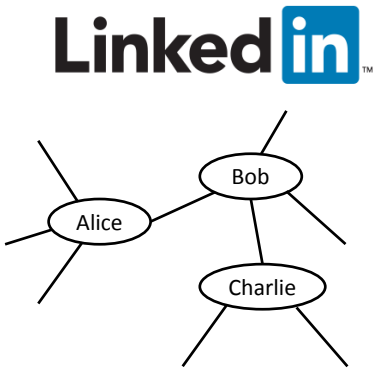
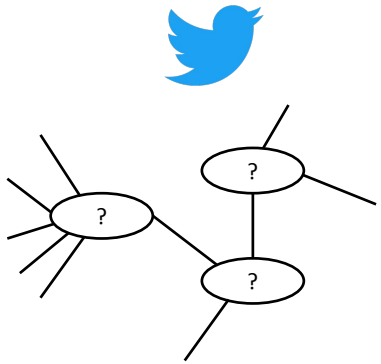
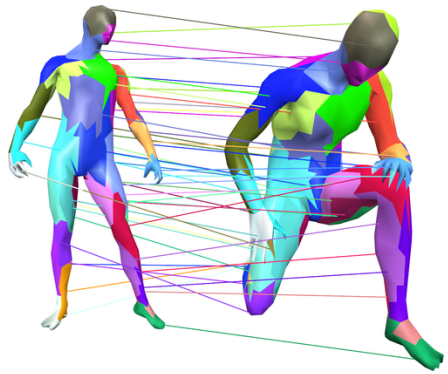
Graph Matching for Stochastic Block Models

Jian Ding, Peking University

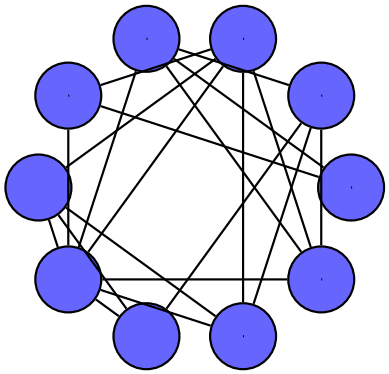
with Guanyi Chen, Shuyang Gong, Zhangsong Li

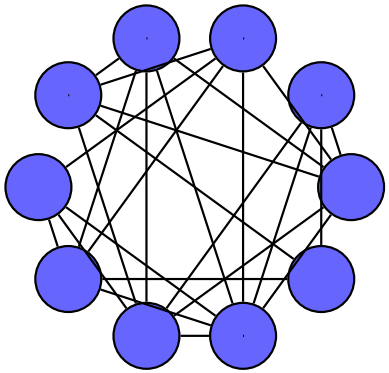






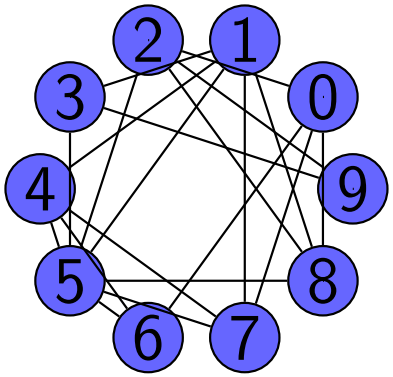
What is graph matching?



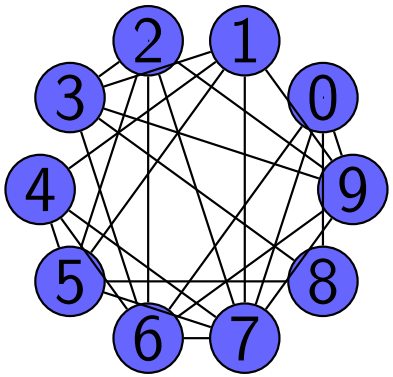


- Graphs: abstract models of objects and their connections.

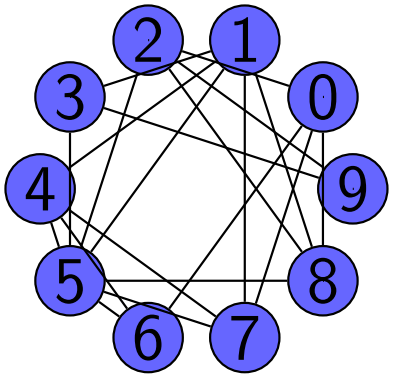
What is graph matching?



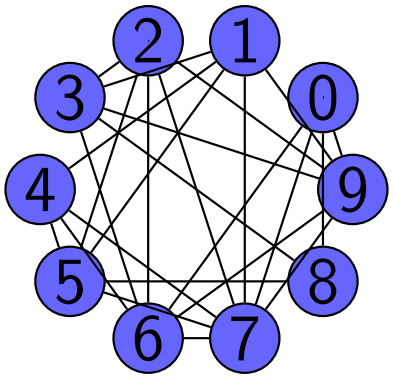
- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.



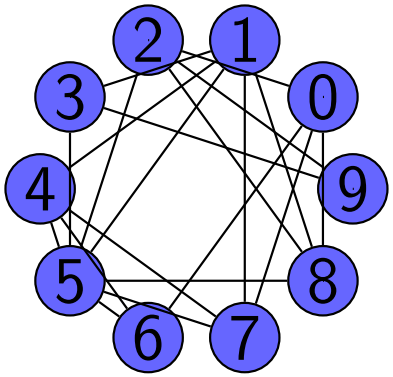
What is graph matching?

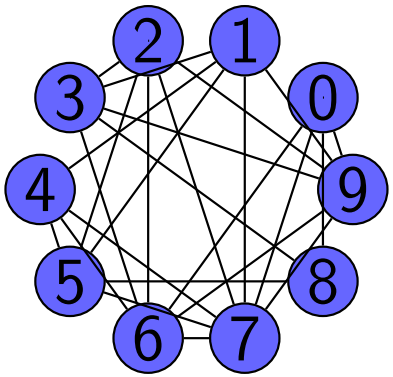


- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.



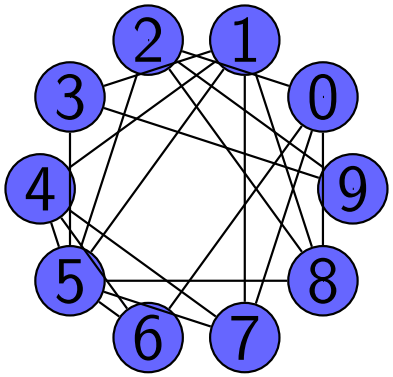
What is graph matching?

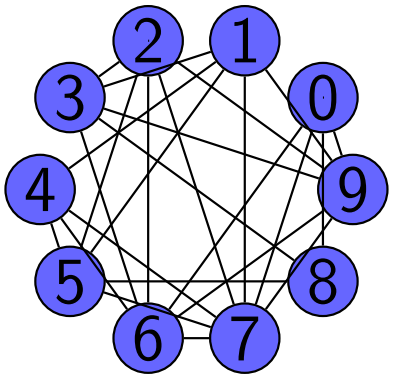




- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.
- Graph matching helps identifying the same object appearing in different networks.

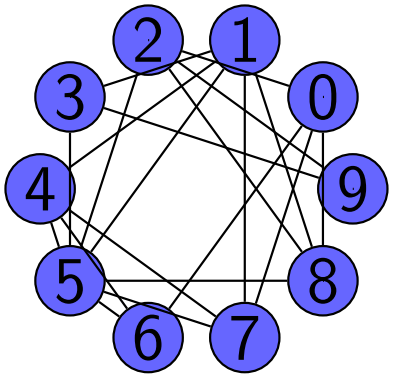
What is graph matching?

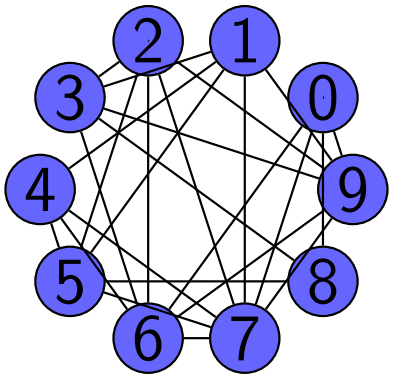




- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.
- Graph matching helps identifying the same object appearing in different networks.
- Various applications in real-world scenes.

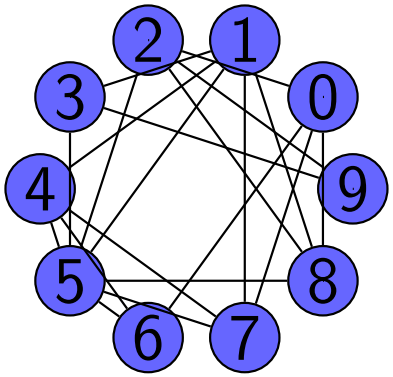
What is graph matching?

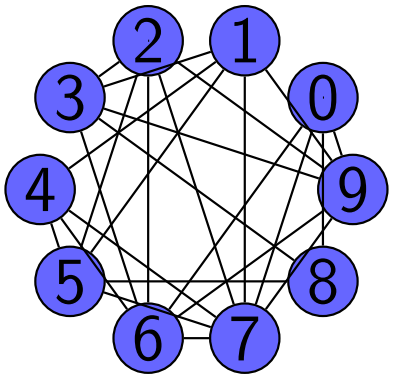




- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.
- Graph matching helps identifying the same object appearing in different networks.
- Various applications in real-world scenes.
  - Network de-anonymization.

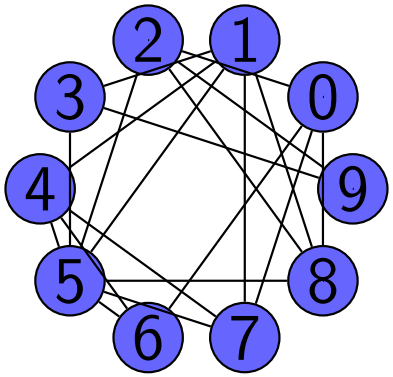
What is graph matching?

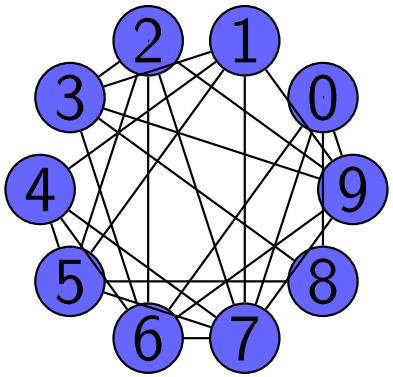




- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.
- Graph matching helps identifying the same object appearing in different networks.
- Various applications in real-world scenes.
  - Network de-anonymization.
  - Protein-protein interaction network.

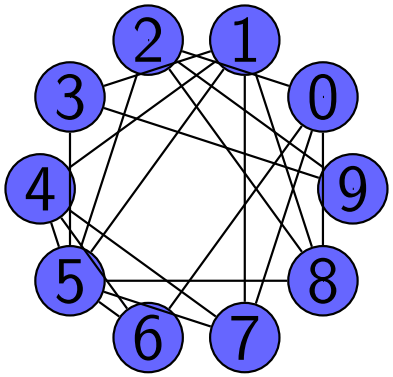
What is graph matching?

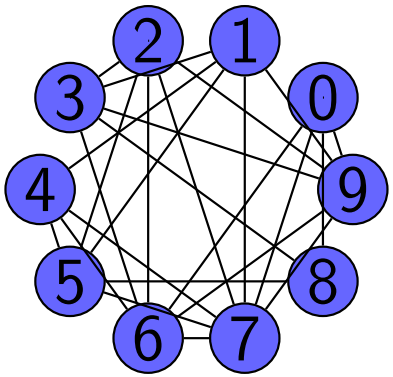




- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.
- Graph matching helps identifying the same object appearing in different networks.
- Various applications in real-world scenes.
  - Network de-anonymization.
  - Protein-protein interaction network.
  - Computer Vision.

What is graph matching?

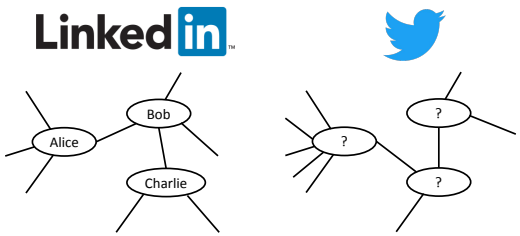




- Graphs: abstract models of objects and their connections.
- Graph matching: identifying the same node in different graphs by similar topological structures.
  - e.g. isomorphic trees, dense subgraphs, etc.
- Graph matching helps identifying the same object appearing in different networks.
- Various applications in real-world scenes.
  - Network de-anonymization.
  - Protein-protein interaction network.
  - Computer Vision.
  - Machine Translation, etc.

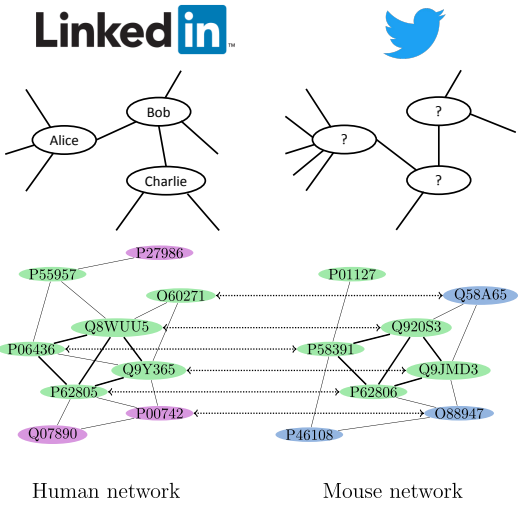
Examples of applications

Examples of applications



- Network de-anonymization: (1) successfully de-anonymize Netflix by matching it to IMDB in Narayanan-Shmatikov '08; (2) correctly identified 30.8% of node mappings between Twitter&Flickr in Narayanan-Shmatikov '09.

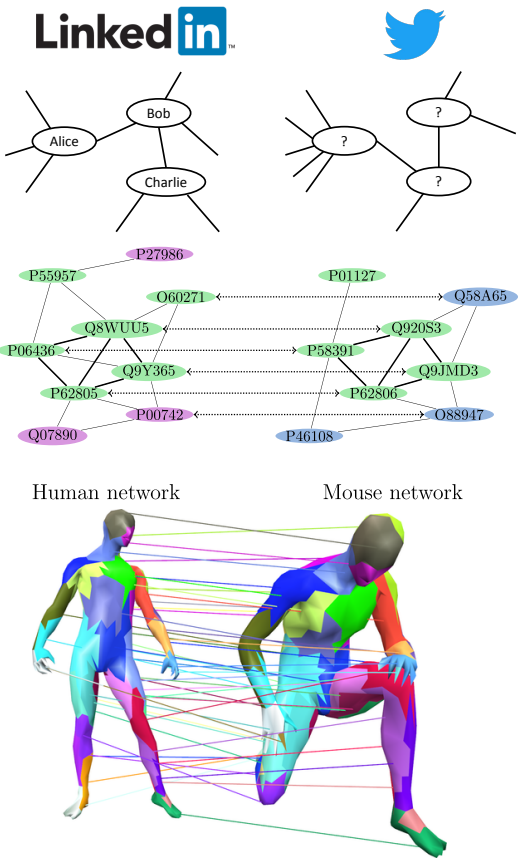
Examples of applications



Human network Mouse network

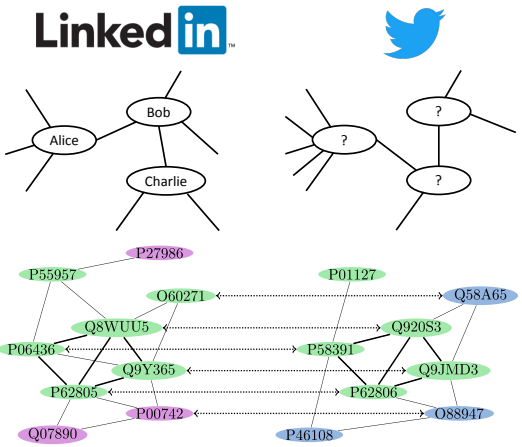
- Network de-anonymization: (1) successfully de-anonymize Netflix by matching it to IMDB in Narayanan-Shmatikov '08; (2) correctly identified 30.8% of node mappings between Twitter&Flickr in Narayanan-Shmatikov '09.
- Protein-protein interaction network: discover proteins with similar functions across different species based on interaction network topology in Kazemi-Hassani-Grossglauser-Modarres '16;

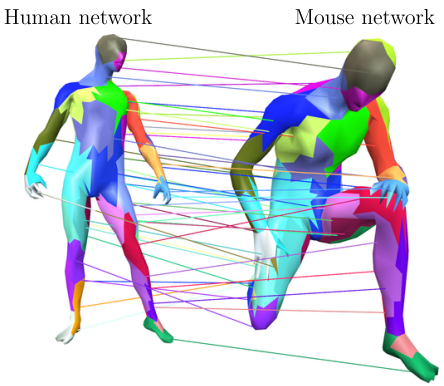
Examples of applications

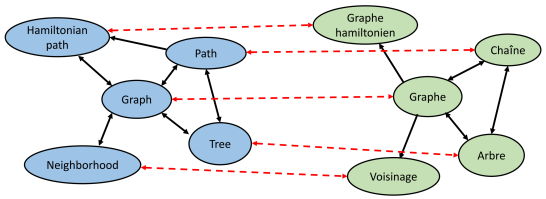


- Network de-anonymization: (1) successfully de-anonymize Netflix by matching it to IMDB in Narayanan-Shmatikov '08; (2) correctly identified 30.8% of node mappings between Twitter&Flickr in Narayanan-Shmatikov '09.
- Protein-protein interaction network: discover proteins with similar functions across different species based on interaction network topology in Kazemi-Hassani-Grossglauser-Modarres '16;
- Computer Vision: detect similar objects that undergo different deformations in Lähner et al '16.

Examples of applications

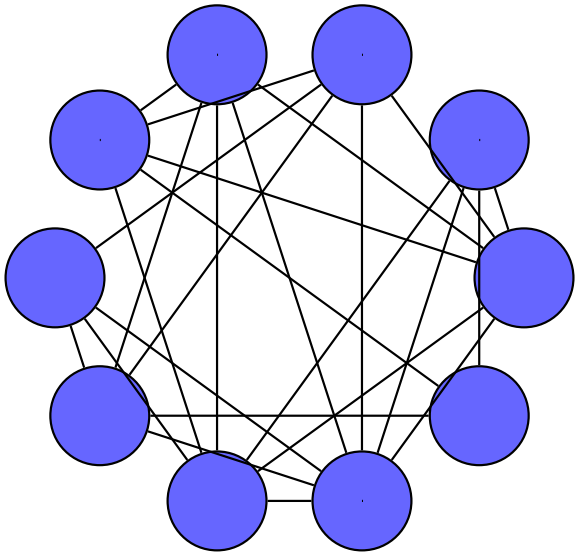


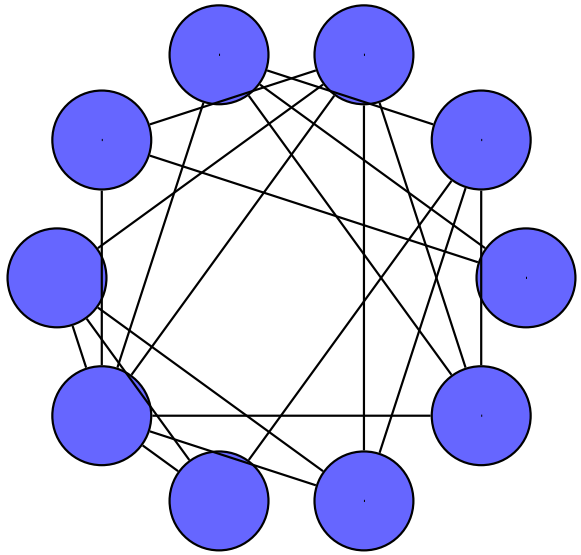




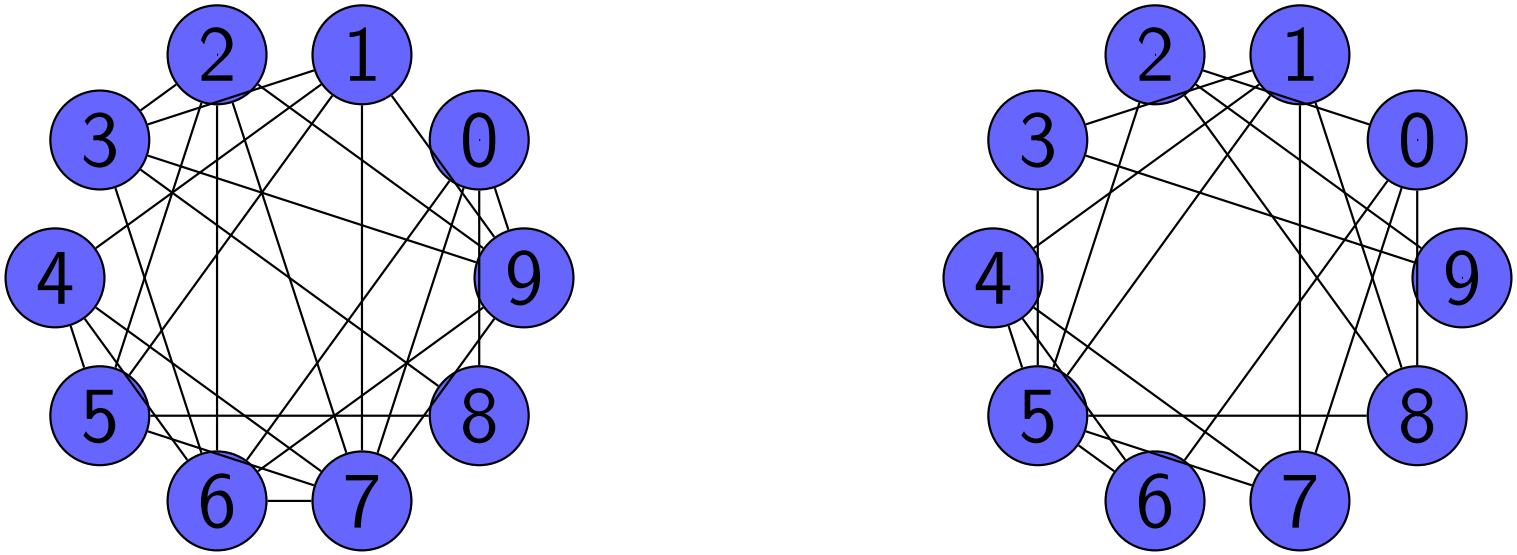
- Network de-anonymization: (1) successfully de-anonymize Netflix by matching it to IMDB in Narayanan-Shmatikov '08; (2) correctly identified 30.8% of node mappings between Twitter&Flickr in Narayanan-Shmatikov '09.
- Protein-protein interaction network: discover proteins with similar functions across different species based on interaction network topology in Kazemi-Hassani-Grossglauser-Modarres '16;
- Computer Vision: detect similar objects that undergo different deformations in Lähner et al '16.
- Machine Translation: automatically find correct corresponding wiki articles in different languages in Fishkind-Adali-Patsolic-Meng-Lyzinski-Priebe '12.

Mathematical formulation of graph matching



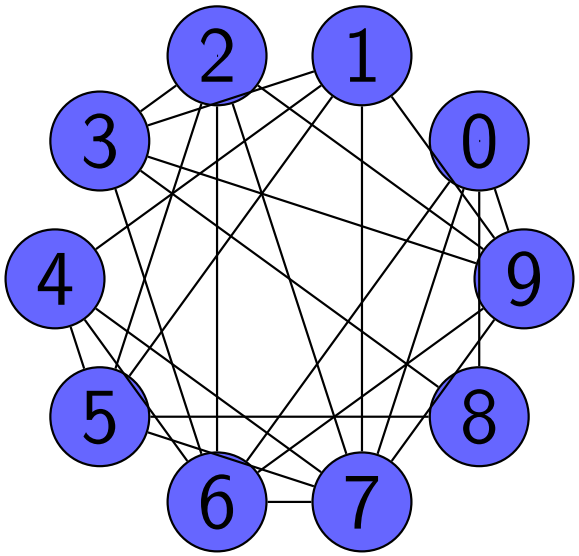


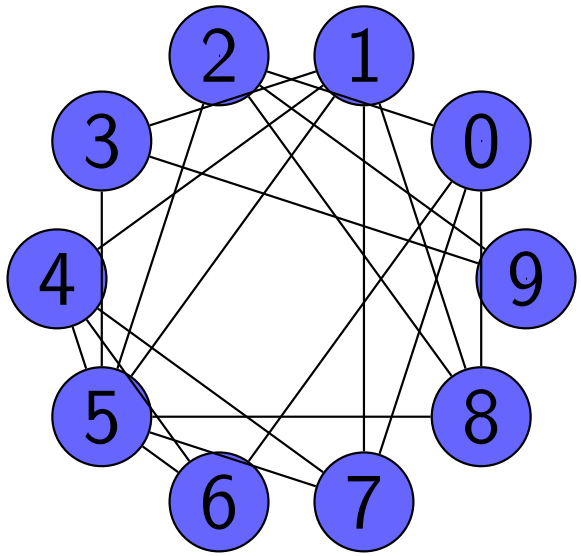
Mathematical formulation of graph matching



• Goal: find a bijection between two vertex sets that maximally align the edges (i.e. minimizes # of adjacency disagreements).

Mathematical formulation of graph matching

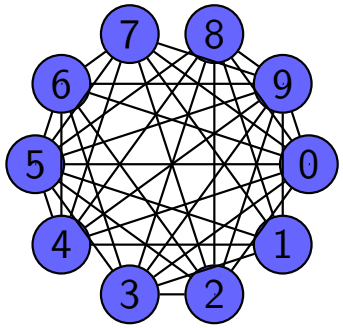




• Goal: find a bijection between two vertex sets that maximally align the edges (i.e. minimizes # of adjacency disagreements).

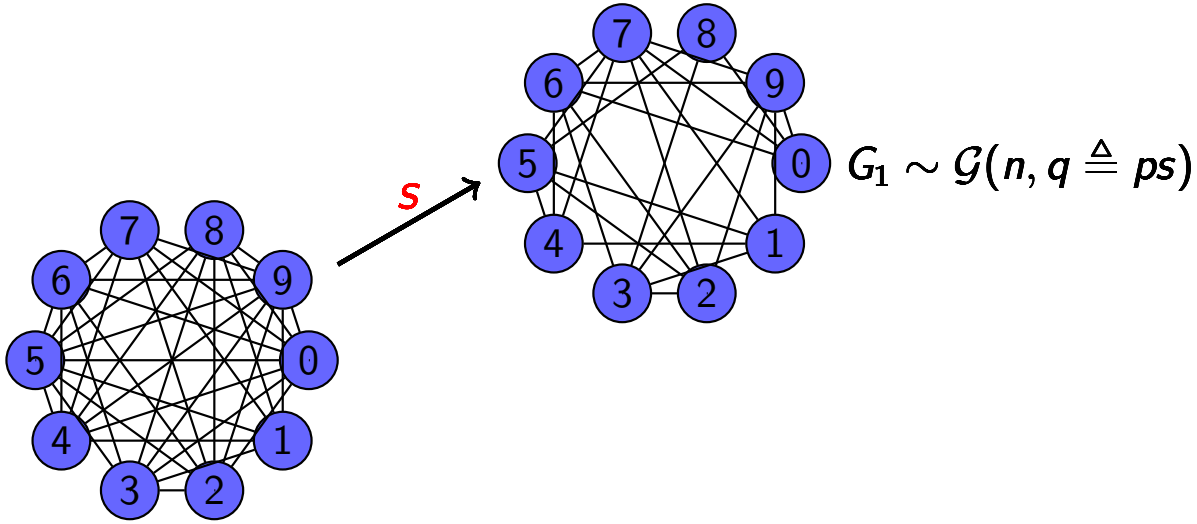
• Since graph matching is a hard optimization problem (NP-hard), we seek help from randomness.

An idealized model: Correlated Erdős-Rényi graphs model

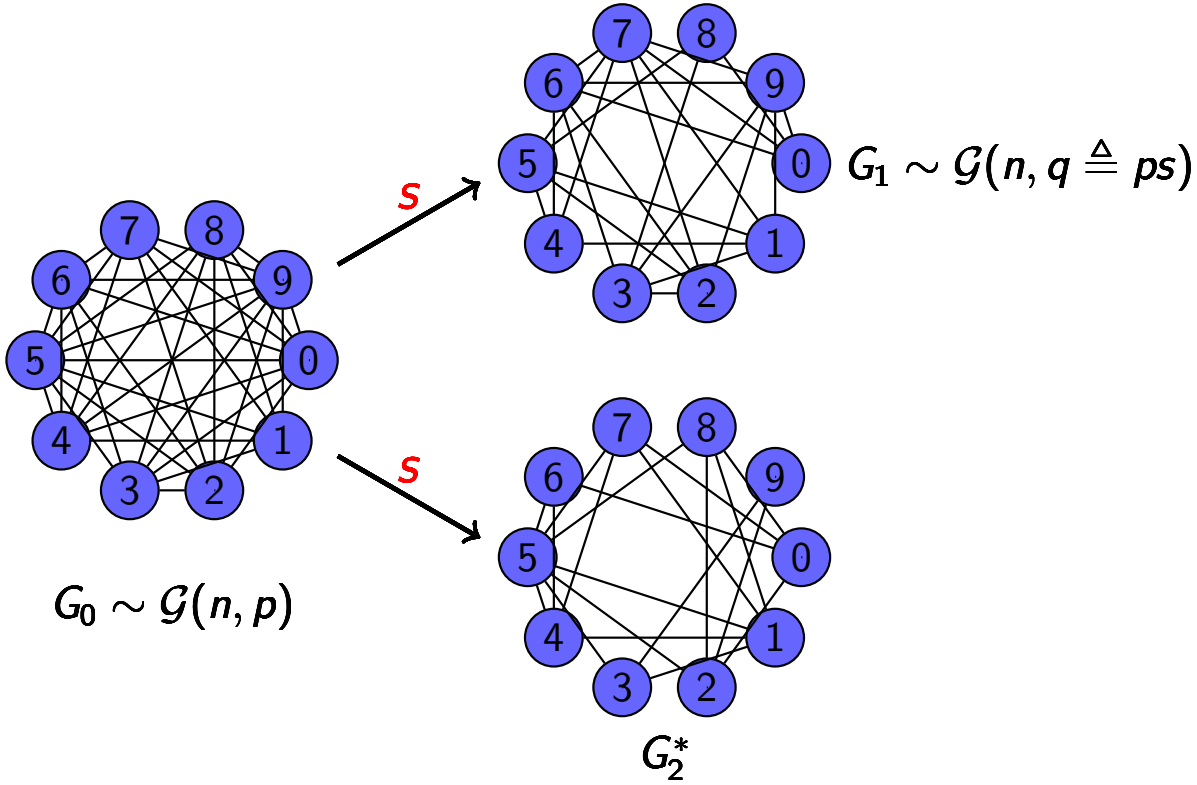


$G_0 \sim \mathcal{G}(n, p)$

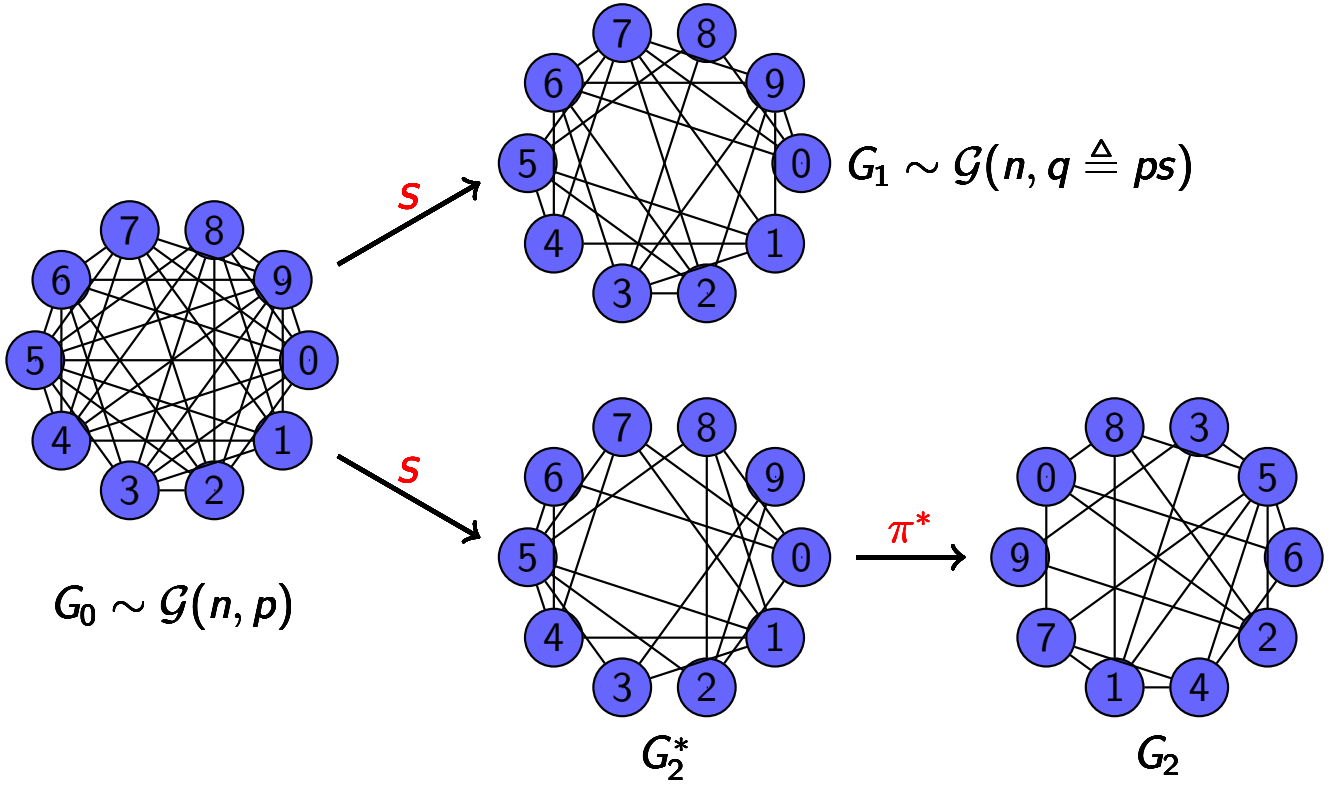
An idealized model: Correlated Erdős-Rényi graphs model



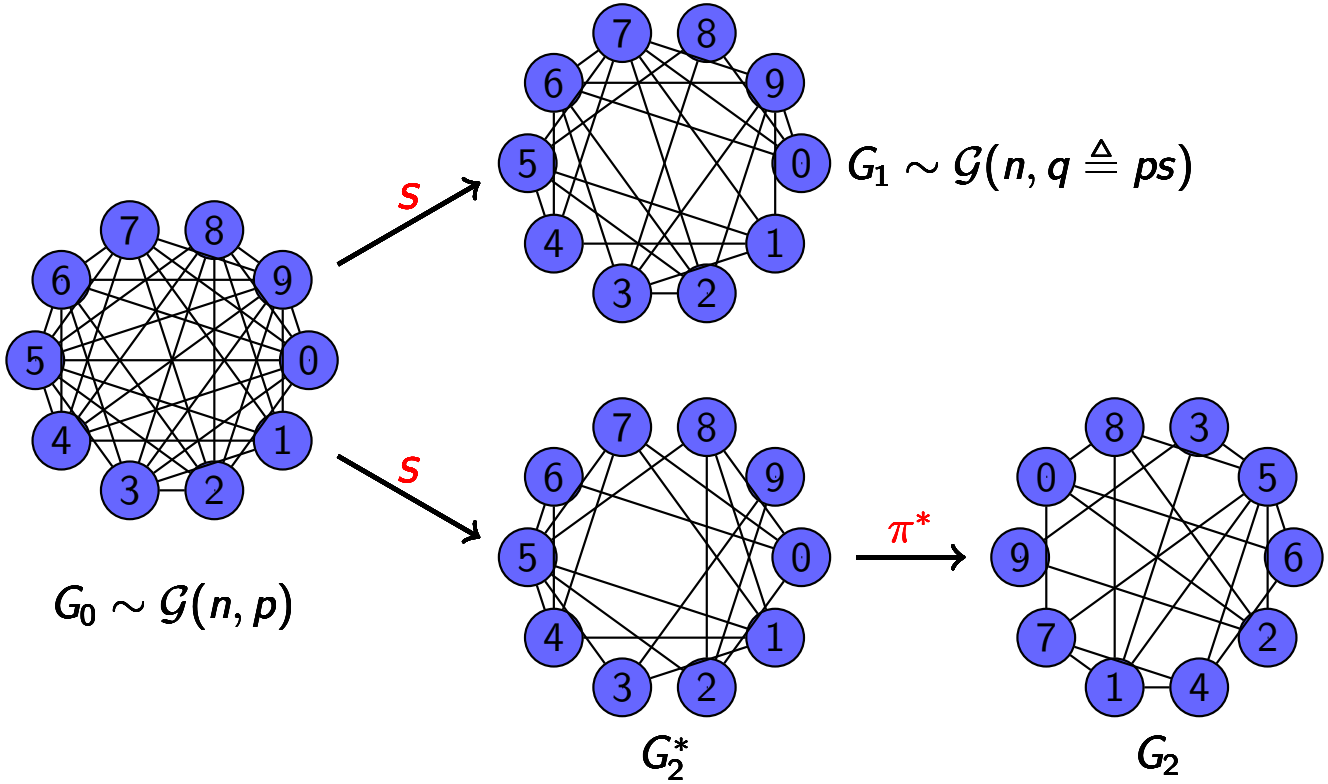
An idealized model: Correlated Erdős-Rényi graphs model



An idealized model: Correlated Erdős-Rényi graphs model

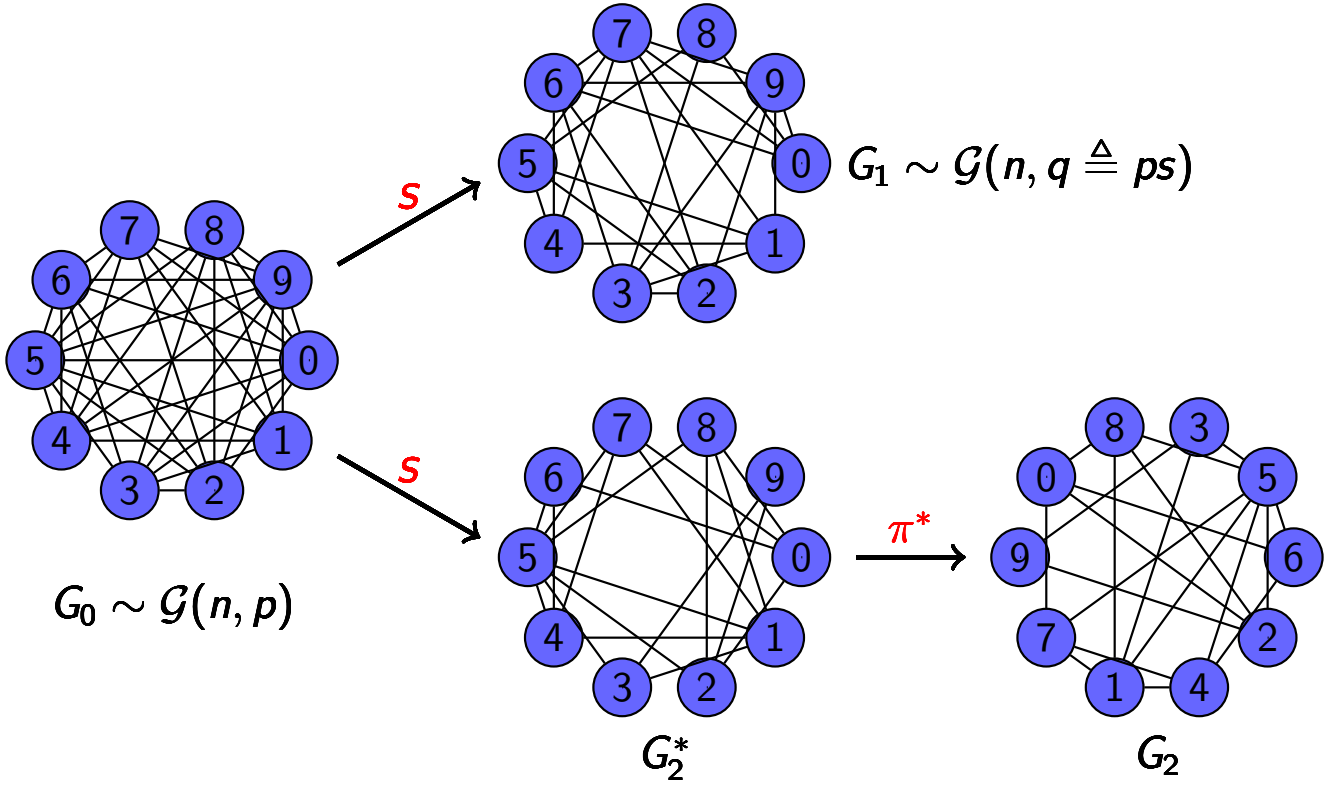


An idealized model: Correlated Erdős-Rényi graphs model



There is no structure in randomness: there is an edge between a pair of vertices with probability p independently.

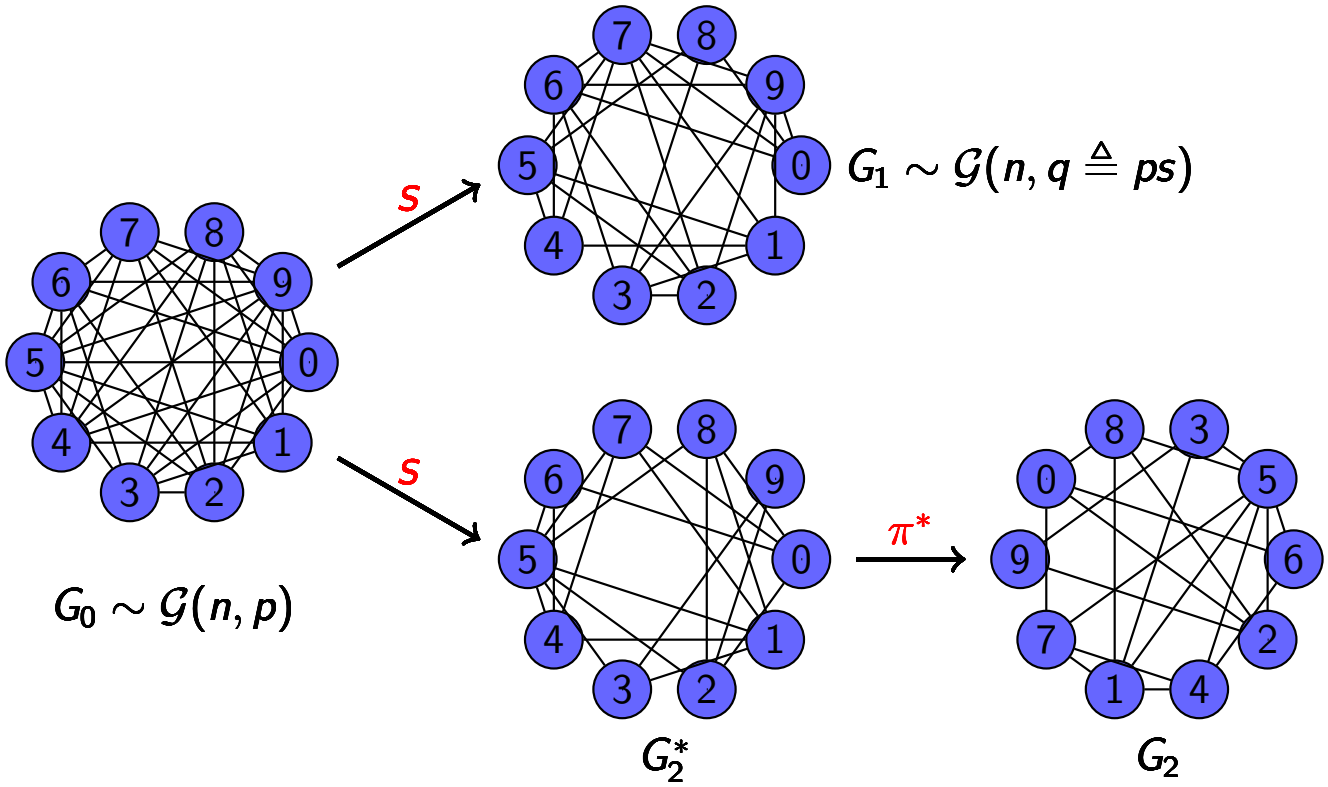
An idealized model: Correlated Erdős-Rényi graphs model



There is no structure in randomness: there is an edge between a pair of vertices with probability p independently.

Advantage: simple probabilistic model; suitable playground for developing mathematical theory.

An idealized model: Correlated Erdős-Rényi graphs model

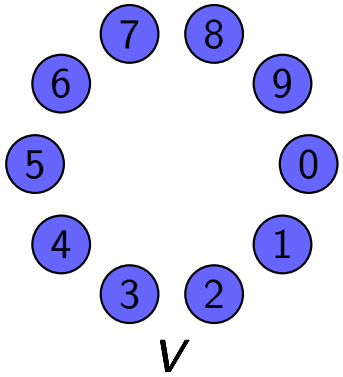


There is no structure in randomness: there is an edge between a pair of vertices with probability p independently.

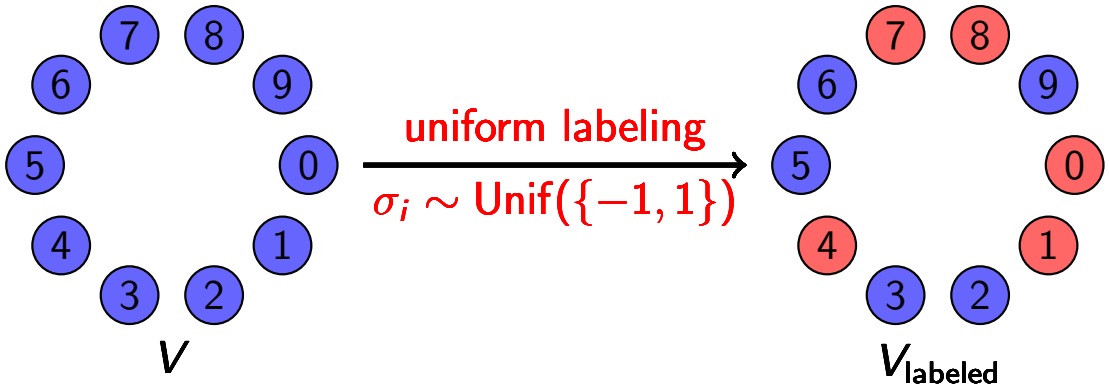
Advantage: simple probabilistic model; suitable playground for developing mathematical theory.

Disadvantage: almost all realistic networks are not Erdős-Rényi.

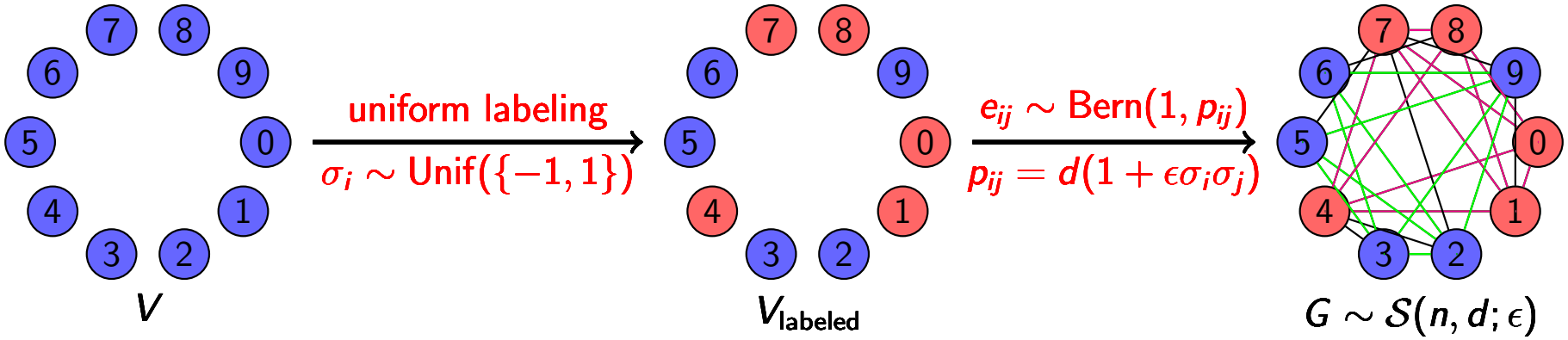
A more realistic variant: Stochastic block model



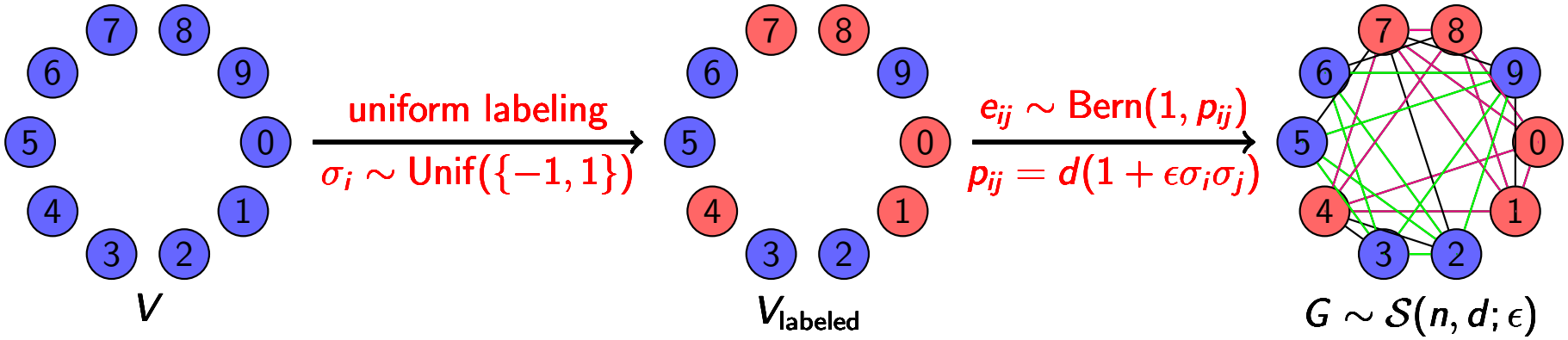
A more realistic variant: Stochastic block model



A more realistic variant: Stochastic block model

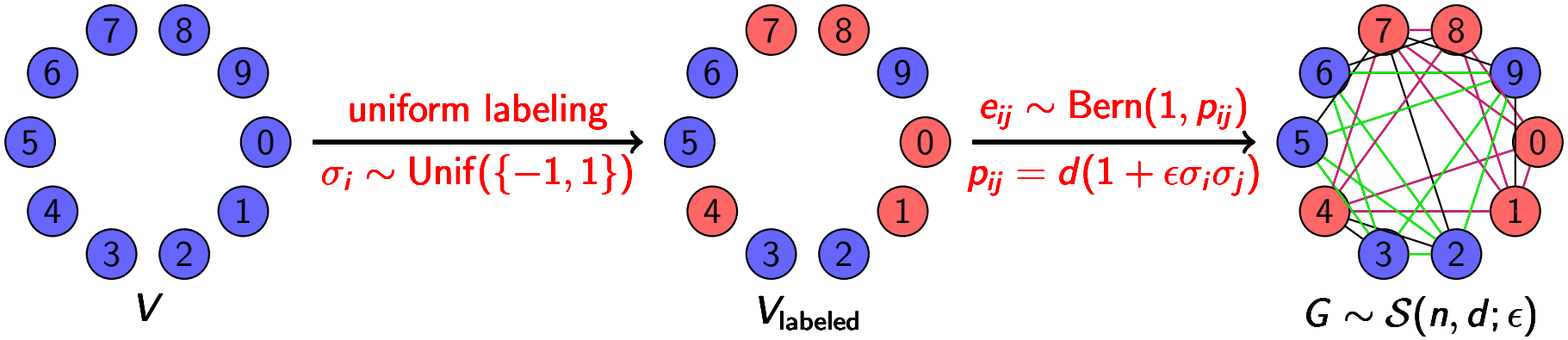


A more realistic variant: Stochastic block model



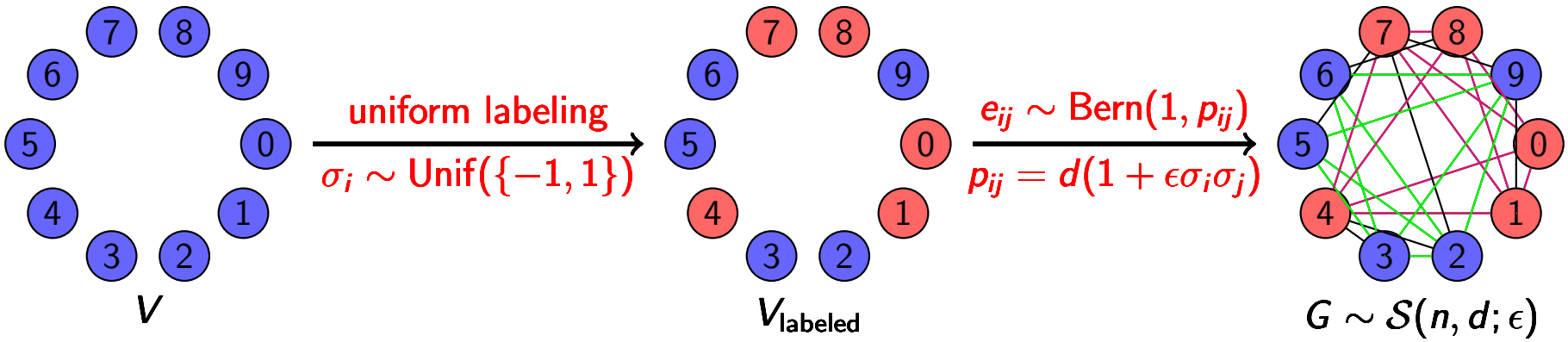
- community structure: an edge joins i and j with probability $p_{ij} = d(1 + \epsilon\sigma_i\sigma_j)$ depending on wether i and j belong to the same community or not.

A more realistic variant: Stochastic block model



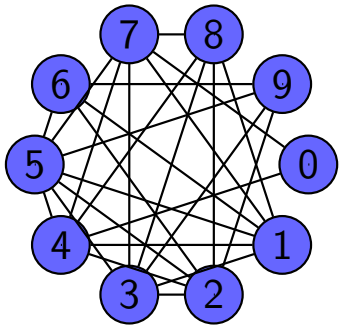
- community structure: an edge joins i and j with probability $p_{ij} = d(1 + \epsilon\sigma_i\sigma_j)$ depending on wether i and j belong to the same community or not.
- Introduced in Holland-Laskey-Leinhardt83 and has attracted significant attentions in physics, statistics, probability, and TCS. (See Decelle-Krzakala-Moore-Zdeborová'11, Mossel-Neeman-Sly'12,13a,13b,14, Abbé-Sandon 15a,15b...)

A more realistic variant: Stochastic block model



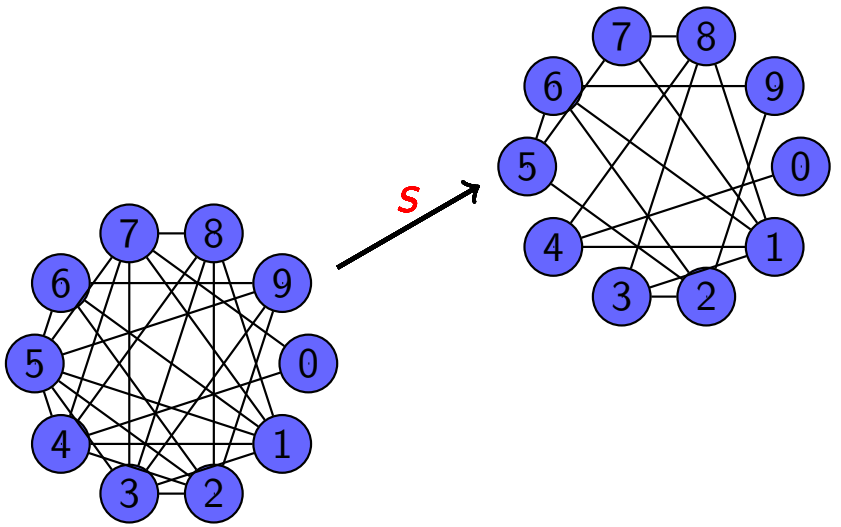
- community structure: an edge joins i and j with probability $p_{ij} = d(1 + \epsilon\sigma_i\sigma_j)$ depending on wether i and j belong to the same community or not.
- Introduced in Holland-Laskey-Leinhardt83 and has attracted significant attentions in physics, statistics, probability, and TCS. (See Decelle-Krzakala-Moore-Zdeborová'11, Mossel-Neeman-Sly'12,13a,13b,14, Abbé-Sandon 15a,15b...)
- Generalization to k communities (denoted by $\mathcal{S}(n, d; k, \epsilon)$):
  - $\sigma_i \sim \mathrm{Unif}(\{1, 2, \ldots, k\})$;
  - $p_{ij} = d(1 + \epsilon\omega(\sigma_i, \sigma_j))$ where $\omega(\sigma_i, \sigma_j) = -1 + k \cdot \mathbf{1}_{\{\sigma_i = \sigma_j\}}$.

Correlated stochastic block model



$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

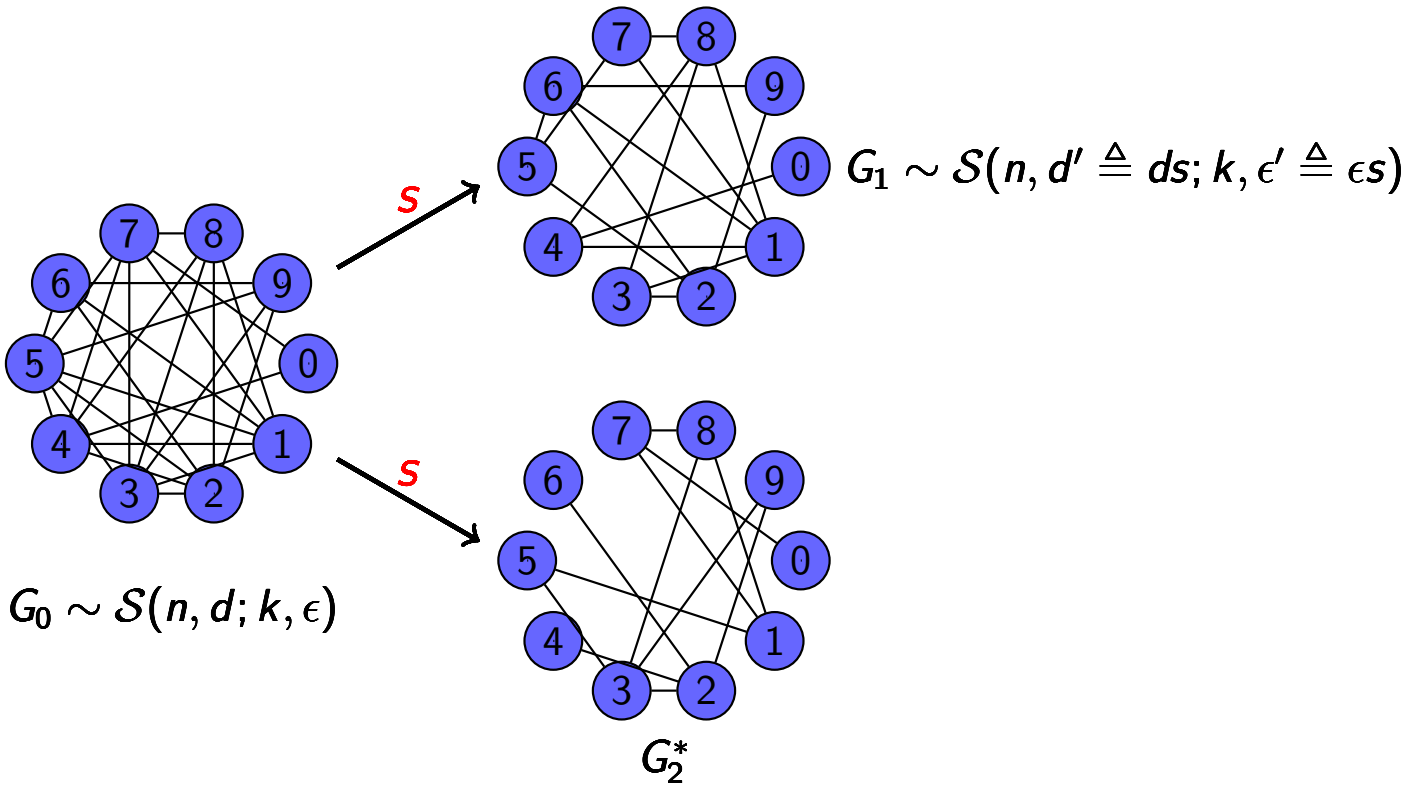
Correlated stochastic block model



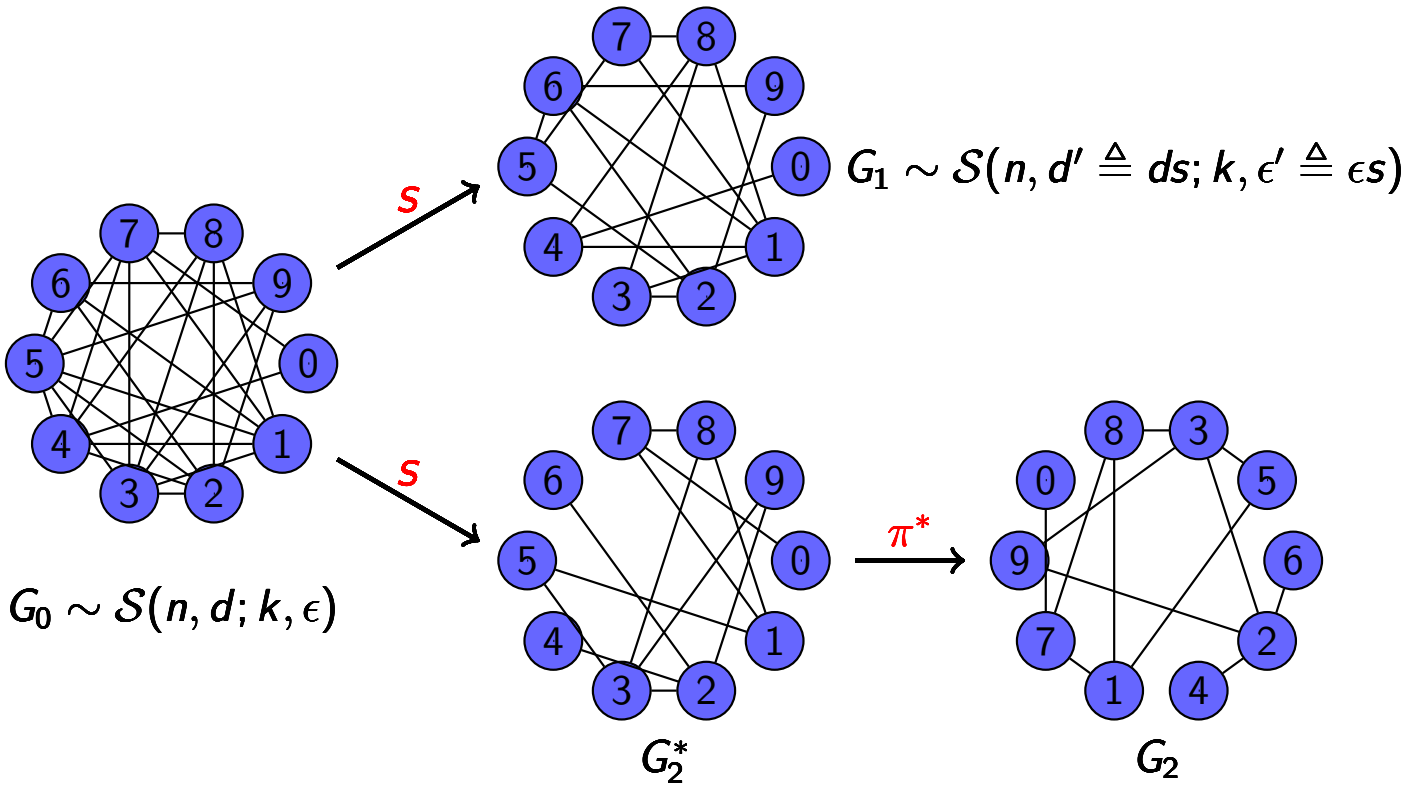
$G_1 \sim \mathcal{S}(n, d' \triangleq ds; k, \epsilon' \triangleq \epsilon s)$

$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

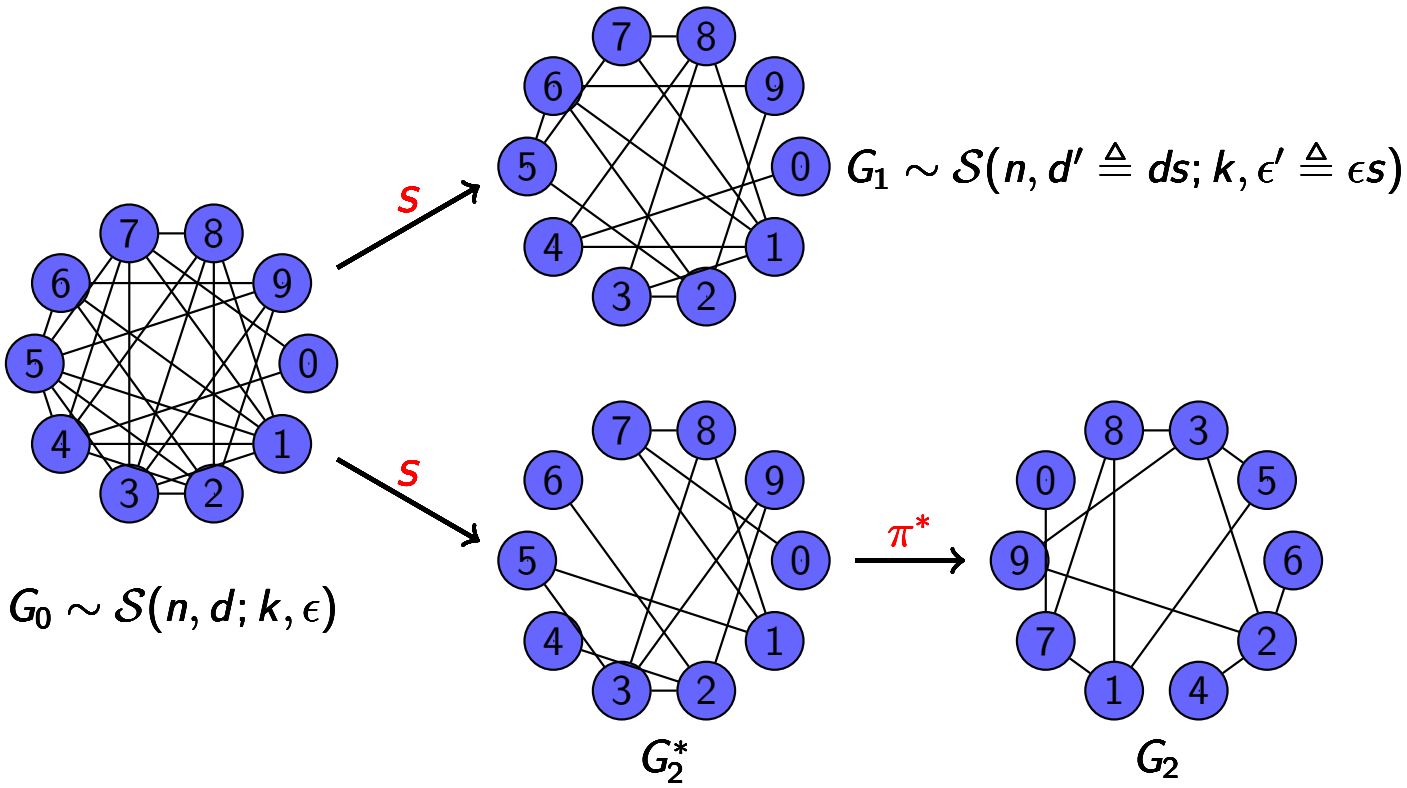
Correlated stochastic block model



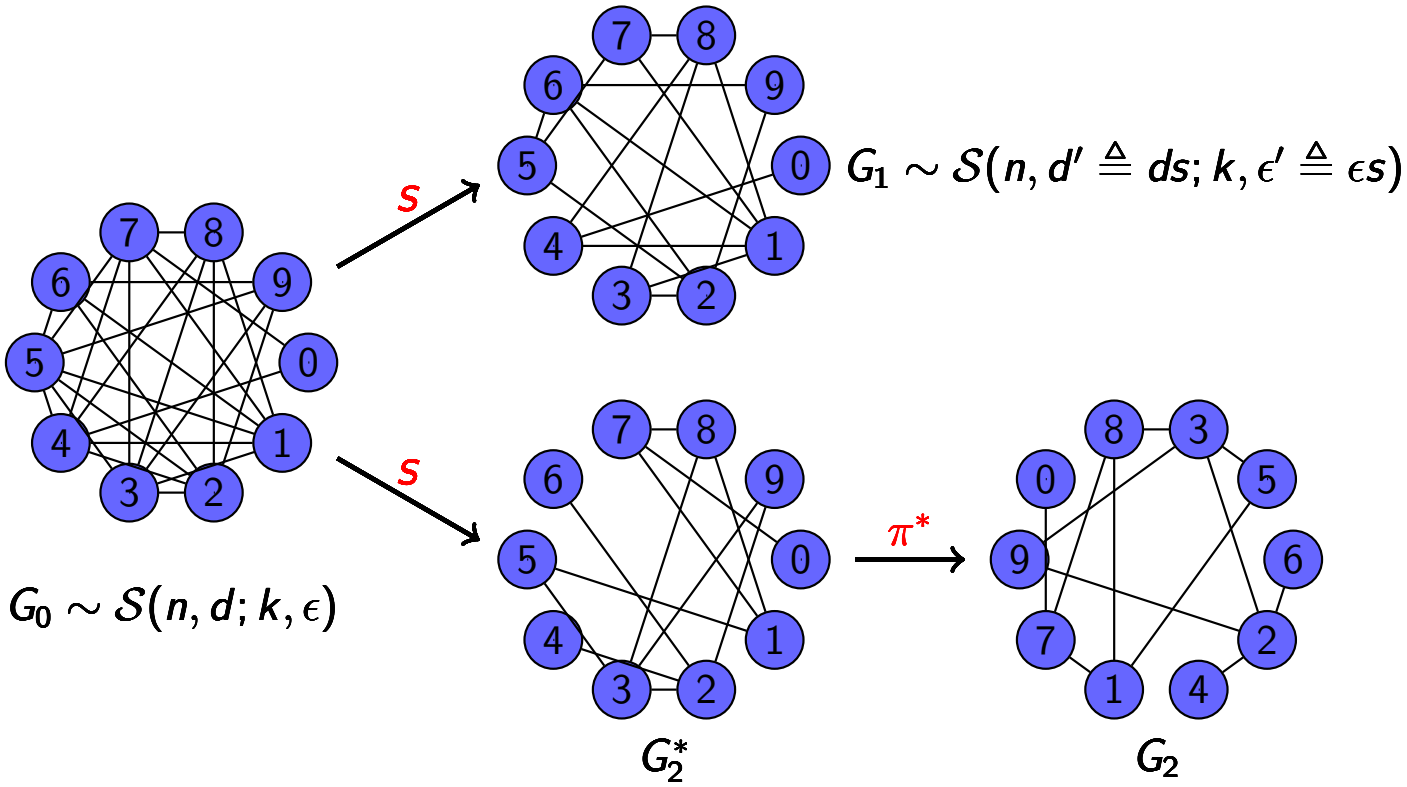
Correlated stochastic block model



Correlated stochastic block model



Correlated stochastic block model



- Generated by independently subsampling $G_0 \sim \mathcal{S}(n, d; k, \epsilon)$ with probability s.
- Denoted by $\mathcal{S}(n, d; k, \epsilon; s)$.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Previous results on matching correlated Erdős-Rényi graphs:

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Previous results on matching correlated Erdős-Rényi graphs:

- Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).
  - Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.
    - Detection: $|\mathcal{E}|$ is large $\Rightarrow$ correlation.
    - Matching: estimate π^* by $\hat{\pi}$.

Information threshold: what is the teststone?

Three thresholds: detection, exact recovery, partial recovery.

- Detection: test correlation against independence.
- Exact recovery: correctly match all vertices.
- Partial recovery: correctly match a positive fraction of vertices.

Previous results on matching correlated Erdős-Rényi graphs:

- Wu-Xu-Yu'20, 21: progress based on maximal common graph (see Ganassali-Massoulié-Lelarge for $p \approx 1/n$).
  - Methods: let $\hat{\pi}$ be the bijection that maximizes the number of common edges $\mathcal{E}$.
    - Detection: $|\mathcal{E}|$ is large $\Rightarrow$ correlation.
    - Matching: estimate π^* by $\hat{\pi}$.
- D.-Du'23, Du25+: Exact detection and partial recovery threshold in the non-dense regime, via densest subgraph.

Matching algorithms for correlated graphs

Matching algorithms for correlated graphs

Results prior to 2021 (requiring correlation tending to 1): Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19, D.-Ma-Wu-Xu'21, Fan-Mao-Wu-Xu'2022.

Matching algorithms for correlated graphs

Results prior to 2021 (requiring correlation tending to 1): Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19, D.-Ma-Wu-Xu'21, Fan-Mao-Wu-Xu'2022.

Recent progress on matching algorithms:

Matching algorithms for correlated graphs

Results prior to 2021 (requiring correlation tending to 1): Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19, D.-Ma-Wu-Xu'21, Fan-Mao-Wu-Xu'2022.

Recent progress on matching algorithms:

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).

Matching algorithms for correlated graphs

Results prior to 2021 (requiring correlation tending to 1): Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19, D.-Ma-Wu-Xu'21, Fan-Mao-Wu-Xu'2022.

Recent progress on matching algorithms:

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.

Matching algorithms for correlated graphs

Results prior to 2021 (requiring correlation tending to 1): Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19, D.-Ma-Wu-Xu'21, Fan-Mao-Wu-Xu'2022.

Recent progress on matching algorithms:

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).

Matching algorithms for correlated graphs

Results prior to 2021 (requiring correlation tending to 1): Dai-Cullina-Kiyavash-Grossglauser'18, Barak-Chou-Lei-Schramm-Sheng'19, D.-Ma-Wu-Xu'21, Fan-Mao-Wu-Xu'2022.

Recent progress on matching algorithms:

- Mao-Rudelson-Tikhomirov'21+: poly-time algorithm based some partition trees, when correlation $\geq \mathrm{const}$ (close to 1).
- Ganassali-Massoulié-Lelarge'20+,22+: poly-time partial matching algorithm for sparse graphs based on message passing, when correlation $> \sqrt{\text{Otter's constant}} \approx \sqrt{0.3383}$.
- Mao-Wu-Xu-Yu'22+: poly-time algorithm when correlation $> \sqrt{\text{Otter's constant}}$, based on a carefully curated family of rooted trees called chandeliers (substantially improving MRT21+, and covers much wider parameter regime).
- D.-Li'22+, 23+: poly-time iterative algorithm when correlation is non-vanishing and average degree grows polynomially.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (much more difficult on random instance than for worst-case instance);

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem (D'–Du–Li'23+: low-degree polynomial complexity);

Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem (D'–Du–Li'23+: low-degree polynomial complexity);
  - exhibit similar structural properties as in other hard problems.

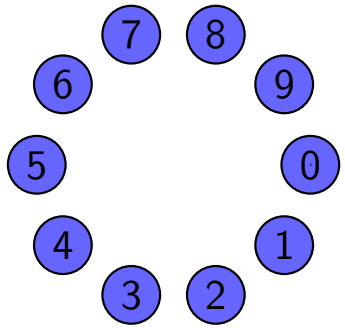
Complexity theory: how to certify computational hardness?

- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem (D'–Du–Li'23+: low-degree polynomial complexity);
  - exhibit similar structural properties as in other hard problems.
- Application in data privacy: how can we perform a minimal change on the Linkedin and Twitter network, so that it would be computationally hard to recover the matching from the this perturbed observation?

Complexity theory: how to certify computational hardness?

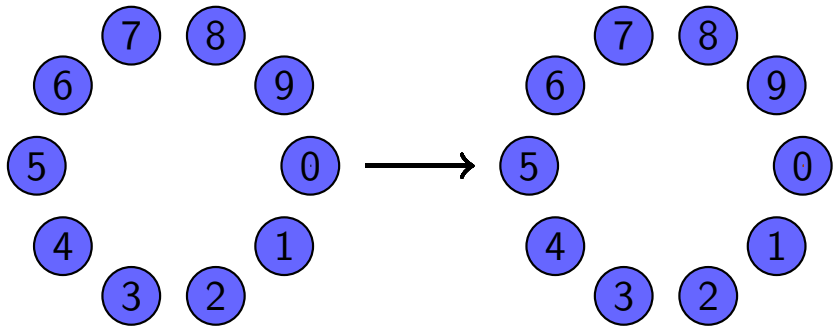
- Traditionally, complexity theory studies hardness of computational problems for worst-case instance.
- Usually certify hardness by reduction: if you could solve this problem, then you can solve some well-known hard problems.
- For problems with random instance, we care about the hardness for a typical instance. **Evidences** of hardness include
  - show as hard as well-known hard problems (much more difficult on random instance than for worst-case instance);
  - show that a wide class of algorithms fail to solve the problem (D'–Du–Li'23+: low-degree polynomial complexity);
  - exhibit similar structural properties as in other hard problems.
- Application in data privacy: how can we perform a minimal change on the Linkedin and Twitter network, so that it would be computationally hard to recover the matching from the this perturbed observation?
- **Information-computation gap**: a **major** challenge in many random combinatorial optimization and constraint satisfaction problems!

Finding communities in stochastic block model



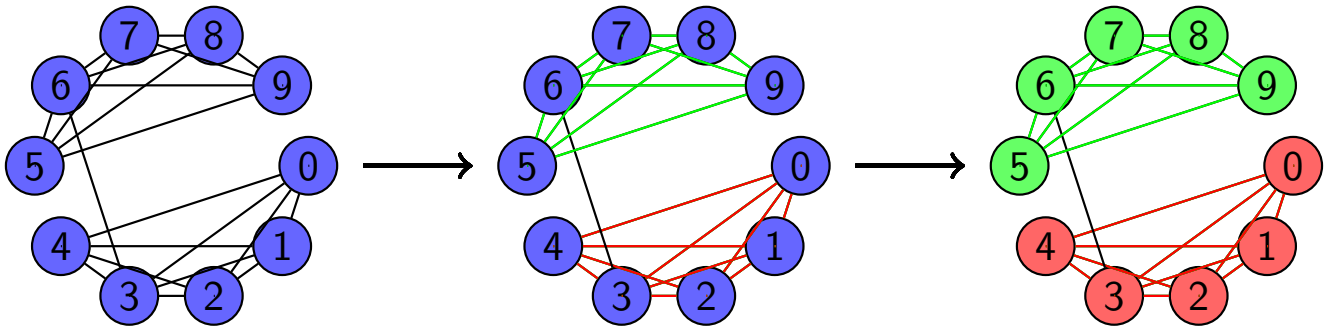
$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

Finding communities in stochastic block model



$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

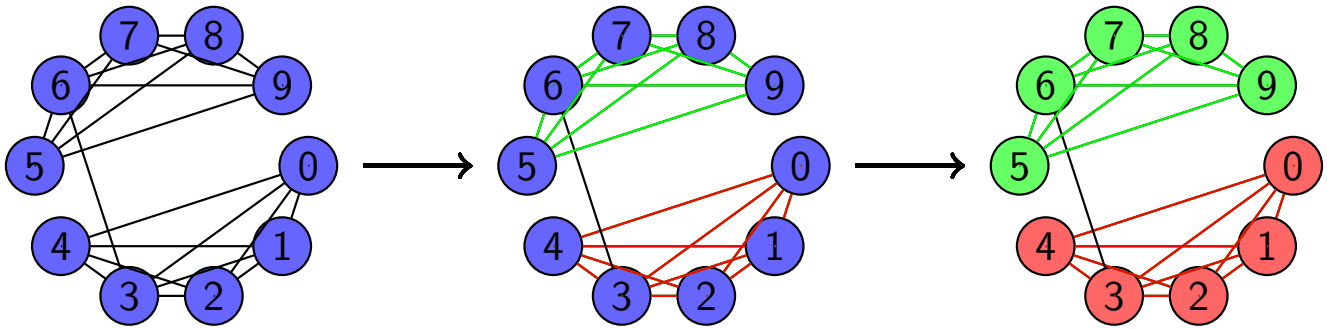
Finding communities in stochastic block model



$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

G_0 with estimated communities

Finding communities in stochastic block model

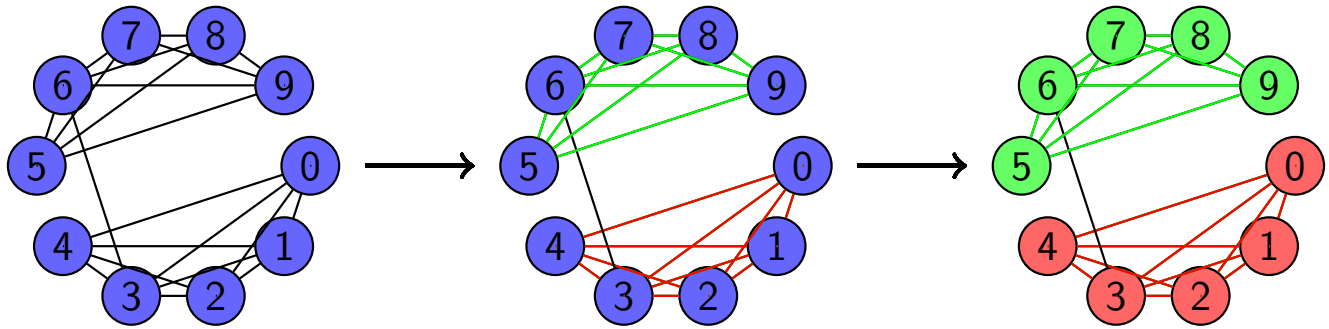


$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

G_0 with estimated communities

- Community recovery/detection: estimate/detect the existence of community structure.

Finding communities in stochastic block model



$G_0 \sim \mathcal{S}(n, d; k, \epsilon)$

G_0 with estimated communities

- Community recovery/detection: estimate/detect the existence of community structure.
- Decelle-Krzakala-Moore-Zdeborová'11:
  - Computational transition (under low-degree conjecture) around Kesten-Stigum threshold, i.e., $\lambda\epsilon^2 = 1$ Hopkins-Steurer'16.
  - Presumably information-computation gap if and only if $k \geq 5$ (Mossel-Neeman-Sly'16, Massoulié'16 for $k = 2$ and Mossel-Sly-Sohn'23,24 for general k).

Prior works in logarithmic regime

Prior works in logarithmic regime

- Rácz-Sridhar'22, Gaudio-Rácz-Sridhar'22, Rácz-Zhang'24: in the logarithmic regime, exact community recovery in multiple correlated block models is (informationally) possible even when:
  - (1) exact community recovery in each single block model is (informationally) impossible;
  - (2) exact graph matching between different graphs is (informationally) impossible.

Prior works in logarithmic regime

- Rácz-Sridhar'22, Gaudio-Rácz-Sridhar'22, Rácz-Zhang'24: in the logarithmic regime, exact community recovery in multiple correlated block models is (informationally) possible even when:
  - (1) exact community recovery in each single block model is (informationally) impossible;
  - (2) exact graph matching between different graphs is (informationally) impossible.

Prior works in logarithmic regime

- Rácz-Sridhar'22, Gaudio-Rácz-Sridhar'22, Rácz-Zhang'24: in the logarithmic regime, exact community recovery in multiple correlated block models is (informationally) possible even when:
  - (1) exact community recovery in each single block model is (informationally) impossible;
  - (2) exact graph matching between different graphs is (informationally) impossible.
- Yang-Shin-Chung'23, Chai-Rácz'24: generalize some graph matching algorithms developed in Erdős-Rényi models to block models.

Prior works in logarithmic regime

- Rácz-Sridhar'22, Gaudio-Rácz-Sridhar'22, Rácz-Zhang'24: in the logarithmic regime, exact community recovery in multiple correlated block models is (informationally) possible even when:
  - (1) exact community recovery in each single block model is (informationally) impossible;
  - (2) exact graph matching between different graphs is (informationally) impossible.
- Yang-Shin-Chung'23, Chai-Rácz'24: generalize some graph matching algorithms developed in Erdős-Rényi models to block models.

Two natural questions:

Prior works in logarithmic regime

- Rácz-Sridhar'22, Gaudio-Rácz-Sridhar'22, Rácz-Zhang'24: in the logarithmic regime, exact community recovery in multiple correlated block models is (informationally) possible even when:
  - (1) exact community recovery in each single block model is (informationally) impossible;
  - (2) exact graph matching between different graphs is (informationally) impossible.
- Yang-Shin-Chung'23, Chai-Rácz'24: generalize some graph matching algorithms developed in Erdős-Rényi models to block models.

Two natural questions:

- What about (arguably more interesting) constant degree regime?

Prior works in logarithmic regime

- Rácz-Sridhar'22, Gaudio-Rácz-Sridhar'22, Rácz-Zhang'24: in the logarithmic regime, exact community recovery in multiple correlated block models is (informationally) possible even when:
  - (1) exact community recovery in each single block model is (informationally) impossible;
  - (2) exact graph matching between different graphs is (informationally) impossible.
- Yang-Shin-Chung'23, Chai-Rácz'24: generalize some graph matching algorithms developed in Erdős-Rényi models to block models.

Two natural questions:

- What about (arguably more interesting) constant degree regime?
- Is there any case that we can break the algorithmic barrier in Erdős-Rényi model?

Computational hardness in constant degree regime

Computational hardness in constant degree regime

Consider two children graphs G_1, G_2 subsampled from a common parent block model with edge-density λ, community number k and subsampling probability s. In the constant degree regime $\lambda = O(1)$:

Computational hardness in constant degree regime

Consider two children graphs G_1, G_2 subsampled from a common parent block model with edge-density λ, community number k and subsampling probability s. In the constant degree regime $\lambda = O(1)$:
Chen-D.-Gong-Li'24+: Assuming the low-degree conjecture, the (algorithmic) detection threshold between this model and two independent Erdős-Rényi model is simply determined by

Computational hardness in constant degree regime

Consider two children graphs G_1, G_2 subsampled from a common parent block model with edge-density λ, community number k and subsampling probability s. In the constant degree regime $\lambda = O(1)$:
Chen-D.-Gong-Li'24+: Assuming the low-degree conjecture, the (algorithmic) detection threshold between this model and two independent Erdős-Rényi model is simply determined by

- (1) community detection threshold in a single block model;

Computational hardness in constant degree regime

Consider two children graphs G_1, G_2 subsampled from a common parent block model with edge-density λ, community number k and subsampling probability s. In the constant degree regime $\lambda = O(1)$:
Chen-D.-Gong-Li'24+: Assuming the low-degree conjecture, the (algorithmic) detection threshold between this model and two independent Erdős-Rényi model is simply determined by

- (1) community detection threshold in a single block model;
- (2) correlation detection threshold in a pair of Erdős-Rényi graphs.

Computational hardness in constant degree regime

Consider two children graphs G_1, G_2 subsampled from a common parent block model with edge-density λ, community number k and subsampling probability s. In the constant degree regime $\lambda = O(1)$:
Chen-D.-Gong-Li'24+: Assuming the low-degree conjecture, the (algorithmic) detection threshold between this model and two independent Erdős-Rényi model is simply determined by

- (1) community detection threshold in a single block model;
- (2) correlation detection threshold in a pair of Erdős-Rényi graphs.

In conclusion, no interplay between community detection and network correlation detection.

Computational hardness in constant degree regime

Consider two children graphs G_1, G_2 subsampled from a common parent block model with edge-density λ, community number k and subsampling probability s. In the constant degree regime $\lambda = O(1)$:
Chen-D.-Gong-Li'24+: Assuming the low-degree conjecture, the (algorithmic) detection threshold between this model and two independent Erdős-Rényi model is simply determined by

- (1) community detection threshold in a single block model;
- (2) correlation detection threshold in a pair of Erdős-Rényi graphs.

In conclusion, no interplay between community detection and network correlation detection.
Using a reduction technique in Li'25+, indicates in the subcritical regime (i.e., when community signal cannot be extracted from a single graph) the detection threshold between correlated/independent block models is still Otter's threshold.

Break Otter's barrier with strong community signal

Break Otter's barrier with strong community signal

Chen-D.-Gong-Li'25+: For two symmetric communities, detect correlation efficiently breaking Otter's threshold if the community signal is "sufficiently large" (i.e., $\lambda\epsilon^2 \gg 1$ and thus possible to recover 99% of community labels).

Break Otter's barrier with strong community signal

Chen-D.-Gong-Li'25+: For two symmetric communities, detect correlation efficiently breaking Otter's threshold if the community signal is "sufficiently large" (i.e., $\lambda\epsilon^2 \gg 1$ and thus possible to recover 99% of community labels).

- First algorithm breaking Otter's threshold in sparse correlated graphs; reveals a new phase transition phenomenon.

Break Otter's barrier with strong community signal

Chen-D.-Gong-Li'25+: For two symmetric communities, detect correlation efficiently breaking Otter's threshold if the community signal is "sufficiently large" (i.e., $\lambda\epsilon^2 \gg 1$ and thus possible to recover 99% of community labels).

- First algorithm breaking Otter's threshold in sparse correlated graphs; reveals a new phase transition phenomenon.
- Expected to extend to a partial matching algorithm and to all supercritical block models with general number of communities.

Break Otter's barrier with strong community signal

Chen-D.-Gong-Li'25+: For two symmetric communities, detect correlation efficiently breaking Otter's threshold if the community signal is "sufficiently large" (i.e., $\lambda\epsilon^2 \gg 1$ and thus possible to recover 99% of community labels).

- First algorithm breaking Otter's threshold in sparse correlated graphs; reveals a new phase transition phenomenon.
- Expected to extend to a partial matching algorithm and to all supercritical block models with general number of communities.

An ongoing challenge: extending the above result to as long as $\lambda\epsilon^2 > 1$.

Perspectives and future directions

Perspectives and future directions

- A step forward of meeting theory and applications:

Perspectives and future directions

- A step forward of meeting theory and applications:
  - Currently, most extensively studied models are idealized.

Perspectives and future directions

- A step forward of meeting theory and applications:
  - Currently, most extensively studied models are idealized.
  - Many algorithms and their analysis are based on unrealistic model assumptions, e.g., local tree structure for social network model.

Perspectives and future directions

- A step forward of meeting theory and applications:
  - Currently, most extensively studied models are idealized.
  - Many algorithms and their analysis are based on unrealistic model assumptions, e.g., local tree structure for social network model.
  - Major challenge 1: propose models with general applicability where theorists can say something meaningful.

Perspectives and future directions

- A step forward of meeting theory and applications:
  - Currently, most extensively studied models are idealized.
  - Many algorithms and their analysis are based on unrealistic model assumptions, e.g., local tree structure for social network model.
  - Major challenge 1: propose models with general applicability where theorists can say something meaningful.
  - Major challenge 2: propose new perspectives and formulations on graph matching from real-world applications (e.g. stability of matching algorithms under perturbations; editing graphs to make matching impossible, etc.)

Perspectives and future directions

- A step forward of meeting theory and applications:
  - Currently, most extensively studied models are idealized.
  - Many algorithms and their analysis are based on unrealistic model assumptions, e.g., local tree structure for social network model.
  - Major challenge 1: propose models with general applicability where theorists can say something meaningful.
  - Major challenge 2: propose new perspectives and formulations on graph matching from real-world applications (e.g. stability of matching algorithms under perturbations; editing graphs to make matching impossible, etc.)
- Bridging what is wanted with what is possible.

Reference: all mentioned works available on arXiv.