Networks & Hierarchies

Daniel Larremore
Assistant Professor
Dept. of Computer Science
& BioFrontiers Institute

June 25-26, 2018
SFI Complex Systems Summer School

University of Colorado **Boulder**

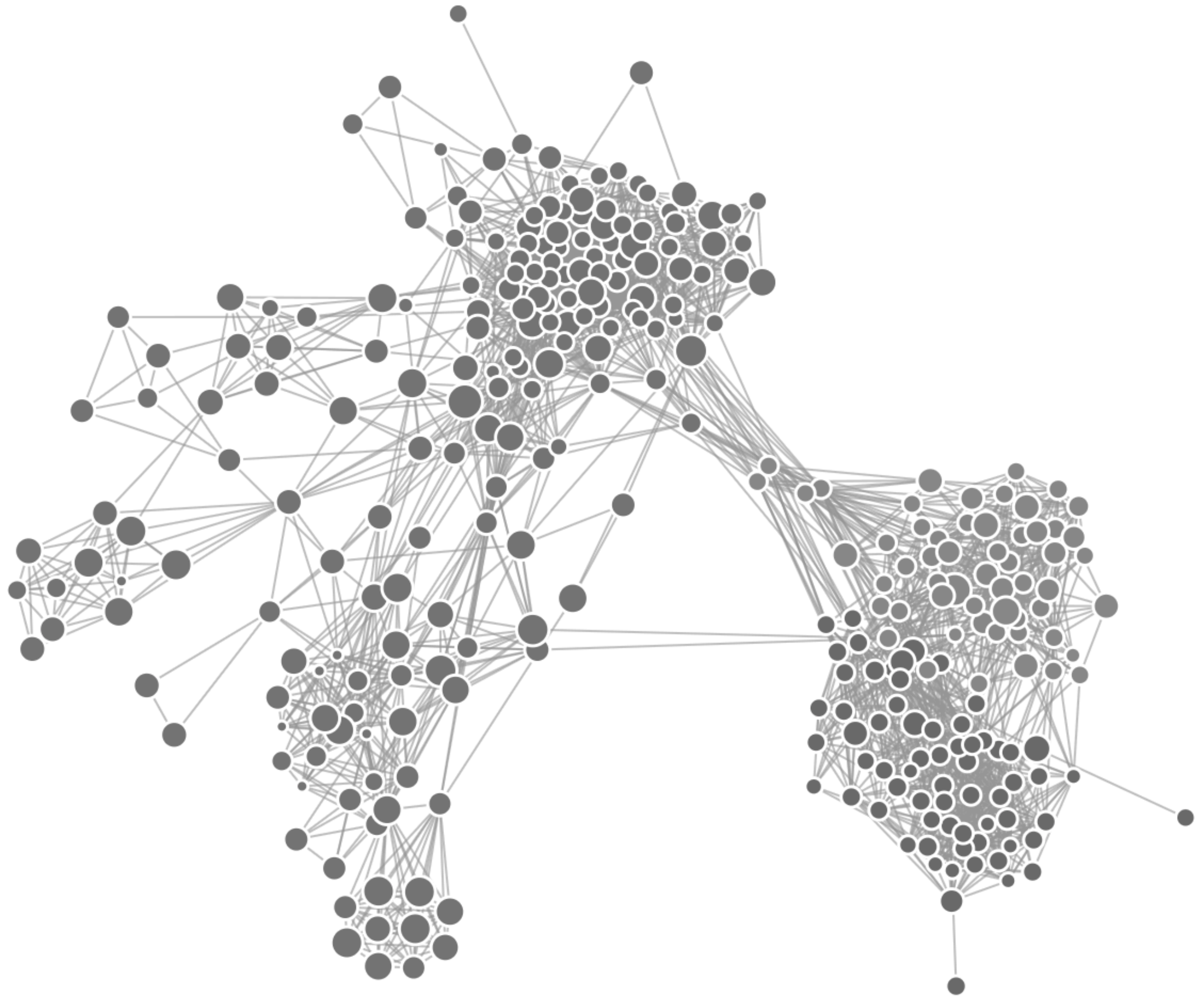
daniel.larremore@colorado.edu
@danlarremore

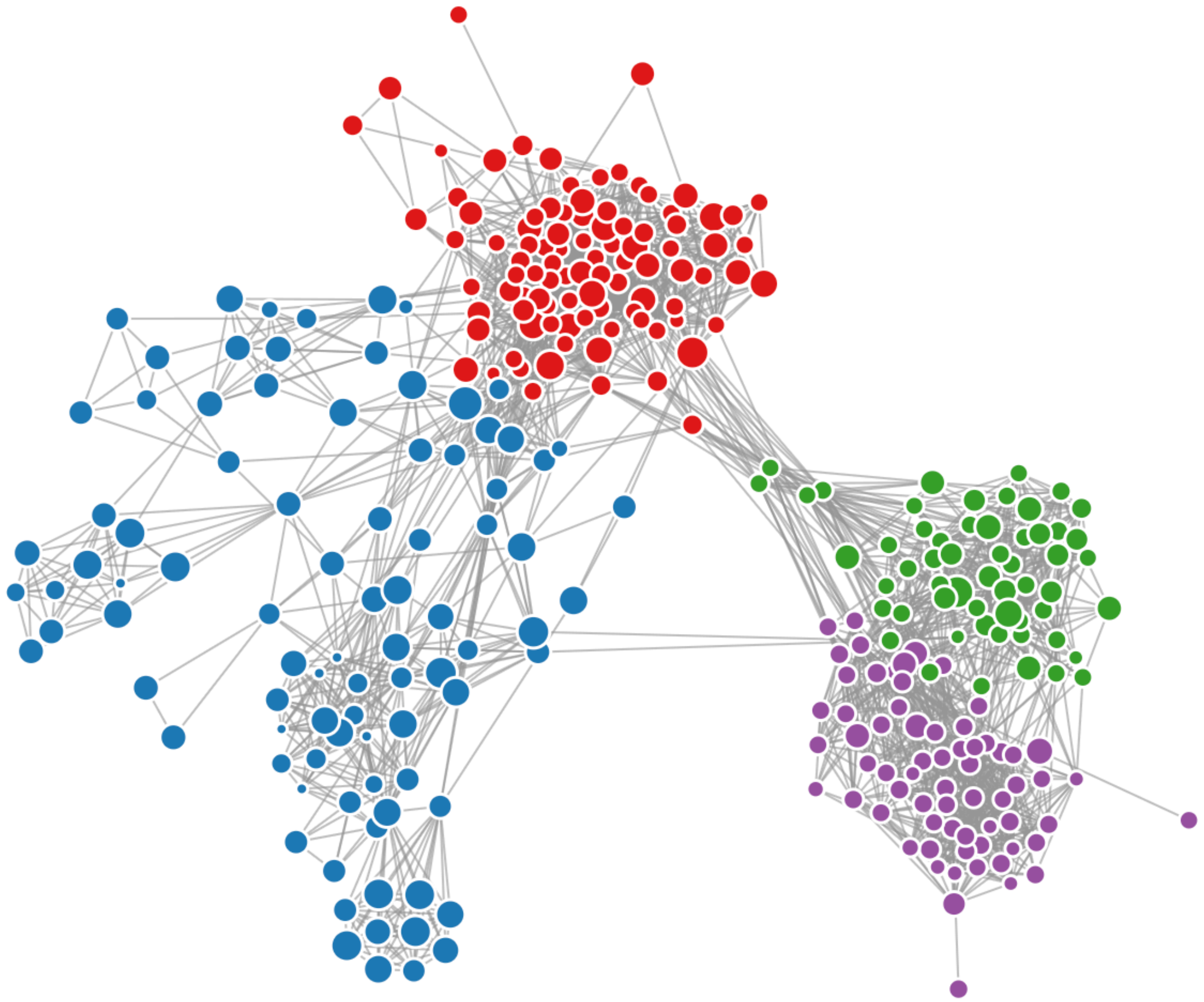
Goals for these two lectures:

1. **Why** do we look for large-scale structure? 🤔
2. **How** do we find linear hierarchies? 🧐
3. **Where** can we read more details? 📚

Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.

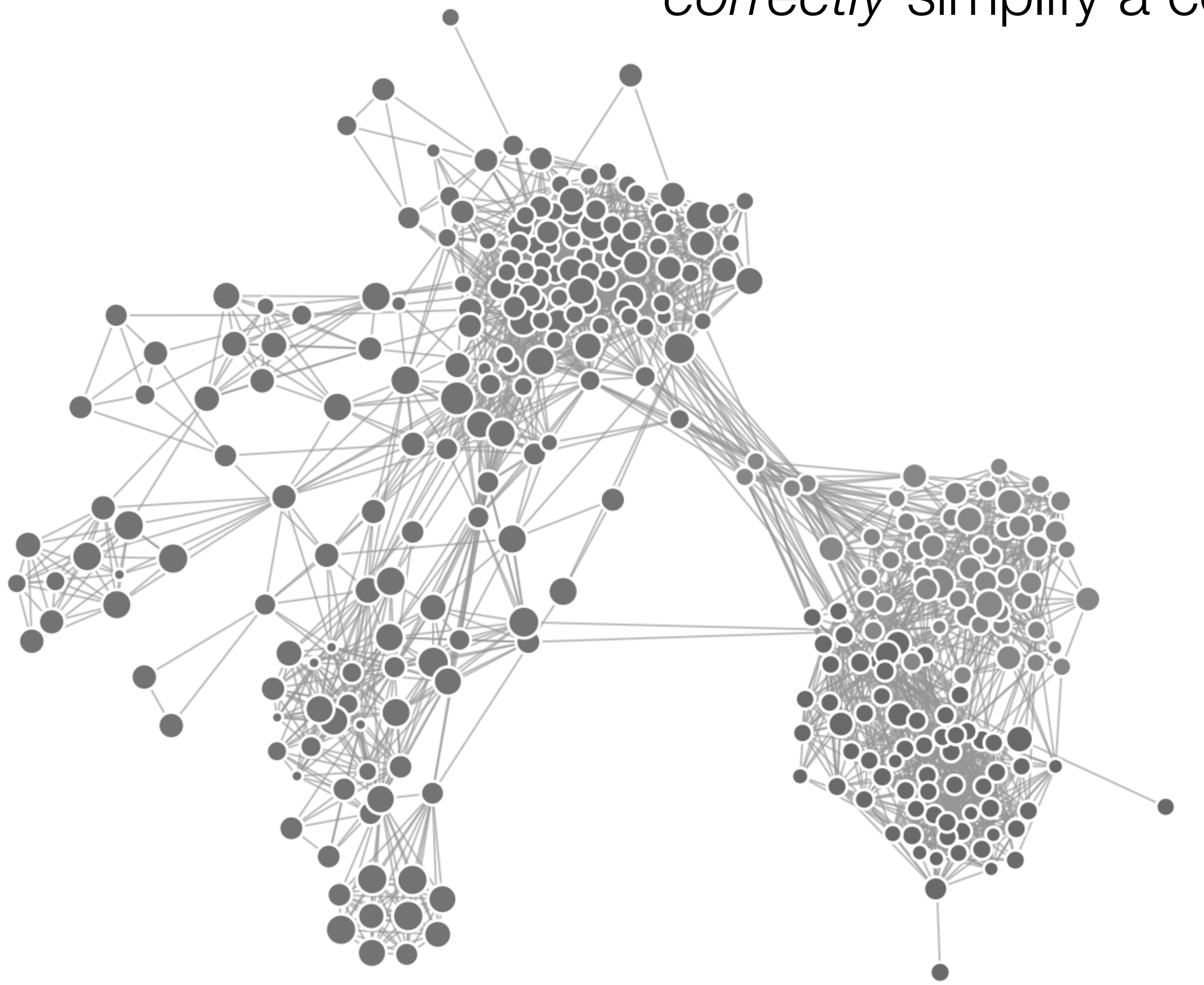
E. W. Dijkstra

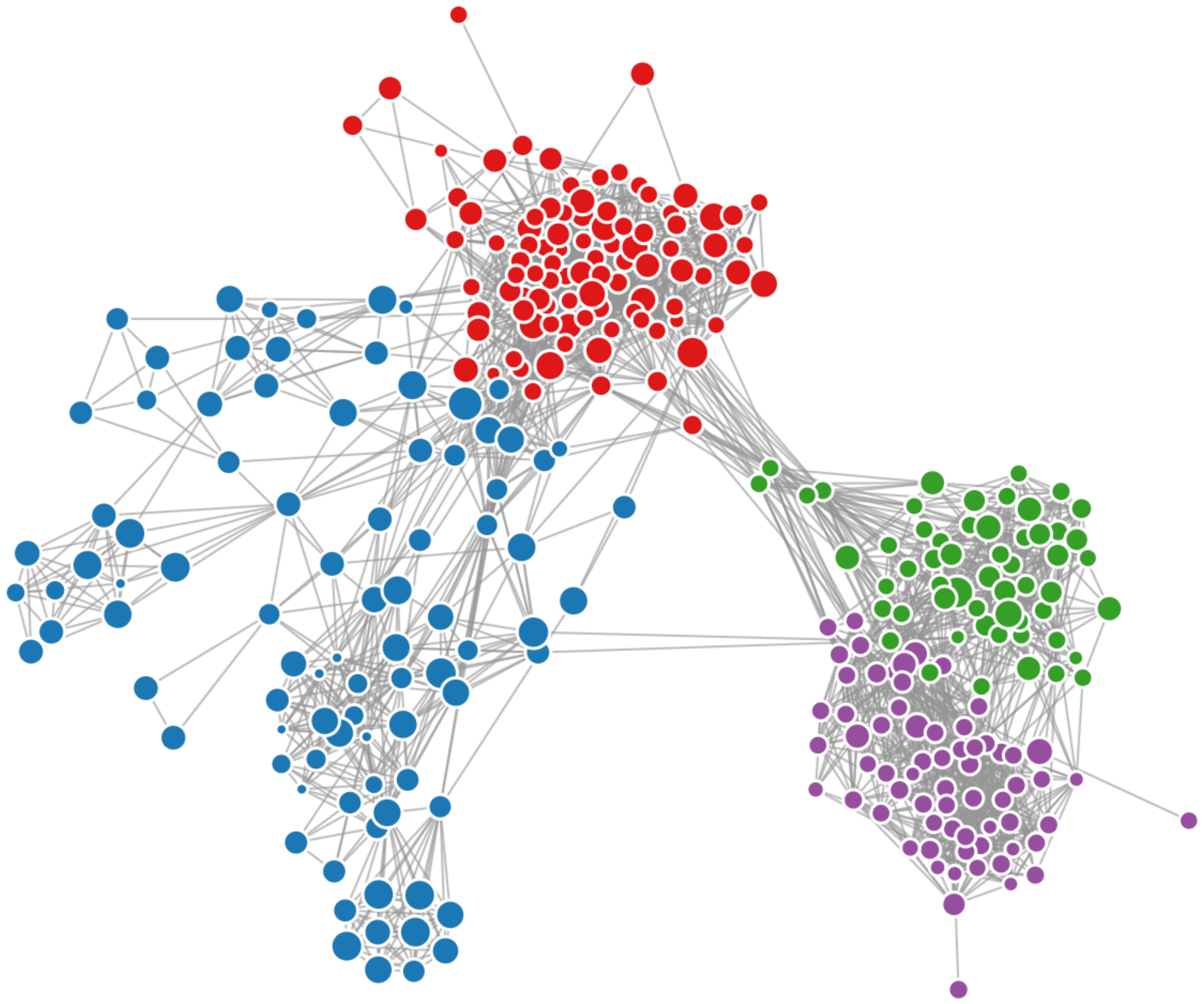




Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.
E. W. Dijkstra

We can interpret this in two ways:
The Cynic: Pictures of networks can be *really cool* but
our goal is to do good science, not make pretty pictures.
The Scientist: The most beautiful science is when we
correctly simplify a complex system.





What do we mean by “large-scale structure” ?

Structure is what makes data different from noise.

It’s what makes a network different from a random graph.

What do we mean by “large-scale structure” ?

Structure is what makes data different from noise.
It’s what makes a network different from a random graph.

Networks are often too large and complex to be adequately summarized by a few scalars, like the number of nodes, the number of edges, or the mean degree.

However, they are also often too large and complex to be analyzed *without* some kind of simplification!

What do we mean by “large-scale structure” ?

Structure is what makes data different from noise.
It’s what makes a network different from a random graph.

Networks are often too large and complex to be adequately summarized by a few scalars, like the number of nodes, the number of edges, or the mean degree.

However, they are also often too large and complex to be analyzed *without* some kind of simplification!

Therefore, understanding what the network *means* requires that we identify key structures.

Searching for large-scale structures in a network reflects a belief that in all the complexity there are patterns that make the network less complicated.

What do we mean by “large-scale structure” ?

Structure is what makes data different from noise.

It’s what makes a network different from a random graph.

Networks are often too large and complex to be adequately summarized by a few scalars, like the number of nodes, the number of edges, or the mean degree.

However, they are also often too large and complex to be analyzed *without* some kind of simplification!

Therefore, understanding what the network *means* requires that we identify key structures.

Searching for large-scale structures in a network reflects a belief that in all the complexity there are patterns that make the network less complicated.

We define these large-scale structures—models, really—to compress complex networks.

Goal: understanding, not a list of parts and dimensions

Finding large-scale structures is the same as anything else:

We want a simplified model of something very complicated.

We want to know what the important pieces are, and how they fit together.

slide more or less stolen from Aaron Clauset

Many uses for models of large-scale structure

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in "space" or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

intuition: compare this list with the list you would write for regression

Many uses for models of large-scale structure

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in "space" or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

intuition: compare this list with the list you would write for regression

Many uses for models of large-scale structure

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in "space" or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

Treat the network like a means to an end; an intermediate data structure:

Useful division. Need groups so that we can assign treatments in an A/B test.
Simplification. Downstream regression model needs ranks or groups.

intuition: compare this list with the list you would write for regression

Community structure

Rankings and linear hierarchies


Rankings and linear hierarchies


NOCOAST


NOCOAST
Dr. Joshua Garland
Time-series Chaos Wizard
Santa Fe Institute


NOCOAST
Ben Garland
Offensive Guard & Little Brother
Atlanta Falcons
Dr. Joshua Garland
Time-series Chaos Wizard
Santa Fe Institute


Brandon Garland
Firefighter & Little Brother
Grand Junction FD
Ben Garland
Offensive Guard & Little Brother
Atlanta Falcons
Dr. Joshua Garland
Time-series Chaos Wizard
Santa Fe Institute

Rankings and linear hierarchies


Ben Garland
Rankings and linear hierarchies


Ben Garland
Misuse of statistics
Rankings and linear hierarchies

The idea of rankings—pervasive!

Assumptions:

1. Competitors have some intrinsic quality (or vector of qualities).
2. Interactions can (stochastically) reveal differences in qualities.
3. Competitions are pair-wise. (Lee Sedol vs. AlphaGo; Astros vs. Dodgers)

The idea of rankings—pervasive!

Assumptions:

1. Competitors have some intrinsic quality (or vector of qualities).
2. Interactions can (stochastically) reveal differences in qualities.
3. Competitions are pair-wise. (Lee Sedol vs. AlphaGo; Astros vs. Dodgers)

In other words: outcomes are generated by a stochastic process, which is some function of the positions of the competitors.

The idea of rankings—pervasive!

Assumptions:

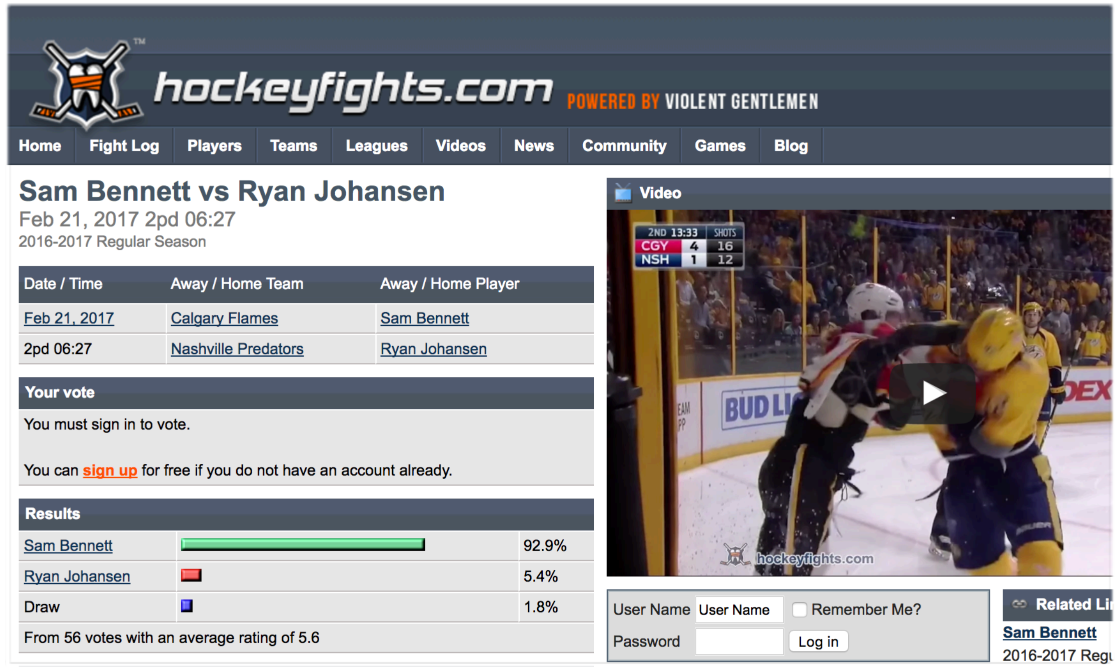
1. Competitors have some intrinsic quality (or vector of qualities).
2. Interactions can (stochastically) reveal differences in qualities.
3. Competitions are pair-wise. (Lee Sedol vs. AlphaGo; Astros vs. Dodgers)

In other words: outcomes are generated by a stochastic process, which is some function of the positions of the competitors.

Systems of dominance

social

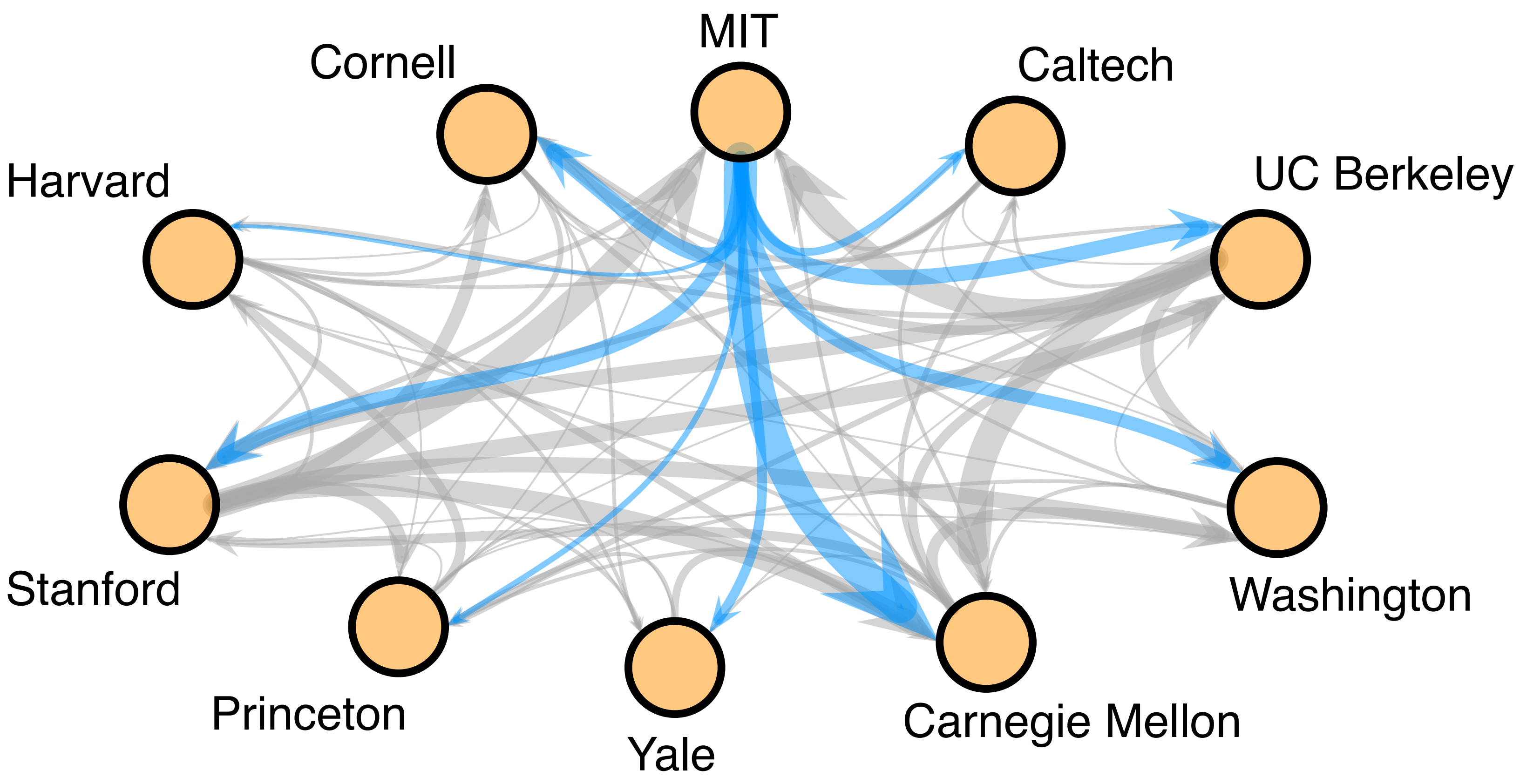
physical



mental

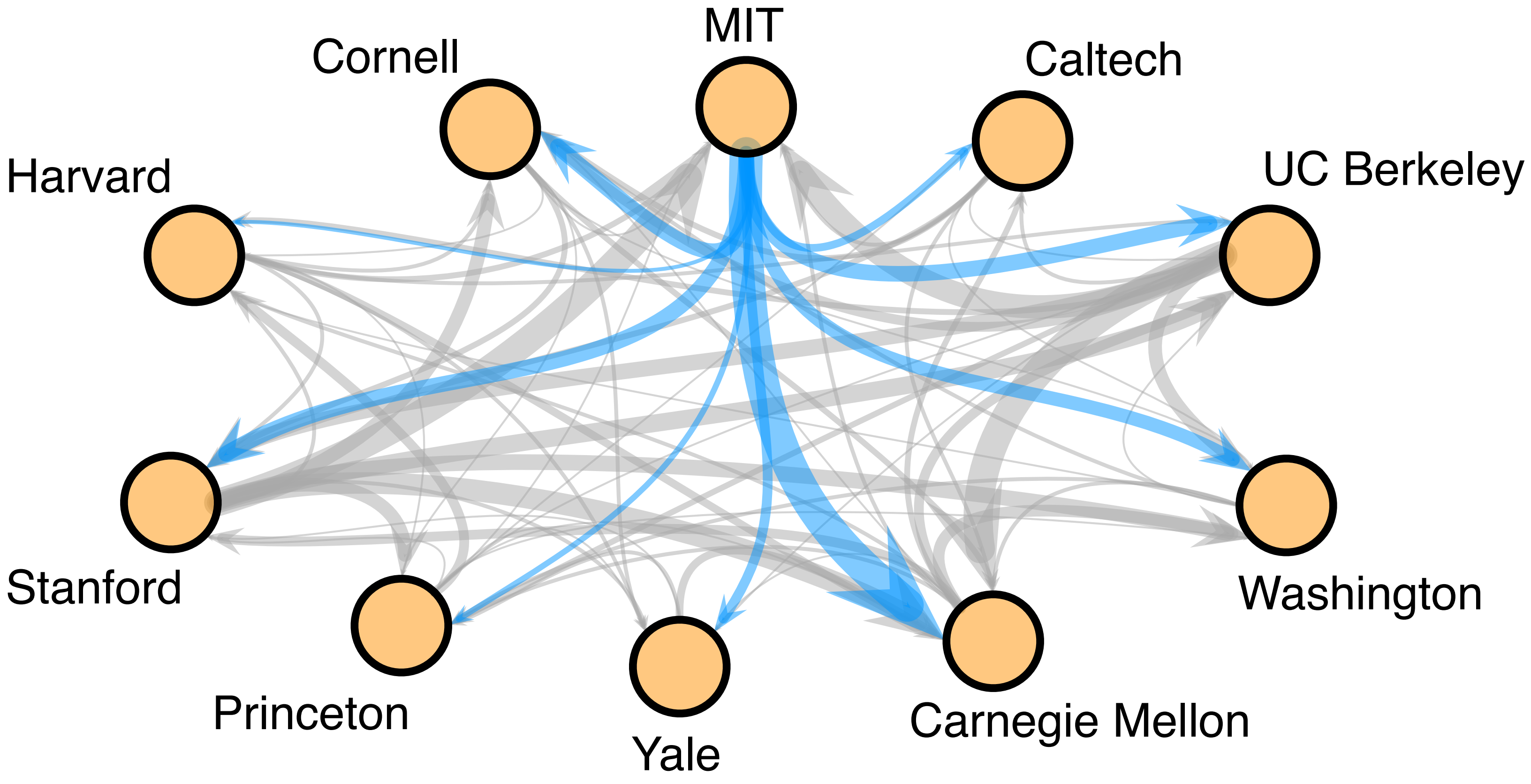
financial

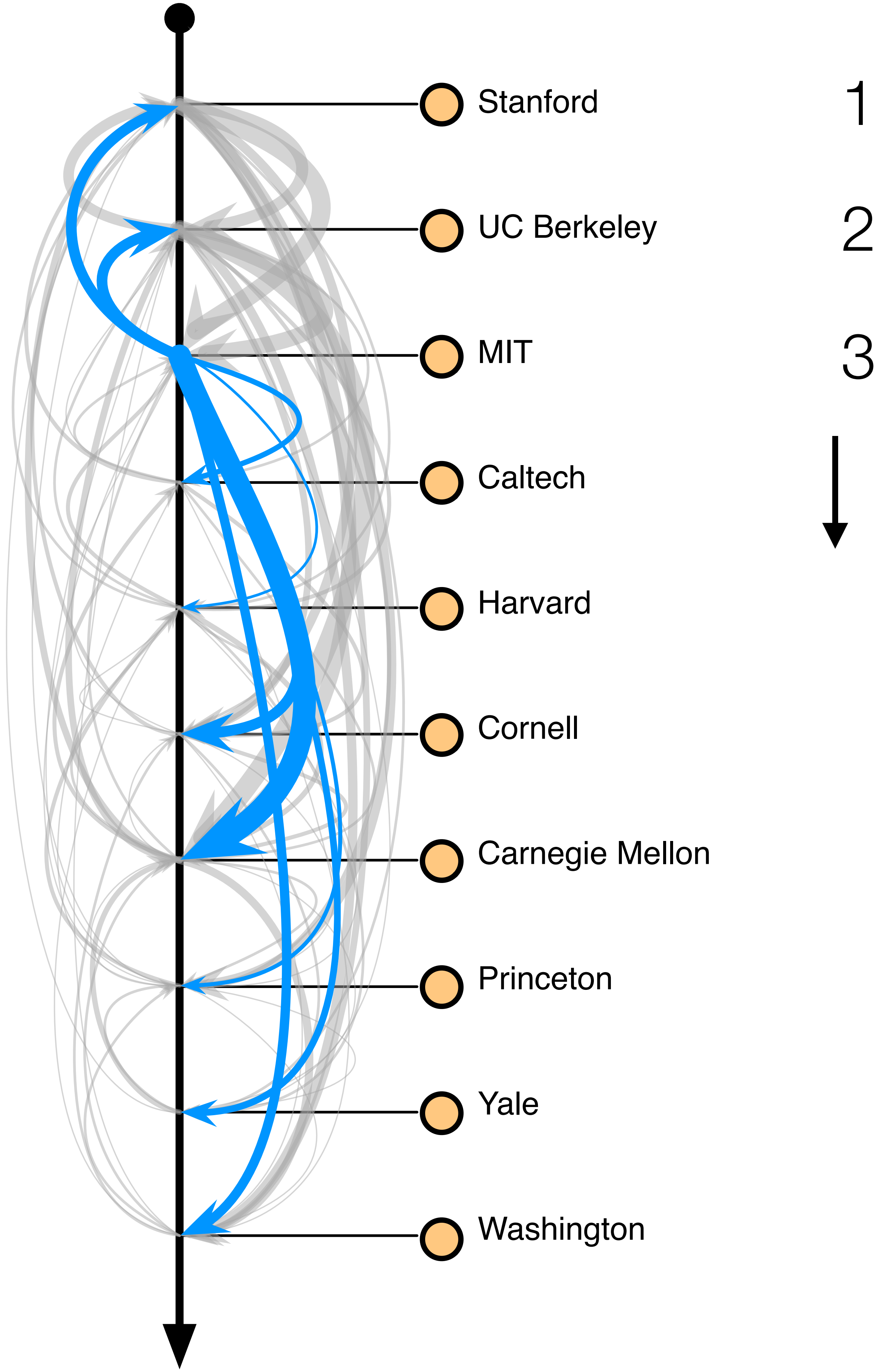
Systems of endorsement



Clauset, Arbesman, Larremore. *Science Advances* 1, e1400005 (2015).

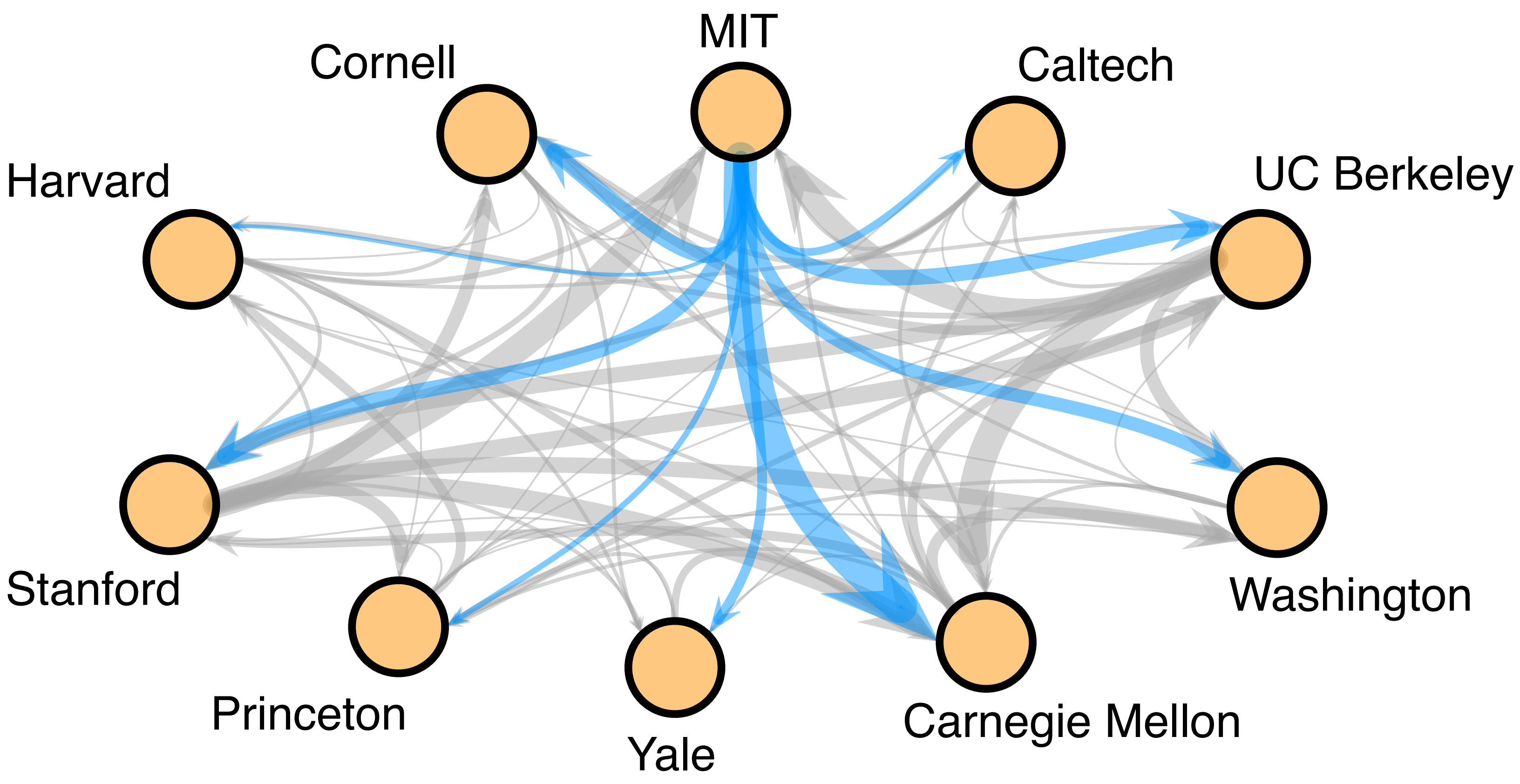
Systems of endorsement

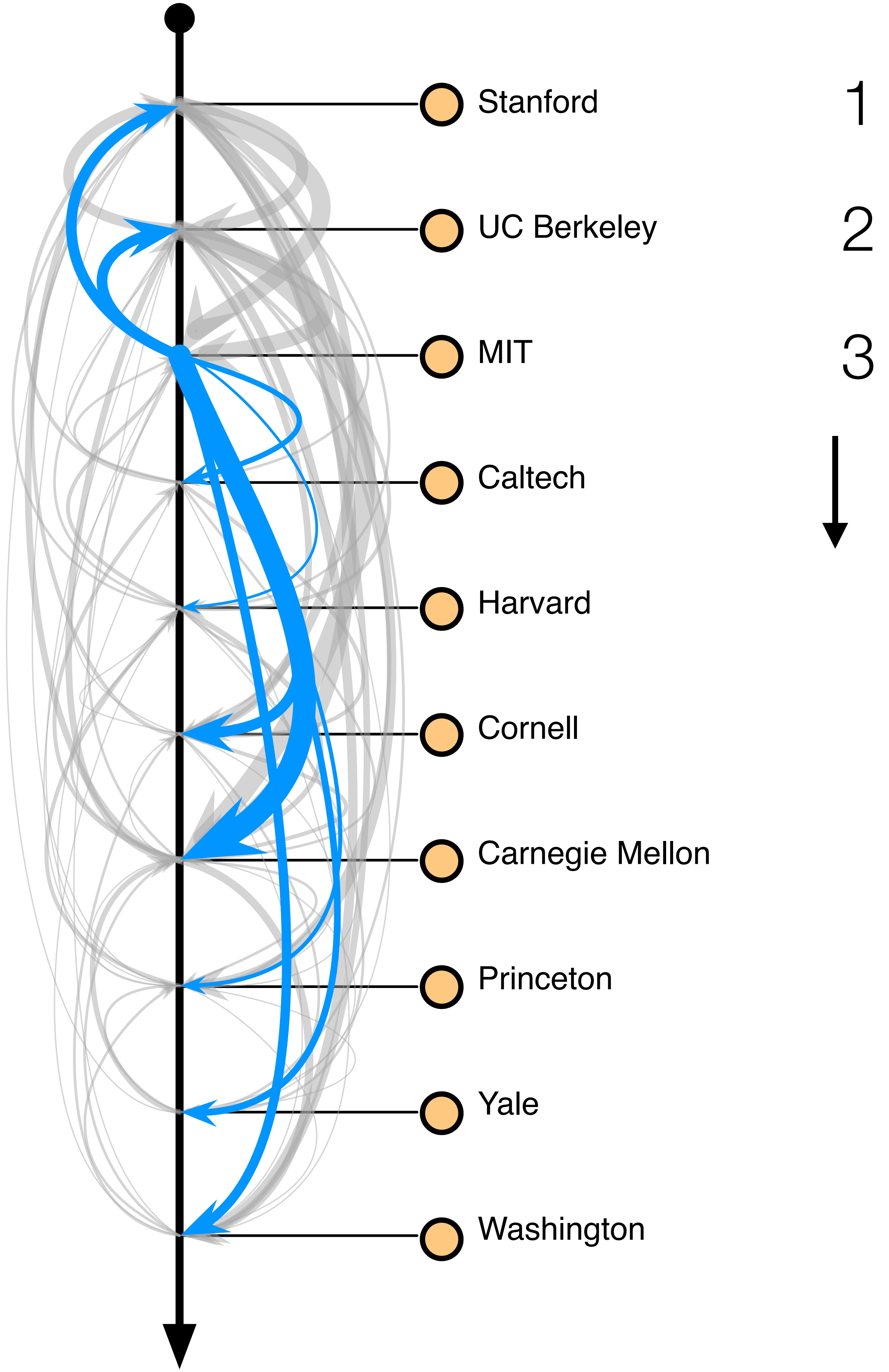




Clauset, Arbesman, Larremore. *Science Advances* 1, e1400005 (2015).

Systems of endorsement



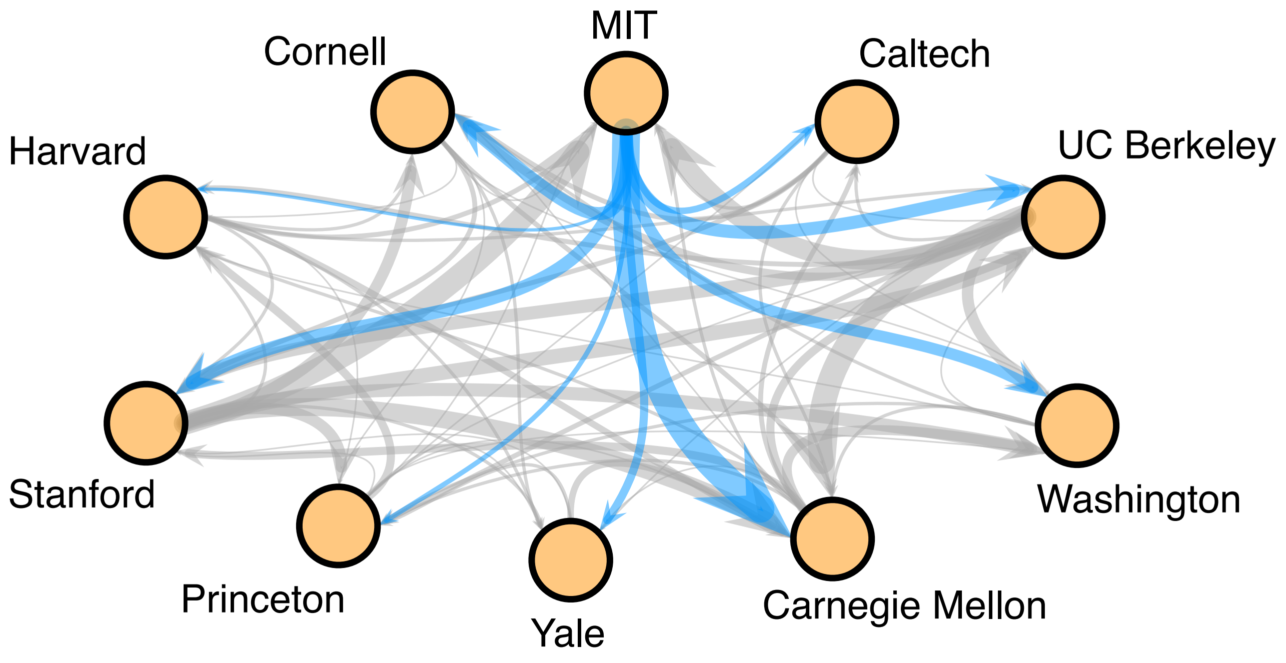


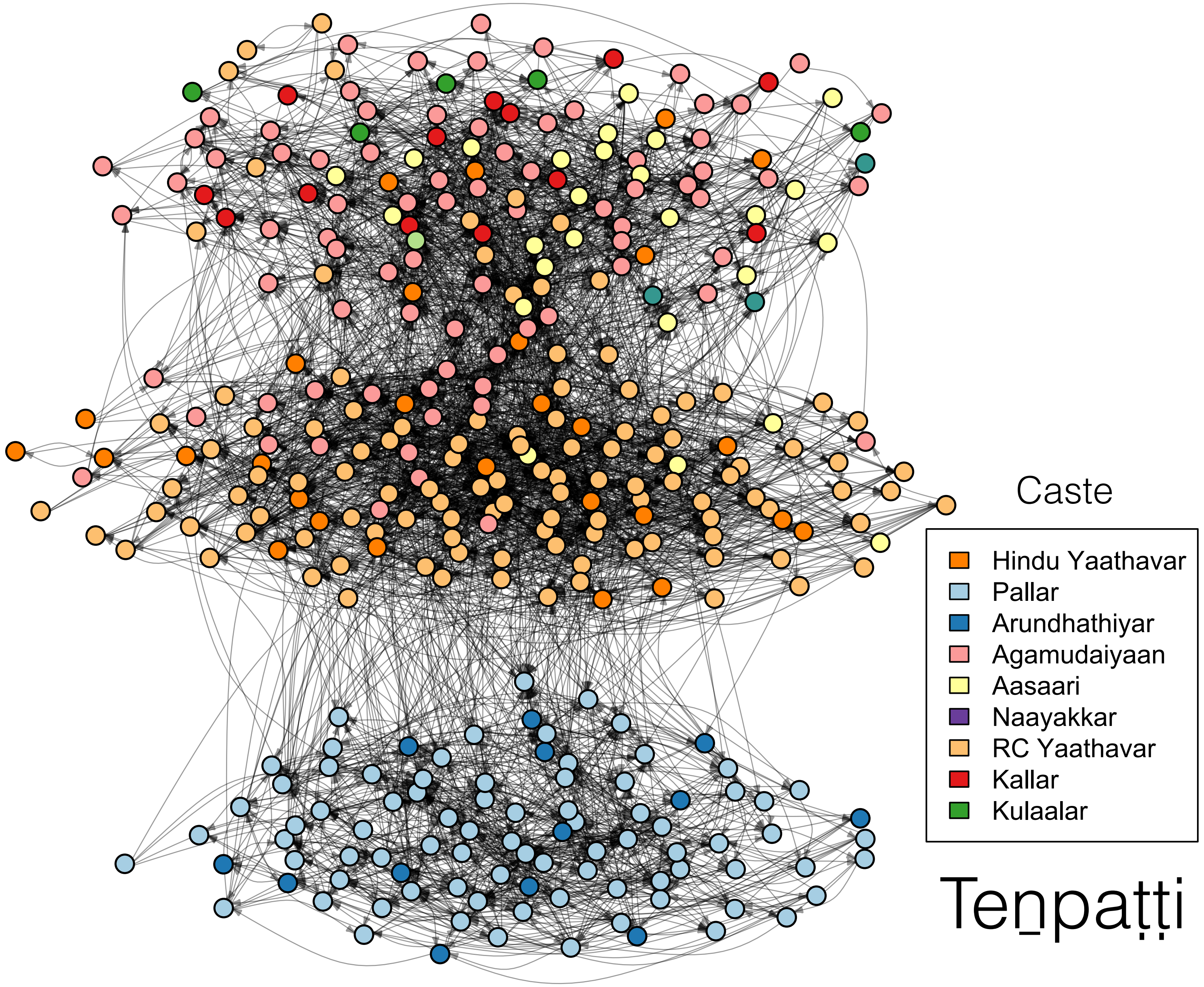
Assumptions:

1. Endorsers have some intrinsic quality.
2. Interactions can reveal differences in qualities.
3. Endorsements are pair-wise.

Clauset, Arbesman, Larremore. *Science Advances* 1, e1400005 (2015).

Systems of endorsement





Latent position can be revealed by **dominance** or **endorsement** interactions.

Figure: Larremore, Hébert-Dufresne, Power. *Draft.* Data: Power. *Nature Human Behaviour* 1, 0057 (2017)

The setup: suppose we have a *directed* network.

Its adjacency matrix is A.

$A_{ij} = A_{i \to j}$ means i beat j or i was endorsed by j

The problem: Rank the nodes.

The setup: suppose we have a *directed* network.

Its adjacency matrix is A .

$A_{ij} = A_{i \to j}$ means i beat j or i was endorsed by j

The problem: Rank the nodes.

Alternative view: there might be no network here. In some cases we're just seeing a network in pairwise comparison data because networks are a convenient data structure.

The setup: suppose we have a *directed* network.

Its adjacency matrix is A .

$A_{ij} = A_{i \to j}$ means i beat j or i was endorsed by j

The problem: Rank the nodes.

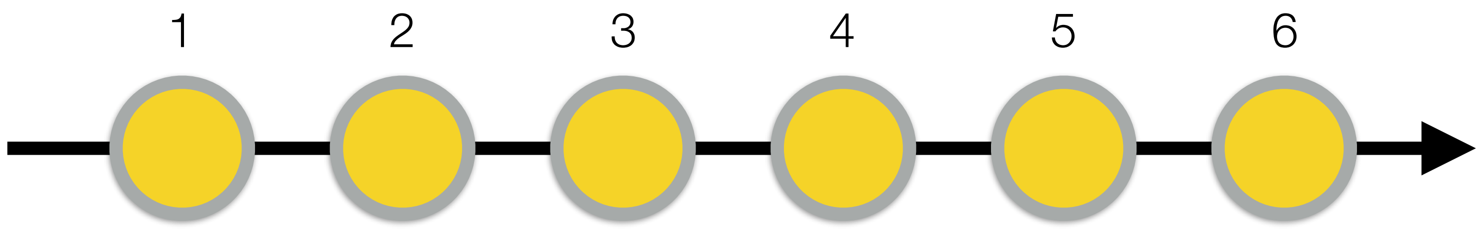
Alternative view: there might be **no network here.** In some cases we're just seeing a network in pairwise comparison data because **networks are a convenient data structure.**

Alternative problem: Which items should be compared next in order to most/best resolve our estimate of the ranks? (sequential tournament design)

Embeddings vs Orderings

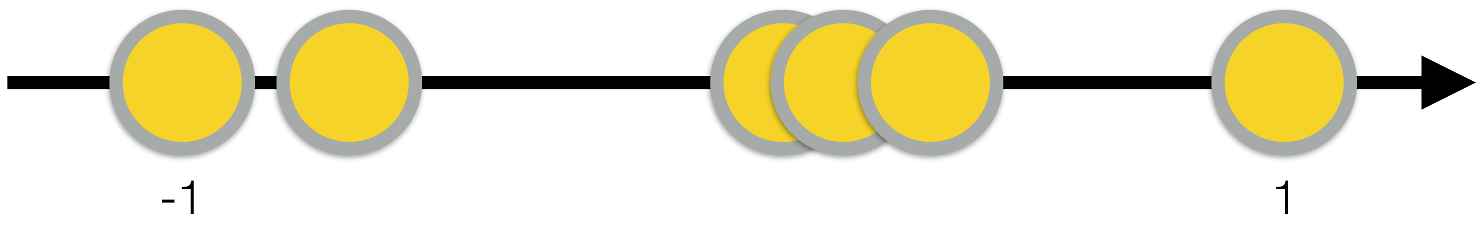
Ordering place the nodes in order:

1, 2, 3, …



Embedding assigns a position to each node:

1, 1.2, 7, 20, 21, 21.2, …



Which one should I use?

> Depends on the use case.

> Is it possible for two nodes to occupy the same rank or position? If so, an embedding is more appropriate. Also better when meaning of 1-rank Δ varies.

> Consider that you can always go from an embedding to an ordering, if you have a rule for breaking ties.

Win-Loss is not satisfactory: schedule matters

Beating the grandmaster counts for more than beating a novice.

Win-Loss is not satisfactory: schedule matters

Beating the grandmaster counts for more than beating a novice.

Win and loss tallies don’t take this “schedule difficulty” into account. Put differently, win-loss records leave information on the table.

Win-Loss is not satisfactory: schedule matters

Beating the grandmaster counts for more than beating a novice.

Win and loss tallies don't take this "schedule difficulty" into account. Put differently, win-loss records **leave information on the table**.

One way to make use of this information:

i **beats** j implies $s_i > s_j$

Therefore if we have a whole list of outcomes, we can try to find a total ordering that breaks as few of these implications as possible.

Win-Loss is not satisfactory: schedule matters

Beating the grandmaster counts for more than beating a novice.

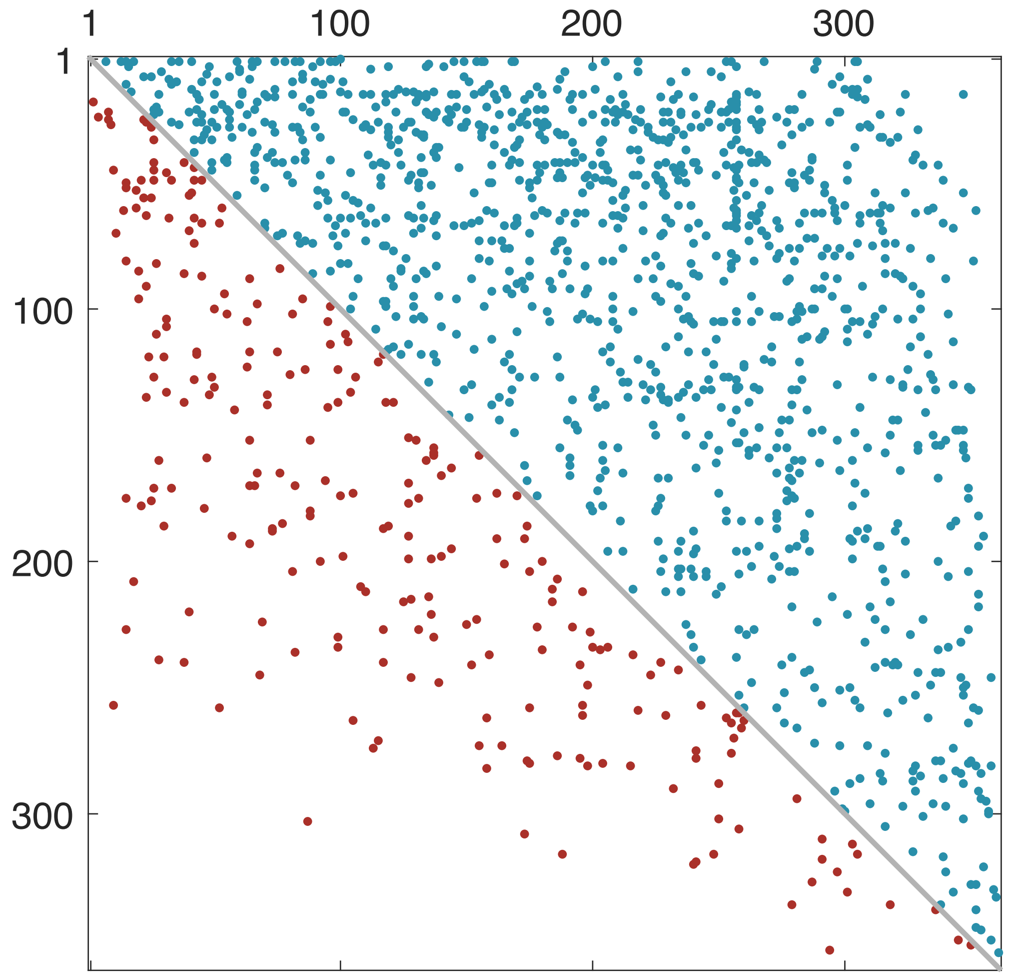
Win and loss tallies don't take this "schedule difficulty" into account. Put differently, win-loss records **leave information on the table.**

One way to make use of this information:

i **beats** *j* implies $s_i > s_j$

Therefore if we have a whole list of outcomes, we can try to find a total ordering that breaks as few of these implications as possible.

$A_{ij} =$ number of times that *i* beat *j*.



minimum violation ranking: sort *A*.

Win-Loss is not satisfactory: schedule matters

How do we find an ordering that minimizes the number of violations (or upsets) ?

Win-Loss is not satisfactory: schedule matters

How do we find an ordering that minimizes the number of violations (or upsets) ?

Recipe (MCMC):

1. Order the nodes randomly.
2. Compute the number of violations. In expectation, this should be 50% of edges.

Win-Loss is not satisfactory: schedule matters

How do we find an ordering that minimizes the number of violations (or upsets) ?

Recipe (MCMC):

1. Order the nodes randomly.
2. Compute the number of violations. In expectation, this should be 50% of edges.
3. Pick two nodes at random and propose to swap their positions.
4. Compute the number of violations in this scenario.

Win-Loss is not satisfactory: schedule matters

How do we find an ordering that minimizes the number of violations (or upsets) ?

Recipe (MCMC):
1. Order the nodes randomly.
2. Compute the number of violations. In expectation, this should be 50% of edges.

3. Pick two nodes at random and propose to swap their positions.
4. Compute the number of violations in this scenario.

5. If #violations decreases or stays the same, keep the swap. Otherwise, reject.
6. Repeat until….?

Win-Loss is not satisfactory: schedule matters

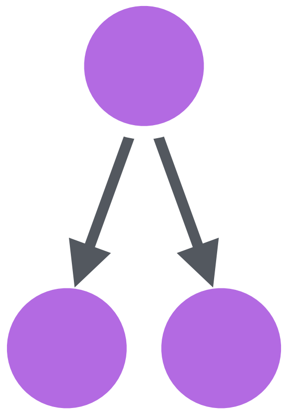
How do we find an ordering that minimizes the number of violations (or upsets) ?

Recipe (MCMC):
1. Order the nodes randomly.
2. Compute the number of violations. In expectation, this should be 50% of edges.

3. Pick two nodes at random and propose to swap their positions.
4. Compute the number of violations in this scenario.

5. If #violations decreases or stays the same, keep the swap. Otherwise, reject.
6. Repeat until….?

Notes:
* The number of violations is non-increasing over time.
* There may be no unique minimum. Consider this scenario:



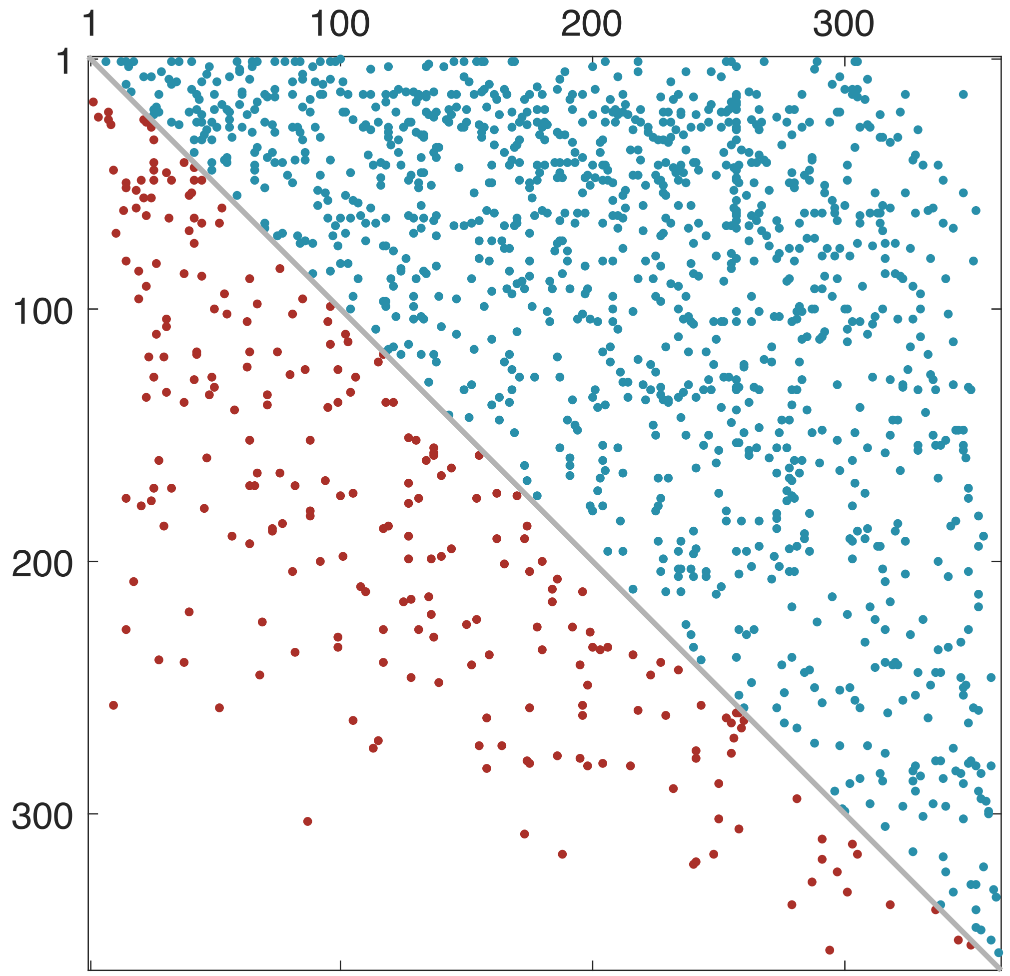
MVR: non-unique & rough optimization landscape

There is no guarantee of a unique minimizing ranking *s.*

Space of ordinal rankings has n! elements—usually use MCMC to search.

Slow.

Ordinal. No ties. No interpretability of rank differences.



minimum violation ranking: sort *A.*

MVR: non-unique & rough optimization landscape

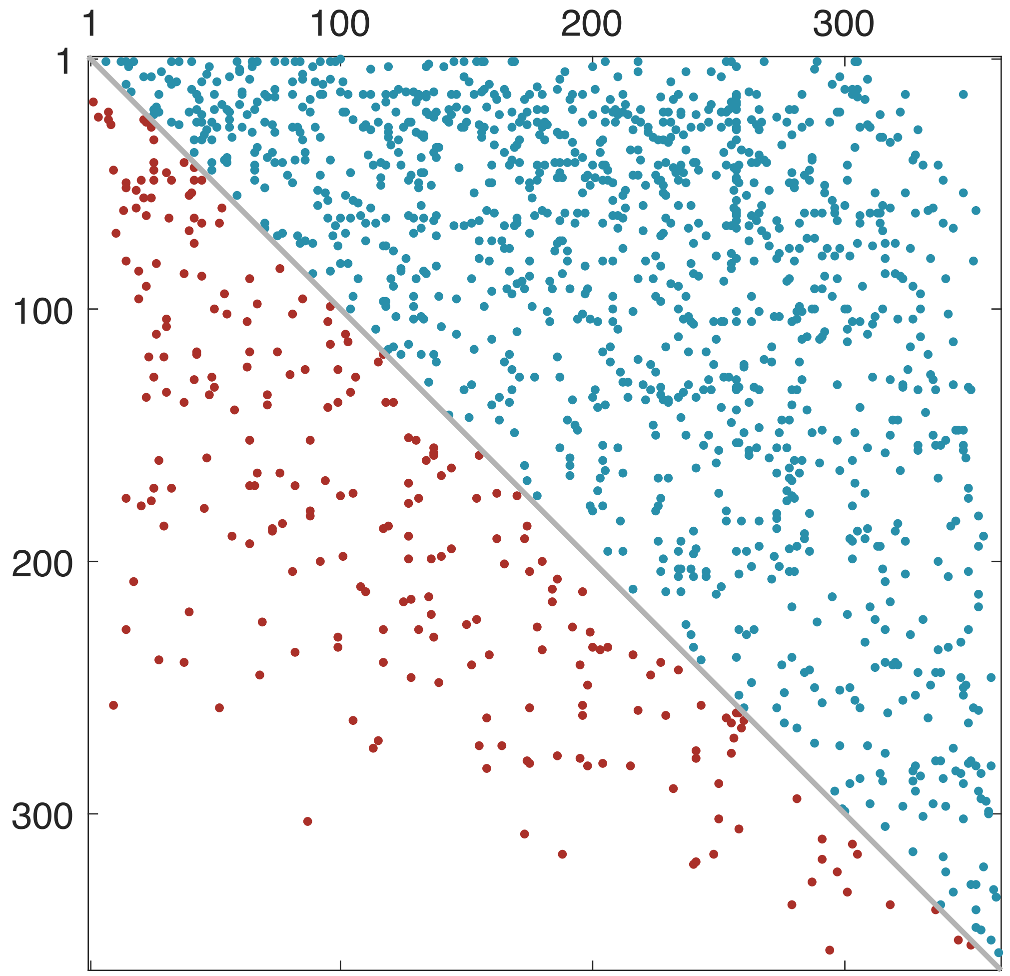
There is no guarantee of a unique minimizing ranking *s.*

Space of ordinal rankings has n! elements—usually use MCMC to search.

Slow.

Ordinal. No ties. No interpretability of rank differences.

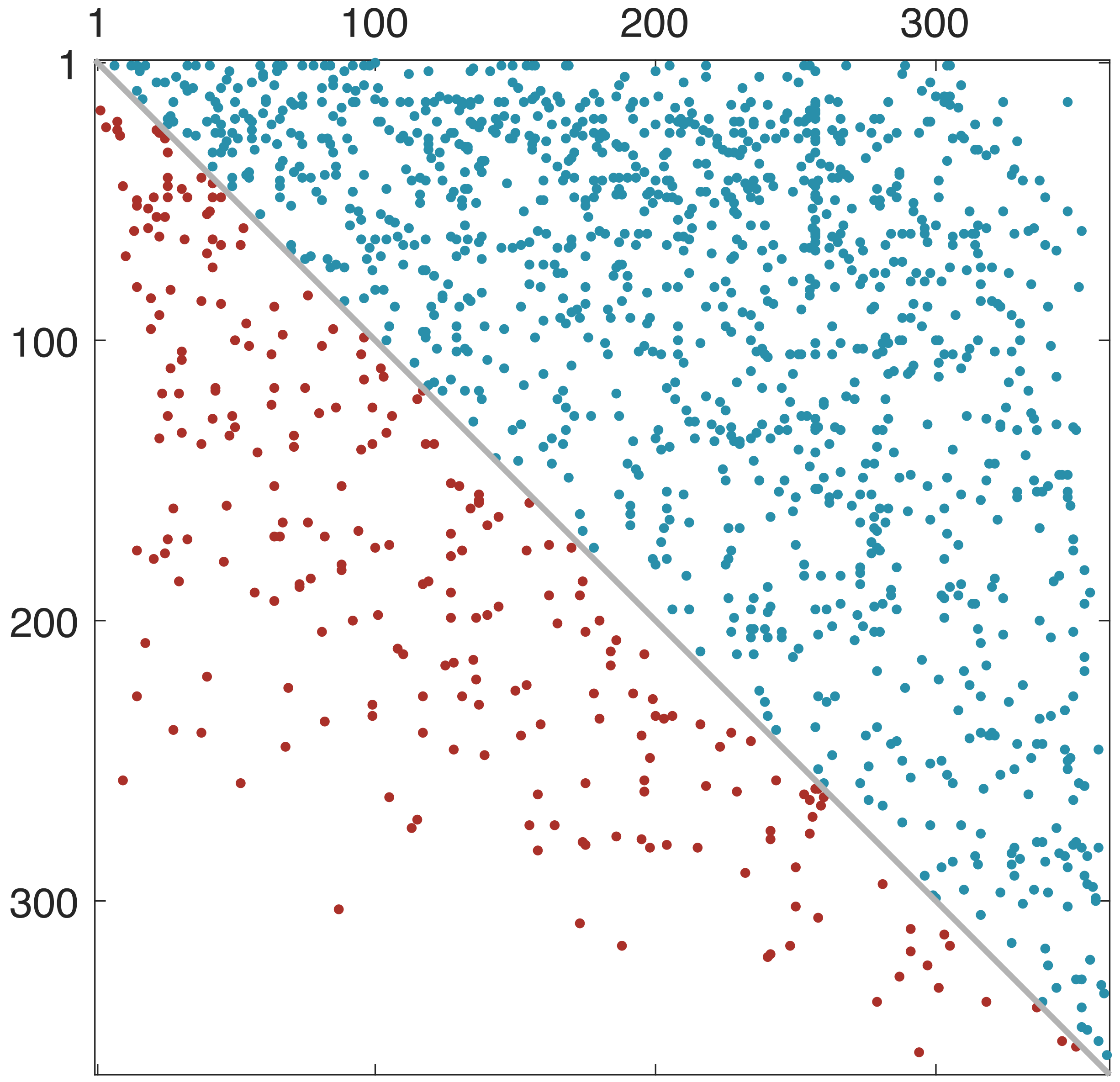
What are other premises on which we can base a ranking model?



minimum violation ranking: sort *A.*

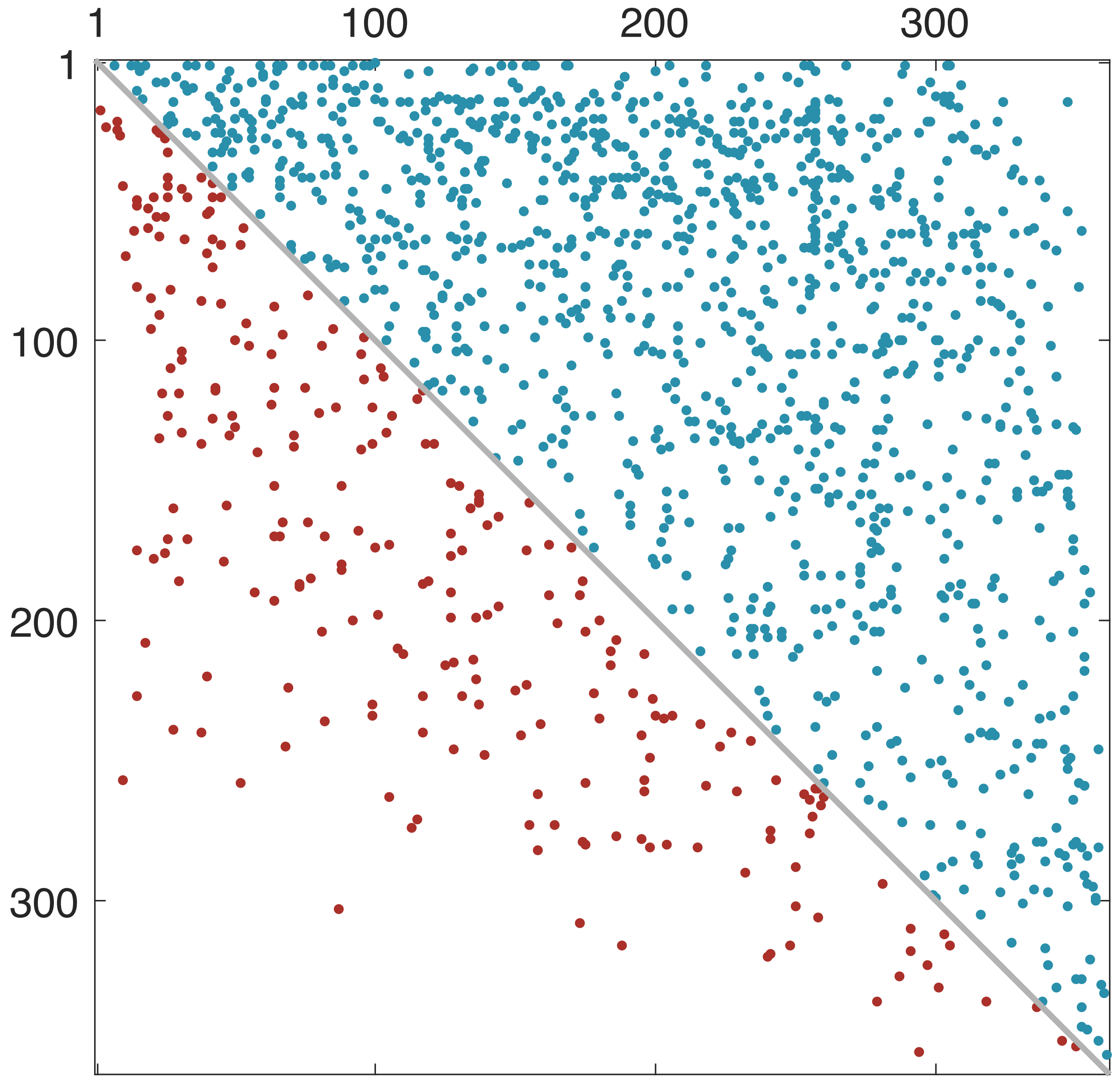
Embeddings and Orderings 1: MVR & Agony

What if you allowed for **ties** and then ran Minimum Violation Ranking (MVR)? What would happen?



Embeddings and Orderings 1: MVR & Agony

What if you allowed for **ties** and then ran Minimum Violation Ranking (MVR)? What would happen?



MVR: uniform cost (1 per edge).
Agony: generic cost function.
  for example, difference in ranks.

Embeddings and Orderings 1: Discrete choice models

Louis Leon Thurstone and Thelma Thurstone

Embeddings and Orderings 1: Discrete choice models

https://xkcd.com/883/

Embeddings and Orderings 1: Discrete choice models

Instead of rating everything from 1 to 10, try *paired comparisons*.

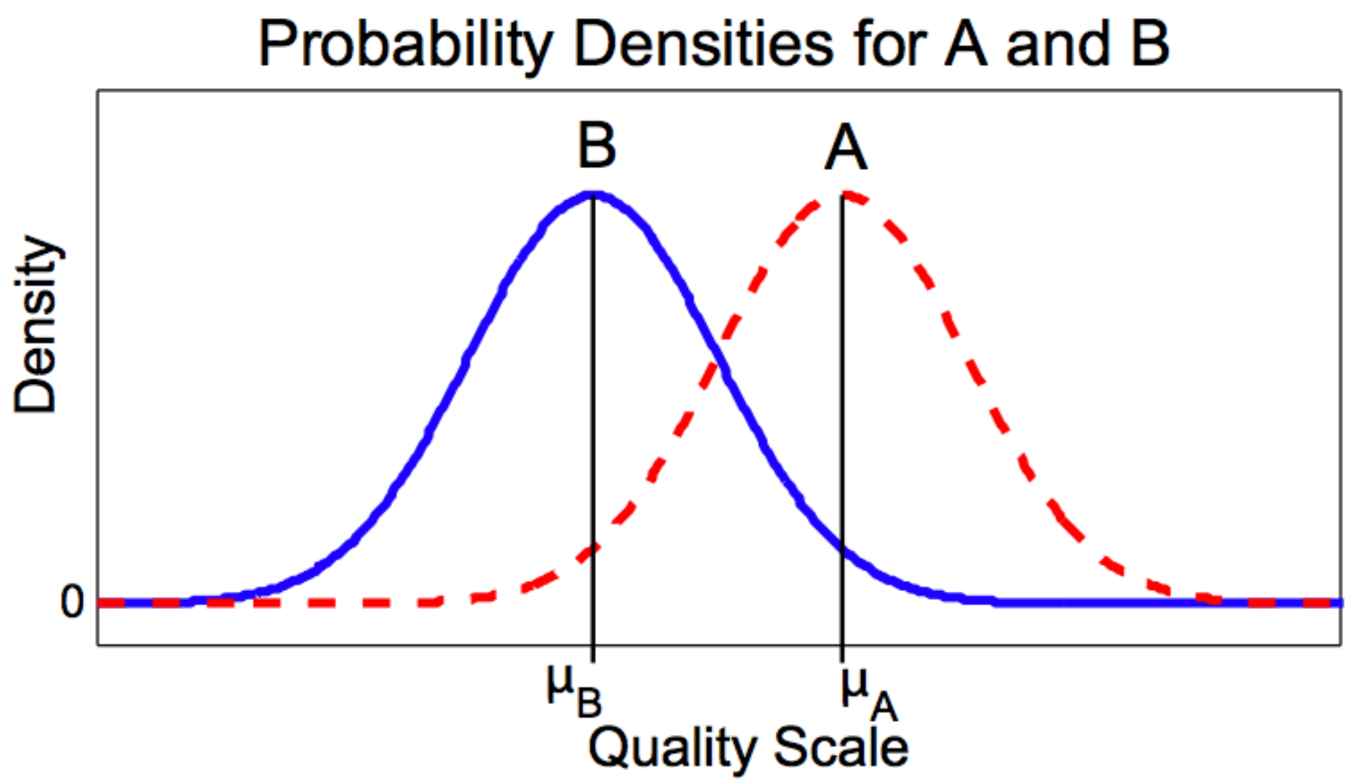
Do you prefer *i* or *j* ?

Why? Consider: My 3 is not your 3. What is 1 and what is 10?

https://xkcd.com/883/

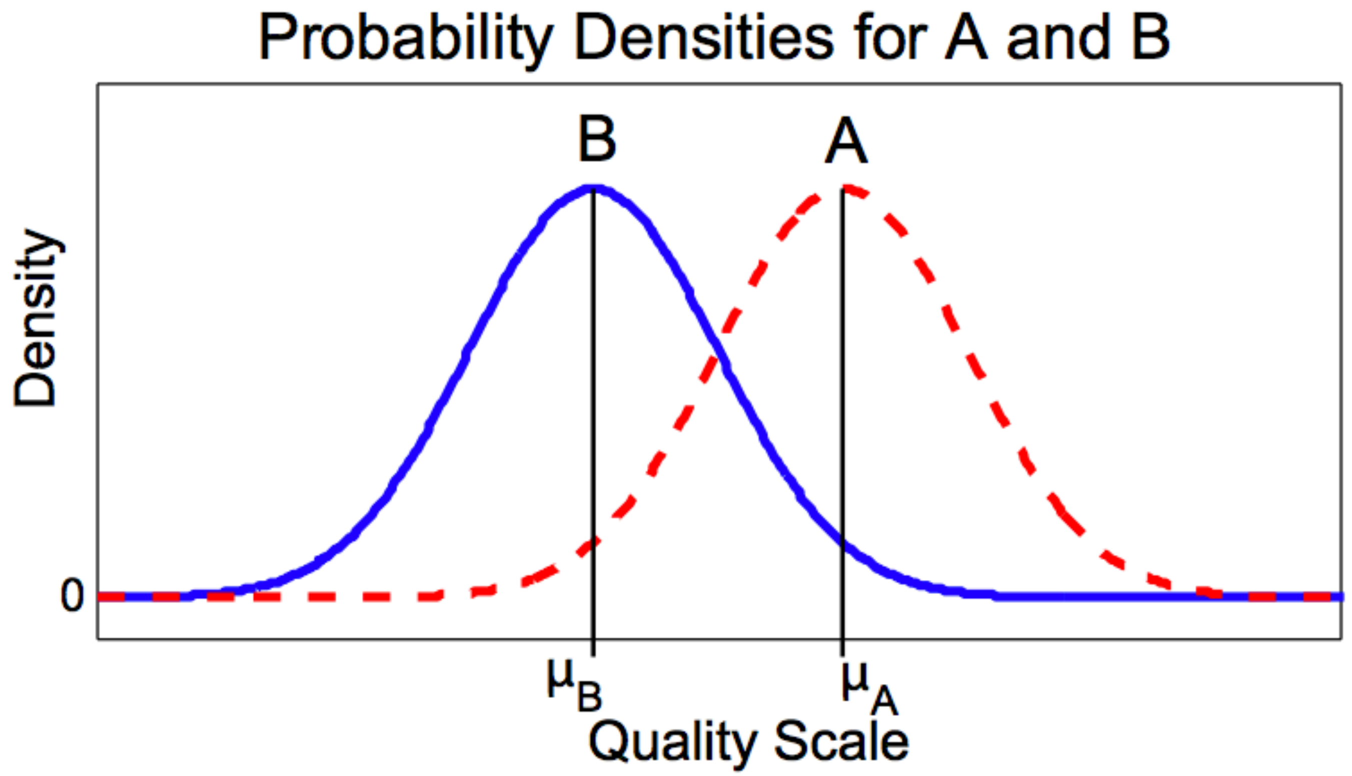
Embeddings and Orderings 1: Discrete choice models

Thurstone: items have quality distributions. When a person judges whether A is better than B they draw from A's distribution and from B's distribution and see which is higher.



Embeddings and Orderings 1: Discrete choice models

Thurstone: items have quality distributions. When a person judges whether A is better than B they draw from A's distribution and from B's distribution and see which is higher.

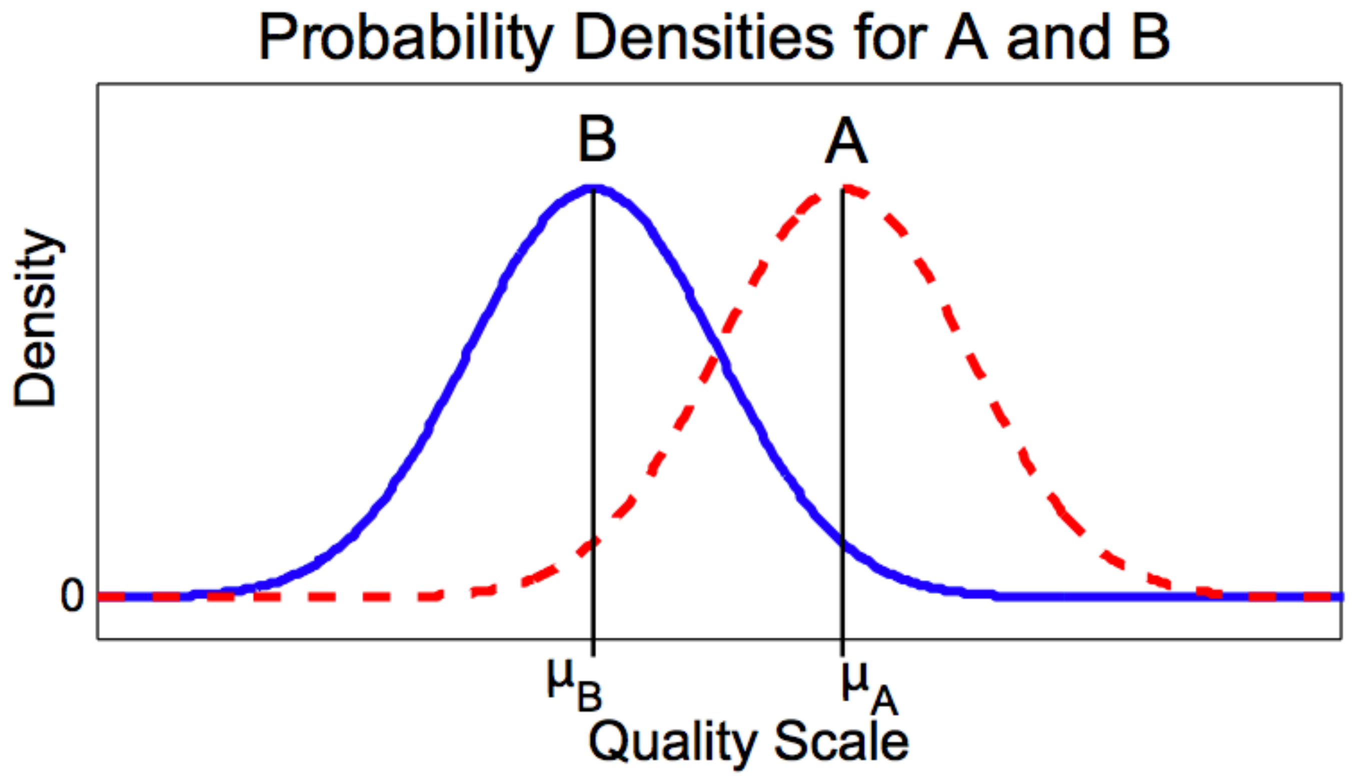


Thurstone modeled these as Gaussians.

$$P(A > B) = P(A - B > 0)$$

Embeddings and Orderings 1: Discrete choice models

Thurstone: items have quality distributions. When a person judges whether A is better than B they draw from A's distribution and from B's distribution and see which is higher.



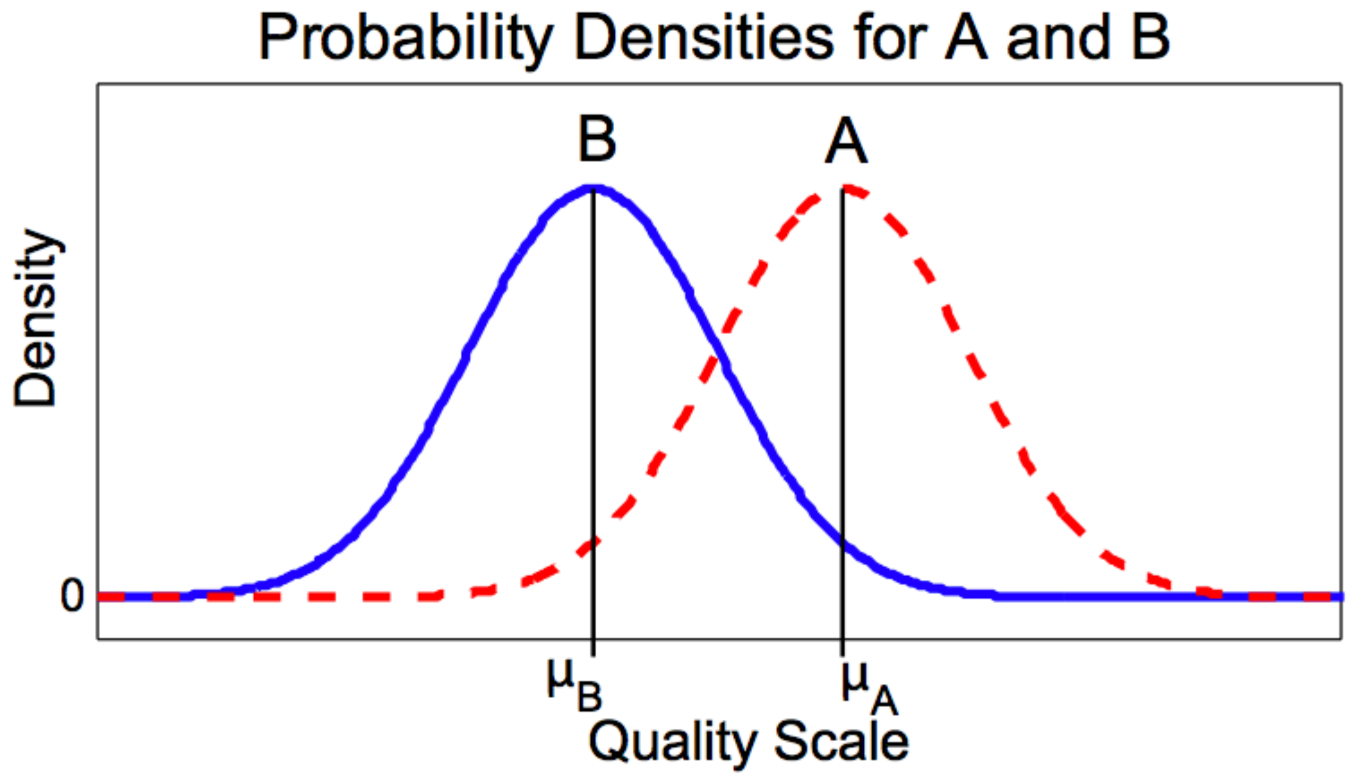
Thurstone modeled these as Gaussians.

$$P(A > B) = P(A - B > 0)$$

Difference of Gaussians is Gaussian.

Embeddings and Orderings 1: Discrete choice models

Thurstone: items have quality distributions. When a person judges whether A is better than B they draw from A's distribution and from B's distribution and see which is higher.



Thurstone modeled these as Gaussians.

$$P(A > B) = P(A - B > 0)$$

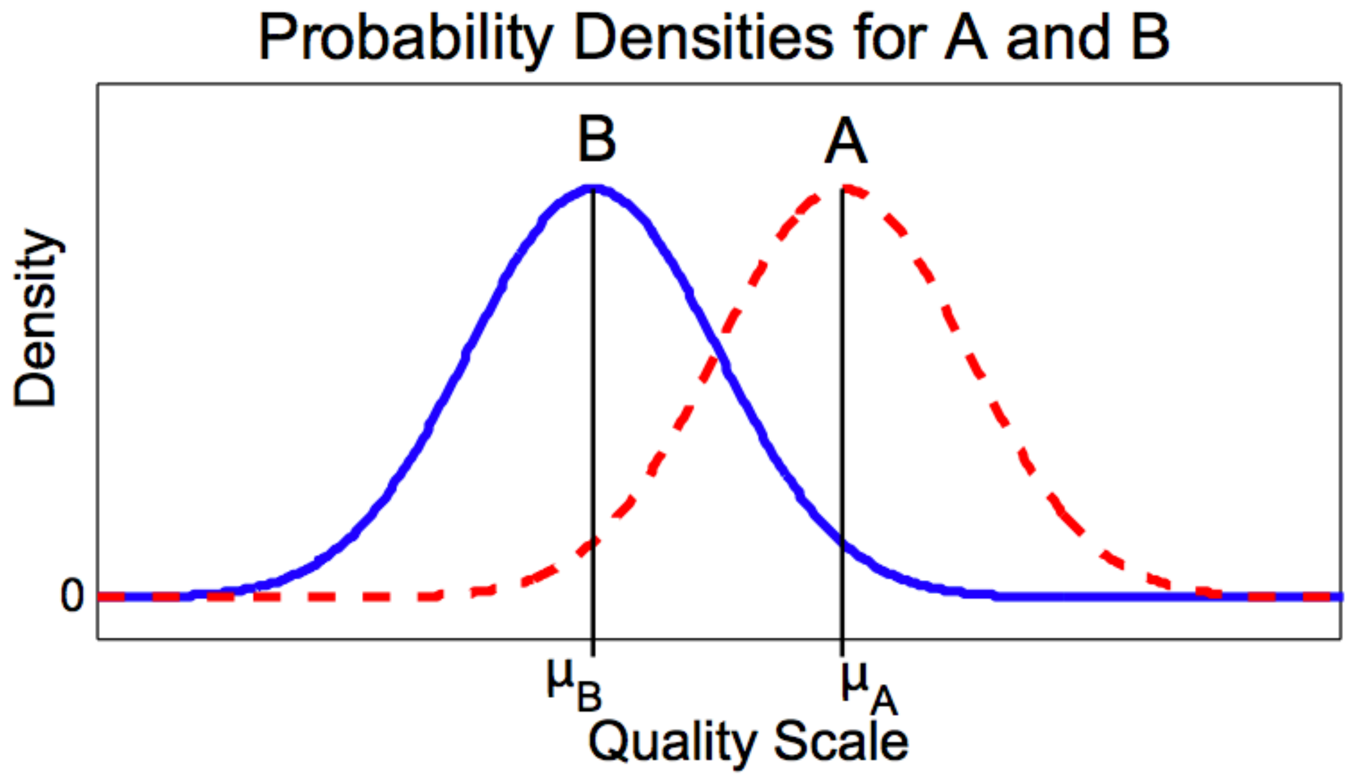
Difference of Gaussians is Gaussian.

$$\hat{\mu}_{AB} = \Phi^{-1}\left(\frac{C_{A\to B}}{C_{A\to B} + C_{B\to A}}\right)$$

Where $\Phi^{-1}(x)$ is the inverse CDF of standard normal, a.k.a. the *probit*.

Embeddings and Orderings 1: Discrete choice models

Thurstone: items have quality distributions. When a person judges whether A is better than B they draw from A's distribution and from B's distribution and see which is higher.



Thurstone modeled these as Gaussians.

$$P(A > B) = P(A - B > 0)$$

Difference of Gaussians is Gaussian.

$$\hat{\mu}_{AB} = \Phi^{-1}\left(\frac{C_{A\to B}}{C_{A\to B} + C_{B\to A}}\right)$$

Where $\Phi^{-1}(x)$ is the inverse CDF of standard normal, a.k.a. the *probit*.

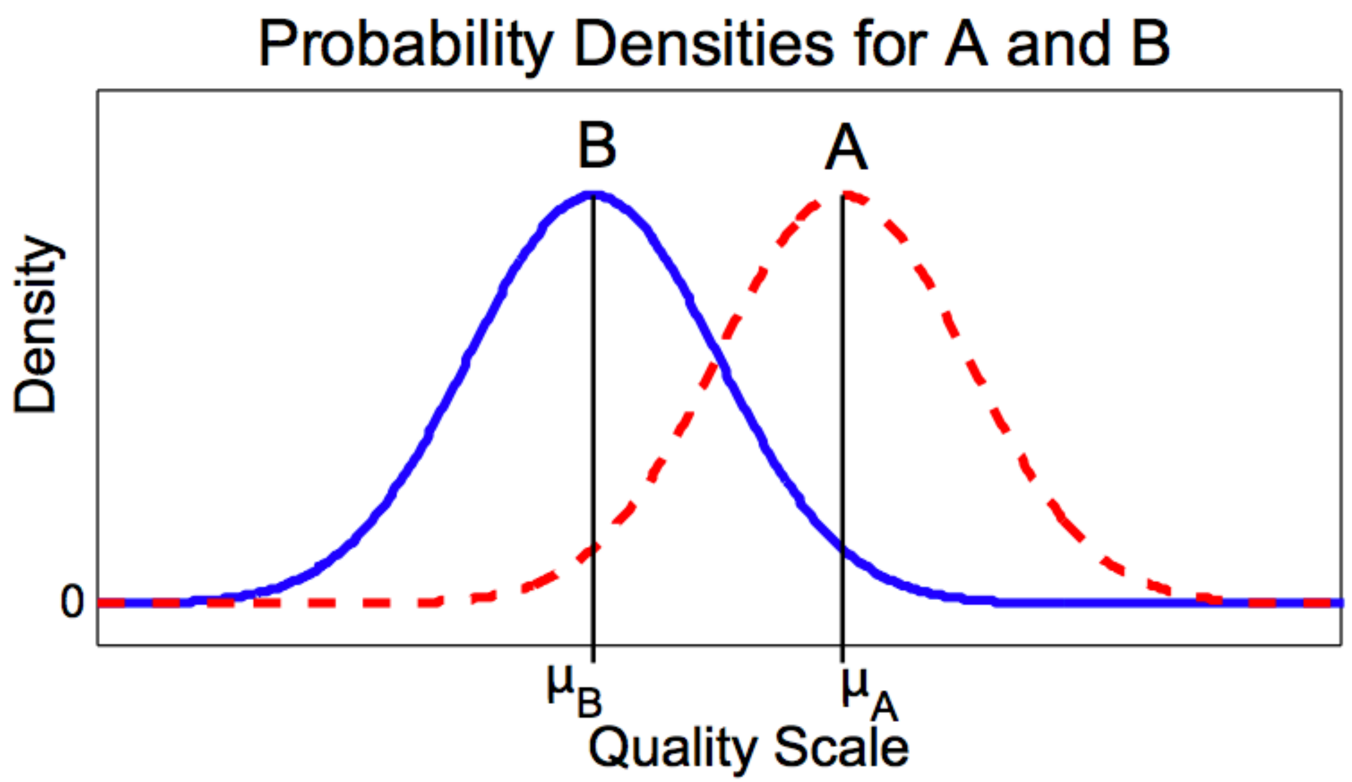
Powerful idea: lots of pairwise comparisons = estimates of all the qualities! An embedding!

Key: pairwise comparisons = directed network. i preferred to $j \;=\; i \rightarrow j$

Finding the qualities of items from pairwise comparisons = Finding embedding of nodes.

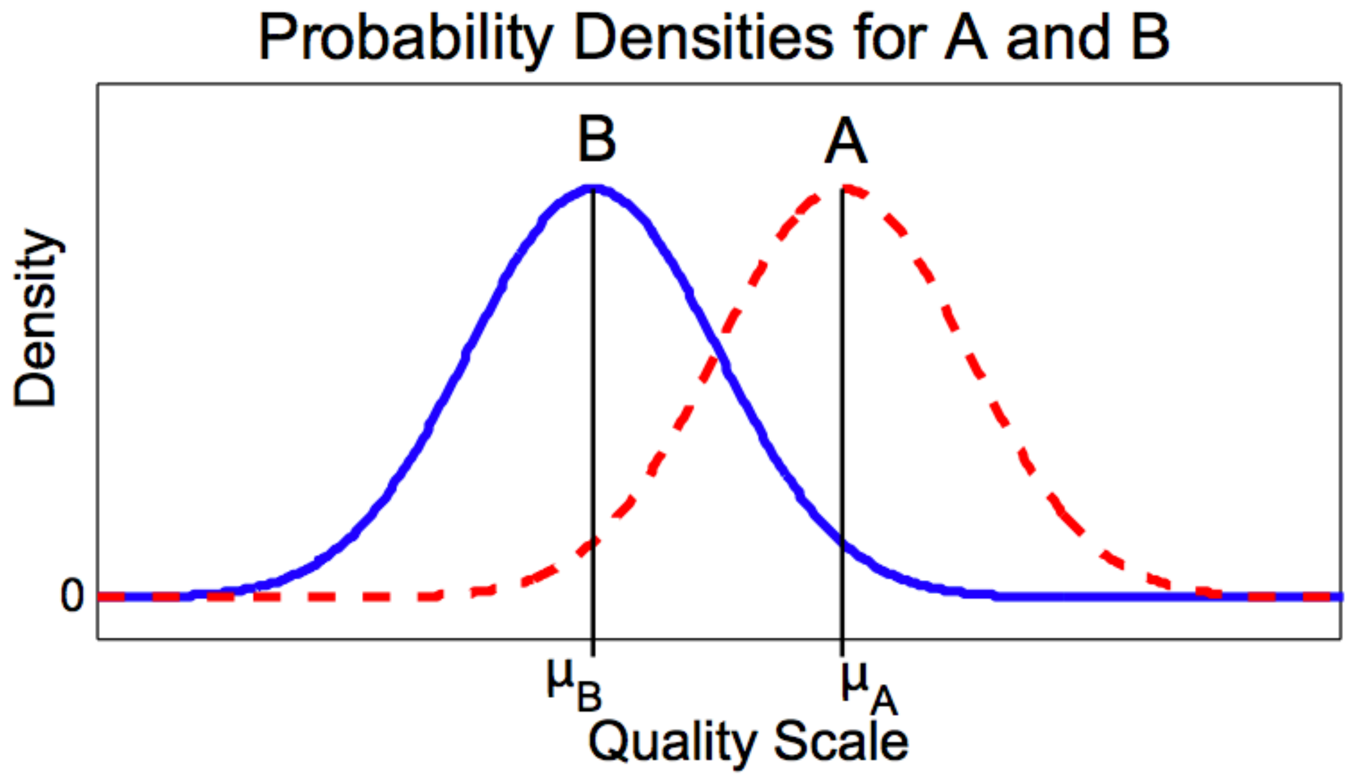
Embeddings and Orderings 1: Discrete choice models

Bradley-Terry & Luce: items have quality distributions. When a person judges whether A is better than B they draw from A's and from B's distribution and see which is higher.



Embeddings and Orderings 1: Discrete choice models

Bradley-Terry & Luce: items have quality distributions. When a person judges whether A is better than B they draw from A's and from B's distribution and see which is higher.

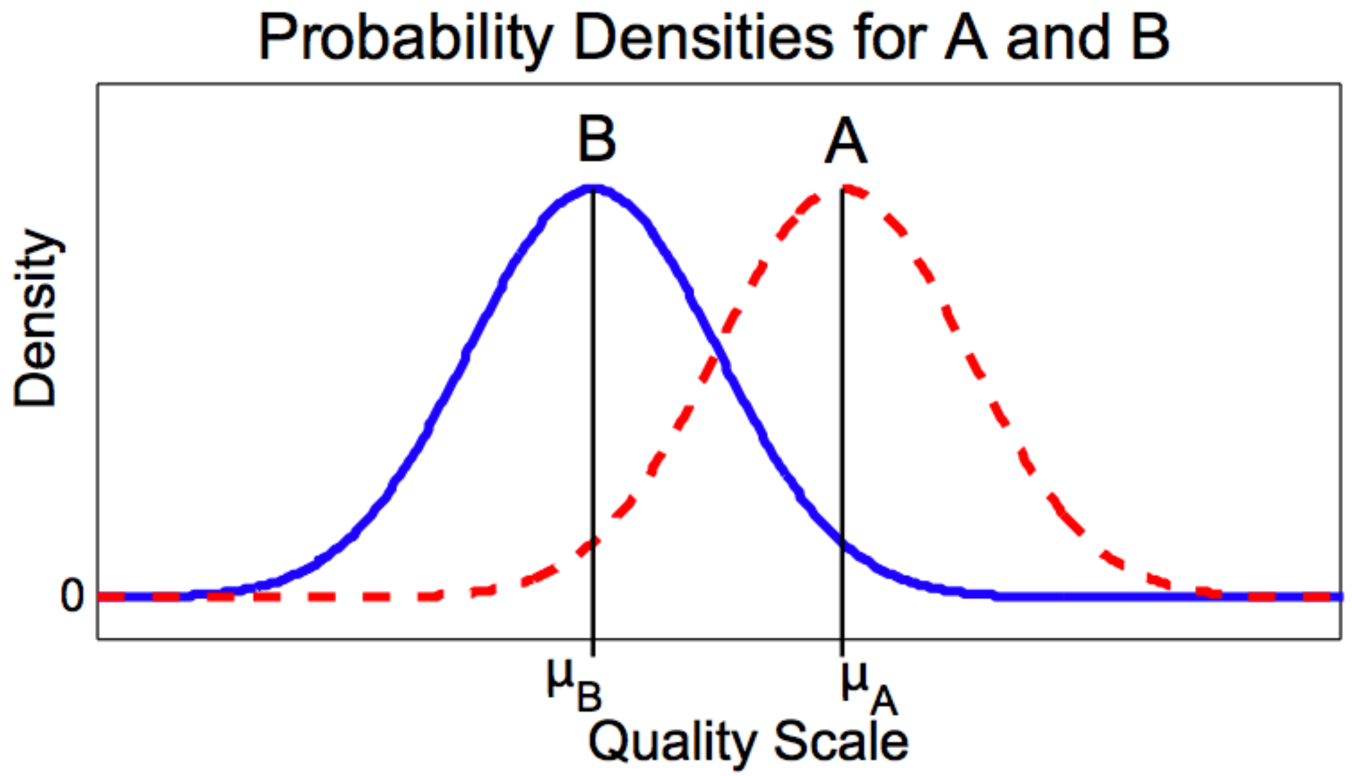


BTL

$$P(A > B) = \frac{\pi_A}{\pi_A + \pi_B}$$

Embeddings and Orderings 1: Discrete choice models

Bradley-Terry & Luce: items have quality distributions. When a person judges whether A is better than B they draw from A's and from B's distribution and see which is higher.



BTL

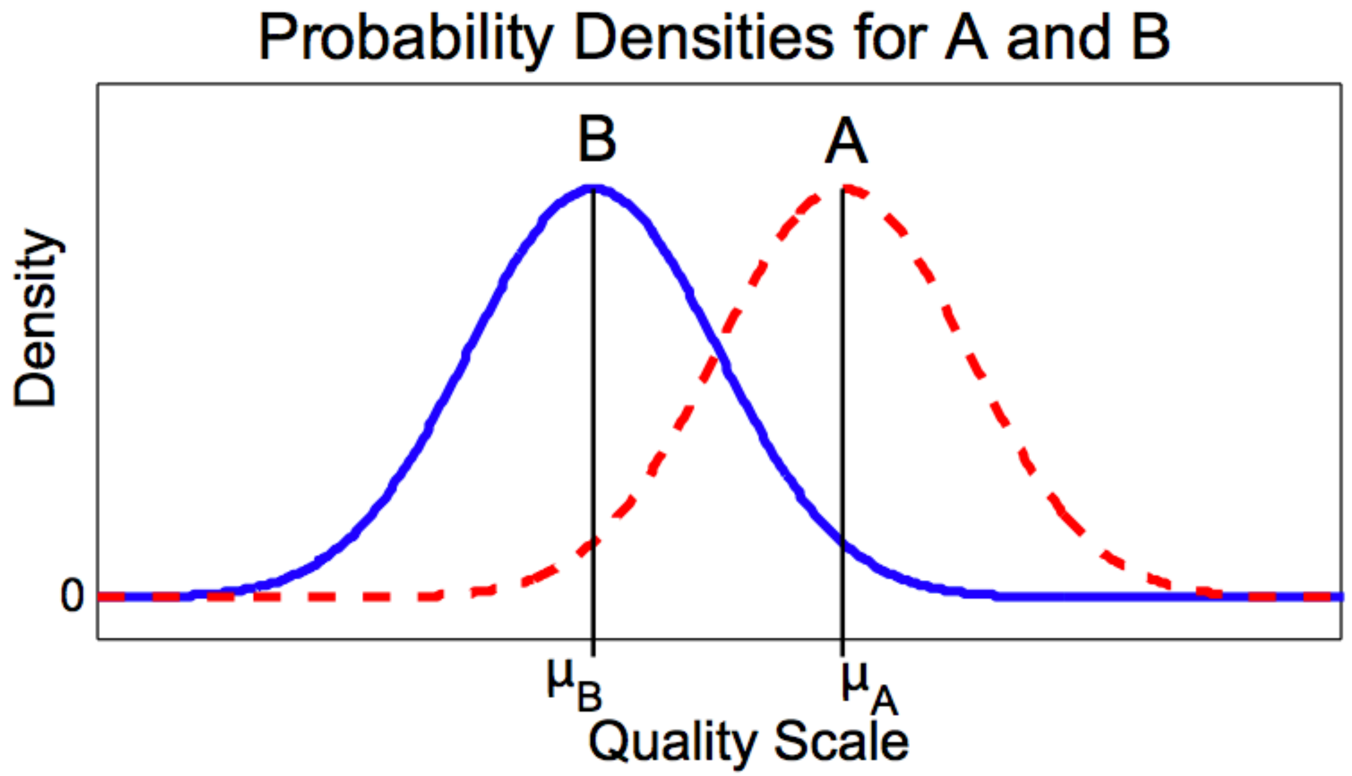
$$P(A > B) = \frac{\pi_A}{\pi_A + \pi_B}$$

Or usually:

$$P(A > B) = \frac{e^{\mu_A/s}}{e^{\mu_A/s} + e^{\mu_B/s}} = \frac{1}{1 + e^{-(\mu_A - \mu_B)/s}}$$

Embeddings and Orderings 1: Discrete choice models

Bradley-Terry & Luce: items have quality distributions. When a person judges whether A is better than B they draw from A's and from B's distribution and see which is higher.



BTL

$$P(A > B) = \frac{\pi_A}{\pi_A + \pi_B}$$

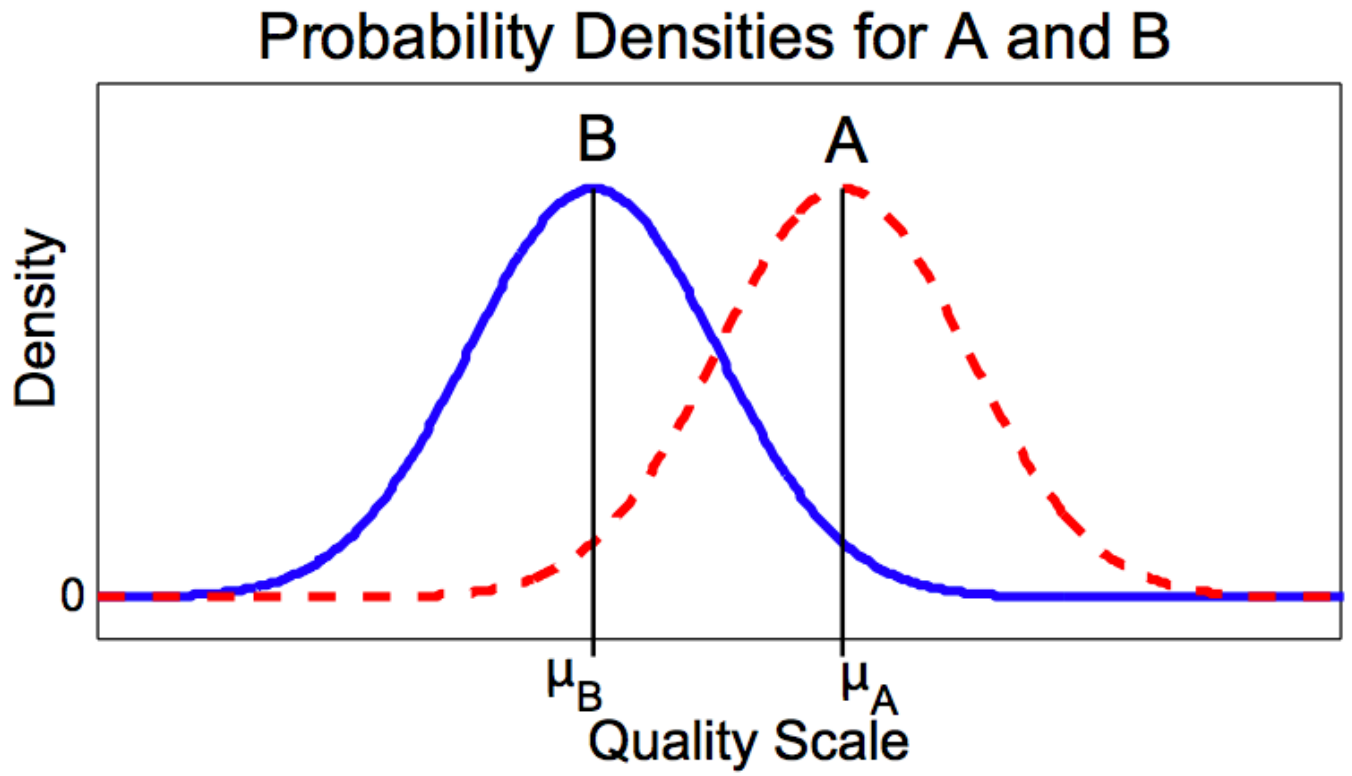
Or usually:

$$P(A > B) = \frac{e^{\mu_A/s}}{e^{\mu_A/s} + e^{\mu_B/s}} = \frac{1}{1 + e^{-(\mu_A - \mu_B)/s}}$$

Same idea; different distribution. (*logit* instead of *probit*; *Gumbel* instead of *Gaussian*)

Embeddings and Orderings 1: Discrete choice models

Bradley-Terry & Luce: items have quality distributions. When a person judges whether A is better than B they draw from A's and from B's distribution and see which is higher.



BTL

$$P(A > B) = \frac{\pi_A}{\pi_A + \pi_B}$$

Or usually:

$$P(A > B) = \frac{e^{\mu_A/s}}{e^{\mu_A/s} + e^{\mu_B/s}} = \frac{1}{1 + e^{-(\mu_A - \mu_B)/s}}$$

Same idea; different distribution. (*logit* instead of *probit*; *Gumbel* instead of *Gaussian*)

Powerful idea: lots of pairwise comparisons = estimates of all the qualities! An embedding!

BTL avoids non-transitivities (aka rock-paper-scissors)

BTL avoids non-transitivities (aka rock-paper-scissors)

Introducing: **non-transitive dice**!

BTL avoids non-transitivities (aka rock-paper-scissors)

Introducing: **non-transitive dice**!

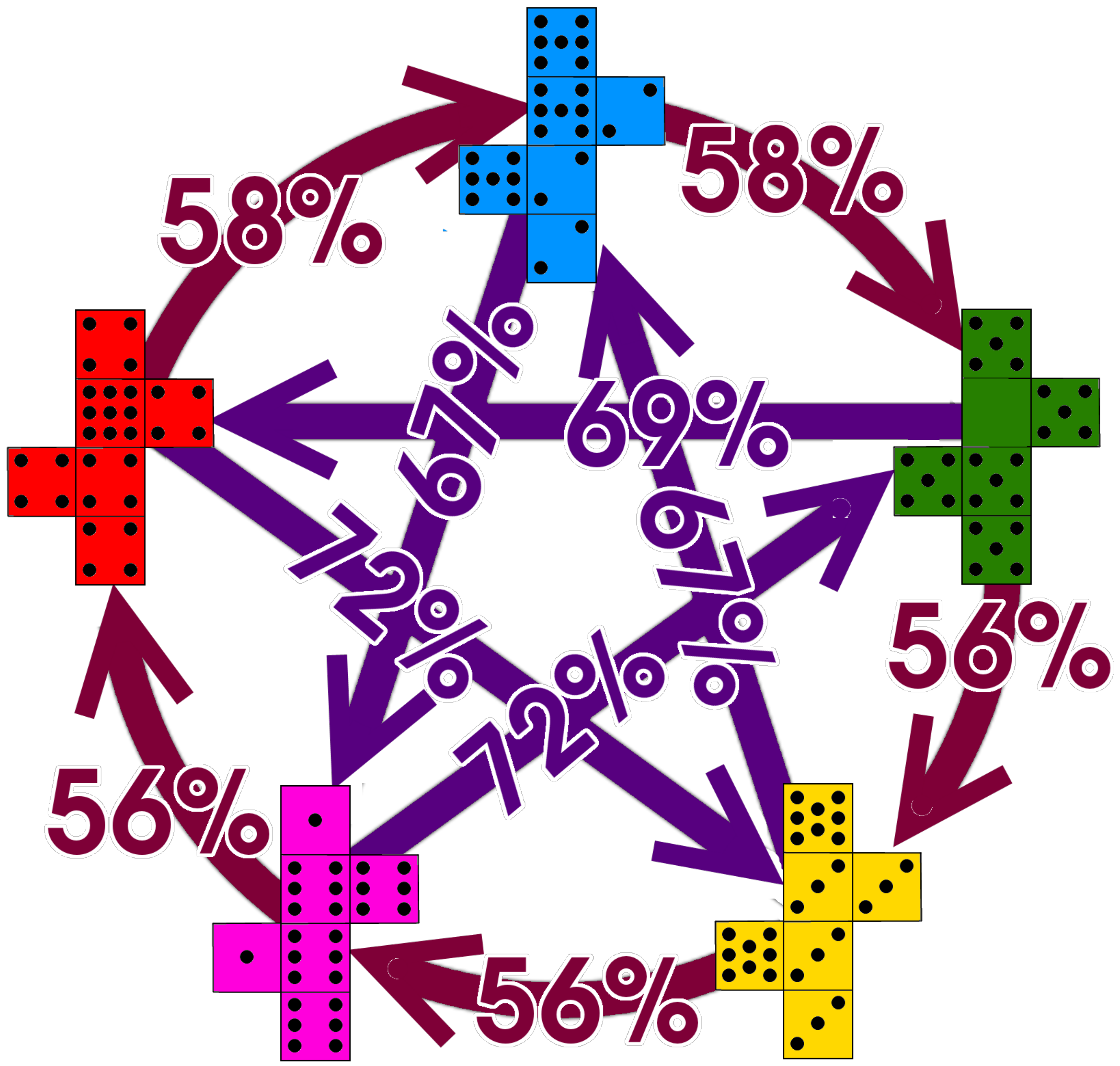
- 3 (or more) dice {A,B,C}
- faces chosen so that they have the property:
    - A>B more than half the time.
    - B>C more than half the time.
    - C>A more than half the time (?!)

BTL avoids non-transitivities (aka rock-paper-scissors)

Introducing: **non-transitive dice**!

- 3 (or more) dice {A,B,C}
- faces chosen so that they have the property:
  - A>B more than half the time.
  - B>C more than half the time.
  - C>A more than half the time (?!)

https://en.wikipedia.org/wiki/Nontransitive_dice

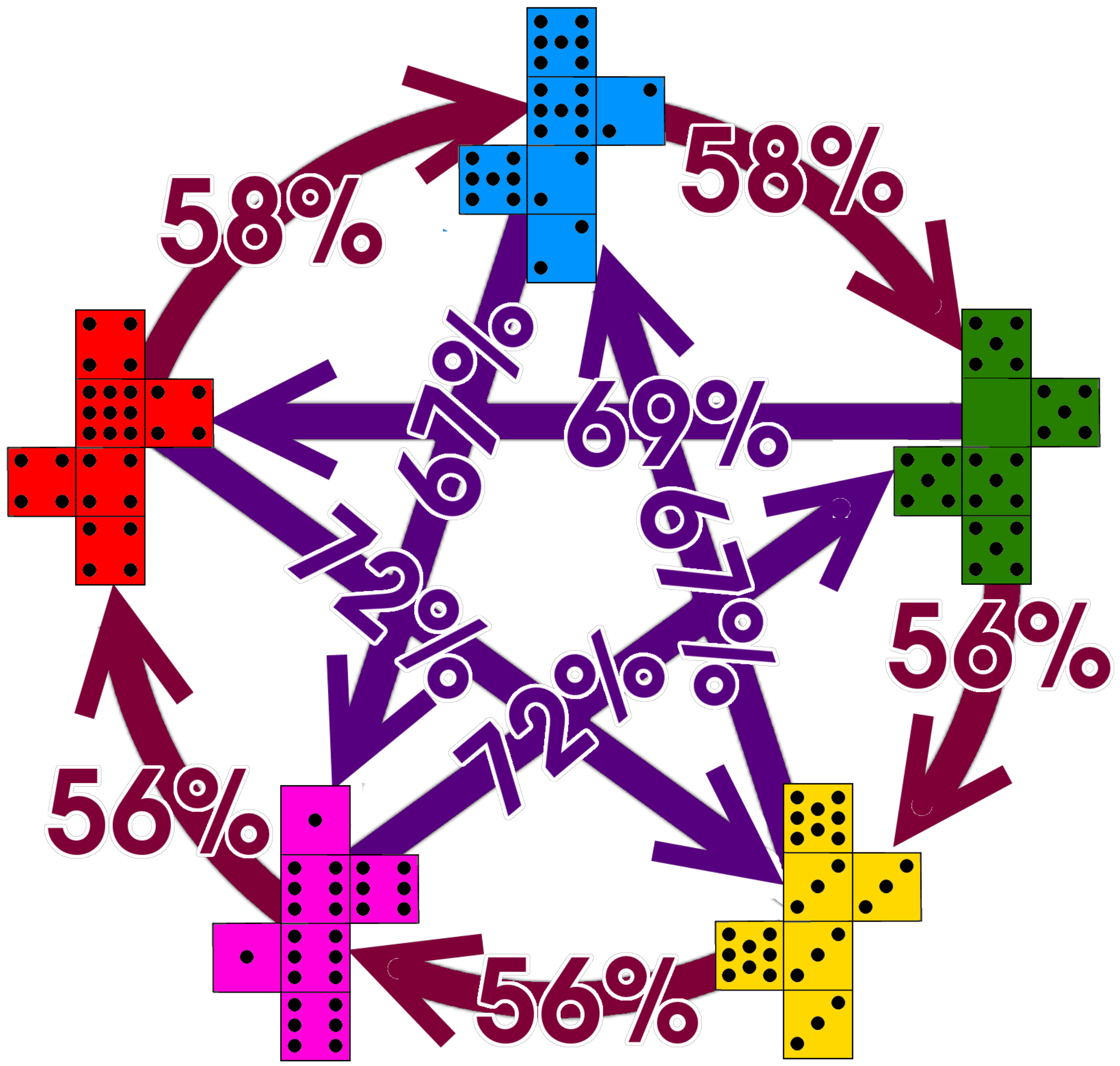


BTL avoids non-transitivities (aka rock-paper-scissors)

Introducing: **non-transitive dice**!

- 3 (or more) dice {A,B,C}
- faces chosen so that they have the property:
  - A>B more than half the time.
  - B>C more than half the time.
  - C>A more than half the time (?!)

https://en.wikipedia.org/wiki/Nontransitive_dice



A great gift for your favorite nerd's desk!
Go to the makerspace and laserbeam your own!

Bradley-Terry-Luce

These methods embed items or players in a 1D space.

- Provably avoids non-transitive properties
- Great when lots of data per interaction.

Pairwise ranking is really nice for ordering large sets of preferences too, and this model specifically models the probability that the preference will be for *i* over *j*.

Iterative algorithms exist. Needs a little regularization so the winningest winners don't fly off to infinity.

$$P(i \to j) = \frac{\gamma_i}{\gamma_i + \gamma_j}$$

Embeddings and Orderings 1: Discrete choice models

Introductory tutorial:

http://mayagupta.org/publications/PairedComparisonTutorialTsukidaGupta.pdf

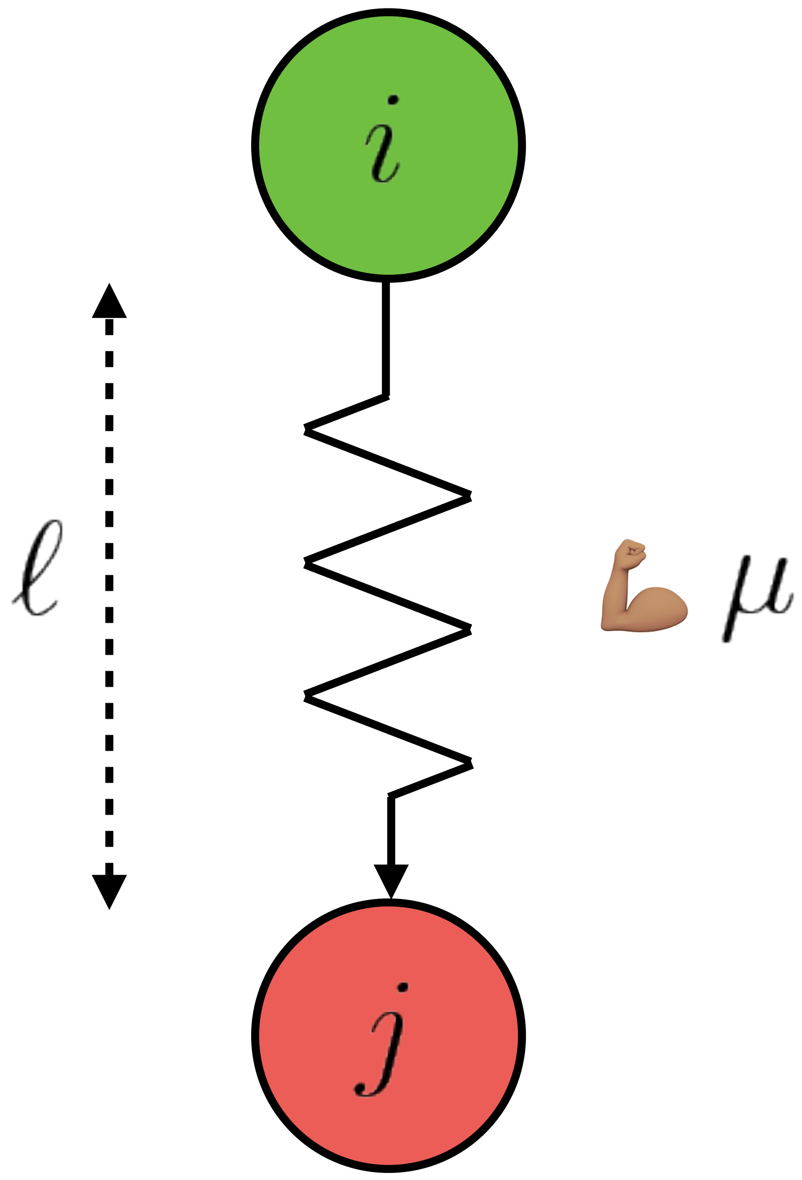
Discrete choice today:

https://web.stanford.edu/~jugander/papers/nips16-pcmc-slides.pdf

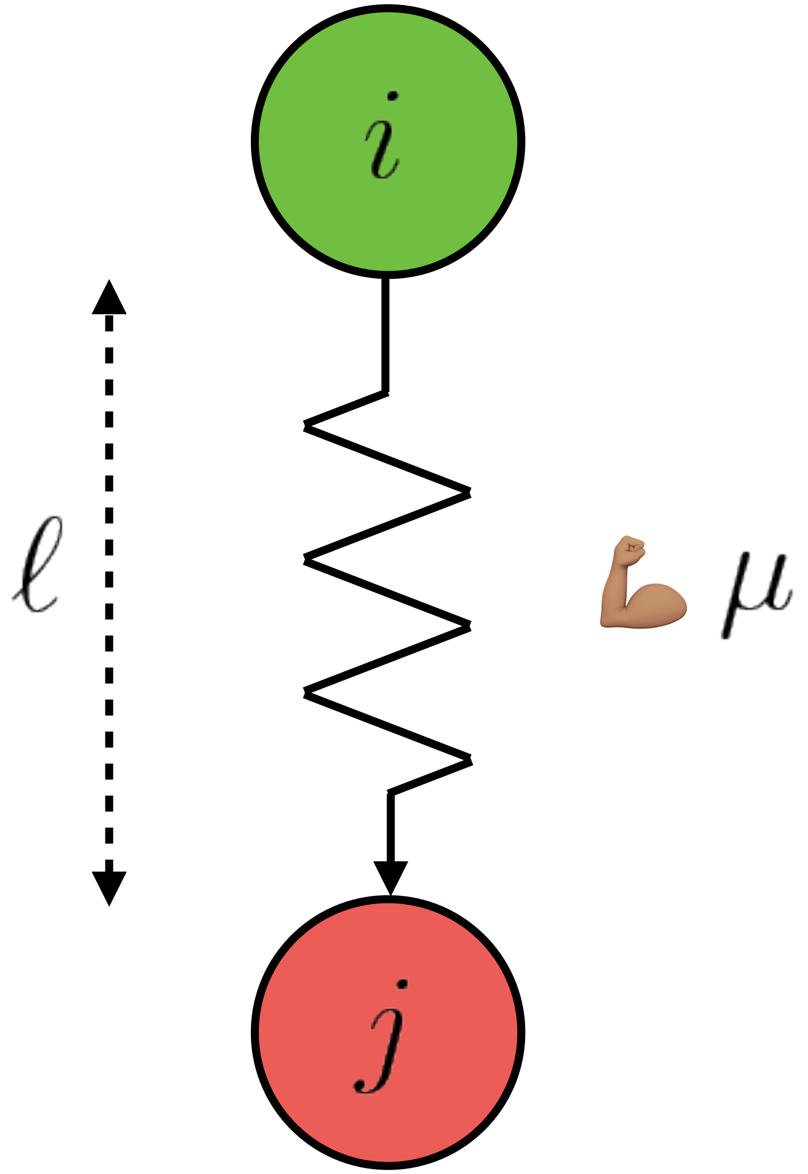
Embeddings & Orderings 2: SpringRank

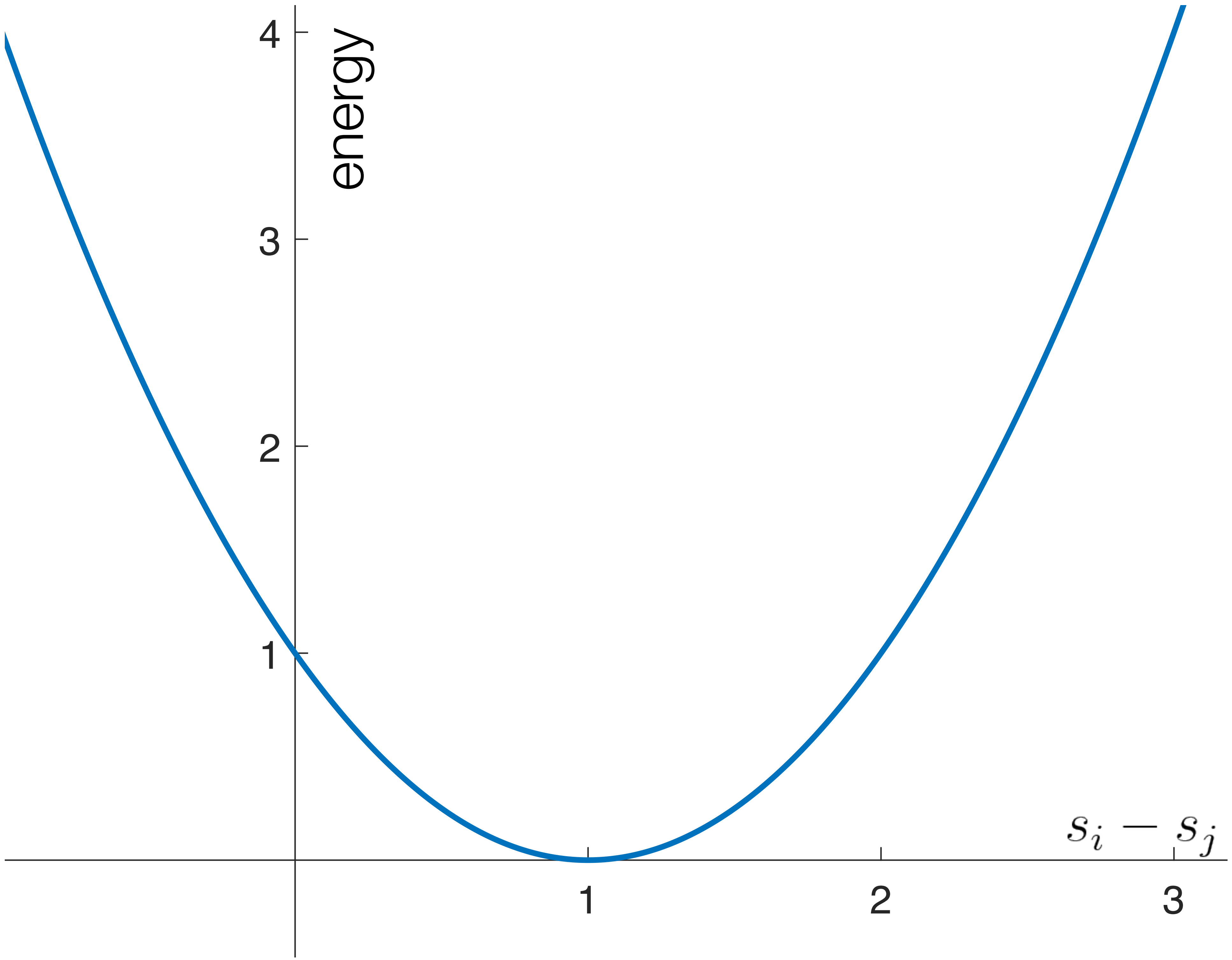
Each directed edge = directed spring

Each directed edge = directed spring

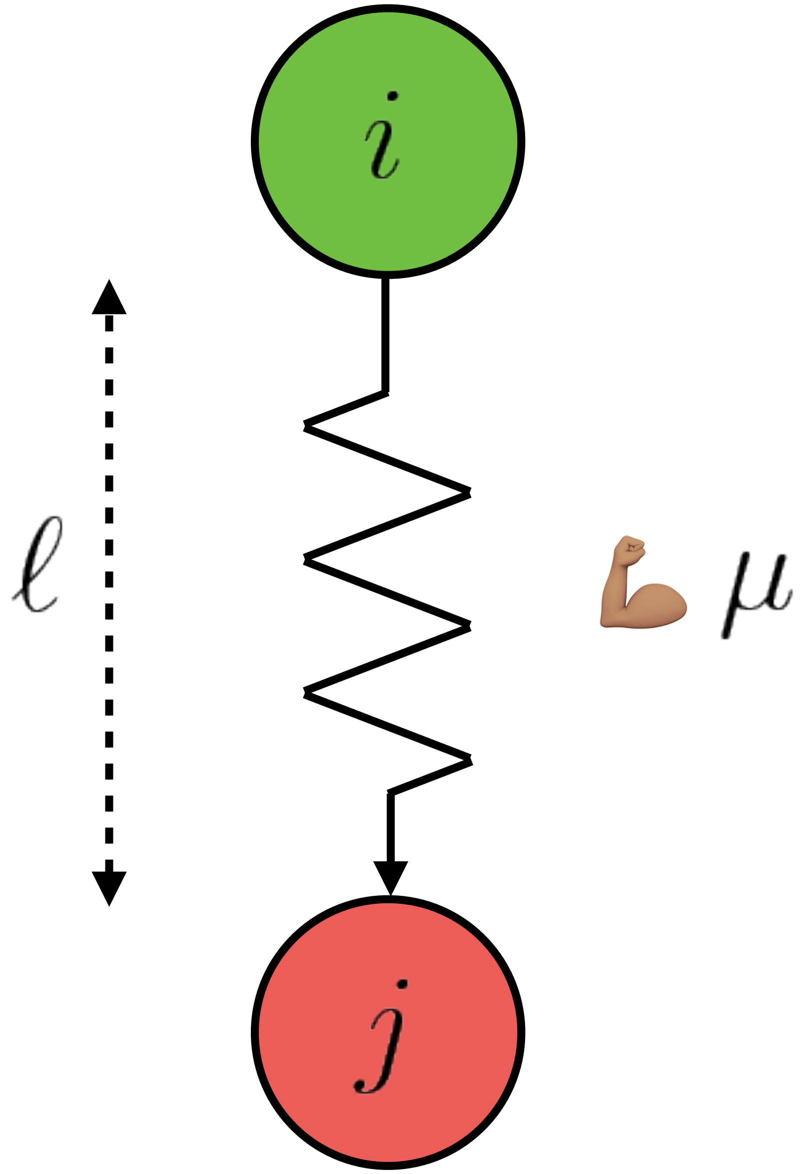


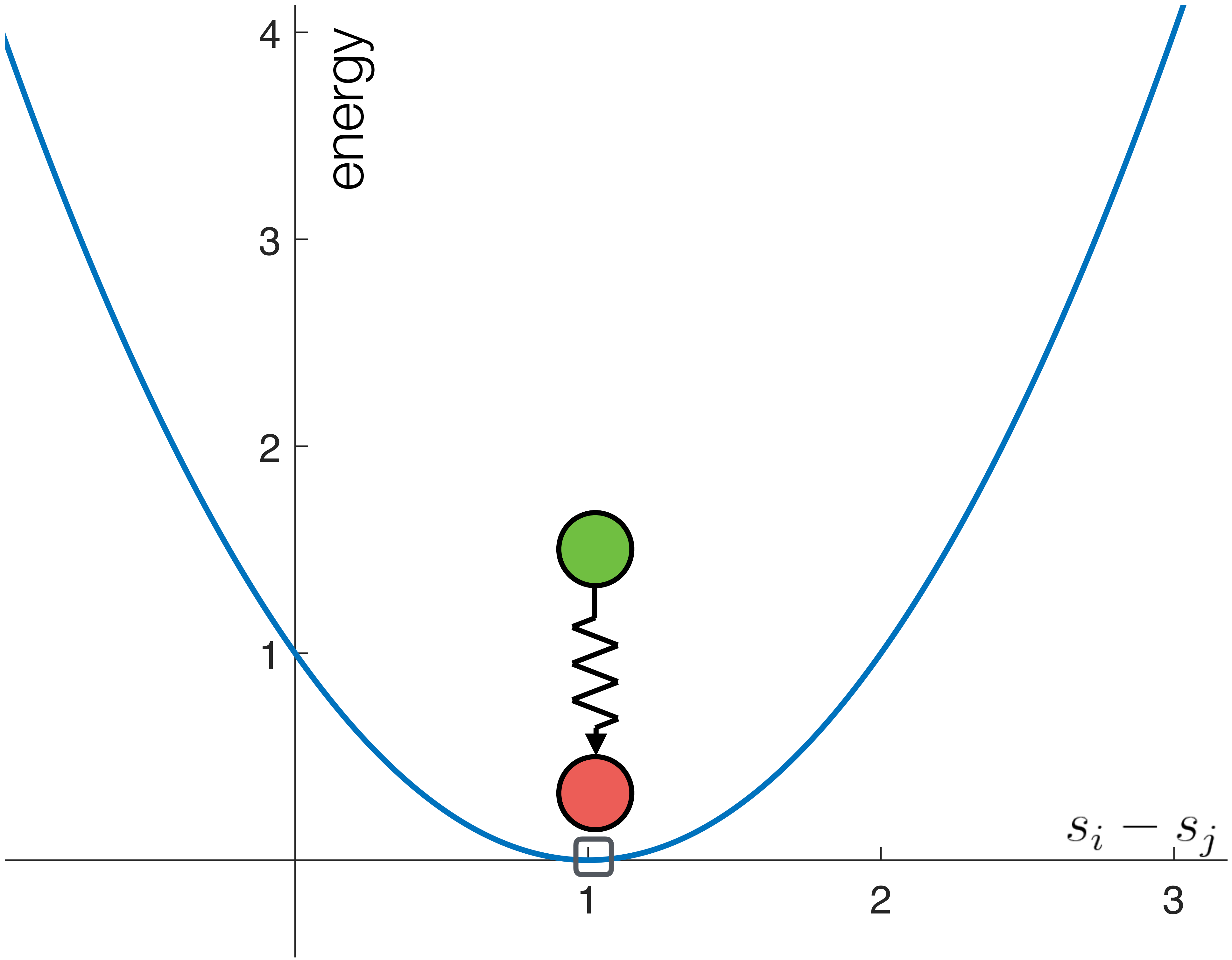
Each directed edge = directed spring



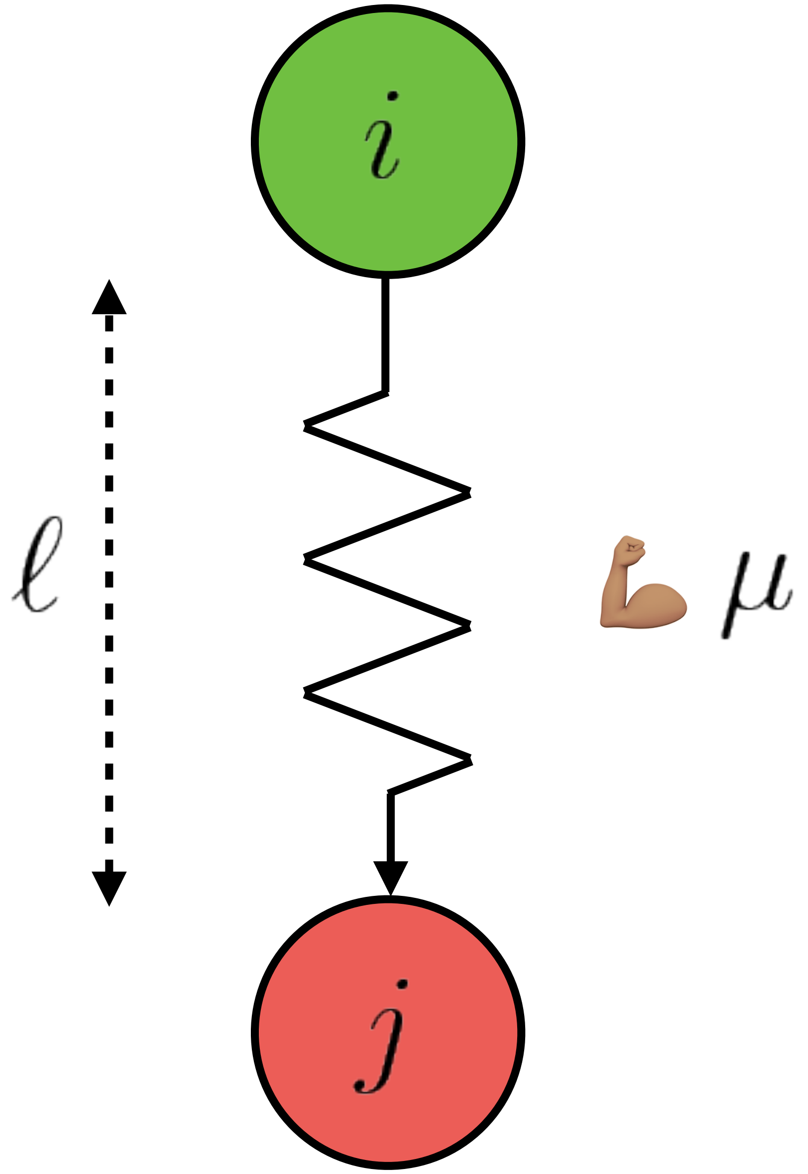


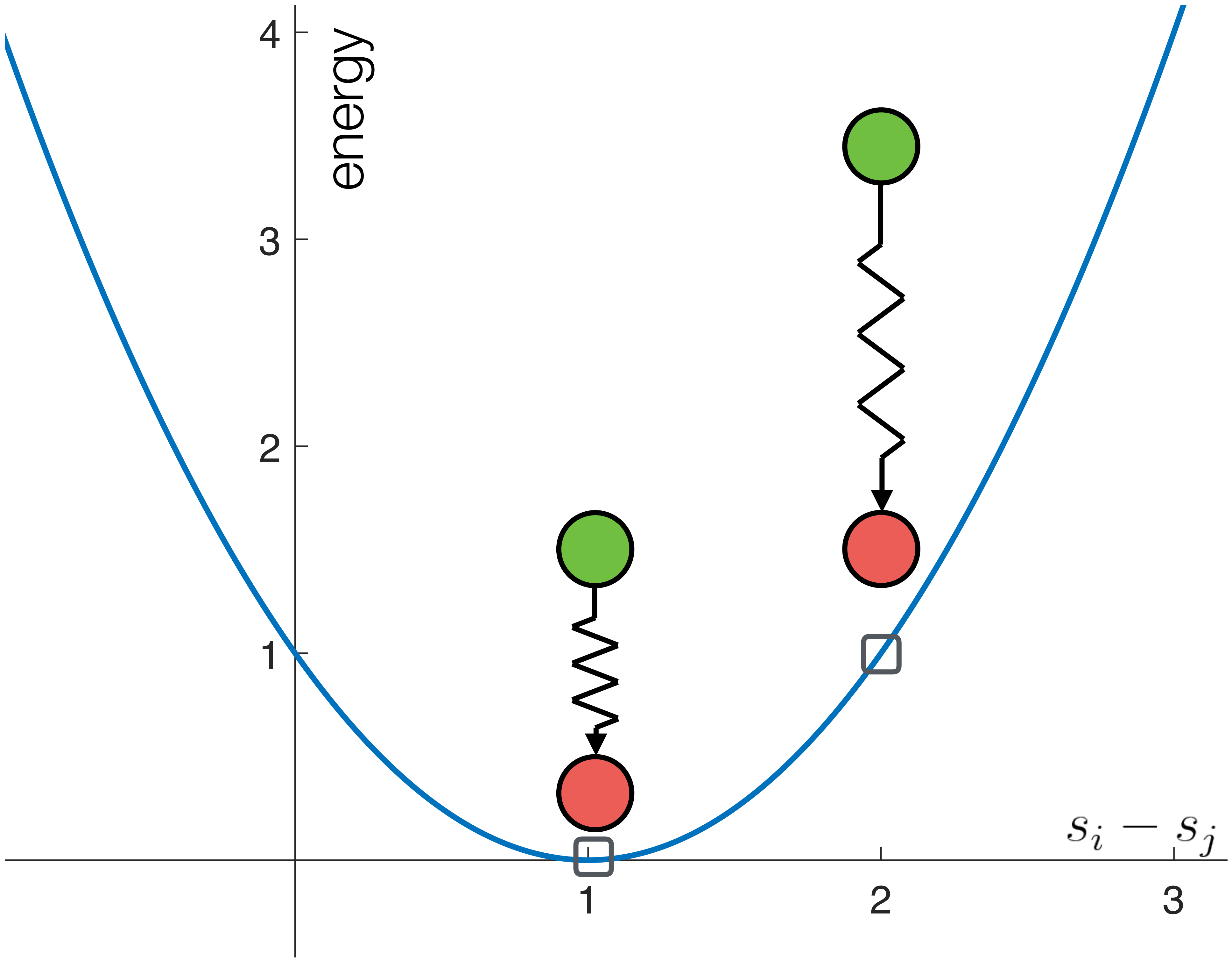
Each directed edge = directed spring



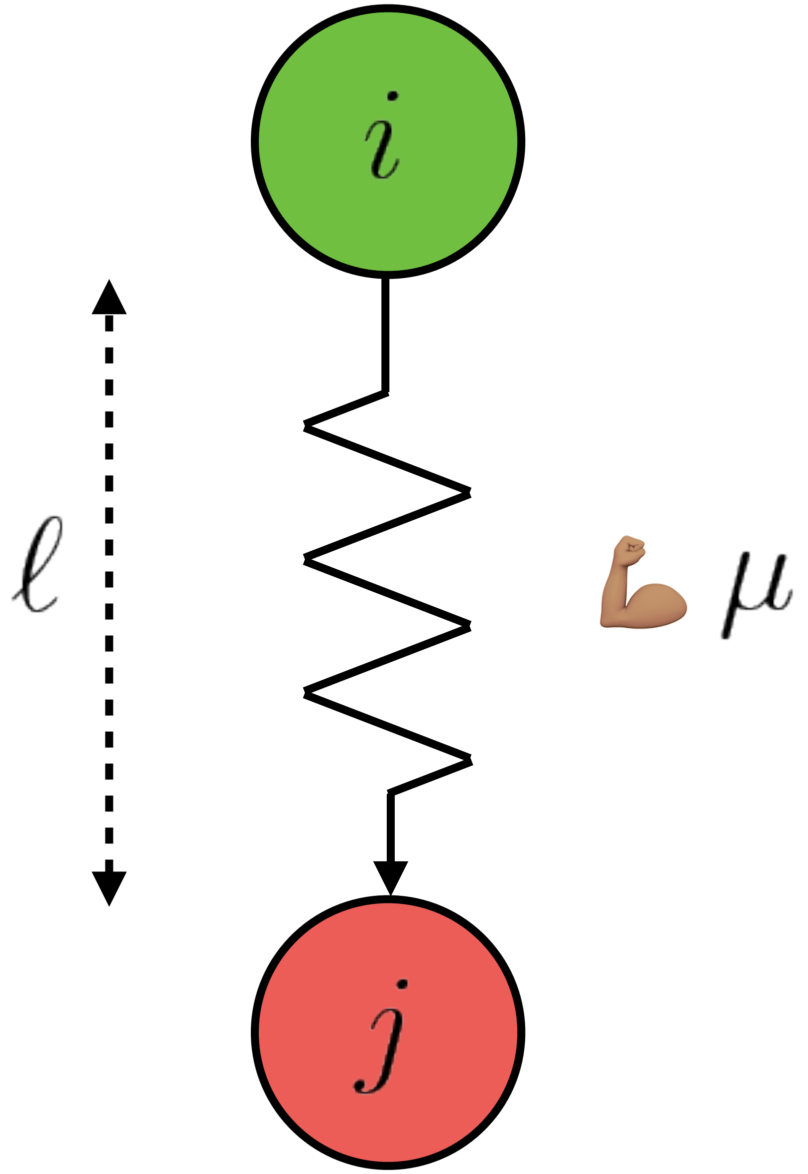


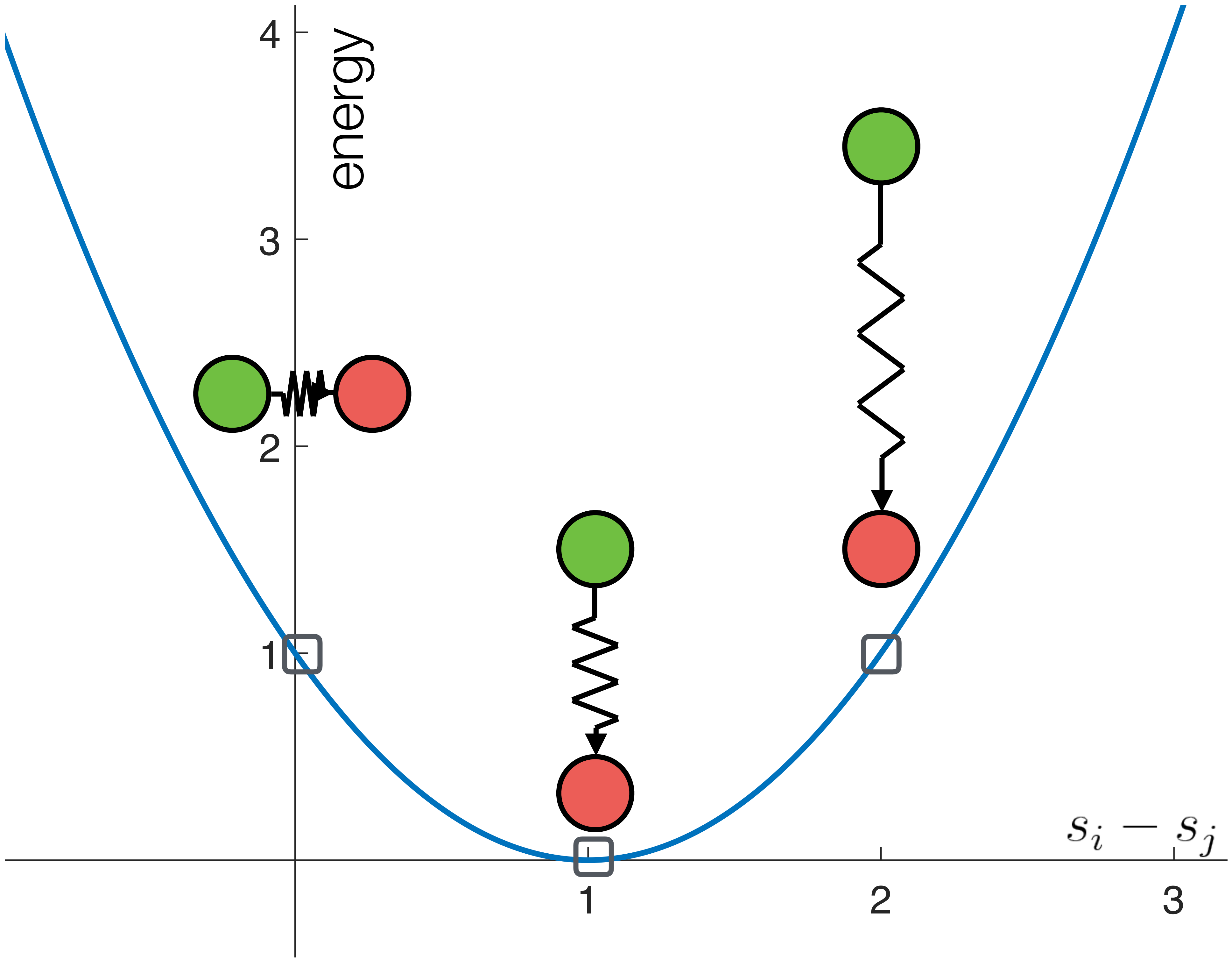
Each directed edge = directed spring



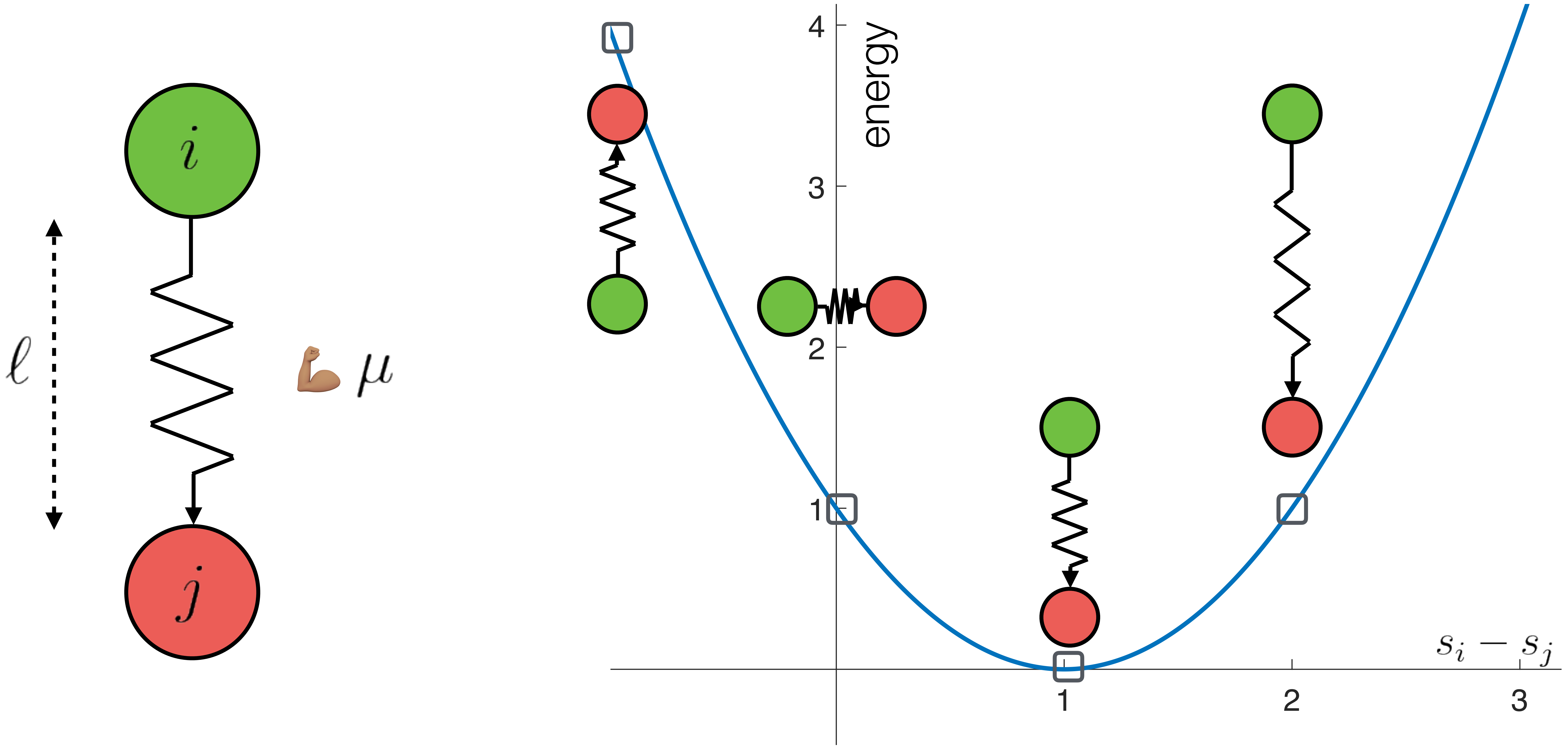


Each directed edge = directed spring





Each directed edge = directed spring



How much energy is this system of springs?

How much energy is this system of springs?

$$(s_i - s_j - 1)^2$$

How much energy is this system of springs?

$$A_{ij}(s_i - s_j - 1)^2$$

How much energy is this system of springs?

$$\sum_{i,j=1}^{N} A_{ij}(s_i - s_j - 1)^2$$

How much energy is this system of springs?

$$H(s) = \frac{1}{2} \sum_{i,j=1}^{N} A_{ij}(s_i - s_j - 1)^2$$

Relax and let the springs decide the ranks

$$H(s) = \frac{1}{2} \sum_{i,j=1}^{N} A_{ij}(s_i - s_j - 1)^2$$

SpringRank Hamiltonian = energy of the system, given the node positions *s*.

Because the springs are linear, the potential is quadratic.

Relax and let the springs decide the ranks

$$H(s) = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij}(s_i - s_j - 1)^2$$

SpringRank Hamiltonian = energy of the system, given the node positions *s.*

Because the springs are linear, the potential is quadratic.

The SR Hamiltonian is *convex* in s.

$$\nabla H(s) = 0$$

The solution is unique…up to an additive constant. (Why?)

Derivatives work out nicely

$$0 = \frac{\partial H}{\partial s_i} = \sum_j A_{ij}(s_i - s_j - 1) - A_{ji}(s_j - s_i - 1)$$

Derivatives work out nicely

$$0 = \frac{\partial H}{\partial s_i} = \sum_j A_{ij}(s_i - s_j - 1) - A_{ji}(s_j - s_i - 1)$$

Rewrite as a linear algebra problem.

$$\left[D^{\text{out}} + D^{\text{in}} - \left(A + A^T\right)\right] s^* = \left[D^{\text{out}} - D^{\text{in}}\right] \mathbf{1}$$

Derivatives work out nicely

$$0 = \frac{\partial H}{\partial s_i} = \sum_j A_{ij}(s_i - s_j - 1) - A_{ji}(s_j - s_i - 1)$$

Rewrite as a linear algebra problem.

$$\left[D^{\text{out}} + D^{\text{in}} - \left(A + A^T\right)\right] s^* = \left[D^{\text{out}} - D^{\text{in}}\right] \mathbf{1}$$

We know *a priori* that the matrix on the left is singular: translational invariance of H(s).

Derivatives work out nicely

$$0 = \frac{\partial H}{\partial s_i} = \sum_j A_{ij}(s_i - s_j - 1) - A_{ji}(s_j - s_i - 1)$$

Rewrite as a linear algebra problem.

$$\left[D^{\text{out}} + D^{\text{in}} - \left(A + A^T\right)\right] s^* = \left[D^{\text{out}} - D^{\text{in}}\right] \mathbf{1}$$

We know *a priori* that the matrix on the left is singular: translational invariance of H(s).
[if *s* is a solution, then *s* + *k* is a solution for any constant *k;* eigenvalue 0, eigenvector **1**]

Derivatives work out nicely

$$0 = \frac{\partial H}{\partial s_i} = \sum_j A_{ij}(s_i - s_j - 1) - A_{ji}(s_j - s_i - 1)$$

Rewrite as a linear algebra problem.

$$\left[D^{\text{out}} + D^{\text{in}} - \left(A + A^T\right)\right] s^* = \left[D^{\text{out}} - D^{\text{in}}\right] \mathbf{1}$$

We know *a priori* that the matrix on the left is singular: translational invariance of H(s).
[if *s* is a solution, then *s* + *k* is a solution for any constant *k;* eigenvalue 0, eigenvector **1**]

Notice: the matrix on the left is the *graph Laplacian* of the *undirected* network.

Derivatives work out nicely

$$0 = \frac{\partial H}{\partial s_i} = \sum_j A_{ij}(s_i - s_j - 1) - A_{ji}(s_j - s_i - 1)$$

Rewrite as a linear algebra problem.

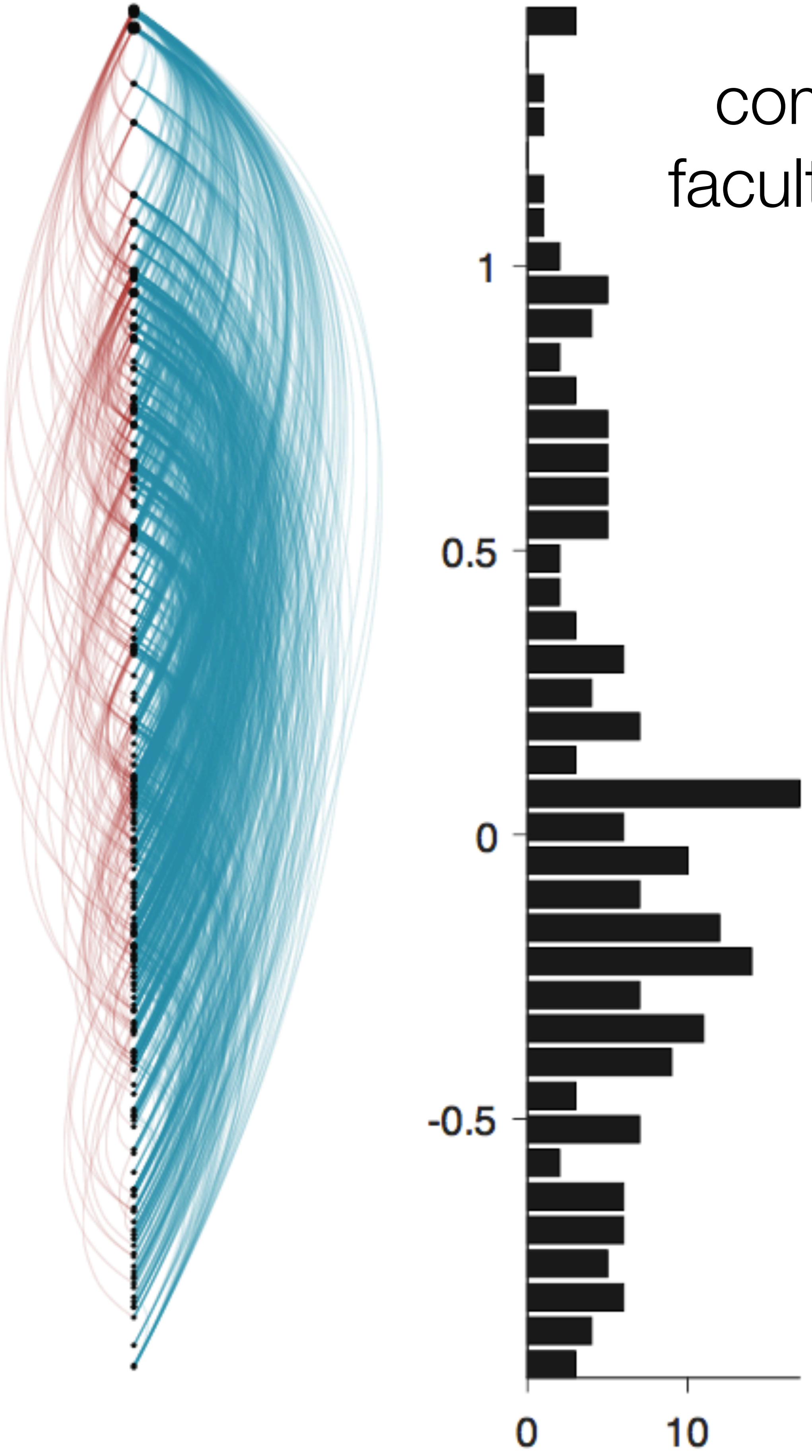
$$\left[D^{\text{out}} + D^{\text{in}} - \left(A + A^T\right)\right] s^* = \left[D^{\text{out}} - D^{\text{in}}\right] \mathbf{1}$$

We know *a priori* that the matrix on the left is singular: translational invariance of H(s).
[if *s* is a solution, then *s* + *k* is a solution for any constant *k;* eigenvalue 0, eigenvector **1**]

Notice: the matrix on the left is the *graph Laplacian* of the *undirected* network.

Uniqueness: Set s_1=0, min(*s*)=0, or mean(*s*)=0. Or use a pseudoinverse. Or regularize.

It works!

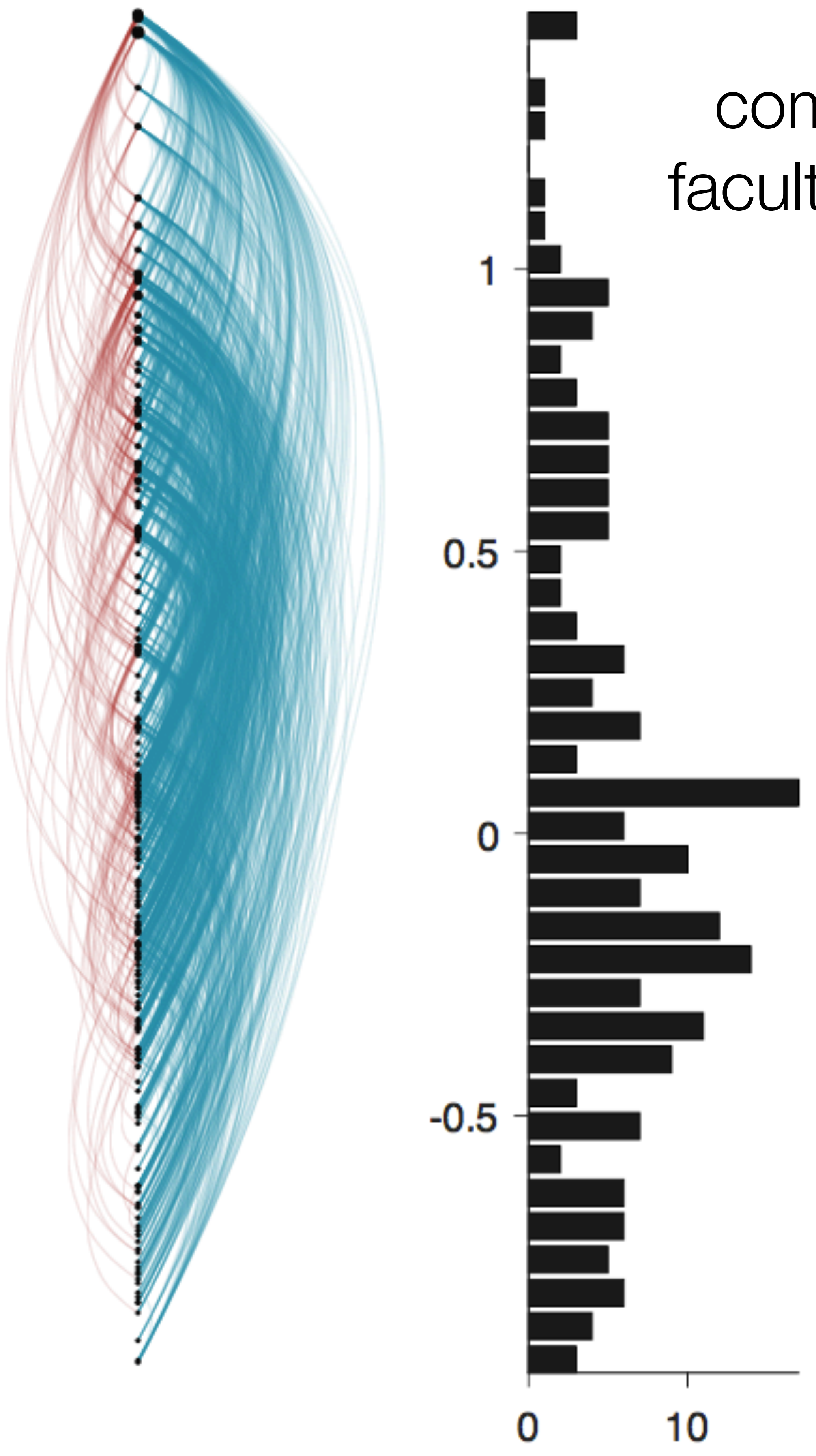


computer science
faculty hiring network

It works!

Real networks tend to be sparse…
our linear algebra problem is sparse…
we can use sparse iterative solvers…
millions of edges in **seconds**.

Even better: it's a linear-Laplacian system.
🚀 Near-linear-time (in |edges|) solutions.



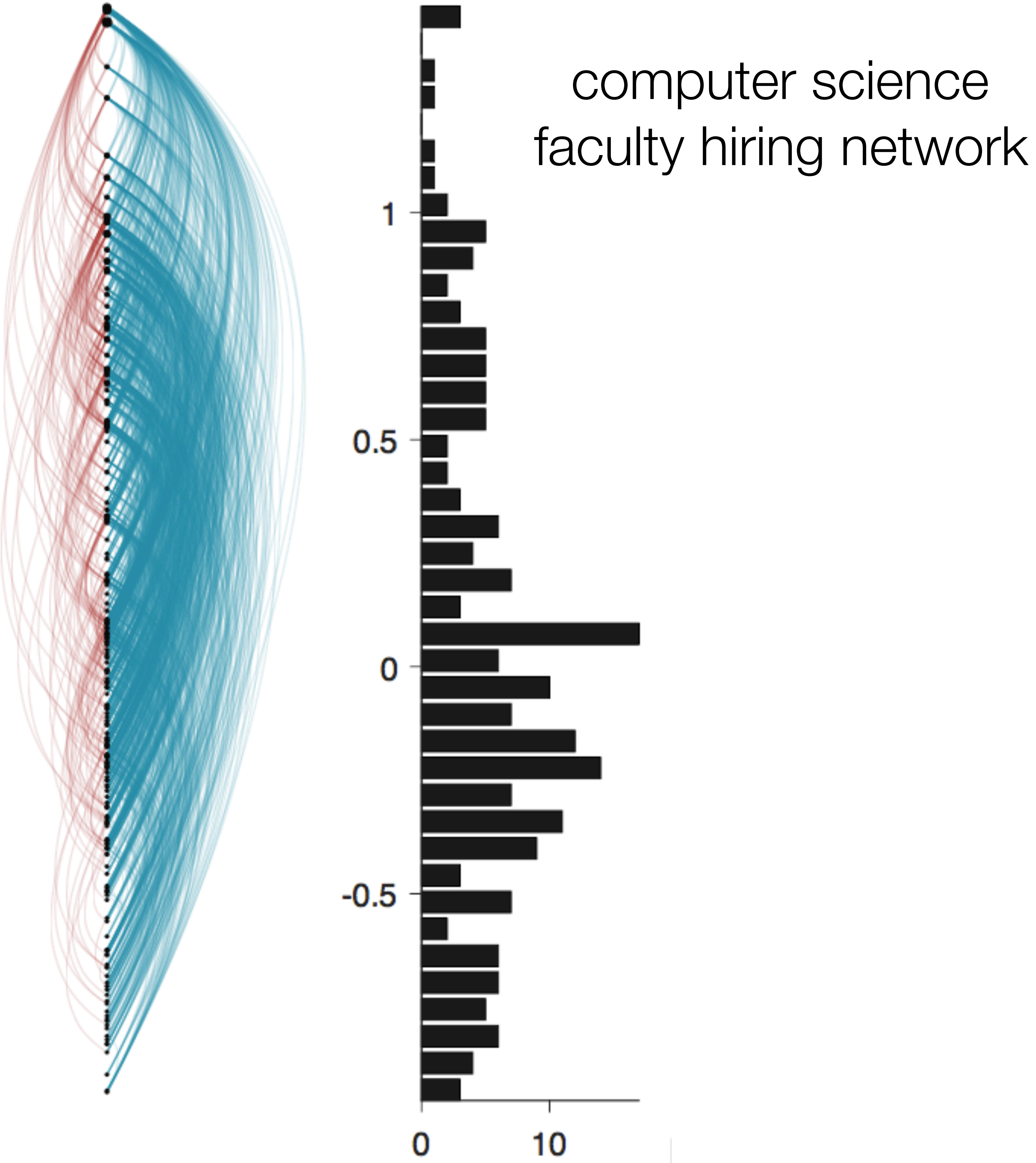
computer science
faculty hiring network

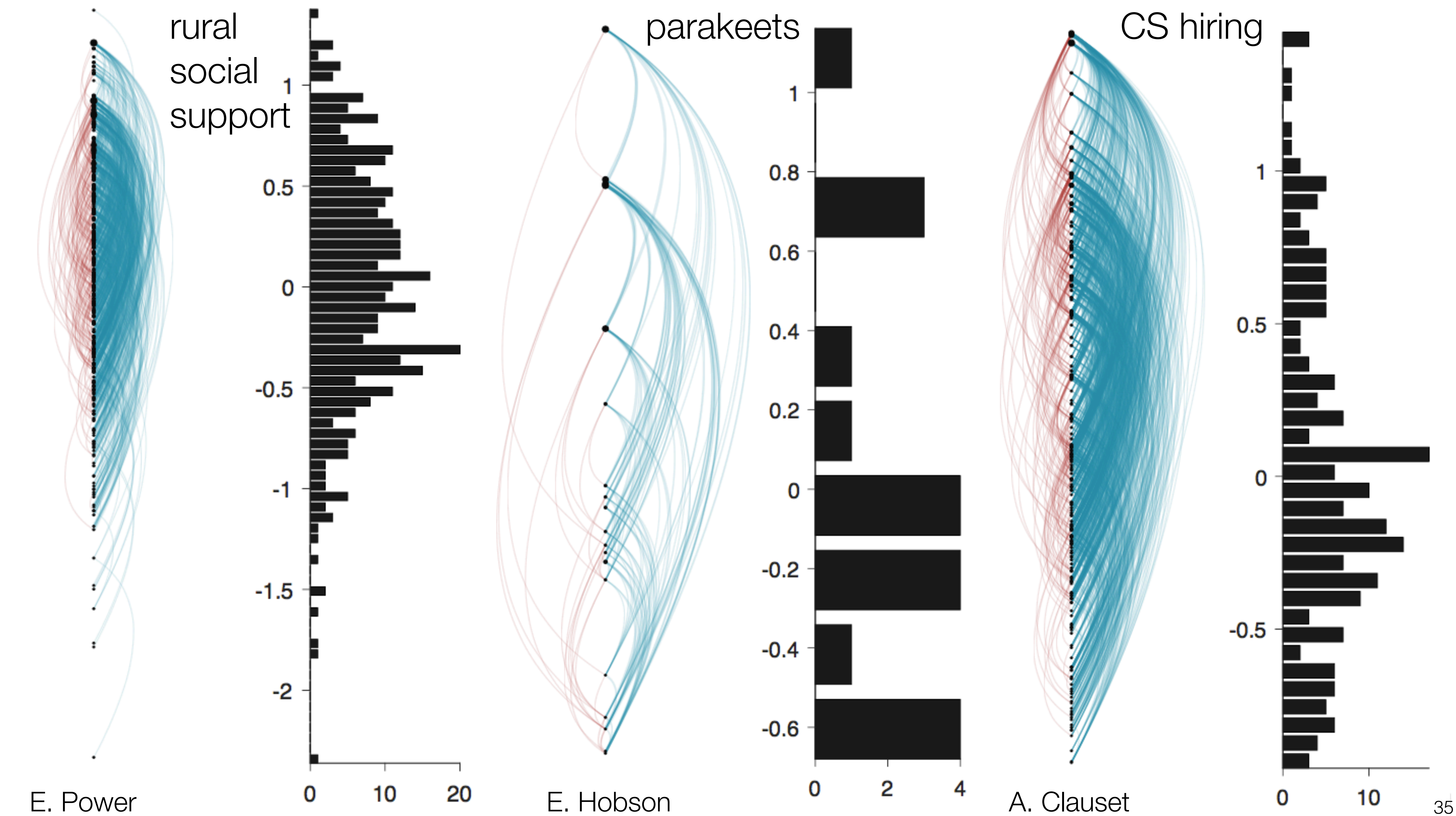
It works!

Real networks tend to be sparse…
our linear algebra problem is sparse…
we can use sparse iterative solvers…
millions of edges in **seconds**.

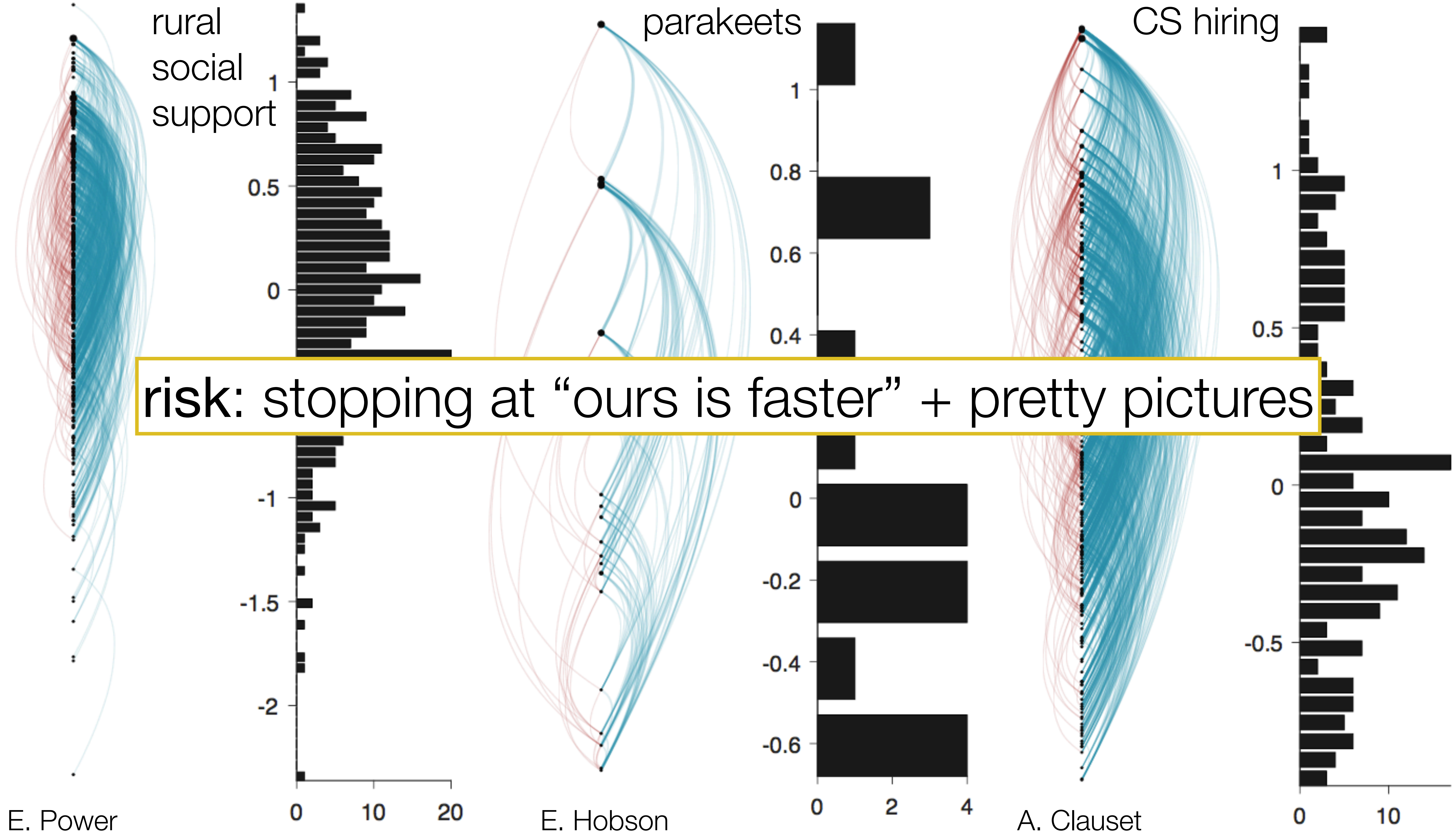
Even better: it's a linear-Laplacian system.
🚀 Near-linear-time (in |edges|) solutions.

Note that node positions can be clumpy, since this is an *embedding*.





rural
social
support
1
0.5
0
-0.5
-1
-1.5
-2
0
10
20
E. Power
parakeets
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
2
4
E. Hobson
CS hiring
1
0.5
0
-0.5
0
10
A. Clauset



risk: stopping at “ours is faster” + pretty pictures

Cross validation: train on 80%, predict 20%

In a linear hierarchy the key quantity to predict is *edge direction,* given *edge existence*.

If *i* and *j* were to face off, who would win?

Cross validation: train on 80%, predict 20%

In a linear hierarchy the key quantity to predict is *edge direction,* given *edge existence*.

If *i* and *j* were to face off, who would win?

I'll give you *undirected(A)*, and you predict *directed(A).*

Cross validation: train on 80%, predict 20%

In a linear hierarchy the key quantity to predict is *edge direction,* given *edge existence*.

If *i* and *j* were to face off, who would win?

I'll give you *undirected(A)*, and you predict *directed(A).*

Setup: learn *s* from 80% of *A*. Then predict edge directions for remaining 20% of *A.*

Cross validation: train on 80%, predict 20%

In a linear hierarchy the key quantity to predict is *edge direction,* given *edge existence*.

If *i* and *j* were to face off, who would win?

I'll give you *undirected(A)*, and you predict *directed(A).*

Setup: learn *s* from 80% of *A*. Then predict edge directions for remaining 20% of *A.*

SpringRank predicts edge direction based on the relative direction probabilities:

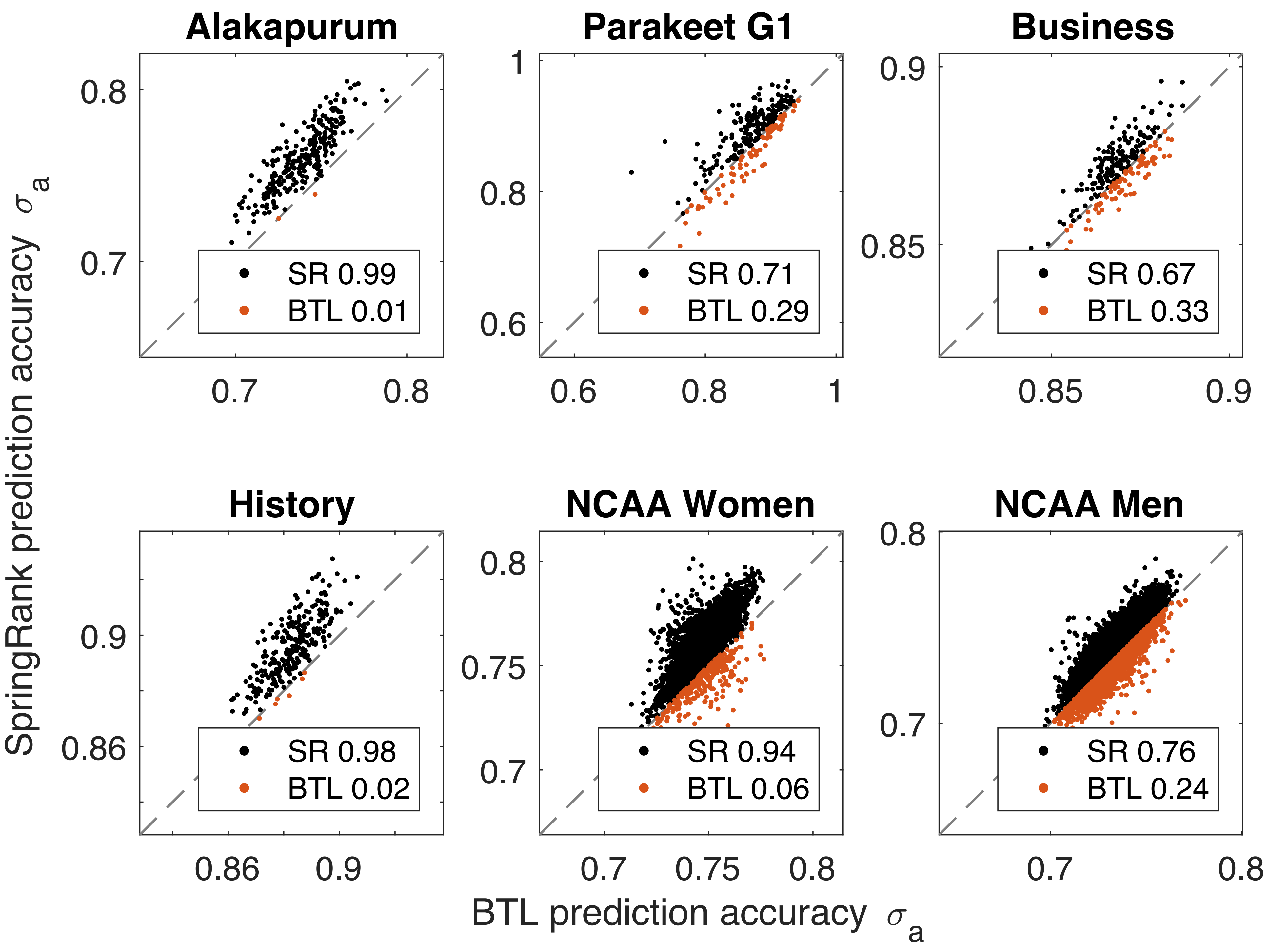
$$P_{ij}(\beta) = \frac{e^{-\beta H_{ij}}}{e^{-\beta H_{ij}} + e^{-\beta H_{ji}}} = \frac{1}{1 + e^{-2\beta(s_i - s_j)}}$$

Cross validation vs BTL: SR makes better predictions

Accuracy:

$$\sigma_a = 1 - \frac{1}{2M} \sum_{i,j} |A_{ij} - (A_{ij} + A_{ji})\, P_{ij}|$$

Goal: maximize the number of correctly predicted edge directions.



50 independent trials of 5-fold cross validation (250 folds)

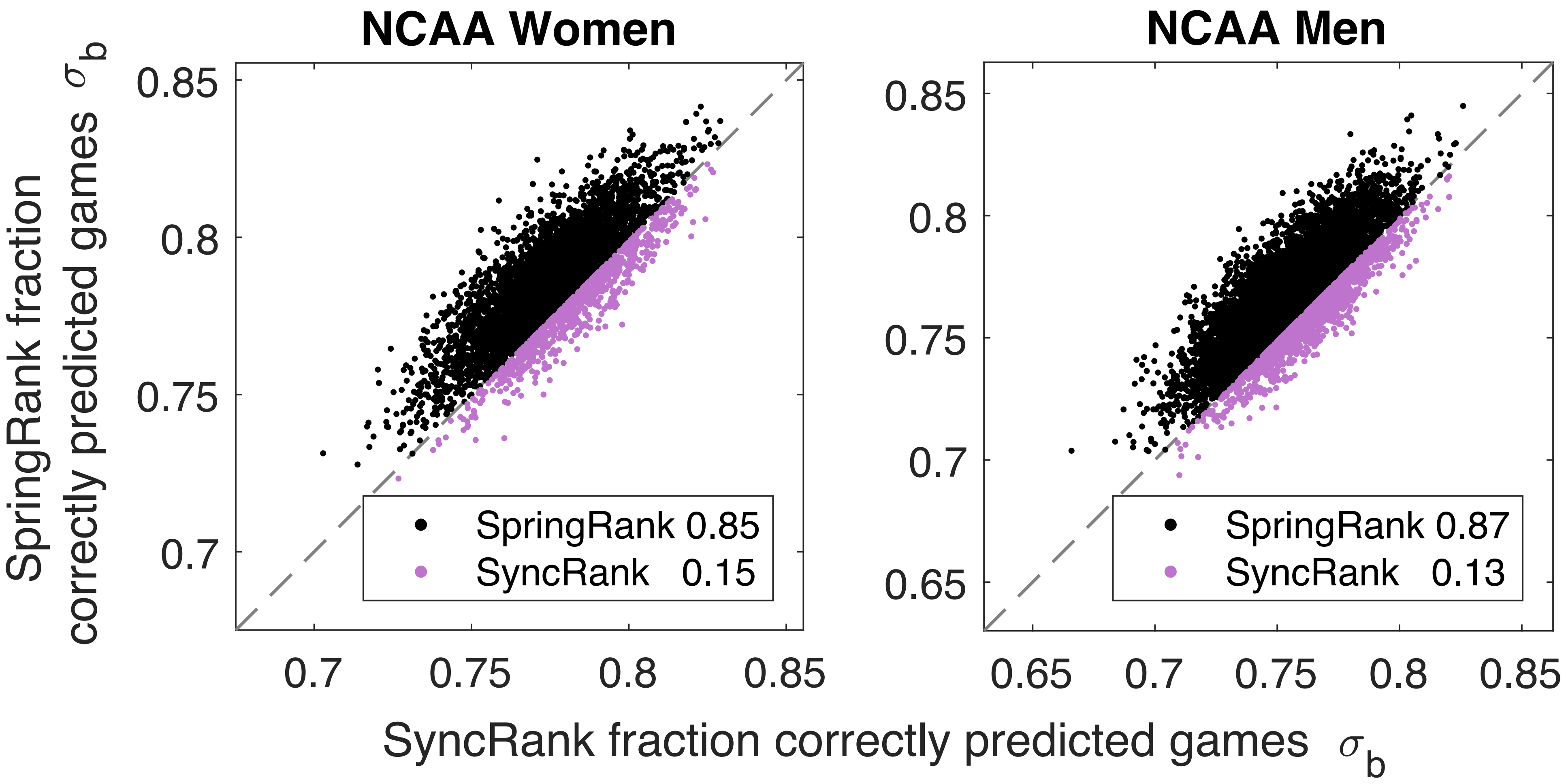
Cross validation vs SyncRank: SR makes better predictions

“One-bit” Accuracy:

Higher ranked player always wins.

- No probabilistic prediction.
- Bad for gambling.

Goal: maximize the number of correctly predicted edge directions.



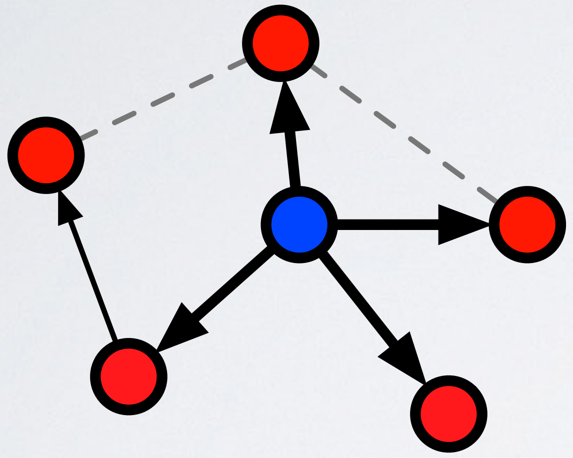
50 independent trials of 5-fold cross validation (250 folds)

Why/when would a model of springs make better predictions than a model of the choices themselves?🧐

Which nodes are important… we’ve heard this before!

Clauset lecture 2

describing networks



position = centrality:
structural vs. dynamical importance

geometric	harmonic centrality
	closeness centrality
	betweenness centrality
connectivity	degree centrality
	eigenvector centrality
	PageRank
	Katz centrality
	many many more…

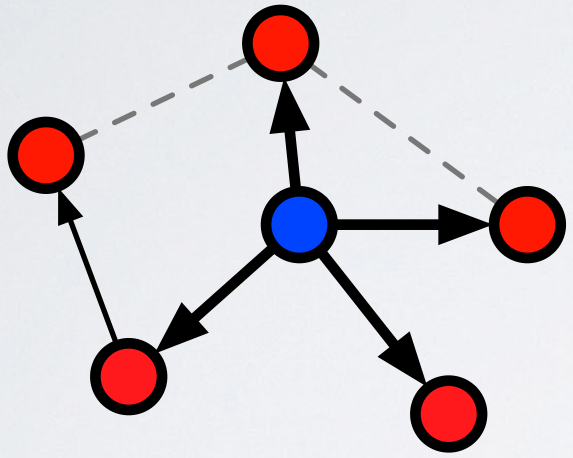
structural importance = cheap estimate of dynamical importance (aka "influence")

Boldi & Vigna, arxiv:1308.2140 (2013)
Borgatti, *Social Networks* **27**, 55–71 (2005)

Which nodes are important… we've heard this before!

Clauset lecture 2

describing networks



position = centrality:
harmonic, closeness centrality

importance = being in "center" of the network

harmonic $$c_i = \frac{1}{n-1} \sum_{j \neq i} \frac{1}{d_{ij}}$$

length of shortest path

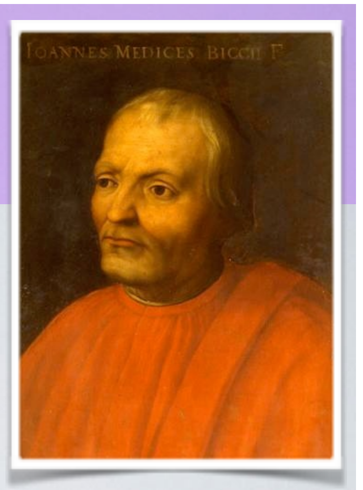
distance: $$d_{ij} = \begin{cases} \ell_{ij} & \text{if } j \text{ reachable from } i \\ \infty & \text{otherwise} \end{cases}$$

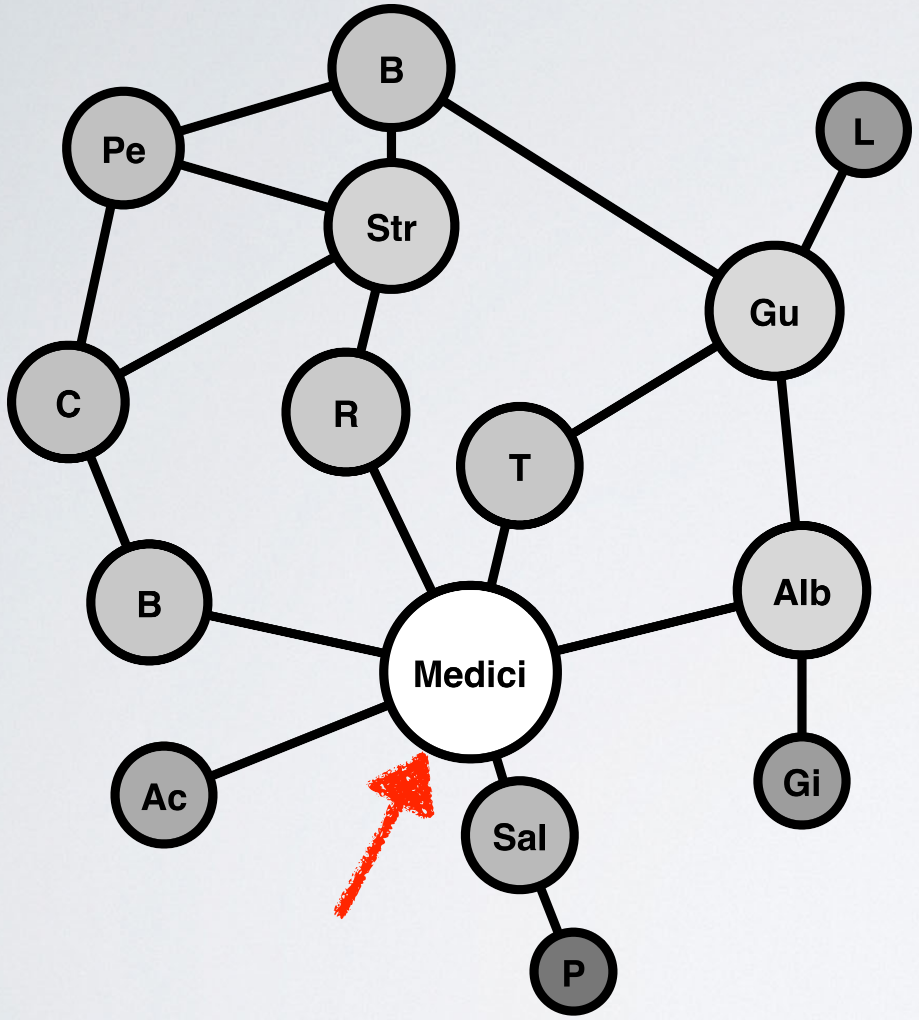
Boldi & Vigna, arxiv:1308.2140 (2013)
Borgatti, *Social Networks* **27**, 55–71 (2005)

Which nodes are important… we've heard this before!

Clauset lecture 2

network position: harmonic



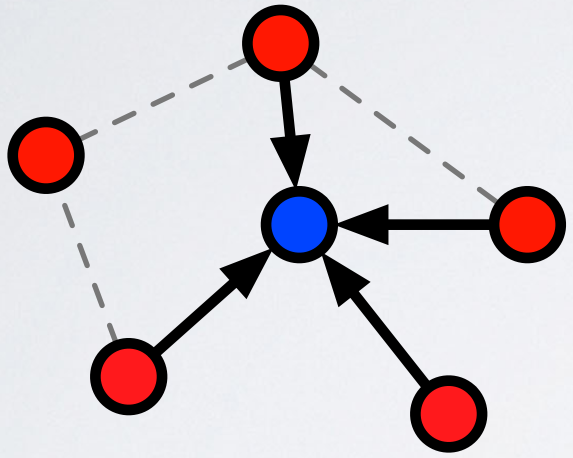


Medici	9.5
Guadagni	7.92
Albizzi	7.83
Strozzi	7.67
Ridolfi	7.25
Bischeri	7.2
Tornabuoni	7.17
Barbadori	7.08
Peruzzi	6.87
Castellani	6.87
Salviati	6.58
Acciaiuoli	5.92
Ginori	5.33
Lamberteschi	5.28
Pazzi	4.77

Which nodes are important… we've heard this before!

Clauset lecture 2

describing networks



position = centrality:
PageRank, Katz, eigenvector centrality

importance = sum of importances* of nodes that point at you

$$I_i = \sum_{j \to i} \frac{I_j}{k_j}$$

or, the left eigenvector of

$$\mathbf{A}\mathbf{x} = \lambda\,\mathbf{x}$$

Boldi & Vigna, arxiv:1308.2140 (2013)
Borgatti, *Social Networks* **27**, 55–71 (2005)

*modulo several technical details

Embeddings and Orderings 3: PageRank

PageRank defines scalar rank recursively:

important pages are those that are linked to by important pages.

- Great at finding the top 3 but limited predictions available using the PageRank scores.

The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

define a transition matrix

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

$$p_i = \frac{1-d}{N} + d \sum_j p_j \pi_{ji}$$

define a transition matrix   write the equation

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

define a transition matrix

$$p_i = \frac{1-d}{N} + d\sum_j p_j \pi_{ji}$$

write the equation

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

define a transition matrix

$$p_i = \frac{1-d}{N} + d \sum_j p_j \pi_{ji}$$

write the equation

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

define a transition matrix

$$p_i = \frac{1-d}{N} + d \sum_j p_j \pi_{ji}$$

write the equation

$$\mathbf{p} = \left(\frac{1-d}{N}\right)\mathbf{1} + d\pi^T \mathbf{p}$$

matrix-vector form

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

define a transition matrix

$$p_i = \frac{1-d}{N} + d \sum_j p_j \pi_{ji}$$

write the equation

$$\mathbf{p} = \left(\frac{1-d}{N}\right)\mathbf{1} + d\pi^T \mathbf{p}$$

matrix-vector form

Alternative: stationary distribution of random walk on the network + weak all-to-all links

Embeddings and Orderings 3: PageRank

We imagine a web surfer who choose a starting webpage at random.

From that webpage, she looks at the links on the page, and either
(a) clicks on a random link or
(b) stops surfing; when she returns, she starts at a new random page.

What's the probability that she's at a particular page? *That's PageRank*.

$$\pi_{ji} = \frac{A_{ji}}{k_j}$$

define a transition matrix

$$p_i = \frac{1-d}{N} + d\sum_j p_j \pi_{ji}$$

write the equation

$$\mathbf{p} = \left(\frac{1-d}{N}\right)\mathbf{1} + d\pi^T\mathbf{p}$$

matrix-vector form

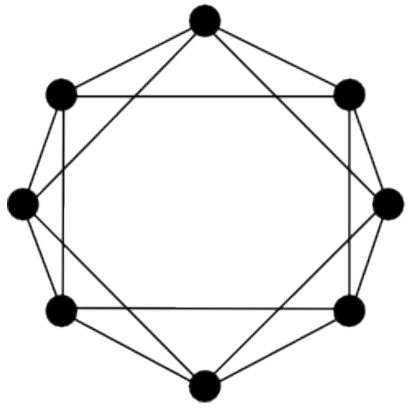
Alternative: stationary distribution of random walk on the network + weak all-to-all links

Jeremy Kun: http://www.infinitelooper.com/?v=K3pT0gTaDec&p=n

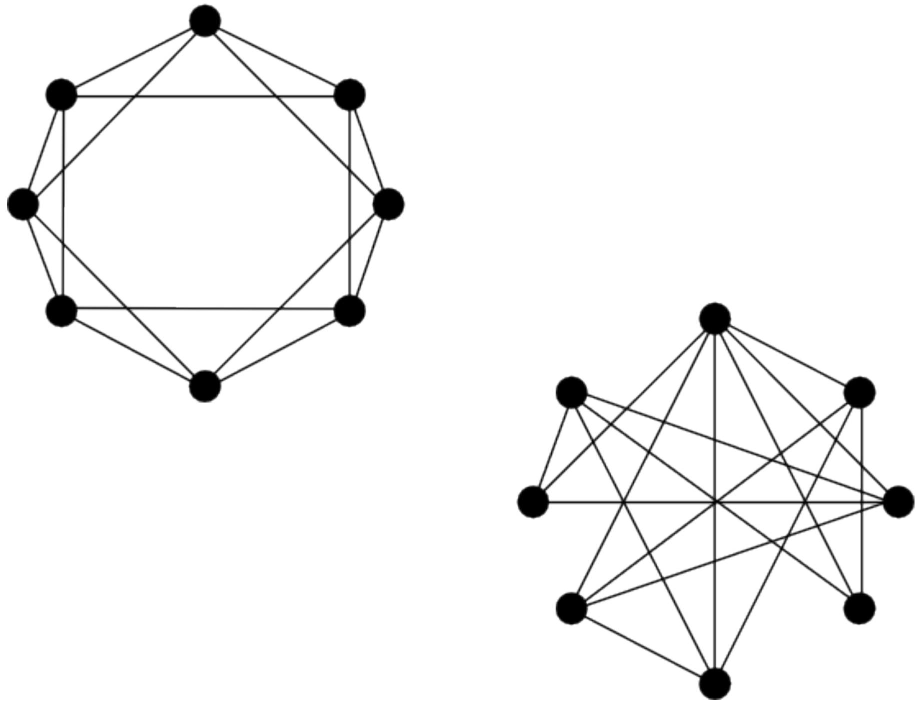
sin/cos: random graph ensembles

Stochastic models, sets, and distributions

- a generative model is just a recipe:
  choose parameters→make the network



Stochastic models, sets, and distributions

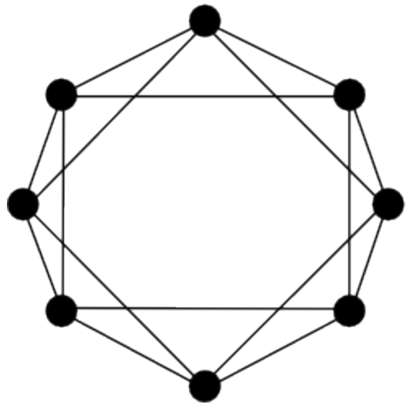


- a generative model is just a recipe:
  choose parameters→make the network

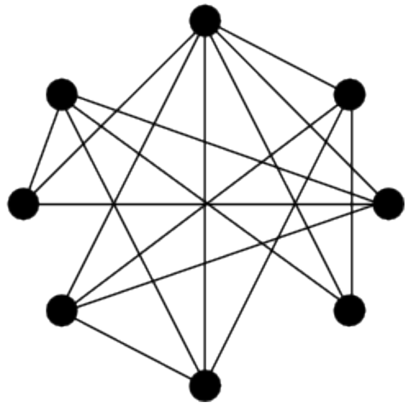
- a **stochastic** generative model is also just a recipe:
  choose parameters→draw **a** network

Stochastic models, sets, and distributions

- a generative model is just a recipe:
  choose parameters→make the network

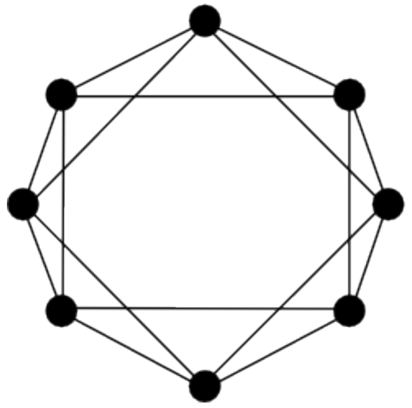


- a **stochastic** generative model is also just a recipe:
  choose parameters→draw **a** network

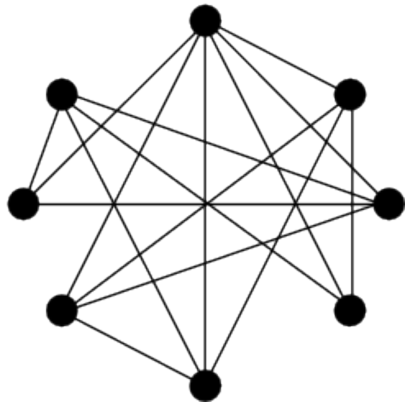


- since a single stochastic generative model can generate many networks, the model itself corresponds to a **set of networks**.

Stochastic models, sets, and distributions



- a generative model is just a recipe:
  choose parameters→make the network

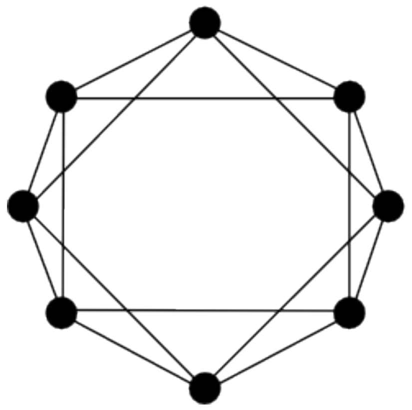


- a **stochastic** generative model is also just a recipe:
  choose parameters→draw **a** network

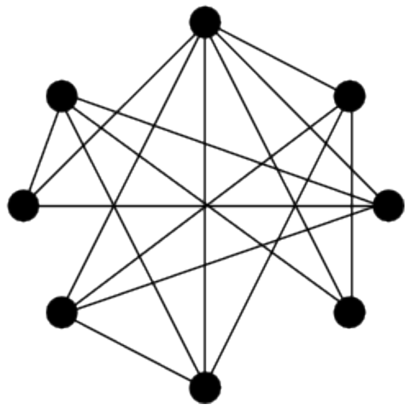
- since a single stochastic generative model can generate many networks, the model itself corresponds to a **set of networks**.

- and since the generative model itself is some combination or composition of random variables, a **random graph model** is a set of possible networks, each with an associated probability, i.e., a distribution.

Stochastic models, sets, and distributions



- a generative model is just a recipe:
  choose parameters→make the network



- a **stochastic** generative model is also just a recipe:
  choose parameters→draw **a** network

- since a single stochastic generative model can generate many networks, the model itself corresponds to a **set of networks**.

- and since the generative model itself is some combination or composition of random variables, a **random graph model** is a set of possible networks, each with an associated probability, i.e., a distribution.

this tangent will introduce the second most popular model for networks:
configuration model(s)—uniform distributions over networks w/ fixed deg. seq.

Why care about random graphs w/ fixed degree sequence?

Since many networks have broad or peculiar degree sequences, these random graph distributions are commonly used for:

Hypothesis testing:

Can a particular network's properties be explained by the degree sequence alone?

Why care about random graphs w/ fixed degree sequence?

Since many networks have broad or peculiar degree sequences, these random graph distributions are commonly used for:

Hypothesis testing:

Can a particular network's properties be explained by the degree sequence alone?

Modeling:

How does the degree distribution affect the epidemic threshold for disease transmission?

Why care about random graphs w/ fixed degree sequence?

Since many networks have broad or peculiar degree sequences,
these random graph distributions are commonly used for:

Hypothesis testing:

Can a particular network's properties be explained by the degree sequence alone?

Modeling:

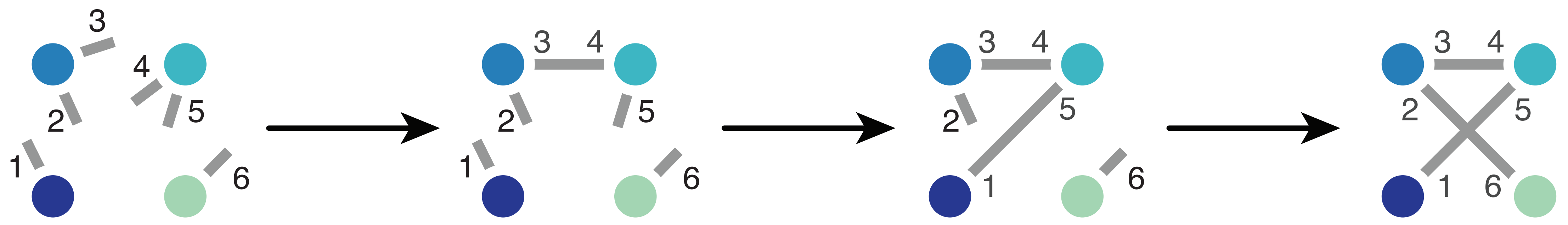
How does the degree distribution affect the epidemic threshold for disease transmission?

Null model for Modularity, Stochastic Block Model:

Compare an empirical graph with (possibly) community structure to the ensemble of random graphs with the same vertex degrees.

Stub Matching to draw from the config. model

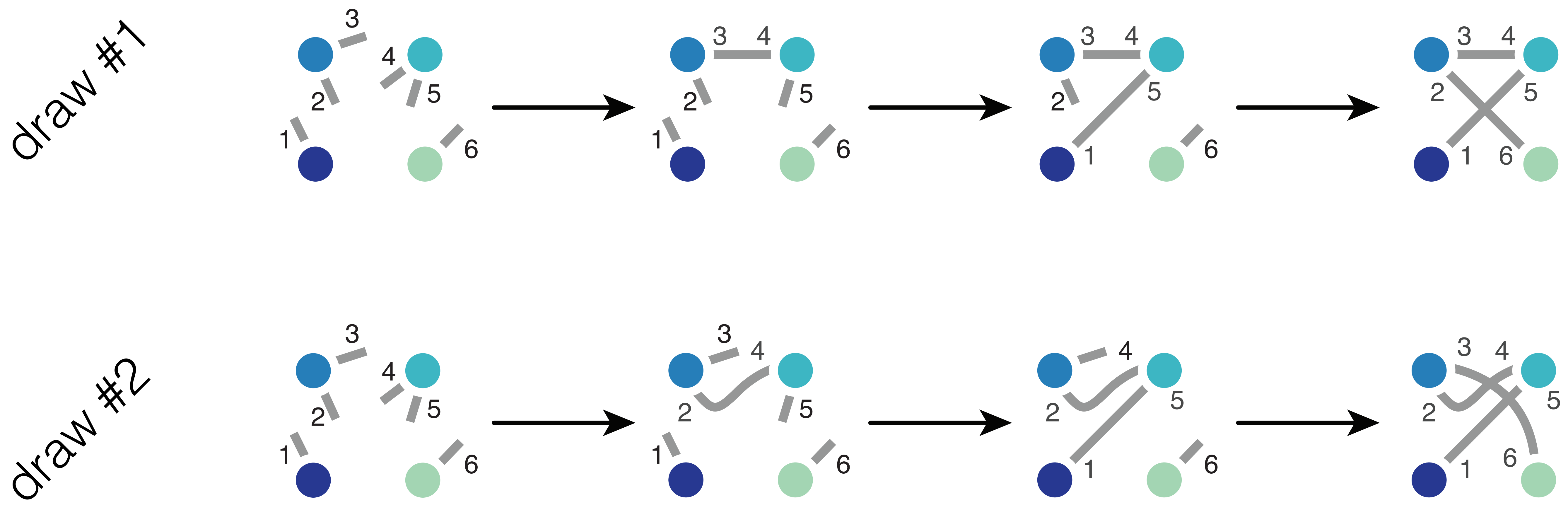
$$\vec{k} = \{1, 2, 2, 1\}$$

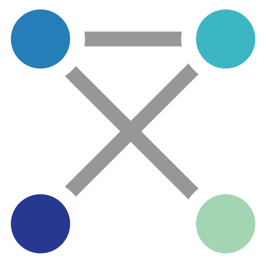


the standard algorithm:
draw from the distribution by sequential "Stub Matching"

1. initialize each node n with k_n half-edges or stubs.
2. choose two stubs uniformly at random and join to form an edge.

Stub Matching to draw from the config. model

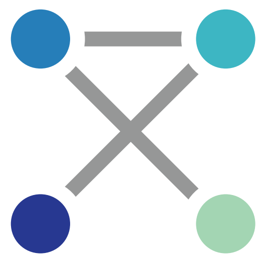


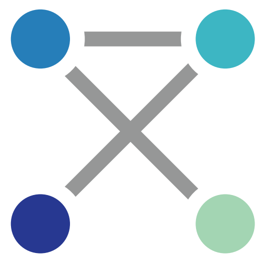


Are these two different networks? or the same network?

Are these two different networks? or the same network?

Are stubs distinguishable or not?





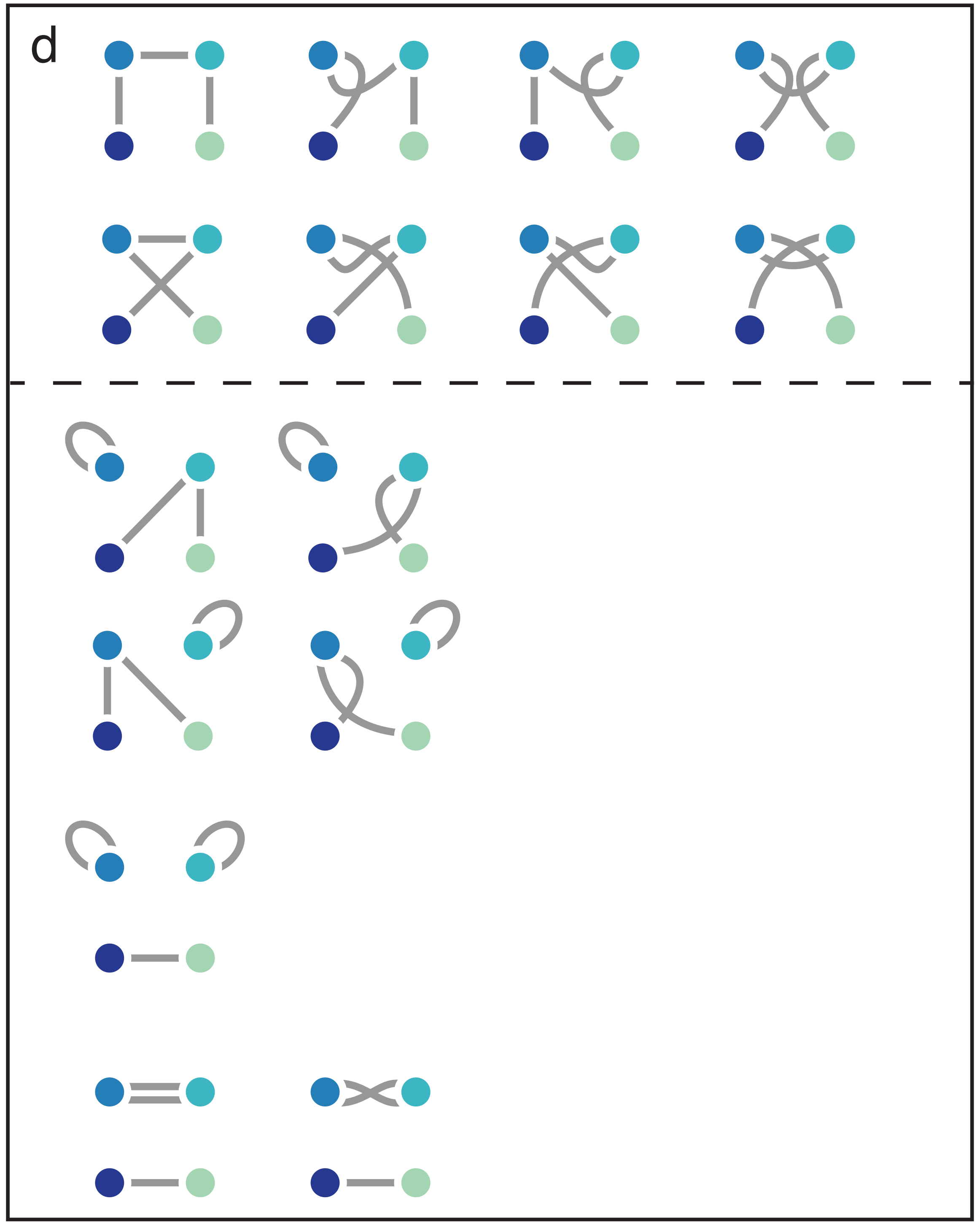
Are these two different networks? or the same network?

Are stubs distinguishable or not?

When it comes to doing science, the answer matters.

The distribution according to stub-matching

When we draw a graph using stub matching,
this is the set of graphs that we uniformly sample.

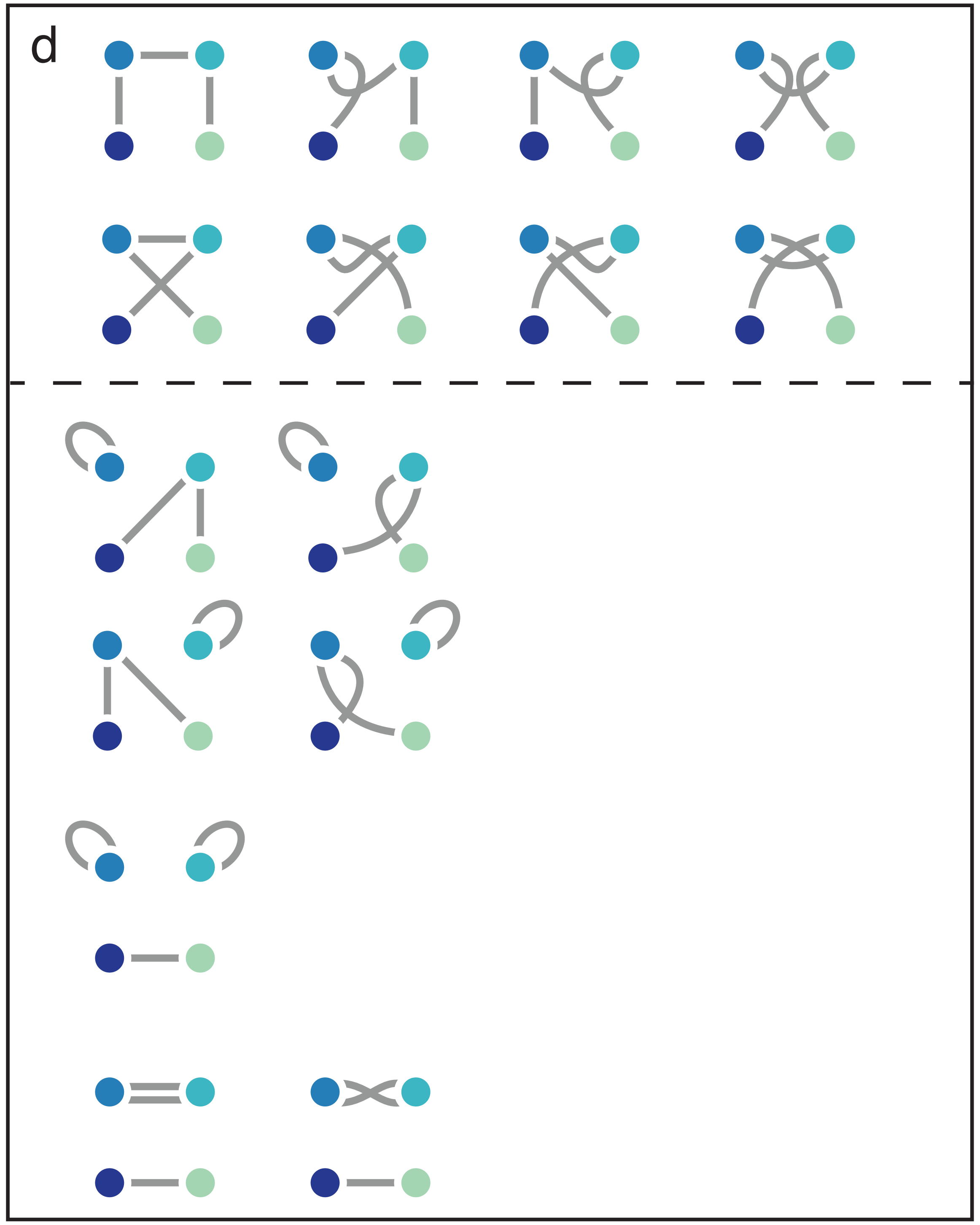


stub-labeled

The distribution according to stub-matching

When we draw a graph using stub matching, this is the set of graphs that we uniformly sample.

8 of the graphs are simple, while the other 7 have self-loops or multiedges.



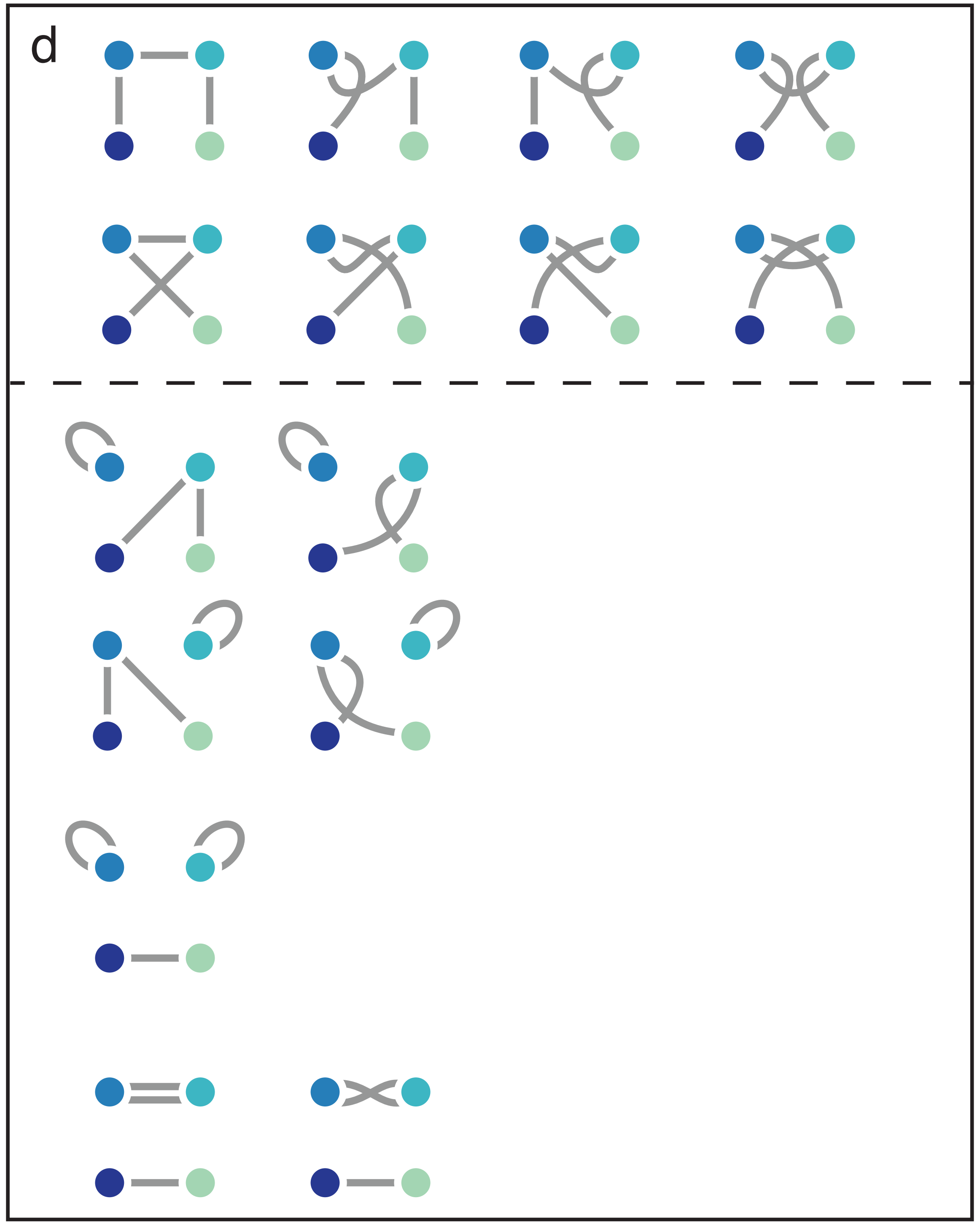
stub-labeled

The distribution according to stub-matching

When we draw a graph using stub matching, this is the set of graphs that we uniformly sample.

8 of the graphs are simple, while the other 7 have self-loops or multiedges.

We therefore say that stub matching uniformly samples space of **stub-labeled loopy multigraphs**.



stub-labeled

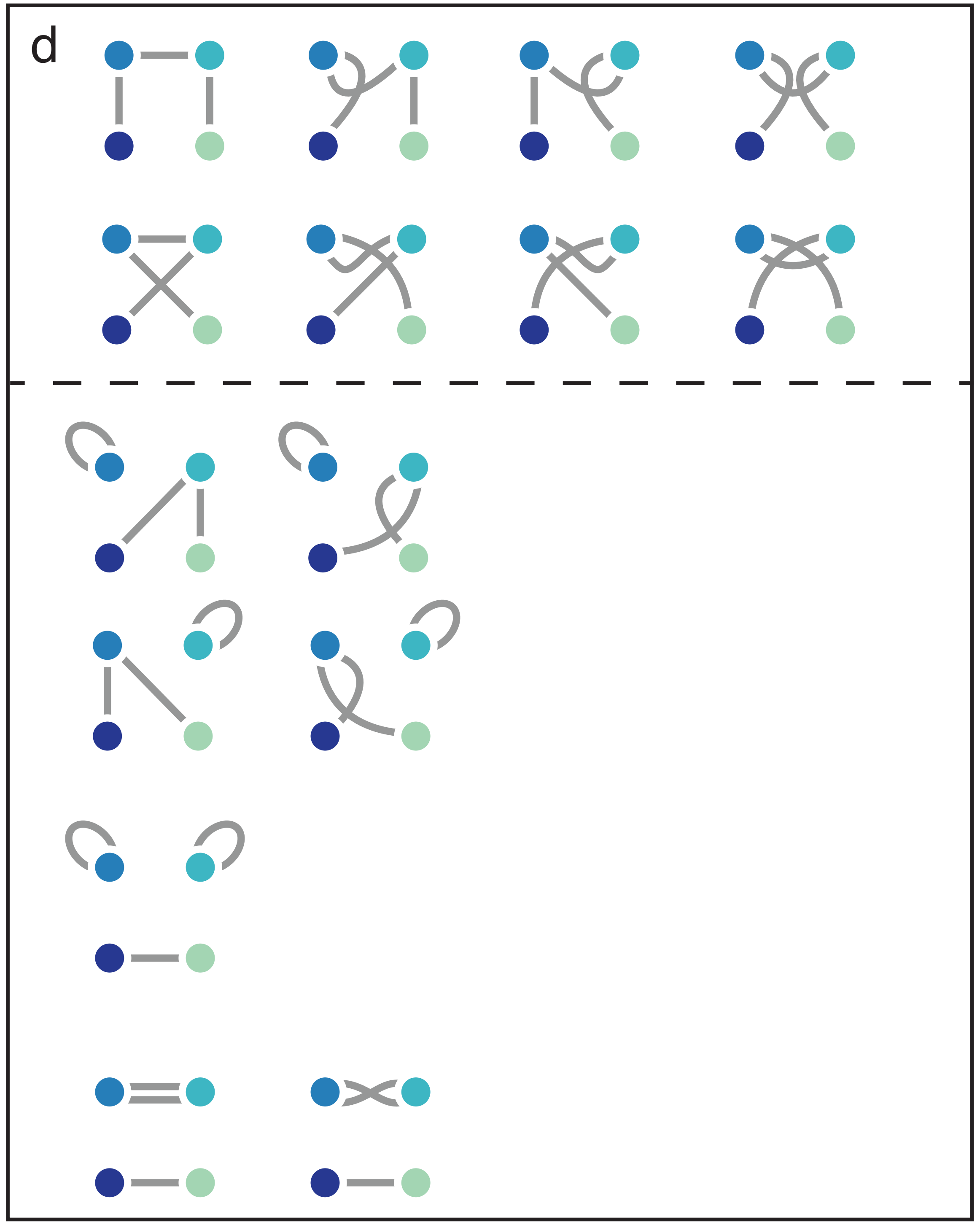
The distribution according to stub-matching

When we draw a graph using stub matching, this is the set of graphs that we uniformly sample.

8 of the graphs are simple, while the other 7 have self-loops or multiedges.

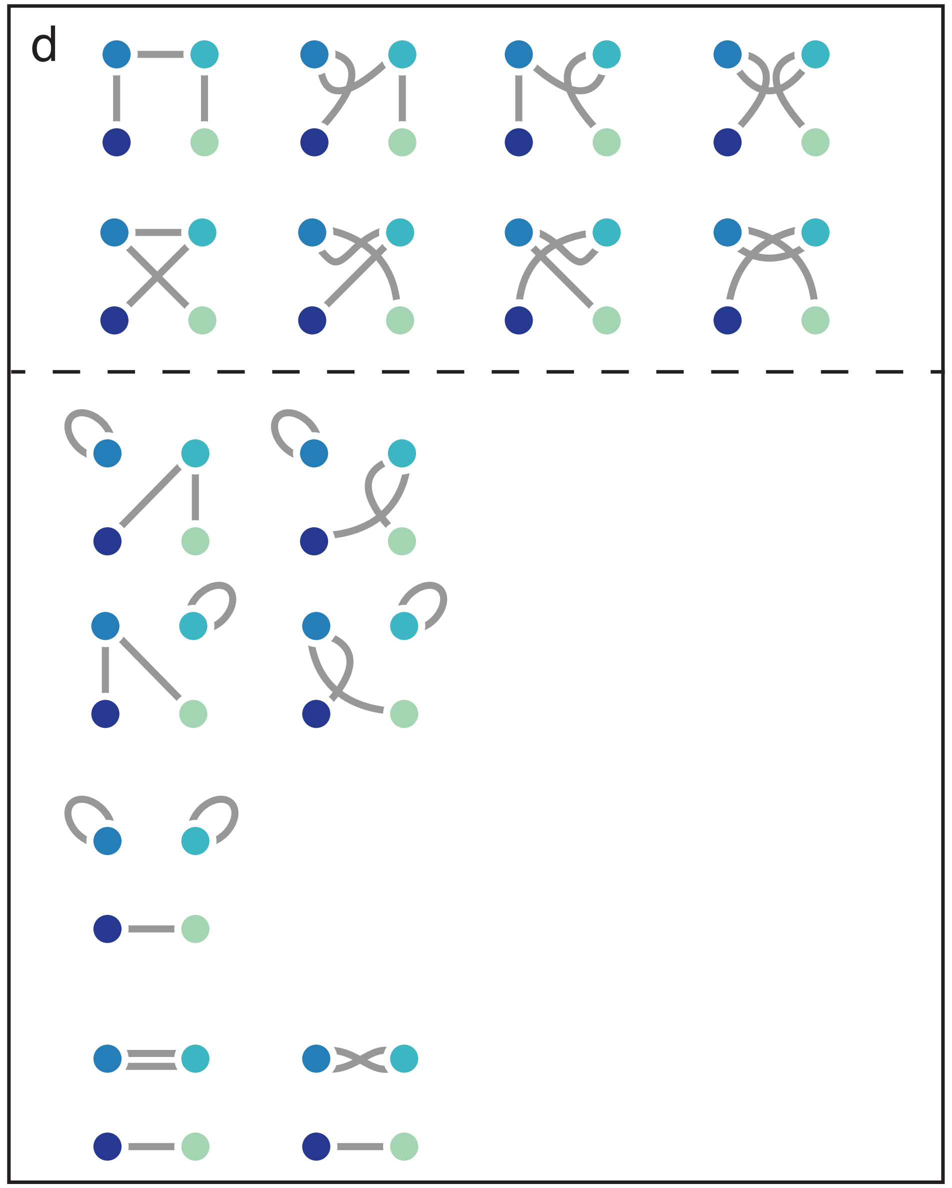
We therefore say that stub matching uniformly samples space of **stub-labeled loopy multigraphs**.

Note, however, that this is not a uniform sample over adjacency matrices (rows).



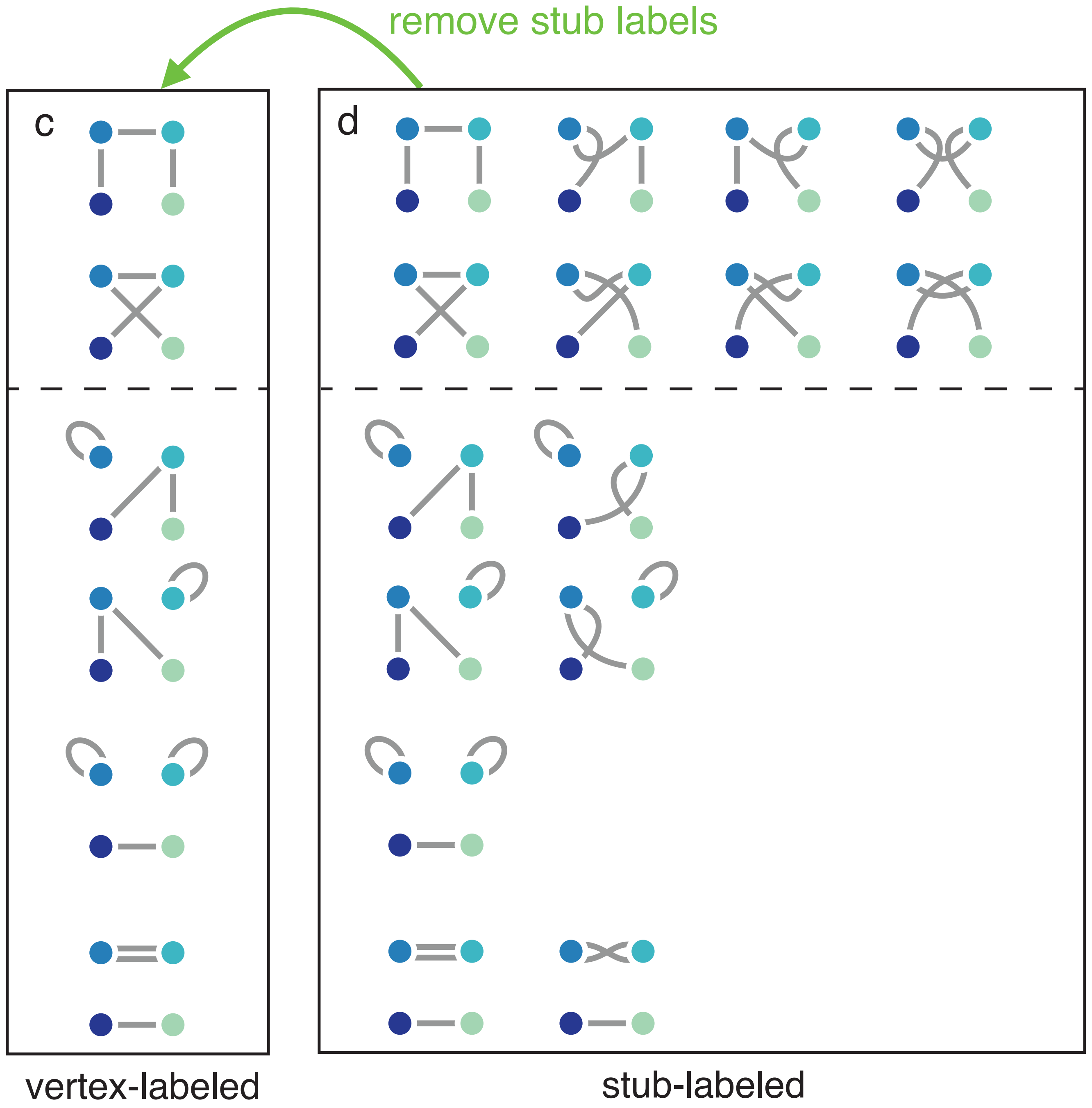
stub-labeled

The importance of uniform distributions

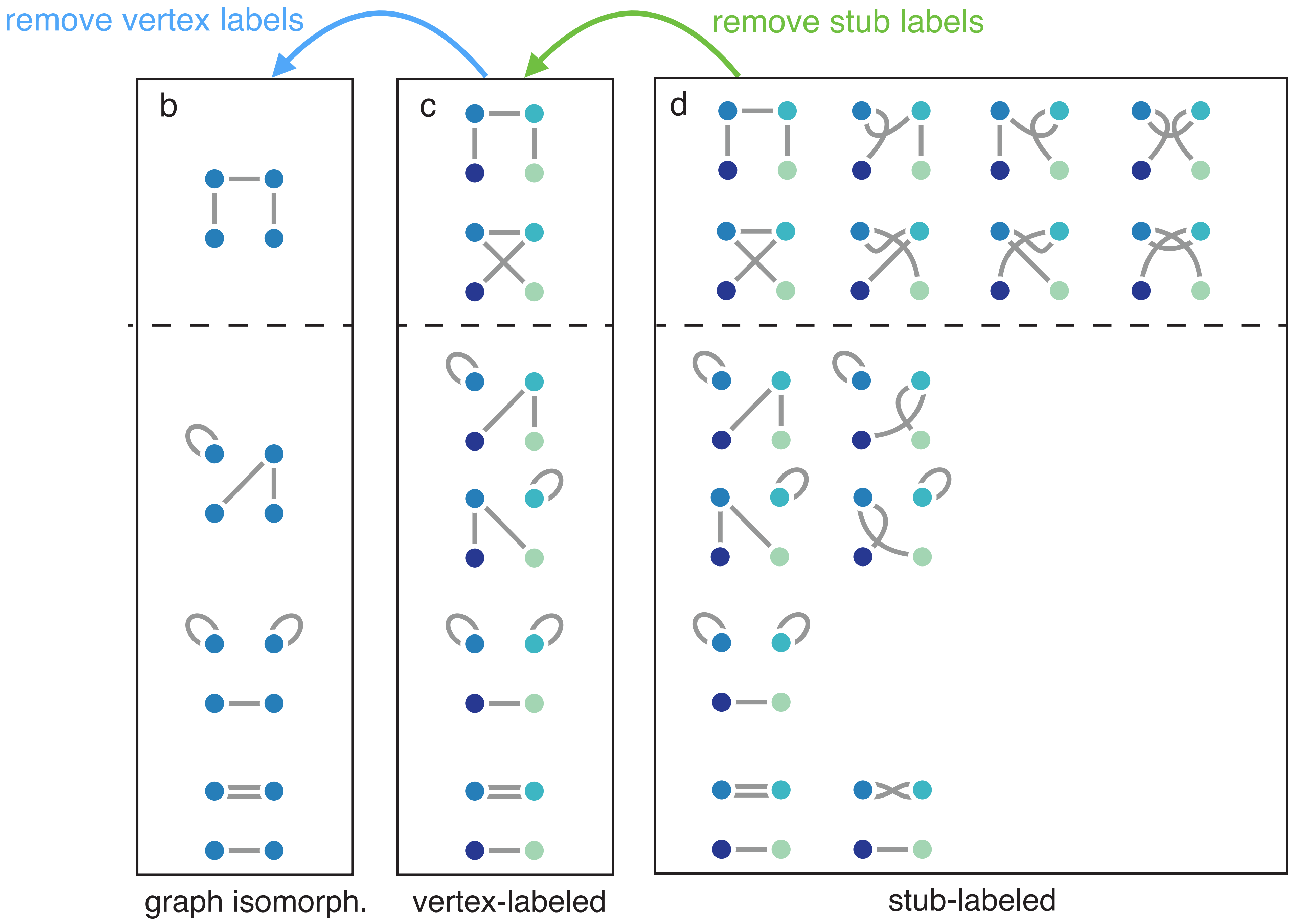


stub-labeled

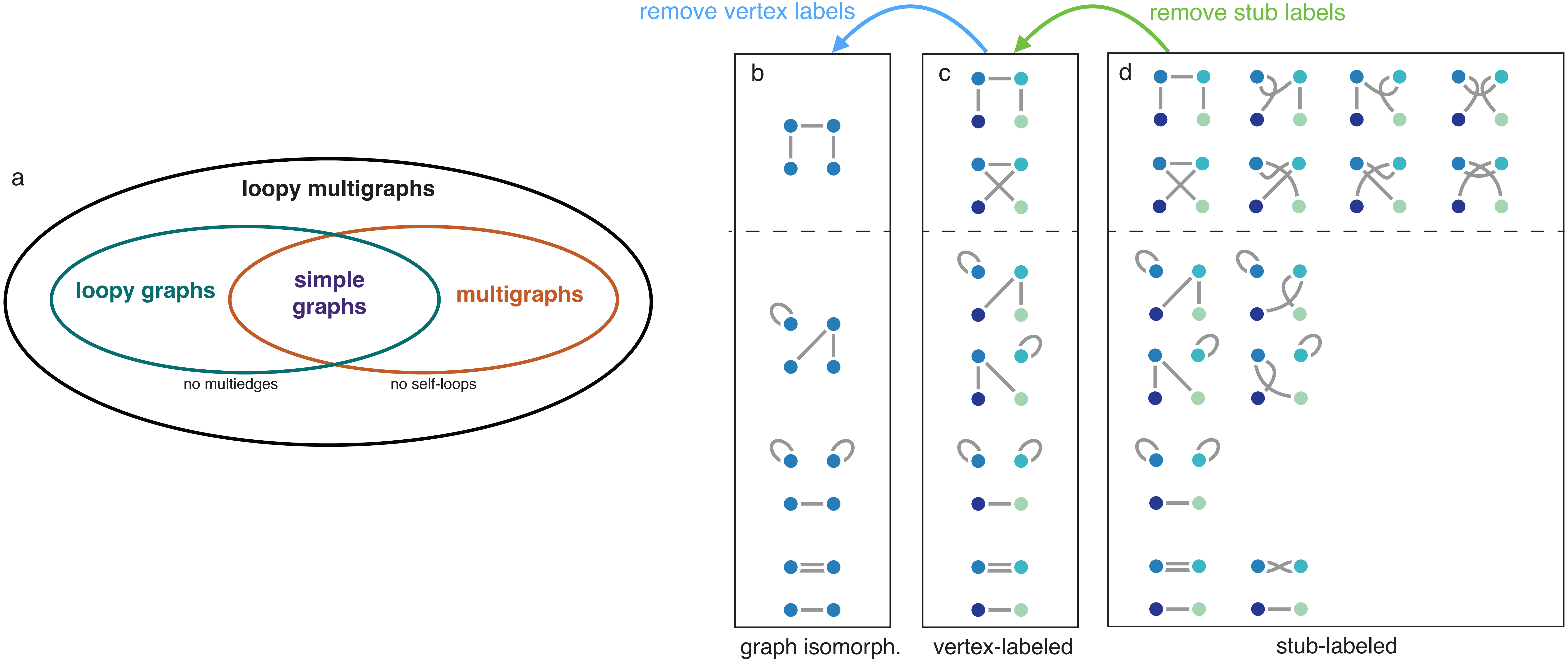
The importance of uniform distributions



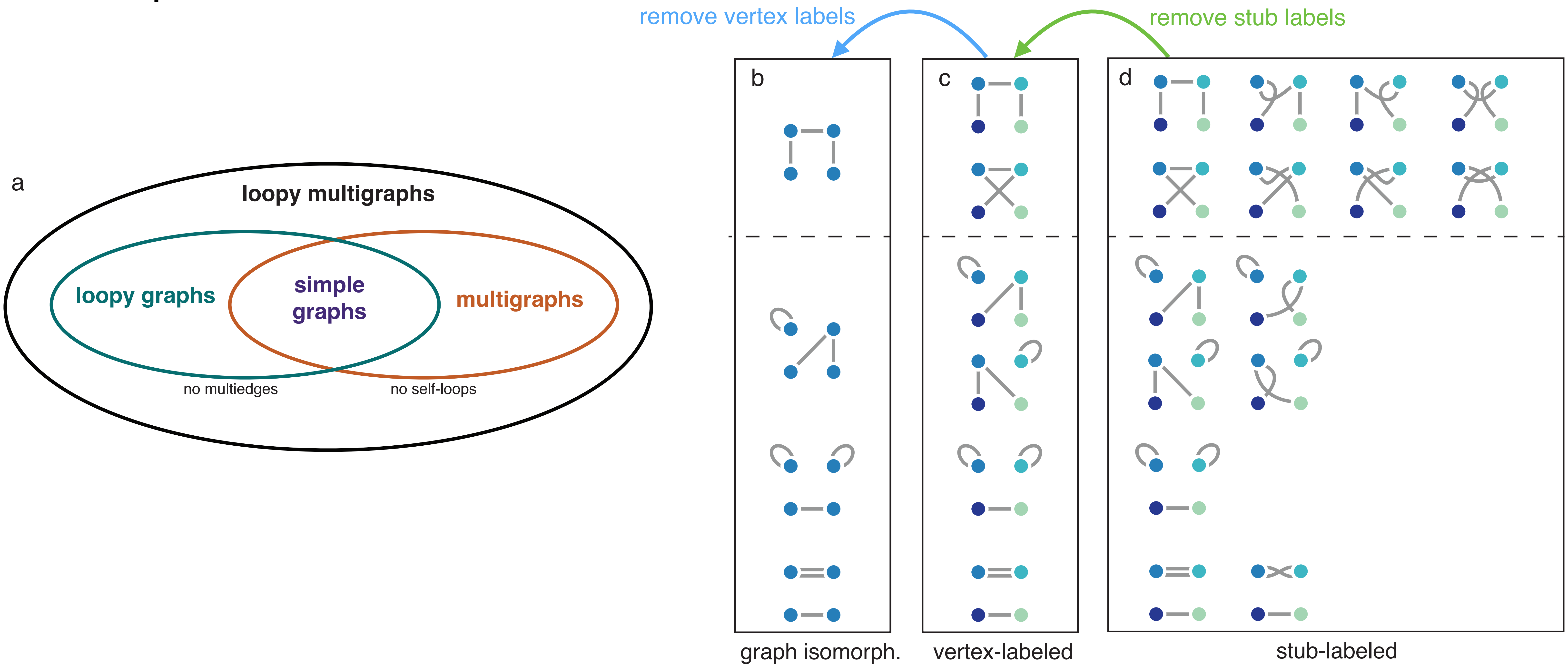
The importance of uniform distributions



The importance of uniform distributions



The importance of uniform distributions



null model hypothesis testing needs uniform sampling for all 8 spaces:
loopy{0,1} x **multigraph**{0,1} x {**stub-**,**vertex-**}

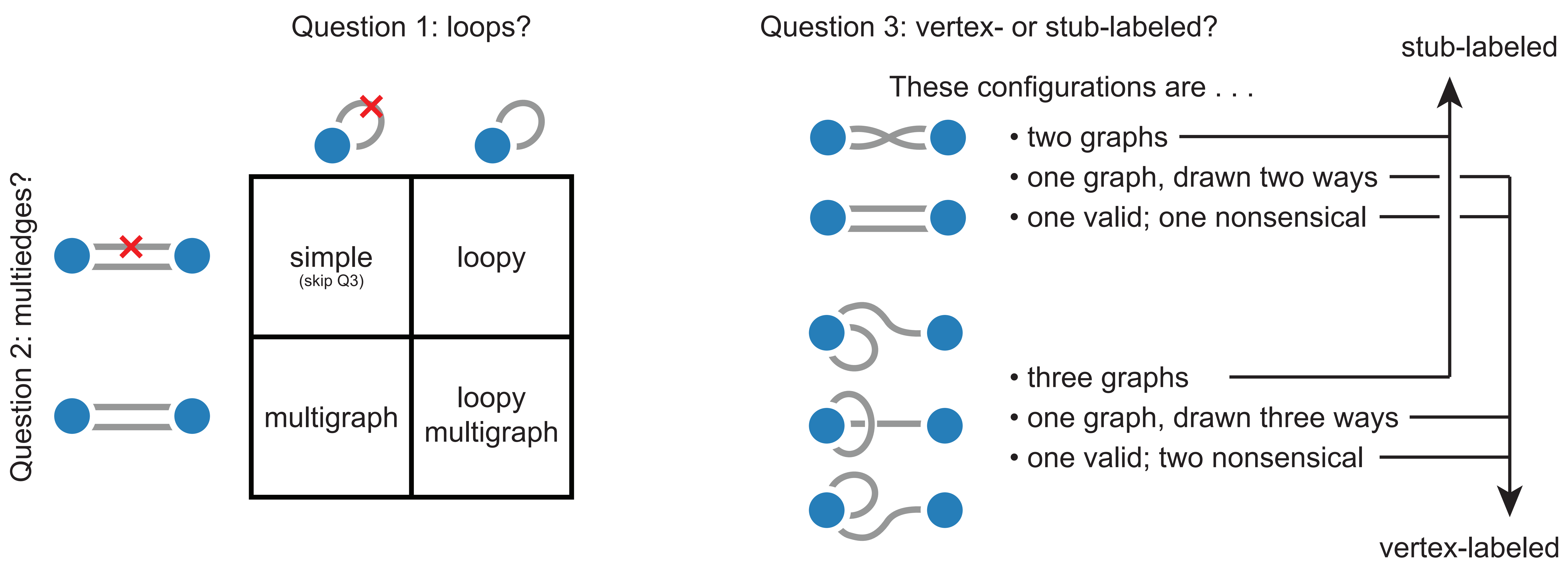
Hypothesis testing

Do barn swallows **tend to associate** with other swallows **of similar color**?

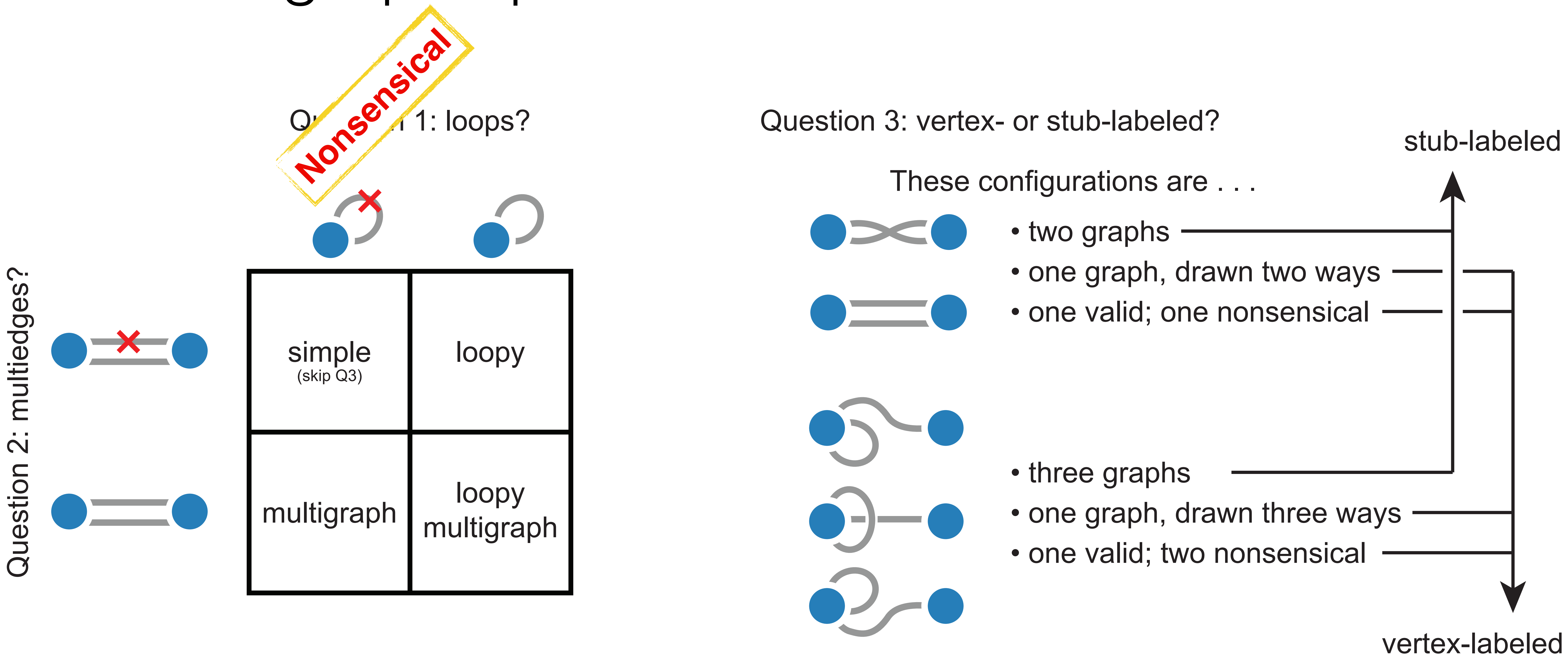
Data: **bird interactions, bird colors**.

Compute **color assortativity** [correlation over edges]

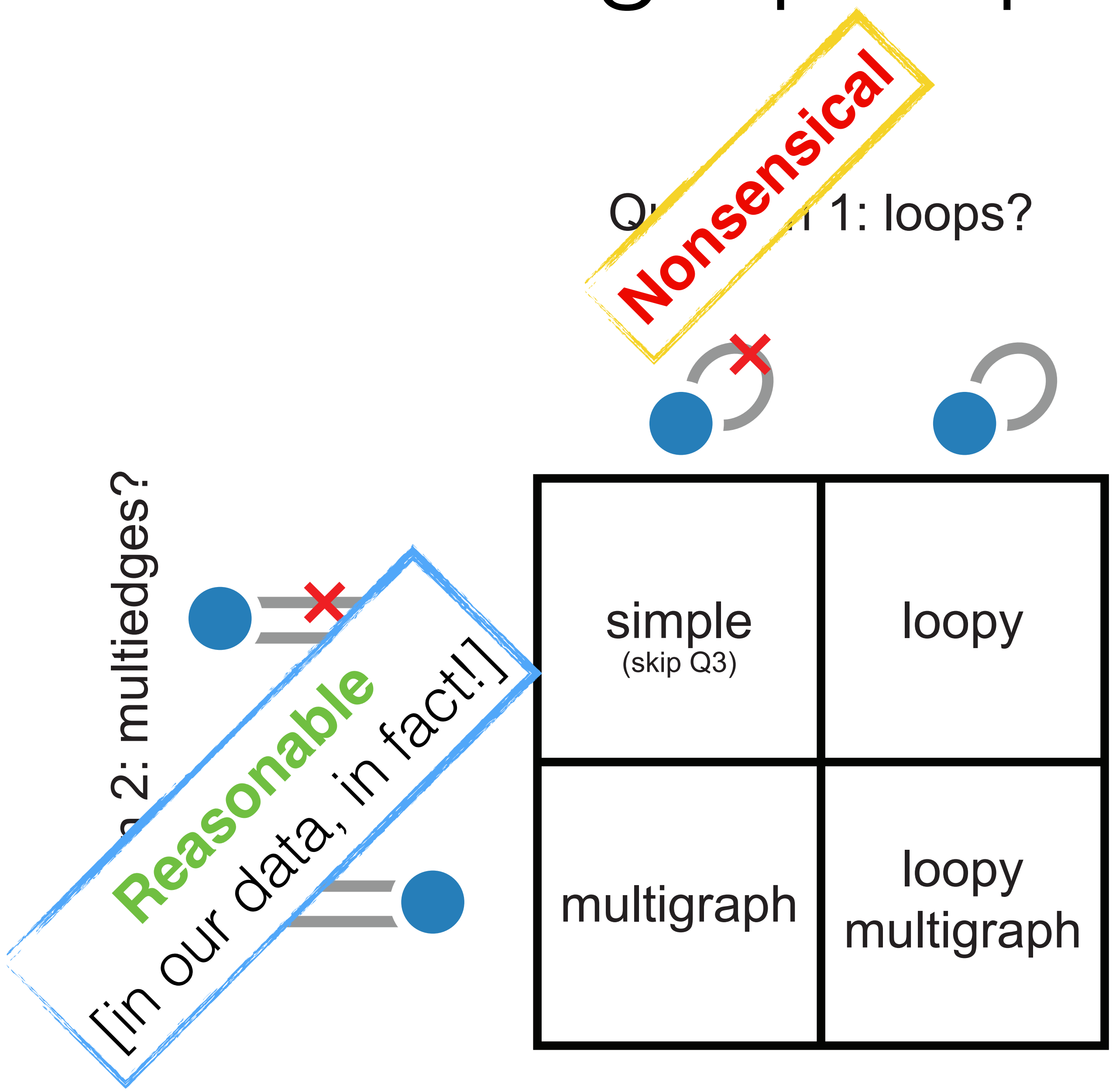
Choose a graph space for barn swallows

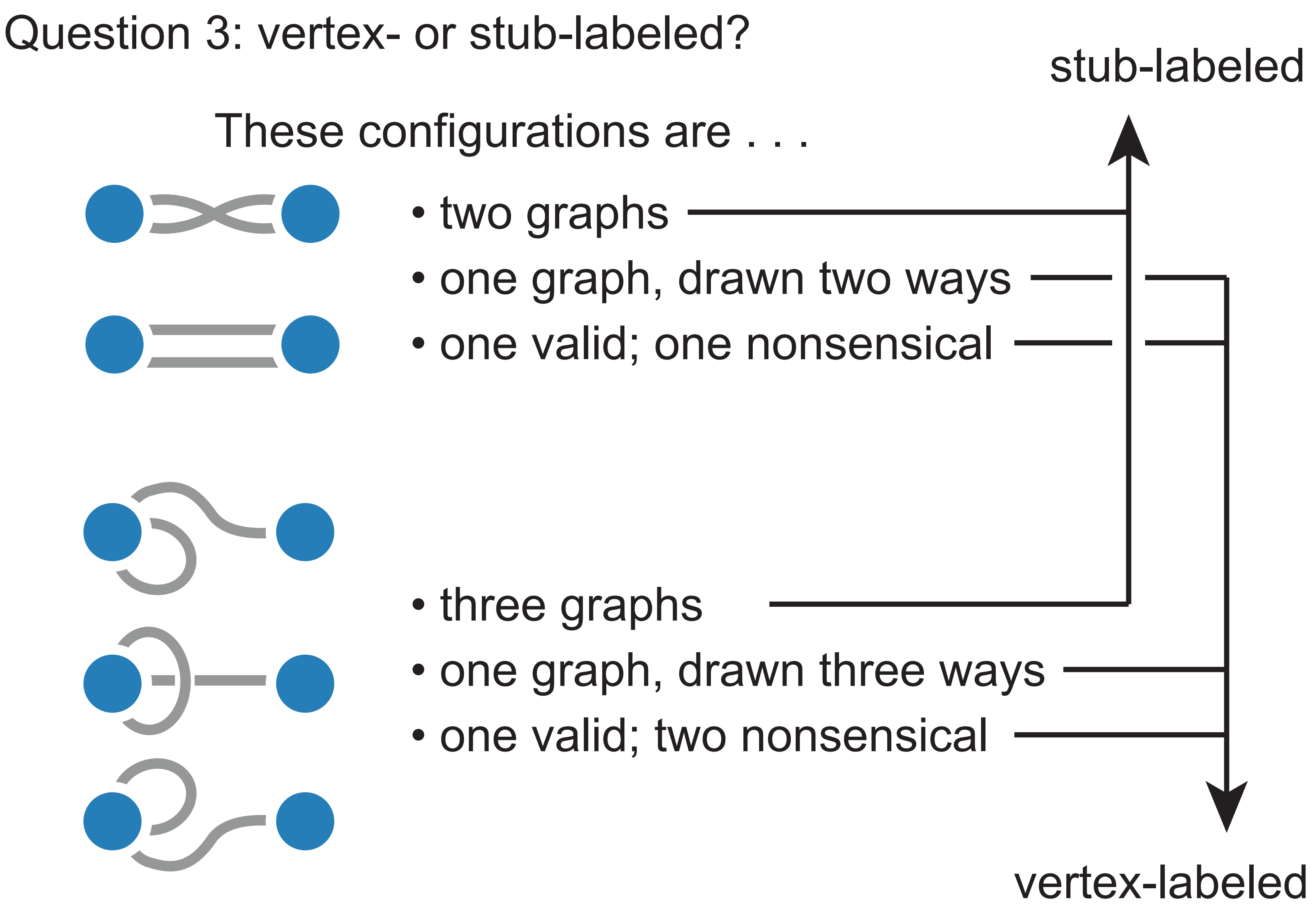


Choose a graph space for barn swallows

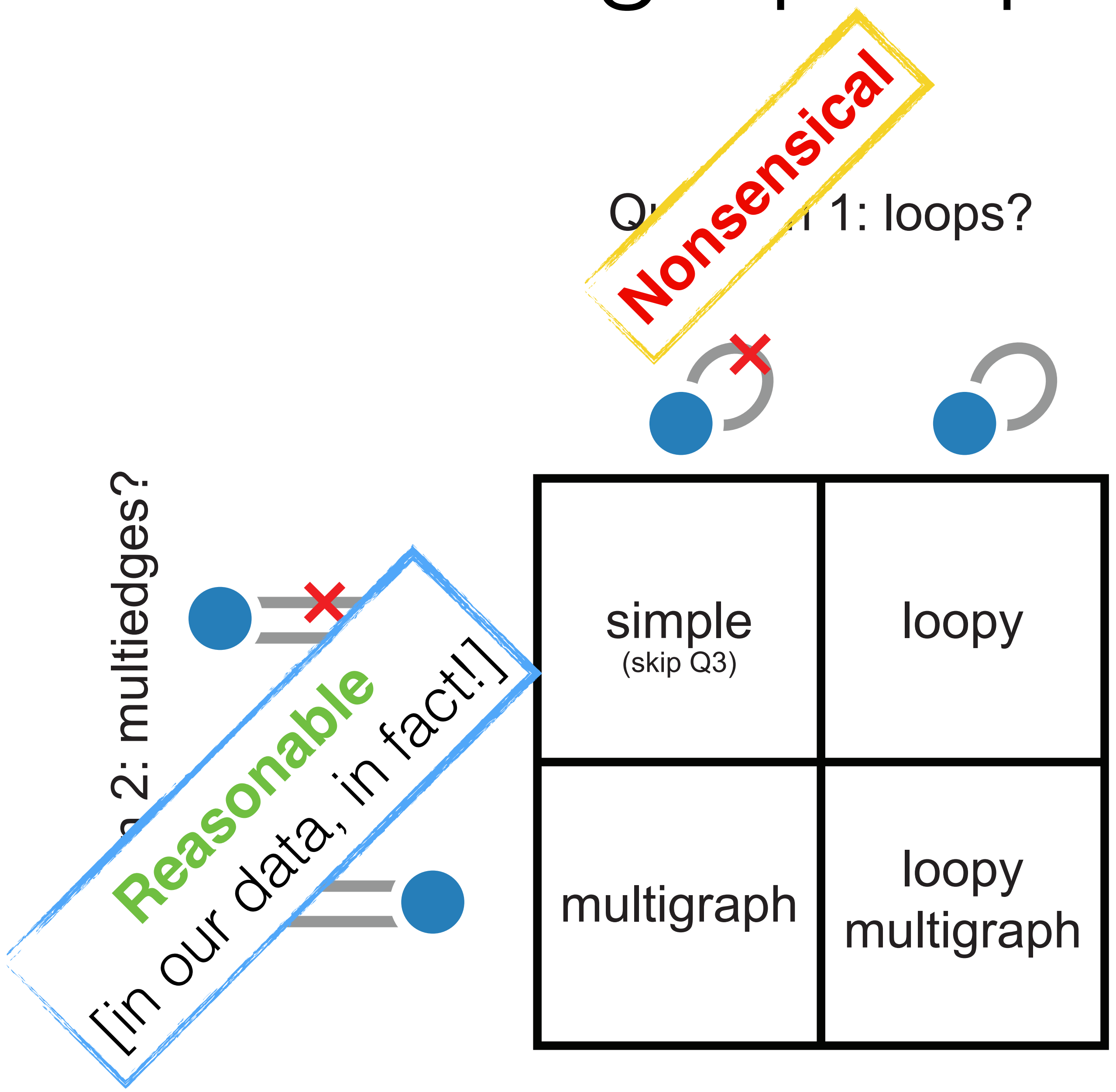


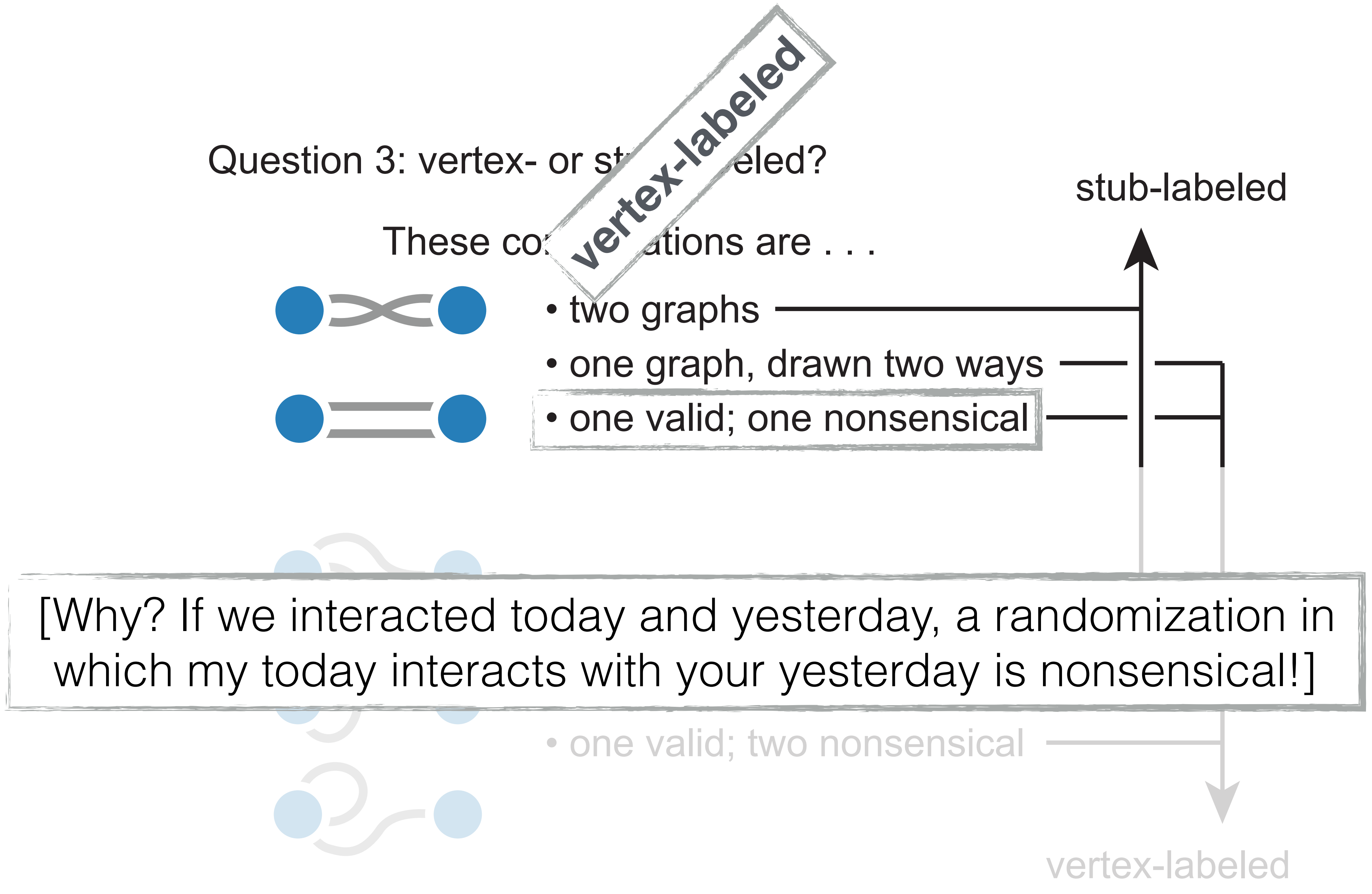
Choose a graph space for barn swallows





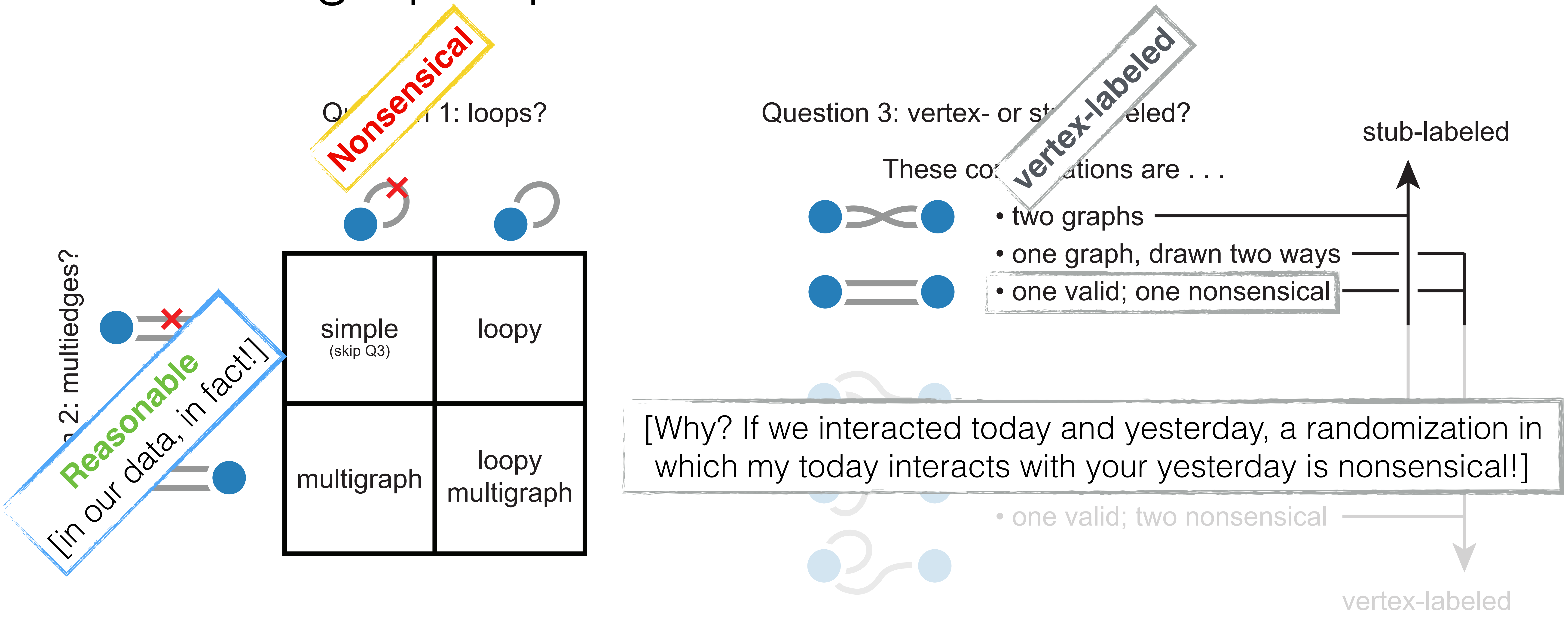
Choose a graph space for barn swallows





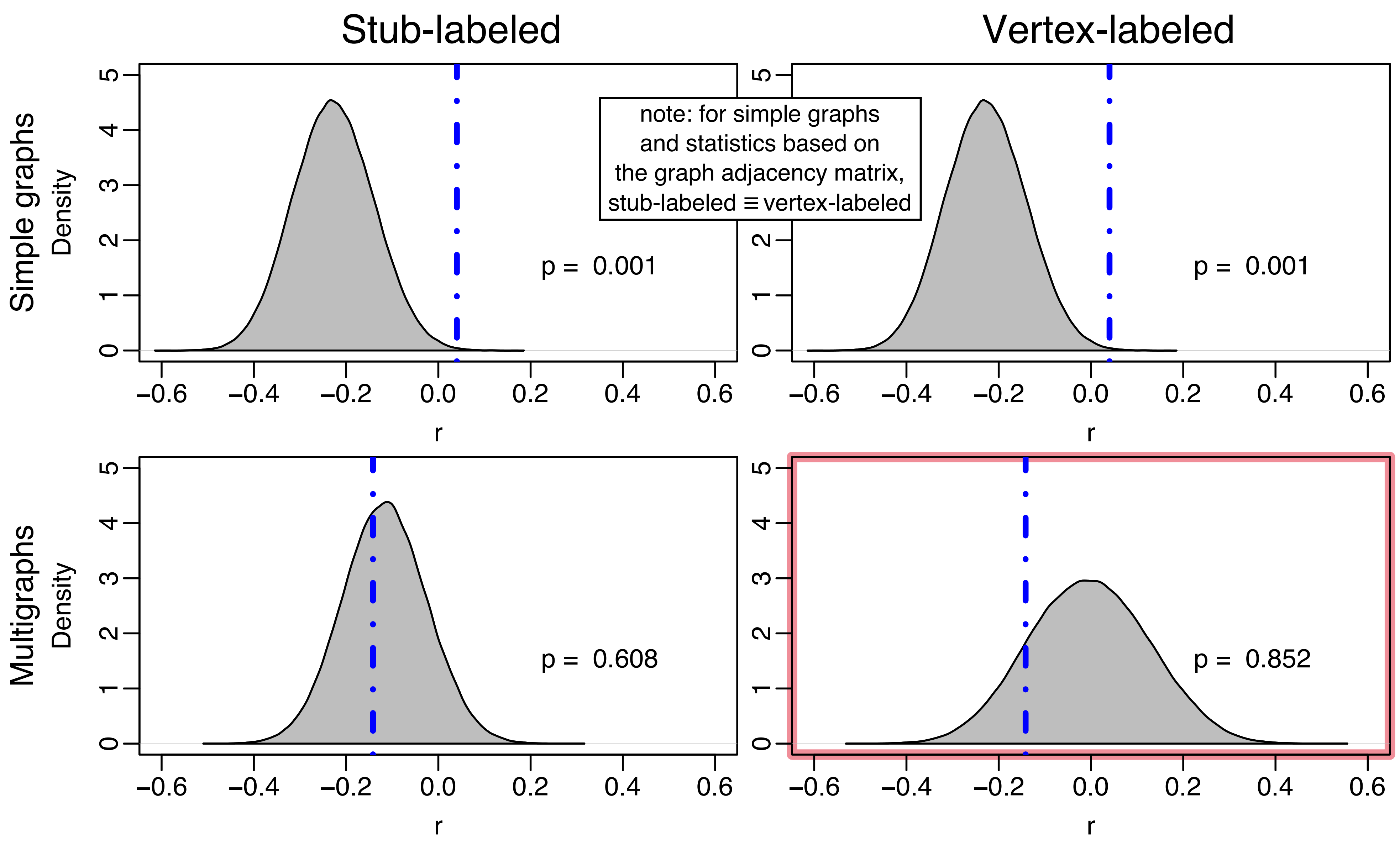
[Why? If we interacted today and yesterday, a randomization in which my today interacts with your yesterday is nonsensical!]

Choose a graph space for barn swallows



This should be modeled as a **vertex-labeled multigraph**.

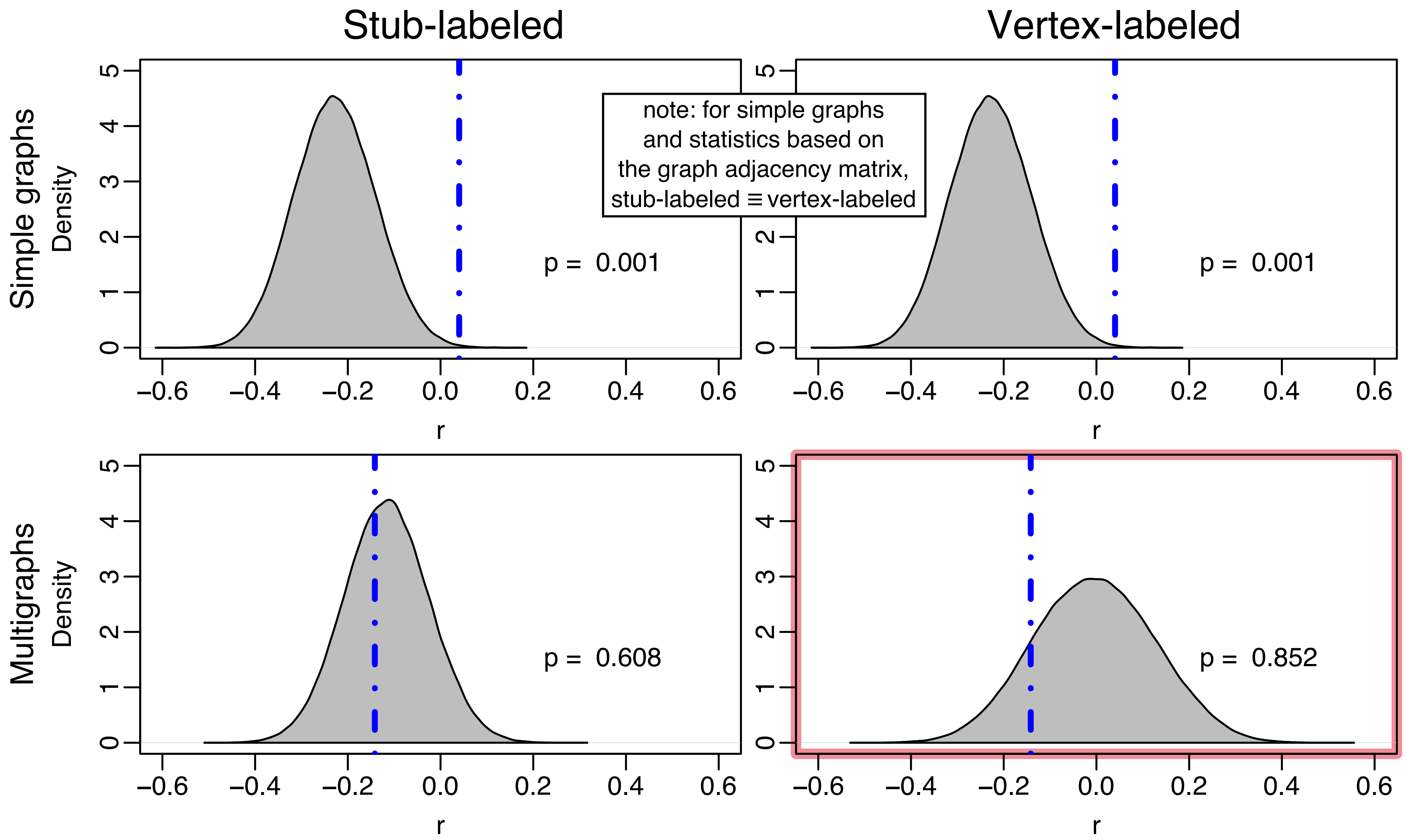
Assortative pairing of barn swallows



Sanity check:
should be = for simple

Fosdick, Larremore, Nishimura, Ugander. SIAM Review 60 (2) 315-355 (2018).

Assortative pairing of barn swallows

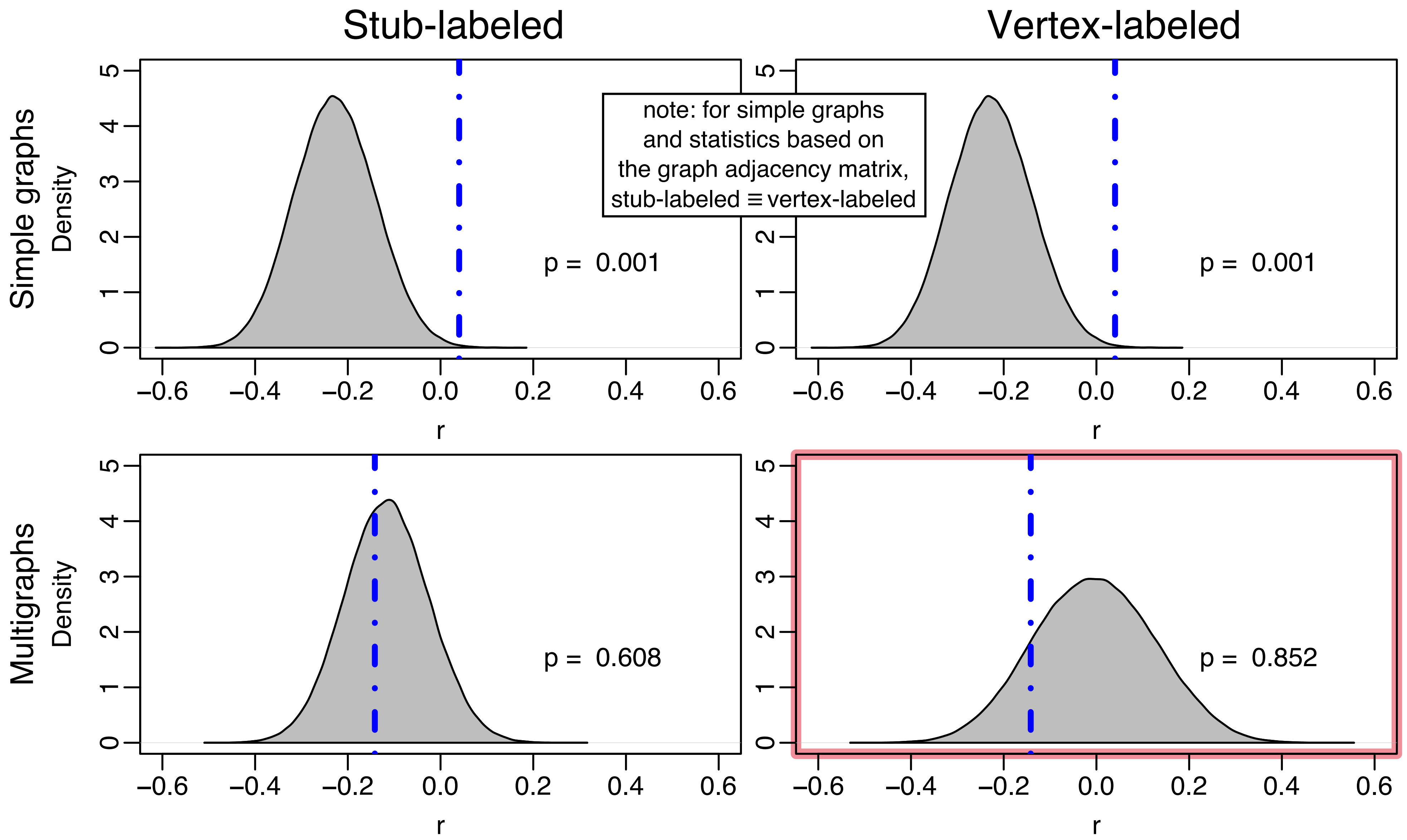


Sanity check:
should be = for simple

NONE of these is centered at zero.
Correct space is *meaningfully* different.

Fosdick, Larremore, Nishimura, Ugander. SIAM Review 60 (2) 315-355 (2018).

Assortative pairing of barn swallows



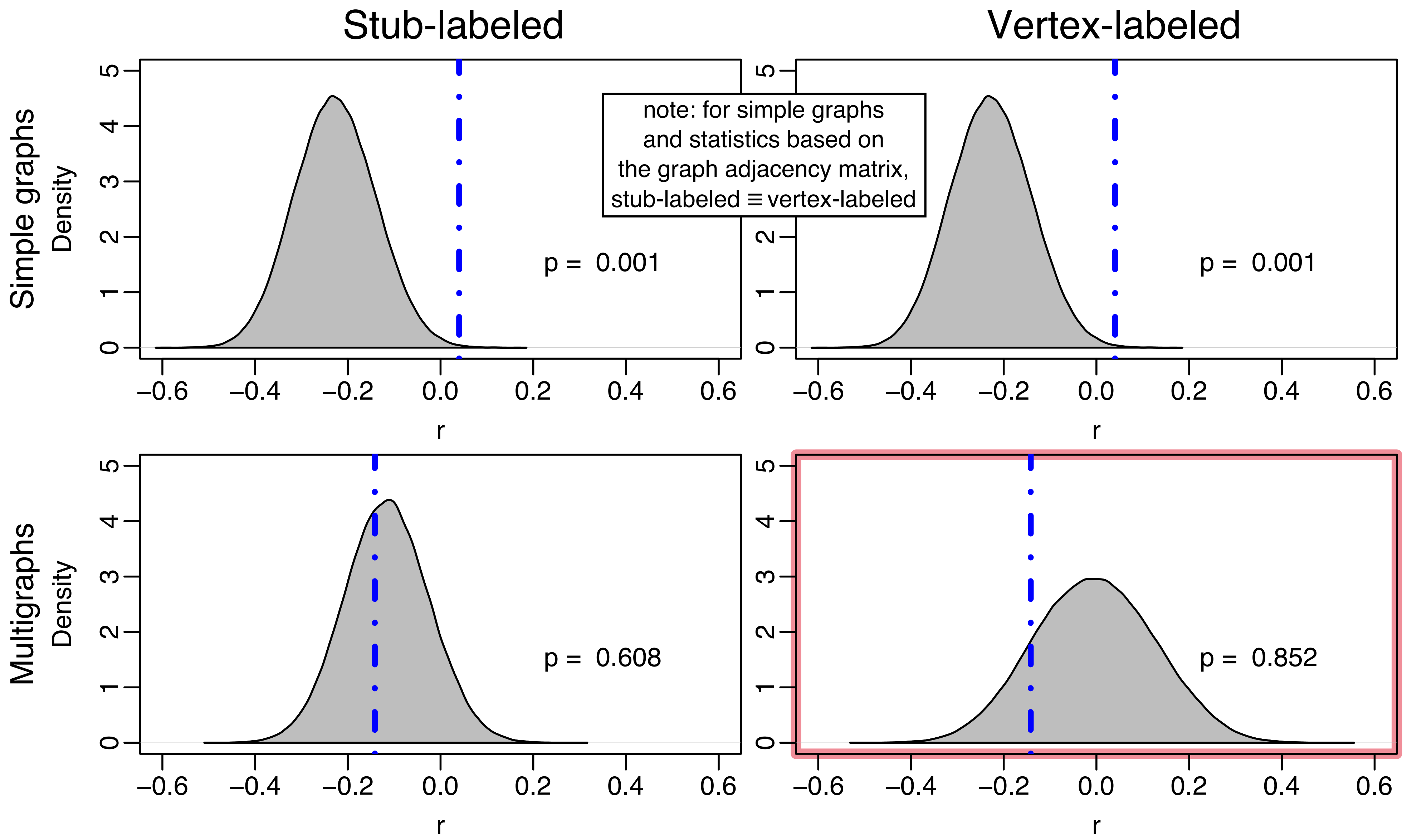
Sanity check:
should be = for simple

NONE of these is centered at zero.
Correct space is *meaningfully* different.

Uniform sampling means we can compare empirical value to null distribution to draw scientific conclusions.

Fosdick, Larremore, Nishimura, Ugander. SIAM Review 60 (2) 315-355 (2018).

Assortative pairing of barn swallows



Sanity check:
should be = for simple

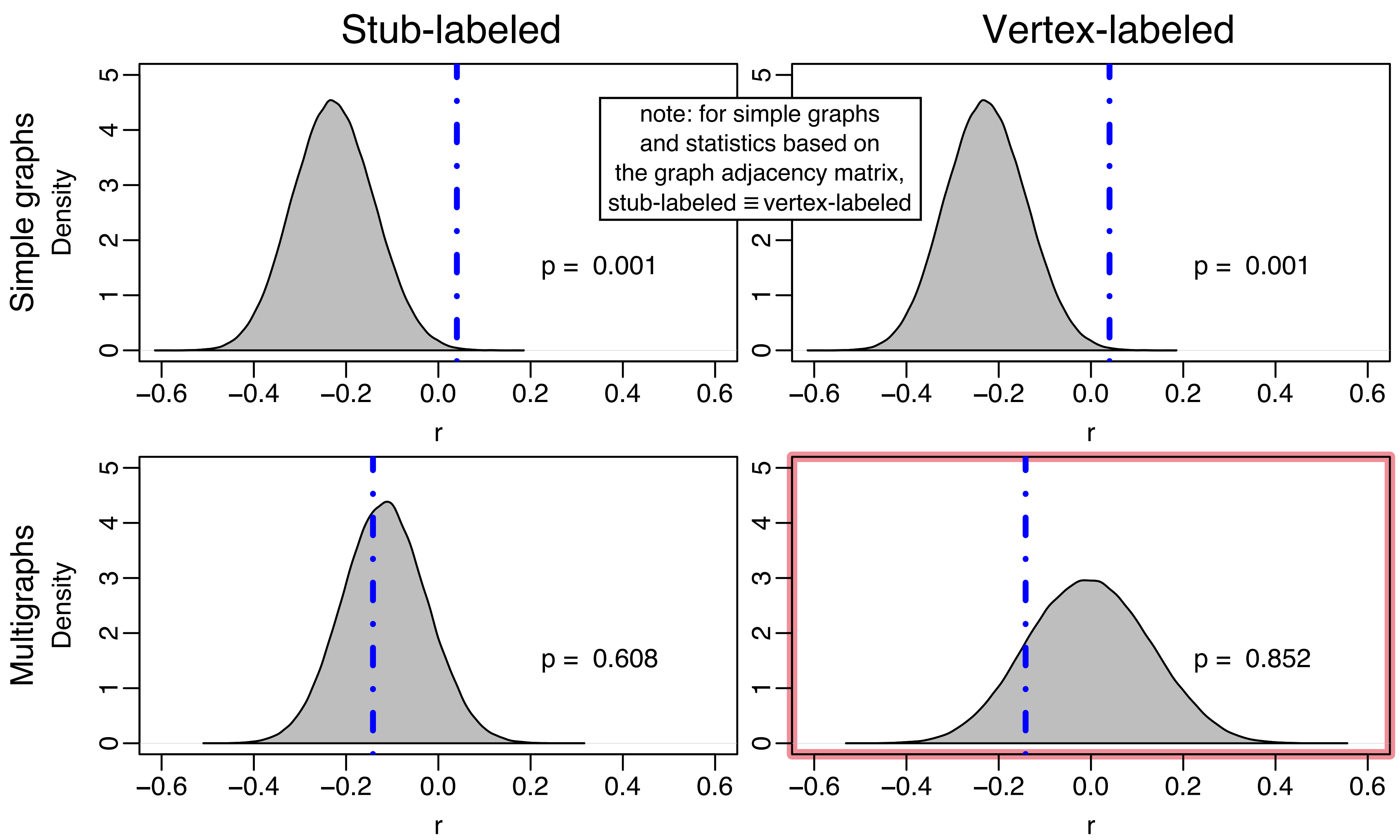
NONE of these is centered at zero.
Correct space is *meaningfully* different.

Uniform sampling means we can compare empirical value to null distribution to draw scientific conclusions.

The choice of graph space matters—careful choice & sampling can flip conclusions!

Fosdick, Larremore, Nishimura, Ugander. SIAM Review 60 (2) 315-355 (2018).

Assortative pairing of barn swallows



Sanity check:
should be = for simple

NONE of these is centered at zero.
Correct space is *meaningfully* different.

Uniform sampling means we can compare empirical value to null distribution to draw scientific conclusions.

Another example: Clauset 4:71-75

The choice of graph space matters—careful choice & sampling can flip conclusions!

Fosdick, Larremore, Nishimura, Ugander. SIAM Review 60 (2) 315-355 (2018).

So, why this tangent?

Random graph models are recipes. You pick the parameters and cook up an ensemble of networks.

How much of [cool property] that we observe is actually just typical of networks from [particular ensemble]?

Null hypothesis: none.
Test requires uniform distribution.

olive the dog

So, why this tangent?

Random graph models are recipes. You pick the parameters and cook up an ensemble of networks.

How much of [cool property] that we observe is actually just typical of networks from [particular ensemble]?

Null hypothesis: none.
Test requires uniform distribution.

olive the dog

What if we wrote down model whose non-uniformity over its ensemble was a *feature*?

Generative models for network structure

Generate the structure you wish to infer.

We like generative models because they open the door to inference:

Generative models for network structure

Generate the structure you wish to infer.

We like generative models because they open the door to inference:

GM + parameters —stochastic draw→ Data

Generative models for network structure

Generate the structure you wish to infer.

We like generative models because they open the door to inference:

GM + parameters —(stochastic draw)→ Data

GM + parameters ←(inference)— Data

Generative models for network structure

Generate the structure you wish to infer.

We like generative models because they open the door to inference:

GM + parameters —(stochastic draw)→ Data

GM + parameters ←(inference)— Data

In other words: let's write down a model whose ensemble's distribution is not uniform but **highly peaked** around networks with structures that we want to see.

The stochastic block model

GM + parameters —stochastic draw→ Data

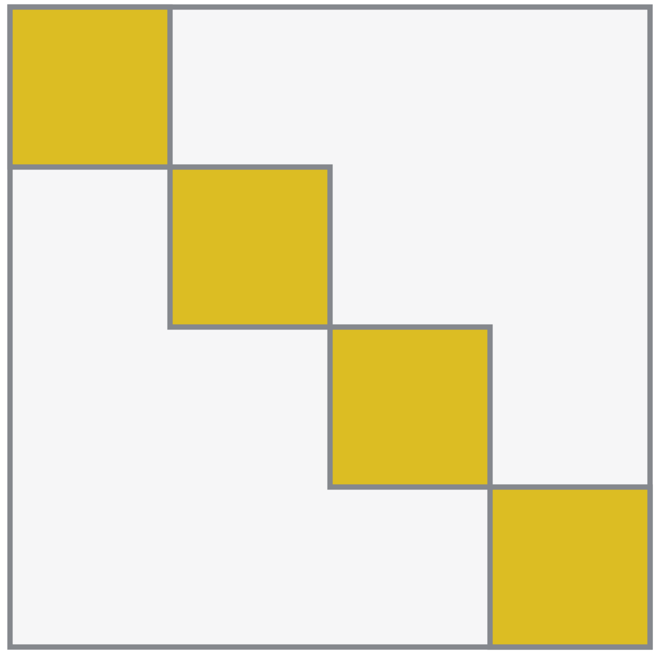
Assign each node to one of *B* blocks.

b_i

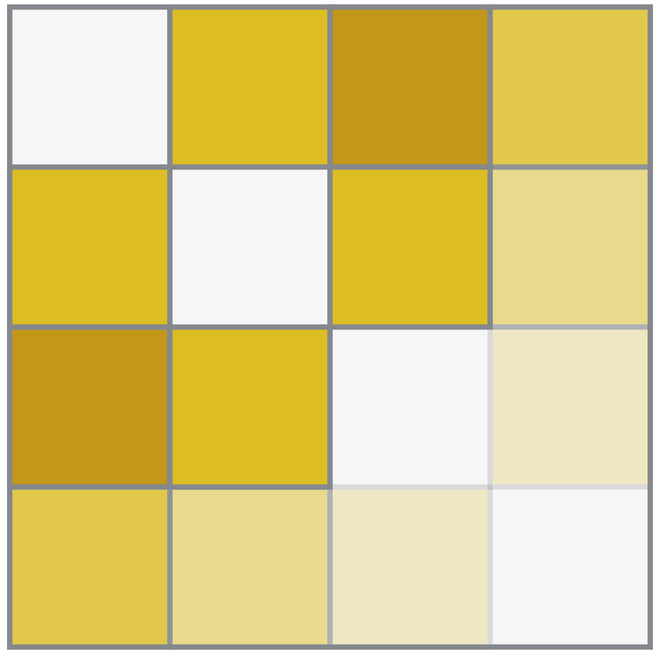
Let the probability that two nodes connect depend *only* on their blocks:

$$Pr(A_{ij}|b_i, b_j) = \omega_{b_i,b_j}$$

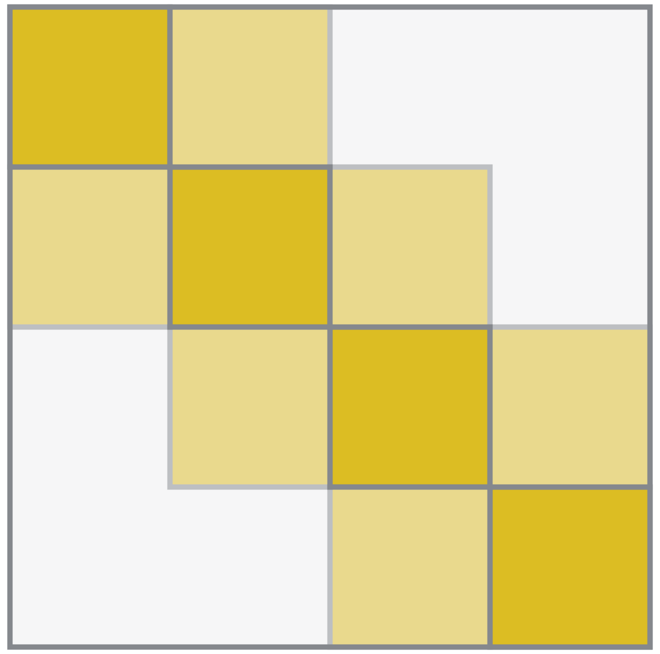
Then we can choose the matrix ω to have whatever structure we want!



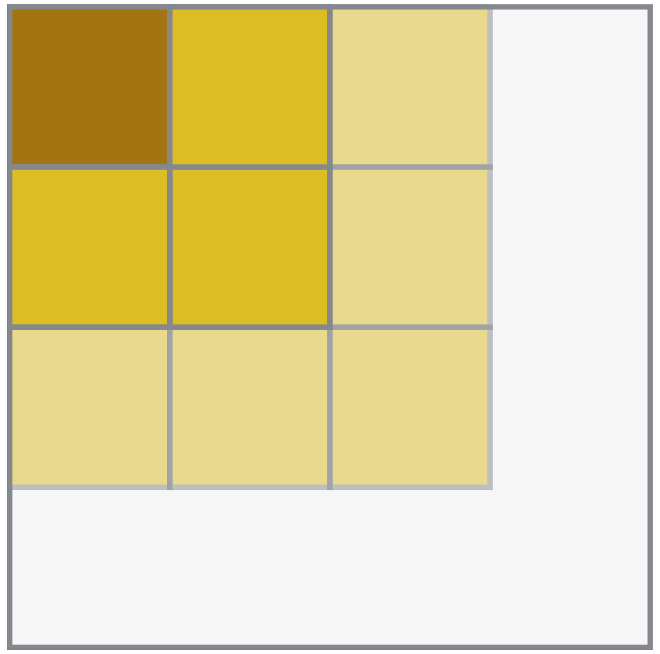
Assortative



Disassortative



Ordered



Core-periphery

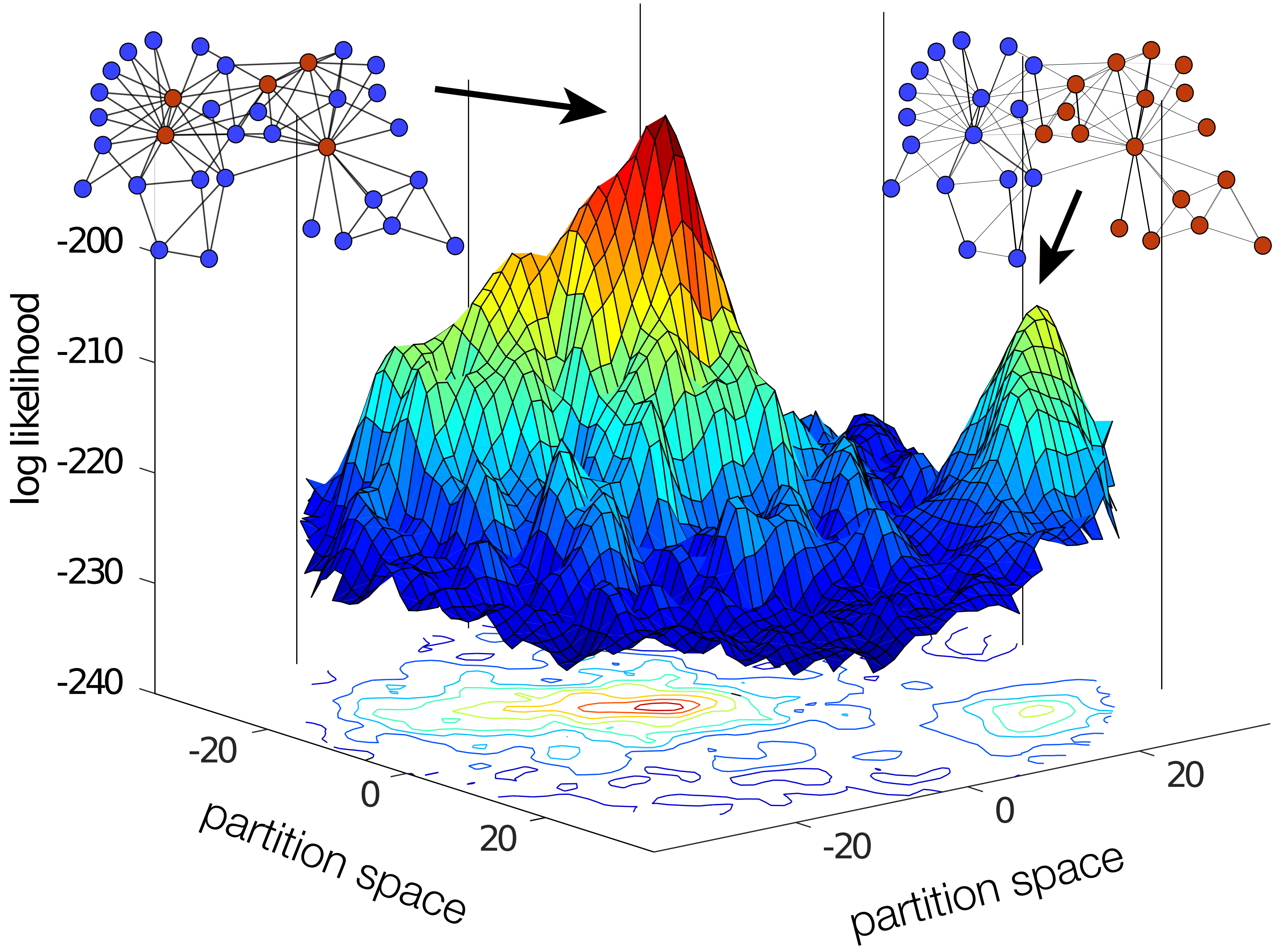
The stochastic block model

GM + parameters ⟵ inference ⟵ Data

When we run the generative process in reverse (aka inference), we find community structure.

This is nothing more than a statistically principled approach to **fitting a model to data.**

But instead of fitting a line to a scatter of (x,y) data, we're fitting a model for networks with community structure to data.



Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

IRL, not all friendships are reciprocated 😭
So let’s generate undirected AND directed edges:

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

IRL, not all friendships are reciprocated 😭
So let's generate undirected AND directed edges:

$$P(i \leftrightarrow j) \;=\; \alpha(r_i - r_j)$$

$$P(i \rightarrow j) \;=\; \beta(r_i - r_j)$$

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

IRL, not all friendships are reciprocated 😭
So let's generate undirected AND directed edges:

$$P(i \leftrightarrow j) \;=\; \alpha(r_i - r_j)$$

A gaussian centered at 0

$$P(i \rightarrow j) \;=\; \beta(r_i - r_j)$$

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

IRL, not all friendships are reciprocated 😭
So let's generate undirected AND directed edges:

$$P(i \leftrightarrow j) \;=\; \alpha(r_i - r_j)$$

A gaussian centered at 0

$$P(i \rightarrow j) \;=\; \beta(r_i - r_j)$$

Fourier cosine series, keeping five terms & squaring to enforce nonnegativity, plus an additional Gaussian peak at the origin.

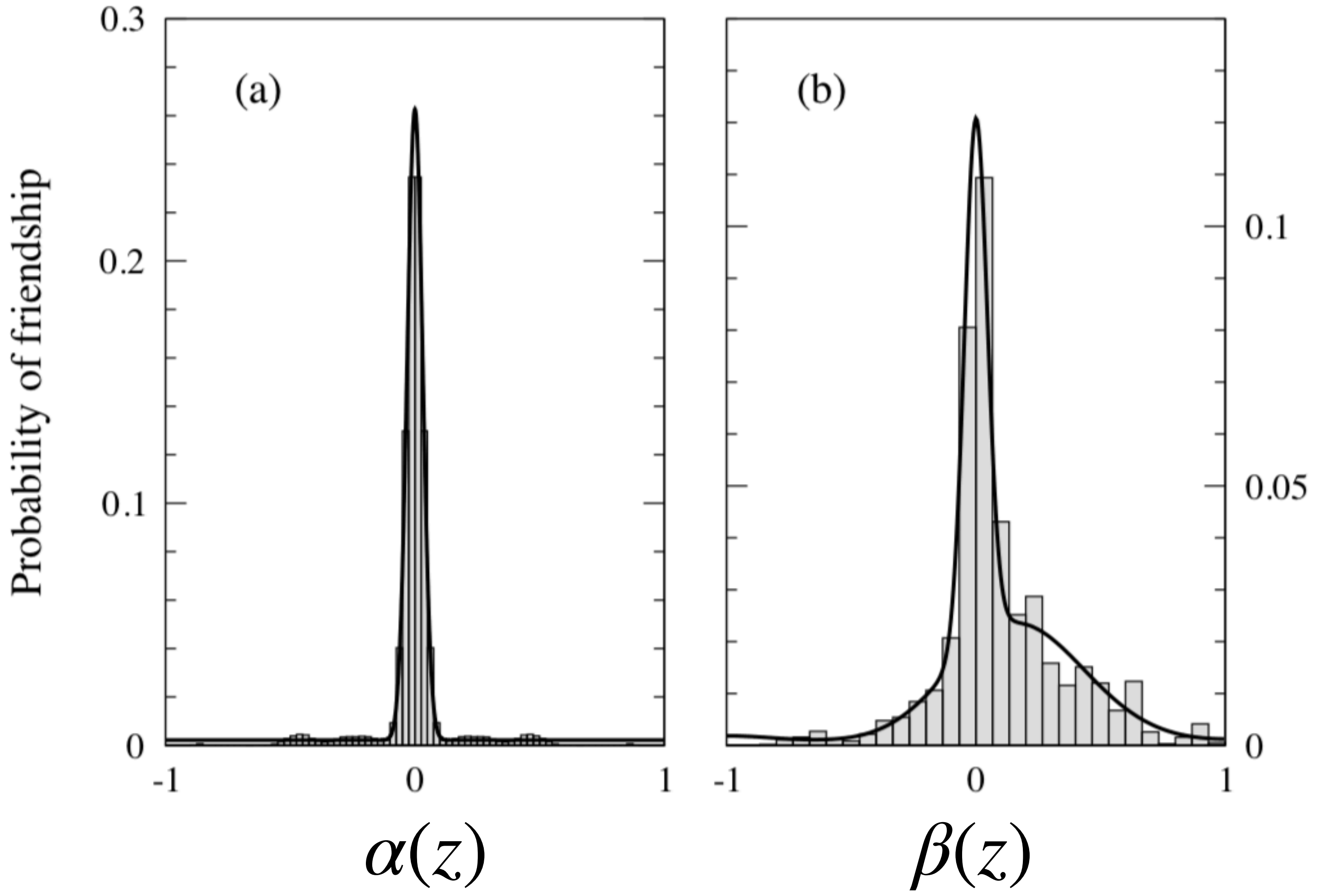
Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

Generative model:
Generate the patterns that you want to identify.

Create N nodes.
Assign each node an integer rank *r*, from 1 to N.

IRL, not all friendships are reciprocated 😭
So let's generate undirected AND directed edges:



$$P(i \leftrightarrow j) \;=\; \alpha(r_i - r_j)$$

A gaussian centered at 0

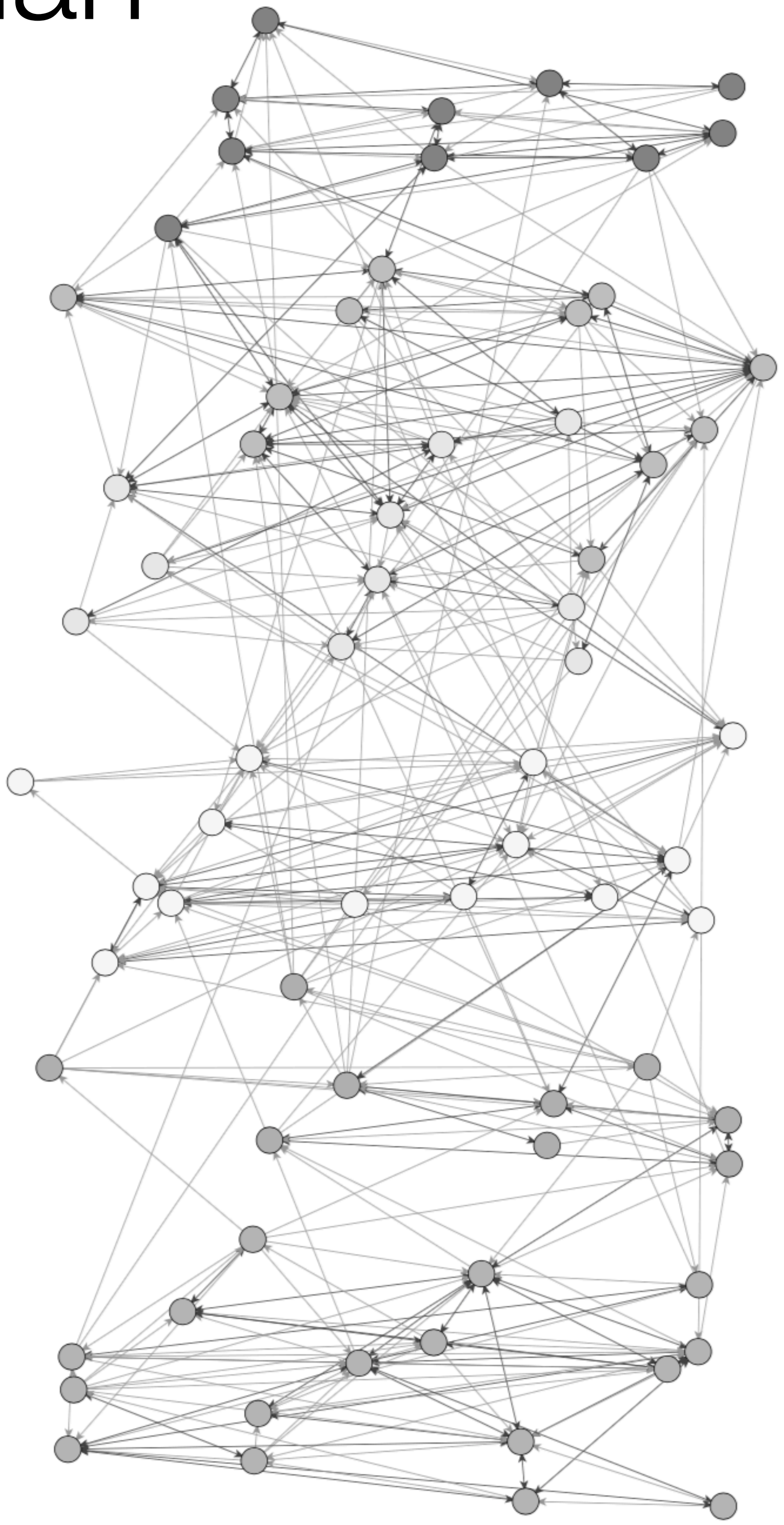
$$P(i \rightarrow j) \;=\; \beta(r_i - r_j)$$

Fourier cosine series, keeping five terms & squaring to enforce nonnegativity, plus an additional Gaussian peak at the origin.

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

GM + parameters ← *inference* — Data



Inferred parameters of people's attachment preferences & ranks.

- Identified the need to learn from reciprocated friendships.
- Found that in AddHealth data, teens link to others of *nearby* social status.

Ball, Newman. Network Science 1, 16-30 (2013)

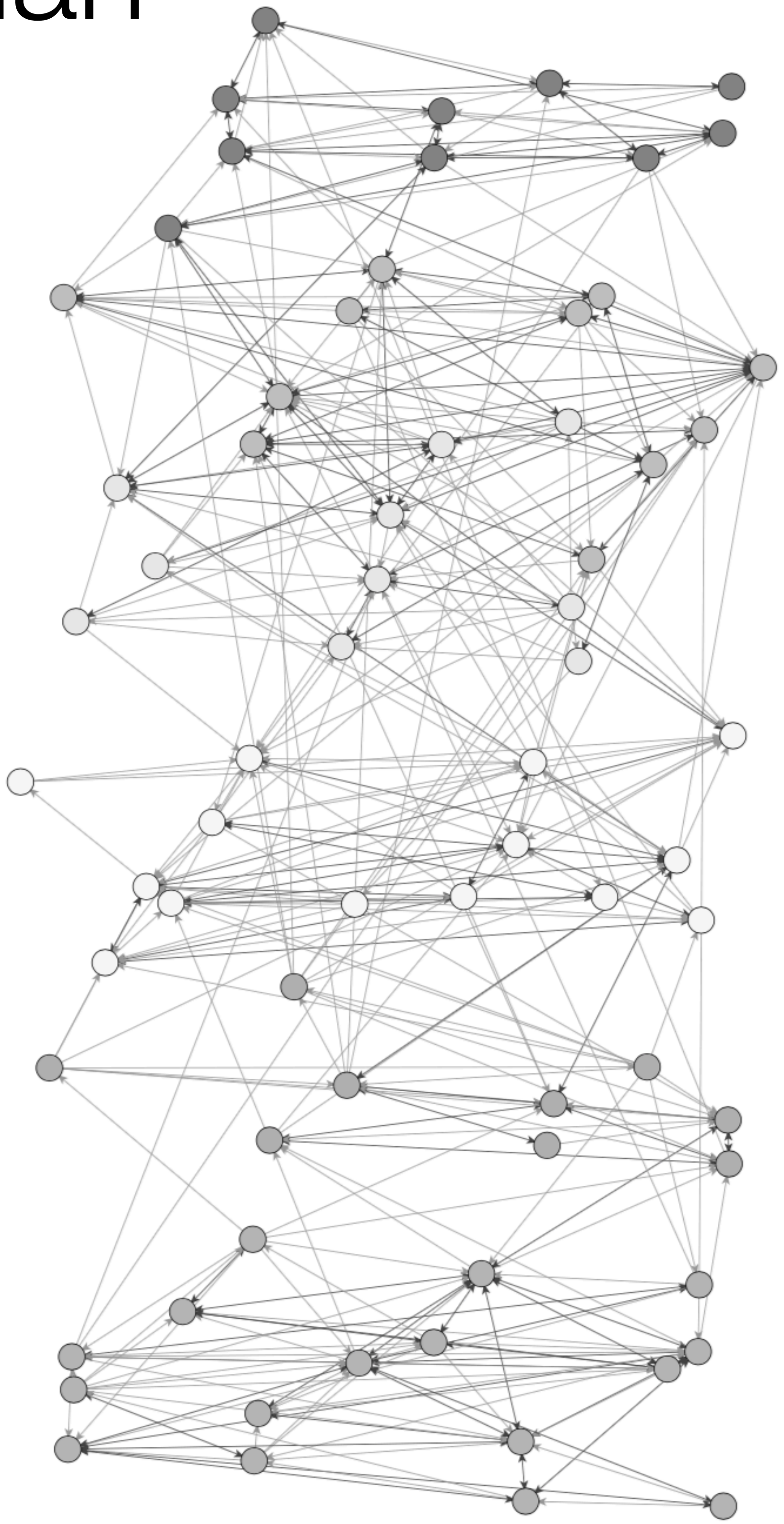
Embeddings and Orderings 4: Ball & Newman

GM + parameters ← inference ← Data

α

β

ranks



Inferred parameters of people's attachment preferences & ranks.

- Identified the need to learn from reciprocated friendships.
- Found that in AddHealth data, teens link to others of *nearby* social status.

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 4: Ball & Newman

GM + parameters

α

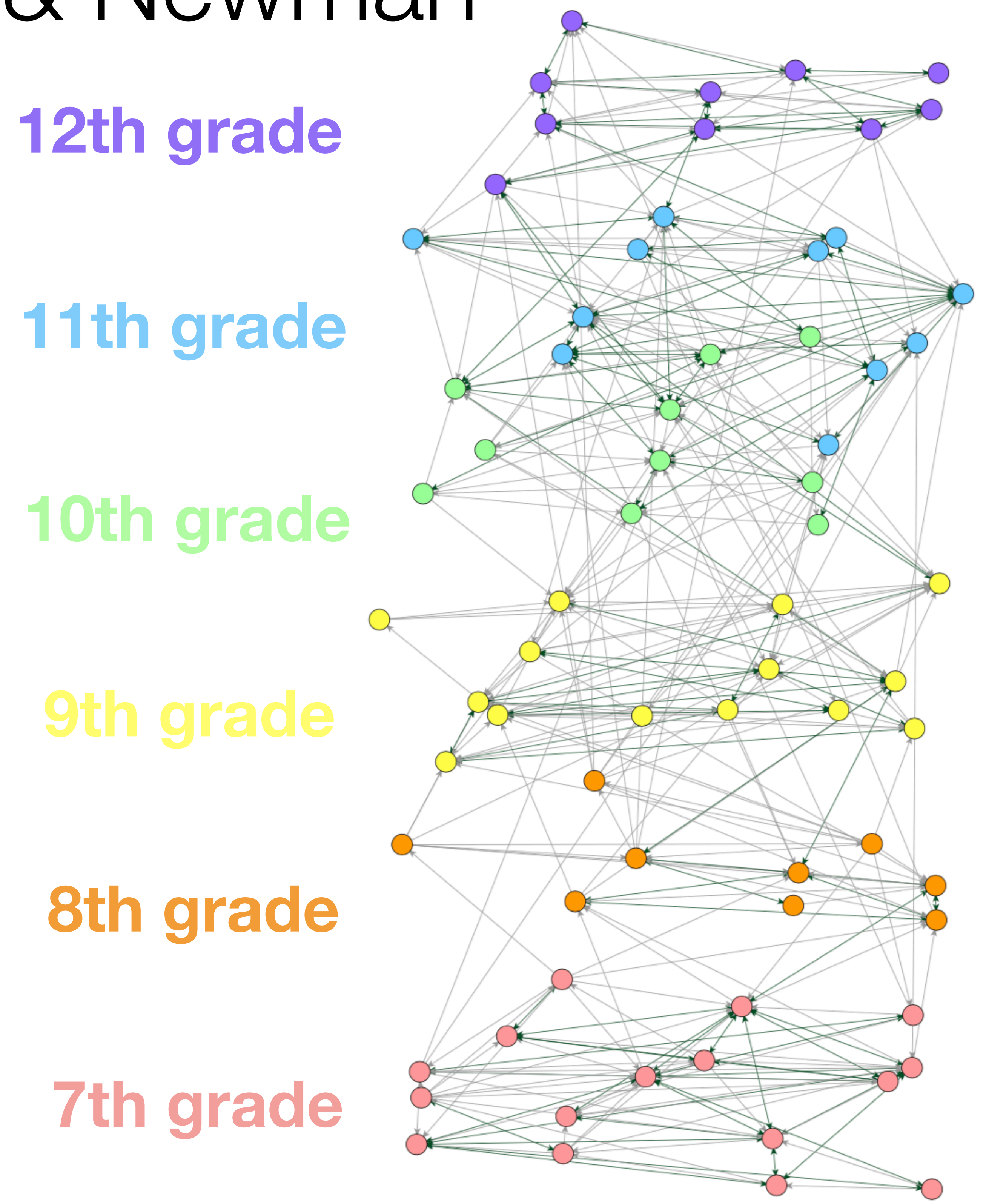
β

ranks

← inference — Data

Inferred parameters of people's attachment preferences & ranks.

- Identified the need to learn from reciprocated friendships.
- Found that in AddHealth data, teens link to others of *nearby* social status.



Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 5: Niche Models

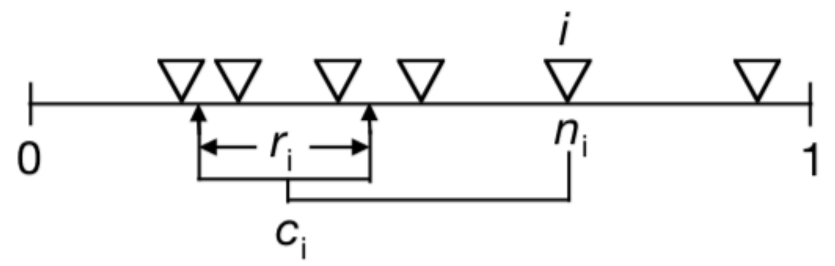


Figure 1 Diagram of the niche model. Each of ***S*** species (for example, ***S*** = 6, each shown as an inverted triangle) is assigned a 'niche value' parameter (n_i) drawn uniformly from the interval [0,1]. Species ***i*** consumes all species falling in a range (r_i) that is placed by uniformly drawing the centre of the range (c_i) from [r_i/2, n_i]. This permits looping and cannibalism by allowing up to half of r_i to include values $\geqslant n_i$. The size of r_i is assigned by using a beta function to randomly draw values from [0,1] whose expected value is 2***C*** and then multiplying that value by n_i [expected $E(n_i) = 0.5$] to obtain the desired ***C***. A beta distribution with $\alpha = 1$ has the form $f(x|1, \beta) = \beta(1-x)^{\beta-1}$, $0 < x < 1$, 0 otherwise, and $E(X) = 1/(1+\beta)$. In this case, $x = 1-(1-y)^{1/\beta}$ is a random variable from the beta distribution if y is a uniform random variable and β is chosen to obtain the desired expected value. We chose this form because of its simplicity and ease of calculation. The fundamental generality of species i is measured by r_i. The number of species falling within r_i measures realized generality. Occasionally, model-generated webs contain completely disconnected species or trophically identical species. Such species are eliminated and replaced until the web is free of such species. The species with the smallest n_i has $r_i = 0$ so that every web has at least one basal species.

Embeddings and Orderings 5: Niche Models

Niche Models embed species in a latent space based on feeding preferences:

most species feed from narrow range in a 1-dim. space (~body size).

- Great for food webs. Inference models v slow for all but small networks.

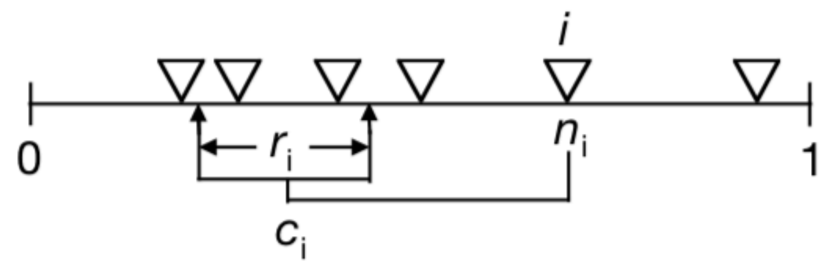


Figure 1 Diagram of the niche model. Each of $\boldsymbol{S}$ species (for example, $\boldsymbol{S} = 6$, each shown as an inverted triangle) is assigned a 'niche value' parameter (n_i) drawn uniformly from the interval [0,1]. Species $\boldsymbol{i}$ consumes all species falling in a range (r_i) that is placed by uniformly drawing the centre of the range (c_i) from [r_i/2, n_i]. This permits looping and cannibalism by allowing up to half of r_i to include values $\geqslant n_i$. The size of r_i is assigned by using a beta function to randomly draw values from [0,1] whose expected value is 2$\boldsymbol{C}$ and then multiplying that value by n_i [expected $E(n_i) = 0.5$] to obtain the desired $\boldsymbol{C}$. A beta distribution with $\alpha = 1$ has the form $f(x|1, \beta) = \beta(1-x)^{\beta-1}$, $0 < x < 1$, 0 otherwise, and $E(X) = 1/(1+\beta)$. In this case, $x = 1-(1-y)^{1/\beta}$ is a random variable from the beta distribution if y is a uniform random variable and β is chosen to obtain the desired expected value. We chose this form because of its simplicity and ease of calculation. The fundamental generality of species i is measured by r_i. The number of species falling within r_i measures realized generality. Occasionally, model-generated webs contain completely disconnected species or trophically identical species. Such species are eliminated and replaced until the web is free of such species. The species with the smallest n_i has $r_i = 0$ so that every web has at least one basal species.

Williams & Martinez. Nature 404.6774 (2000).

Embeddings and Orderings 5: Niche Models

Niche Models embed species in a latent space based on feeding preferences:

most species feed from narrow range in a 1-dim. space (~body size).

- Great for food webs. Inference models v slow for all but small networks.

Want more? Jen Dunne, Cris Moore

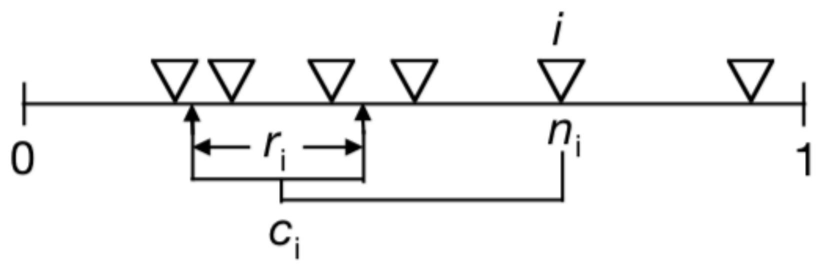


Figure 1 Diagram of the niche model. Each of ***S*** species (for example, ***S*** = 6, each shown as an inverted triangle) is assigned a 'niche value' parameter (n_i) drawn uniformly from the interval [0,1]. Species ***i*** consumes all species falling in a range (r_i) that is placed by uniformly drawing the centre of the range (c_i) from [r_i/2, n_i]. This permits looping and cannibalism by allowing up to half of r_i to include values $\geqslant n_i$. The size of r_i is assigned by using a beta function to randomly draw values from [0,1] whose expected value is 2***C*** and then multiplying that value by n_i [expected $E(n_i) = 0.5$] to obtain the desired ***C***. A beta distribution with $\alpha = 1$ has the form $f(x|1, \beta) = \beta(1-x)^{\beta-1}$, $0 < x < 1$, 0 otherwise, and $E(X) = 1/(1+\beta)$. In this case, $x = 1-(1-y)^{1/\beta}$ is a random variable from the beta distribution if y is a uniform random variable and β is chosen to obtain the desired expected value. We chose this form because of its simplicity and ease of calculation. The fundamental generality of species i is measured by r_i. The number of species falling within r_i measures realized generality. Occasionally, model-generated webs contain completely disconnected species or trophically identical species. Such species are eliminated and replaced until the web is free of such species. The species with the smallest n_i has $r_i = 0$ so that every web has at least one basal species.

Williams & Martinez. Nature 404.6774 (2000).

</methods>

<applications>

Many uses for the same techniques. cf regression

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in “space” or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

Treat the network like a means to an end; an intermediate data structure:

Useful division. Need groups so that we can assign treatments in an A/B test.
Simplification. Downstream regression model needs ranks or groups.

Following

I have a problem with academics using academic data to show something about academics.

9:27 AM - 20 Jun 2018

1 Retweet **4** Likes

1 1 4

Dan Larremore @DanLarremore · Jun 20

Replying to @OverheardAtSFI

Well there goes my research agenda. 😭

6

Structure and inequality in academic hiring

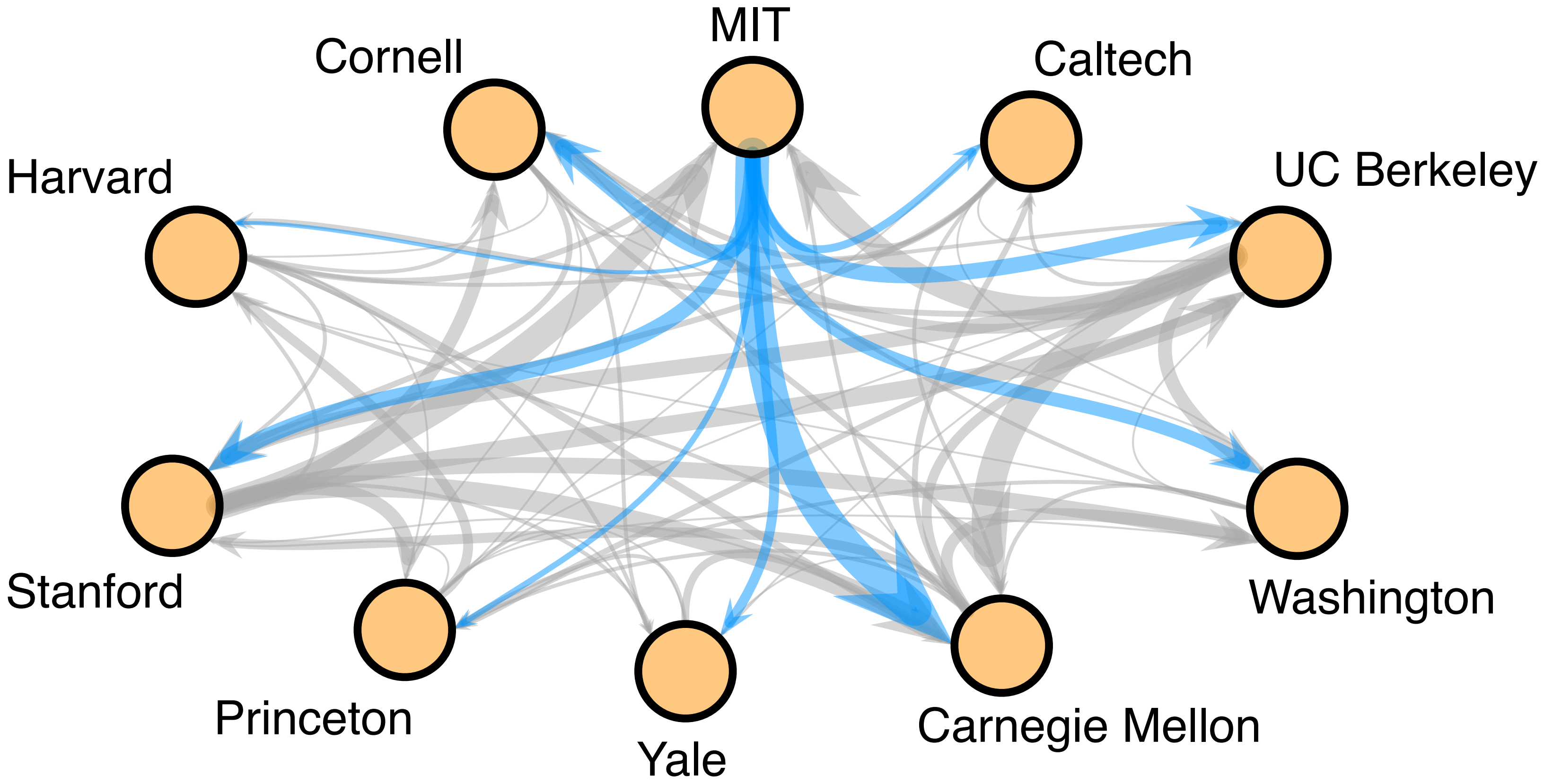
Collect the data (by hand 😭)

CVs of all US & Canadian tenure-track faculty in CS, Business, History: 2011-2013.

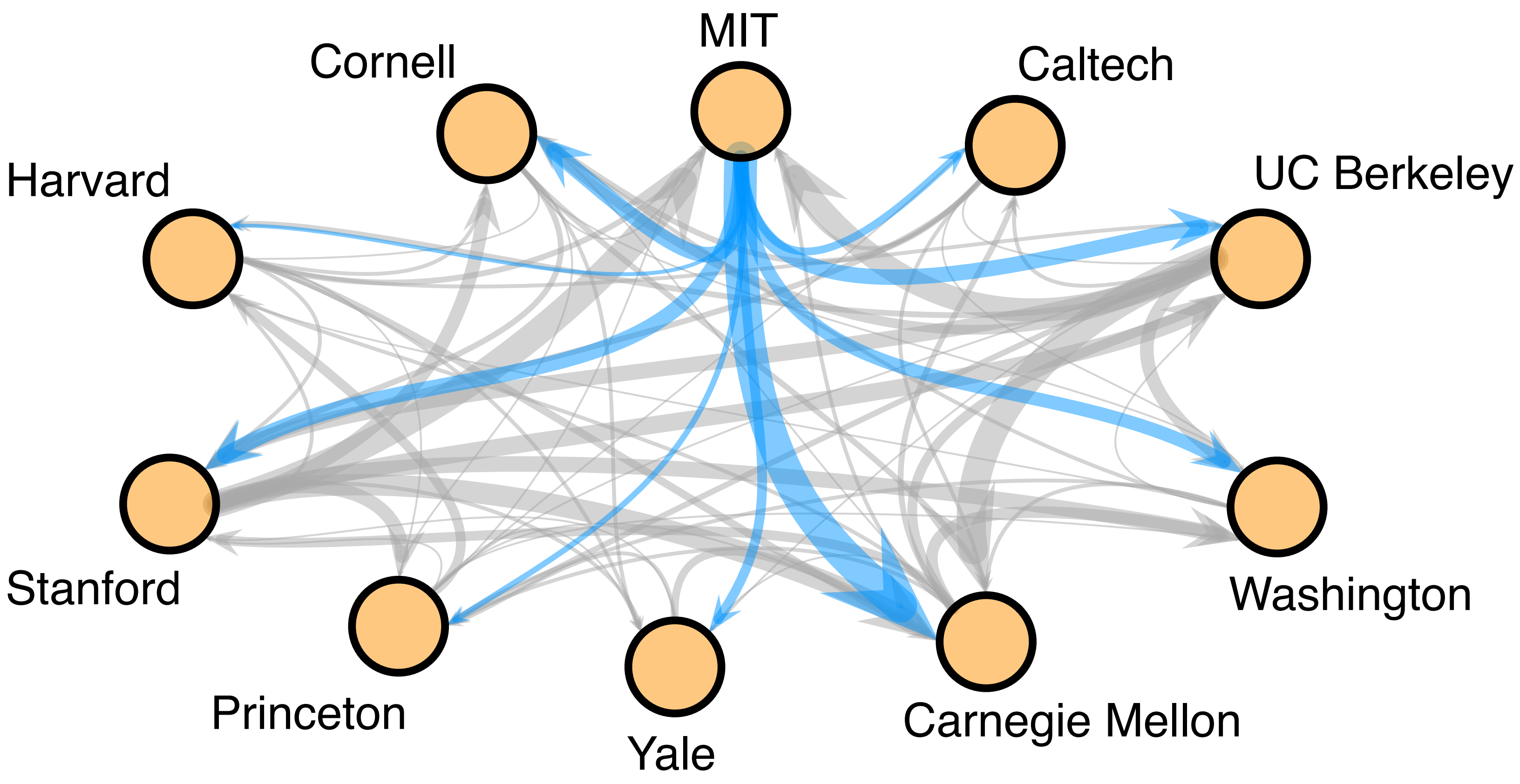
	Computer Science	Business	History
institutions	205	112	144
tenure-track faculty	5032	9336	4556
mean size	25	83	32
female	15%	22%	36%

total: **18,924** CVs

Faculty hiring networks



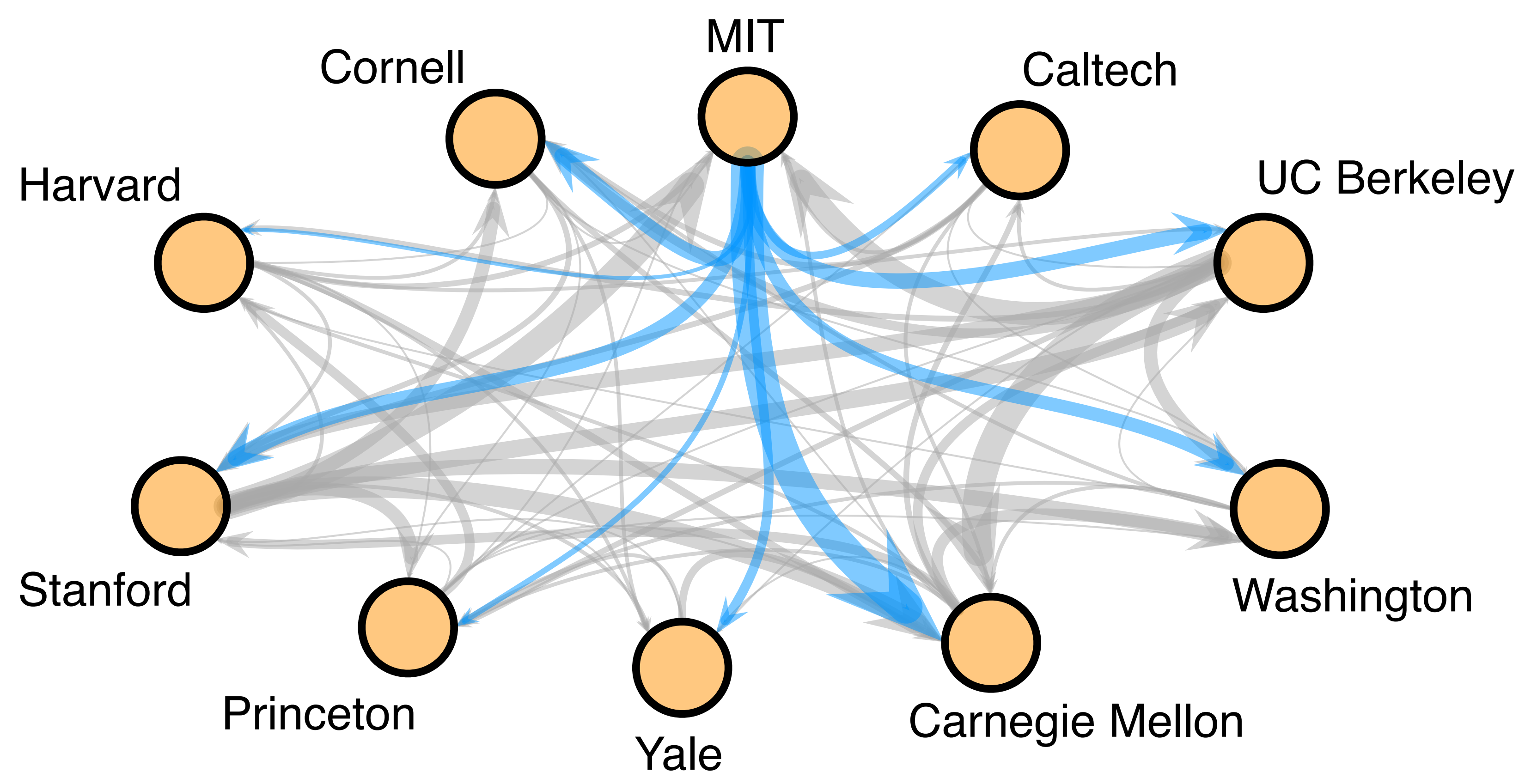
Faculty hiring networks



Premises:

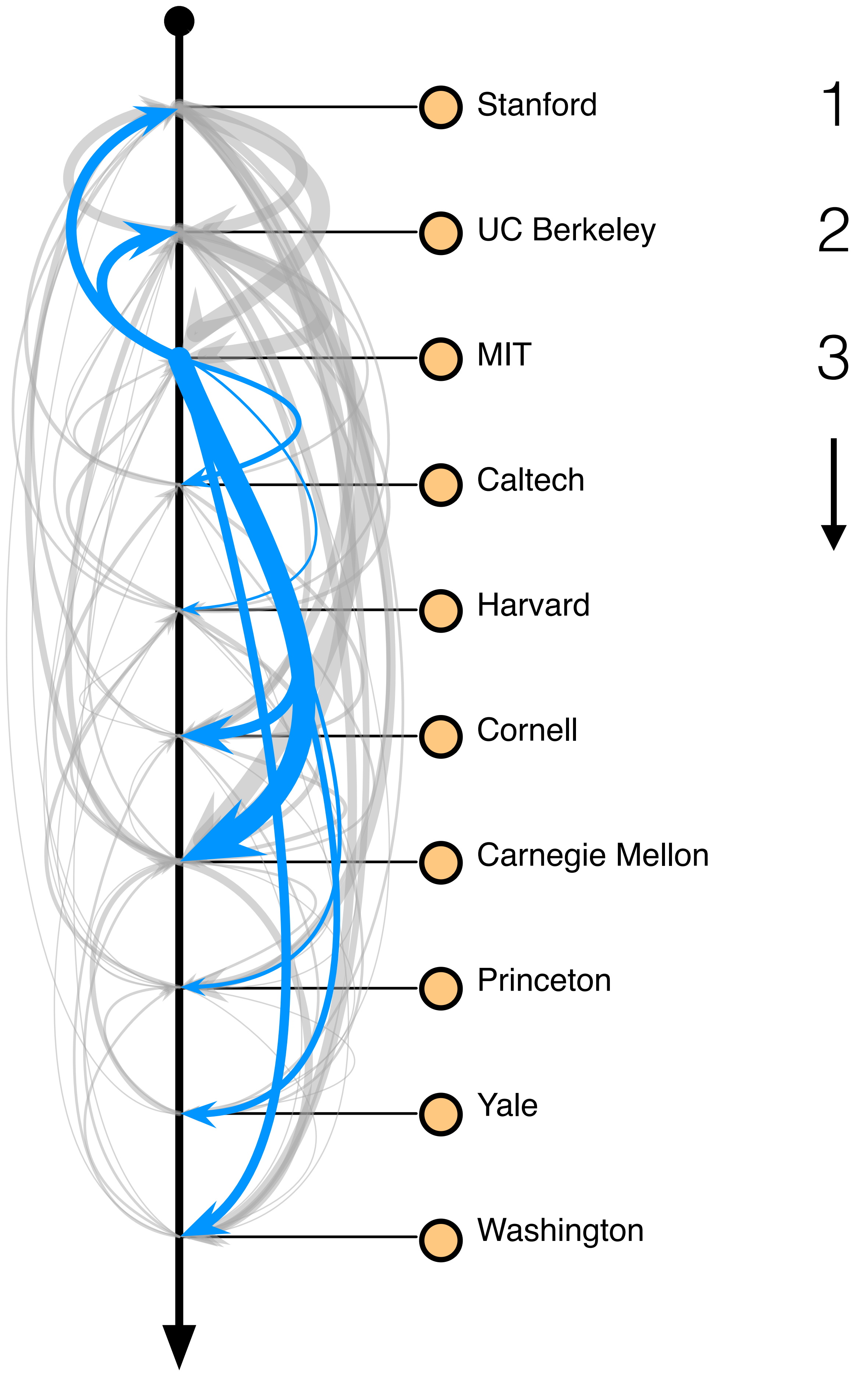
1. Each hiring committee wants to hire **the best**.
2. Entire network reveals **collective preferences**.

Faculty hiring networks



Premises:

1. Each hiring committee wants to hire **the best**.
2. Entire network reveals **collective preferences**.



Faculty hiring networks

systematic

90% of hiring movement is “down” the hierarchy

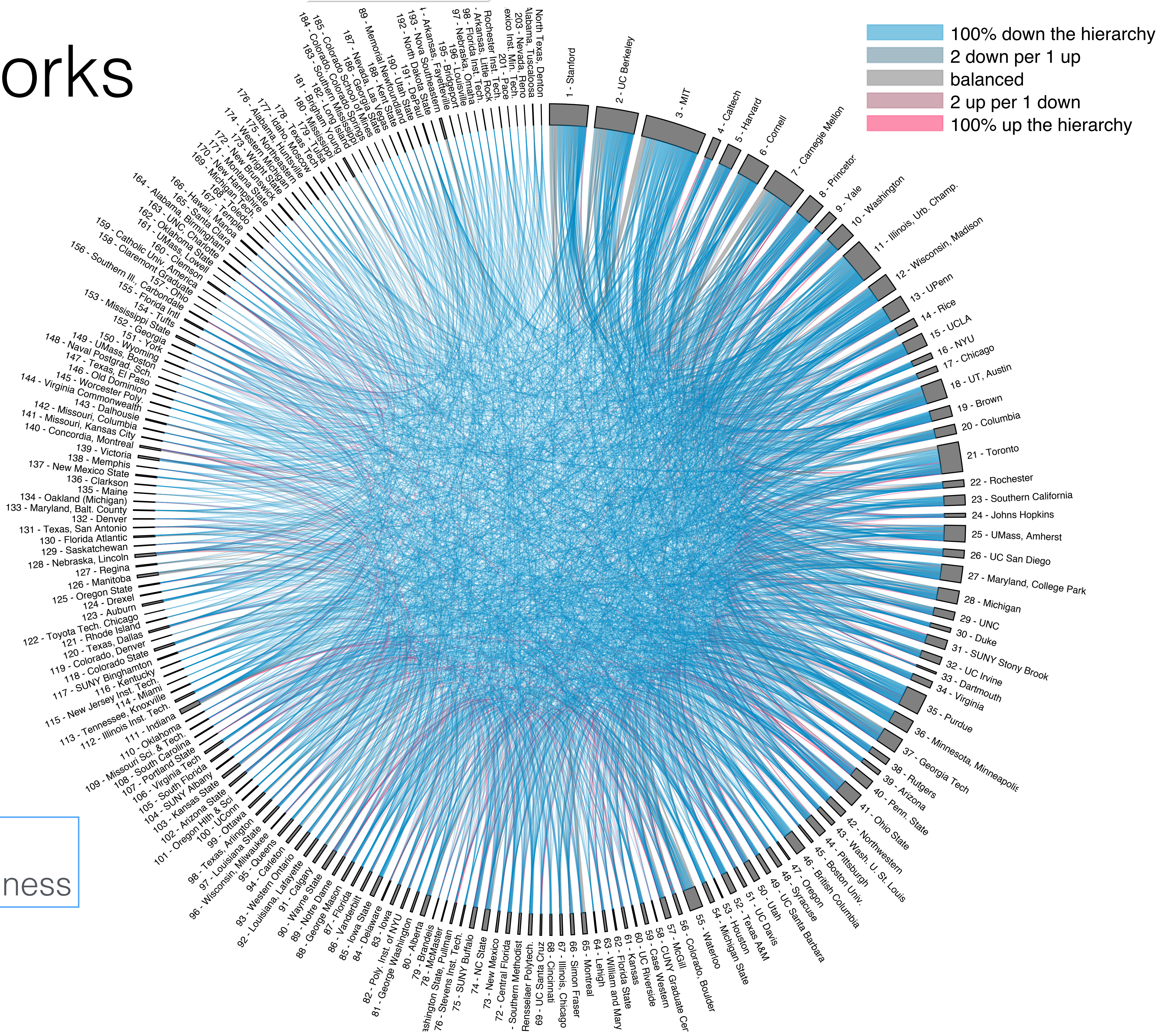
steep

< 7% of faculty have PhD from lower 75% of universities

biased

median change for women ~3 ranks worse than men

danlarremore.com/**faculty**/
explore 19,000 hires for History, CS, Business

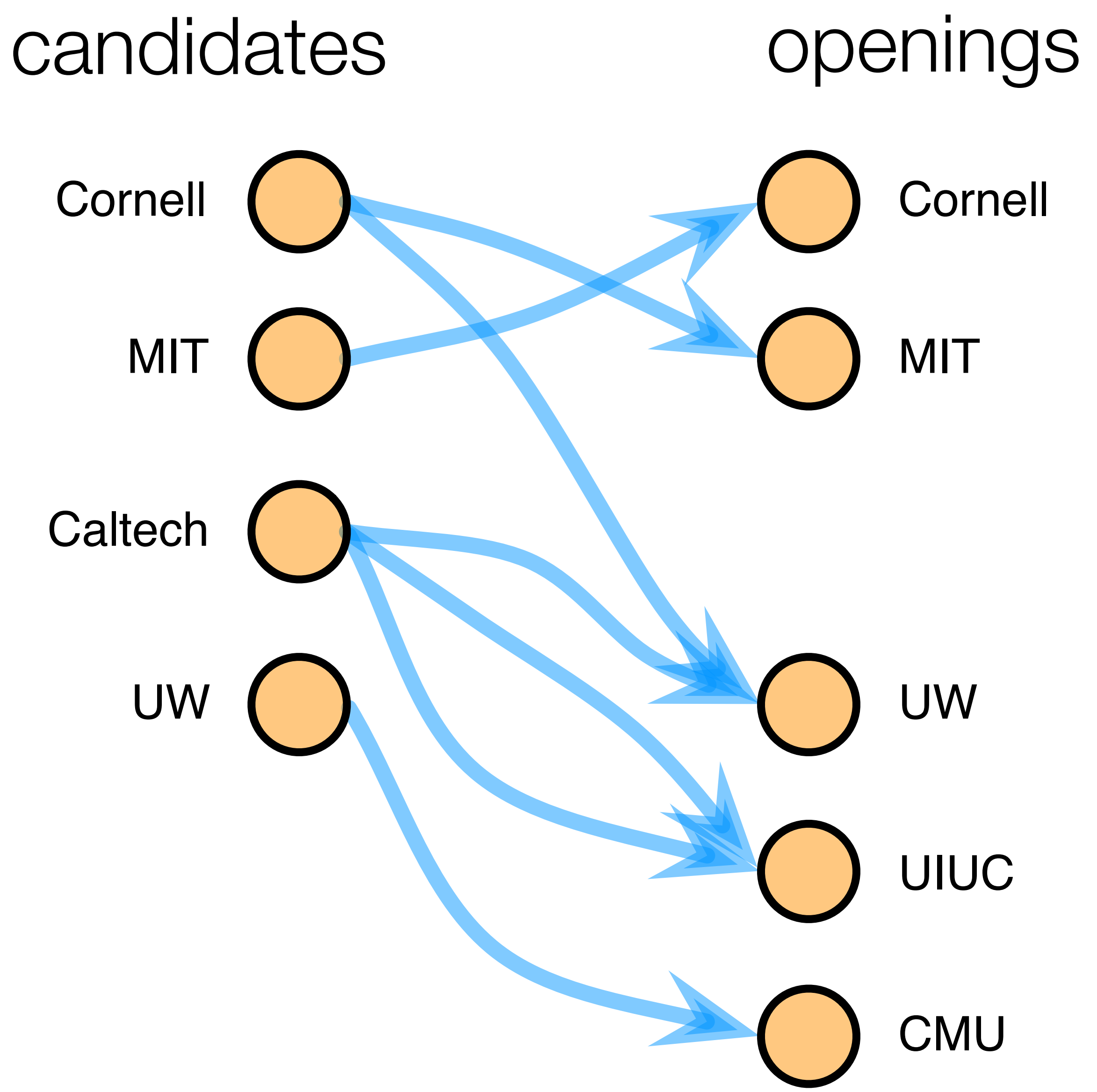


What else explains movement in this labor market?

Generative model:

prestige
productivity
postdoc experience

gender
geography

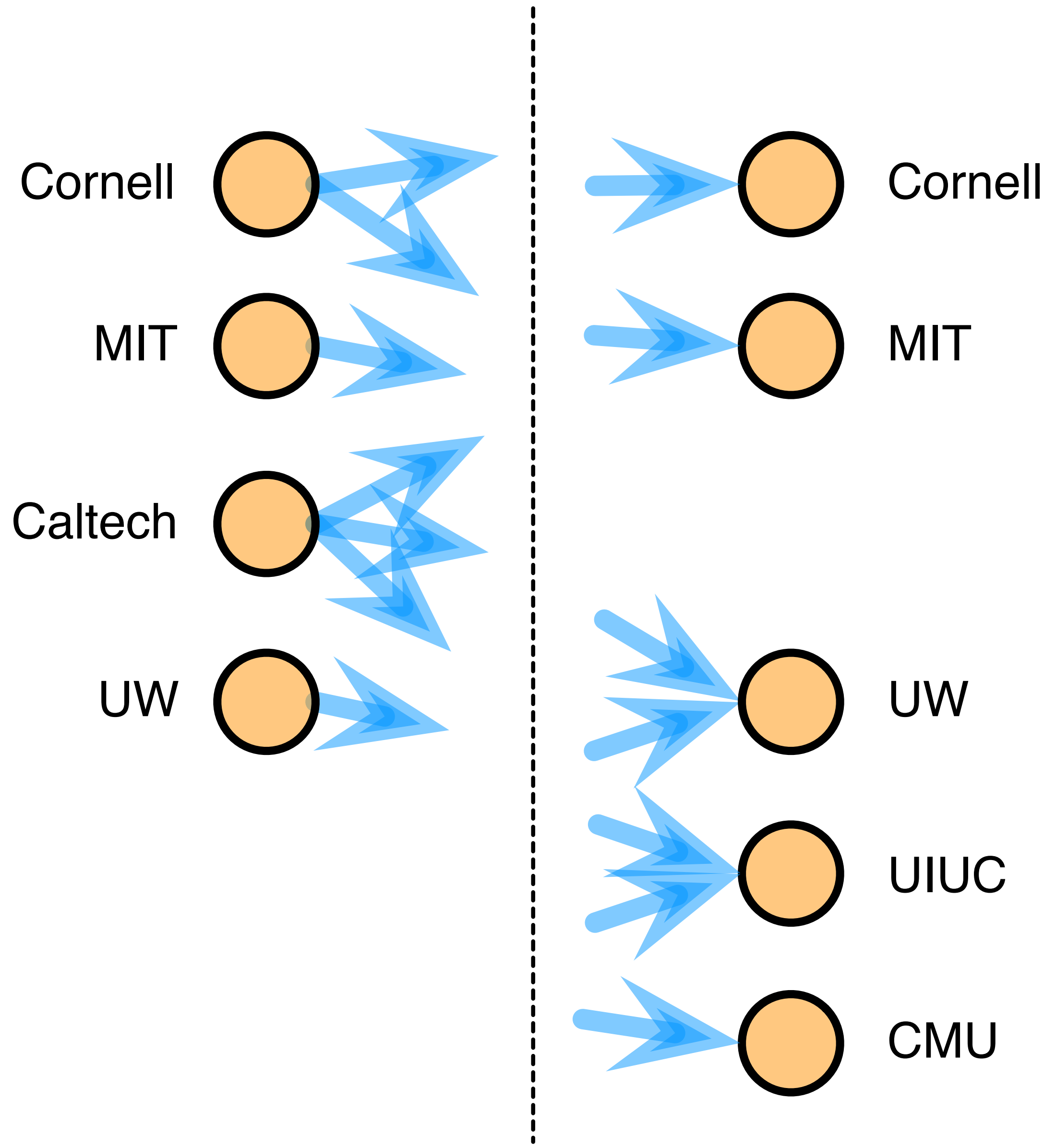


What else explains movement in this labor market?

Generative model:

prestige
productivity
postdoc experience

gender
geography



accurately generate the links!

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD

2. **Productivity**

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD
2. **Productivity**
3. **Prestige of Faculty Job**

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD
2. **Productivity**
3. **Prestige of Faculty Job**
4. **Postdoc experience** + **geography** (together)

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD
2. **Productivity**
3. **Prestige of Faculty Job**
4. **Postdoc experience** + **geography** (together)
5. **Gender**.

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD

2. **Productivity**

3. **Prestige of Faculty Job**

4. **Postdoc experience** + **geography** (together)

5. **Gender**.

Gender bias is not **uniformly**, systematically affecting all hires. But…

What else explains movement in this labor market?

1. **Prestige difference**: Faculty Job vs PhD
2. **Productivity**
3. **Prestige of Faculty Job**
4. **Postdoc experience** + **geography** (together)
5. **Gender**.

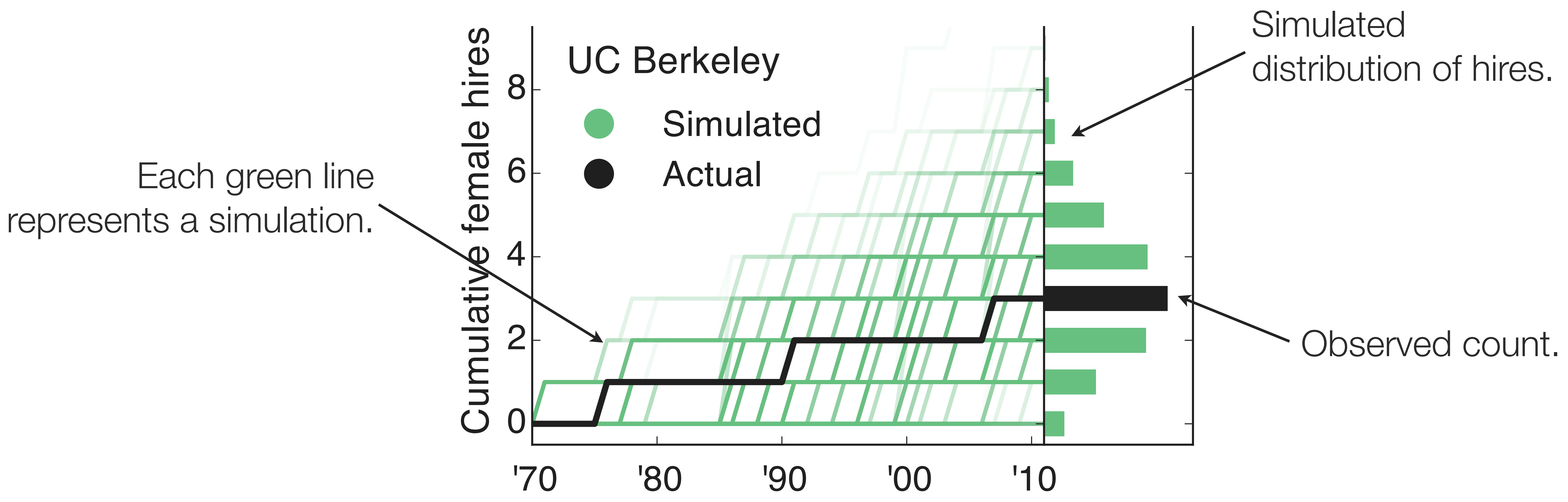
Gender bias is not **uniformly**, systematically affecting all hires. But…

a woman on the job market must have published ~1 additional paper to be placed the same as an equally qualified man.

Institution-level results

Using 40 years of actual hiring data, **simulate** hiring patterns for **each institution.**

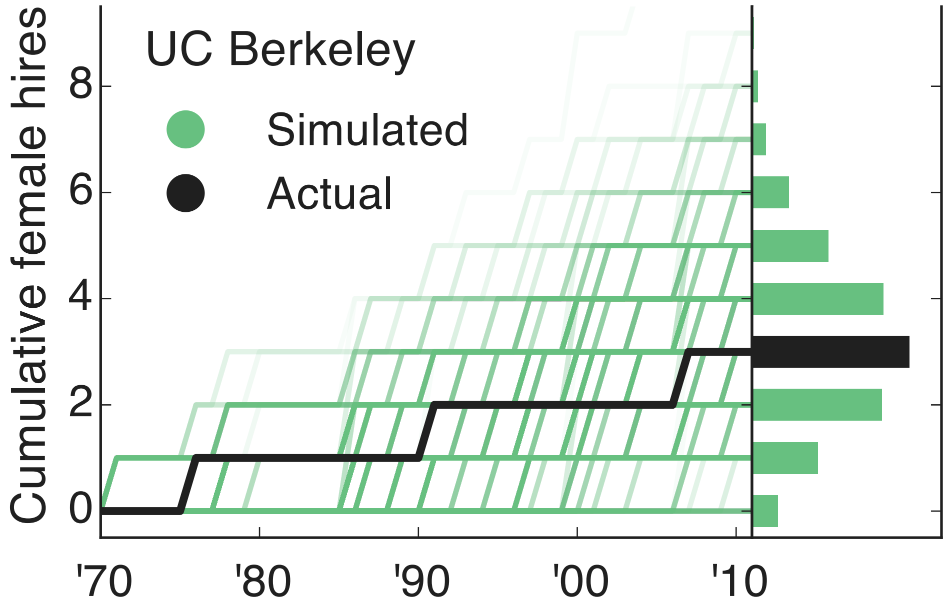
Compare actual vs. expected number of female hires.



Institution-level results

Using 40 years of actual hiring data, **simulate** hiring patterns for **each institution**.

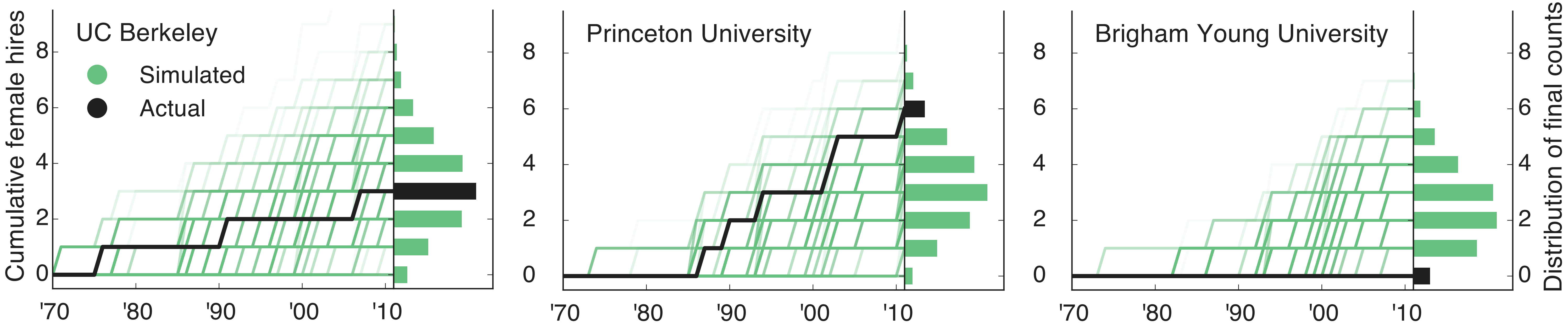
Compare actual vs. expected number of female hires.



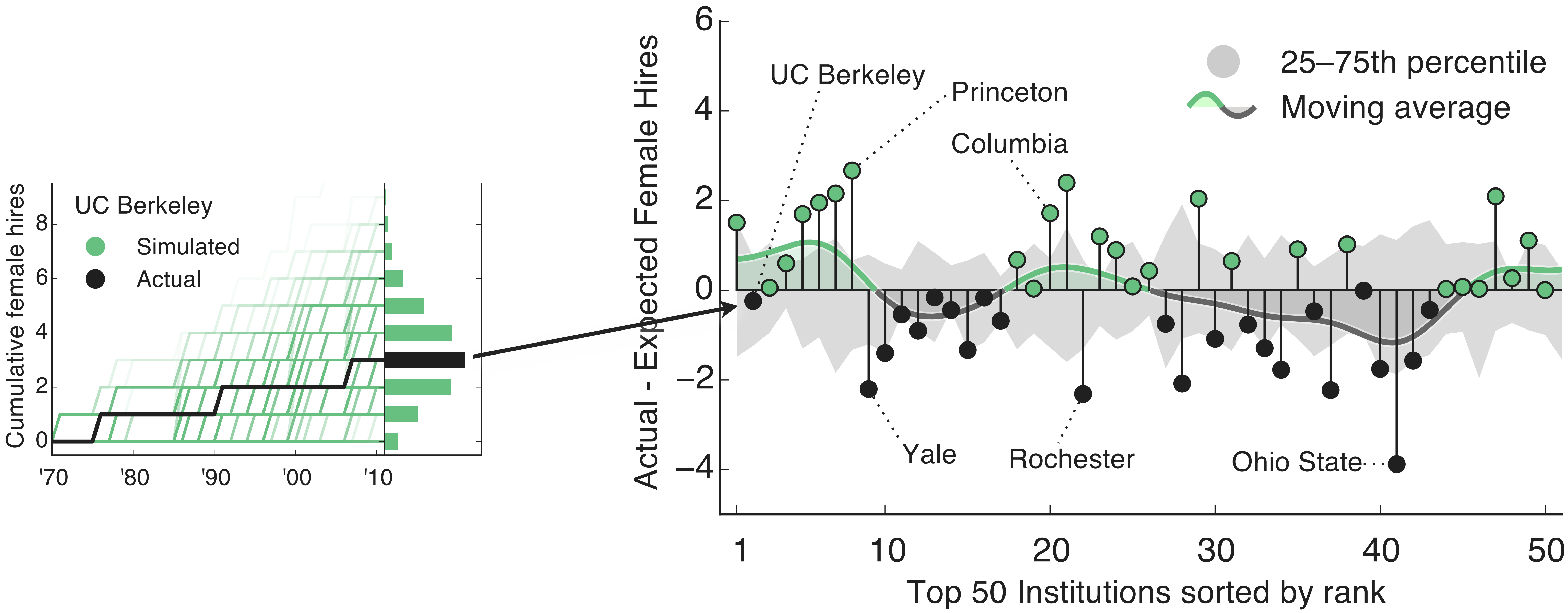
Institution-level results

Using 40 years of actual hiring data, **simulate** hiring patterns for **each institution**.

Compare actual vs. expected number of female hires.

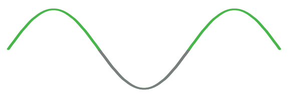


Institution-level results



Institution-level results

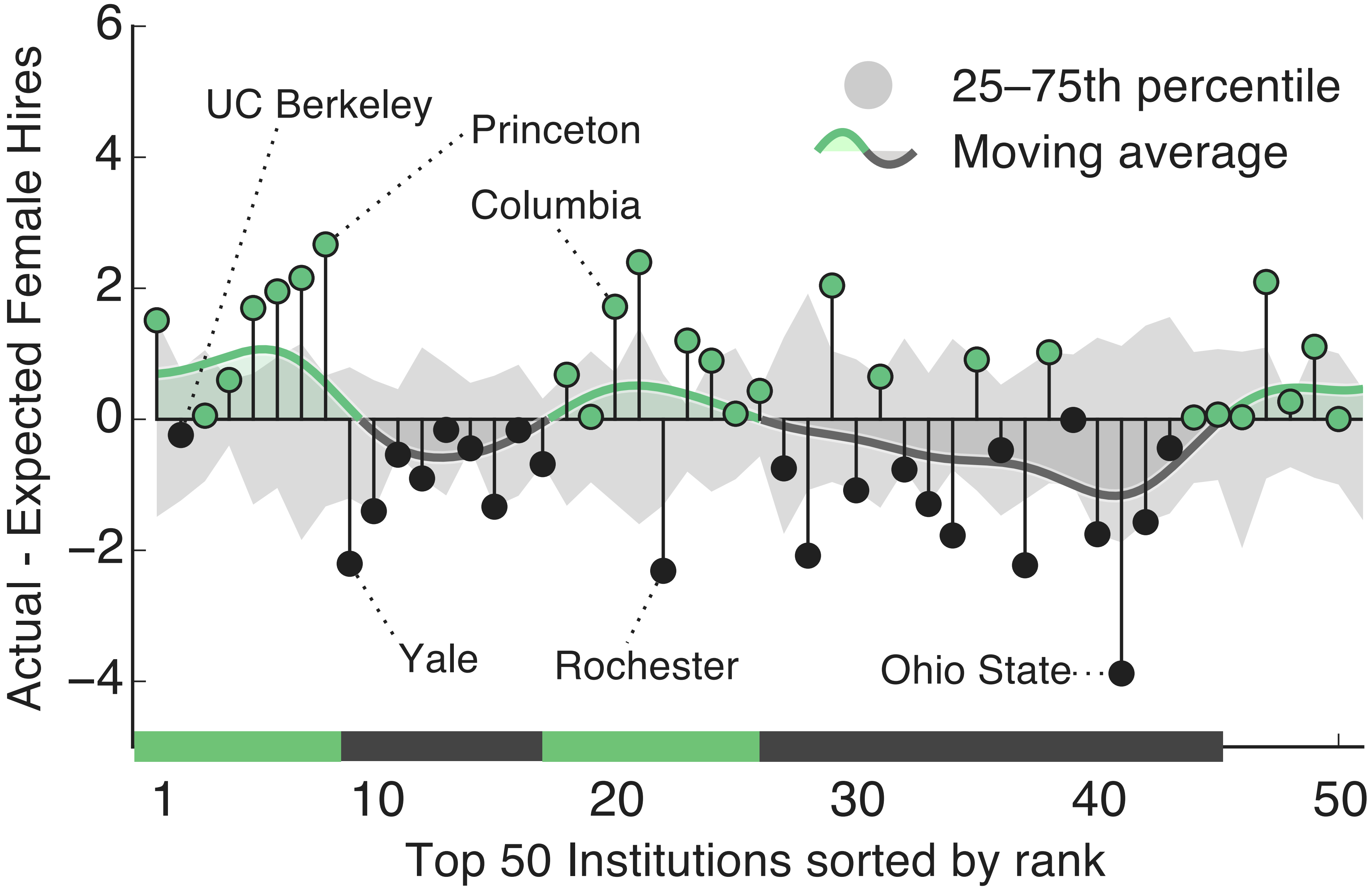
For the top 50 institutions,
we see an **oscillation**.



Why?

An interference effect?
Two distinct candidate pools?

Is it real?

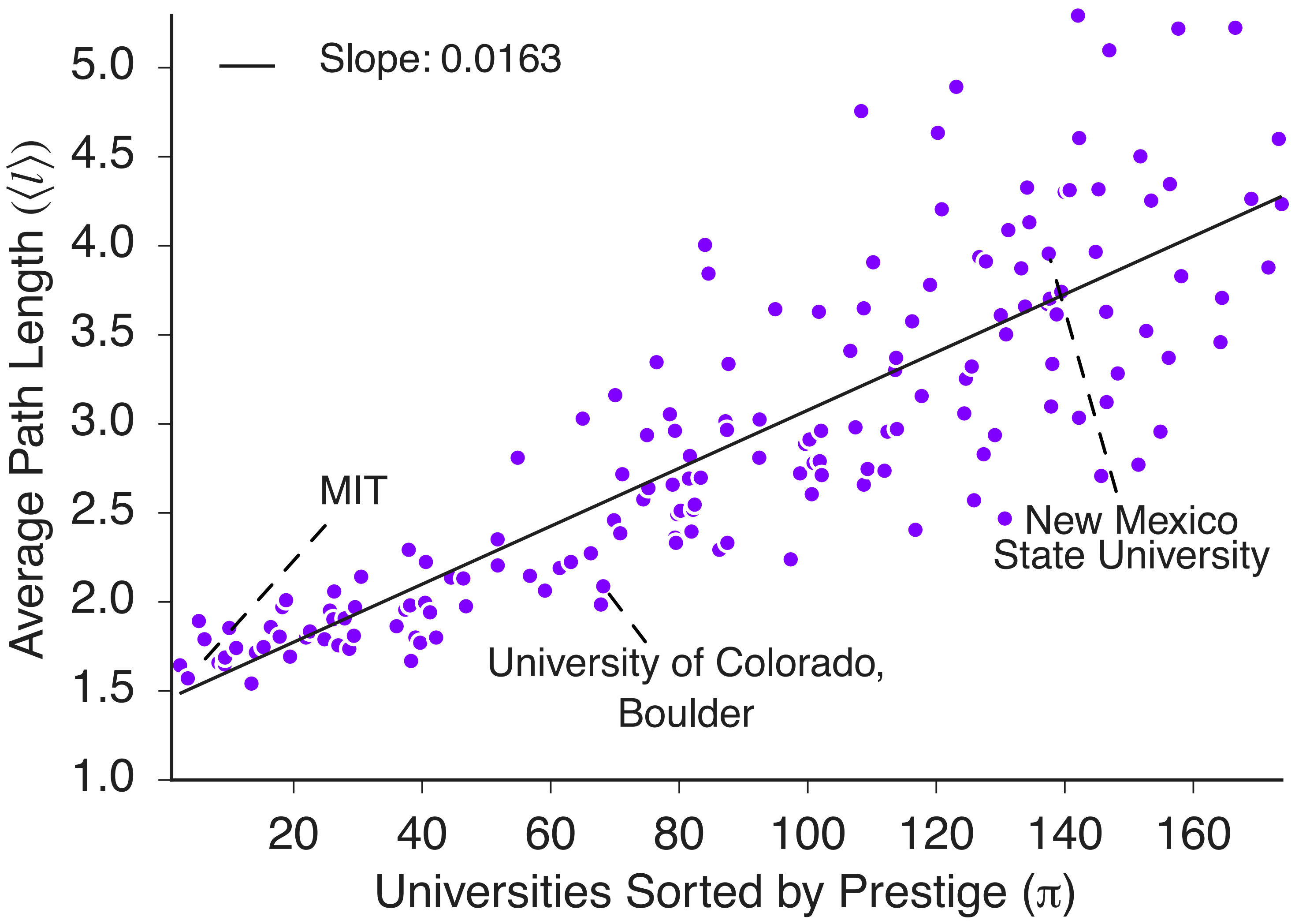


Does the structure of this network affect *ideas?*

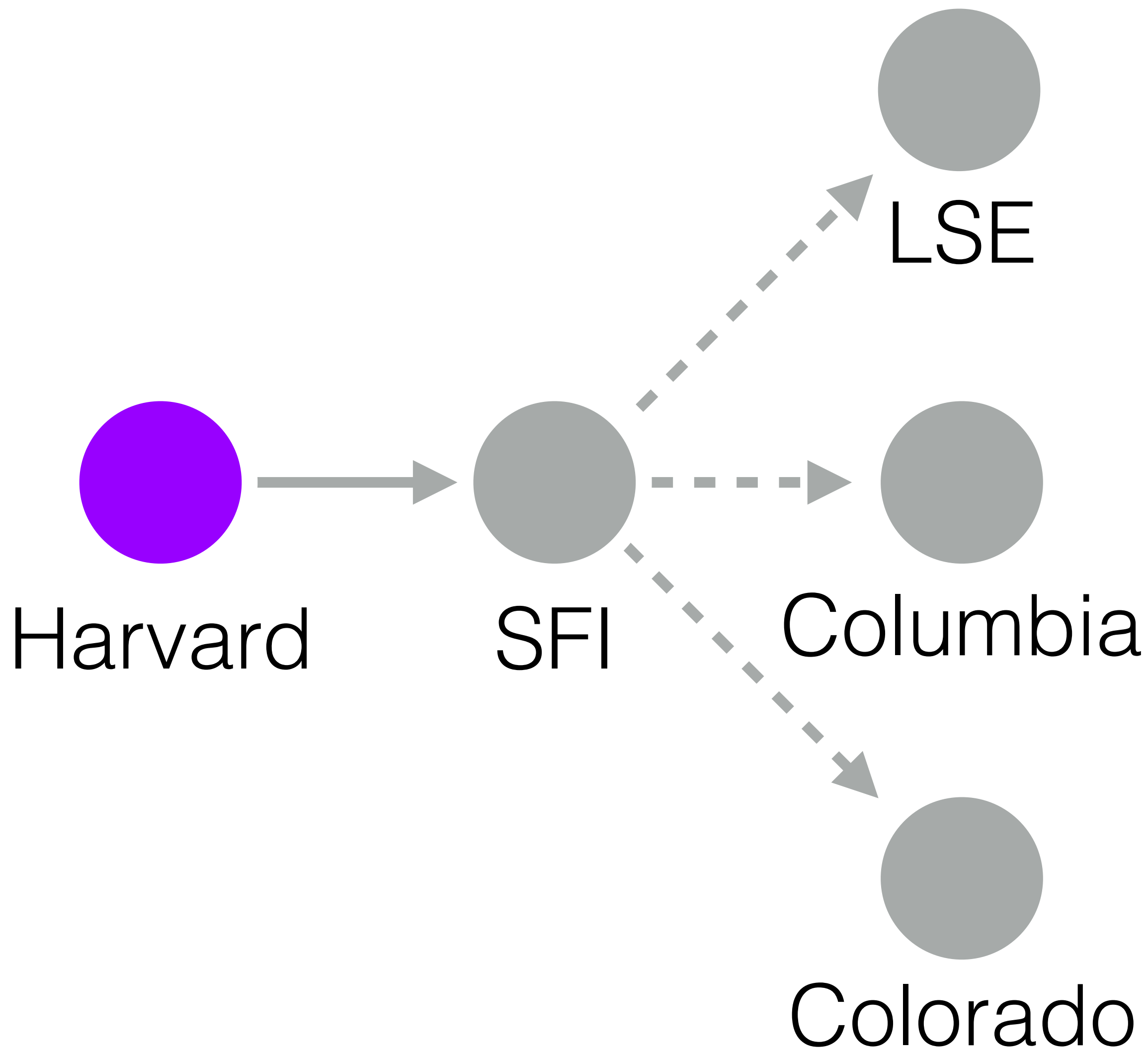
Prestigious institutions are **closer** to all other institutions.

What implications does this have for the **exchange** & **filtration** of ideas?

Does the prestige hierarchy lead to **epistemic inequality**?



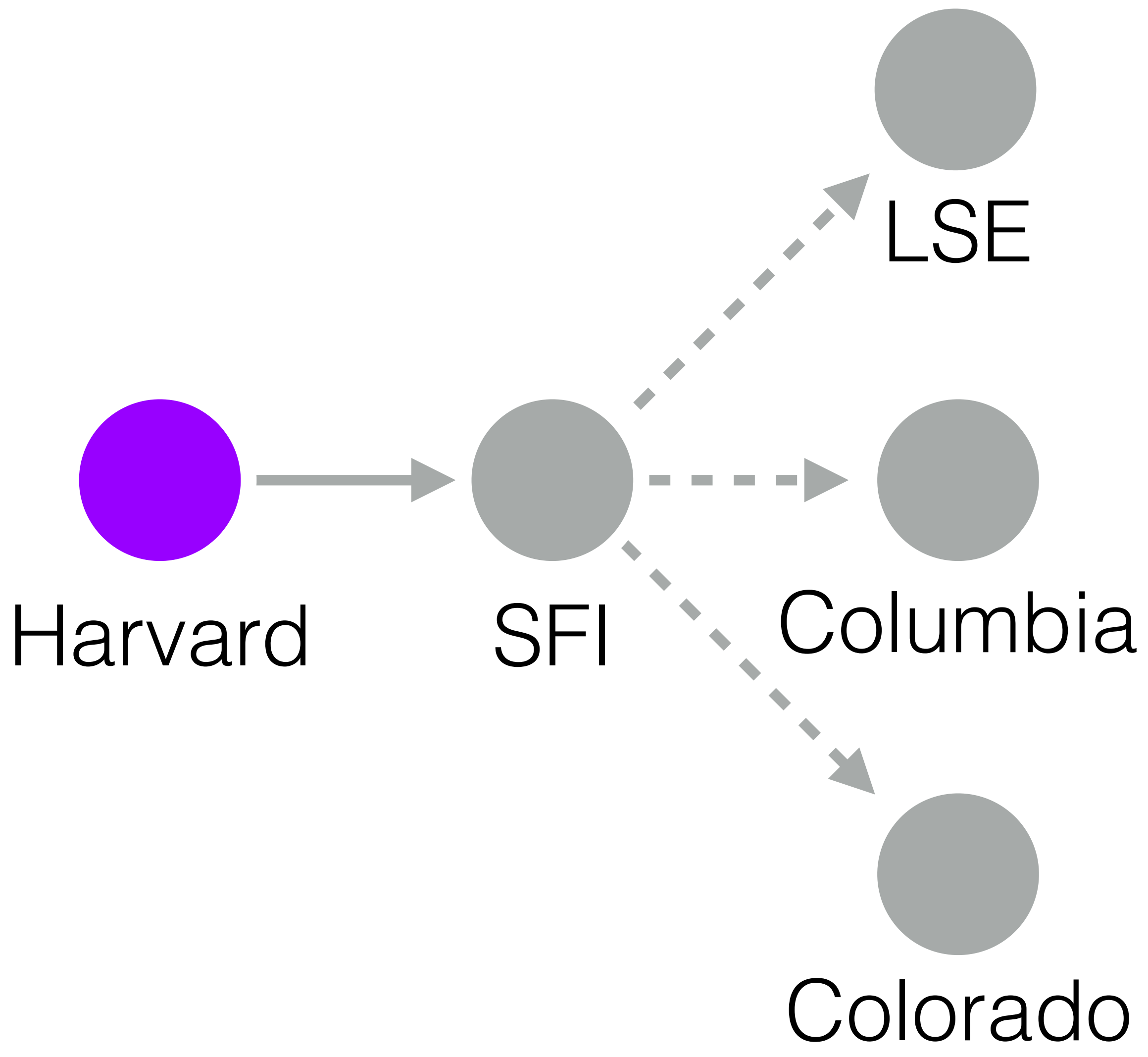
New hires as vectors for infectious ideas?



Do new hires *actually* bring ideas with them?
[or would popular topics get there anyway?]

Are some universities better idea exporters?

New hires as vectors for infectious ideas?



Do new hires *actually* bring ideas with them?
[or would popular topics get there anyway?]

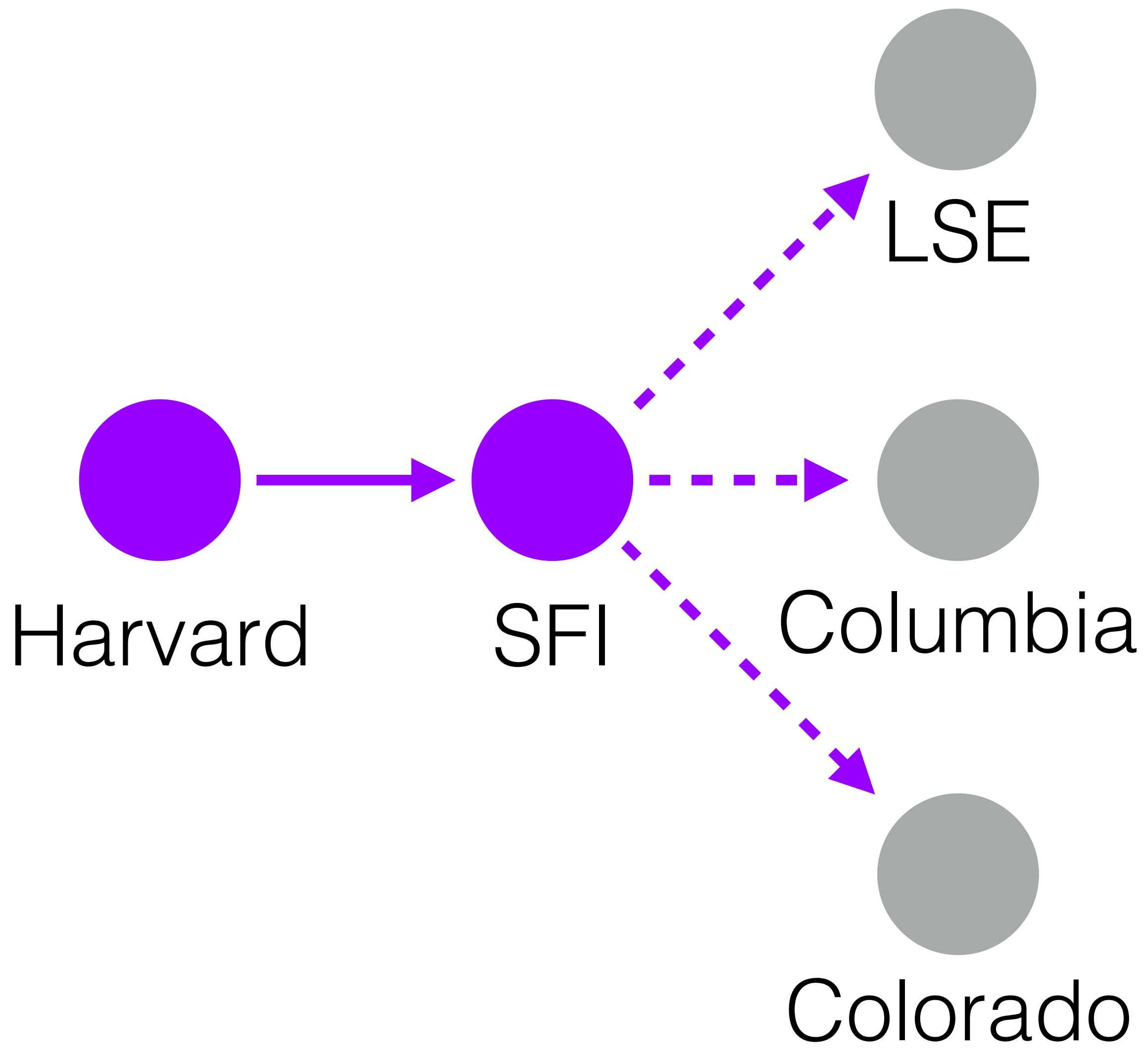
Are some universities better idea exporters?

Epidemic model: treat the idea as an infection, and a new hire as "infectious."

The probability that a hire transmits the idea to and uninfected university: p [idea quality]

Morgan, Economou, Way, Clauset *In preparation* (2018)

New hires as vectors for infectious ideas?



Do new hires *actually* bring ideas with them?
[or would popular topics get there anyway?]

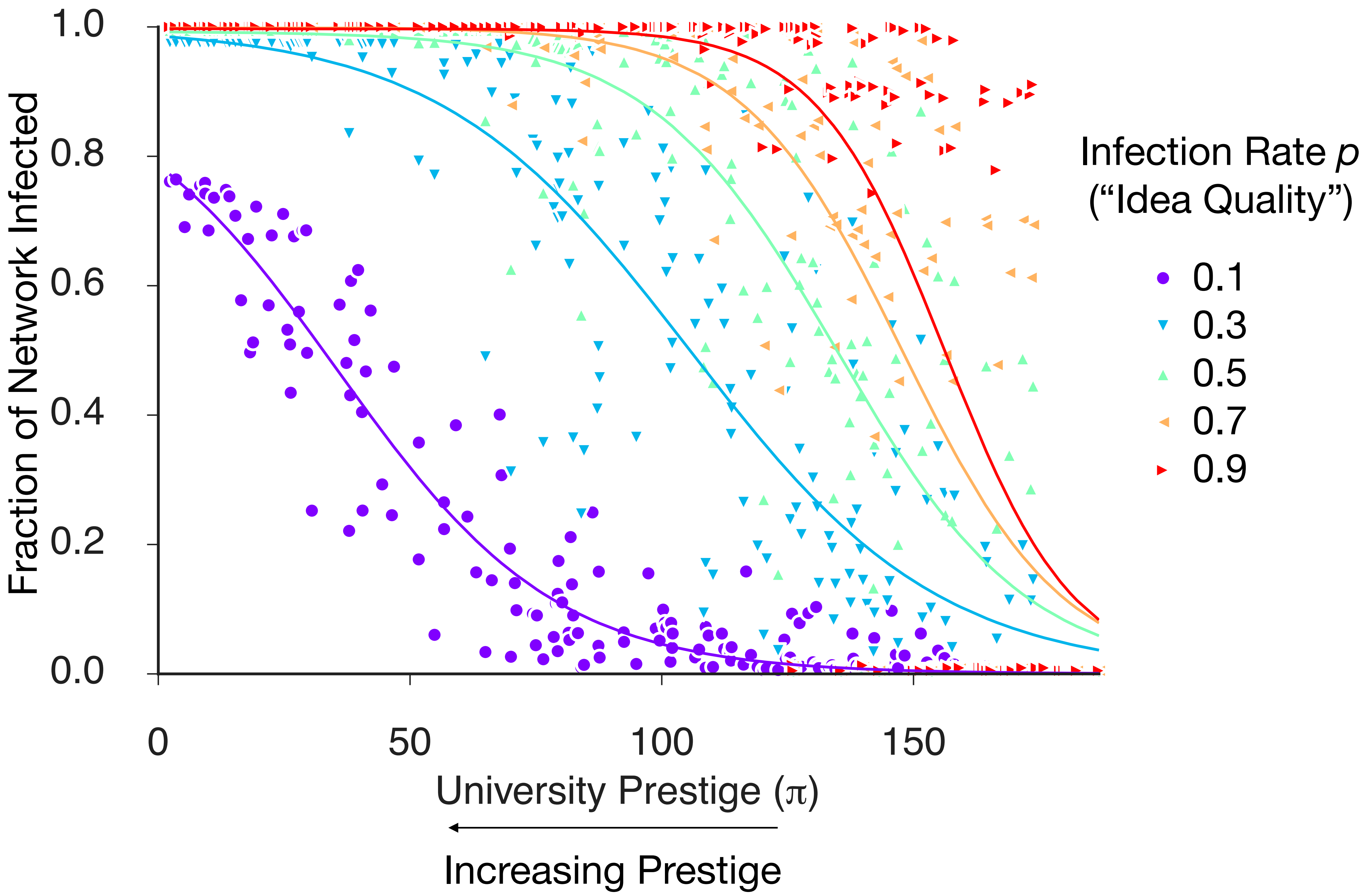
Are some universities better idea exporters?

Epidemic model: treat the idea as an infection, and a new hire as "infectious."

The probability that a hire transmits the idea to an uninfected university: p [idea quality]

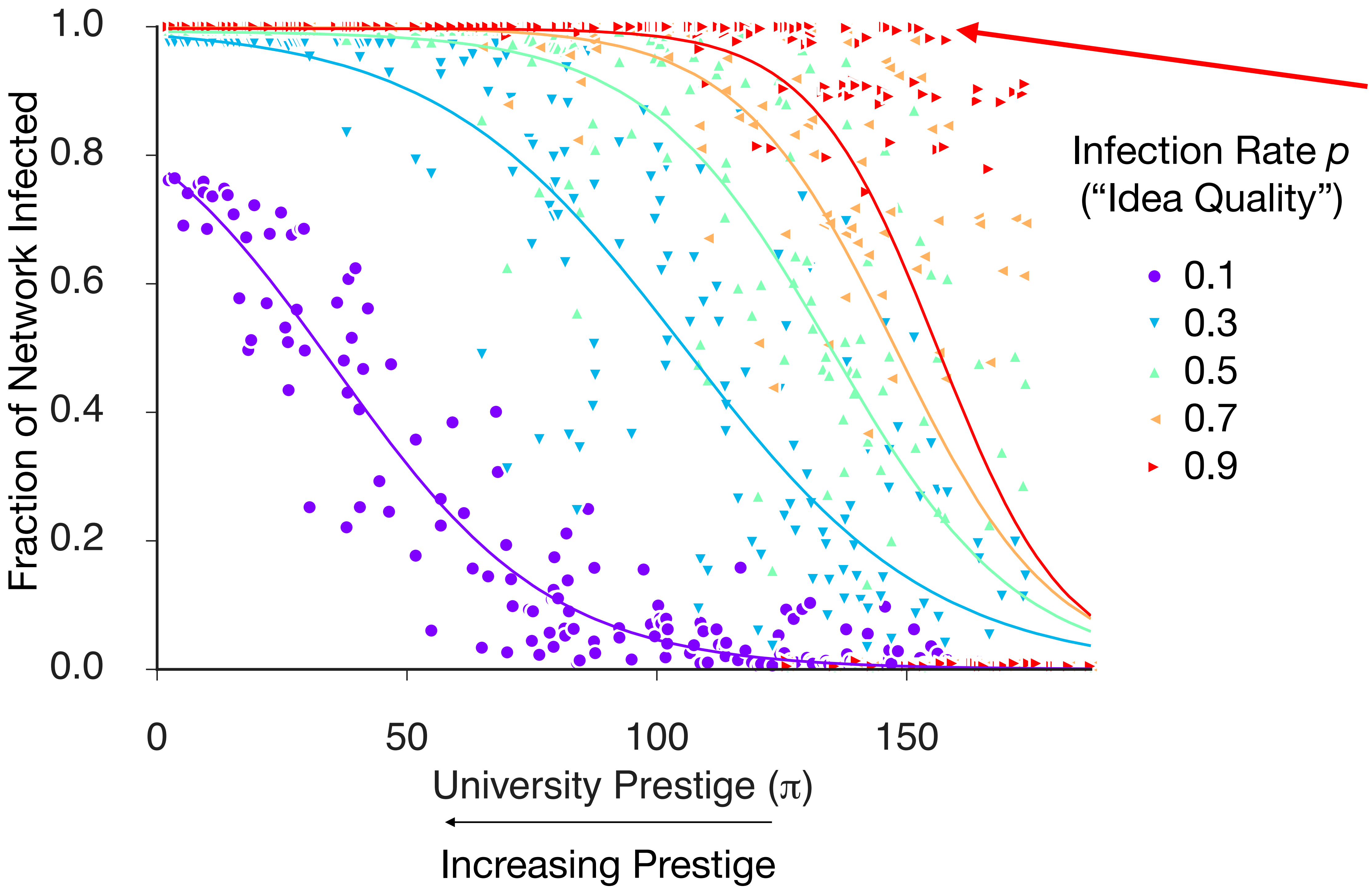
Morgan, Economou, Way, Clauset *In preparation* (2018)

Network position & the spread of ideas



Morgan, Economou, Way, Clauset *In preparation* (2018)

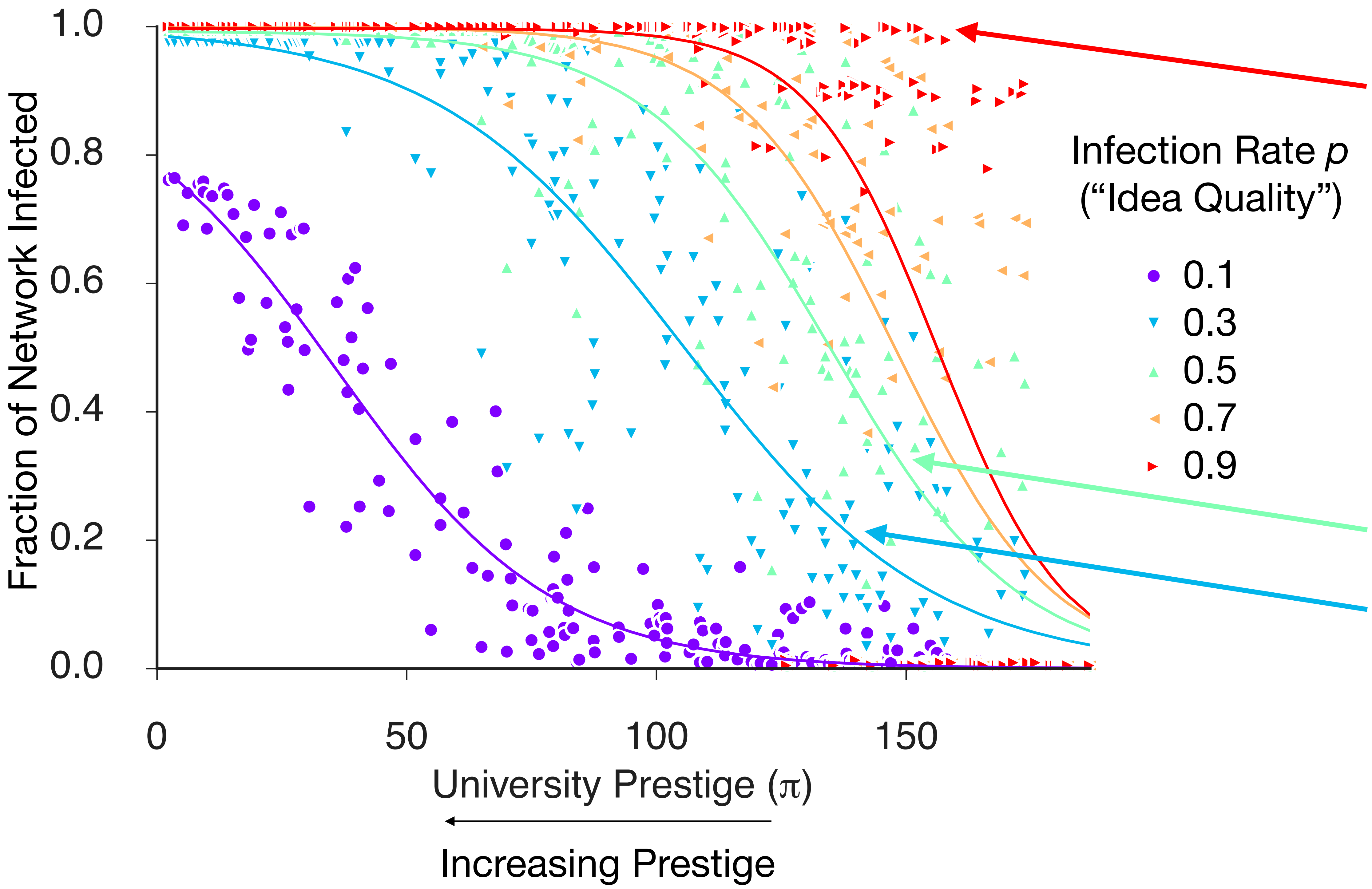
Network position & the spread of ideas



great ideas take root, regardless of starting place

Morgan, Economou, Way, Clauset *In preparation* (2018)

Network position & the spread of ideas



great ideas take root, regardless of starting place

good ideas spread easily from high-prestige univs, but poorly otherwise.

Morgan, Economou, Way, Clauset *In preparation* (2018)

Do *real* ideas spread along hiring links?

Analyzed over 200,000 computer science publications and over 2,500 hires.

Flagged publications on topic modeling, incremental computing, deep learning.

Morgan, Economou, Way, Clauset *In preparation* (2018)

Do *real* ideas spread along hiring links?

Analyzed over 200,000 computer science publications and over 2,500 hires.

Flagged publications on topic modeling, incremental computing, deep learning.

Identified faculty who brought [topic] with them when they were hired.
Identified faculty who began working on [topic] only 2+ yrs after being hired.

Morgan, Economou, Way, Clauset *In preparation* (2018)

Do *real* ideas spread along hiring links?

Analyzed over 200,000 computer science publications and over 2,500 hires.

Flagged publications on topic modeling, incremental computing, deep learning.

Identified faculty who brought [topic] with them when they were hired.
Identified faculty who began working on [topic] only 2+ yrs after being hired.

Compared relative rates of hiring-link spread vs spontaneous spread. (vs random)

Morgan, Economou, Way, Clauset *In preparation* (2018)

Do *real* ideas spread along hiring links?

Analyzed over 200,000 computer science publications and over 2,500 hires.
Flagged publications on **topic modeling, incremental computing, deep learning**.

Identified faculty who brought [topic] with them when they were hired.
Identified faculty who began working on [topic] only 2+ yrs after being hired.

Compared relative rates of **hiring-link spread** vs **spontaneous spread**. (vs random)

Spread of topic modeling (p=0.01) & incremental computing (p=0.01) significantly tied to infection via hiring. Spread of deep learning (p=0.2) *not* significantly linked to hiring.

Morgan, Economou, Way, Clauset *In preparation* (2018)

Colorado
Sam Way
Aaron Clauset
Allison Morgan
Dimitrios Economou

Kauffman / Lux
Sam Arbesman

luxcapital

Ewing Marion KAUFFMAN Foundation

SANTA FE INSTITUTE

NSF

2018

Allie Morgan

2015

Sam Way

2003

Aaron Clauset

Prestige drives epistemic inequality in the diffusion of scientific ideas
Morgan, Economou, Way, Clauset. *Submitted* (2018).

The misleading narrative of the canonical faculty productivity trajectory
Way, Morgan, Clauset, Larremore. *PNAS* (2017).

Data-driven predictions in the science of science
Clauset, Larremore, Sinatra. *Science* (2017).

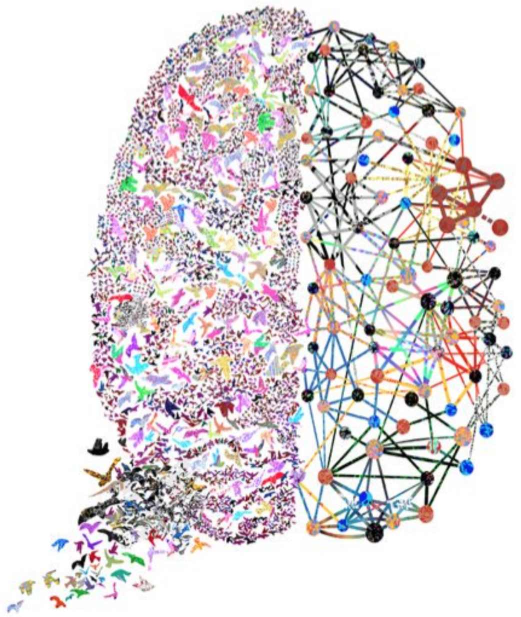
Gender, productivity, and prestige in computer science faculty hiring networks
Way, Larremore, Clauset. *Proc. WWW* (2016).

Systematic inequality and hierarchy in faculty hiring networks
Clauset, Arbesman, Larremore. *Science Advances*. (2015).

Conference on Complex Networks

COMPLENET '18

Hosted by Northeastern University Network Science Institute



BOSTON, MA APRIL 2018

Xindi Wang

Northeastern University

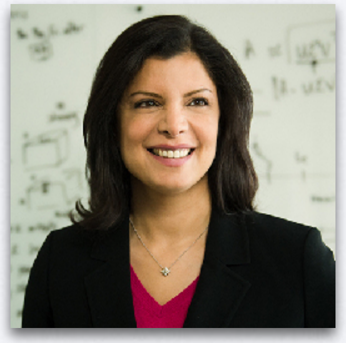
Network Science Institute

LEARNING TO PLACE OBJECTS: A NETWORK-BASED APPROACH

Xindi Wang

Onur Varol

Tina Eliassi-Rad

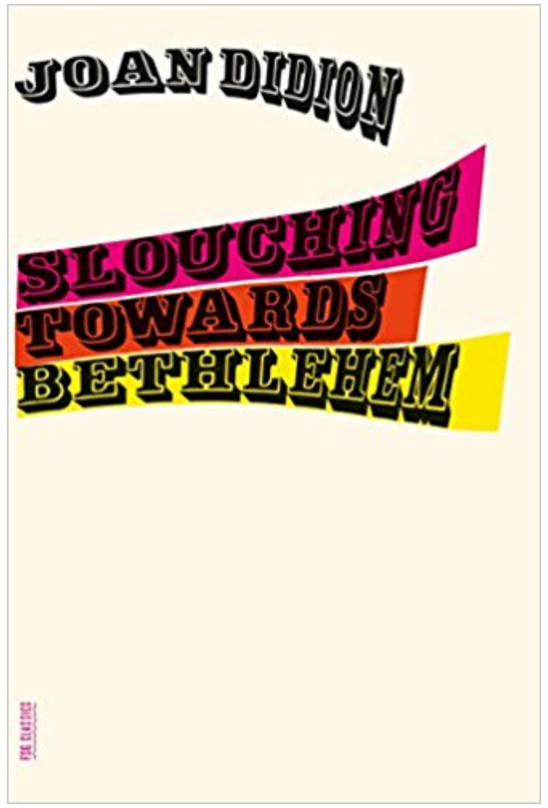


Albert-László Barabási

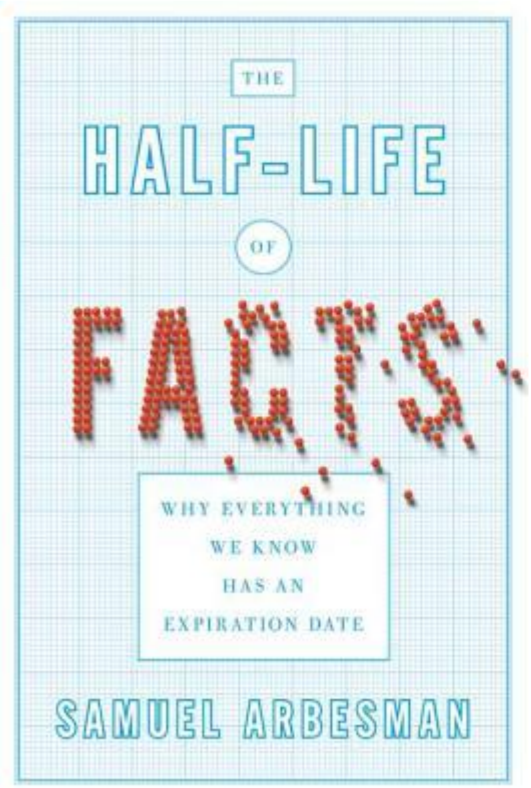
Suppose I give you a book. Predict its sales.

Existing data: books and their sales.

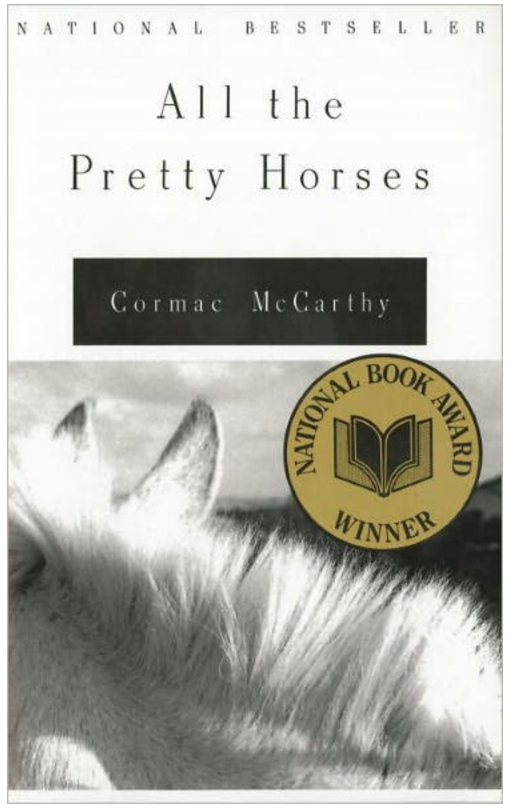
1. turn books into feature vectors.



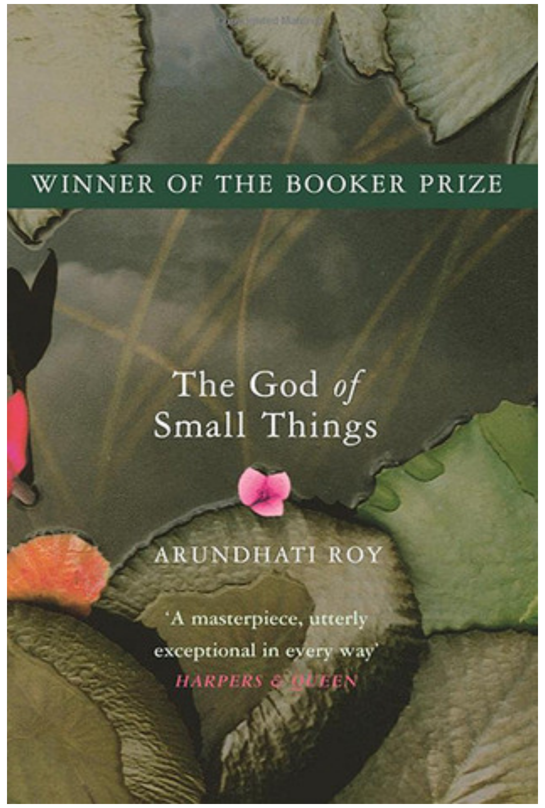
$\overrightarrow{x}_1$



$\overrightarrow{x}_2$



$\overrightarrow{x}_3$



$\overrightarrow{x}_4$

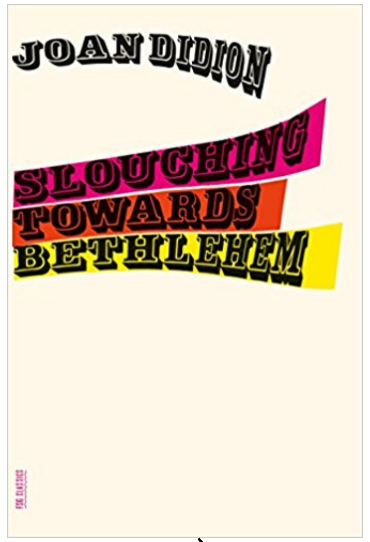
2. Train a model:

$$P(\text{book } i > \text{book } j \mid \overrightarrow{x}_i, \overrightarrow{x}_j, \theta)$$

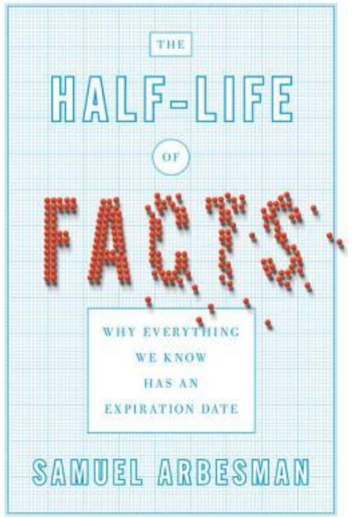
Suppose I give you a book. Predict its sales.

Existing data: books and their sales.

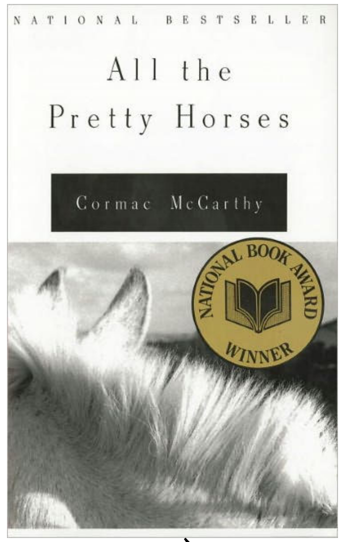
1. turn books into feature vectors.



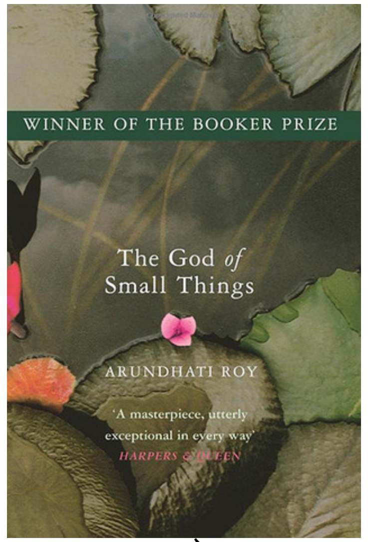
$\vec{x}_1$



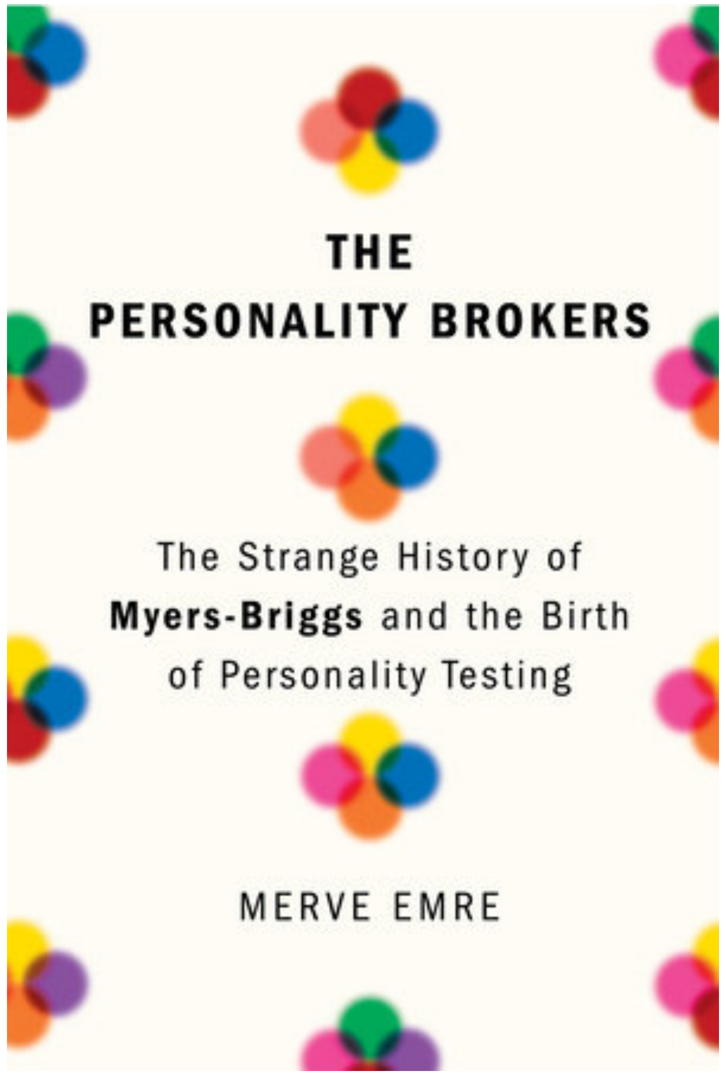
$\vec{x}_2$



$\vec{x}_3$



$\vec{x}_4$



$\vec{x}_5$

2. Train a model.

3. Use the model to simulate pairwise competitions.

   $P(\text{book } i > \text{book } 5 \mid \vec{x}_i, \vec{x}_5, \theta)$

4. Use [**your favo(u)rite algorithm**] to infer rank_5 from pairwise comparisons.

Rankings rankings

Area under the receiver-operator curve (AUC)

Method		AUC on Fiction	AUC on Biography
	KNN	0.759	0.815
Pairwise +	Cohen et al.	0.892	0.871
	WTG wave	**0.910**	**0.892**
	Voting	**0.915**	**0.891**
	FAS-PIVOT	**0.907**	**0.892**

Want to learn more? xindi.w1993@gmail.com

Rankings rankings

Area under the receiver-operator curve (AUC)

Method		AUC on Fiction	AUC on Biography
	KNN	0.759	0.815
Pairwise +	Cohen et al.	0.892	0.871
	WTG wave	**0.910**	**0.892**
	Voting	**0.915**	**0.891**
	FAS-PIVOT	**0.907**	**0.892**
	SpringRank	**0.908**	**0.893**

Want to learn more? xindi.w1993@gmail.com

Rankings rankings

Area under the receiver-operator curve (AUC)

Method		AUC on Fiction	AUC on Biography
	KNN	0.759	0.815
Pairwise +	Cohen et al.	0.892	0.871
	WTG wave	**0.910**	**0.892**
	Voting	**0.915**	**0.891**
	FAS-PIVOT	**0.907**	**0.892**
	SpringRank	**0.908**	**0.893**

Want to learn more? xindi.w1993@gmail.com

Rankings rankings

Area under the receiver-operator curve (AUC)

Method		AUC on Fiction	AUC on Biography
	KNN	0.759	0.815
Pairwise +	Cohen et al.	0.892	0.871
	WTG wave	**0.910**	**0.892**
	Voting	**0.915**	**0.891**
	FAS-PIVOT	**0.907**	**0.892**
	SpringRank	**0.908**	**0.893**

Want to learn more? xindi.w1993@gmail.com

Rankings rankings

Area under the receiver-operator curve (AUC)

Method		AUC on Fiction	AUC on Biography
	KNN	0.759	0.815
Pairwise +	Cohen et al.	0.892	0.871
	WTG wave	**0.910**	**0.892**
	Voting	**0.915**	**0.891**
	FAS-PIVOT	**0.907**	**0.892**
	SpringRank	**0.908**	**0.893**

Want to learn more? xindi.w1993@gmail.com

But actually… many methods performed well!

Now the question is: why do the top four algorithms perform similarly?

What does that tell us about the **structure of the problem** and the **structure of the space** over which we are ranking?

But actually… many methods performed well!

Now the question is: why do the top four algorithms perform similarly?

What does that tell us about the **structure of the problem** and
the **structure of the space** over which we are ranking?

Use the consistency of the ranking results across
algorithms to learn about the system itself.

What does it mean for a space or problem or set to be easily ordered or rankable?

But actually… many methods performed well!

Now the question is: why do the top four algorithms perform similarly?

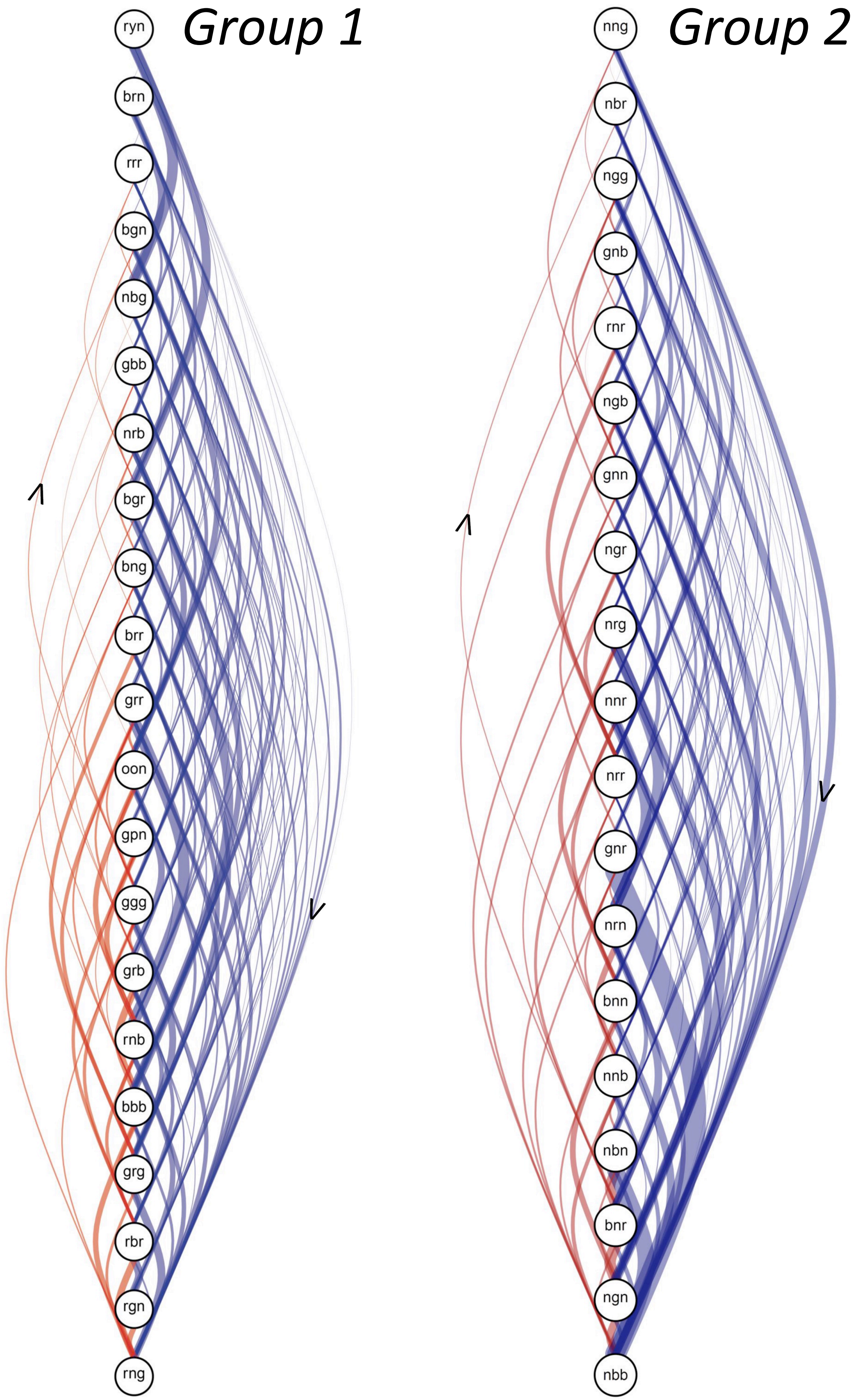
What does that tell us about the **structure of the problem** and
the **structure of the space** over which we are ranking?

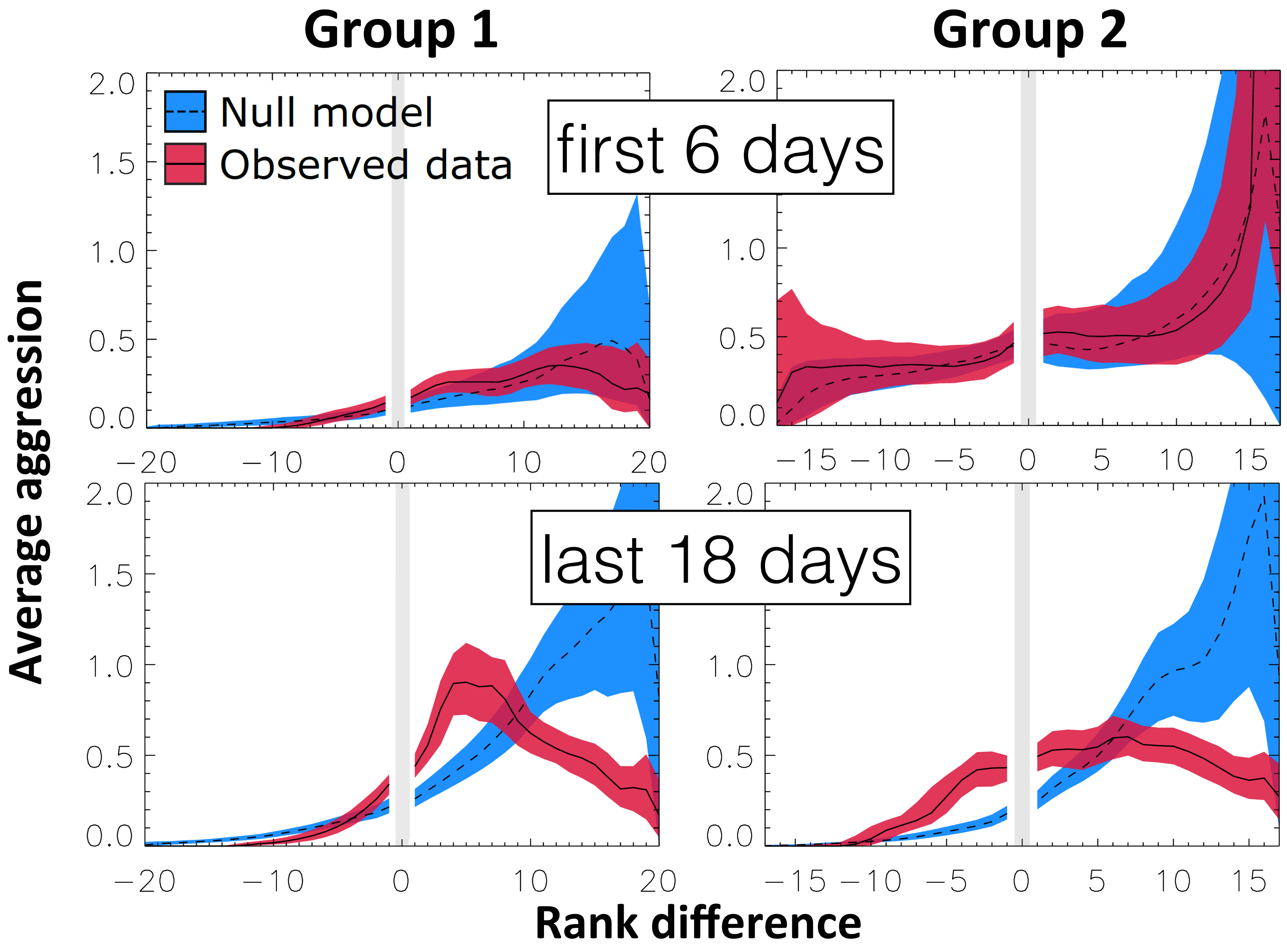
Use the consistency of the ranking results across
algorithms to learn about the system itself.

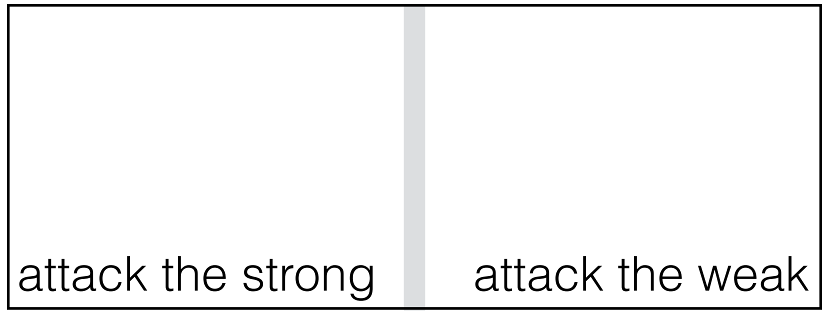
What does it mean for a space or problem or set to be easily ordered or rankable?

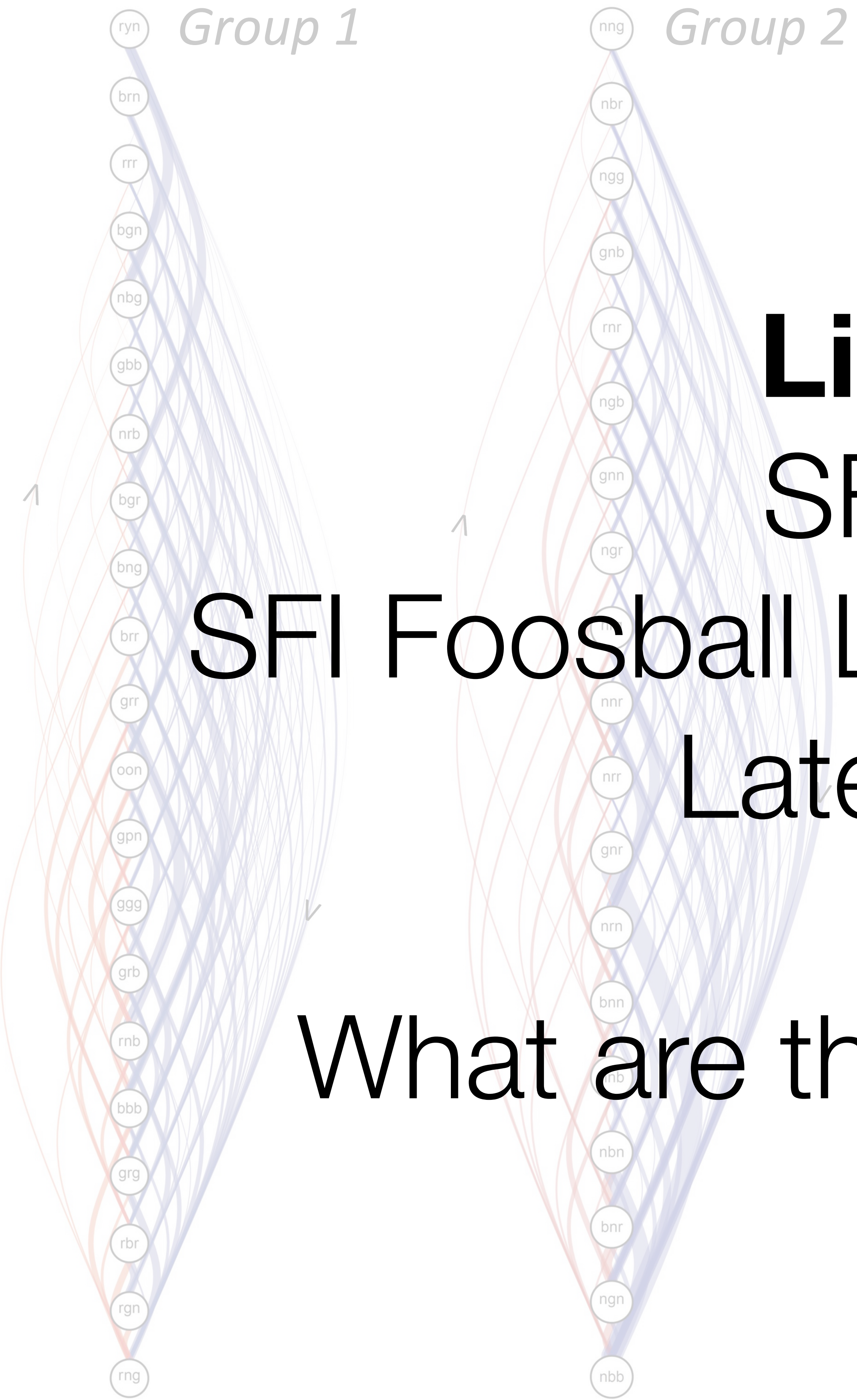
Want to learn more? xindi.w1993@gmail.com


Hierarchy and cognition


Group 1
Group 2
ryn
brn
rrr
bgn
nbg
gbb
nrb
bgr
bng
brr
grr
oon
gpn
ggg
grb
rnb
bbb
grg
rbr
rgn
rng
nng
nbr
ngg
gnb
rnr
ngb
gnn
ngr
nrg
nnr
nrr
gnr
nrn
bnn
nnb
nbn
bnr
ngn
nbb


Group 1
Group 2
Null model
Observed data
first 6 days
last 18 days
Average aggression
Rank difference
2.0
1.5
1.0
0.5
0.0
−20
−10
0
10
20
−15
−10
−5
5
15


attack the strong
attack the weak



Liz Hobson

SFI Postdoc

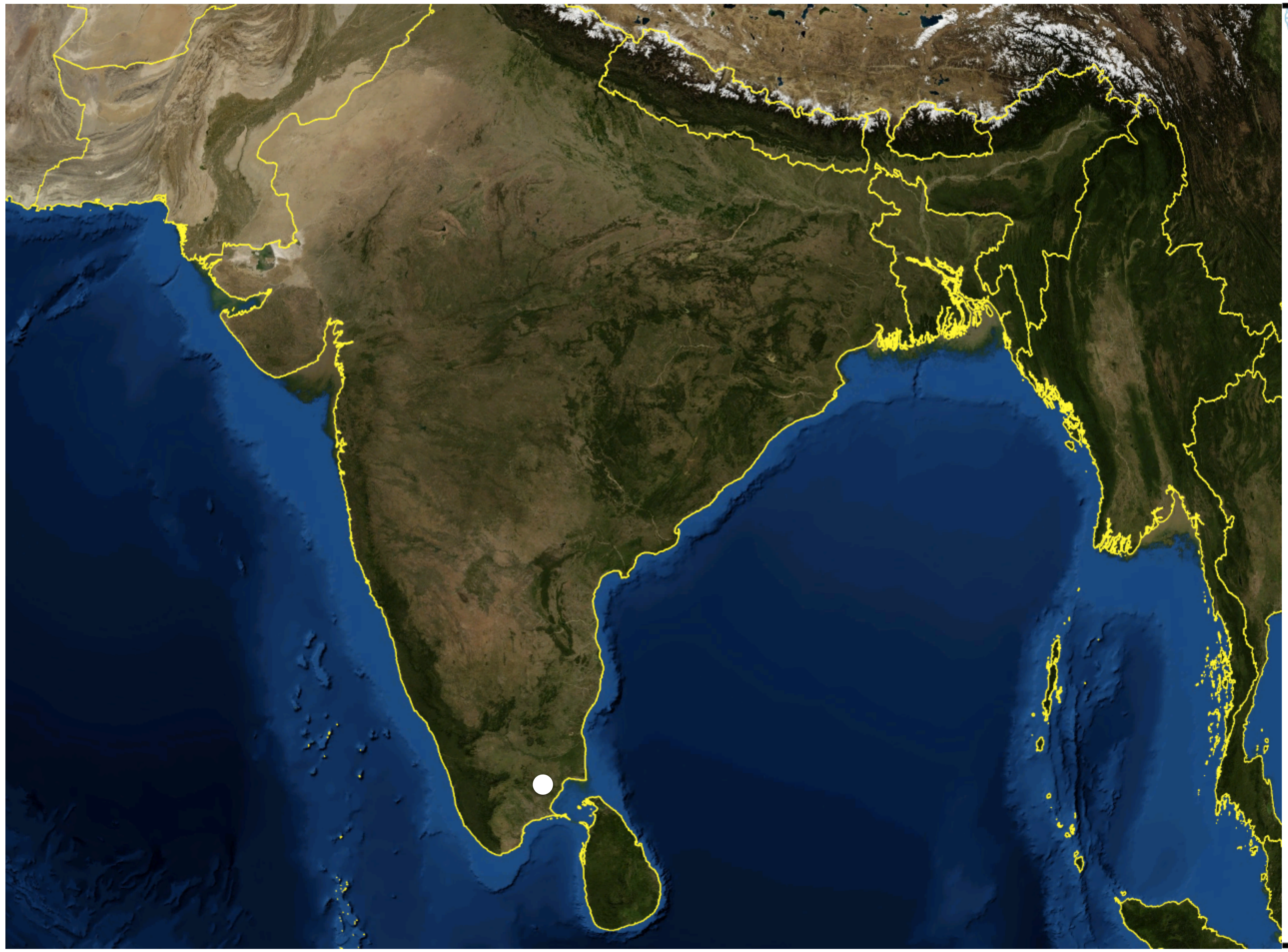
SFI Foosball League Commissioner

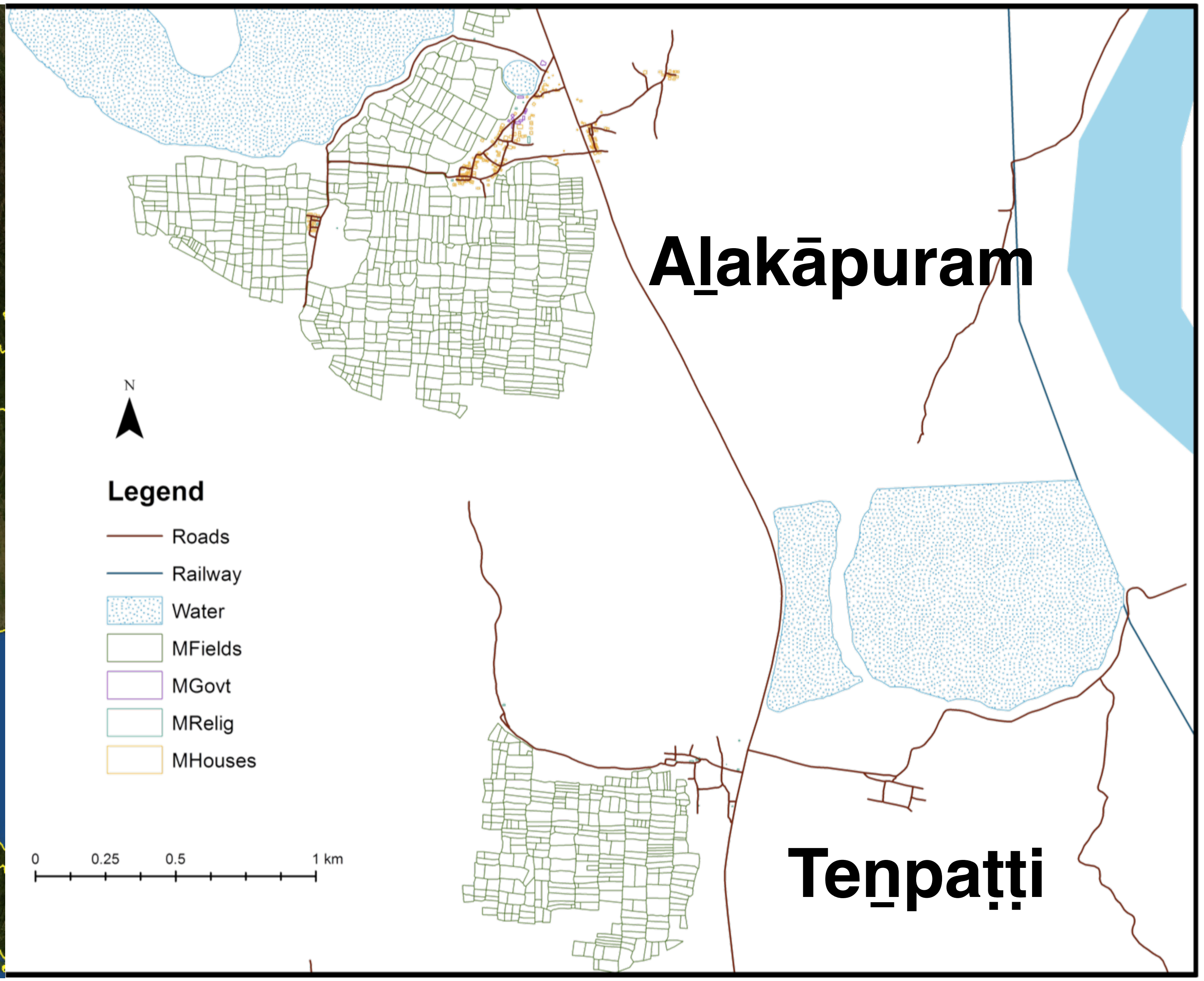
Later this week!

What are these birds thinking?!


Groups and ordered structures

South Indian networks: Teṉpaṭṭi and Aḻakāpuram





1964 question of Srinavas and Béteille: beyond ethnographic investigations?

Ranked order quality, R

We propose to measure the **quality** of a ranked ordering by R

$$R = \frac{1}{m} \sum_{ij} (A_{ij} - E_{ij})\, \mathbf{1}_{g_i \leq g_j}$$

links

network adjacency matrix

expectation (null model)

indicator
group of i ≤ group of j

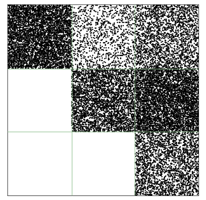
Larremore, Hébert-Dufresne, Power. *In prep.* (2018)

Ranked order quality, *R*

We propose to measure the **quality** of a ranked ordering by ***R***

$$R = \frac{1}{m} \sum_{ij} (A_{ij} - E_{ij}) \, \mathbf{1}_{g_i \leq g_j}$$

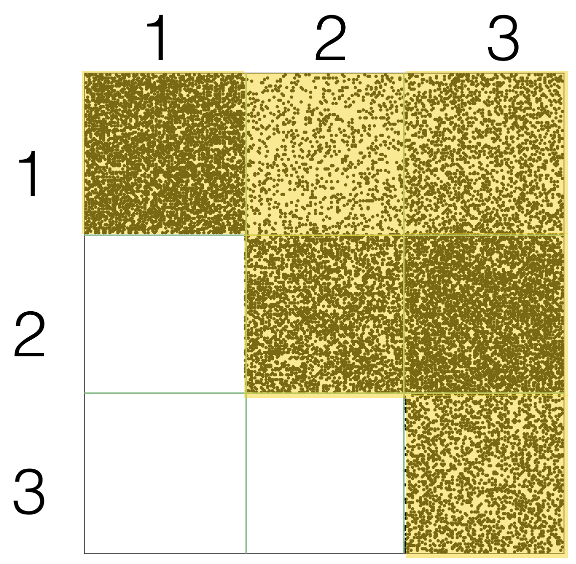
links



network adjacency matrix

expectation (null model)

indicator
group of i ≤ group of j



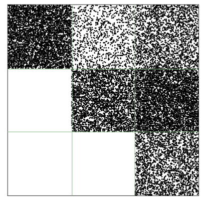
endorsement quality, R

Ranked order quality, *R*

We propose to measure the **quality** of a ranked ordering by R

$$R = \frac{1}{m} \sum_{ij} (A_{ij} - E_{ij})\, \mathbf{1}_{g_i \le g_j}$$

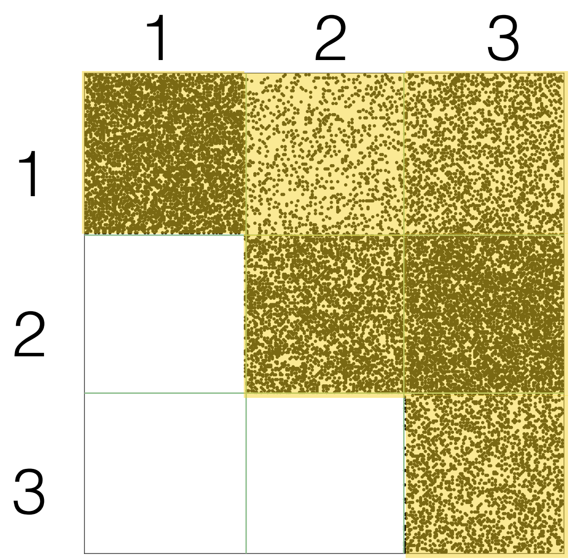
links



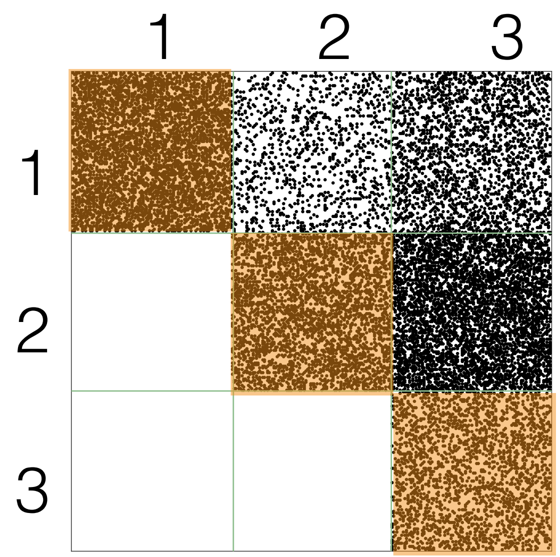
network adjacency matrix

expectation (null model)

indicator
group of i ≤ group of j



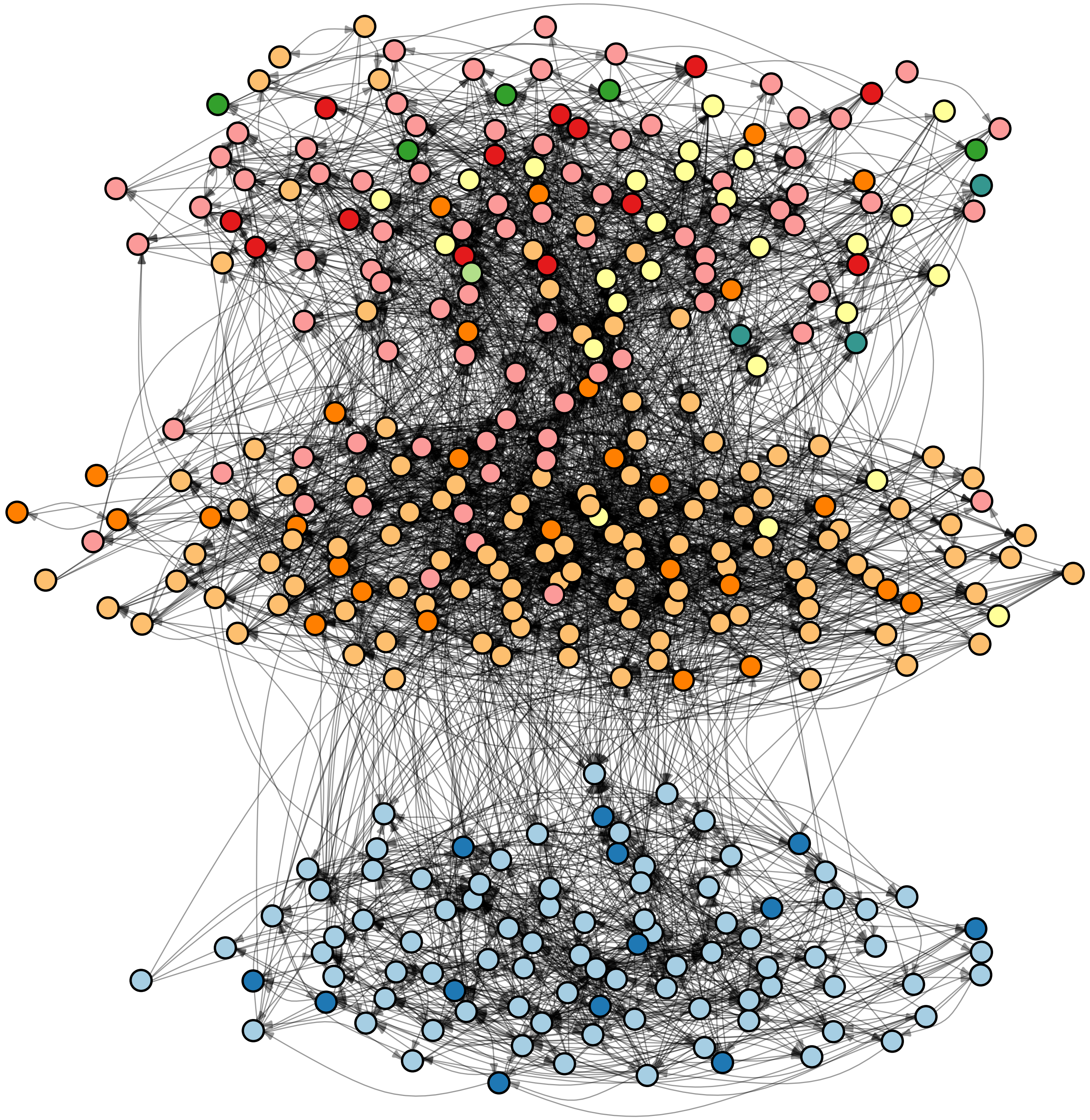
endorsement quality, R



modularity, Q

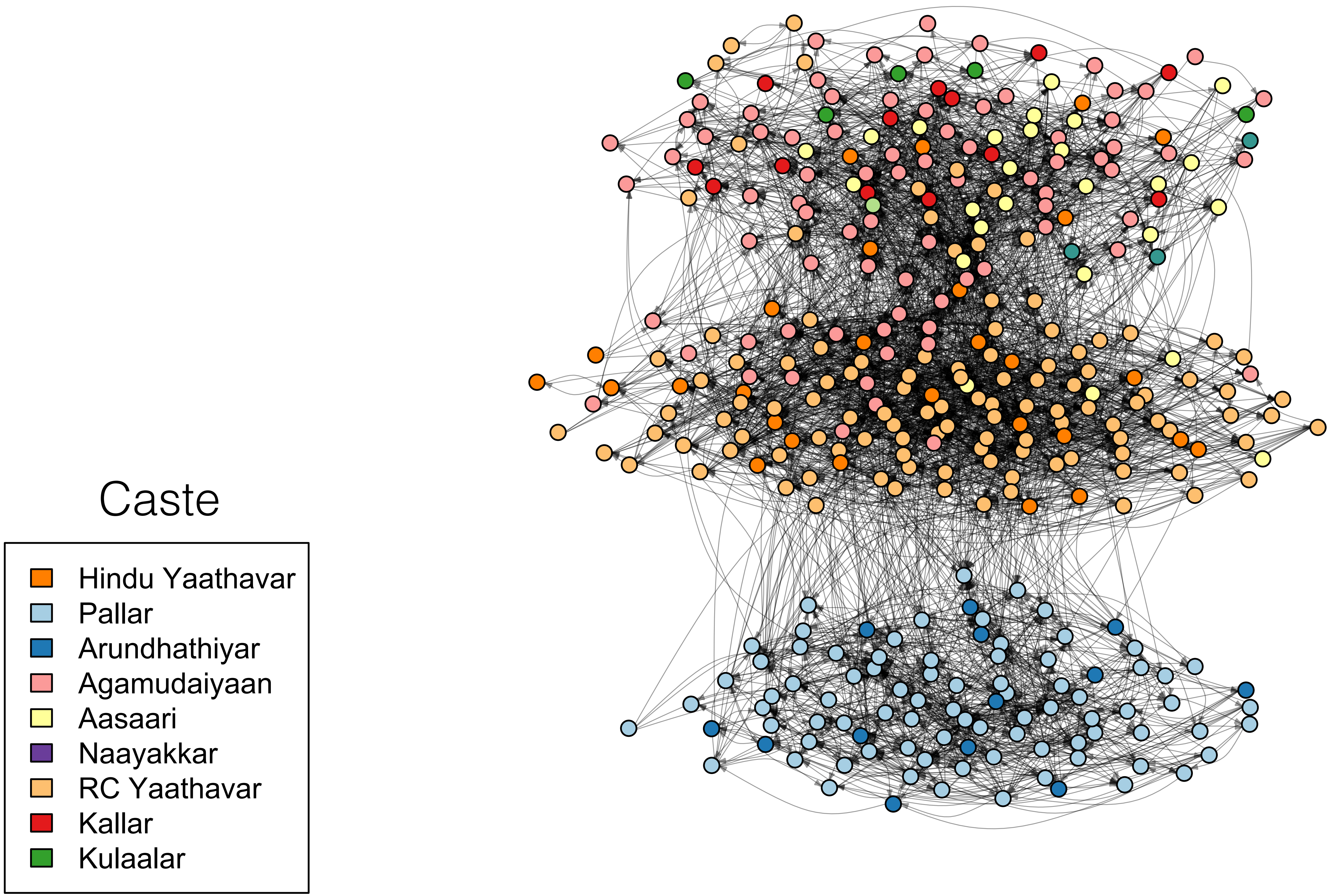
Larremore, Hébert-Dufresne, Power. *In prep.* (2018)

Teṉpaṭṭi



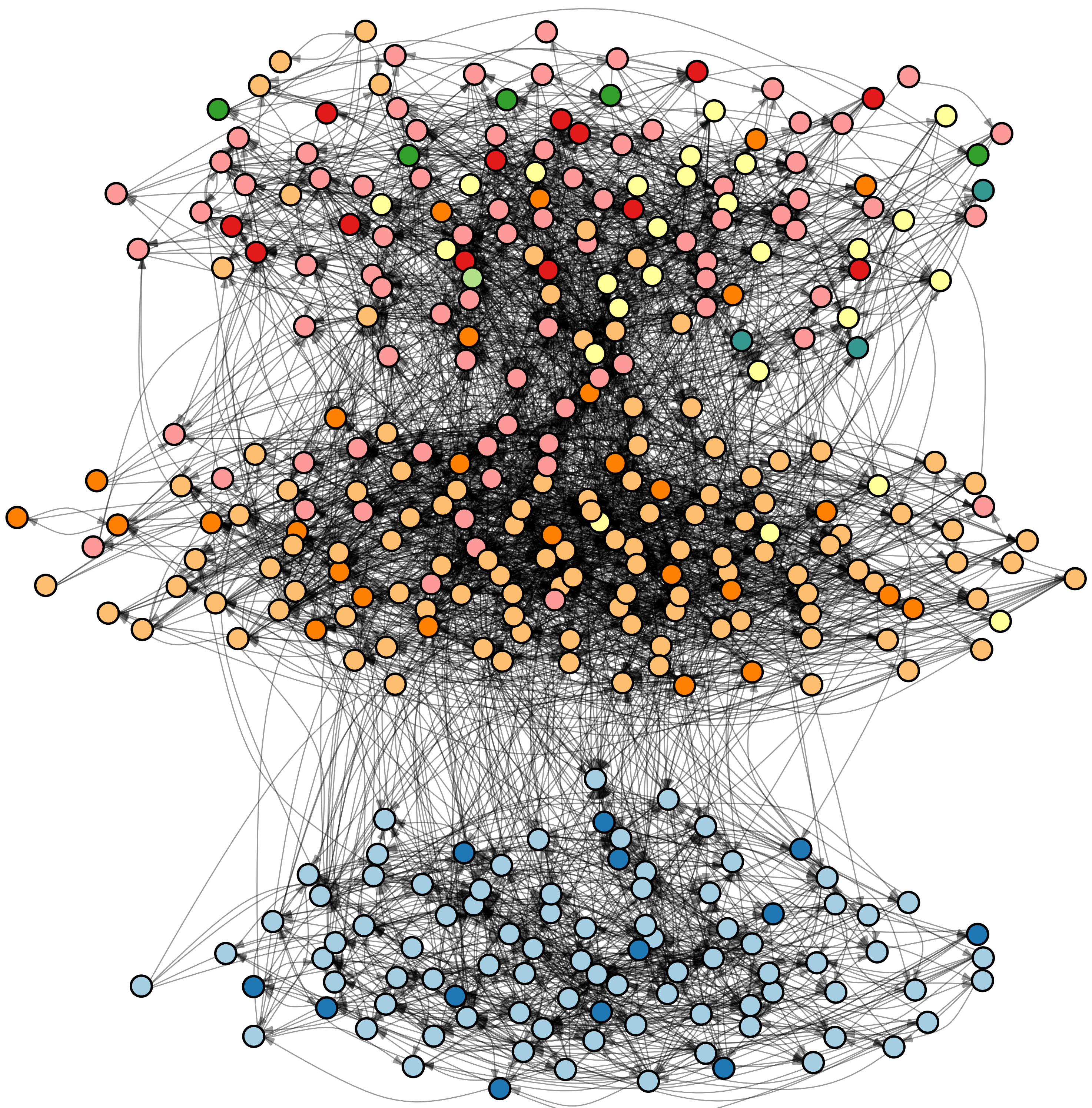
Larremore, Hébert-Dufresne, Power. *In prep.* (2018)

Teṉpaṭṭi



Larremore, Hébert-Dufresne, Power. *In prep.* (2018)

Teṉpaṭṭi

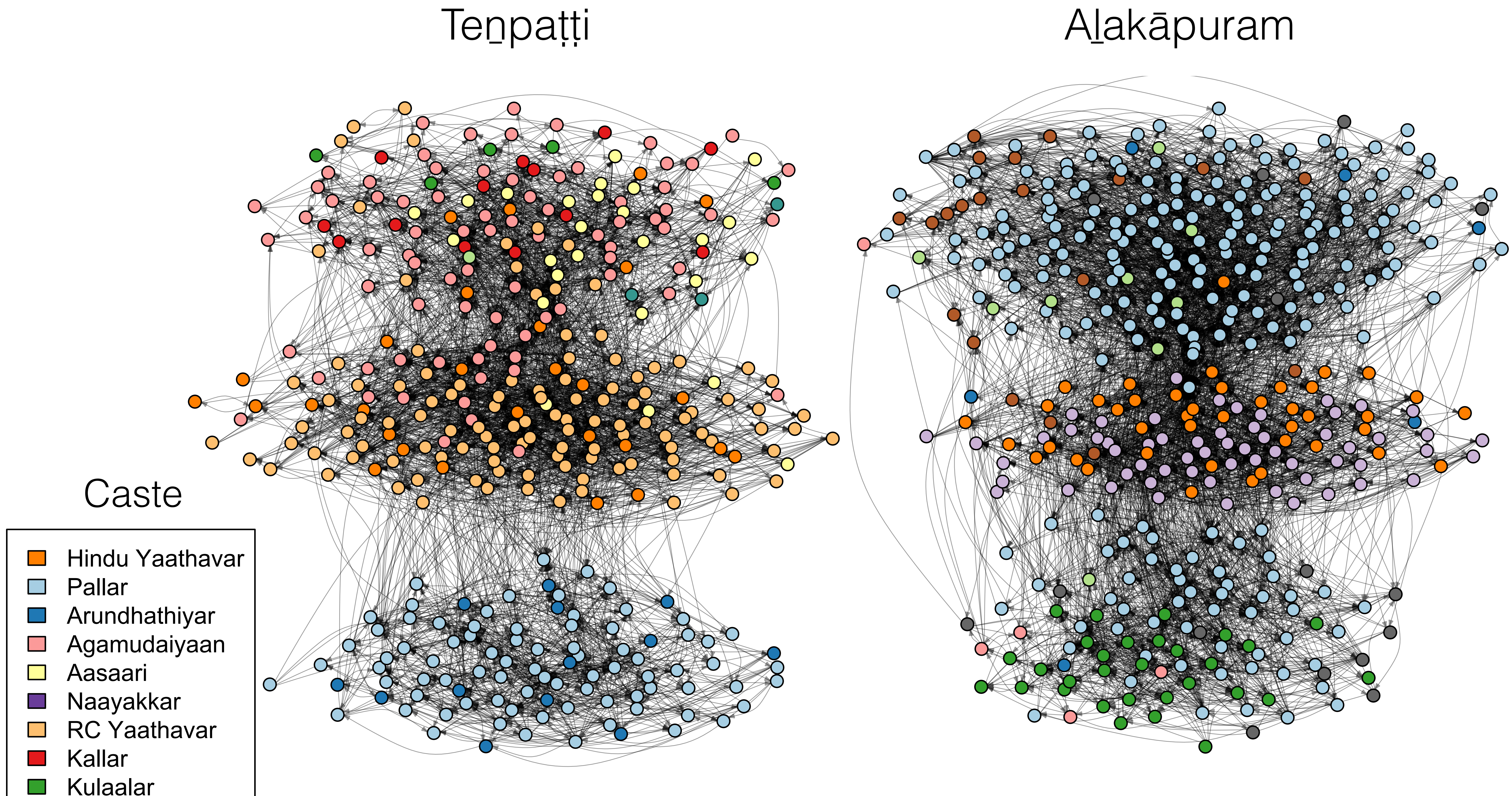


Caste

- Hindu Yaathavar
- Pallar
- Arundhathiyar
- Agamudaiyaan
- Aasaari
- Naayakkar
- RC Yaathavar
- Kallar
- Kulaalar

Scheduled castes—*dalit*
“untouchable”

Larremore, Hébert-Dufresne, Power. *In prep.* (2018)



Larremore, Hébert-Dufresne, Power. *In prep.* (201[illegible])

Many uses for models of large-scale structure

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in "space" or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

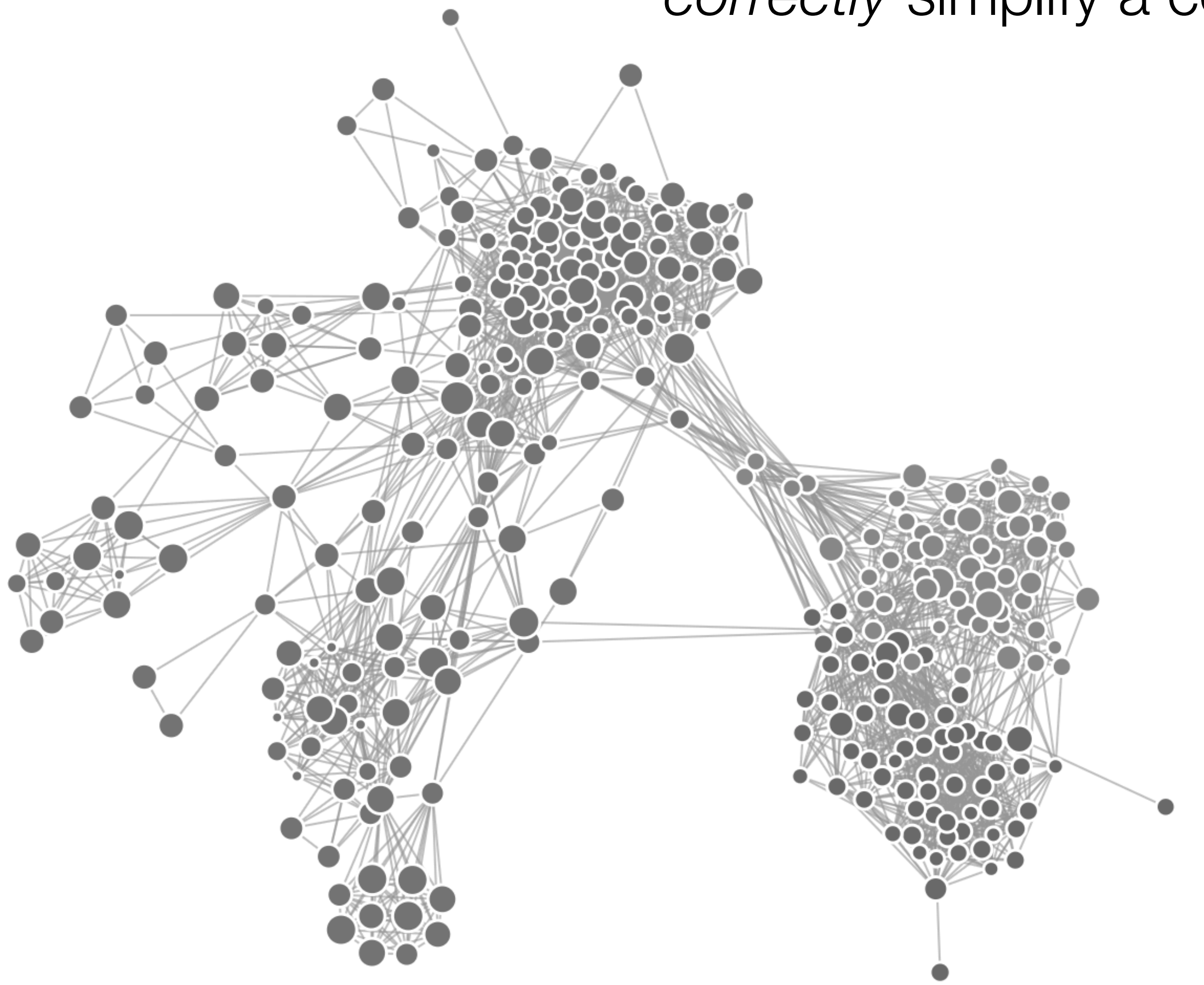
Treat the network like a means to an end; an intermediate data structure:

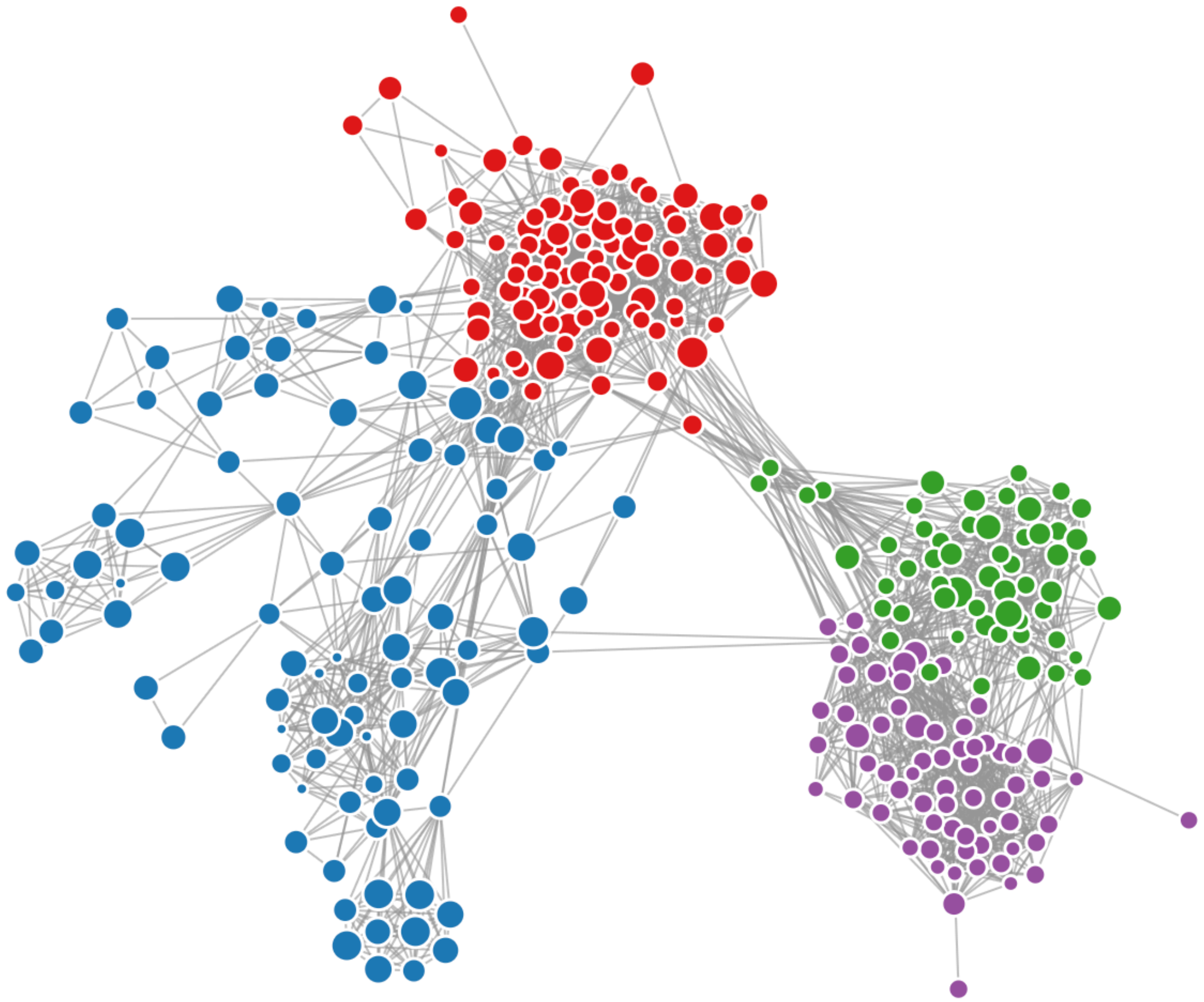
Useful division. Need groups so that we can assign treatments in an A/B test.
Simplification. Downstream regression model needs ranks or groups.

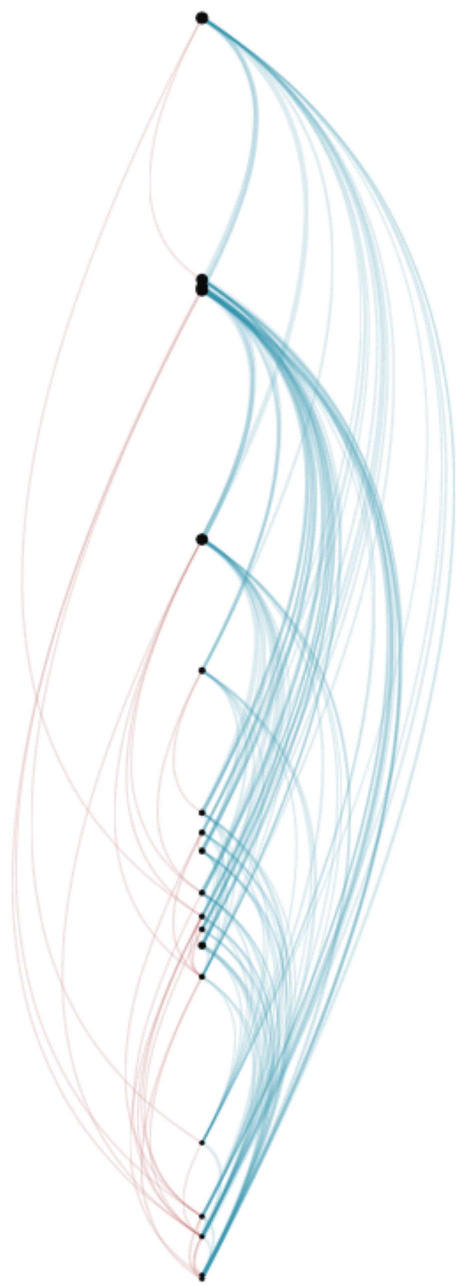
intuition: compare this list with the list you would write for regression

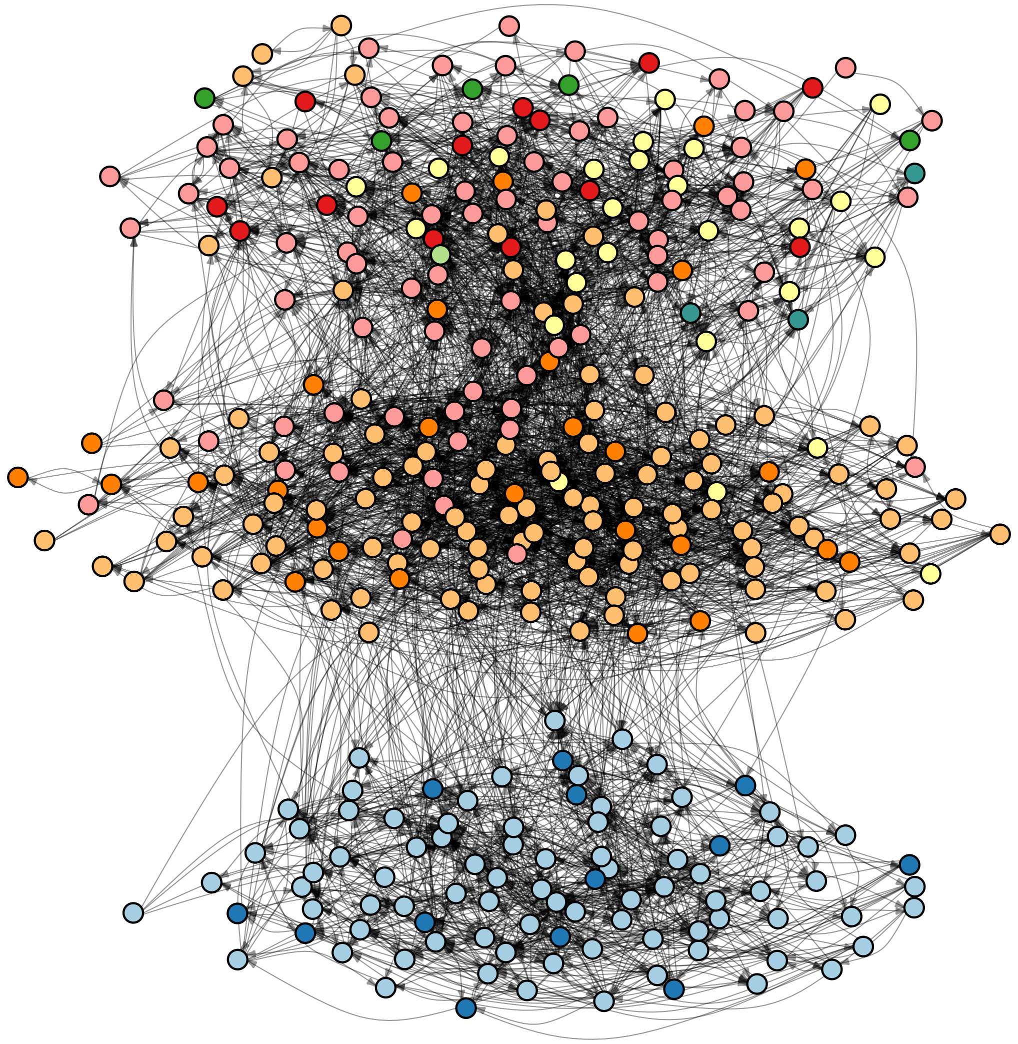
Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.
E. W. Dijkstra

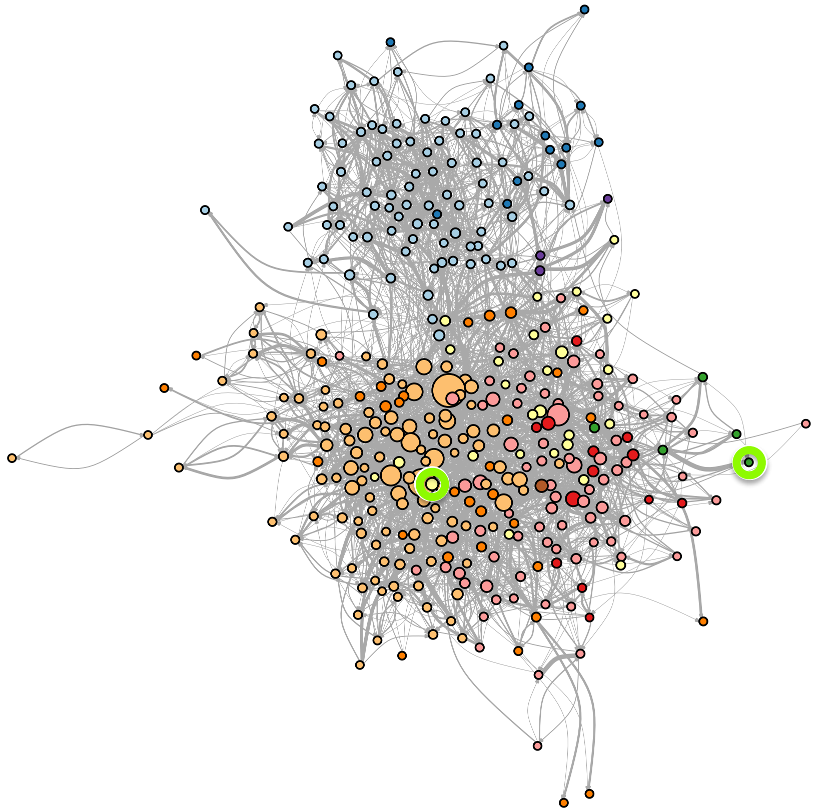
We can interpret this in two ways:
The Cynic: Pictures of networks can be *really cool* but
our goal is to do good science, not make pretty pictures.
The Scientist: The most beautiful science is when we
correctly simplify a complex system.

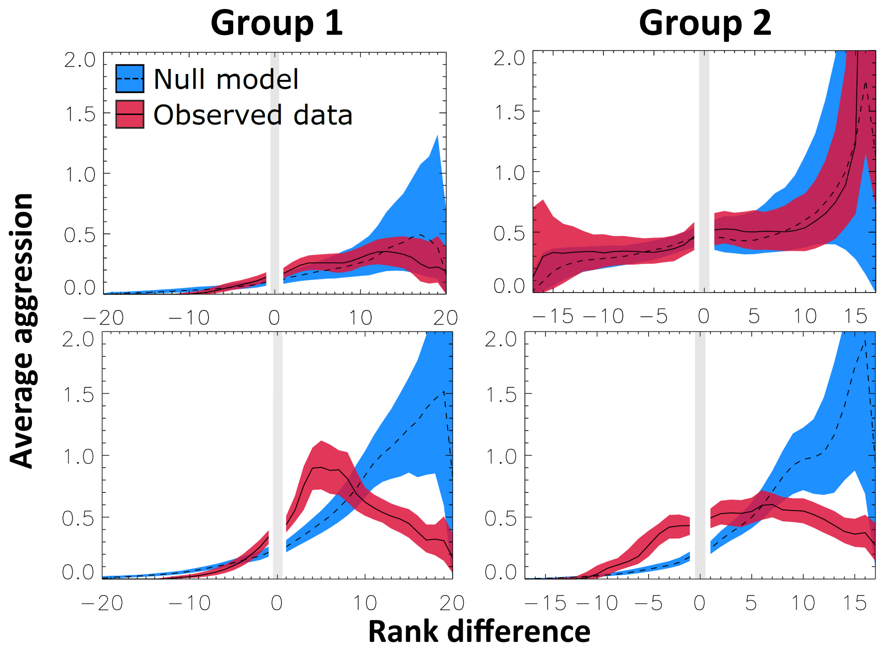


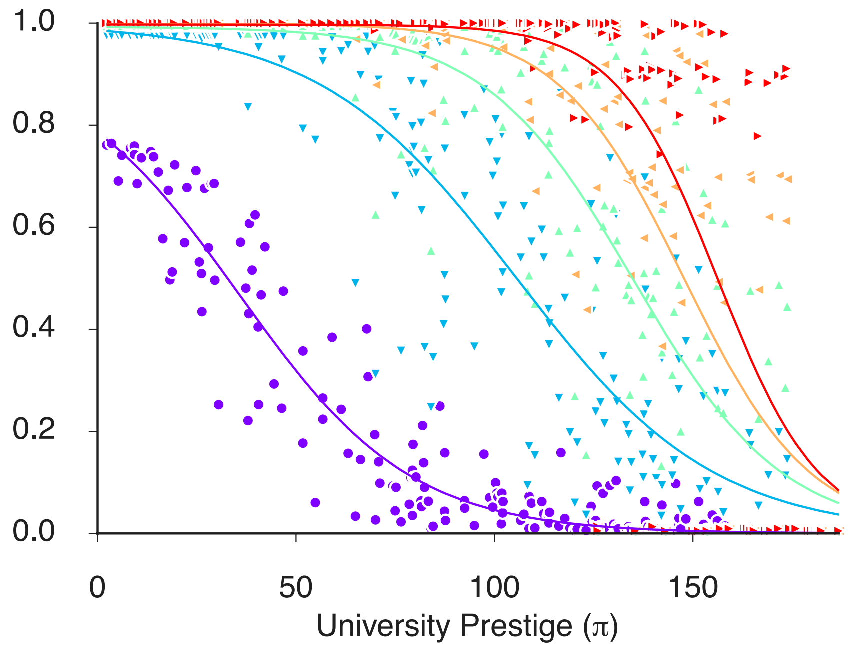


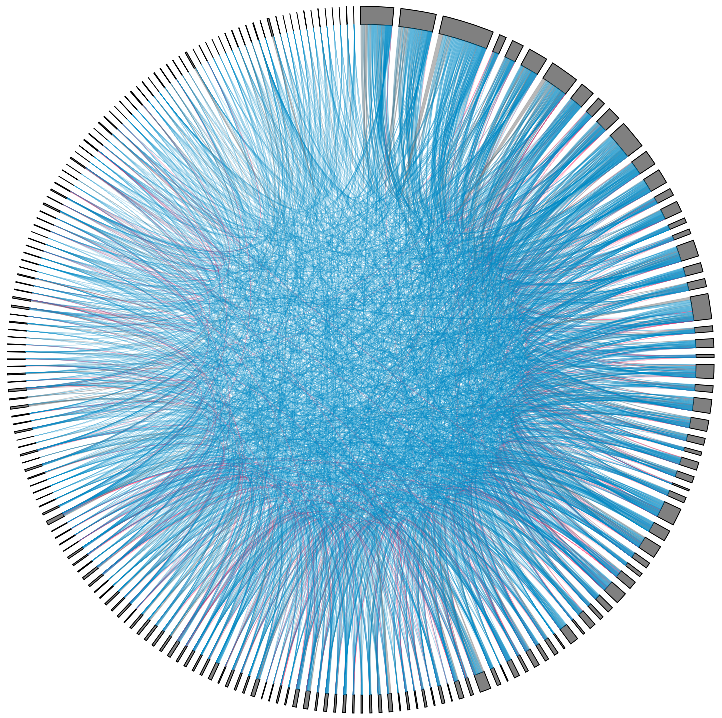












Prestige and status structures emerge in networks & we can identify them.

Beyond pictures: these things matter. traps, formation, ideas, & inequalities.

Thank you

@danlarremore
daniel.larremore@colorado.edu