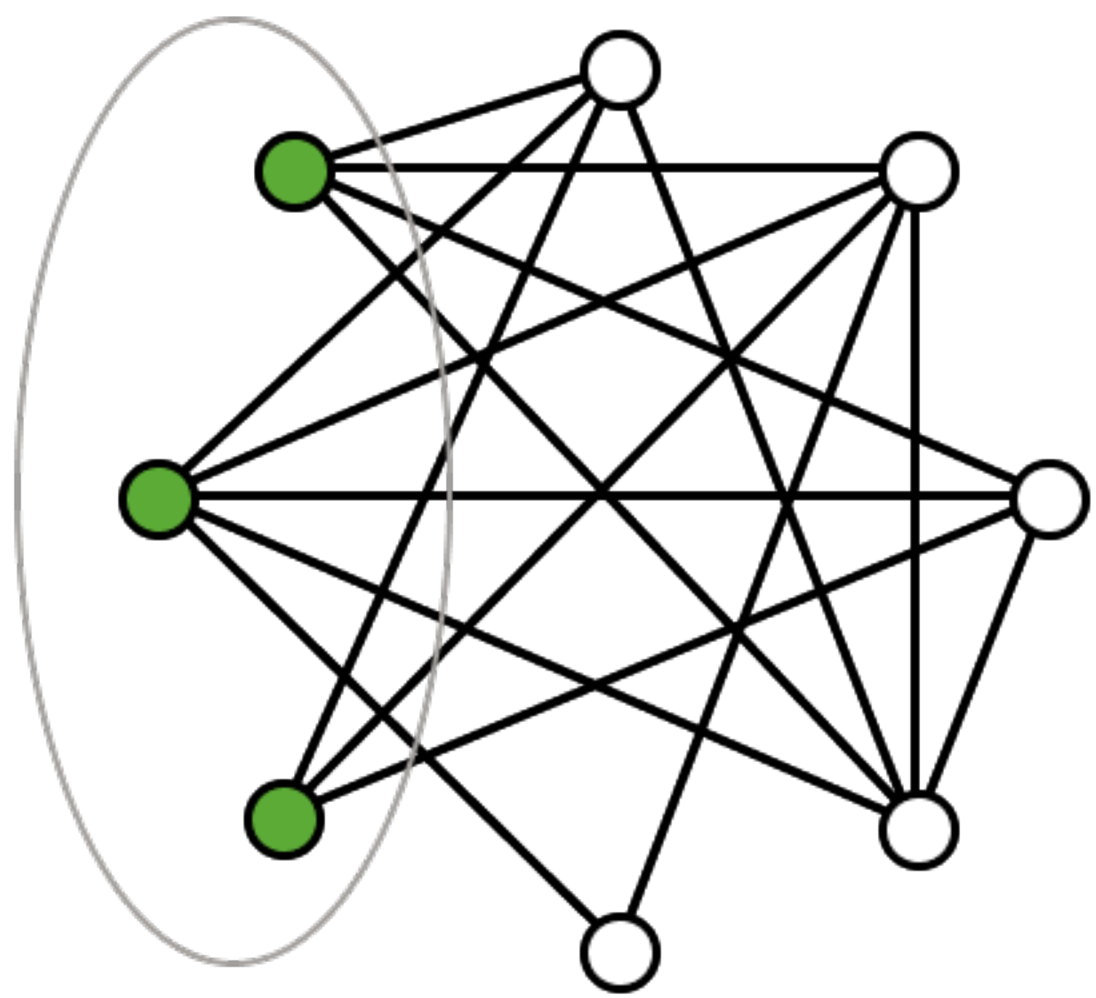
Independent Set Problem

Given a graph G on n vertices, does it have an independent set of size ρn?



Independent Set Problem

Given a graph G on n vertices, does it have an independent set of size ρn?

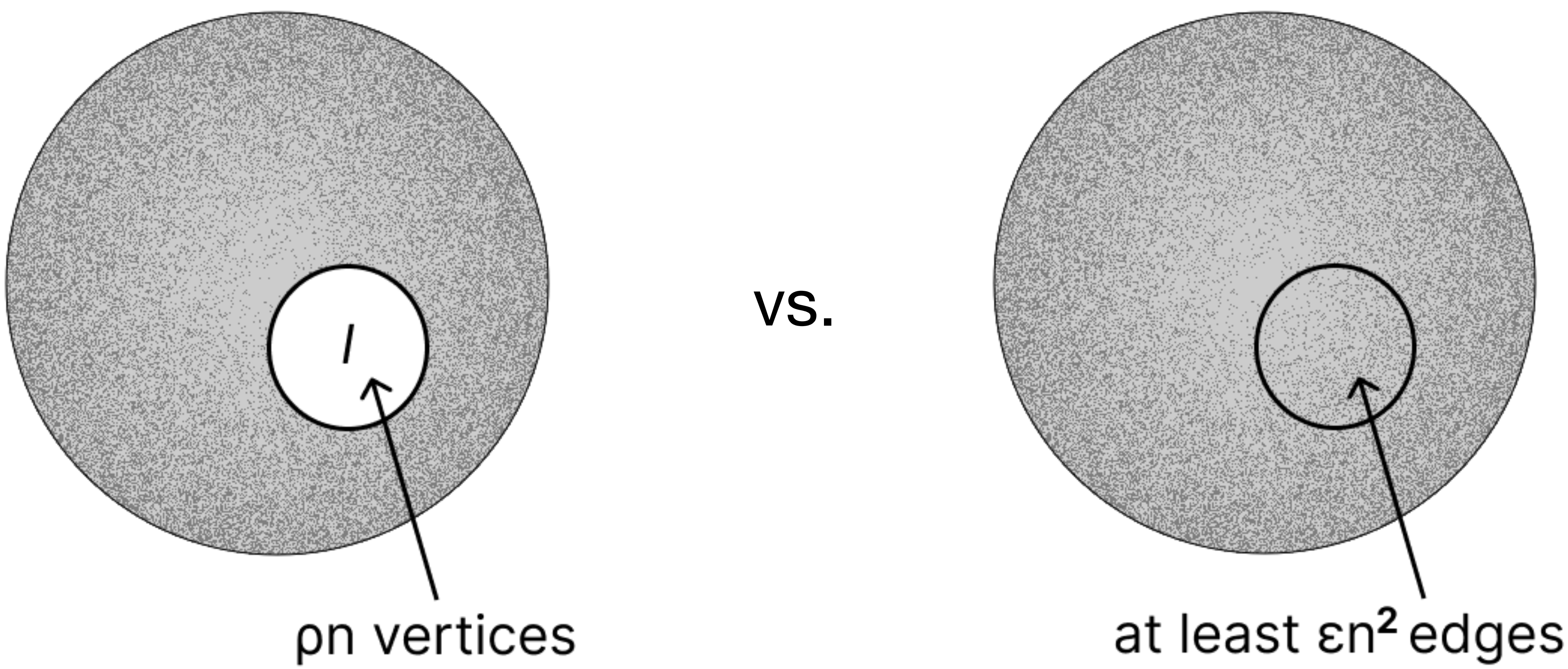


Testing Independent Sets

Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn has at least ϵn^2 edges



Theorem: Inspecting a random subgraph on $\tilde{O}(\rho/\epsilon^4)$ vertices is sufficient for distinguishing between (i) and (ii) (whp).

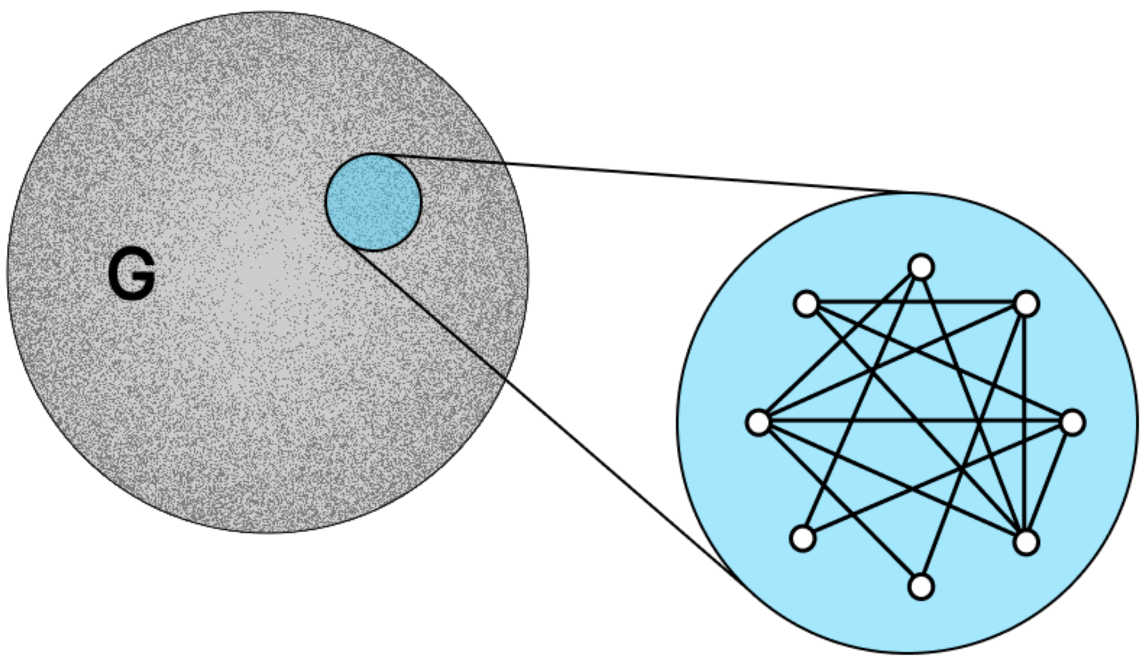
[Goldreich, Goldwasser, Ron '98]

Definitions

An ϵ-tester for the ρn-independent set property is an algorithm that samples a set S of s random vertices, examines the induced subgraph $G[S]$, and distinguishes between the cases (with high probability):

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn has at least ϵn^2 edges (ϵ—far)



s is the sample complexity of the tester.

Testing Independent Sets

Theorem: There exists an $\epsilon-$tester for the ρn independent set property with sample complexity $\tilde{O}(\rho/\epsilon^4)$.

[Goldreich, Goldwasser, Ron ’98]

Testing Independent Sets

Theorem: There exists an $\epsilon-$tester for the ρn independent set property with sample complexity $\tilde{O}(\rho/\epsilon^4)$.

[Goldreich, Goldwasser, Ron '98]

Theorem: There exists an $\epsilon-$tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$, [Blais, Seth '23]

and any such tester has sample complexity $\Omega(\rho^3/\epsilon^2)$.

[Feige, Langberg, Schechtman '04]

Weakness with Standard Testing

Standard Testing Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn has at least ϵn^2 edges

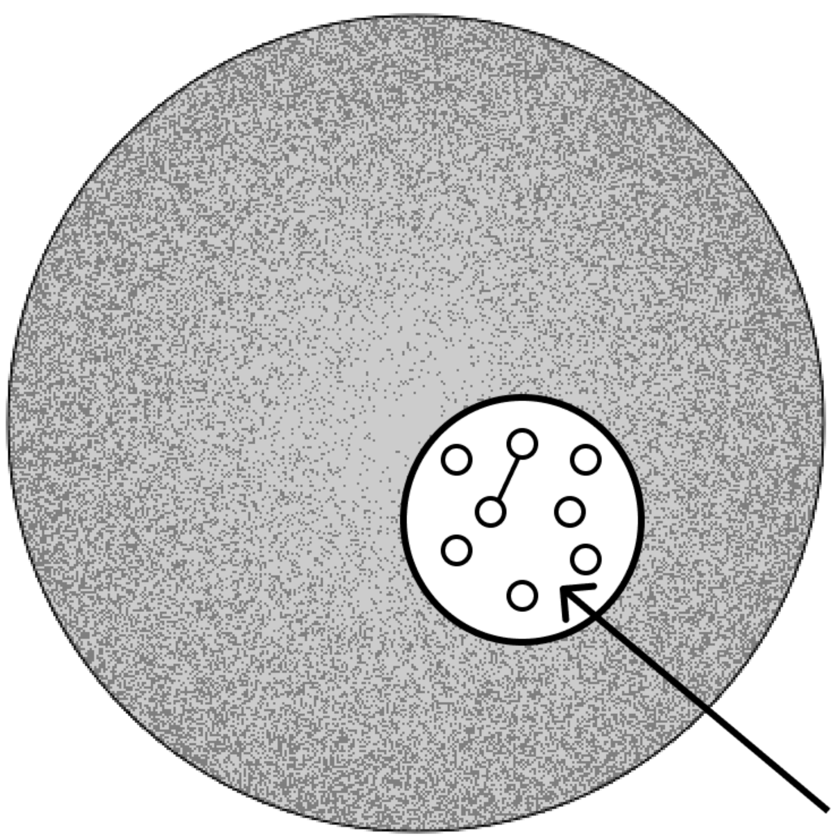
Weakness with Standard Testing

Standard Testing Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn has at least ϵn^2 edges

Question: What if input graph is the following: start with the complete graph and plant a set $U \subset V$ with $|U| = \rho n$ such that $G[U]$ has exactly one edge?



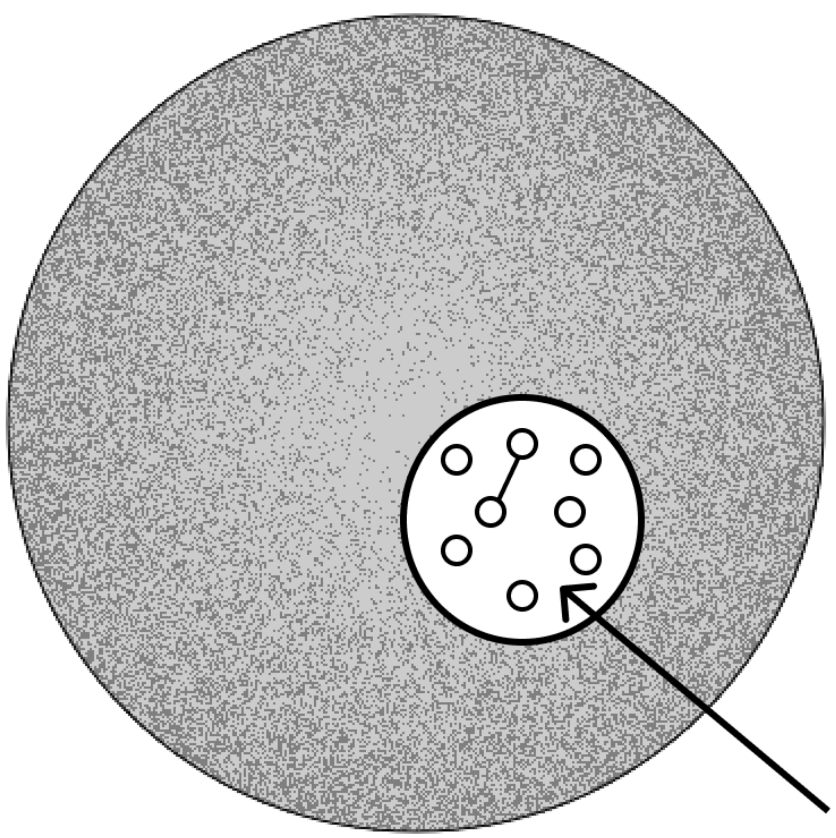
Weakness with Standard Testing

Standard Testing Problem: Distinguish between the cases:

(i) G has a ρn independent set, and

(ii) every induced subgraph of size ρn has at least ϵn^2 edges

Question: What if input graph is the following: start with the complete graph and plant a set $U \subset V$ with $|U| = \rho n$ such that $G[U]$ has exactly one edge?



ρn vertices with 1 edge

Answer: Testing algorithms have no guarantee on this type of input!

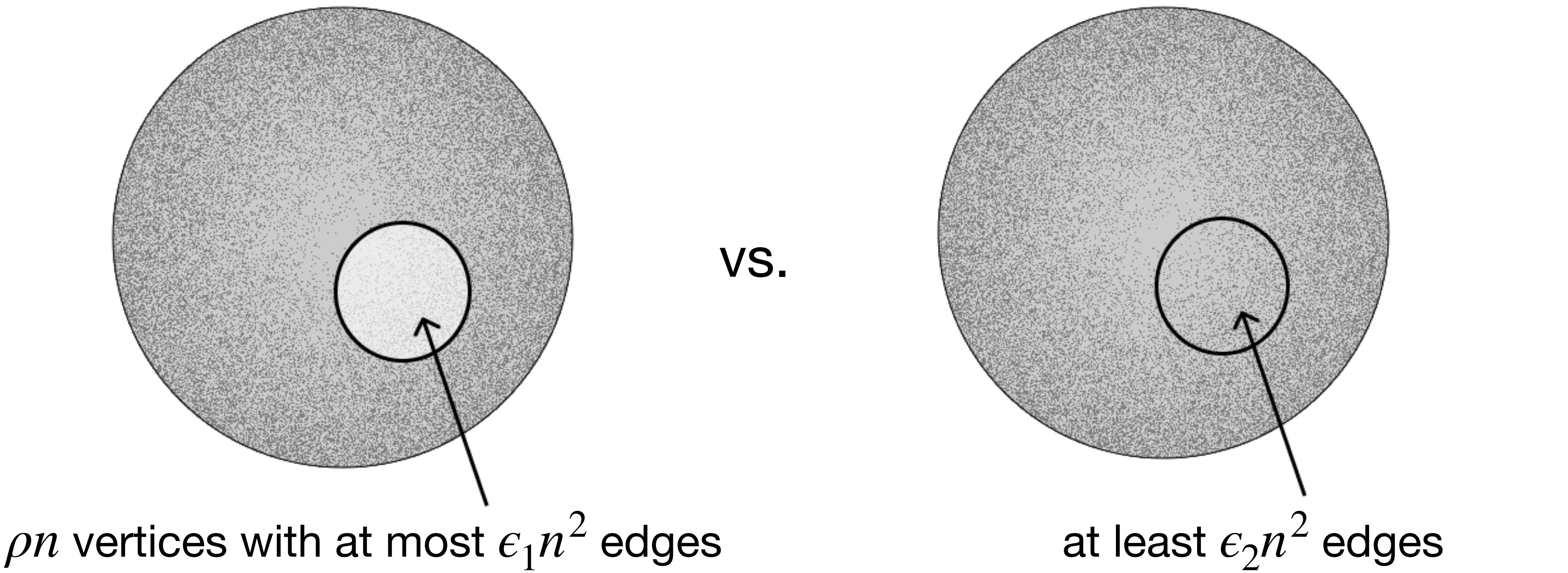
Ideally we would like the algorithm to accept this type of graph.

Tolerant Testing Independent Sets

Problem: For $\epsilon_1 < \epsilon_2$, distinguish between the cases:

(i) G has an induced subgraph of size ρn with fewer than $\epsilon_1 n^2$ edges (ϵ_1-close)

(ii) Every induced subgraph of size ρn has at least $\epsilon_2 n^2$ edges (ϵ_2-far)



An algorithm that, with high probability, distinguishes between (i) and (ii) is called an $(\epsilon_1, \epsilon_2)-$tester.

[Parnas, Ron, Rubinfeld '06]

Tolerant Testing Independent Sets

Problem: For $\epsilon_1 < \epsilon_2$, distinguish between the cases:

(i) G has an induced subgraph of size ρn with fewer than $\epsilon_1 n^2$ edges ($\epsilon_1 -$close)

(ii) Every induced subgraph of size ρn has at least $\epsilon_2 n^2$ edges ($\epsilon_2 -$far)

An algorithm that, with high probability, distinguishes between (i) and (ii) is called an $(\epsilon_1, \epsilon_2)-$tester.

Tolerant Testing Independent Sets

Problem: For $\epsilon_1 < \epsilon_2$, distinguish between the cases:

(i) G has an induced subgraph of size ρn with fewer than $\epsilon_1 n^2$ edges (ϵ_1-close)

(ii) Every induced subgraph of size ρn has at least $\epsilon_2 n^2$ edges (ϵ_2-far)

An algorithm that, with high probability, distinguishes between (i) and (ii) is called an $(\epsilon_1, \epsilon_2)-$tester.

Remarks:

- Generalizes the standard testing problem ($\epsilon_1 = 0$)
- In general ϵ_1 may be a function of ϵ_2
- In some other settings (bounded degree model, boolean strings), there is exponential gap between the query complexity of $\epsilon-$testing and $(\tilde{\Theta}(\epsilon), \epsilon)-$tolerant testing.

[Fischer, Fortnow '05]

[Goldreich, Wigderson '22]

Main Result

Theorem: There is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

[This Work]

Main Result

Theorem: There is a $\left(\frac{\epsilon}{\mathrm{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

[This Work]

Remarks:

- Matches the (optimal) sample complexity bound for $\epsilon-$testing
- Generalizes container method approach of Blais, Seth '23
- Best prior result is from a general result for all graph partition properties, which gives sample complexity of roughly $(1/\epsilon)^{12}$

[Fiat, Ron '21]

[Goldreich, Goldwasser, Ron '98]

Outline of Talk

- Main technique to prove theorem: graph container method
    - What is the container method?
    - How to use the container method to prove testing results

[Blais, Seth ’23]

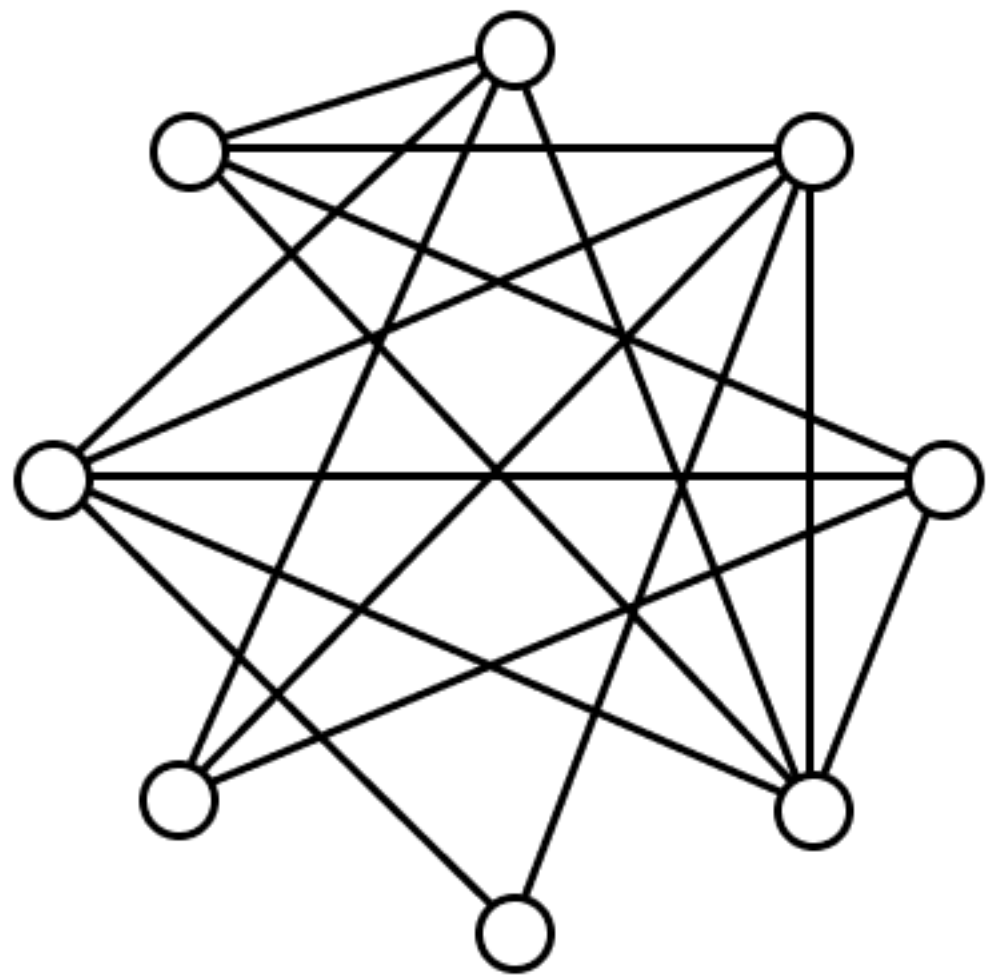
- A new graph container lemma for sparse subgraphs

[This Work]

    - Proof ideas of new container lemma
    - Another application of new container lemma

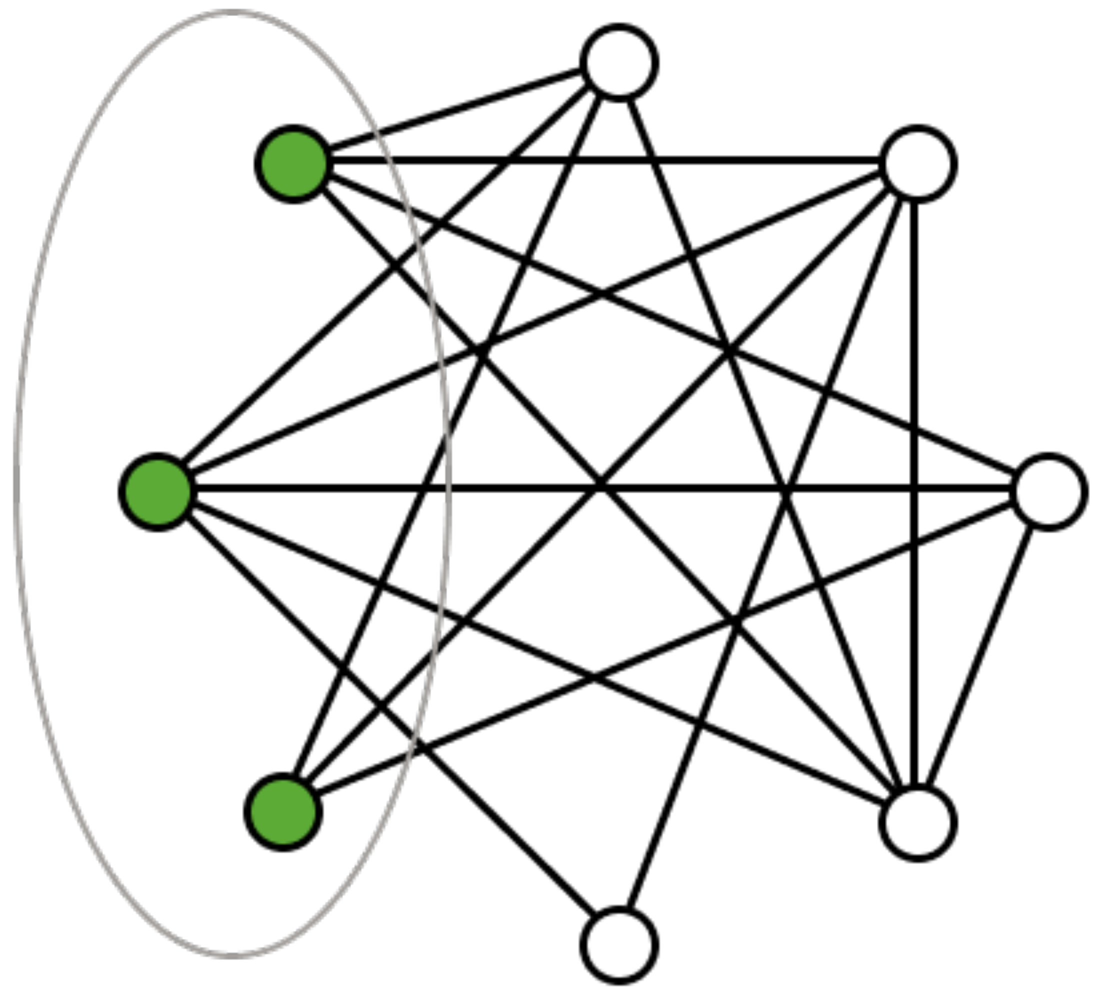
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



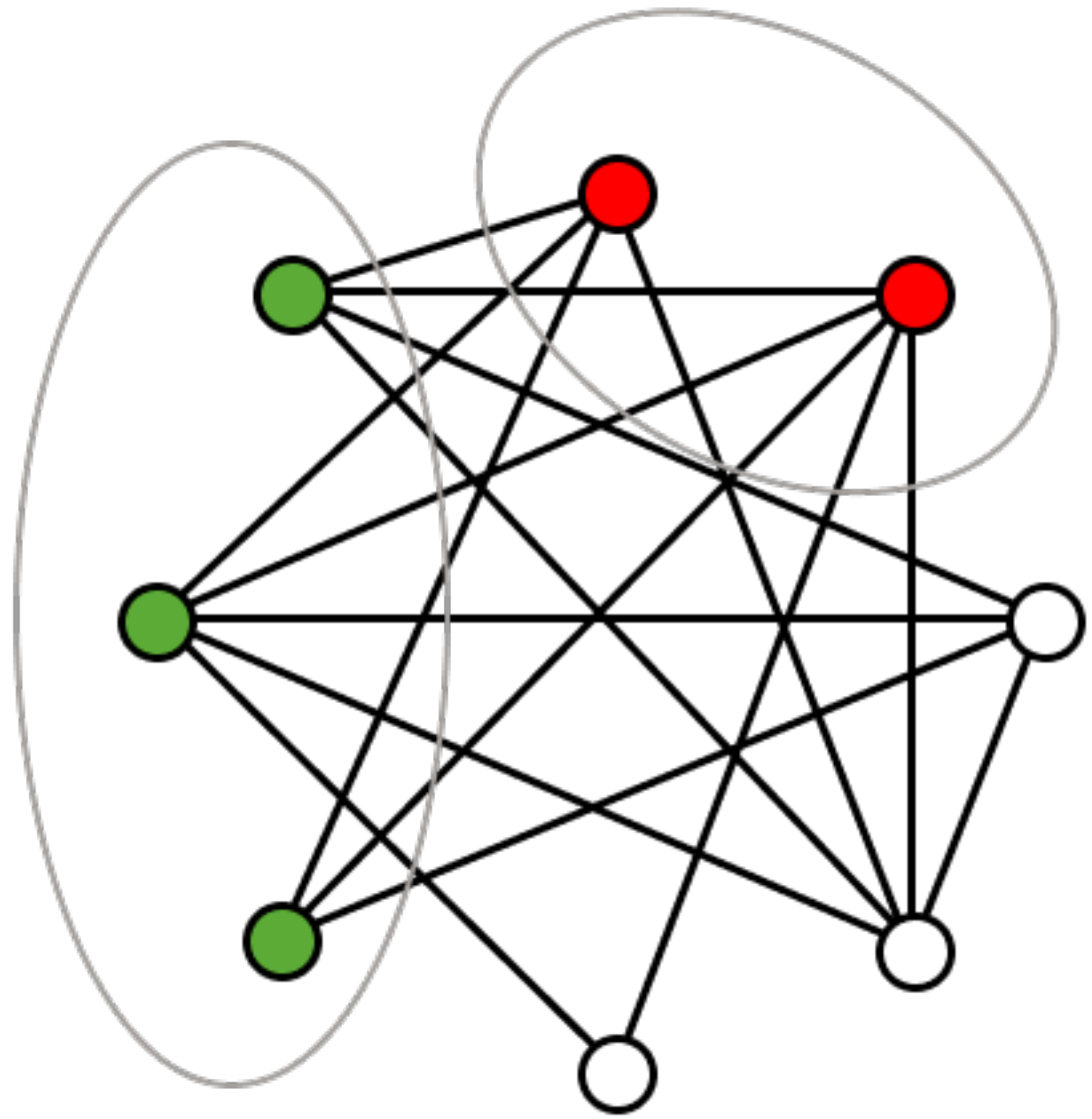
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



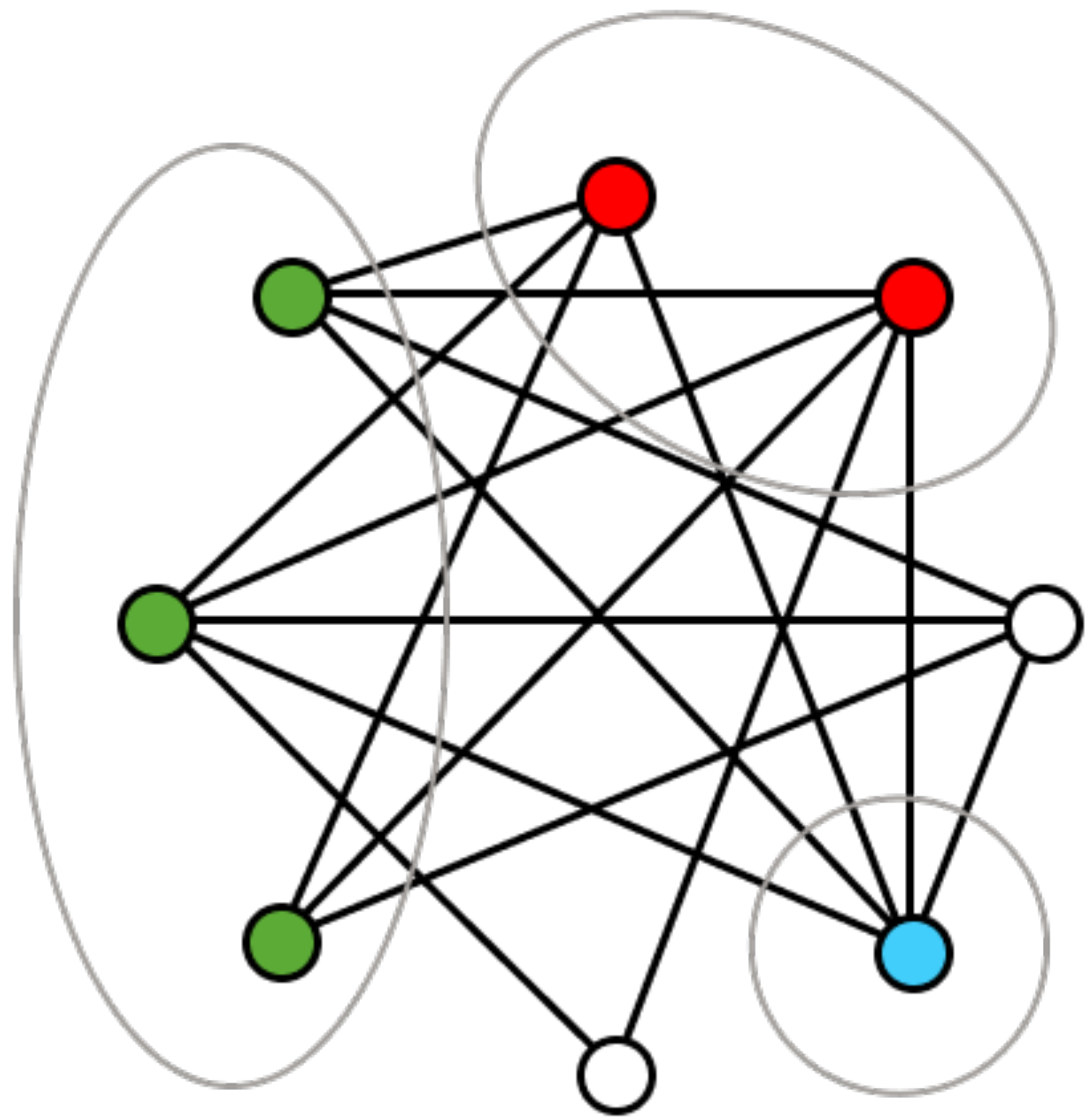
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



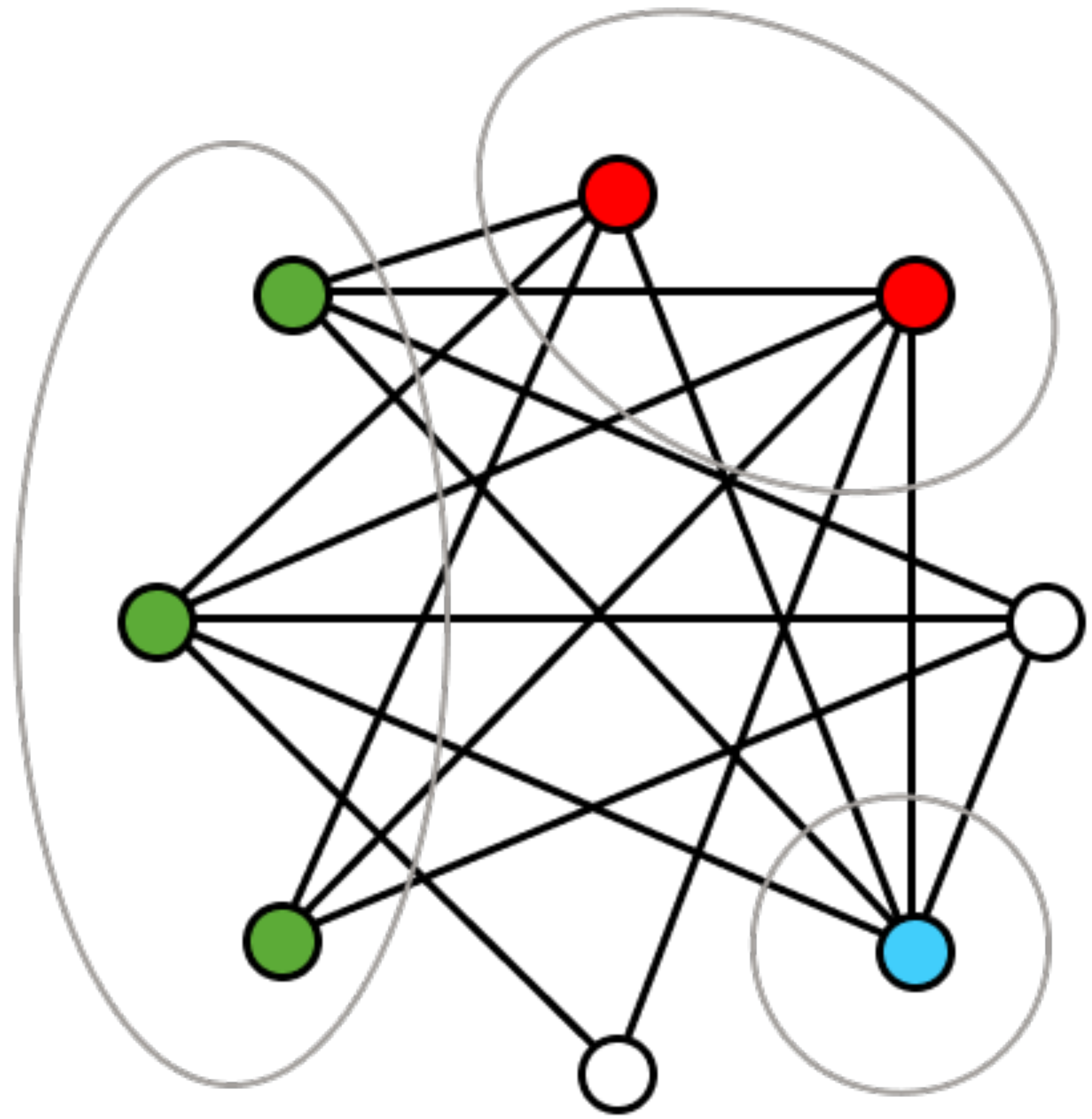
What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



What is the Container Method?

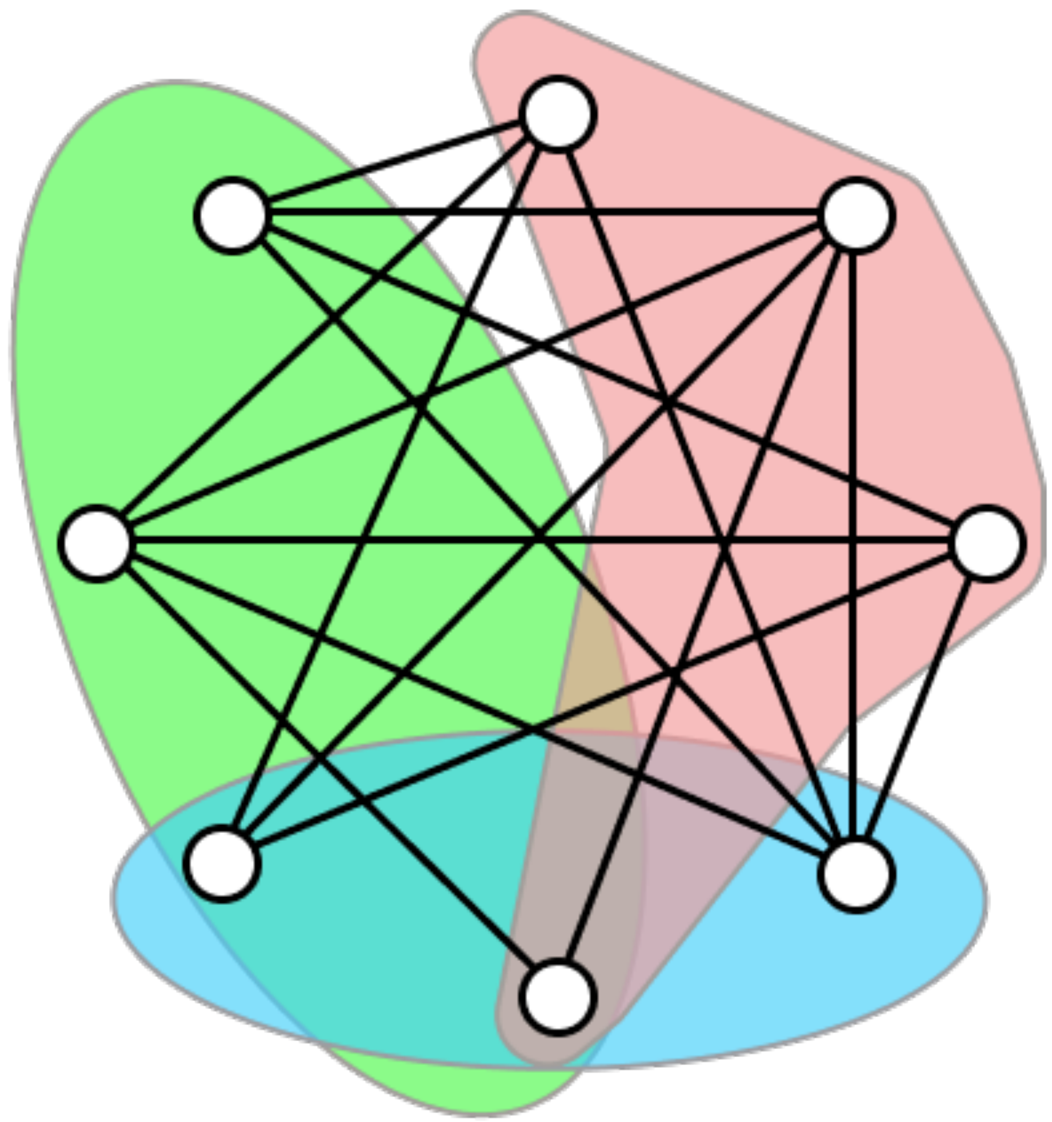
Answer: A tool for characterizing independent sets in some graphs.



Informal Idea: For any graph satisfying some “nice” conditions, all independent sets in the graph can be covered by a small number of containers (each container is a subset of vertices).

What is the Container Method?

Answer: A tool for characterizing independent sets in some graphs.



Informal Idea: For any graph satisfying some "nice" conditions, all independent sets in the graph can be covered by a small number of containers (each container is a subset of vertices).

An Initial Graph Container Lemma

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim (1 - \epsilon)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

An Initial Graph Container Lemma

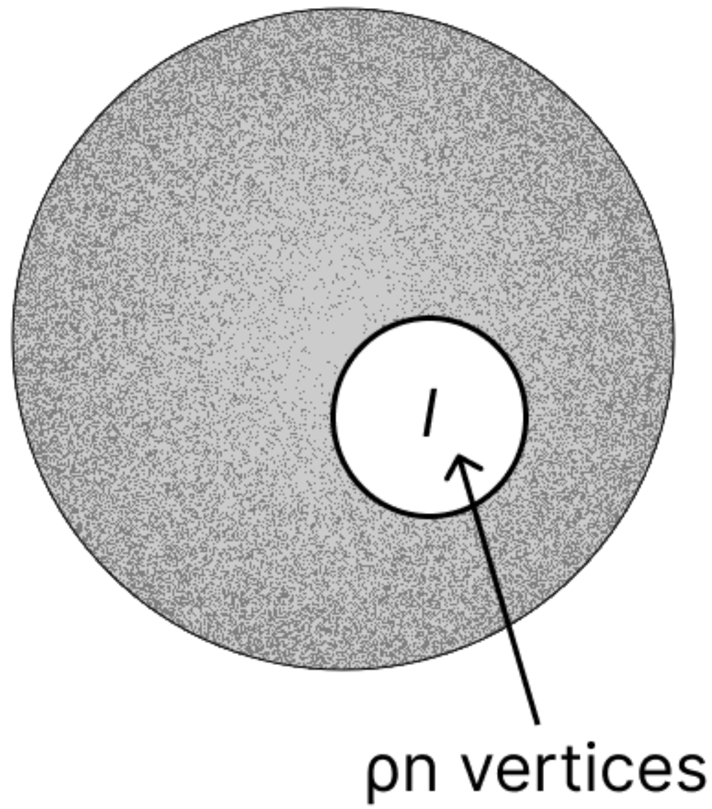
Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| \lesssim (1-\epsilon)\rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

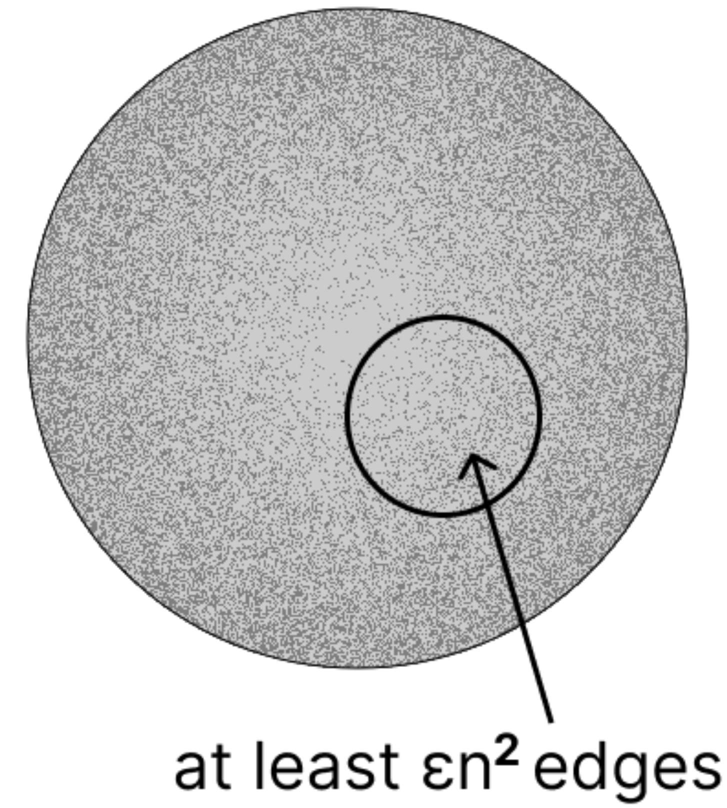
[Kleitman, Winston '82]

Note: for survey of combinatorial applications see "Counting Indepedent Sets in Graphs" by Samotij or "The method of hypergraph containers" by Balogh, Morris, and Samotij.

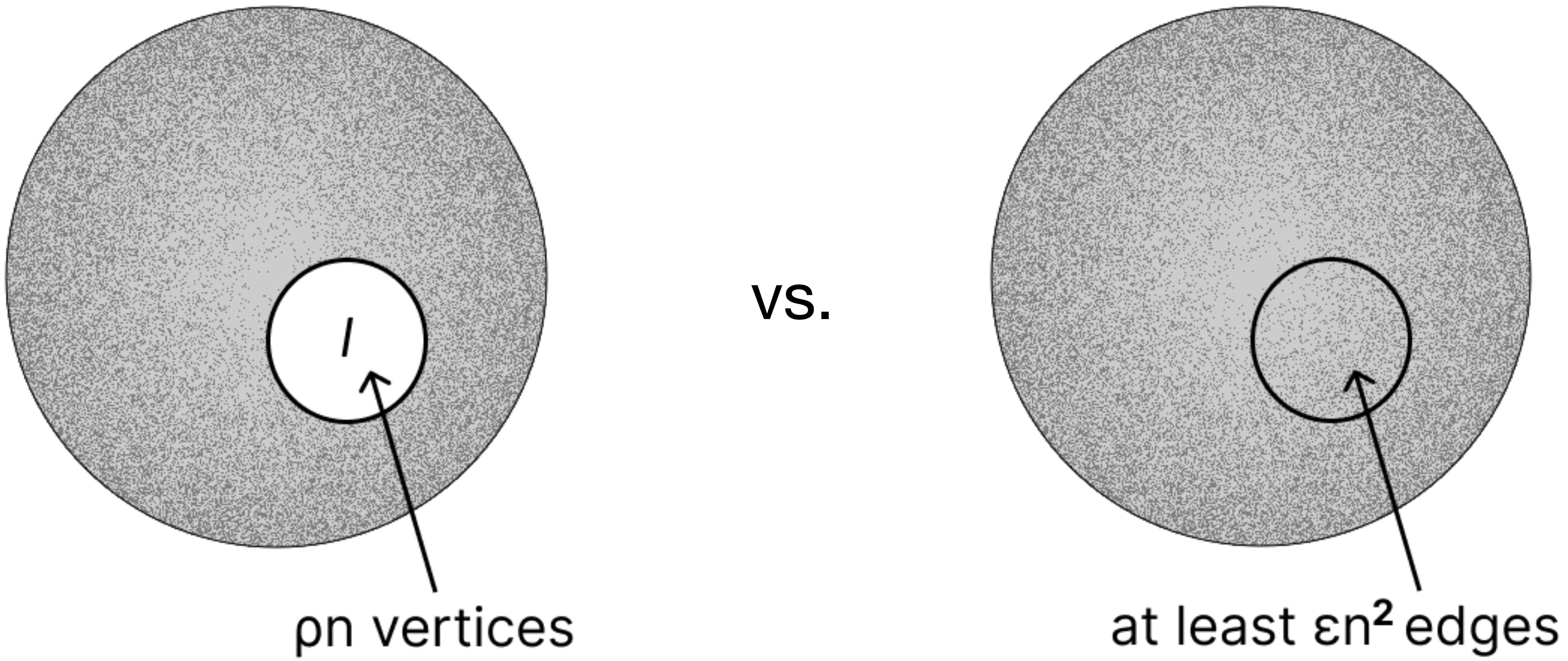
Application to Non-Tolerant Testing



vs.



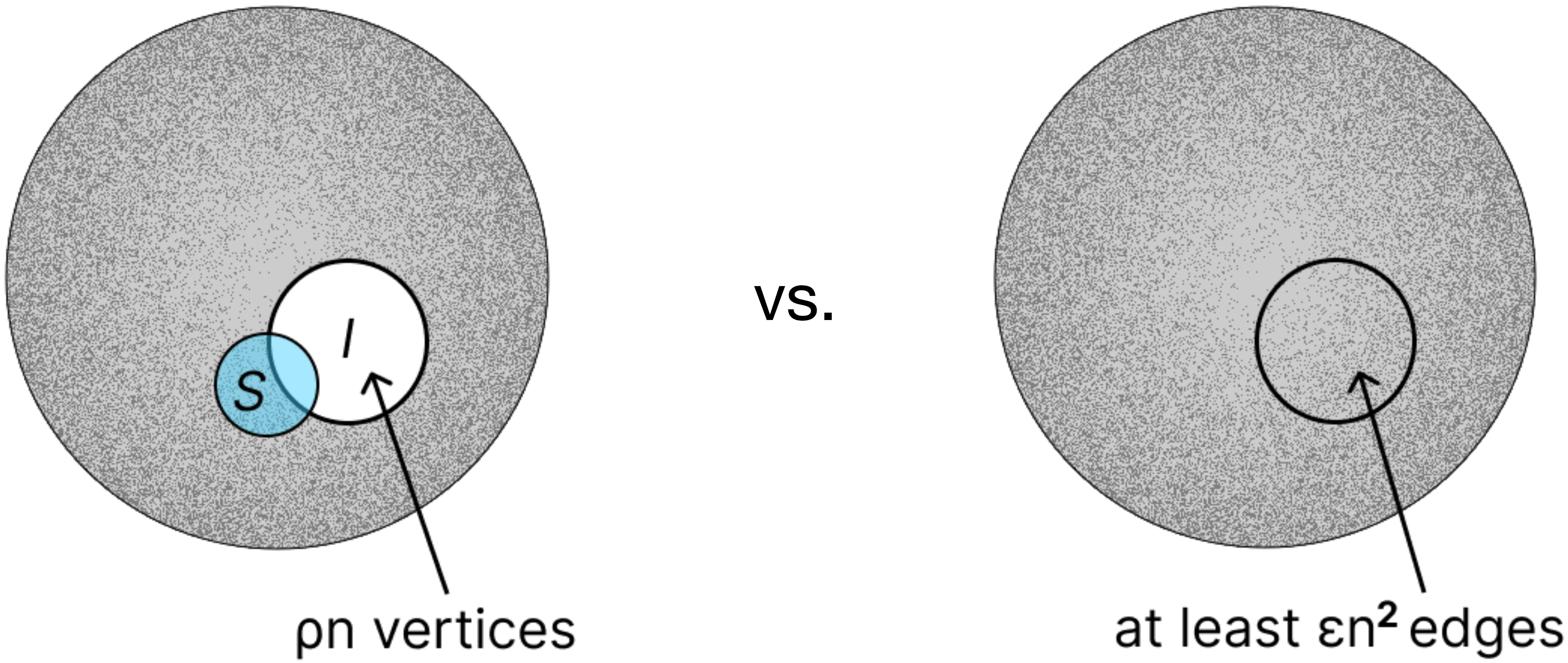
Application to Non-Tolerant Testing



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

[Blais, Seth ’23]

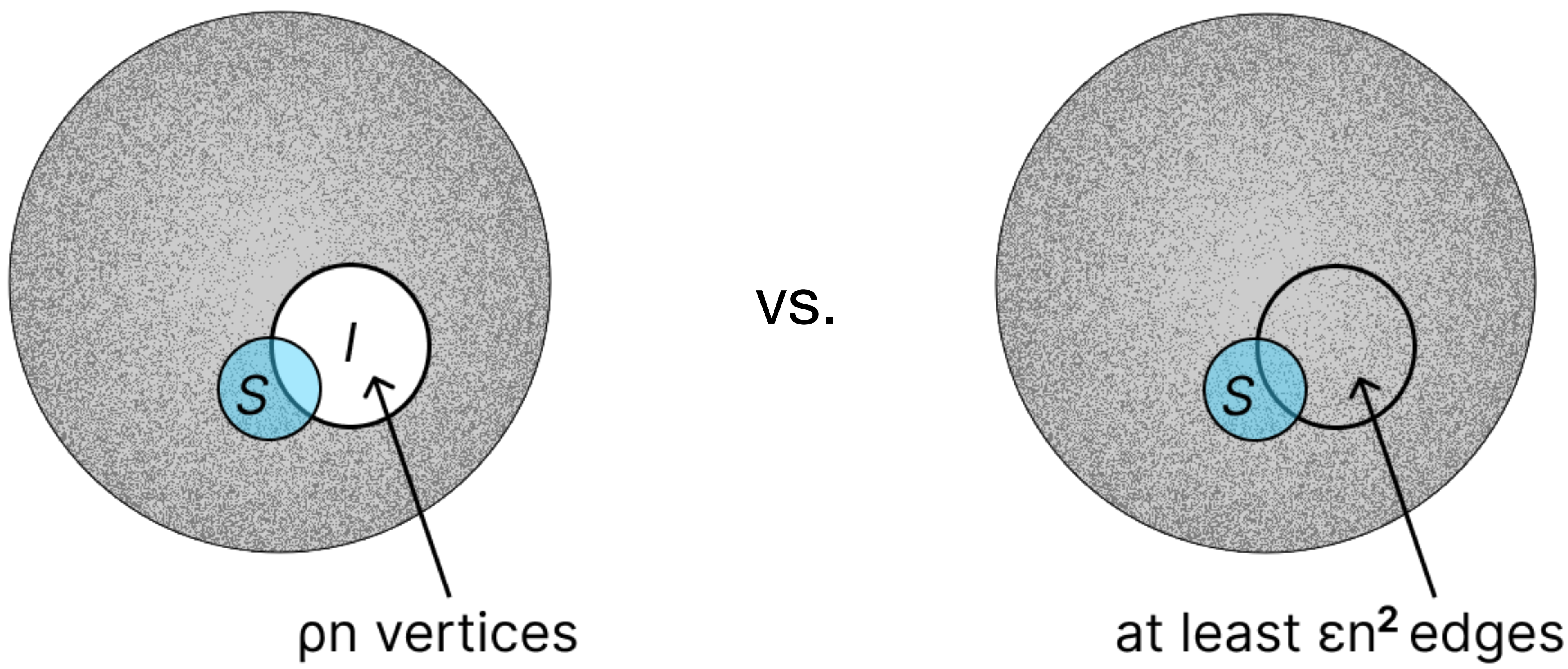
Application to Non-Tolerant Testing



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

[Blais, Seth ’23]

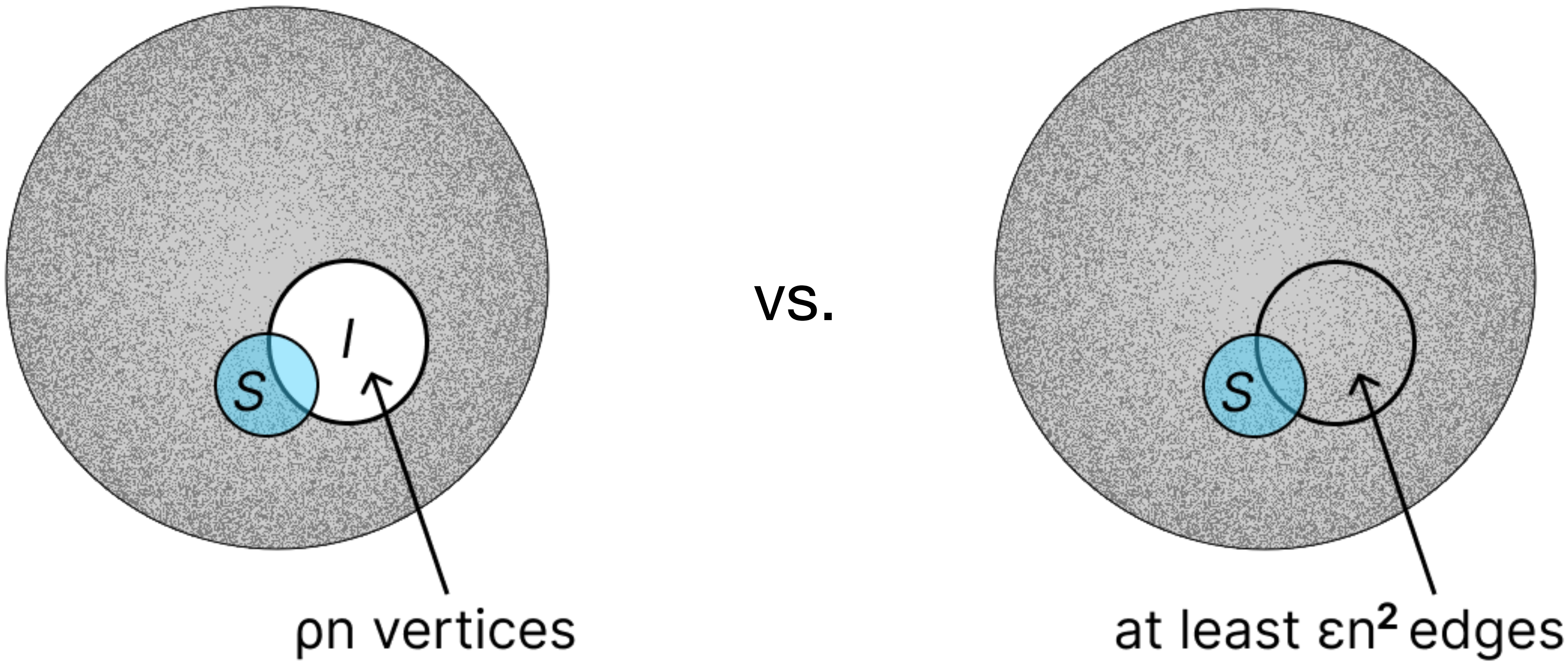
Application to Non-Tolerant Testing



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

[Blais, Seth '23]

Application to Non-Tolerant Testing



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has a ρs independent set.

[Blais, Seth '23]

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

* [Blais, Seth '23] proved a stronger container lemma compared to the lemma of [Kleitman, Winston '82]

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$\Pr[S$ contains a ρs independent set$]$

* [Blais, Seth '23] proved a stronger container lemma compared to the lemma of [Kleitman, Winston '82]

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

* [Blais, Seth '23] proved a stronger container lemma compared to the lemma of [Kleitman, Winston '82]

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

$$\Pr[S \text{ contains a } \rho s \text{ independent set}]$$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

$$\lesssim \binom{n}{1/\epsilon} e^{-\epsilon^2 s}$$

* [Blais, Seth '23] proved a stronger container lemma compared to the lemma of [Kleitman, Winston '82]

Application of Container Lemma

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs independent set with only small probability.

Use Container Lemma*: Every independent set is contained in a container! Each container is of size at most $|C| < (1-\epsilon)\rho n$ and there are at most $\binom{n}{1/\epsilon}$ of them.

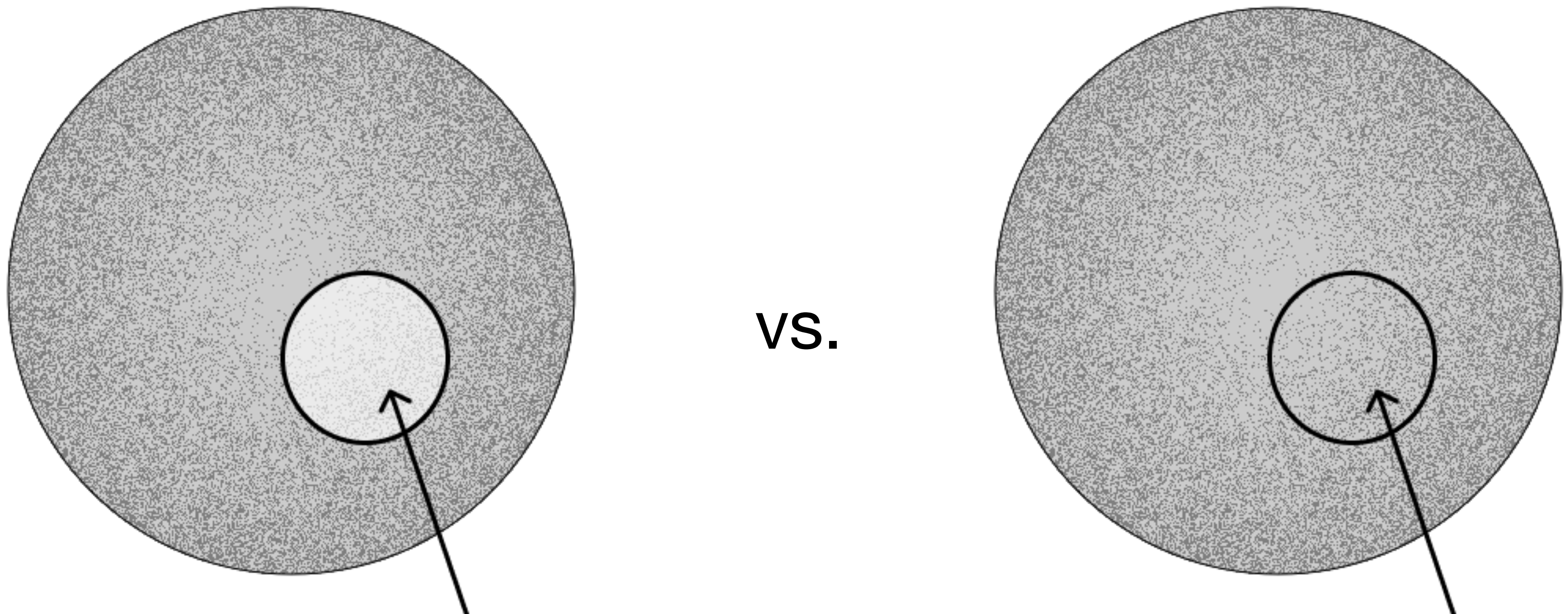
$\Pr[S$ contains a ρs independent set$]$

$$\leq \sum_{C \in \mathscr{C}} \Pr[S \text{ contains at least } \rho s \text{ vertices from } C]$$

$$\lesssim \binom{n}{1/\epsilon} e^{-\epsilon^2 s} \quad \leq 1/3, \text{ as long as } s \gtrsim \frac{\log n}{\epsilon^3}.$$

* [Blais, Seth '23] proved a stronger container lemma compared to the lemma of [Kleitman, Winston '82]

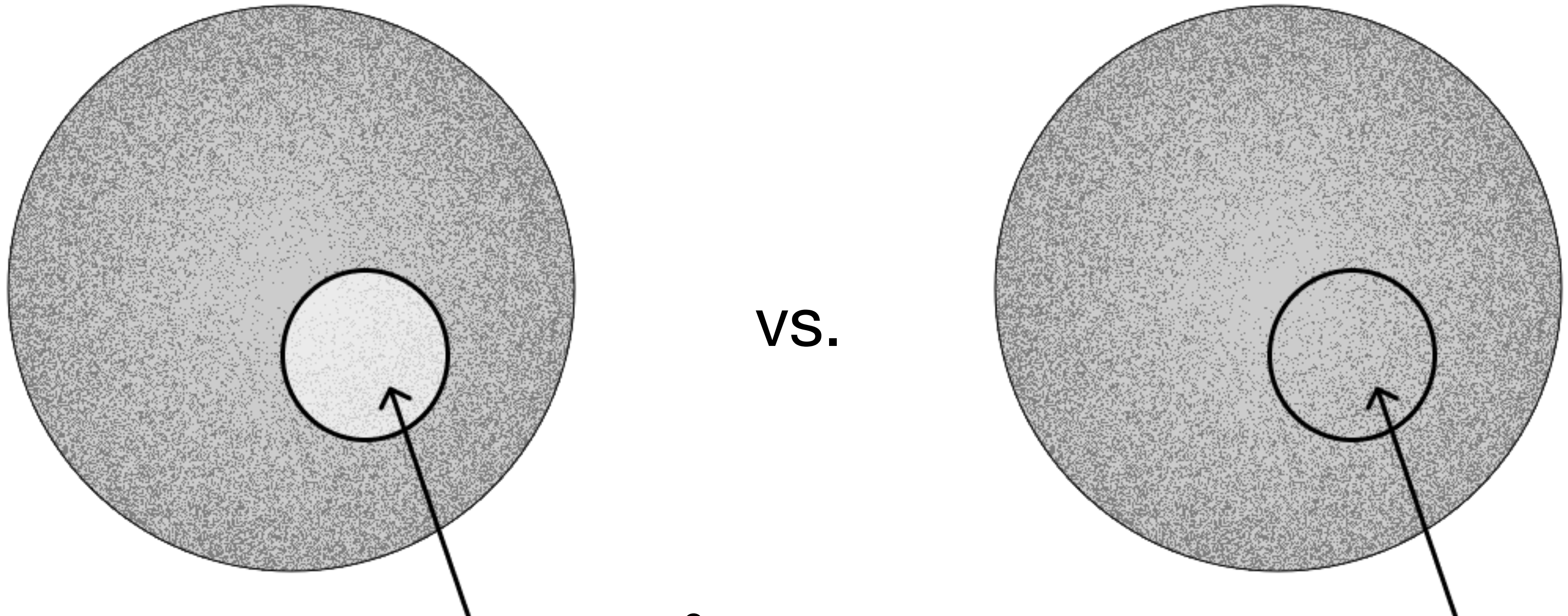
Towards a Tolerant Tester



ρn vertices with at most $\frac{\epsilon}{\text{polylog}(1/\epsilon)} n^2$ edges

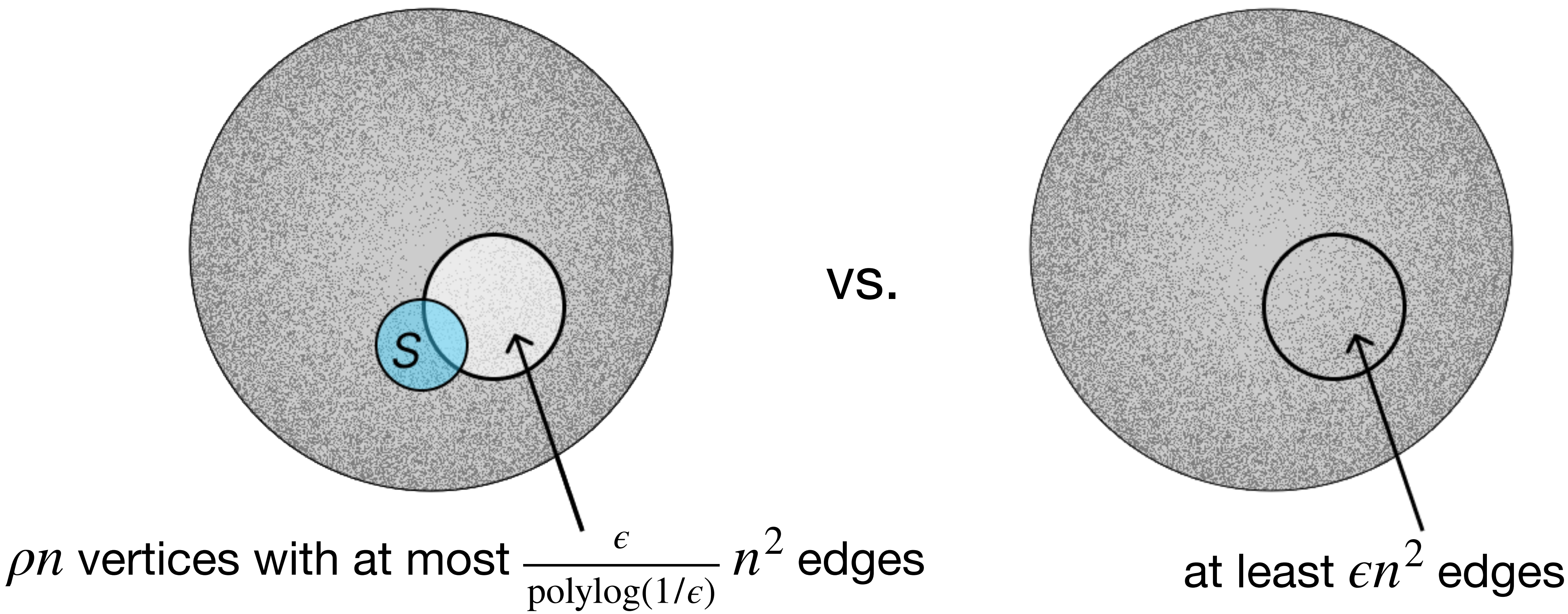
at least ϵn^2 edges

Towards a Tolerant Tester



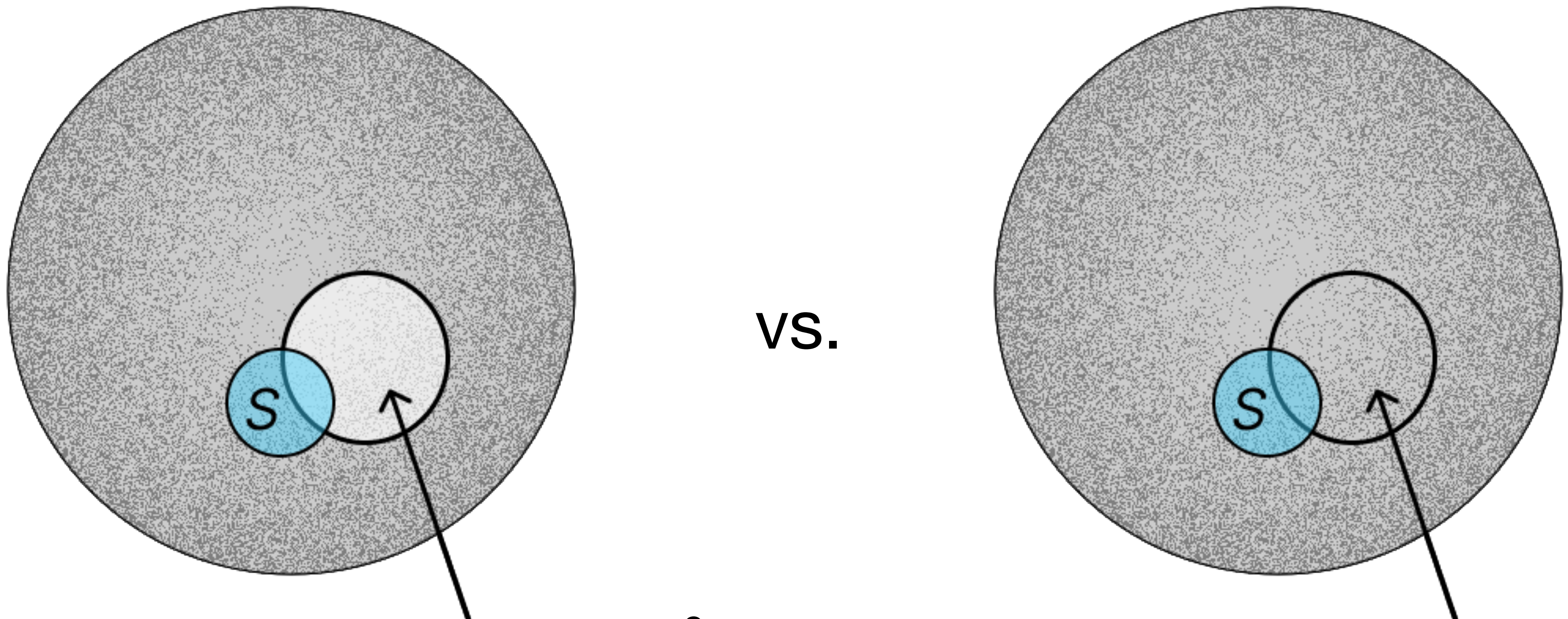
Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has an induced subgraph of size ρs with fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} s^2$ edges.

Towards a Tolerant Tester



Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has an induced subgraph of size ρs with fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} s^2$ edges.

Towards a Tolerant Tester

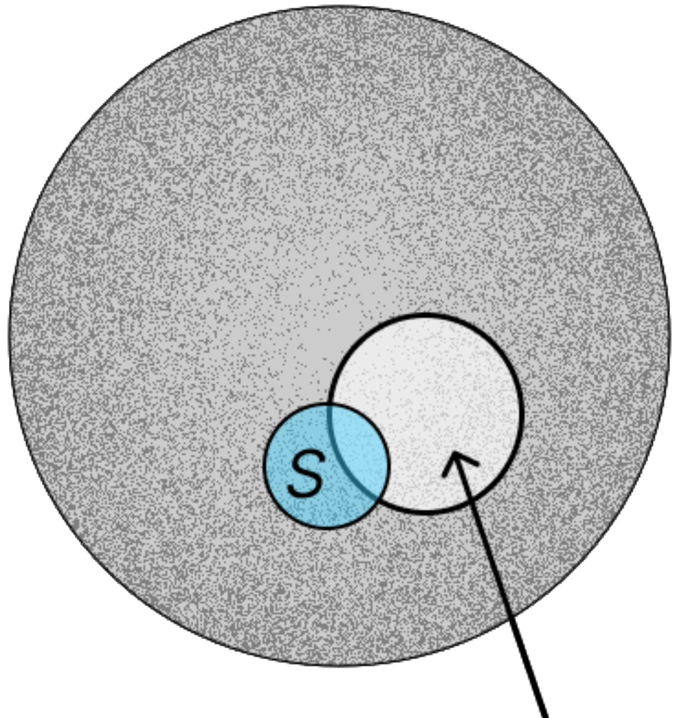


ρn vertices with at most $\frac{\epsilon}{\text{polylog}(1/\epsilon)}\, n^2$ edges

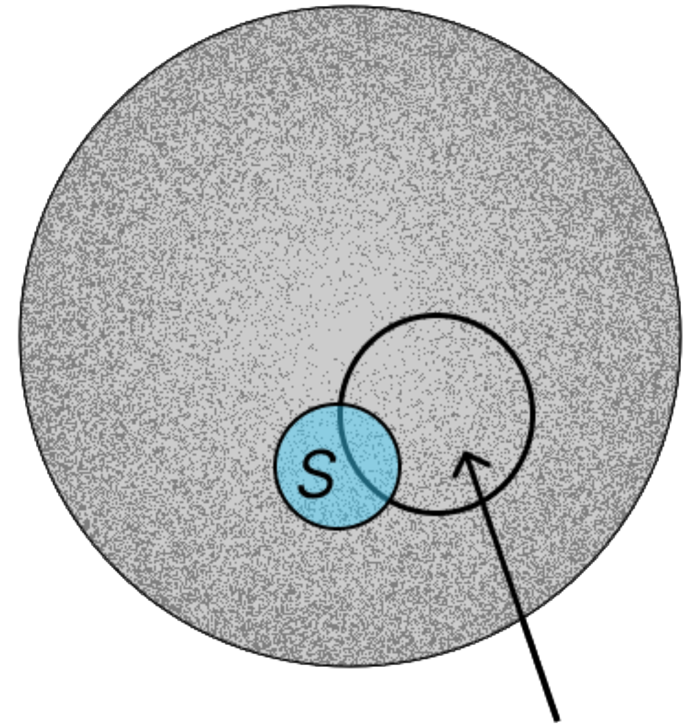
at least ϵn^2 edges

Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has an induced subgraph of size ρs with fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)}\, s^2$ edges.

Towards a Tolerant Tester



vs.



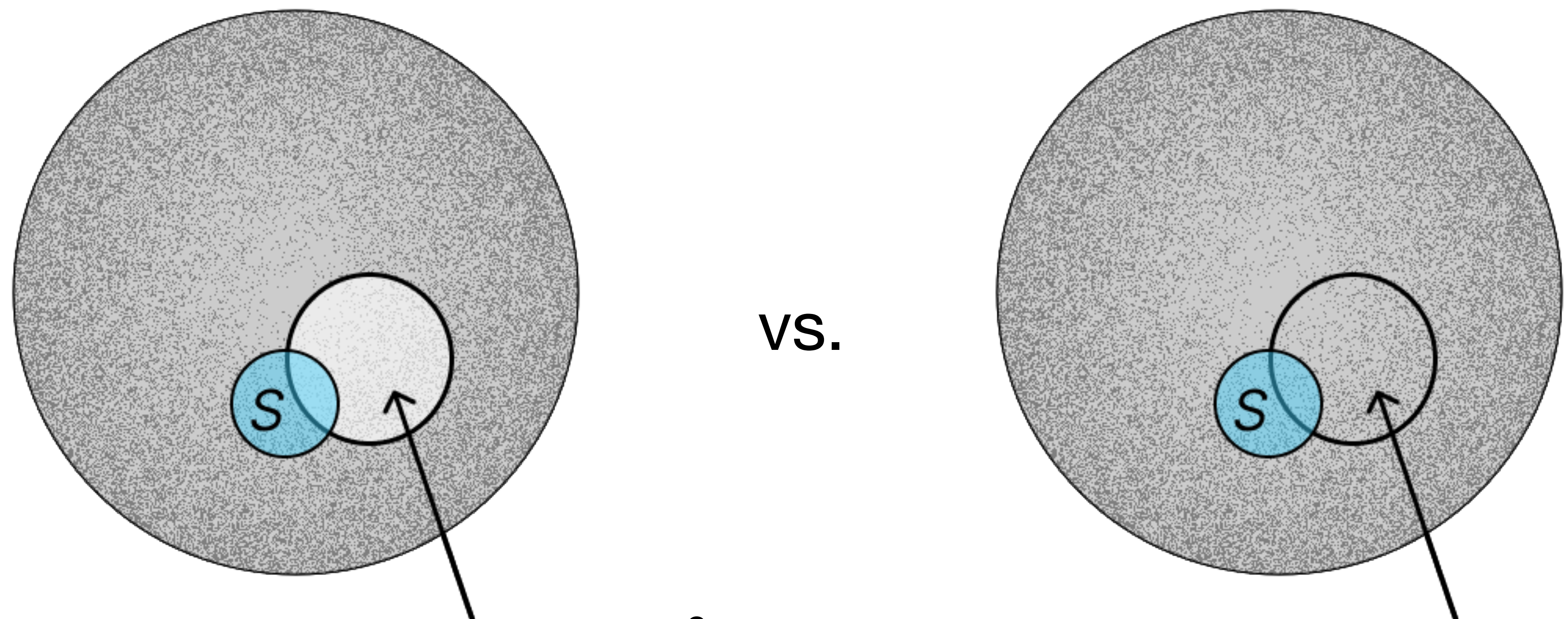
ρn vertices with at most $\frac{\epsilon}{\text{polylog}(1/\epsilon)} n^2$ edges

at least ϵn^2 edges

Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has an induced subgraph of size ρs with fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} s^2$ edges.

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs sparse set with only small probability.

Towards a Tolerant Tester



ρn vertices with at most $\frac{\epsilon}{\text{polylog}(1/\epsilon)} n^2$ edges

at least ϵn^2 edges

Testing Algorithm: Take a random sample S of $s \sim \rho^3/\epsilon^2$ vertices, check if the induced subgraph $G[S]$ has an induced subgraph of size ρs with fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} s^2$ edges.

Key Challenge: If G is ϵ-far from having a ρn independent set, show that S has a ρs sparse set with only small probability.

Induced subgraph has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} s^2$ edges

A Container Lemma For Sparse Sets

Desired Lemma?: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ with $J \subseteq C$.

A Container Lemma For Sparse Sets

Desired Lemma?: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ with $J \subseteq C$.

This is not possible
(more details in a few slides)

A Container Lemma For Sparse Sets

Desired Lemma?: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ with $J \subseteq C$.

No prior results of this form. Only similar results on "sparse subgraphs" apply to subgraphs with smaller density or bounded max degree.

[Nenadov '24] [Saxton, Thomason '15]

This is not possible (more details in a few slides)

A Container Lemma For Sparse Sets

Lemma*: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ and α such that $|C| \leq (1-\alpha)\rho n$ and

$$|C \cap J| \geq \left(1 - \frac{\alpha}{2}\right)|J|.$$

* Omitting some details

A Container Lemma For Sparse Sets

Lemma*: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ and α such that $|C| \leq (1-\alpha)\rho n$ and
$$|C \cap J| \geq \left(1 - \frac{\alpha}{2}\right)|J|.$$

Informally: each sparse set is "mostly" contained in a container

* Omitting some details

A Container Lemma For Sparse Sets

Lemma*: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

Using this new container lemma instead of standard container lemma used by Blais, Seth ('23) we can the prove theorem.

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ and α such that $|C| \leq (1-\alpha)\rho n$ and

$$|C \cap J| \geq \left(1 - \frac{\alpha}{2}\right)|J|.$$

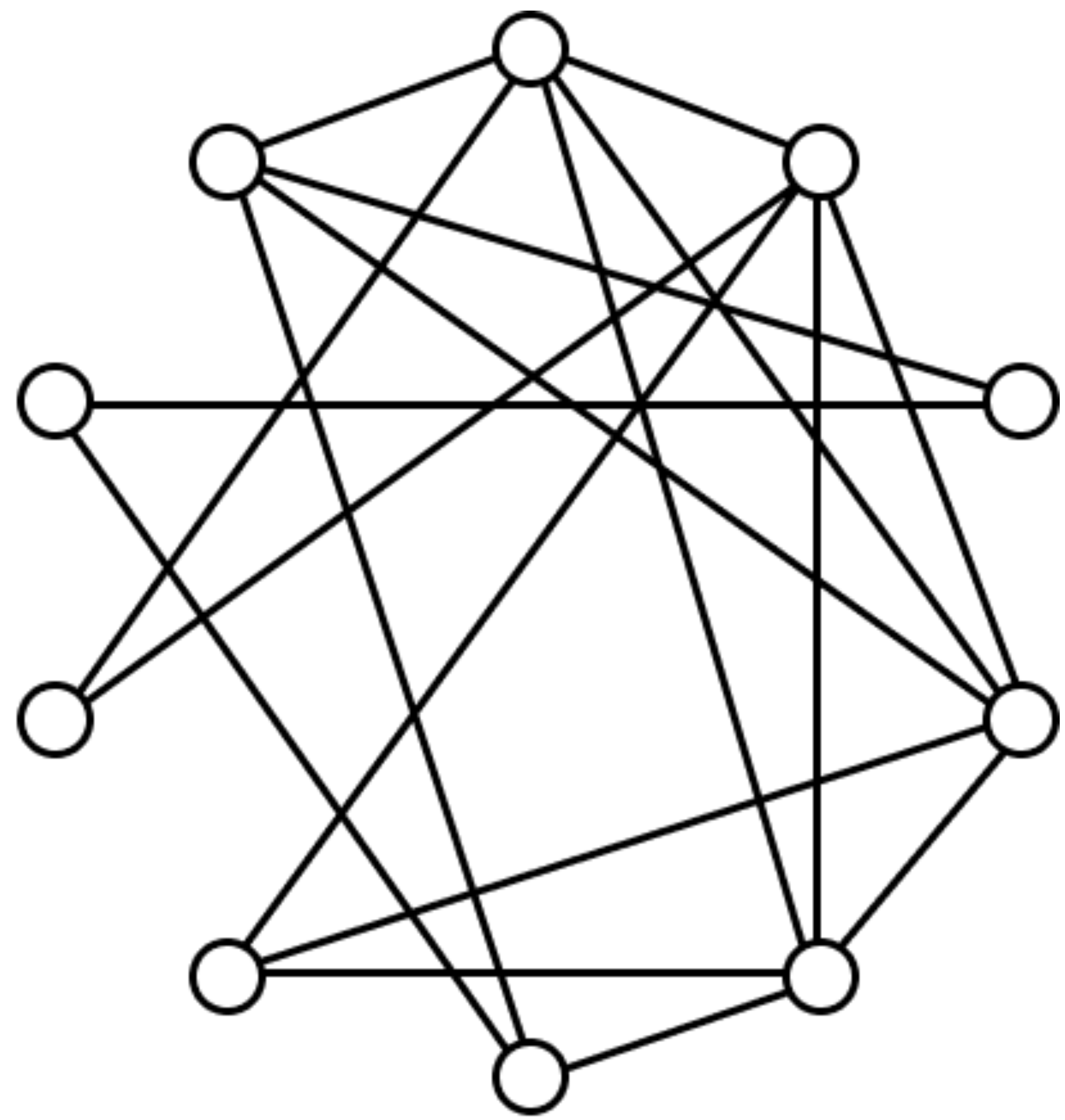
Informally: each sparse set is "mostly" contained in a container

* Omitting some details

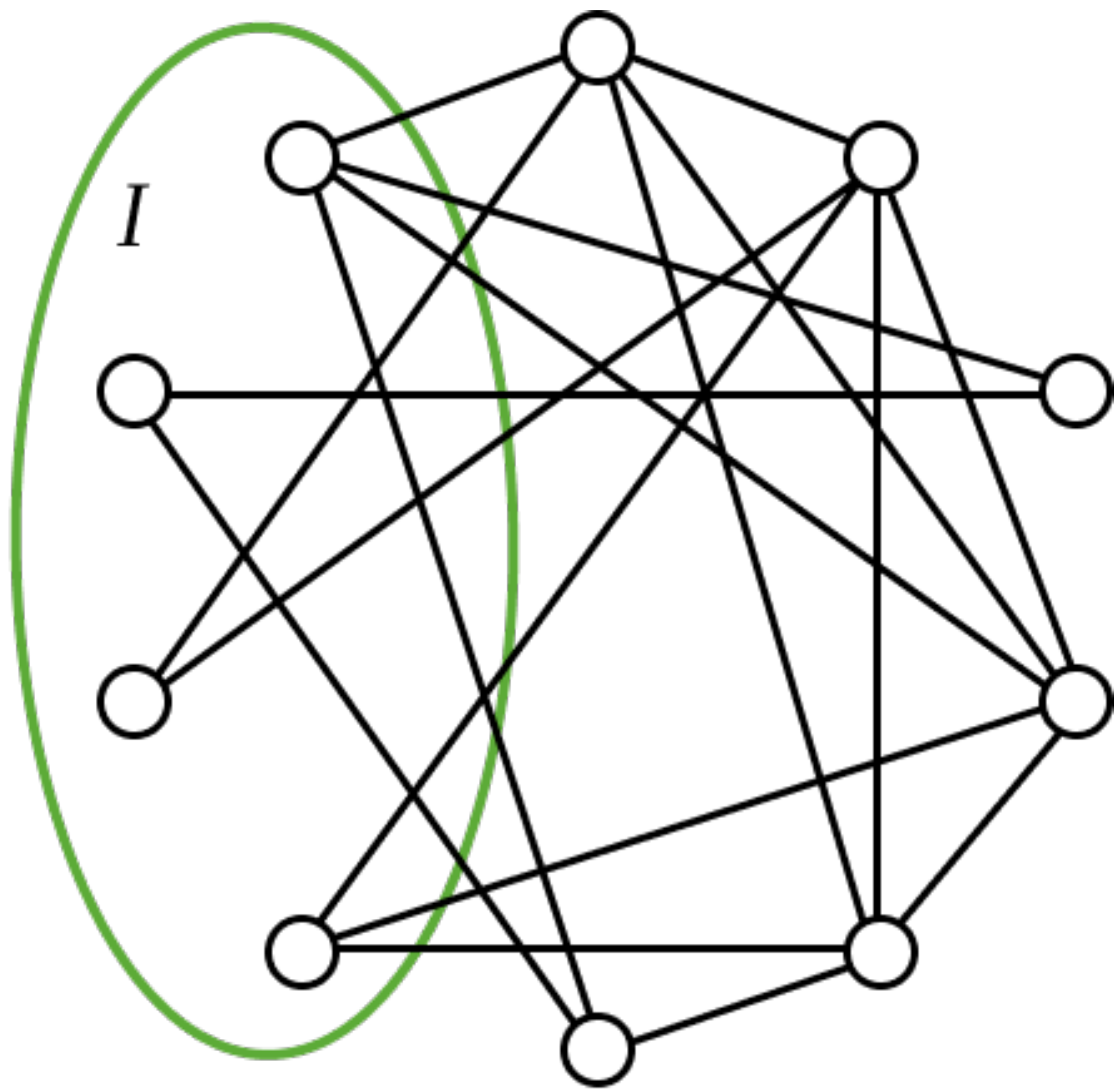
Warmup: How to prove Container Lemma for Independent Sets - An Encoding Argument

Encoding Independent Sets

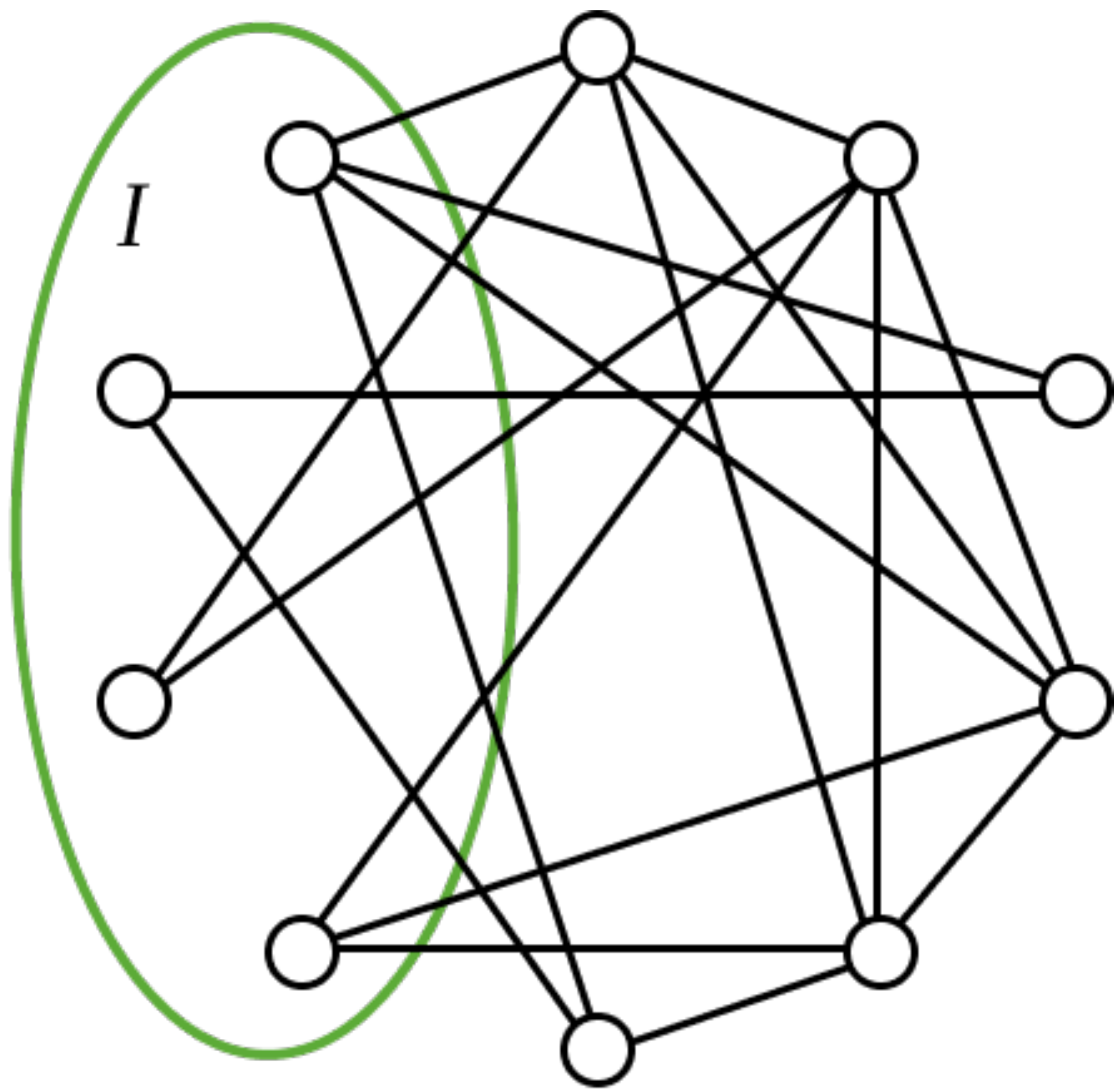
Encoding Independent Sets



Encoding Independent Sets

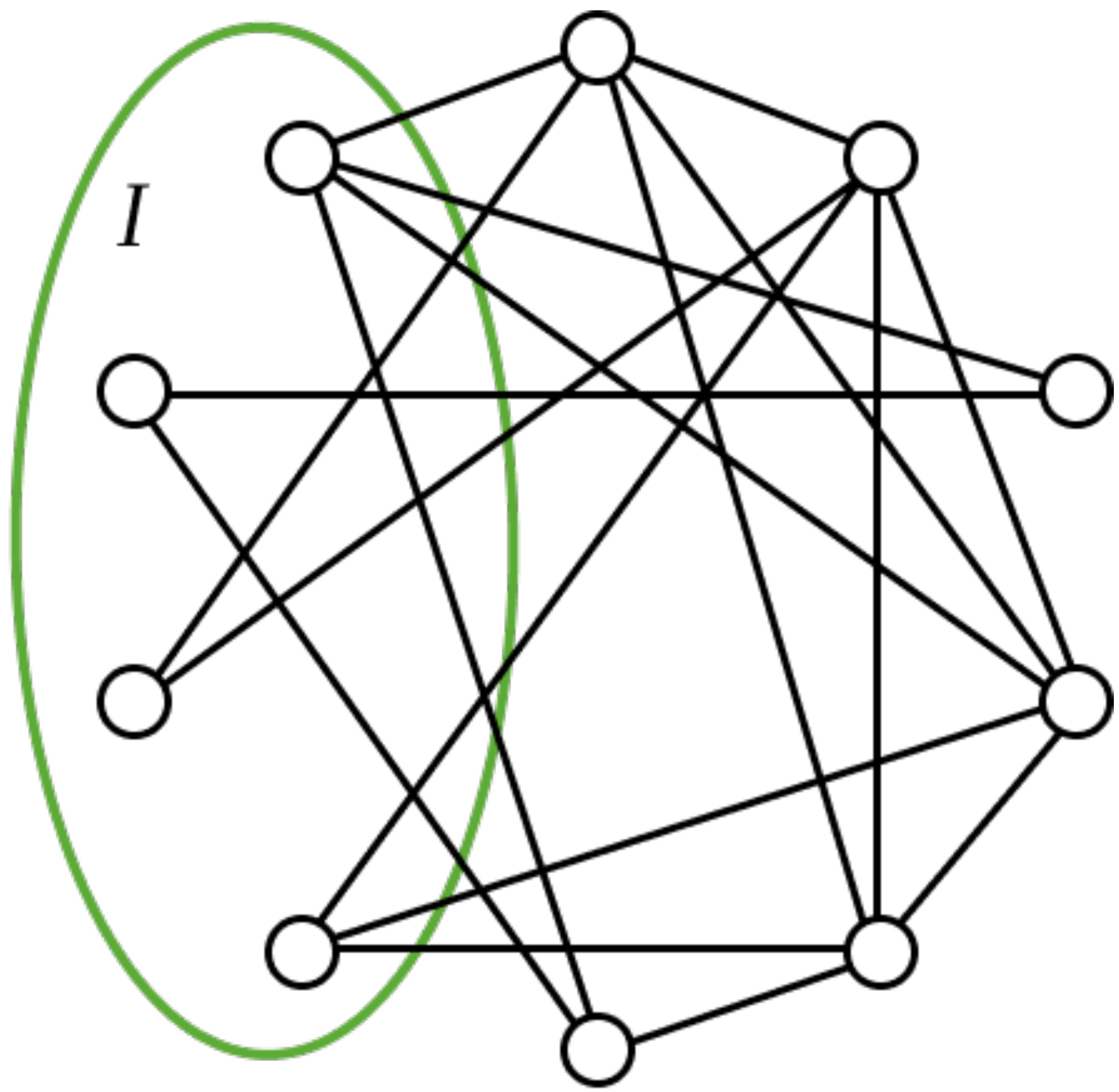


Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

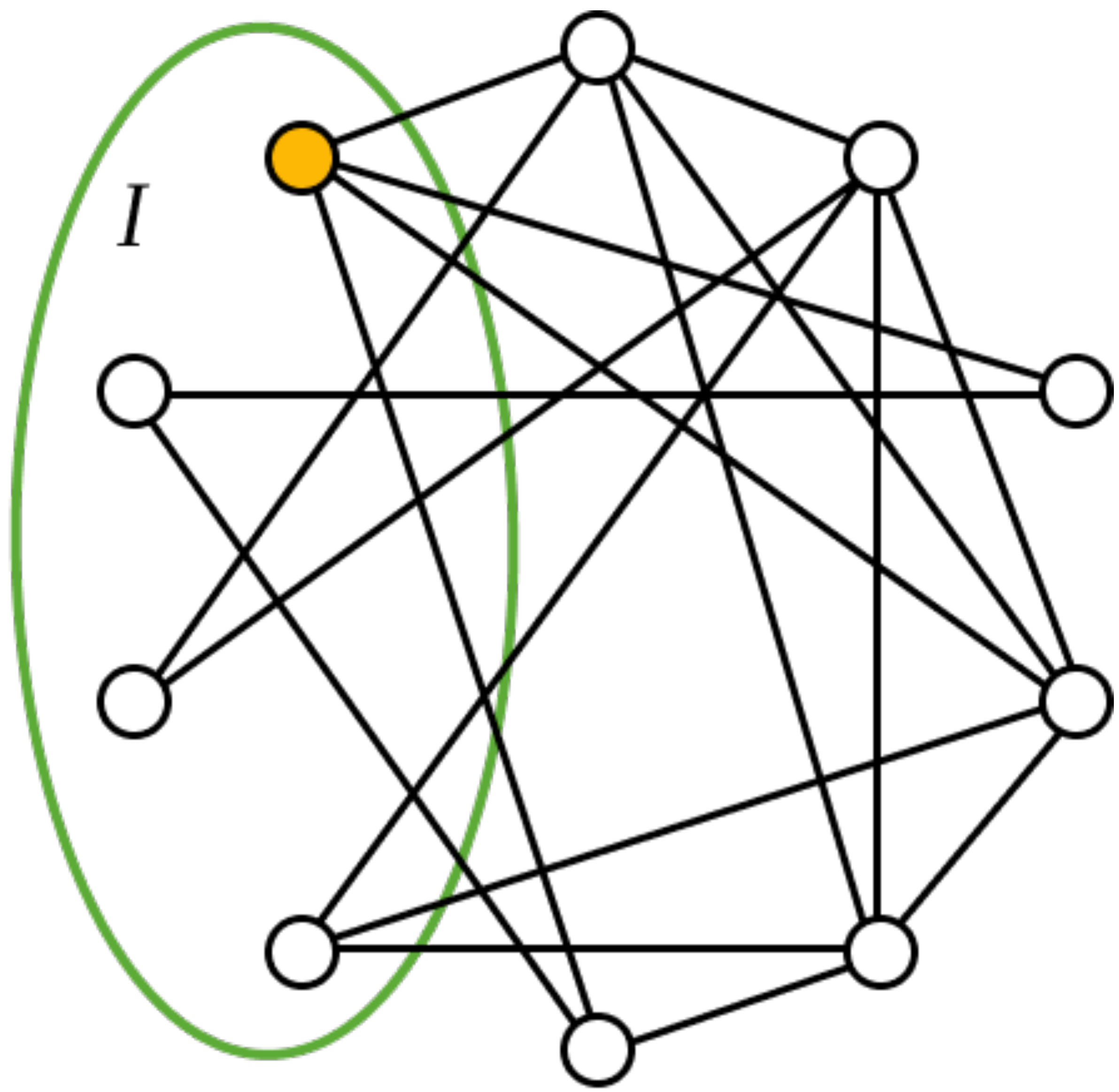
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

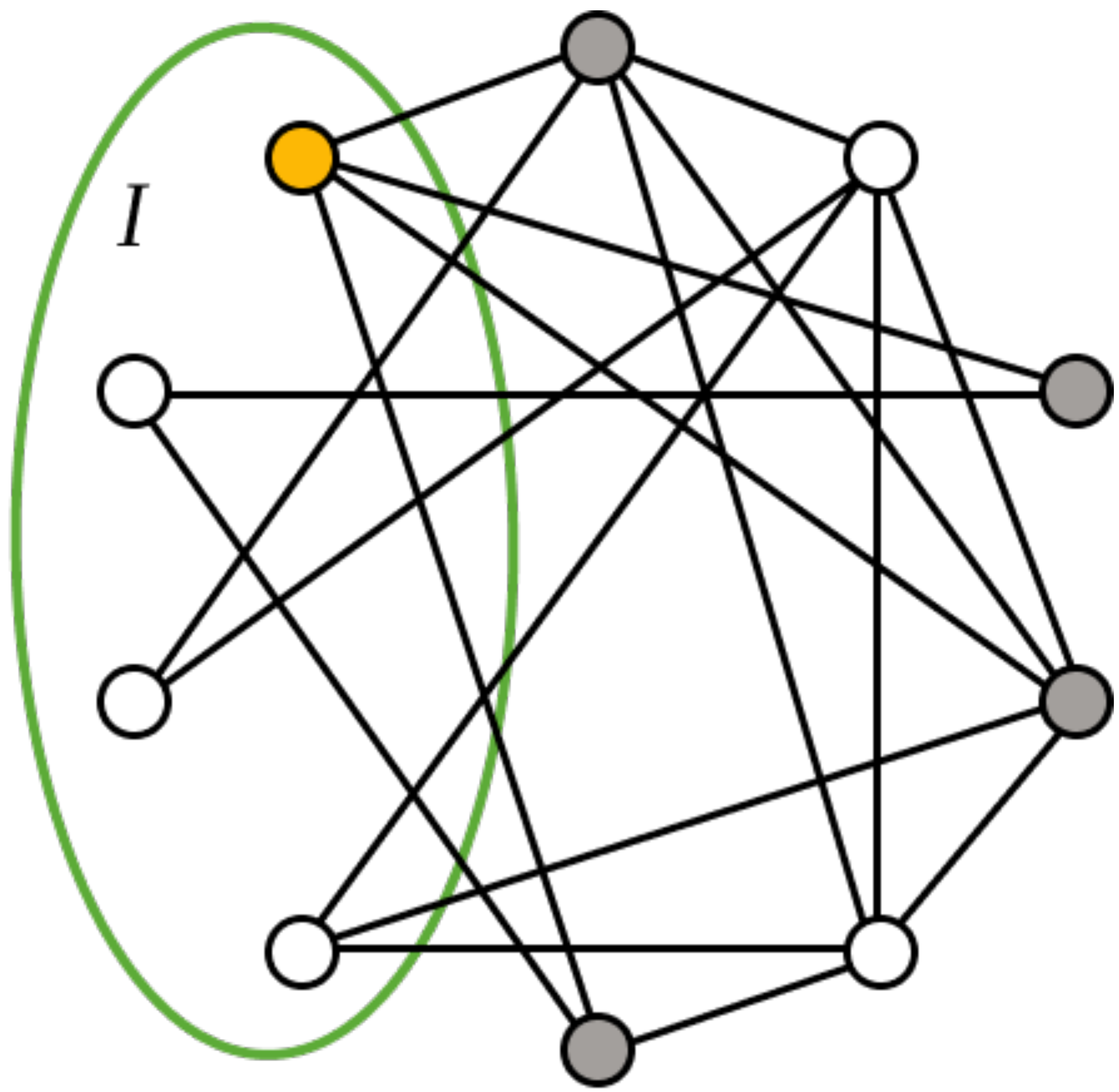
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

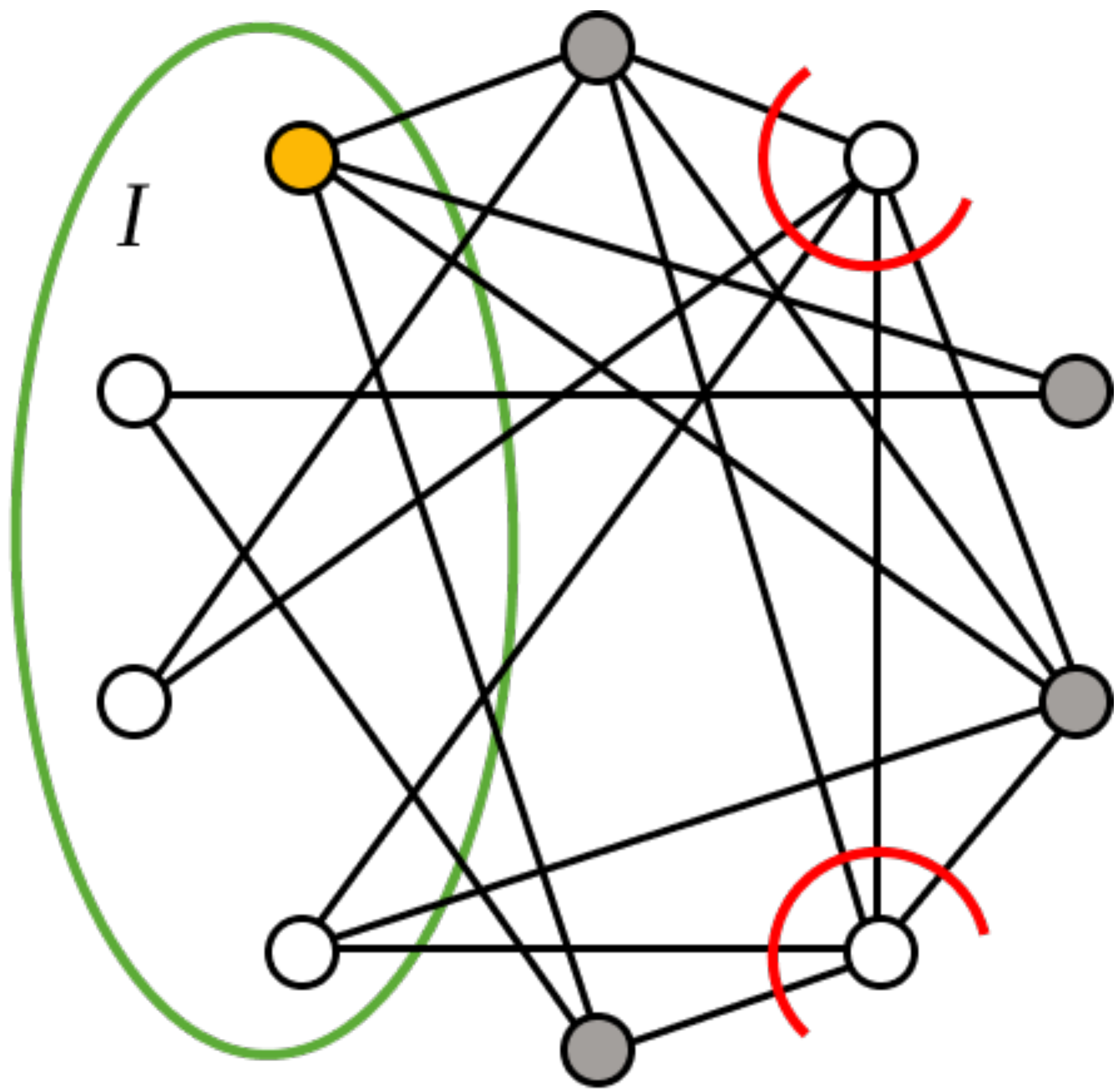
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

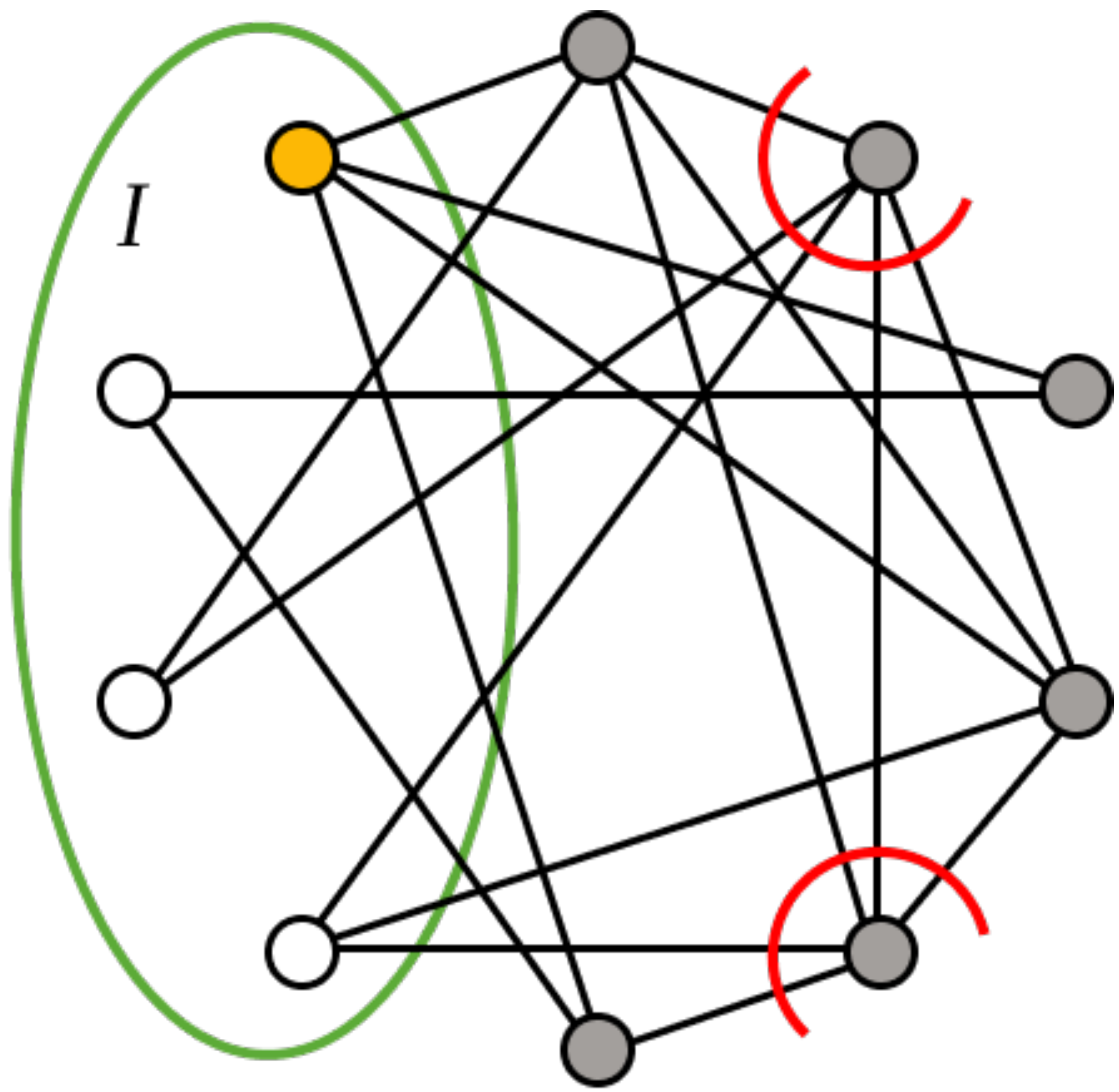
Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Encoding Independent Sets



How can I give you the most information about an independent set I by just telling you about one of the vertices in I?

Answer: Send the vertex $v \in I$ with highest degree.

Container Generator Algorithm

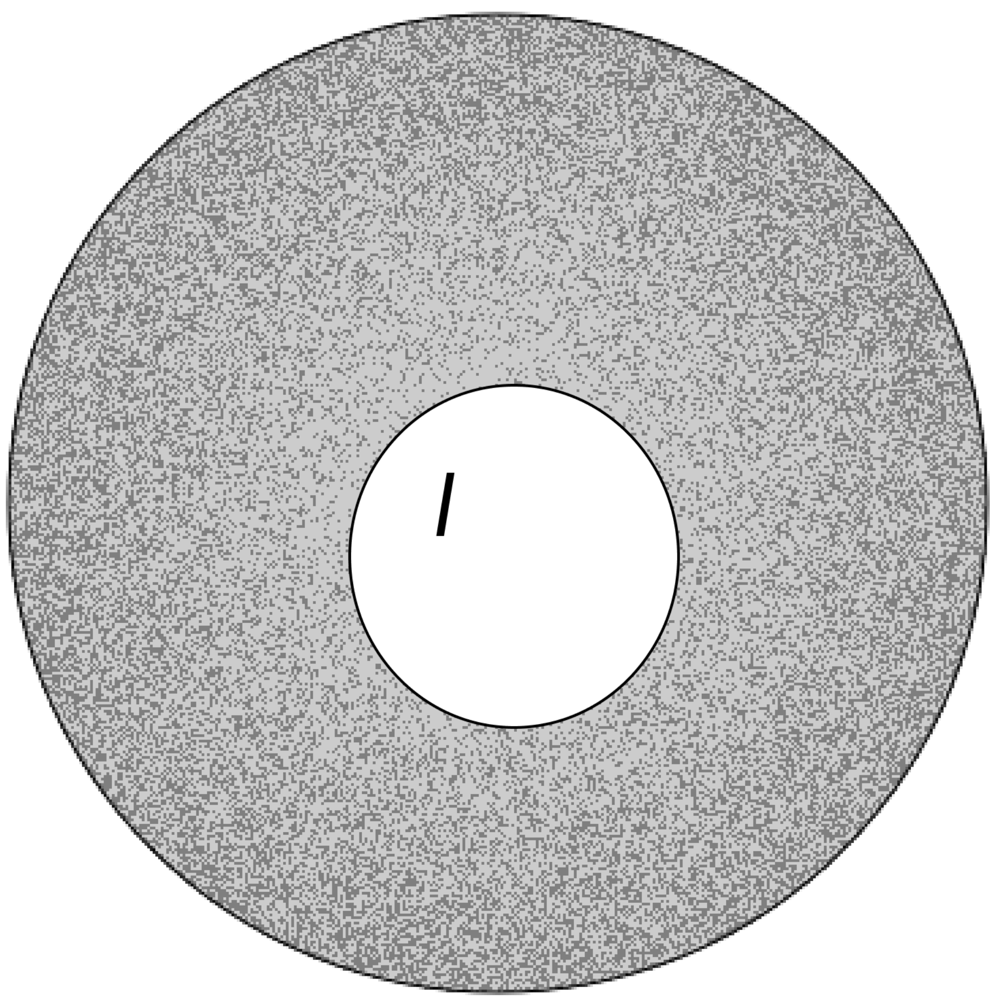
Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

Container Generator Algorithm

Input: Graph G and an independent set I

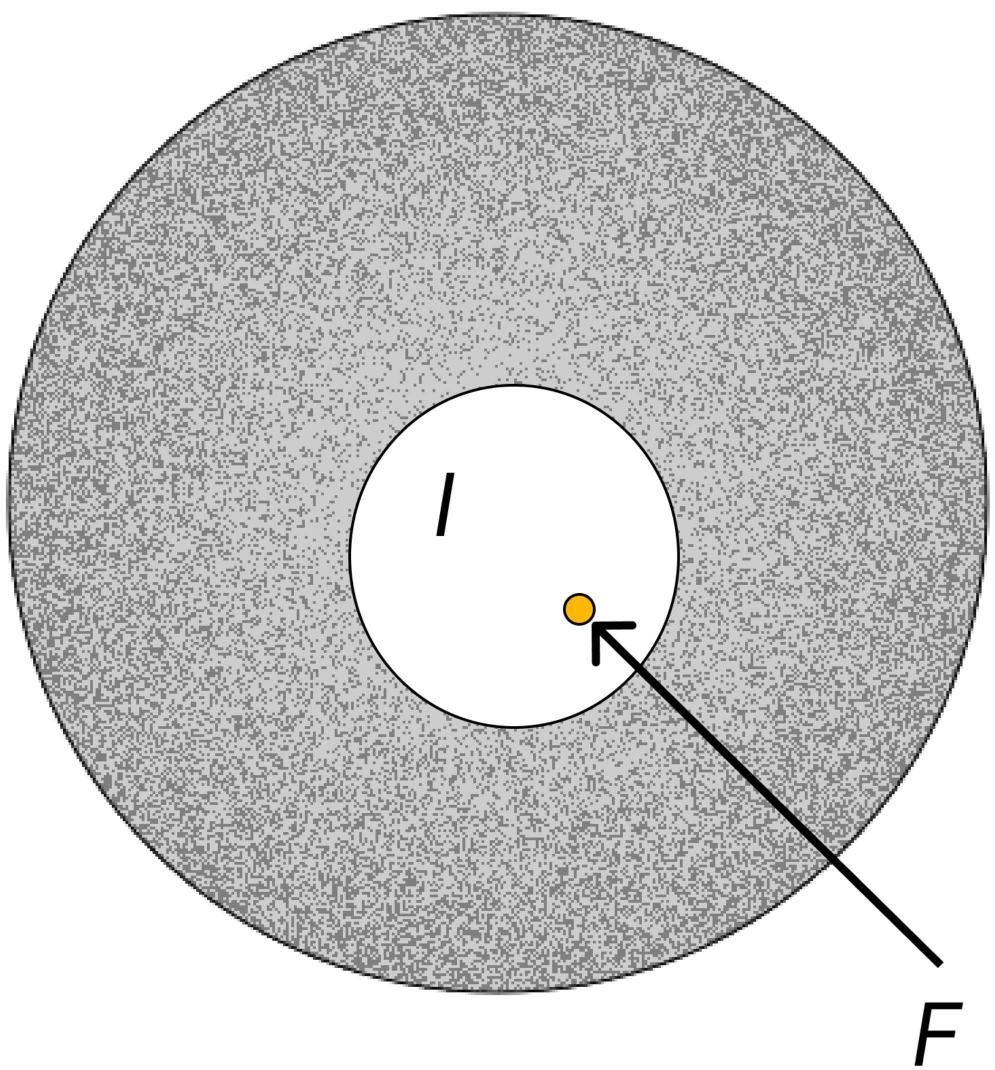
- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.



Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

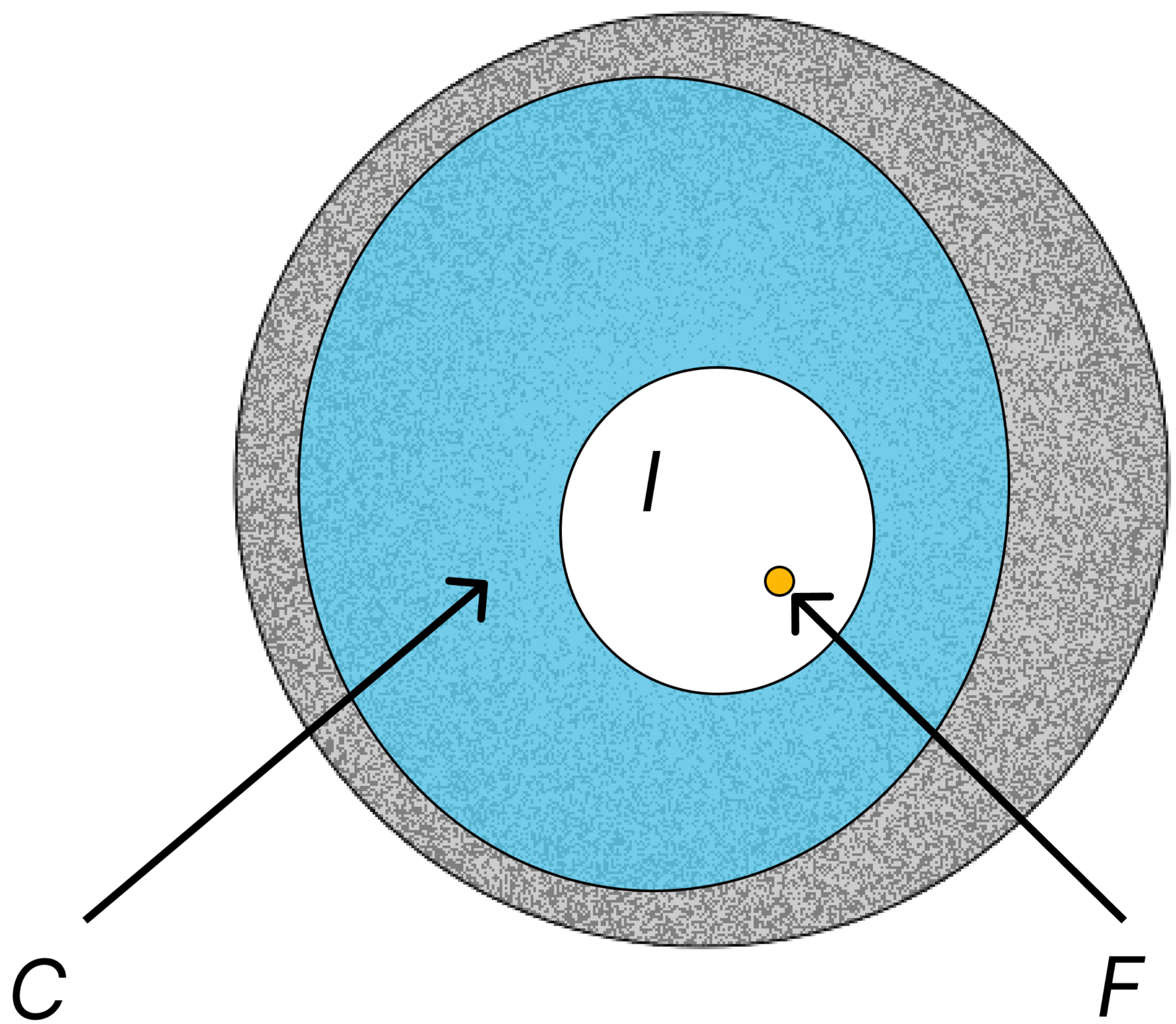


1st Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

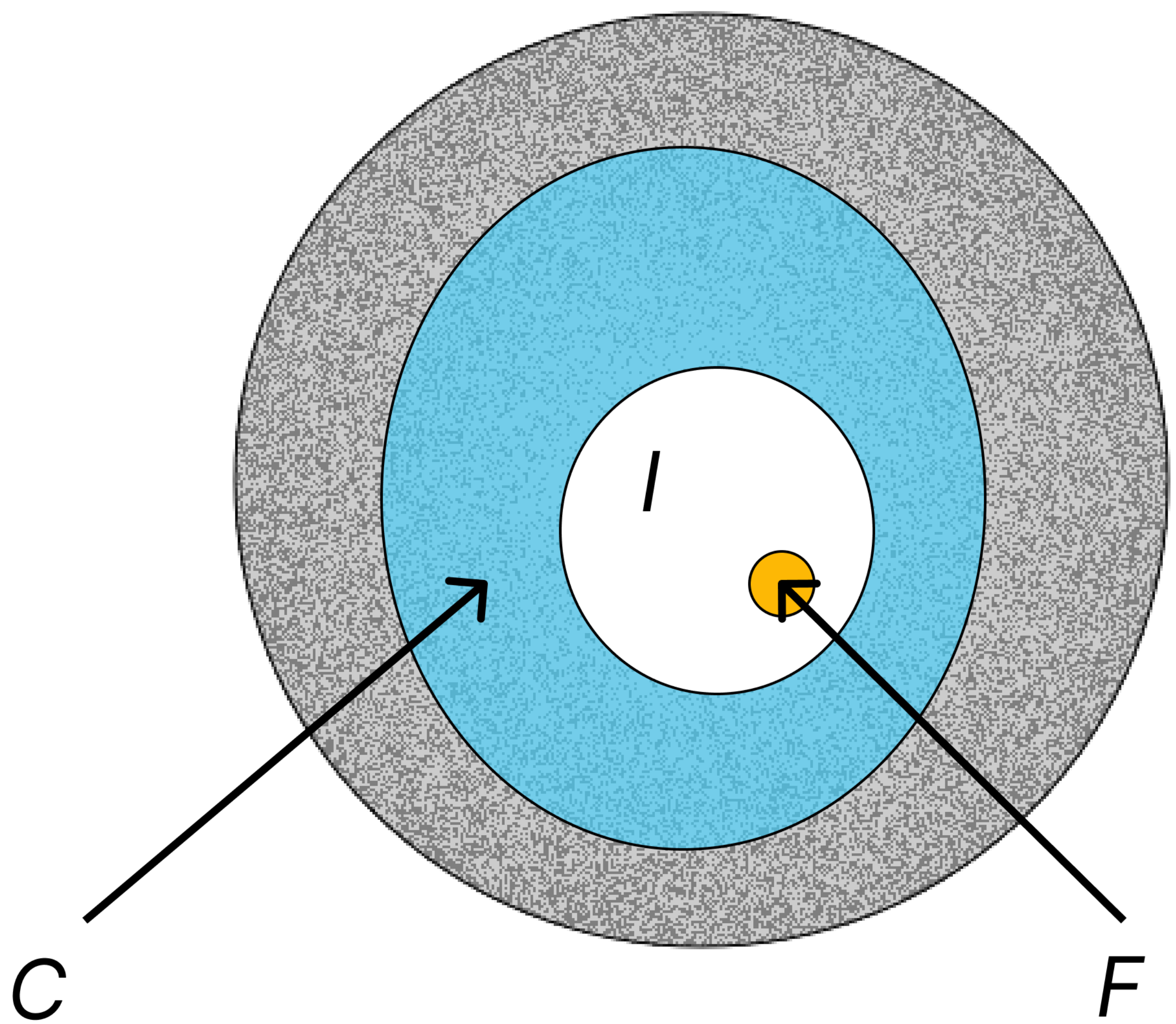


1st Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

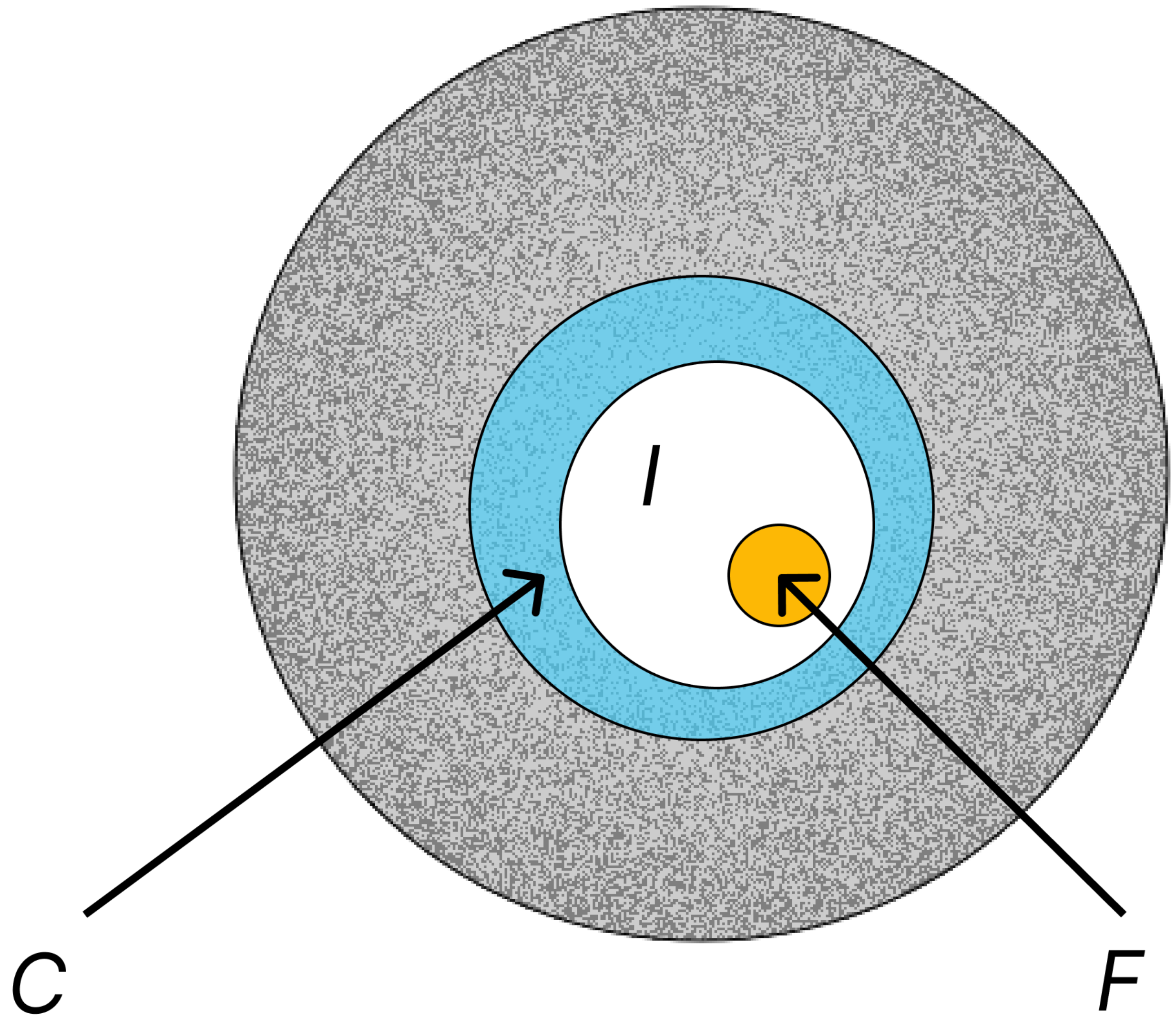


2nd Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

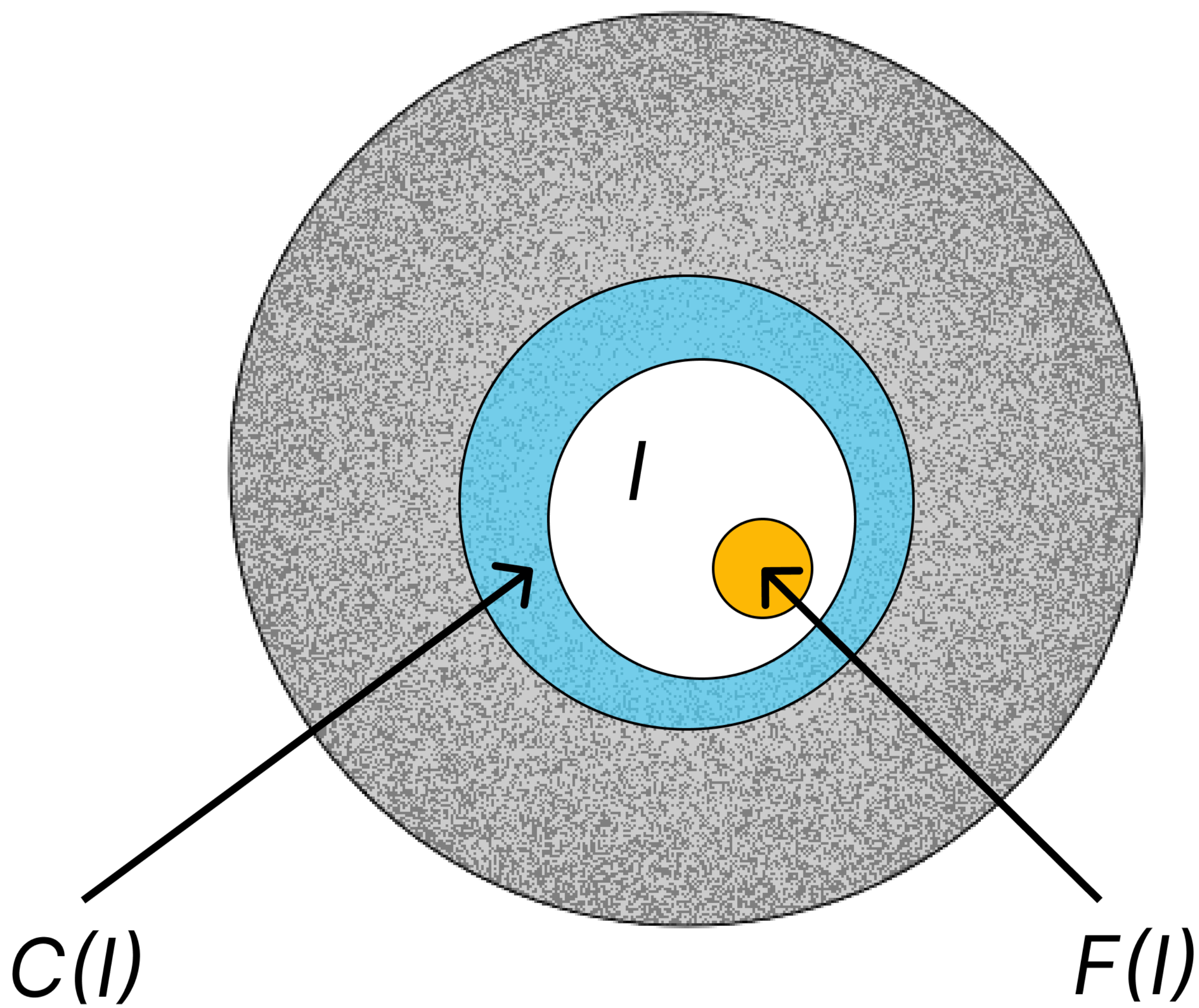


Final Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.

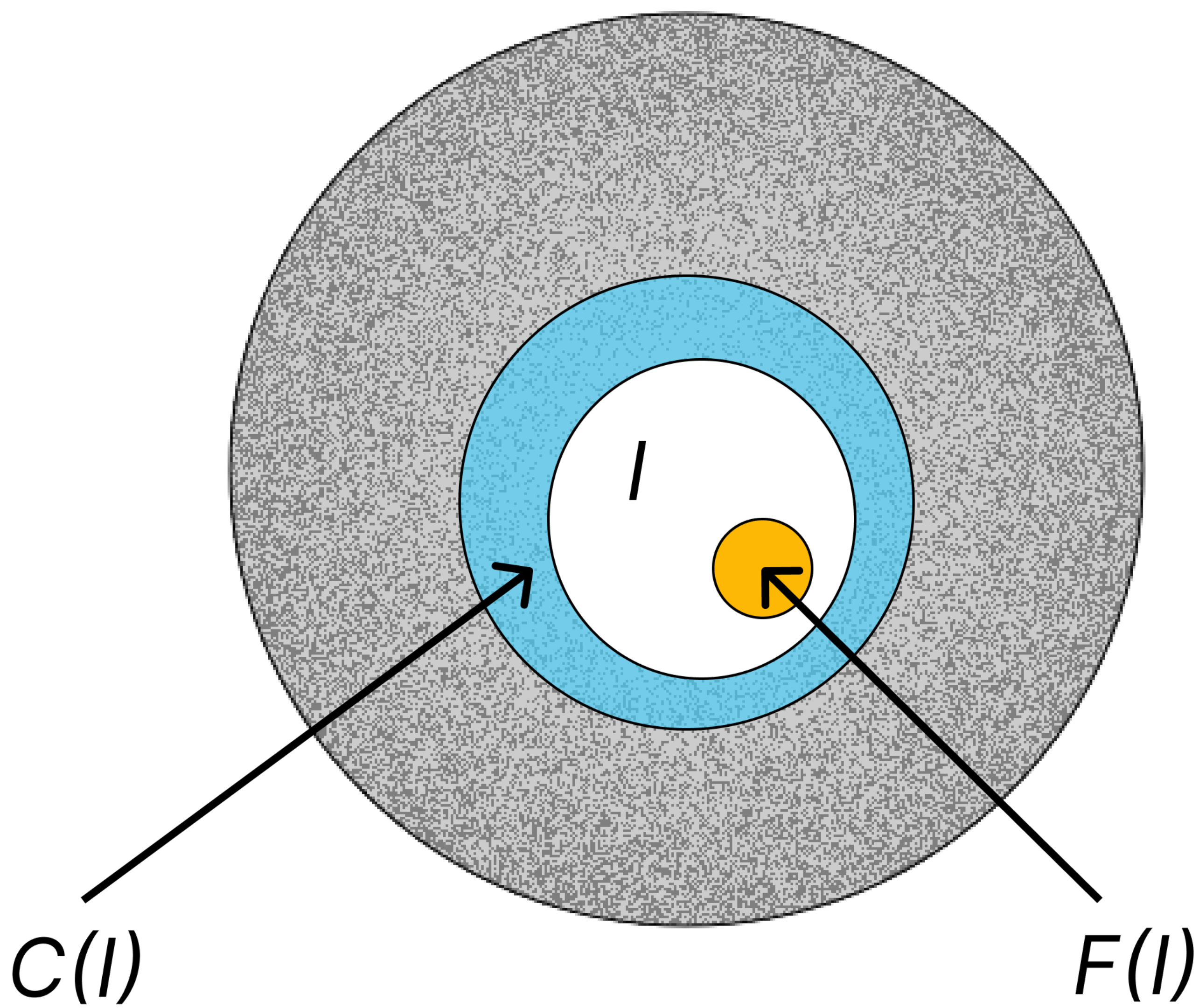


Final Iteration

Container Generator Algorithm

Input: Graph G and an independent set I

- Initialize fingerprint $F = \emptyset$ and container $C = V$
- Repeatedly select a vertex $v \in I$ with highest degree in $G[C]$ and add it to F. Remove $N(v)$ and all vertices with higher degree than v from C.



Final Iteration

Key Observations:

- $I \subseteq C(I)$
- $C(I) = C(F(I))$

Container Lemma for Independent Sets - Restated

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Container Lemma for Independent Sets - Restated

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: how do we construct the collection of containers?

Container Lemma for Independent Sets - Restated

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: how do we construct the collection of containers?

Answer:

Container Lemma for Independent Sets - Restated

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: how do we construct the collection of containers?

Answer:

Let $\mathscr{C} = \{C(I) : I \text{ is an independent set in } G\}$

Container Lemma for Independent Sets - Restated

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: how do we construct the collection of containers?

Answer:

$$\text{Let } \mathscr{C} = \{C(I) : I \text{ is an independent set in } G\}$$
$$= \{C(F(I)) : I \text{ is an independent set in } G\}$$

Container Lemma for Independent Sets - Restated

Lemma: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. for every independent set I, there exists $C \in \mathscr{C}$ with $I \subseteq C$.

[Kleitman, Winston '82]

Question: how do we construct the collection of containers?

Answer:

$$\begin{aligned} \text{Let } \mathscr{C} &= \{C(I) : I \text{ is an independent set in } G\} \\ &= \{C(F(I)) : I \text{ is an independent set in } G\} \\ &= \{C(F) : F = F(I) \text{ for some independent set } I \text{ in } G\} \end{aligned}$$

How to prove new Container Lemma for Sparse Sets - An Encoding Argument

A Container Lemma For Sparse Sets (restated)

Lemma*: For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq 2^V$ of containers that satisfies:

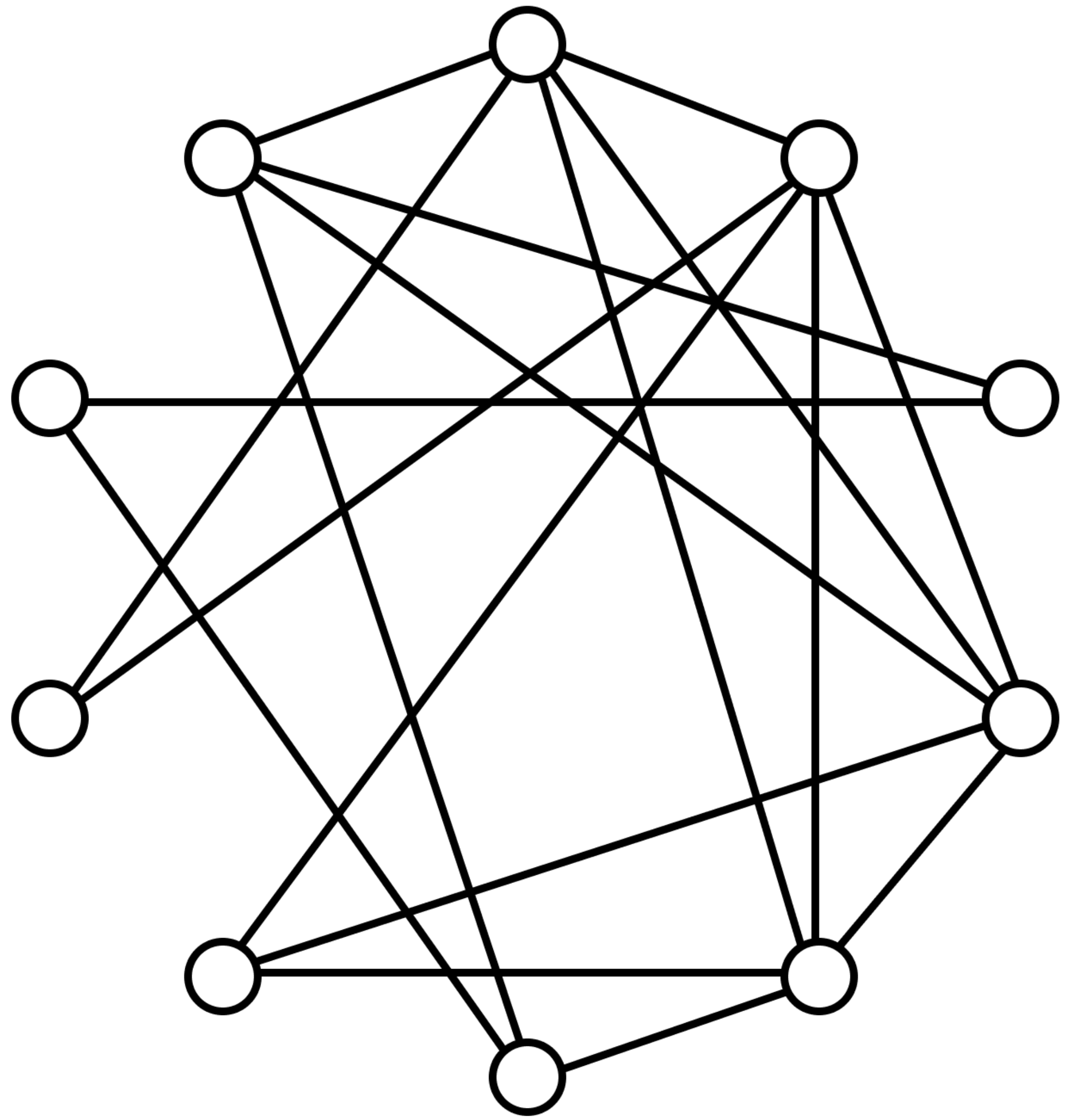
1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ and α such that $|C| \leq (1-\alpha)\rho n$ and
$$|C \cap J| \geq \left(1 - \frac{\alpha}{2}\right)|J|.$$

Informally: each sparse set is "mostly" contained in a container

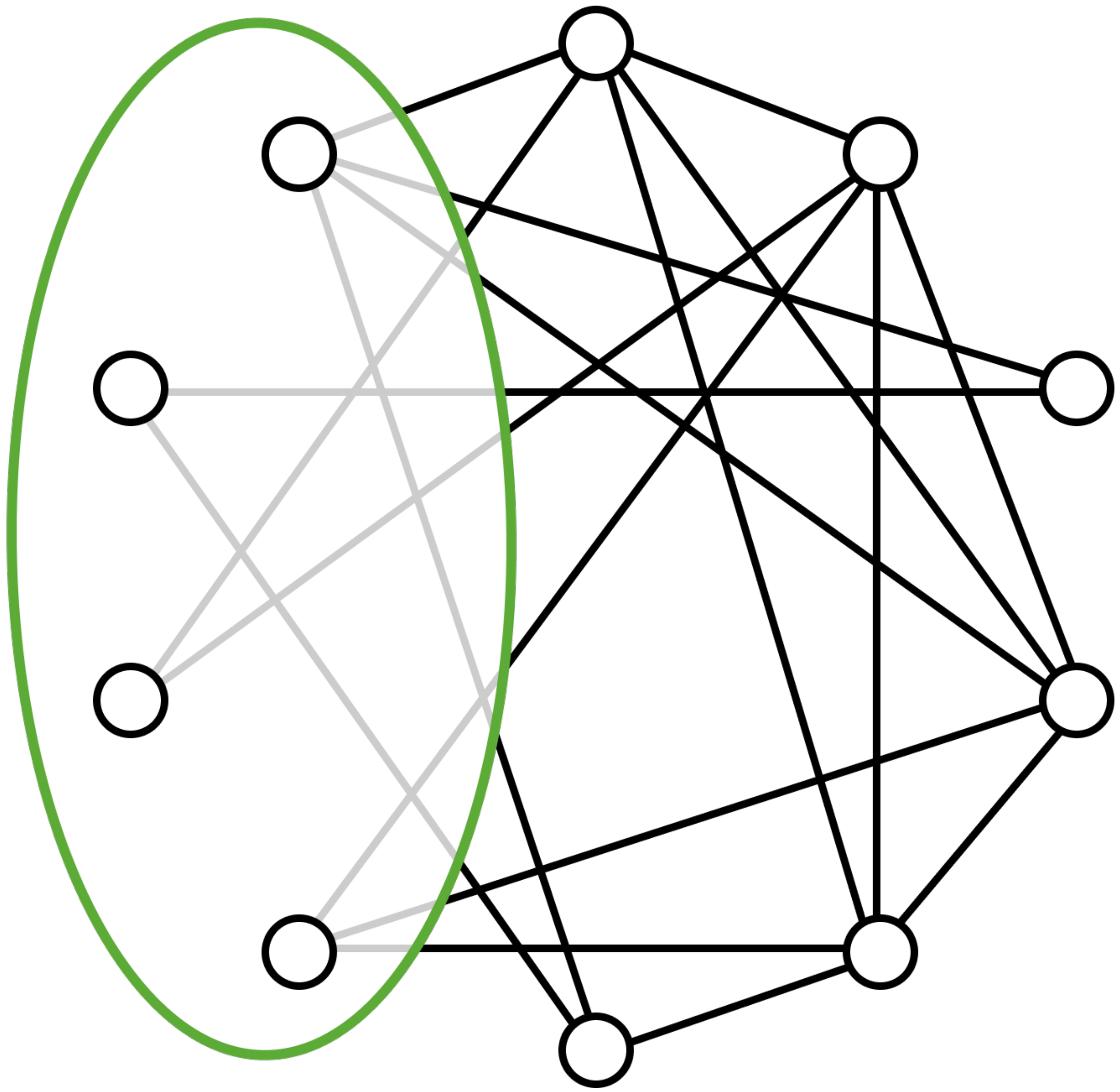
* Omitting some details

Encoding Sparse Sets

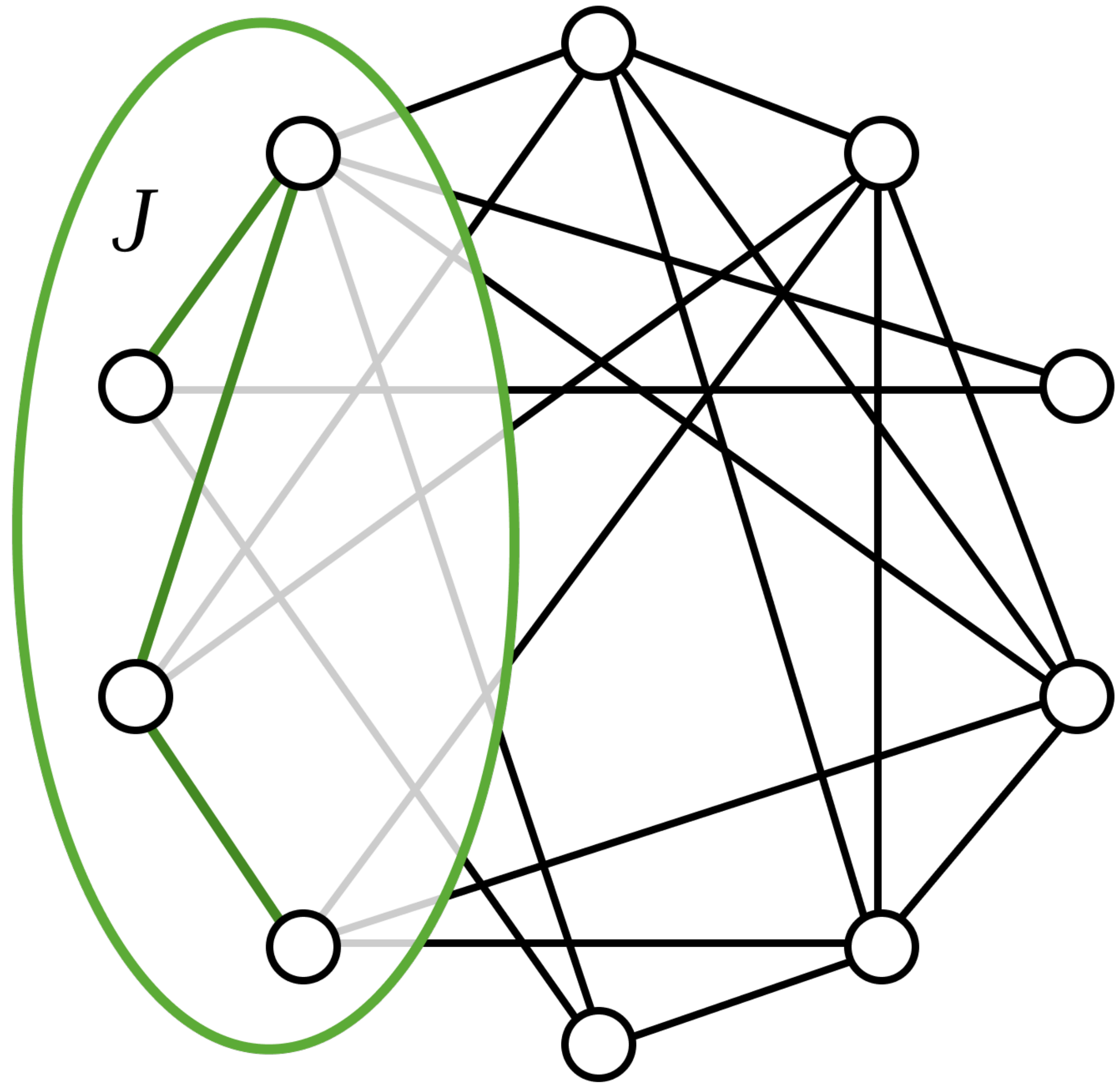
Encoding Sparse Sets



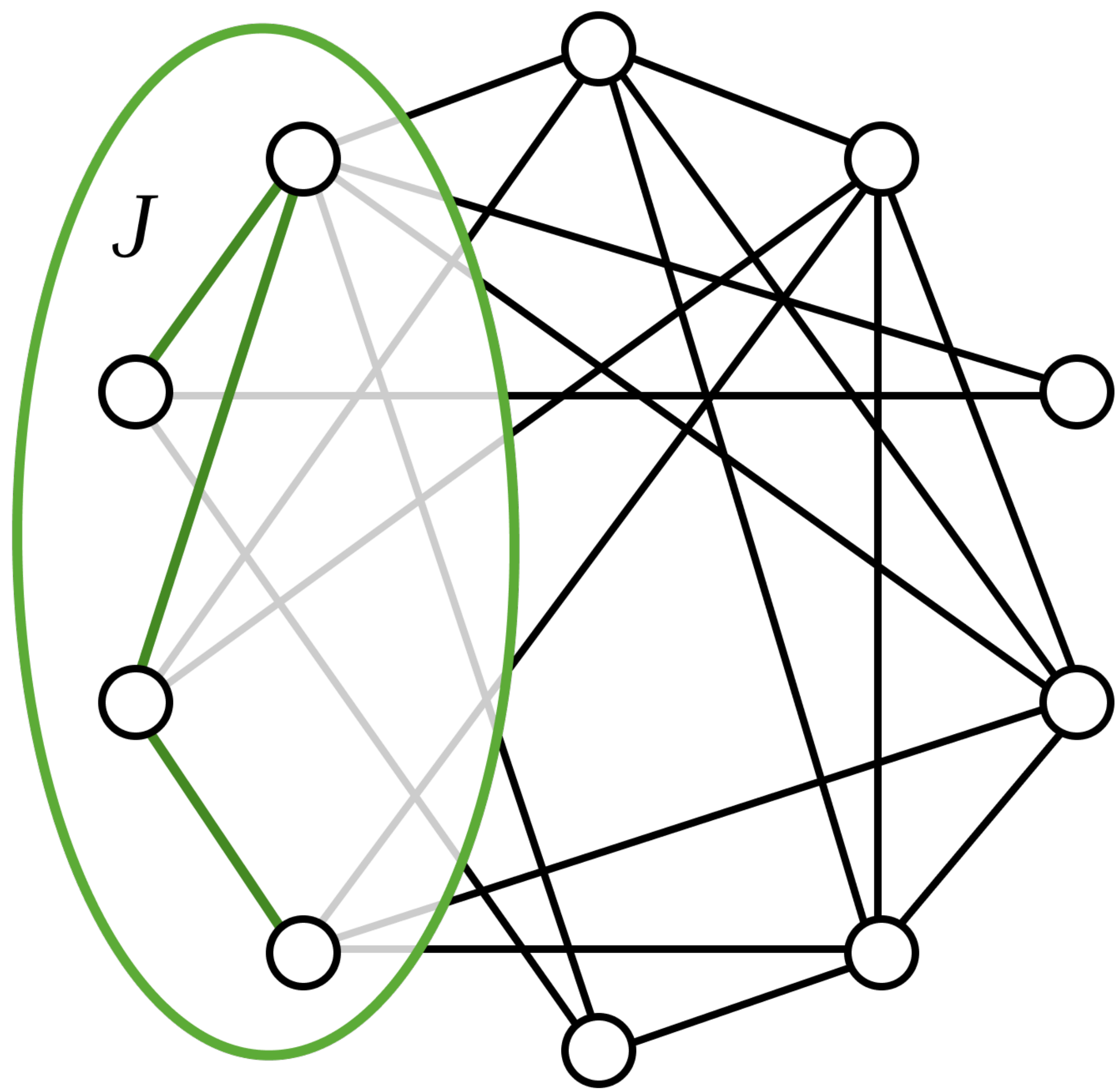
Encoding Sparse Sets



Encoding Sparse Sets

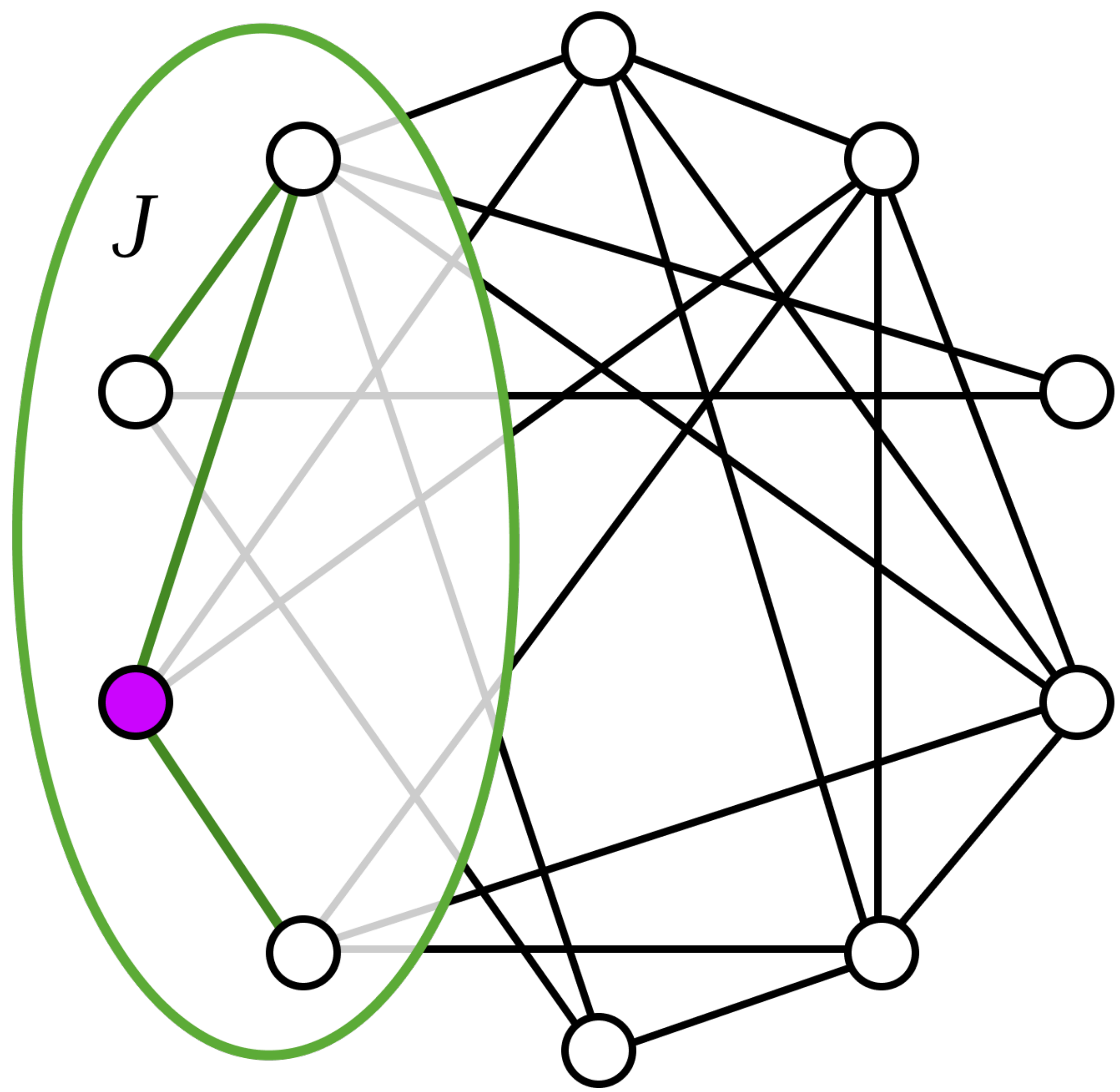


Encoding Sparse Sets

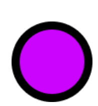


How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

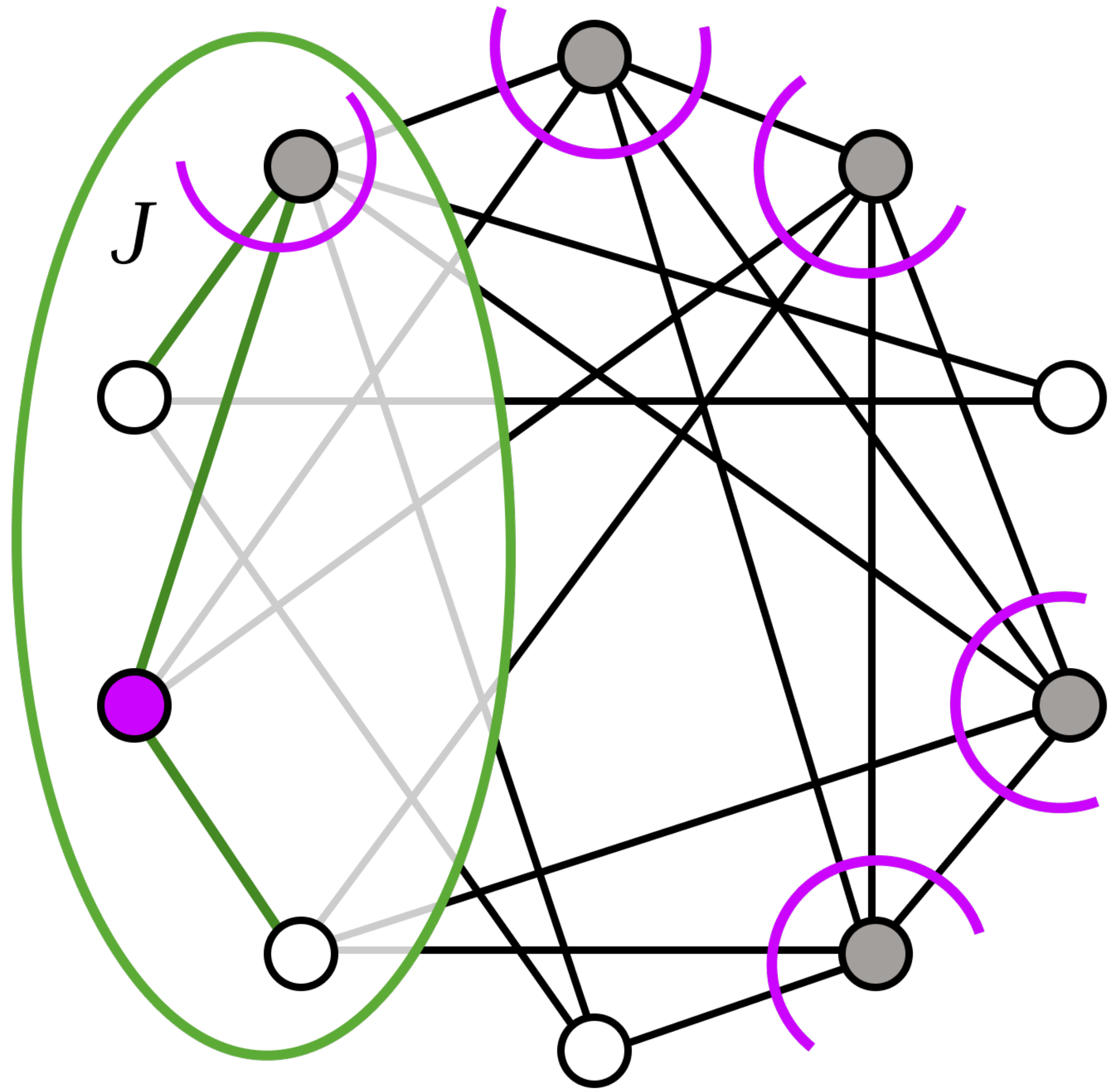
Encoding Sparse Sets



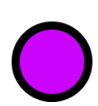
How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send  and remove higher degree vertices

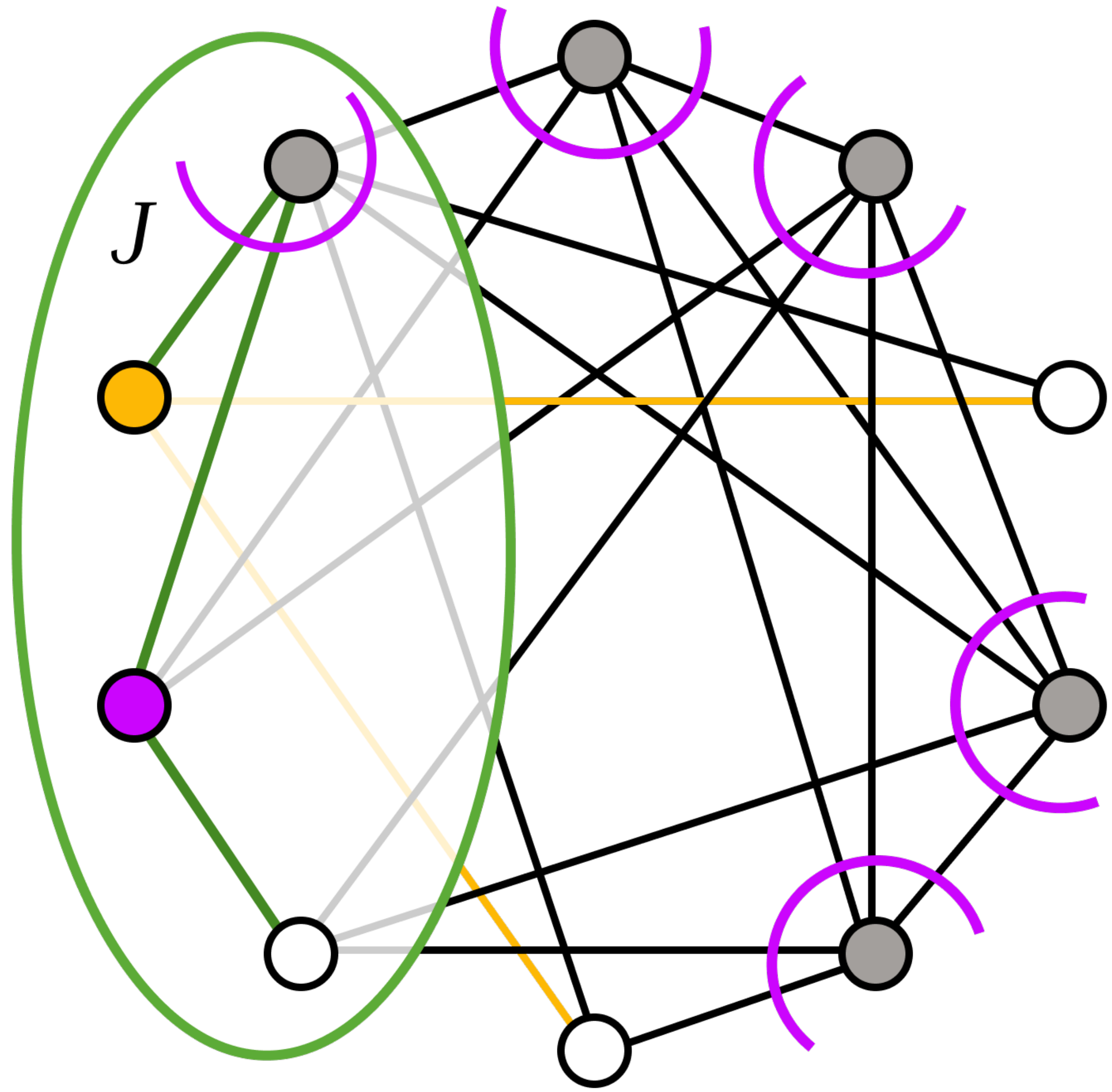
Encoding Sparse Sets



How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send  and remove higher degree vertices

Encoding Sparse Sets

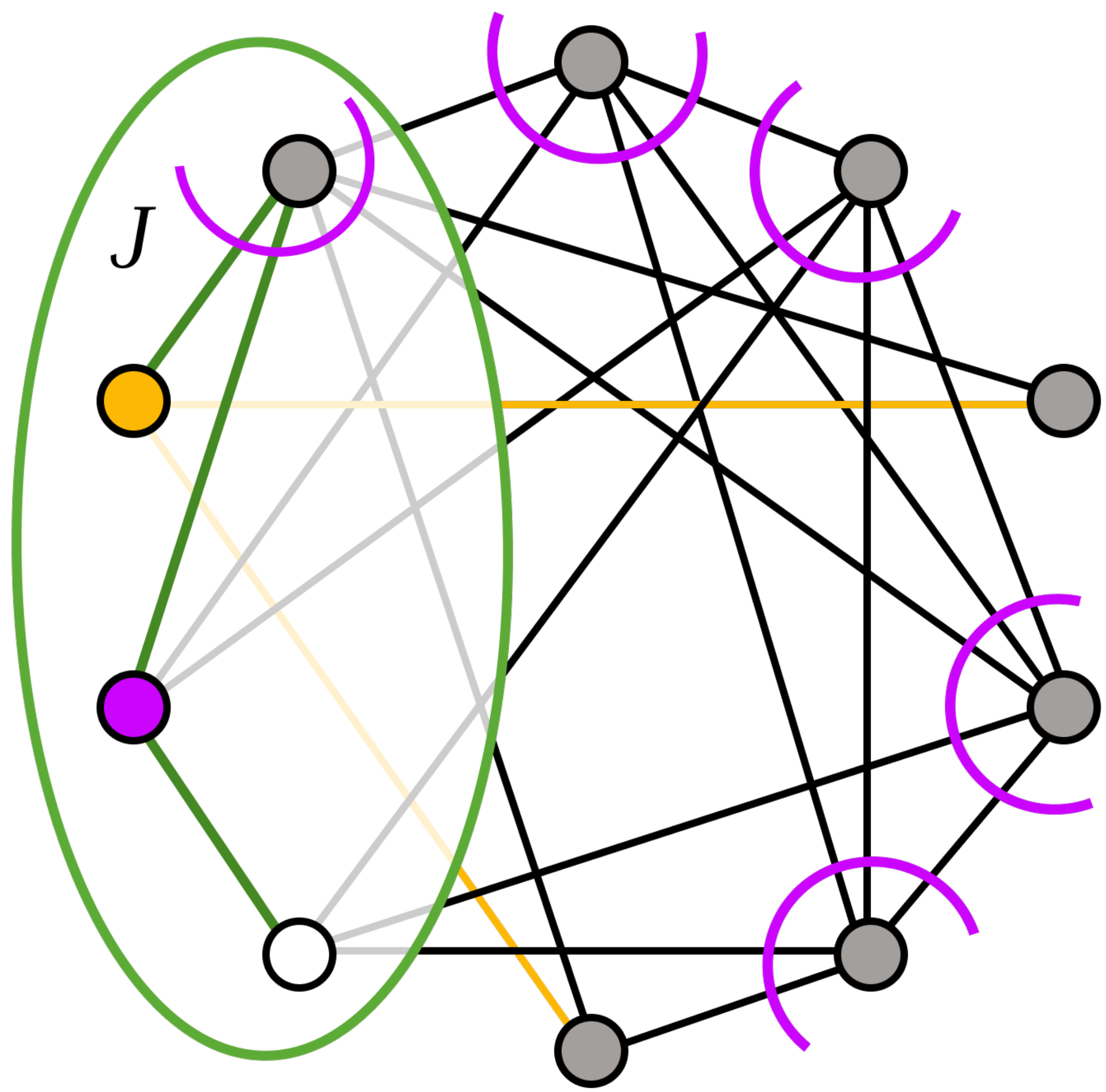


How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send  and remove higher degree vertices

Send  and remove neighbours

Encoding Sparse Sets



How can I give you the most information about a sparse set J by just telling you about one (or two) of the vertices in J?

Answer: Send and remove higher degree vertices

Send and remove neighbours

Container Generator Algorithm - Sparse Sets

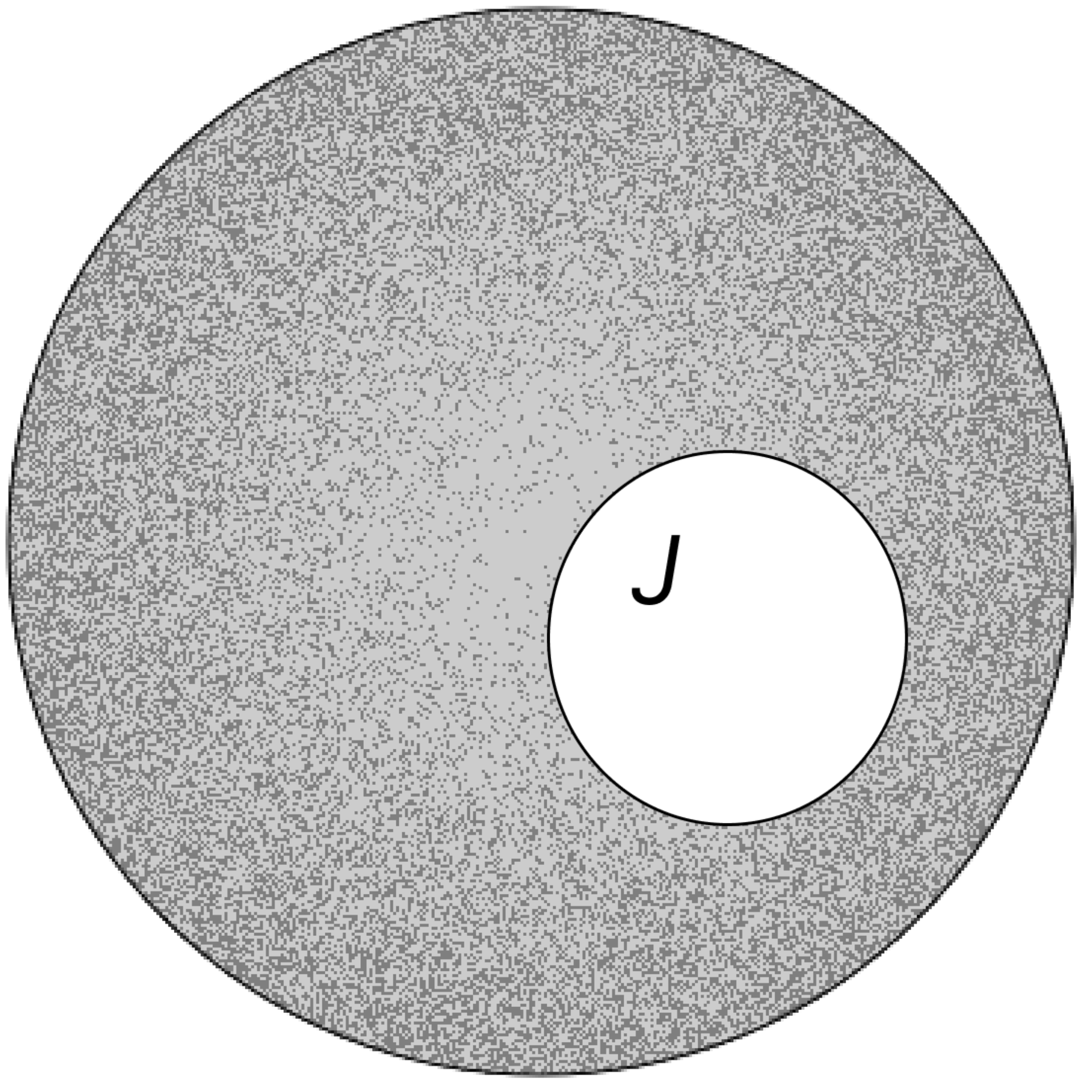
Input: Graph G and a sparse set J

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Container Generator Algorithm - Sparse Sets



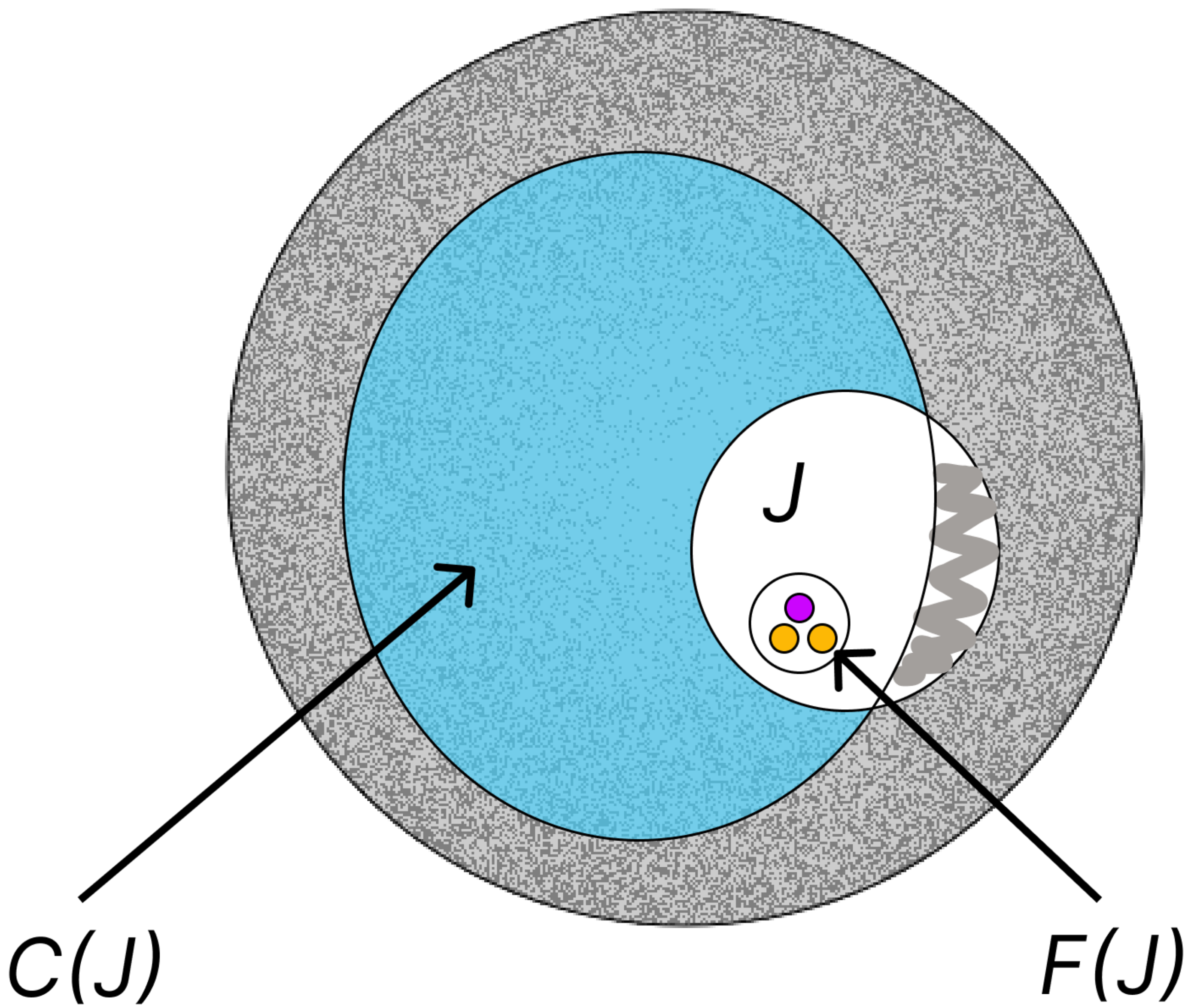
Input: Graph G and a sparse set J

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
    - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

    - apply operation to container, add v and operation to F

Container Generator Algorithm - Sparse Sets



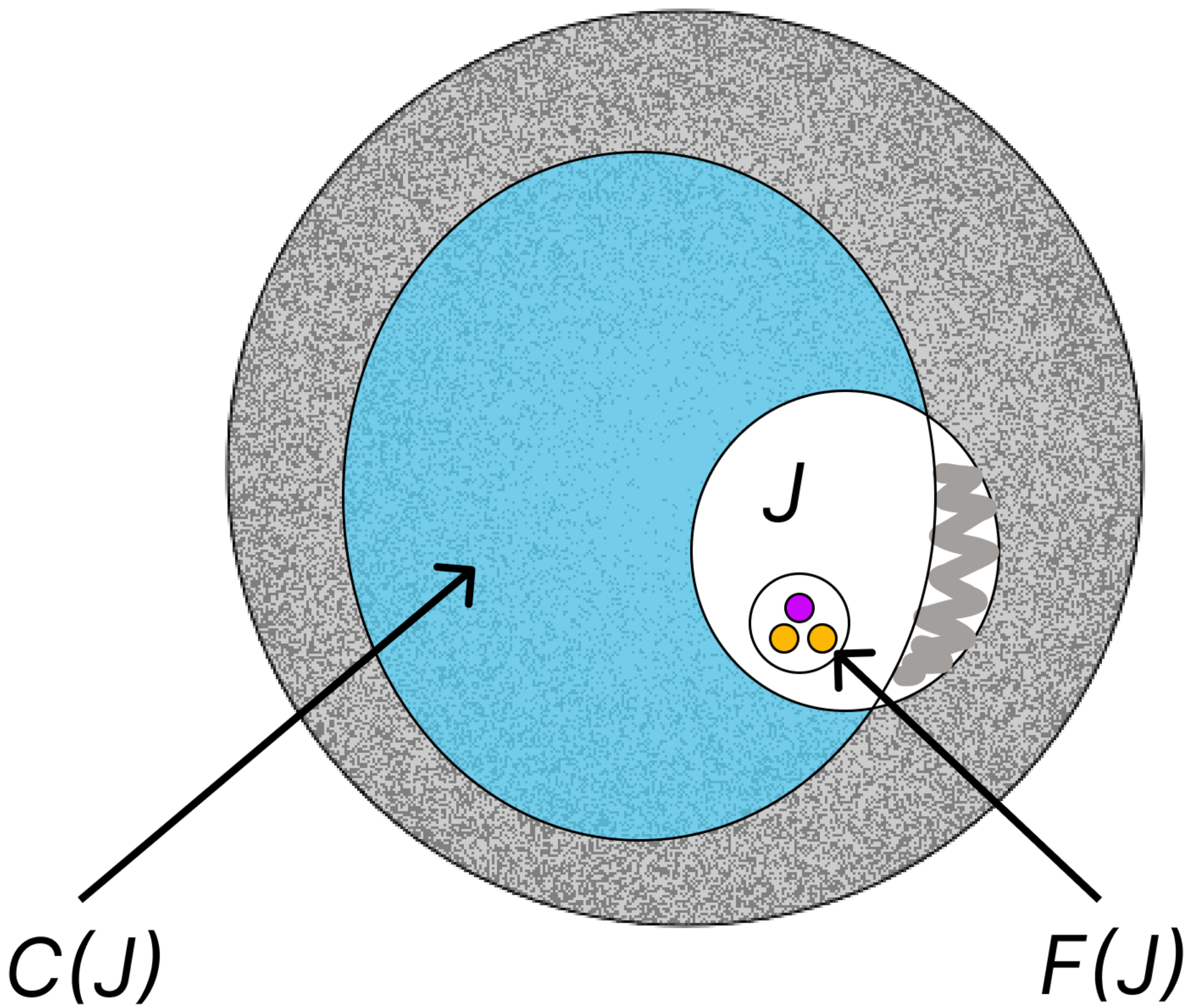
Input: Graph G and a sparse set J

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\#\text{ vertices removed from } C \text{ by operation}}{\#\text{ vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Container Generator Algorithm - Sparse Sets



Input: Graph G and a sparse set J

- Initialize fingerprint $F = \varnothing$ and container $C = V$
- Repeat while $G[C]$ has more than ϵn^2 edges
  - select a vertex $v \in J$ and an operation of "remove neighbours" or "remove higher degree vertices" that maximizes the following:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}}$$

  - apply operation to container, add v and operation to F

Observations:

- $C(J)$ can be reconstructed from $F(J)$
- J is NOT fully contained in $C(J)$

Intuition: Procedure makes Progress

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Intuition: Procedure makes Progress

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Informally: in every step the procedure removes “much” more from the current container C than from J (relatively).

Intuition: Procedure makes Progress

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Informally: in every step the procedure removes “much” more from the current container C than from J (relatively).

For example, if $\frac{n}{2}$ vertices are removed from C then only $\frac{|J|}{2\log(1/\epsilon)}$ vertices of J are removed from C.

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

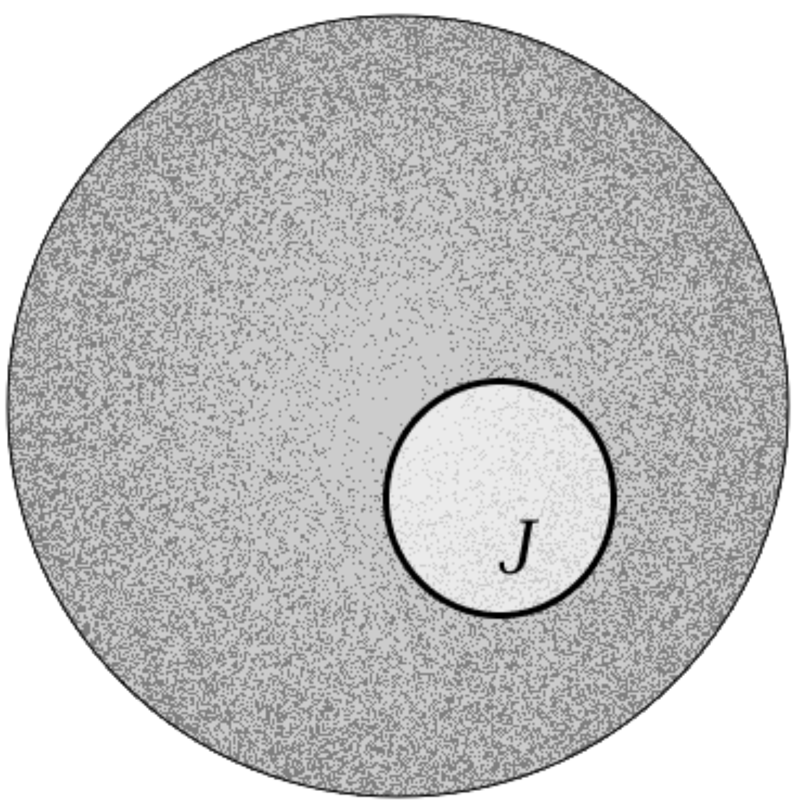
Example 1: Suppose $G[C]$ is the random graph on n vertices with edge density ϵ, and plant the sparse set J (has fewer than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|^2$ edges).

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\#\text{ vertices removed from } C \text{ by operation}}{\#\text{ vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 1: Suppose $G[C]$ is the random graph on n vertices with edge density ϵ, and plant the sparse set J (has fewer than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|^2$ edges).

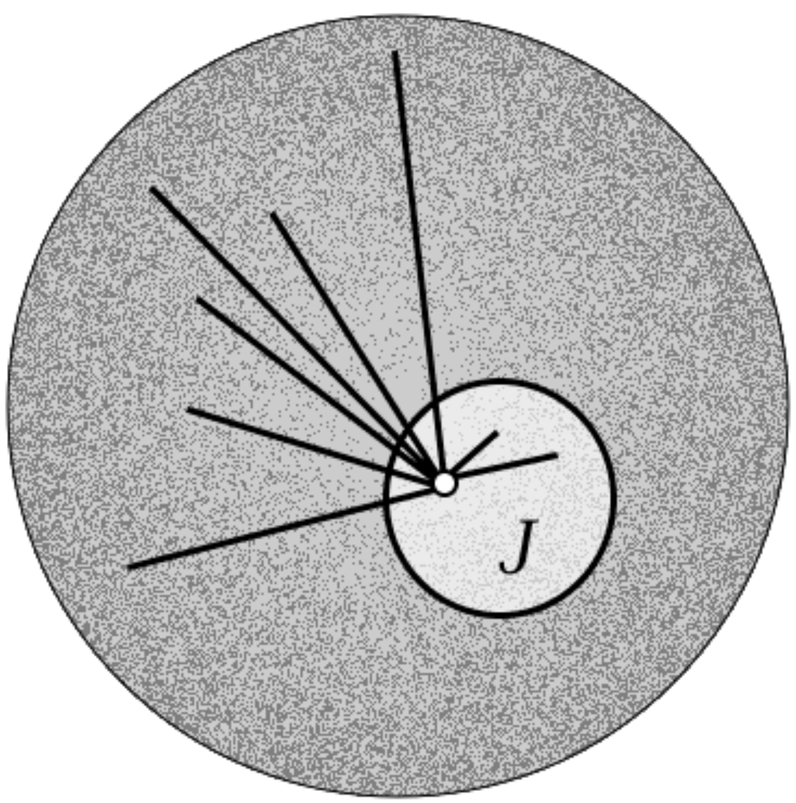


Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 1: Suppose $G[C]$ is the random graph on n vertices with edge density ϵ, and plant the sparse set J (has fewer than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|^2$ edges).



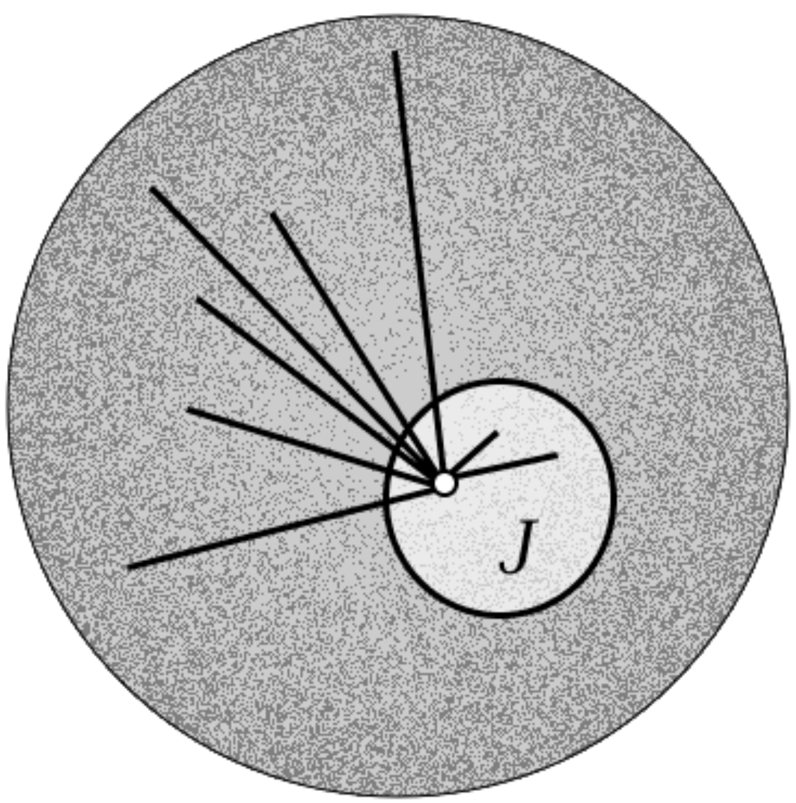
- Pick a vertex $v \in J$ with degree less than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|$ in $G[J]$.

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 1: Suppose $G[C]$ is the random graph on n vertices with edge density ϵ, and plant the sparse set J (has fewer than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|^2$ edges).



- Pick a vertex $v \in J$ with degree less than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|$ in $G[J]$.

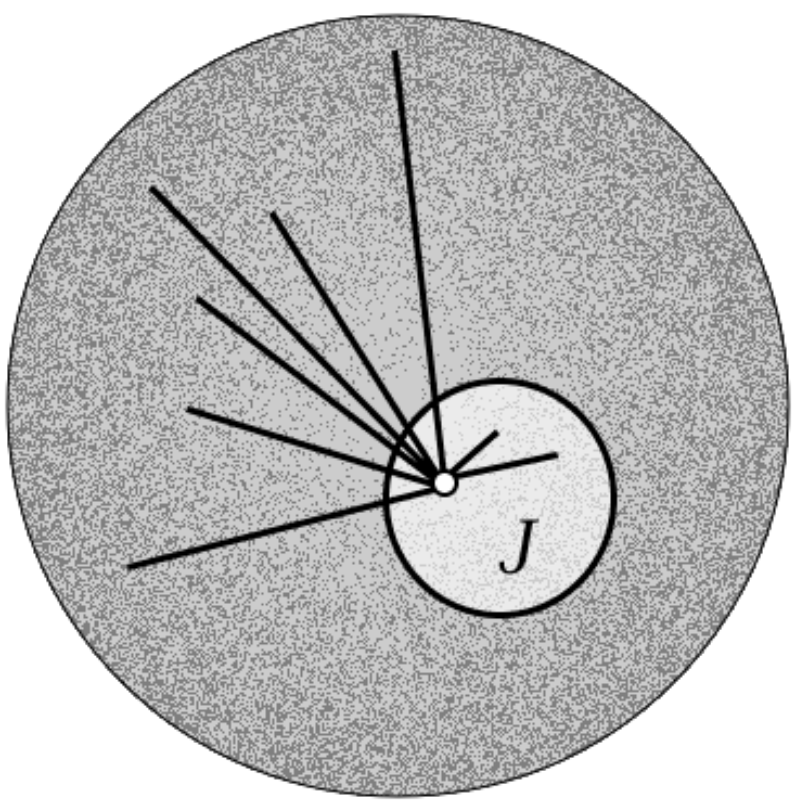
Such a vertex exists because this is the average degree in $G[J]$

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 1: Suppose $G[C]$ is the random graph on n vertices with edge density ϵ, and plant the sparse set J (has fewer than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|^2$ edges).



- Pick a vertex $v \in J$ with degree less than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|$ in $G[J]$.

  Such a vertex exists because this is the average degree in $G[J]$

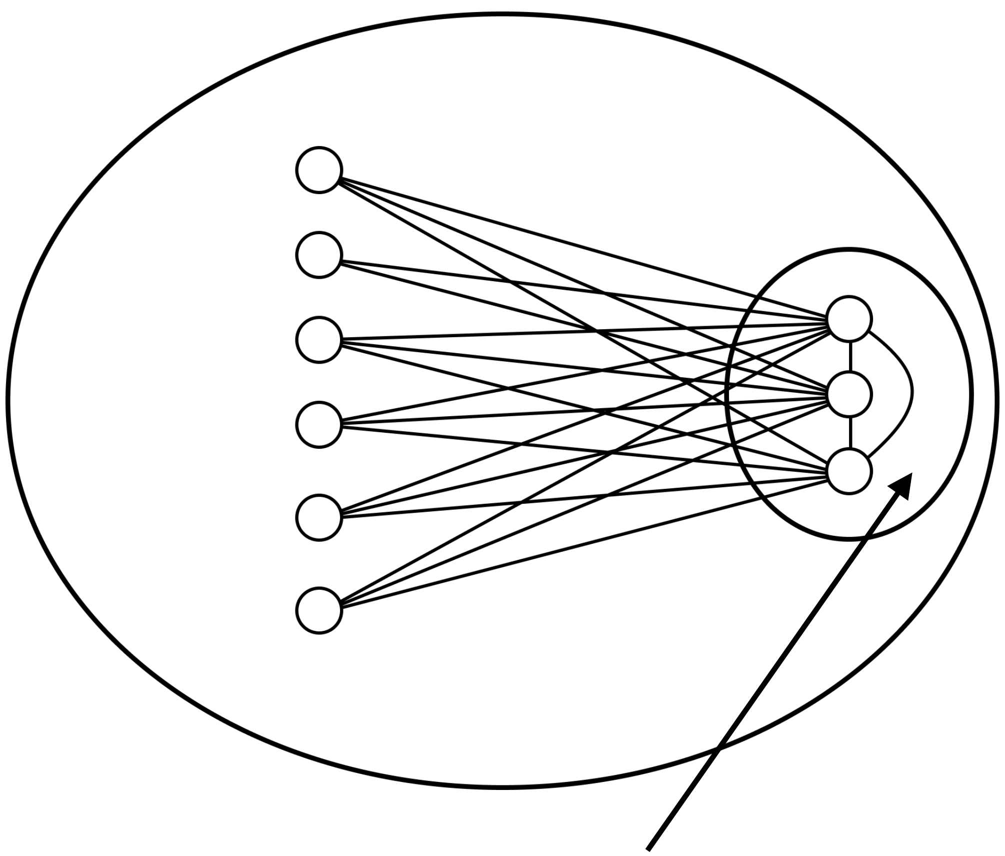
- Using "remove neighbours" operation, can remove ϵn vertices from C and less than $\frac{\epsilon}{\log^2(1/\epsilon)}|J|$ vertices of J from C.

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 2: Suppose $G[C]$ has ϵn vertices with degree n.



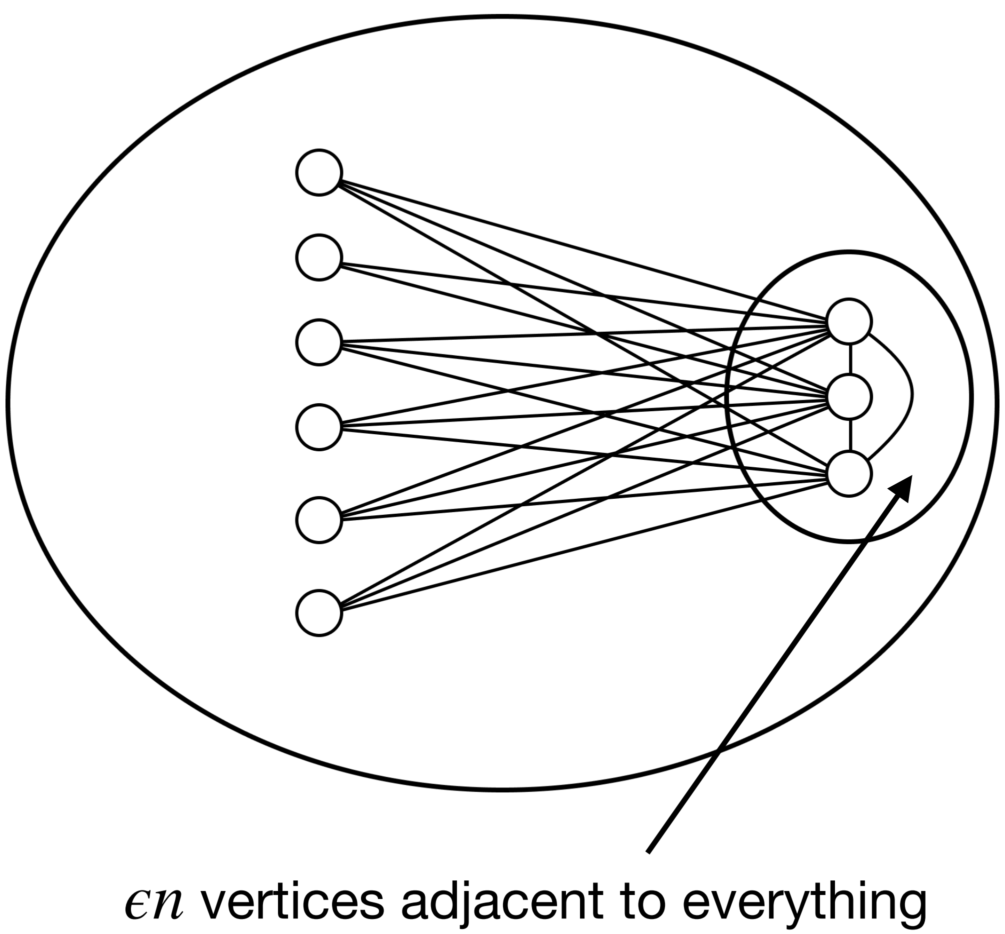
ϵn vertices adjacent to everything

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 2: Suppose $G[C]$ has ϵn vertices with degree n.



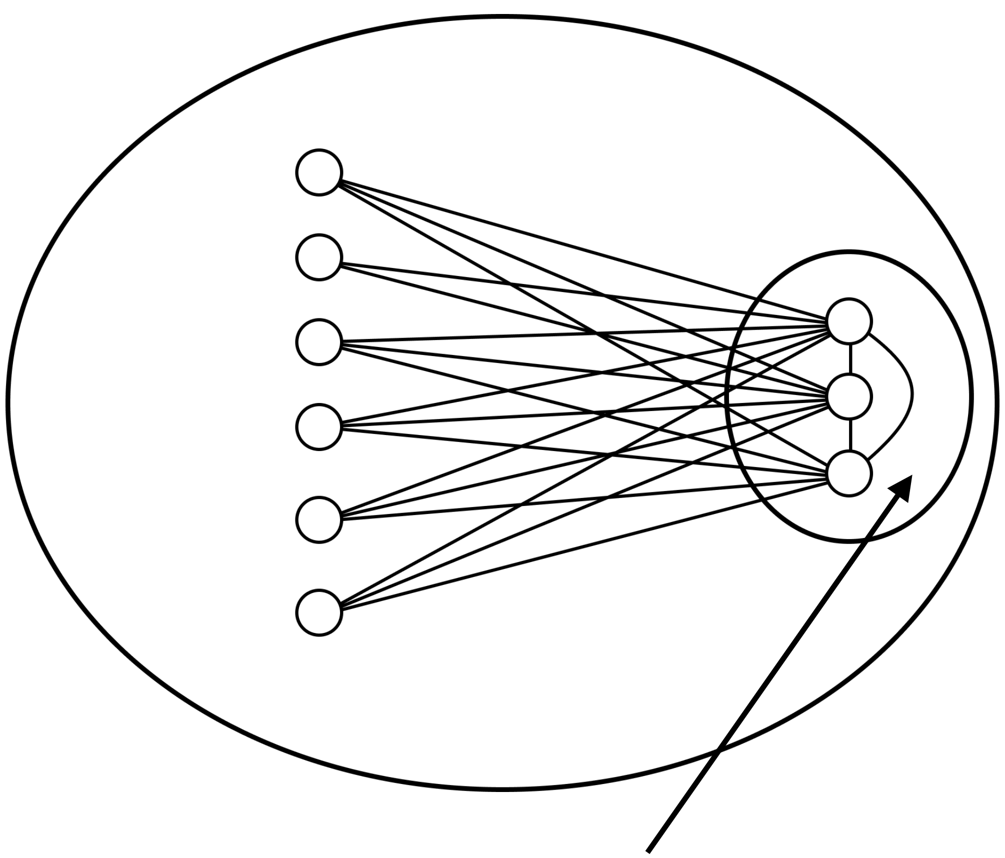
- If there is a vertex $v \in J$ with degree less than $\frac{1}{\log(1/\epsilon)}|J|$ in $G[J]$ and degree n in $G[C]$ then select v and "remove neighbours" operation

Proof of Claim: Some Examples

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Example 2: Suppose $G[C]$ has ϵn vertices with degree n.



- If there is a vertex $v \in J$ with degree less than $\frac{1}{\log(1/\epsilon)}|J|$ in $G[J]$ and degree n in $G[C]$ then select v and "remove neighbours" operation

- Otherwise, select $v \in J$ with degree less than n in $G[C]$ and "remove higher degree vertices" operation, will remove ϵn vertices from C and less than $\frac{\epsilon}{\log(1/\epsilon)}|J|$ vertices of J from C

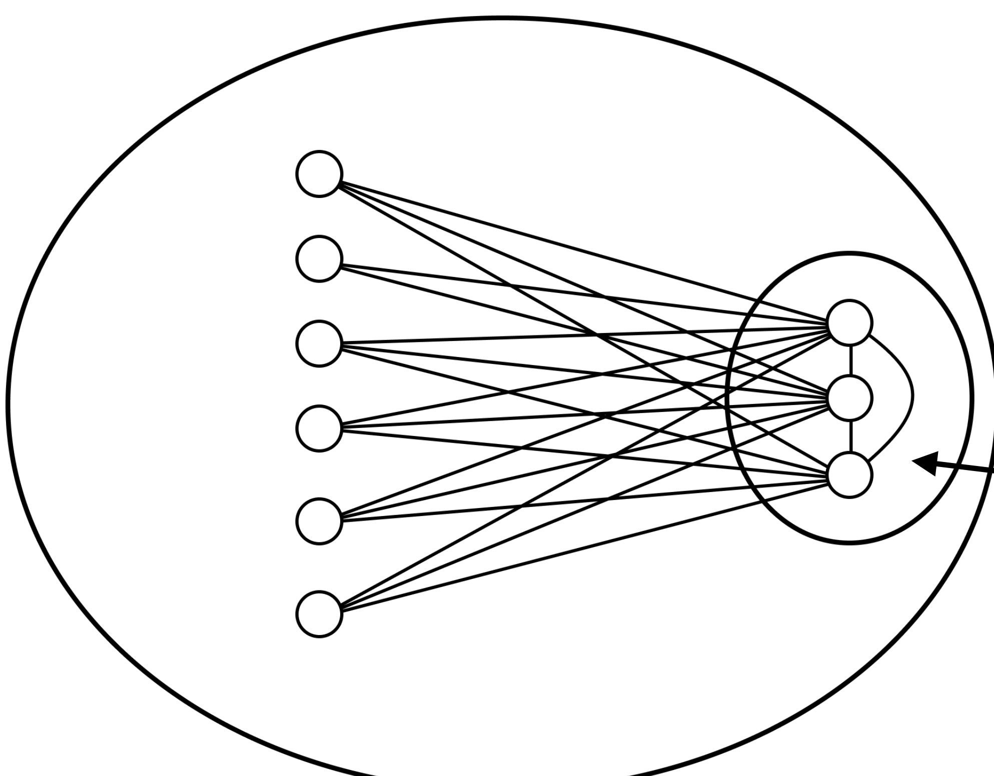
Proof of Claim

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Proof: Use the following lemma to generalize 2 cases.

Lemma: If $G[C]$ has more than ϵn^2 edges, then there exists $U \subseteq V$ such that every vertex in U has degree at least (roughly) $\frac{\epsilon n^2}{|U|}$ in $G[C]$.



$|U|$ vertices each with degree $\gtrsim \frac{\epsilon n^2}{|U|}$

Proof of Claim

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Proof: Use the following lemma to generalize 2 cases.

Lemma: If $G[C]$ has more than ϵn^2 edges, then there exists $U \subseteq V$ such that every vertex in U has degree at least (roughly) $\frac{\epsilon n^2}{|U|}$ in $G[C]$.

Proof of Claim

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Proof: Use the following lemma to generalize 2 cases.

Lemma: If $G[C]$ has more than ϵn^2 edges, then there exists $U \subseteq V$ such that every vertex in U has degree at least (roughly) $\frac{\epsilon n^2}{|U|}$ in $G[C]$.

- If there is a vertex $v \in J$ with degree less than $\frac{\epsilon n}{\log(1/\epsilon)|U|}|J|$ in $G[J]$ and degree at least $\frac{\epsilon n^2}{|U|}$ in $G[C]$ then can select v and "remove neighbours" operation

Proof of Claim

Claim: In every step of procedure where $G[C]$ has more than ϵn^2 edges:

$$\frac{\text{\# vertices removed from } C \text{ by operation}}{\text{\# vertices in } J \text{ removed from } C \text{ by operation}} \gtrsim \log(1/\epsilon) \cdot \frac{n}{|J|}$$

Proof: Use the following lemma to generalize 2 cases.

Lemma: If $G[C]$ has more than ϵn^2 edges, then there exists $U \subseteq V$ such that every vertex in U has degree at least (roughly) $\frac{\epsilon n^2}{|U|}$ in $G[C]$.

- If there is a vertex $v \in J$ with degree less than $\frac{\epsilon n}{\log(1/\epsilon)|U|}|J|$ in $G[J]$ and degree at least $\frac{\epsilon n^2}{|U|}$ in $G[C]$ then can select v and "remove neighbours" operation
- Otherwise, select $v \in J$ with degree less than $\frac{\epsilon n^2}{|U|}$ in $G[C]$ and "remove higher degree vertices" operation, will remove $|U|$ vertices from C and less than $\frac{|U|}{\log(1/\epsilon)n}|J|$ vertices of J from C

□

Towards Proving Stronger Lemma

Lemma (Restated): For any ϵ, ρ let $G = (V, E)$ be a graph such that every induced subgraph on ρn vertices has at least ϵn^2 edges. Then, there exists a set $\mathscr{C} \subseteq P(V)$ of containers that satisfies:

1. $|\mathscr{C}| \lesssim \binom{n}{1/\epsilon}$,
2. for every $C \in \mathscr{C}$, $|C| < \rho n$.
3. For every set $J \subseteq V$ such that $G[J]$ has fewer than $\frac{\epsilon}{\text{polylog}(1/\epsilon)} |J|^2$ edges, there exists $C \in \mathscr{C}$ and α such that $|C| \leq (1-\alpha)\rho n$ and
$$|C \cap J| \geq \left(1 - \frac{\alpha}{2}\right)|J|.$$

- Full proof involves showing that when container is large (close to ρn) the container procedure makes faster progress OR we can shrink the container at the end of the process (full details in paper)

Another Application of new Container Lemma

Counting Independent Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Then the number of independent sets in G is at most $2^{\frac{n}{2}\left(1+\frac{1}{d}\right)}$, and there exists a $d-$regular graph achieving this maximum ($\frac{n}{2d}$ copies of $K_{d,d}$).

[Zhao '10] [Alon '91]

Counting Independent Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Then the number of independent sets in G is at most $2^{\frac{n}{2}\left(1+\frac{1}{d}\right)}$, and there exists a $d-$regular graph achieving this maximum ($\frac{n}{2d}$ copies of $K_{d,d}$).

[Zhao '10] [Alon '91]

Conjectured by Alon ('91), proven over series of improvements:

[Sapozhenko '01], [Kahn '01], [Galvin '09]

Counting Independent Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Then the number of independent sets in G is at most $2^{\frac{n}{2}\left(1+\frac{1}{d}\right)}$, and there exists a $d-$regular graph achieving this maximum ($\frac{n}{2d}$ copies of $K_{d,d}$).

[Zhao '10] [Alon '91]

Conjectured by Alon ('91), proven over series of improvements:

[Sapozhenko '01], [Kahn '01], [Galvin '09]

One of the classic applications of container method

Counting Independent Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Then the number of independent sets in G is at most $2^{\frac{n}{2}\left(1+\frac{1}{d}\right)}$, and there exists a $d-$regular graph achieving this maximum ($\frac{n}{2d}$ copies of $K_{d,d}$).

[Zhao '10] [Alon '91]

Conjectured by Alon ('91), proven over series of improvements:

[Sapozhenko '01], [Kahn '01], [Galvin '09]

One of the classic applications of container method

Question: What about sparse sets in a $d-$regular graph? Can we use the new container lemma in place of the standard container lemma used by Sapozhenko?

Counting Sparse Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Let $k \geq \text{polylog}(n)$. Then the number of induced subgraphs in G with edge density less $\frac{1}{k}\frac{d}{n}$ is at most

$$2^{\frac{n}{2}\left(1+O\left(\frac{\text{polylog}(n)}{d}\right)+O\left(\frac{\text{polylog}(n)}{k^{1/3}}\right)\right)}.$$

[This Work]

Counting Sparse Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Let $k \geq \text{polylog}(n)$. Then the number of induced subgraphs in G with edge density less $\frac{1}{k}\frac{d}{n}$ is at most

$$2^{\frac{n}{2}\left(1+O\left(\frac{\text{polylog}(n)}{d}\right)+O\left(\frac{\text{polylog}(n)}{k^{1/3}}\right)\right)}.$$

[This Work]

Remarks:

- Observe that the edge density of G is roughly $\frac{d}{n}$ and there are at least $\frac{1}{2}2^n$ induced subgraphs with edge density at least $\frac{4d}{n}$
- [Zhao '10] says that there are at most $2^{\frac{n}{2}\left(1+\frac{1}{d}\right)}$ induced subgraphs with edge density 0.
- Our theorem shows that even for edge density up to $\frac{1}{\text{polylog}(n)}\frac{d}{n}$ there are still only roughly $2^{\frac{n}{2}}$ induced subgraphs.

Counting Sparse Sets in Regular Graphs

Theorem: Let G be a $d-$regular graph. Let $k \geq \text{polylog}(n)$. Then the number of induced subgraphs in G with edge density less $\frac{1}{k}\frac{d}{n}$ is at most

$$2^{\frac{n}{2}\left(1+O\left(\frac{\text{polylog}(n)}{d}\right)+O\left(\frac{\text{polylog}(n)}{k^{1/3}}\right)\right)}.$$

[This Work]

Remarks:

- Observe that the edge density of G is roughly $\frac{d}{n}$ and there are at least $\frac{1}{2}2^n$ induced subgraphs with edge density at least $\frac{4d}{n}$
- [Zhao '10] says that there are at most $2^{\frac{n}{2}\left(1+\frac{1}{d}\right)}$ induced subgraphs with edge density 0.
- Our theorem shows that even for edge density up to $\frac{1}{\text{polylog}(n)}\frac{d}{n}$ there are still only roughly $2^{\frac{n}{2}}$ induced subgraphs.
- $\frac{n}{2d}$ copies of $K_{d,d}$ gives lower bound of $2^{\frac{n}{2}\left(1+\frac{1}{d}+\frac{1}{k}\right)}$

Summary and Open Questions

Open Questions

Summary: We prove a new container lemma for sparse subgraphs, and use it to show that there is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Open Questions

Summary: We prove a new container lemma for sparse subgraphs, and use it to show that there is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Question 1: What about a fully tolerant tester? (i.e. an $(\epsilon_1, \epsilon_2)-$ tolerant tester for any $\epsilon_1 < \epsilon_2$)

Open Questions

Summary: We prove a new container lemma for sparse subgraphs, and use it to show that there is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Question 1: What about a fully tolerant tester? (i.e. an $(\epsilon_1, \epsilon_2)-$ tolerant tester for any $\epsilon_1 < \epsilon_2$)

Question 2: Can we use similar techniques to (tolerantly) test other graph properties? ($k-$colorability, $\rho-$cut, general partition properties)

Open Questions

Summary: We prove a new container lemma for sparse subgraphs, and use it to show that there is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Question 1: What about a fully tolerant tester? (i.e. an $(\epsilon_1, \epsilon_2)-$ tolerant tester for any $\epsilon_1 < \epsilon_2$)

Question 2: Can we use similar techniques to (tolerantly) test other graph properties? ($k-$colorability, $\rho-$cut, general partition properties)

Question 3: Are there other applications of our new container lemma for sparse subgraphs?

- Counting sparse subgraphs in $d-$regular graphs ✓

Open Questions

Summary: We prove a new container lemma for sparse subgraphs, and use it to show that there is a $\left(\frac{\epsilon}{\text{polylog}(1/\epsilon)}, \epsilon\right)-$ tolerant tester for the ρn independent set property with sample complexity $\tilde{O}(\rho^3/\epsilon^2)$.

Question 1: What about a fully tolerant tester? (i.e. an $(\epsilon_1, \epsilon_2)-$ tolerant tester for any $\epsilon_1 < \epsilon_2$)

Question 2: Can we use similar techniques to (tolerantly) test other graph properties? ($k-$colorability, $\rho-$cut, general partition properties)

Question 3: Are there other applications of our new container lemma for sparse subgraphs?

- Counting sparse subgraphs in $d-$regular graphs ✓

Thank you!