GRAPH RECONSTRUCTION FROM RANDOM SUBGRAPHS

Andrew McGregor & Rik Sengupta

University of Massachusetts Amherst

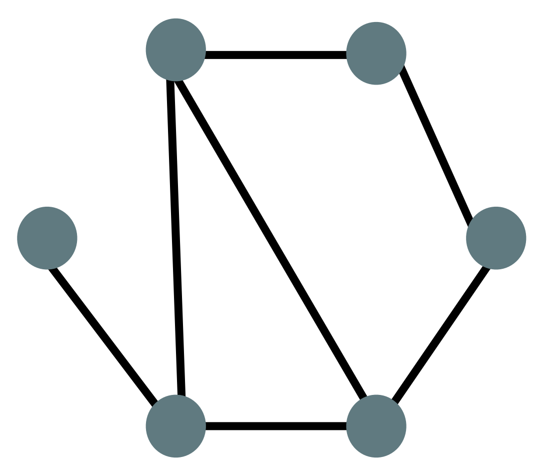
GRAPH RECONSTRUCTION

GRAPH RECONSTRUCTION

- **Model:** Unknown graph *G*, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of *G* is deleted with probability $q = 0.5$.

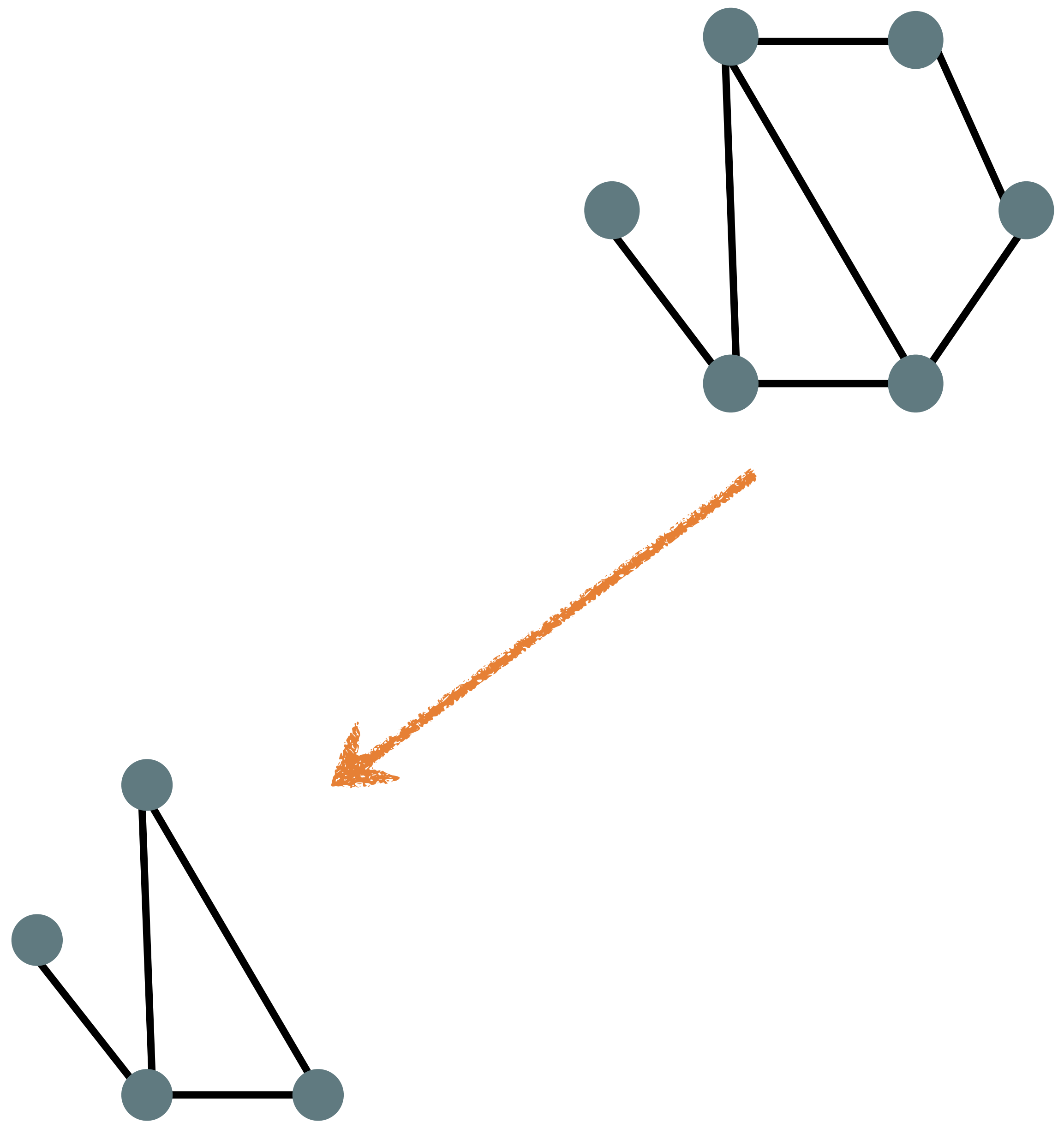
GRAPH RECONSTRUCTION

- **Model:** Unknown graph *G*, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of *G* is deleted with probability $q = 0.5$.



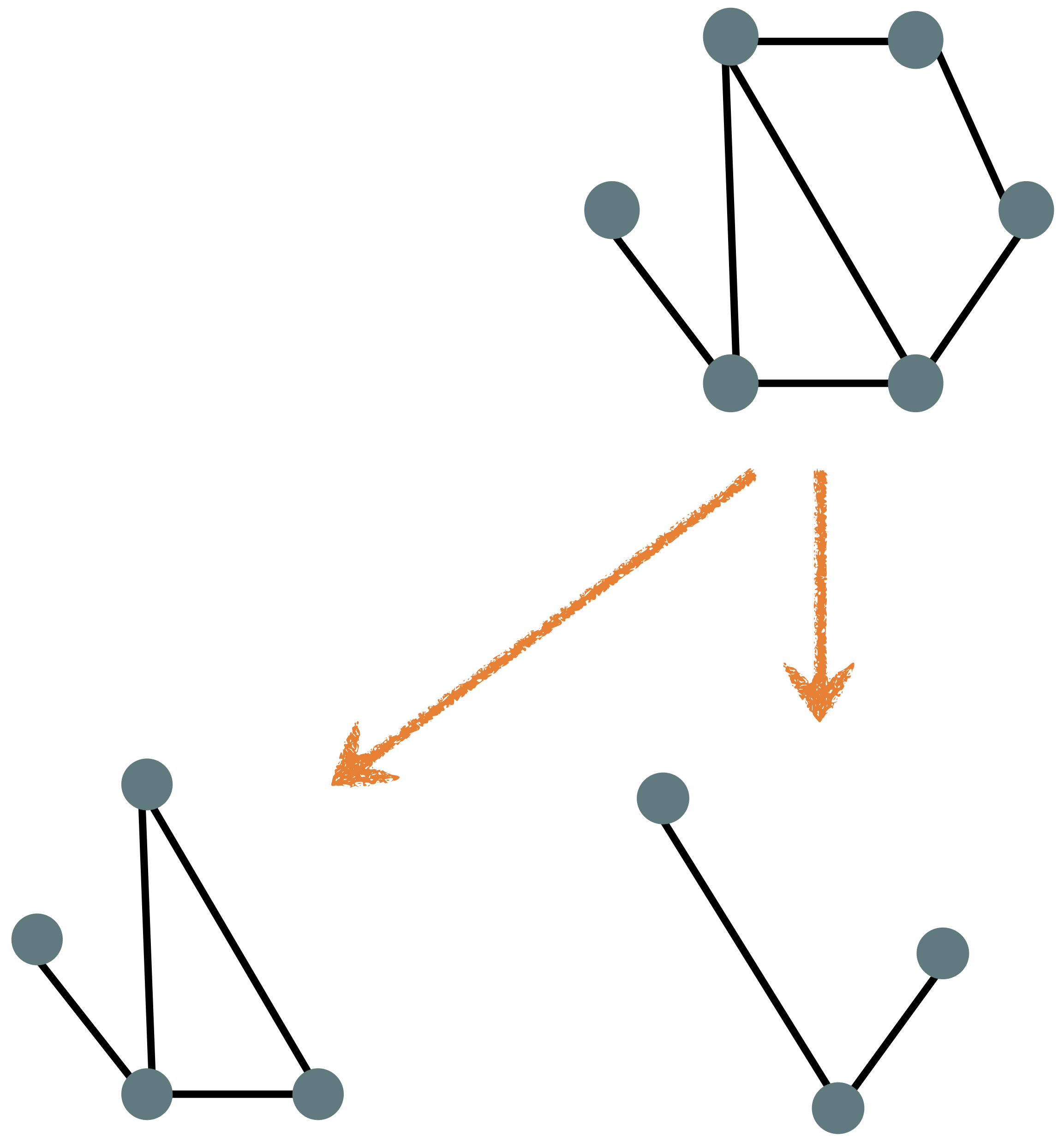
GRAPH RECONSTRUCTION

- **Model:** Unknown graph *G*, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of *G* is deleted with probability $q = 0.5$.



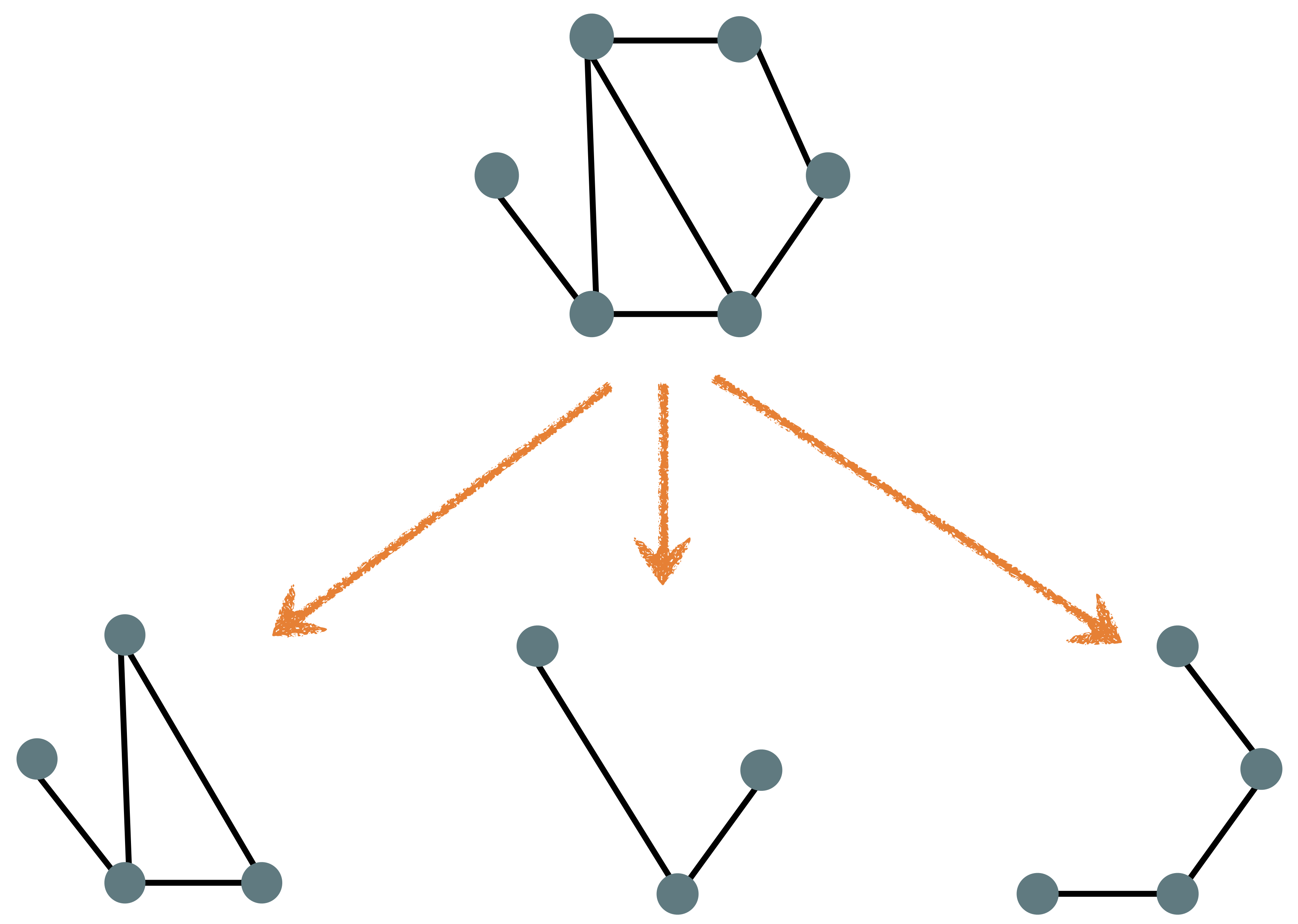
GRAPH RECONSTRUCTION

- **Model:** Unknown graph *G*, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of *G* is deleted with probability $q = 0.5$.



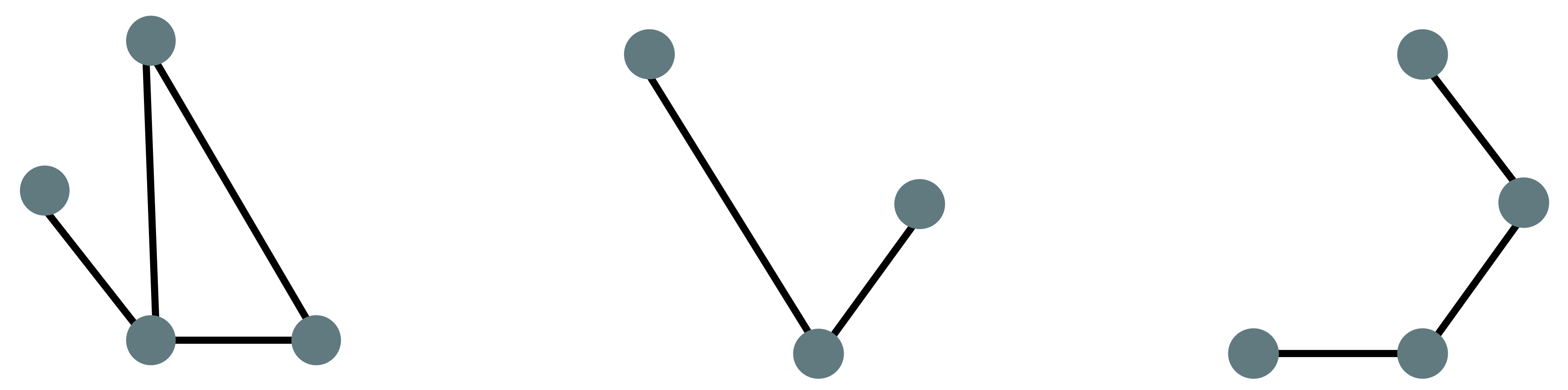
GRAPH RECONSTRUCTION

- **Model:** Unknown graph G, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of G is deleted with probability $q = 0.5$.



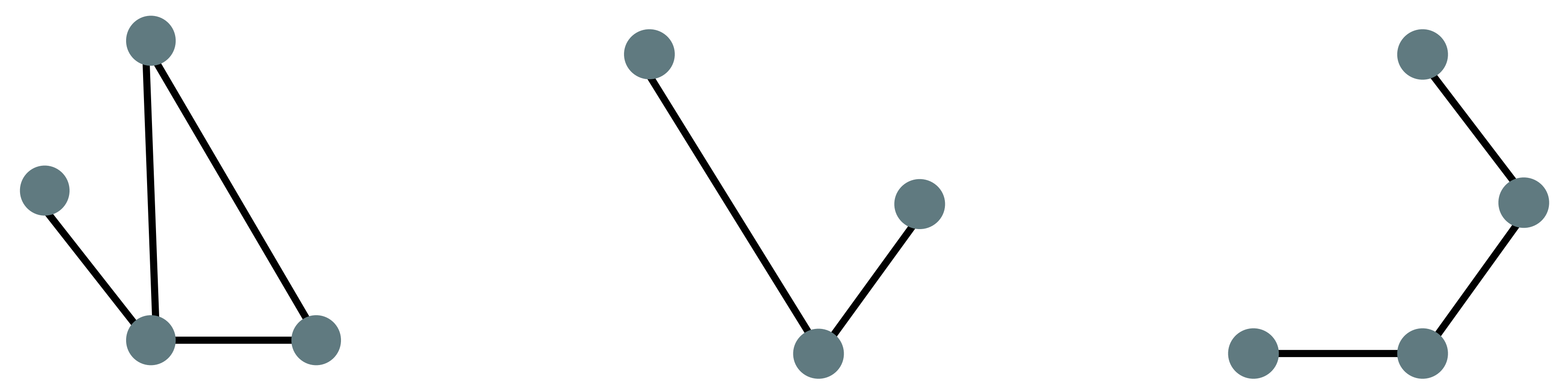
GRAPH RECONSTRUCTION

- **Model:** Unknown graph *G*, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of *G* is deleted with probability $q = 0.5$.



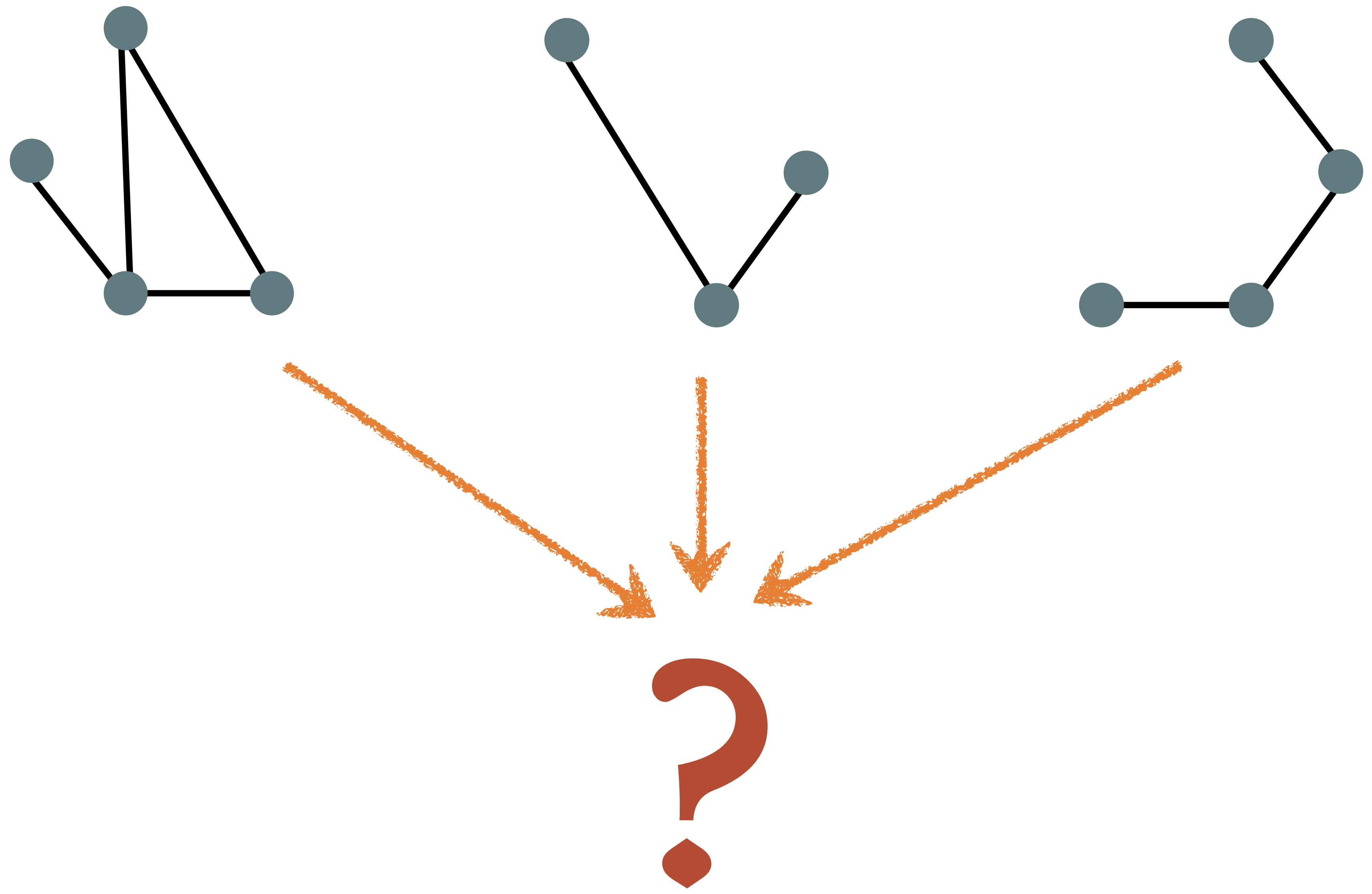
GRAPH RECONSTRUCTION

- **Model:** Unknown graph G, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of G is deleted with probability $q = 0.5$.
- **Big Question:** How many traces do we need in order to reconstruct G with high probability?



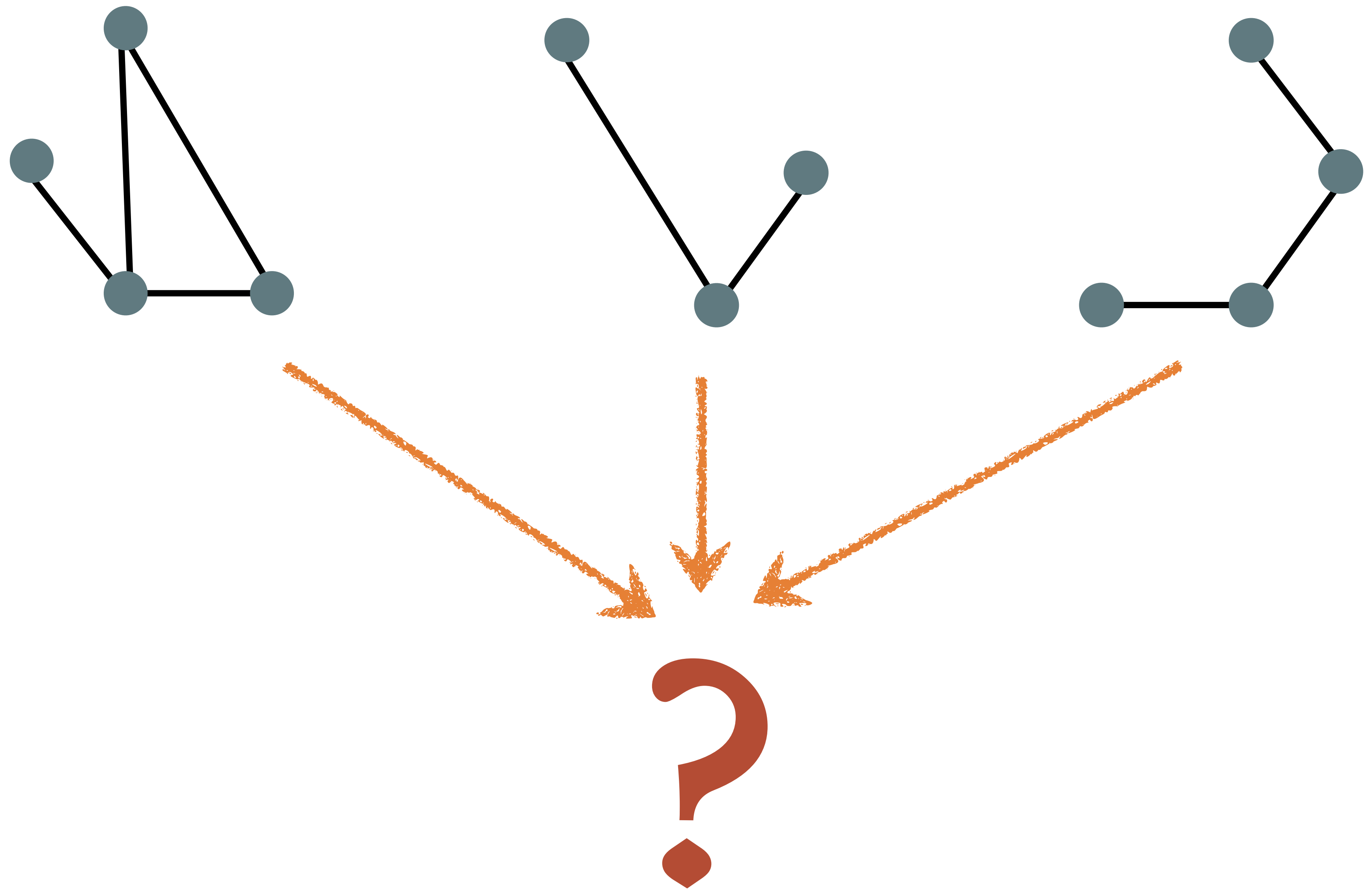
GRAPH RECONSTRUCTION

- **Model:** Unknown graph *G*, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of *G* is deleted with probability $q = 0.5$.
- **Big Question:** How many traces do we need in order to reconstruct *G* with high probability?



GRAPH RECONSTRUCTION

- **Model:** Unknown graph G, with a multiset of *traces*. A trace is a random induced subgraph, obtained after each vertex of G is deleted with probability $q = 0.5$.
- **Big Question:** How many traces do we need in order to reconstruct G with high probability?
- Of course, this is easy if the nodes were *labeled*, but we are assuming we don't know which nodes were retained.



TALK OUTLINE

TALK OUTLINE

- **Arbitrary Graphs:**

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).
- **Random Graphs:**

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).
- **Random Graphs:**
  - For almost all graphs, $O(\log n)$ traces suffice.

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).
- **Random Graphs:**
  - For almost all graphs, $O(\log n)$ traces suffice.
  - This is optimal.

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).
- **Random Graphs:**
  - For almost all graphs, $O(\log n)$ traces suffice.
  - This is optimal.
- **Adjacency Matrices:**

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).
- **Random Graphs:**
  - For almost all graphs, $O(\log n)$ traces suffice.
  - This is optimal.
- **Adjacency Matrices:**
  - For arbitrary graphs, $2^{O\left(n^{2/3}\right)}$ traces suffice.

TALK OUTLINE

- **Arbitrary Graphs:**
  - Some graphs require $2^{\Omega(n)}$ traces (matches trivial upper bound).
- **Random Graphs:**
  - For almost all graphs, $O(\log n)$ traces suffice.
  - This is optimal.
- **Adjacency Matrices:**
  - For arbitrary graphs, $2^{O\left(n^{2/3}\right)}$ traces suffice.
  - For sparse graphs, $2^{O\left(n^{1/3}\right)}$ traces suffice.

ARBITRARY GRAPHS: A LOWER BOUND

ARBITRARY GRAPHS: A LOWER BOUND

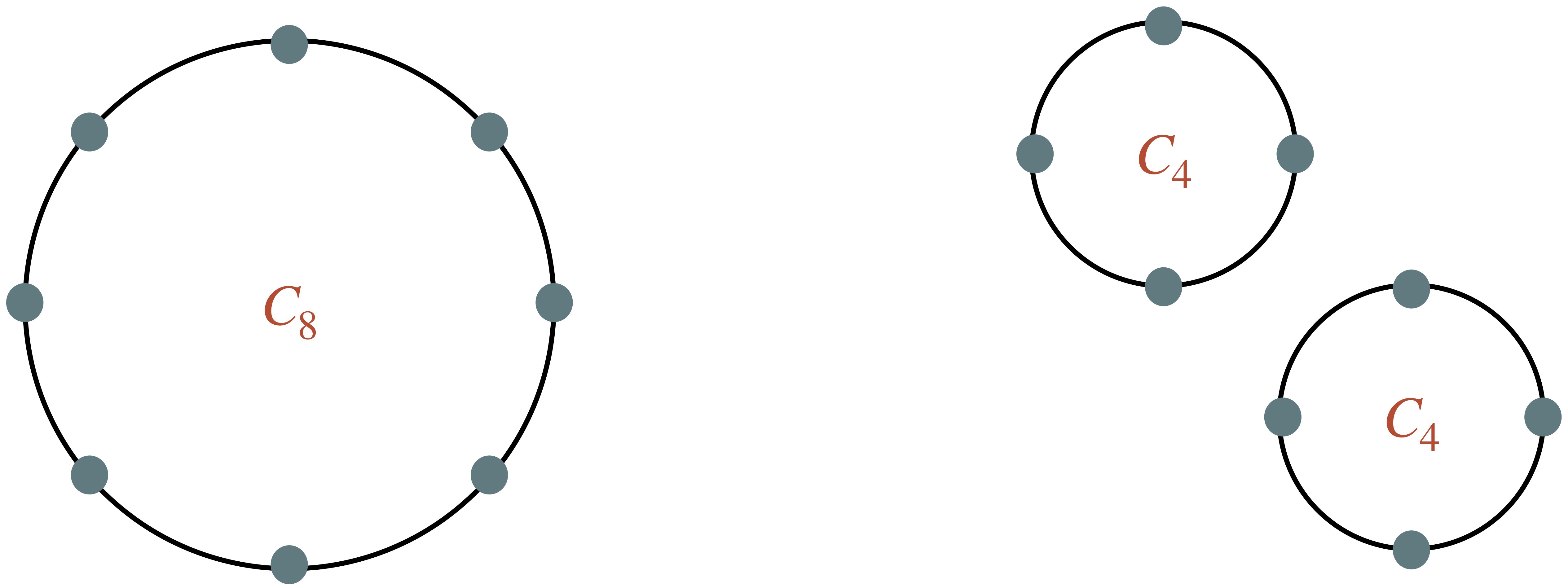
- Theorem. $2^{\Omega(n)}$ traces are required to learn an arbitrary graph with probability 0.9.

ARBITRARY GRAPHS: A LOWER BOUND

- Theorem. $2^{\Omega(n)}$ traces are required to learn an arbitrary graph with probability 0.9.
- Proof. Consider distinguishing an n-node cycle C_n from two disjoint copies of the (n/2)-node cycle $C_{n/2}$.

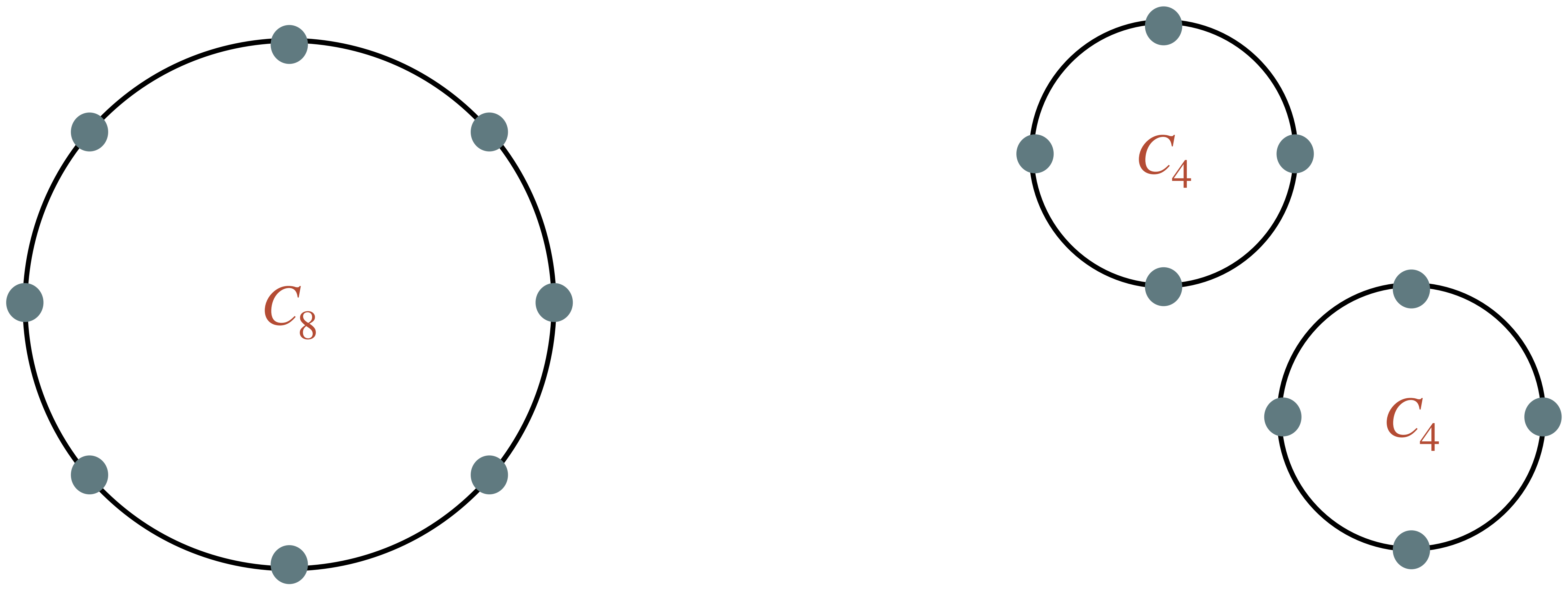
ARBITRARY GRAPHS: A LOWER BOUND

- Theorem. $2^{\Omega(n)}$ traces are required to learn an arbitrary graph with probability 0.9.
- Proof. Consider distinguishing an *n*-node cycle C_n from two disjoint copies of the (*n*/2)-node cycle $C_{n/2}$.



ARBITRARY GRAPHS: A LOWER BOUND

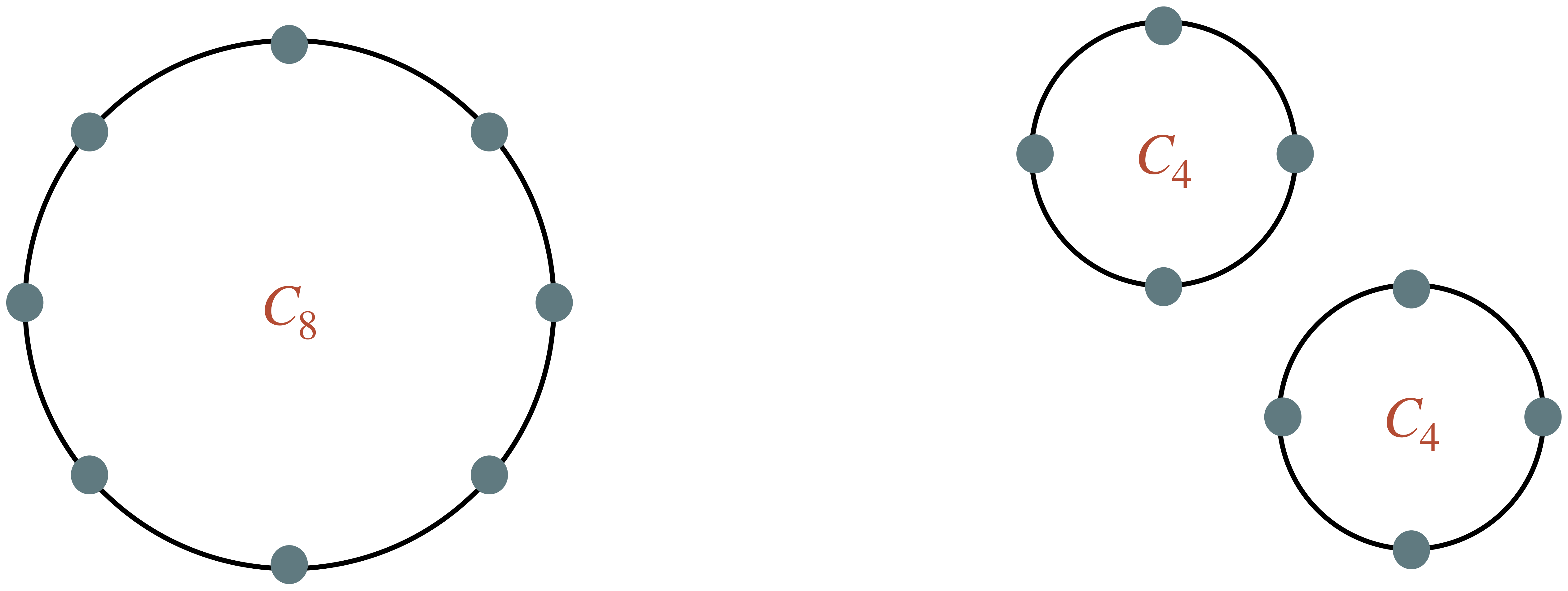
- Theorem. $2^{\Omega(n)}$ traces are required to learn an arbitrary graph with probability 0.9.
- Proof. Consider distinguishing an n-node cycle C_n from two disjoint copies of the $(n/2)$-node cycle $C_{n/2}$.



Cannot be sure which one a trace came from unless $n/2$ adjacent nodes are preserved, which happens in $2^{\Omega(n)}$ traces.

ARBITRARY GRAPHS: A LOWER BOUND

- Theorem. $2^{\Omega(n)}$ traces are required to learn an arbitrary graph with probability 0.9.
- Proof. Consider distinguishing an n-node cycle C_n from two disjoint copies of the $(n/2)$-node cycle $C_{n/2}$.



Cannot be sure which one a trace came from unless $n/2$ adjacent nodes are preserved, which happens in $2^{\Omega(n)}$ traces.

Can show the distribution of subgraphs to be virtually identical, to bound error.

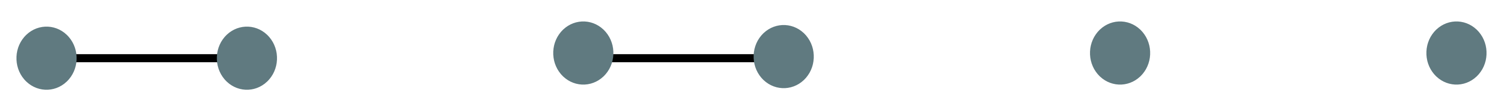
ARBITRARY GRAPHS: POSITIVE RESULTS

ARBITRARY GRAPHS: POSITIVE RESULTS

- **Graphs with Maximum Degree 1**

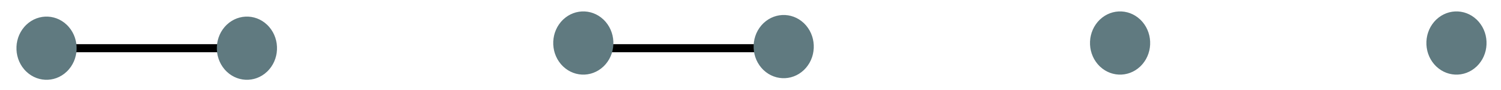
ARBITRARY GRAPHS: POSITIVE RESULTS

- **Graphs with Maximum Degree 1**
  - Theorem. We can recover graphs with maximum degree 1 in $O(n)$ traces whp.



ARBITRARY GRAPHS: POSITIVE RESULTS

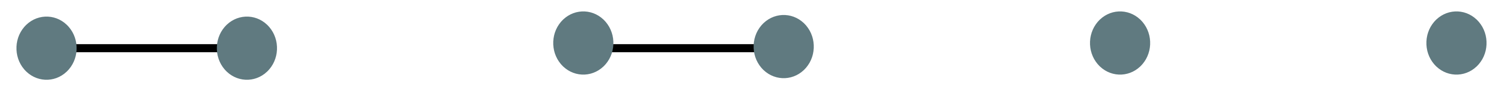
- **Graphs with Maximum Degree 1**
  - Theorem. We can recover graphs with maximum degree 1 in $O(n)$ traces whp.



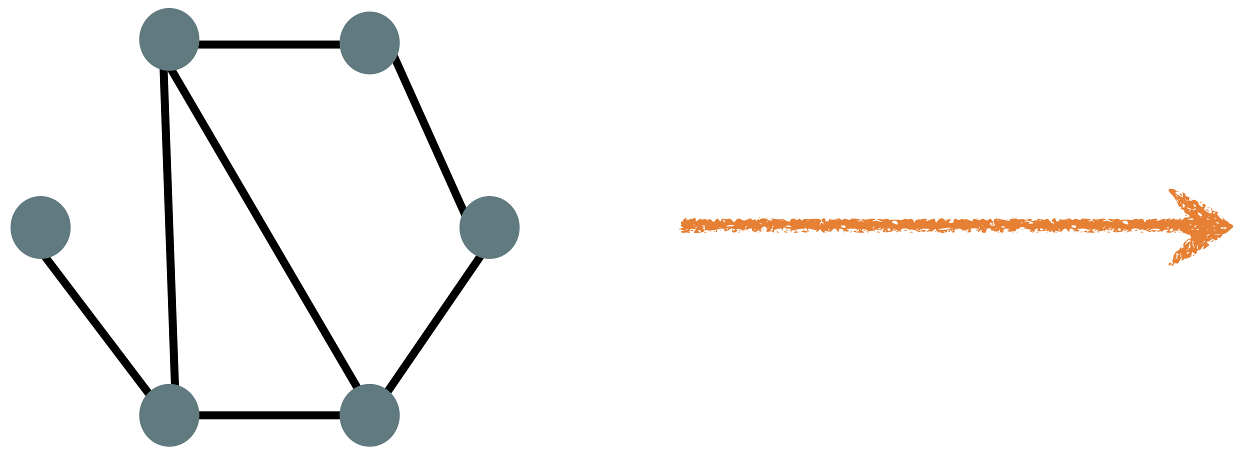
- **Degree Distribution**

ARBITRARY GRAPHS: POSITIVE RESULTS

- **Graphs with Maximum Degree 1**
  - Theorem. We can recover graphs with maximum degree 1 in $O(n)$ traces whp.



- **Degree Distribution**
  - Theorem. For any graph, we can recover the degree distribution in $\exp(O(n^{1/3}))$ traces whp.



$(3,3,3,2,2,1)$

RANDOM GRAPHS: IDEA OF ATTACK

RANDOM GRAPHS: IDEA OF ATTACK

- Theorem (Main). Almost all graphs can be reconstructed whp with $O(\log n)$ traces.

RANDOM GRAPHS: IDEA OF ATTACK

- Theorem (Main). Almost all graphs can be reconstructed whp with $O(\log n)$ traces.
- Optimality. This is the best we could hope for, since we require $\Omega(\log n)$ traces just to ensure every node appears in at least one trace.

RANDOM GRAPHS: IDEA OF ATTACK

- Theorem (Main). Almost all graphs can be reconstructed whp with $O(\log n)$ traces.
- Optimality. This is the best we could hope for, since we require $\Omega(\log n)$ traces just to ensure every node appears in at least one trace.
- Main Idea. Reconstruction would be easy if the nodes came with labels, so we try to identify common substructures to determine a *consistent* labeling of vertices across traces.

RANDOM GRAPHS: IDEA OF ATTACK

- Theorem (Main). Almost all graphs can be reconstructed whp with $O(\log n)$ traces.
- Optimality. This is the best we could hope for, since we require $\Omega(\log n)$ traces just to ensure every node appears in at least one trace.
- Main Idea. Reconstruction would be easy if the nodes came with labels, so we try to identify common substructures to determine a *consistent* labeling of vertices across traces.
- Intermediate Question. How large a common substructure do we need to be sure that they correspond to the same nodes of the original graph?

RANDOM GRAPHS: MAIN LEMMA

RANDOM GRAPHS: MAIN LEMMA

- Lemma (Müller 1976). A random graph has no *identical* subgraphs of size >n/2, and all subgraphs of size >n/2 are *asymmetric*.

RANDOM GRAPHS: MAIN LEMMA

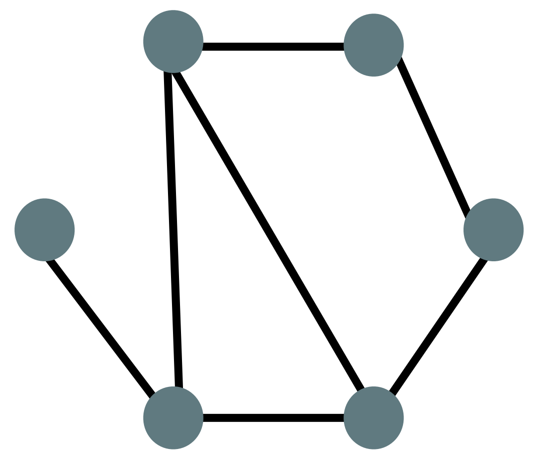
- Lemma (Müller 1976). A random graph has no *identical* subgraphs of size >n/2, and all subgraphs of size >n/2 are *asymmetric*.
- A graph G has *identical* subgraphs of size *t* if it has distinct vertex subsets of size *t* that induce isomorphic graphs.

RANDOM GRAPHS: MAIN LEMMA

- Lemma (Müller 1976). A random graph has no *identical* subgraphs of size >n/2, and all subgraphs of size >n/2 are *asymmetric*.
- A graph G has *identical* subgraphs of size *t* if it has distinct vertex subsets of size *t* that induce isomorphic graphs.
- A graph G is *symmetric* if it has a nontrivial automorphism.

RANDOM GRAPHS: MAIN LEMMA

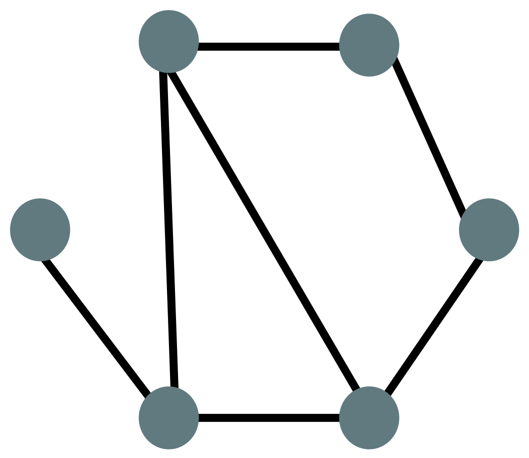
- Lemma (Müller 1976). A random graph has no *identical* subgraphs of size >n/2, and all subgraphs of size >n/2 are *asymmetric*.
- A graph G has *identical* subgraphs of size *t* if it has distinct vertex subsets of size *t* that induce isomorphic graphs.
- A graph G is *symmetric* if it has a nontrivial automorphism.



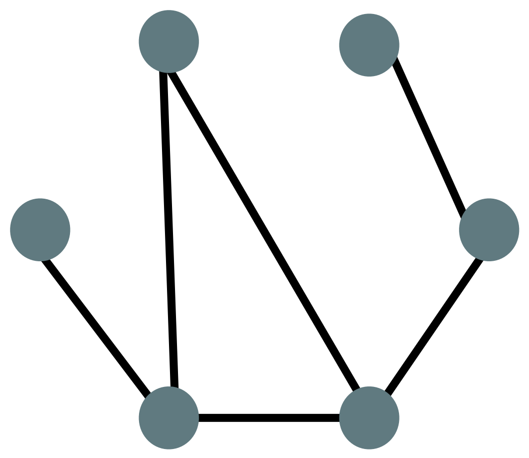
symmetric

RANDOM GRAPHS: MAIN LEMMA

- Lemma (Müller 1976). A random graph has no *identical* subgraphs of size >n/2, and all subgraphs of size >n/2 are *asymmetric*.
- A graph G has *identical* subgraphs of size *t* if it has distinct vertex subsets of size *t* that induce isomorphic graphs.
- A graph G is *symmetric* if it has a nontrivial automorphism.



symmetric



asymmetric

RANDOM GRAPHS: PROOF PART 1

RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.

RANDOM GRAPHS: PROOF PART 1

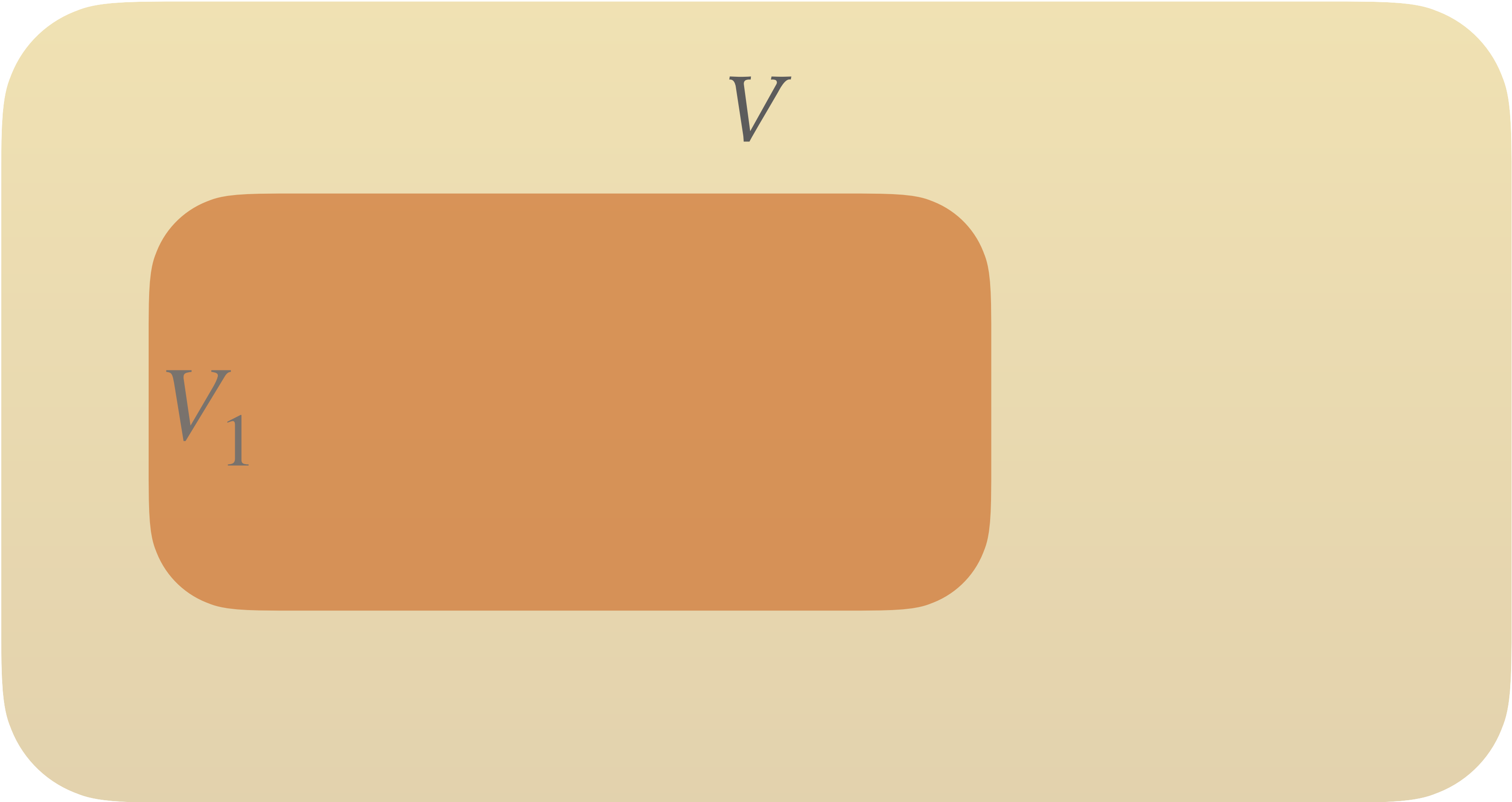
- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).

RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).

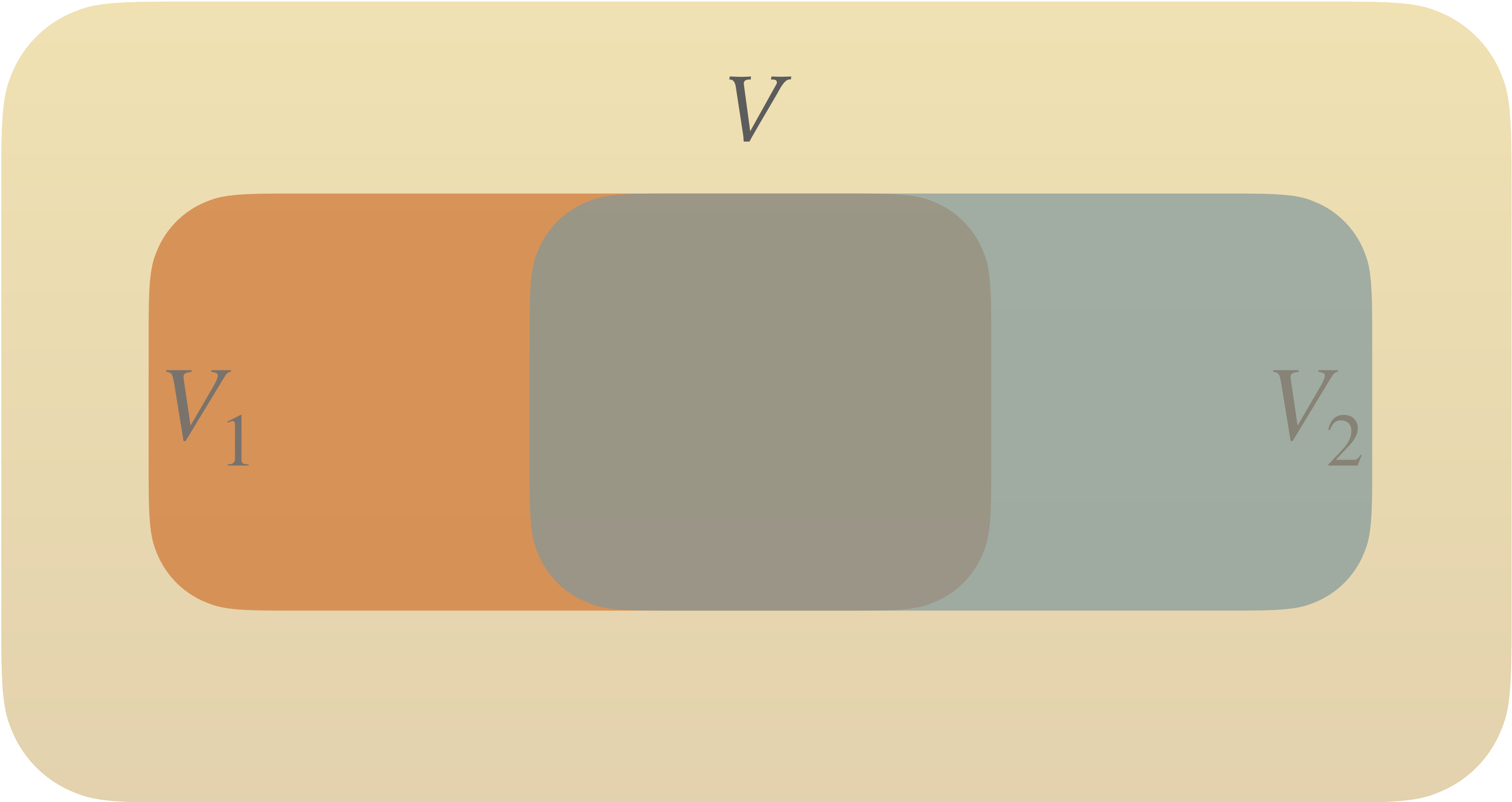
RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).



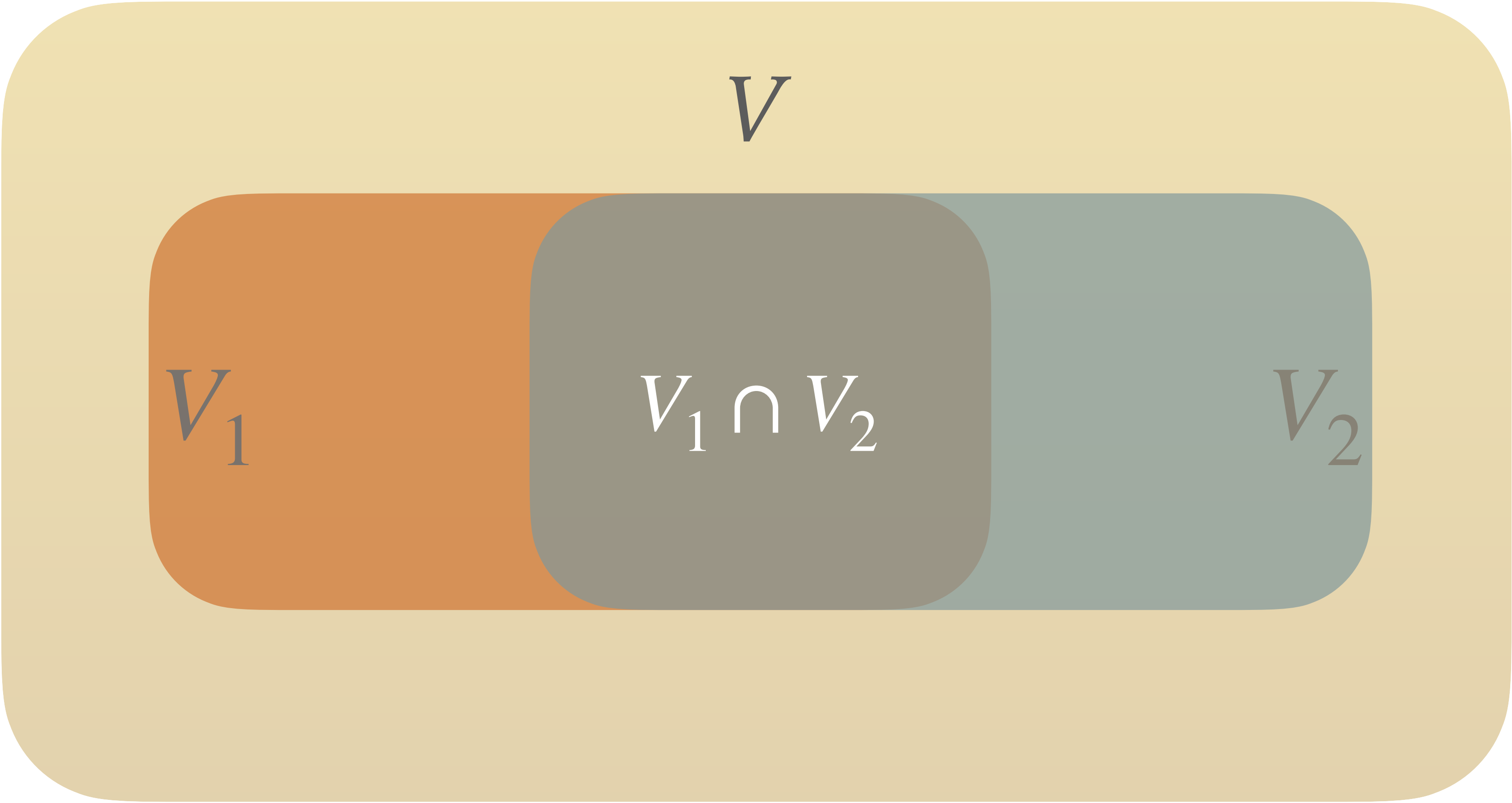
RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).



RANDOM GRAPHS: PROOF PART 1

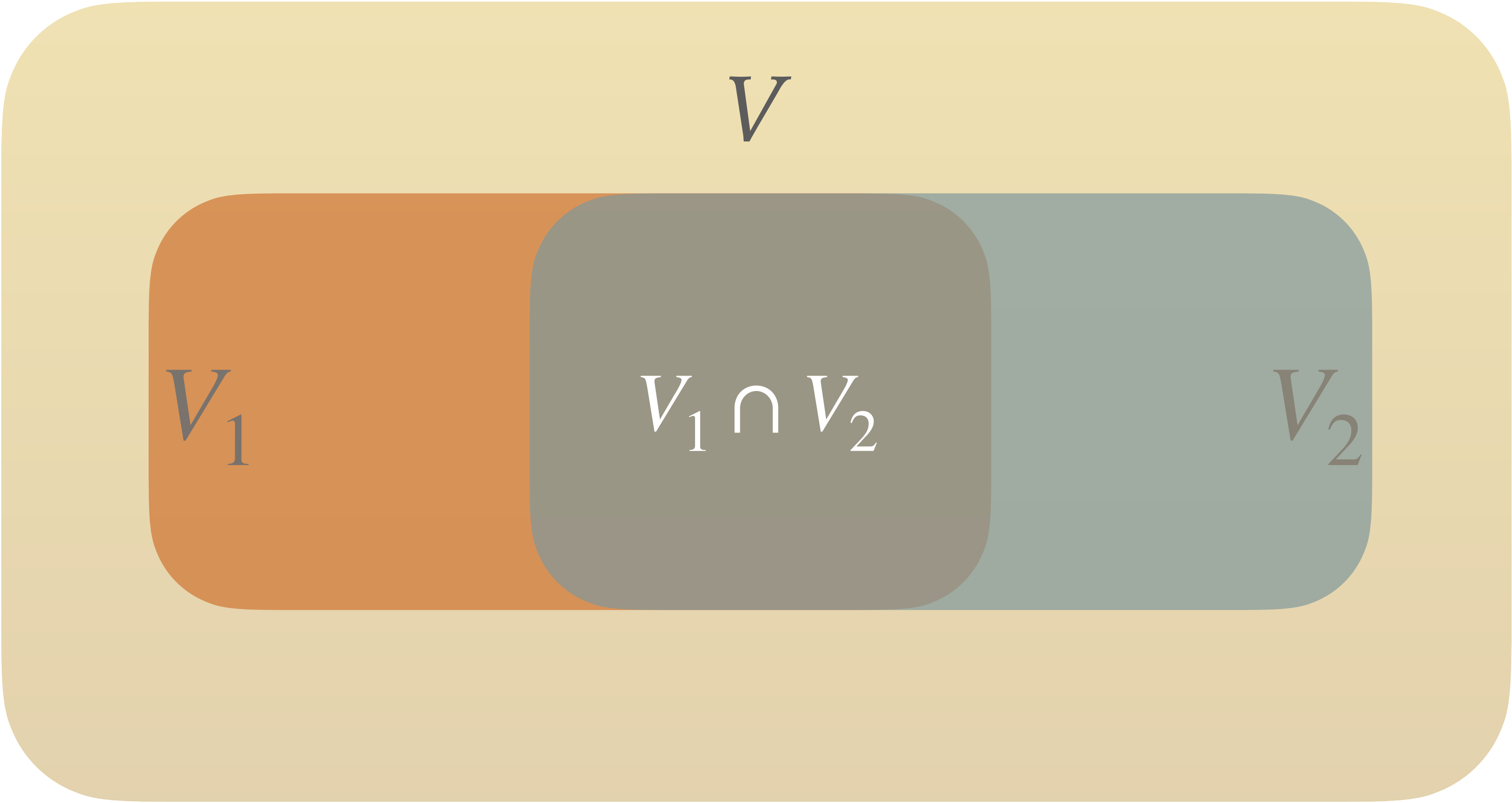
- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?



RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?

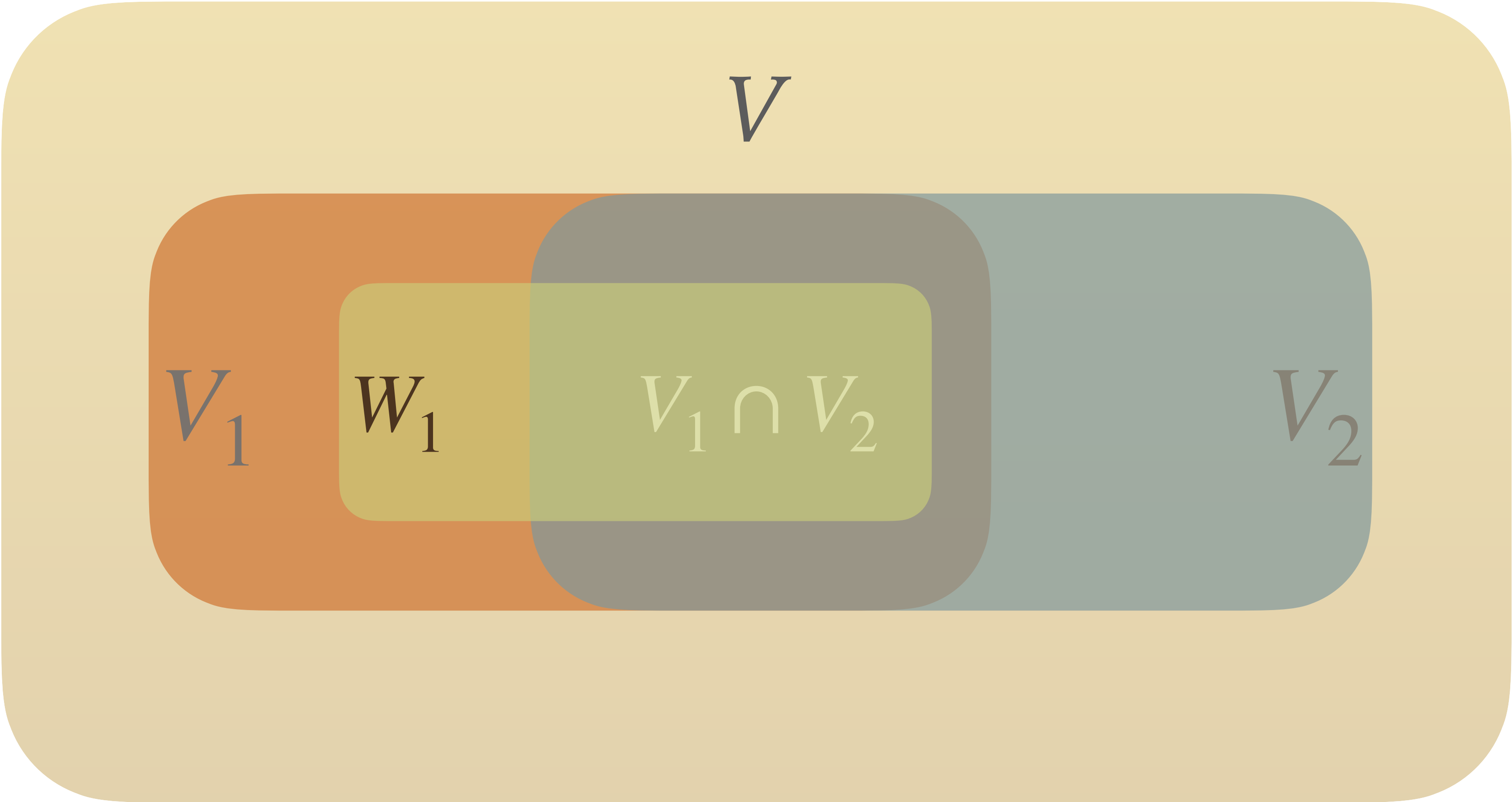
  If $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ give larger subgraphs, then $|W_1| = |W_2| > n/2$, contradicting Müller.



RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?

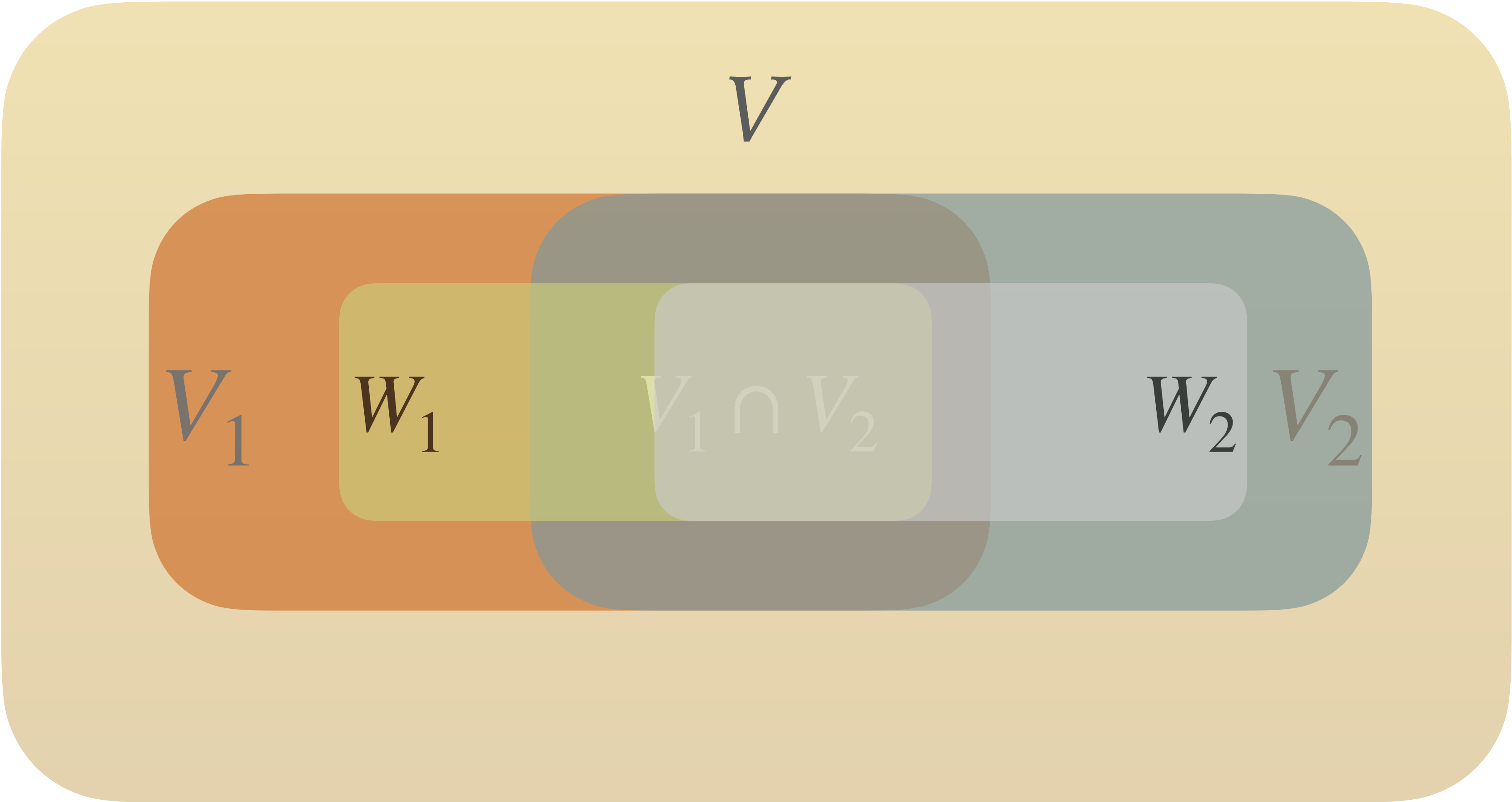
  If $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ give larger subgraphs, then $|W_1| = |W_2| > n/2$, contradicting Müller.



RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?

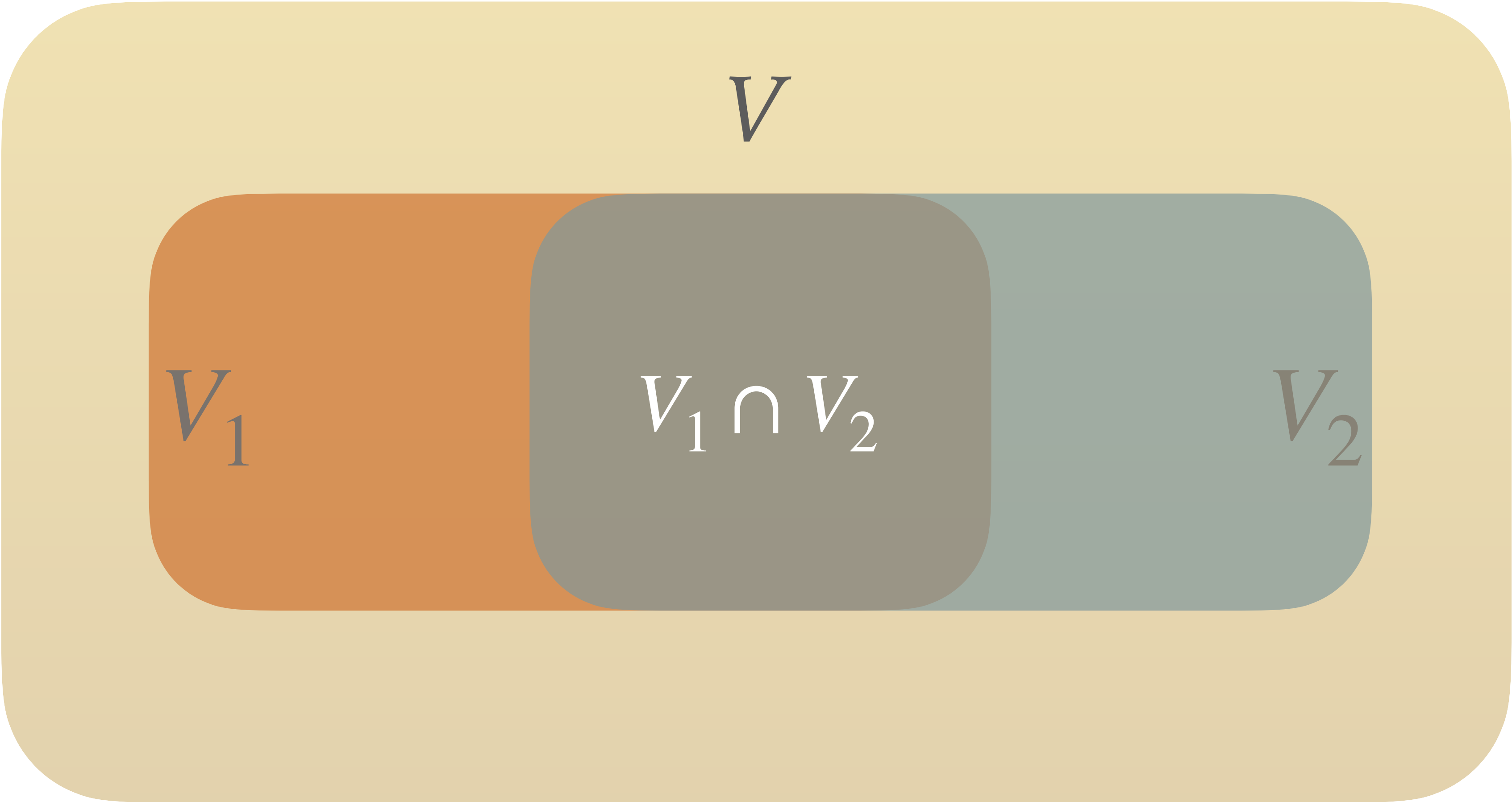
  If $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ give larger subgraphs, then $|W_1| = |W_2| > n/2$, contradicting Müller.



RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?

  If $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ give larger subgraphs, then $|W_1| = |W_2| > n/2$, contradicting Müller.

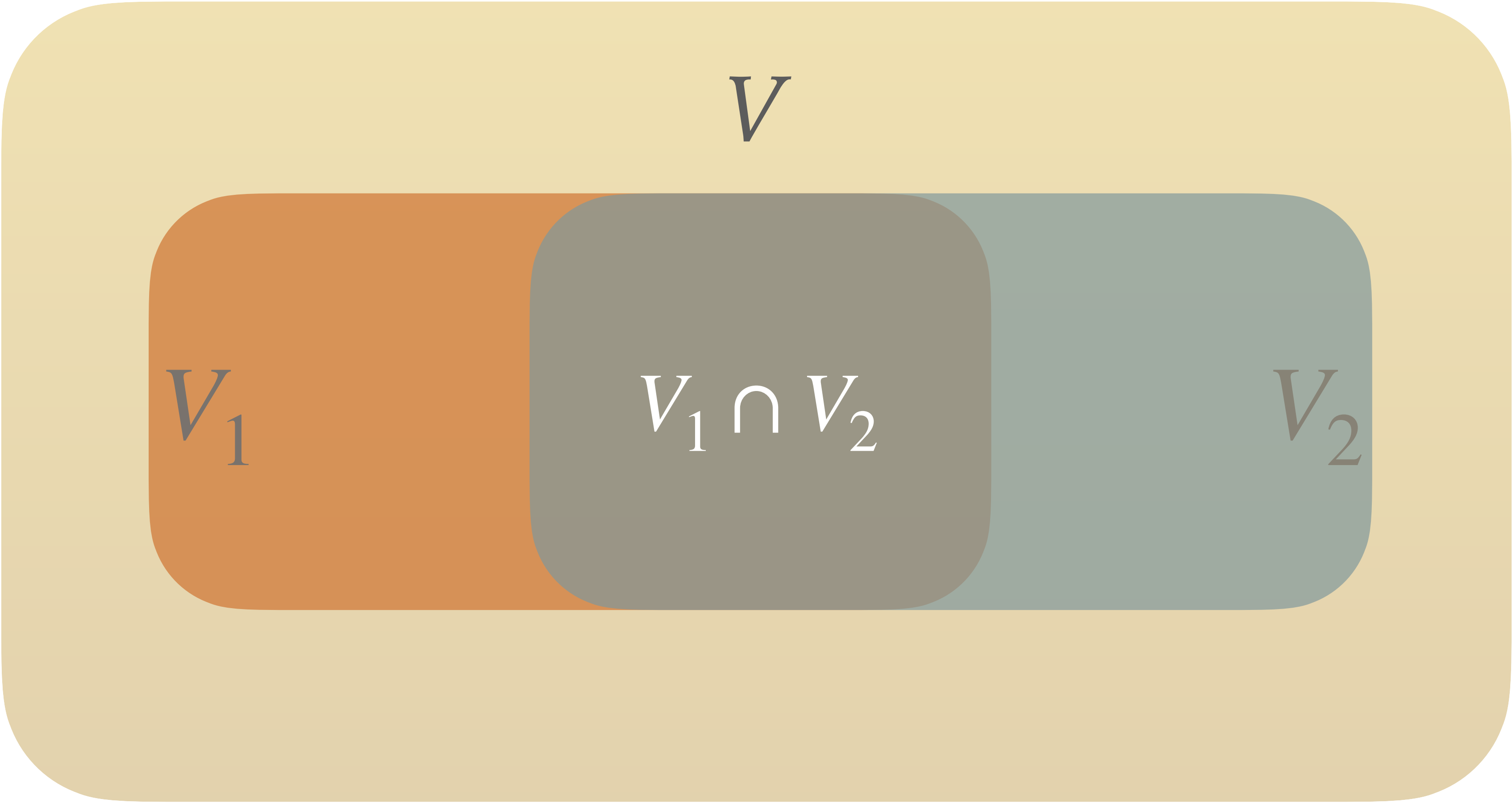


RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?

  If $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ give larger subgraphs, then $|W_1| = |W_2| > n/2$, contradicting Müller.

  Müller also implies that $G[U_1]$ is asymmetric, so there is a unique way to match U_1 with U_2.

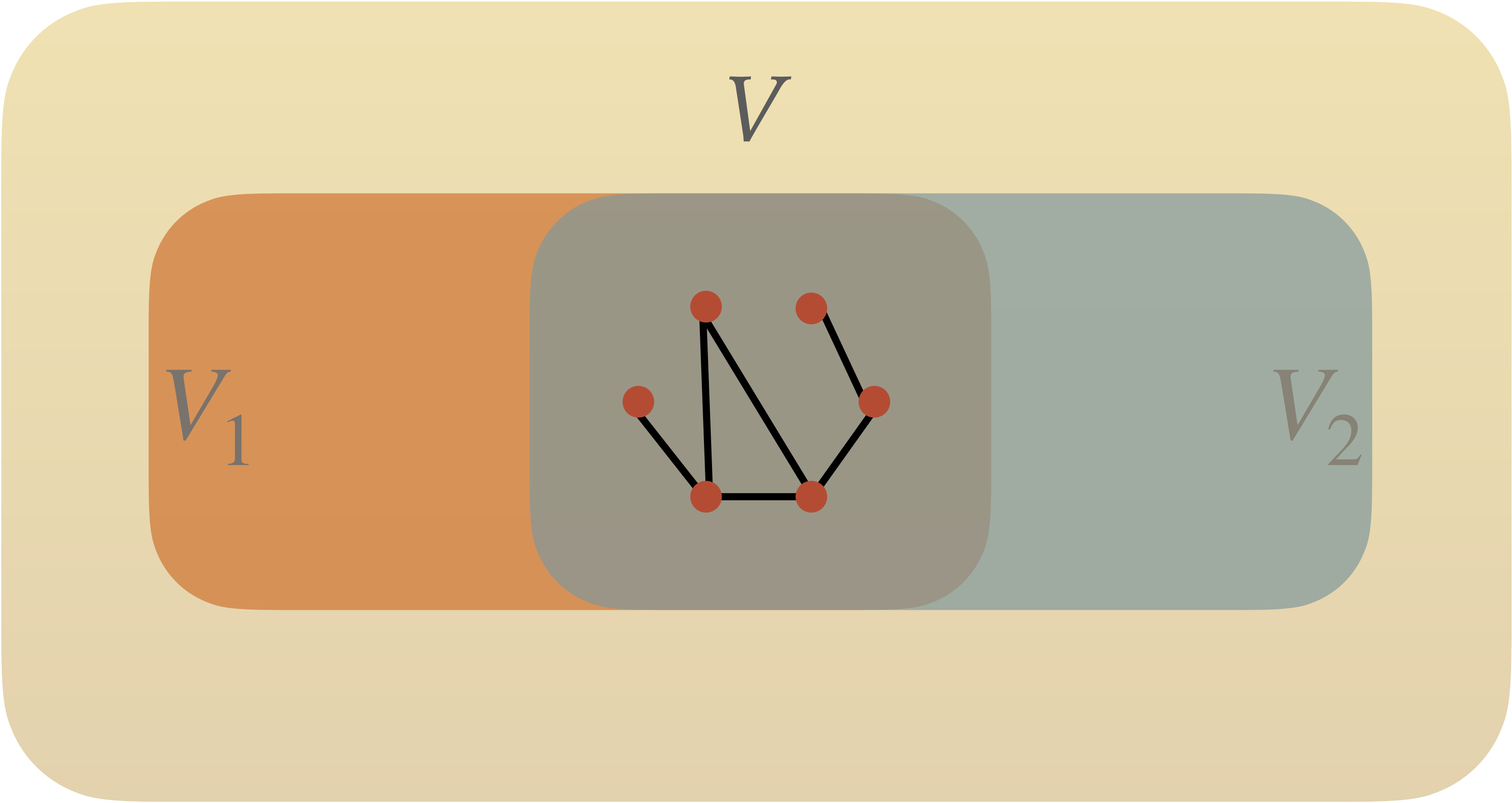


RANDOM GRAPHS: PROOF PART 1

- Temporary Assumption. The deletion probability $q < 1/4$.
- Lemma. For traces $G[V_1]$ and $G[V_2]$, let $U_1 \subseteq V_1$ and $U_2 \subseteq V_2$ give the largest common induced subgraph. Then, $U_1 = U_2$ (they are from the same nodes of G).
- Proof. Of course, $G[V_1 \cap V_2]$ is a common subgraph. But is it the largest?

  If $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ give larger subgraphs, then $|W_1| = |W_2| > n/2$, contradicting Müller.

  Müller also implies that $G[U_1]$ is asymmetric, so there is a unique way to match U_1 with U_2.



RANDOM GRAPHS: PROOF PART 2

RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

RANDOM GRAPHS: PROOF PART 2

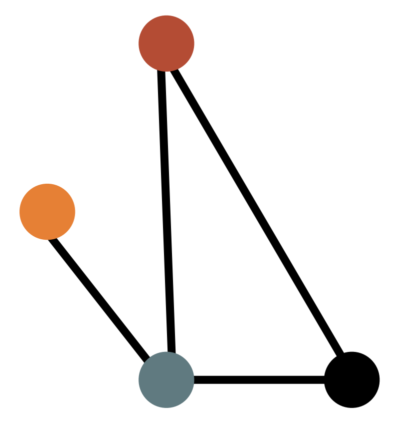
- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.

RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

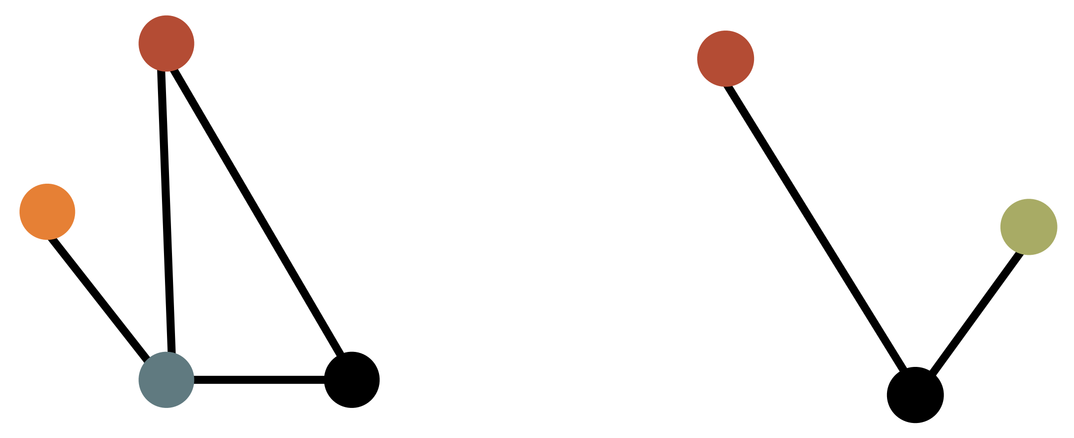
  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.



RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

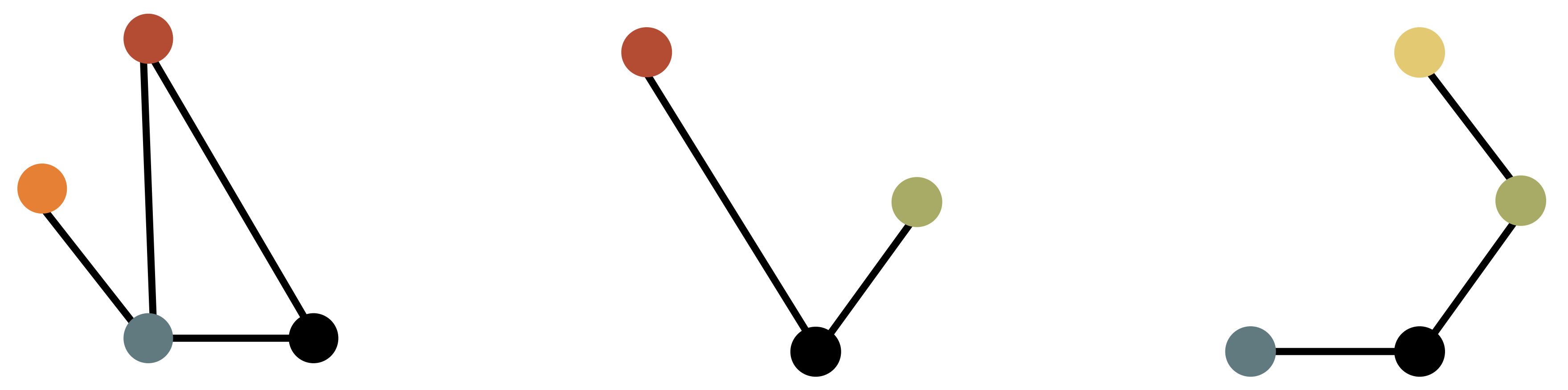
  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.



RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

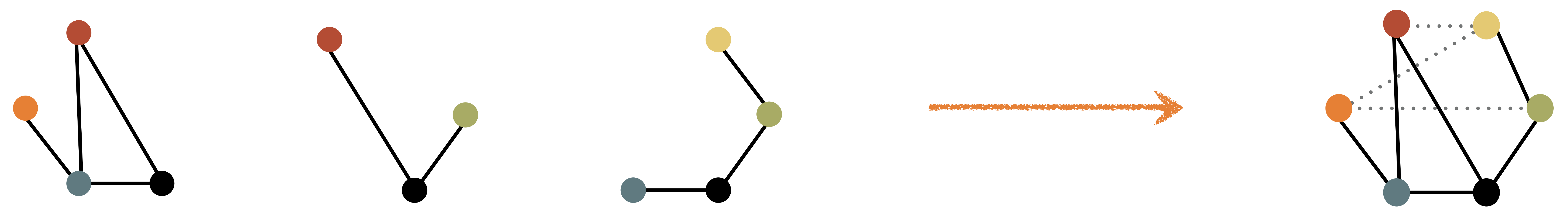
  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.



RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

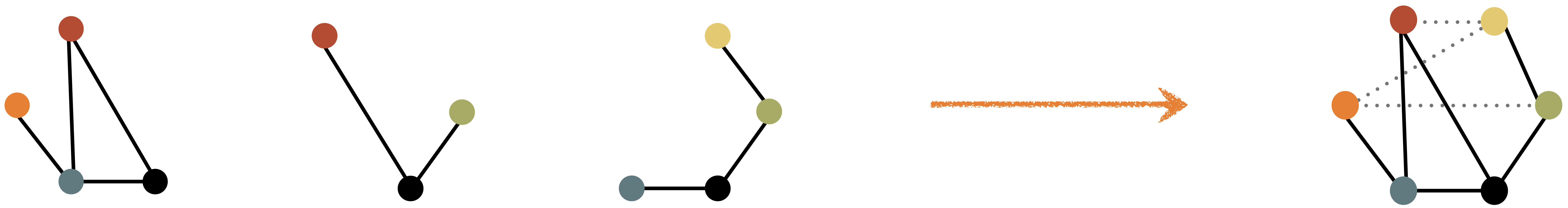
  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.



RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.

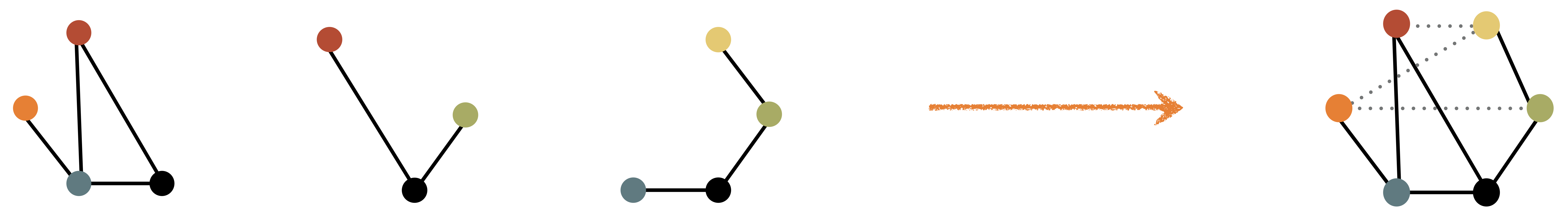


- Theorem (Main). For random graphs, $O(\log n)$ traces suffice for reconstruction.

RANDOM GRAPHS: PROOF PART 2

- Observation. If every time the same node in G shows up in two different traces, we can label it consistently, this means we can "identify" every node we see.

  So, if every vertex shows up in some trace, we will have consistently labeled all vertices, and if every pair of vertices shows up, we will have identified all the edges.



- Theorem (Main). For random graphs, $O(\log n)$ traces suffice for reconstruction.
- Extension. We can extend the result up to $q \approx 1 - \text{poly}(1/n)$ as well, using an extension to Müller, an extra subsampling step, and a modification to the proof.

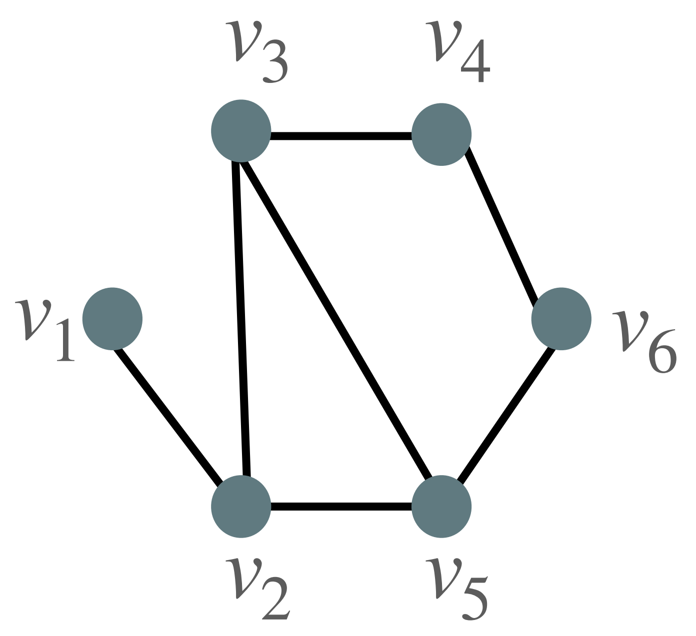
ADJACENCY MATRICES: OVERVIEW

ADJACENCY MATRICES: OVERVIEW

- Definition. The adjacency matrix of a graph $G = (V, E)$ on nodes $V = \{1, \ldots, n\}$ is an $n \times n$ binary matrix A with $A_{ij} = 1$ if and only if $(i, j) \in E$.

ADJACENCY MATRICES: OVERVIEW

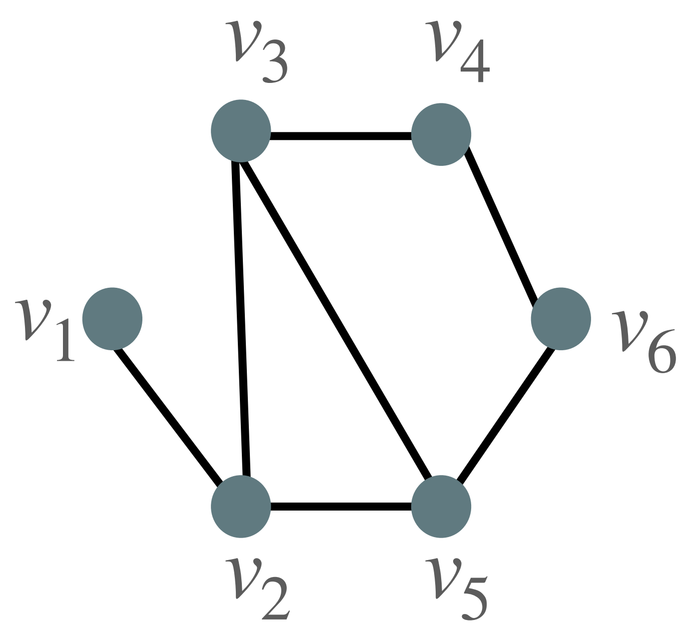
- Definition. The adjacency matrix of a graph $G = (V, E)$ on nodes $V = \{1, \ldots, n\}$ is an $n \times n$ binary matrix A with $A_{ij} = 1$ if and only if $(i, j) \in E$.



ADJACENCY MATRICES: OVERVIEW

- Definition. The adjacency matrix of a graph $G = (V, E)$ on nodes $V = \{1, \ldots, n\}$ is an $n \times n$ binary matrix A with $A_{ij} = 1$ if and only if $(i, j) \in E$.

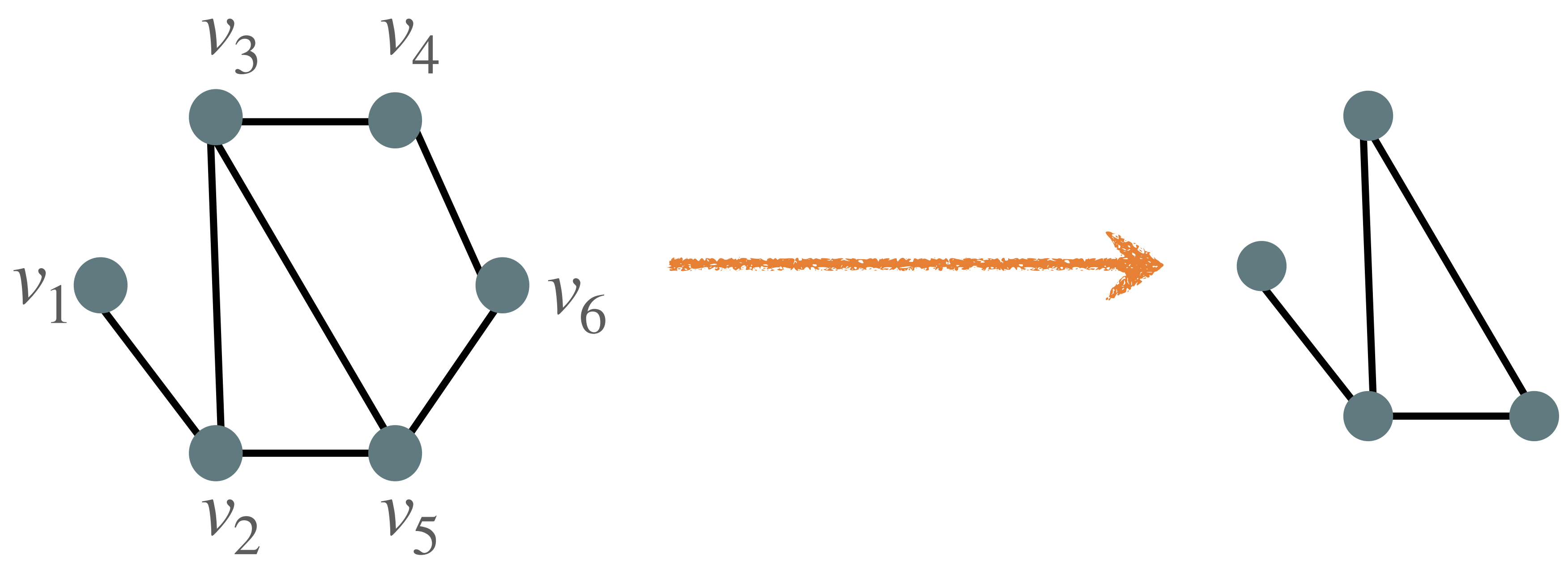
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longleftrightarrow$$



ADJACENCY MATRICES: OVERVIEW

- Definition. The adjacency matrix of a graph $G = (V, E)$ on nodes $V = \{1,\ldots,n\}$ is an $n \times n$ binary matrix A with $A_{ij} = 1$ if and only if $(i,j) \in E$.

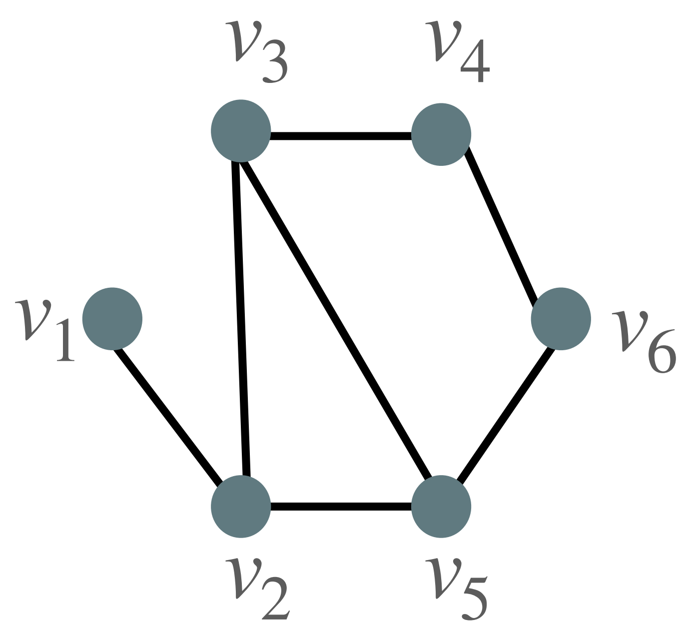
$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longleftrightarrow$$



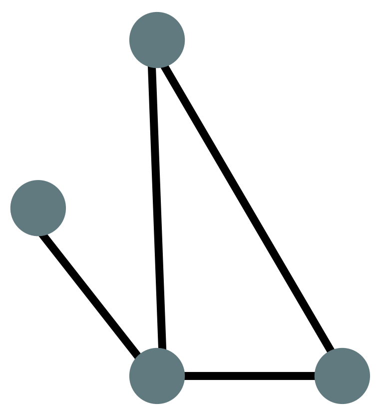
ADJACENCY MATRICES: OVERVIEW

- Definition. The adjacency matrix of a graph $G = (V, E)$ on nodes $V = \{1,\ldots,n\}$ is an $n \times n$ binary matrix A with $A_{ij} = 1$ if and only if $(i, j) \in E$.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longleftrightarrow$$



$$\longrightarrow$$

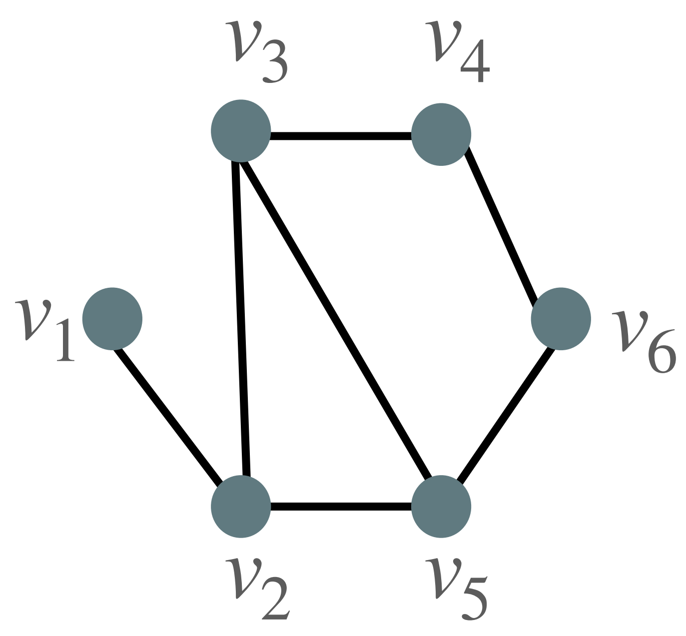


$$\longleftrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

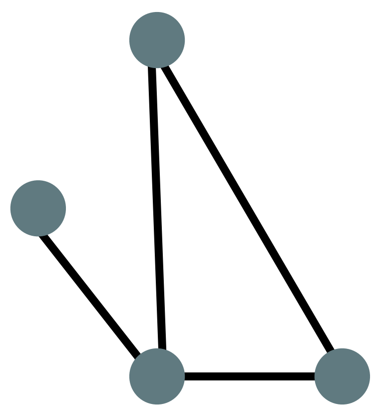
ADJACENCY MATRICES: OVERVIEW

- Definition. The adjacency matrix of a graph $G = (V, E)$ on nodes $V = \{1, \ldots, n\}$ is an $n \times n$ binary matrix A with $A_{ij} = 1$ if and only if $(i, j) \in E$.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longleftrightarrow$$



$\longrightarrow$



$$\longleftrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

- Question. How many random *symmetric* submatrices do we need to reconstruct the original graph?

ADJACENCY MATRICES: RESULT AND OUTLINE

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.
- Main Idea. Reduce the problem to a (modified) string reconstruction problem by finding a suitable encoding method.

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.
- Main Idea. Reduce the problem to a (modified) string reconstruction problem by finding a suitable encoding method.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.
- Main Idea. Reduce the problem to a (modified) string reconstruction problem by finding a suitable encoding method.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.
- Main Idea. Reduce the problem to a (modified) string reconstruction problem by finding a suitable encoding method.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.
- Main Idea. Reduce the problem to a (modified) string reconstruction problem by finding a suitable encoding method.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

ADJACENCY MATRICES: RESULT AND OUTLINE

- Theorem (Main). $\exp(O(n^{2/3}))$ traces suffice for arbitrary graphs, and $\exp(O(n^{1/3}))$ for sparse graphs.
- Main Idea. Reduce the problem to a (modified) string reconstruction problem by finding a suitable encoding method.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Can use complex analysis and moment estimation as in trace reconstruction, albeit with modifications.

OPEN PROBLEMS AND FUTURE DIRECTIONS

OPEN PROBLEMS AND FUTURE DIRECTIONS

- Conjecture. We can extend the results for random graphs to even higher values of q as well, e.g., $q = 1 - o(1)$.

OPEN PROBLEMS AND FUTURE DIRECTIONS

- Conjecture. We can extend the results for random graphs to even higher values of q as well, e.g., $q = 1 - o(1)$.
- Question. What is a lower bound for the sample complexity of graph reconstruction? Can we beat the lower bound for string reconstruction?

OPEN PROBLEMS AND FUTURE DIRECTIONS

- Conjecture. We can extend the results for random graphs to even higher values of q as well, e.g., $q = 1 - o(1)$.
- Question. What is a lower bound for the sample complexity of graph reconstruction? Can we beat the lower bound for string reconstruction?
- Question. What are other structures that can have natural analogues for these questions, and what techniques can we inherit from these results?

THANK YOU!

THANK YOU!