Community detection on networks

Caterina De Bacco

CSSS, Santa Fe, 2018

caterina.debacco@gmail.com https://github.com/cdebacco

The plan

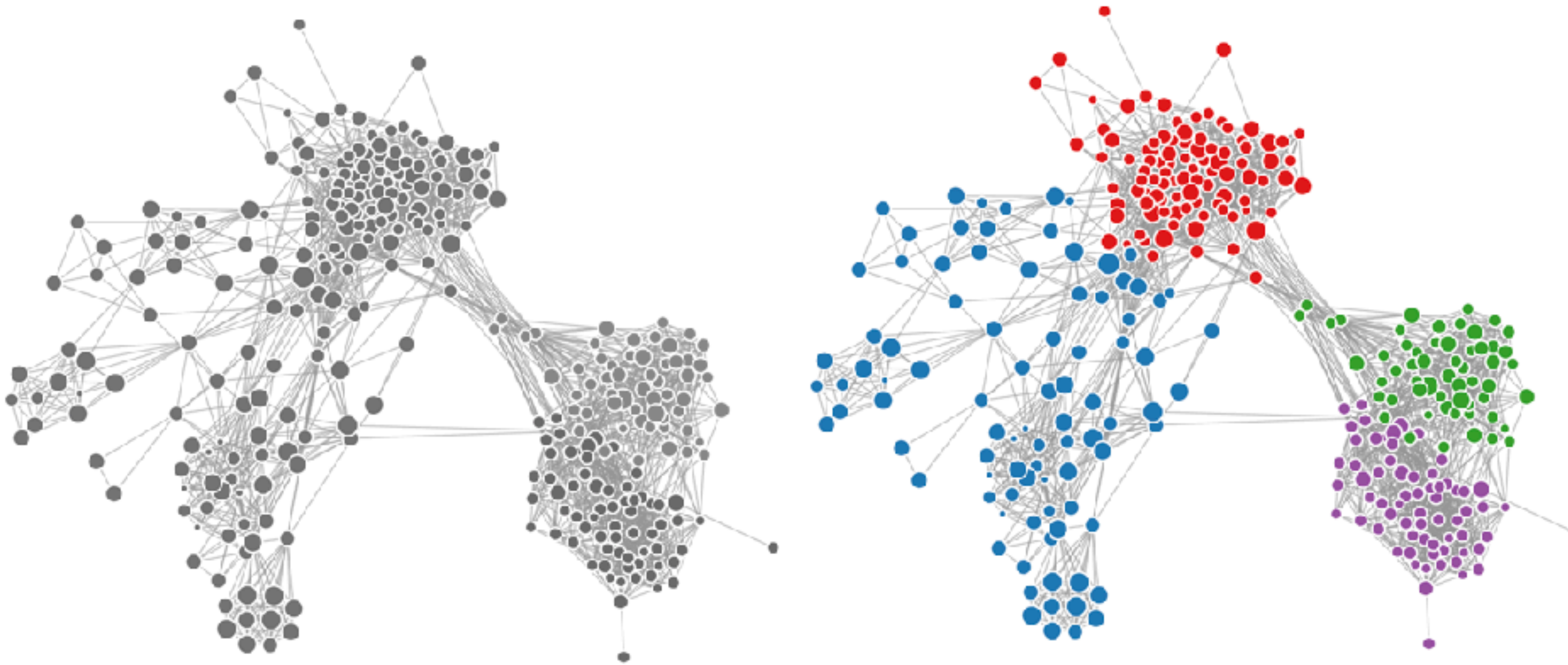
1. **The problem (10mins)**
   - Definition and motivation
   - Example application
2. **The approach (15 mins)**
   - Generative models
3. **Advanced topics: Multilayer networks (20mins)**
   - Mixed-membership factor models
   - *Layer interdependence (if time allows)*

Community detection: *the problem.*

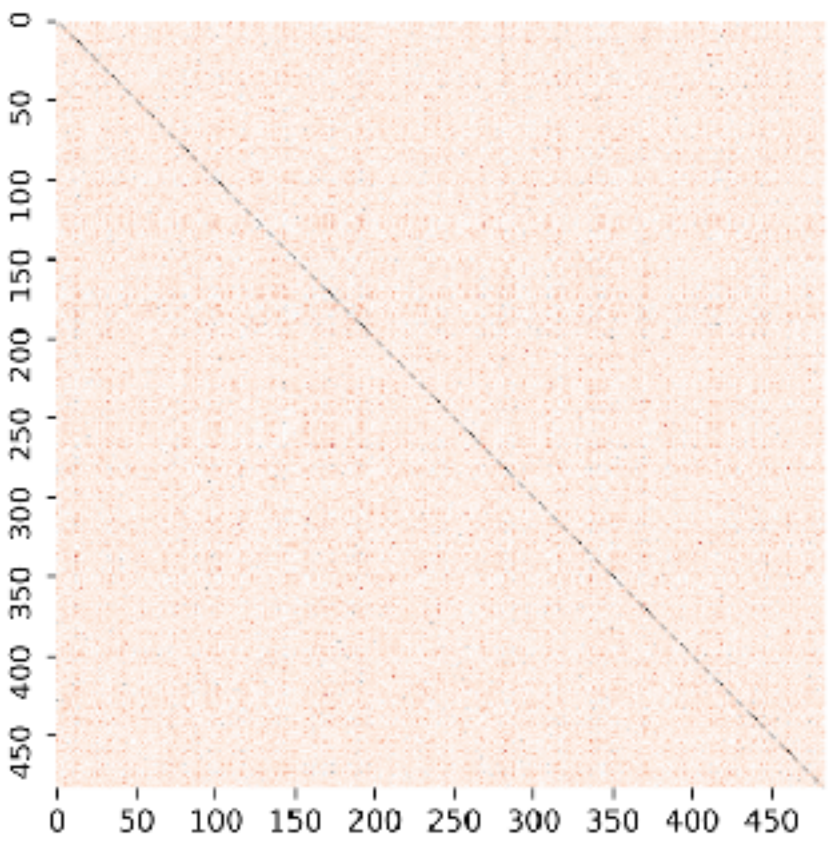
Finding groups of nodes that are more similar within the group than with those in the others.

Community detection: *the problem.*

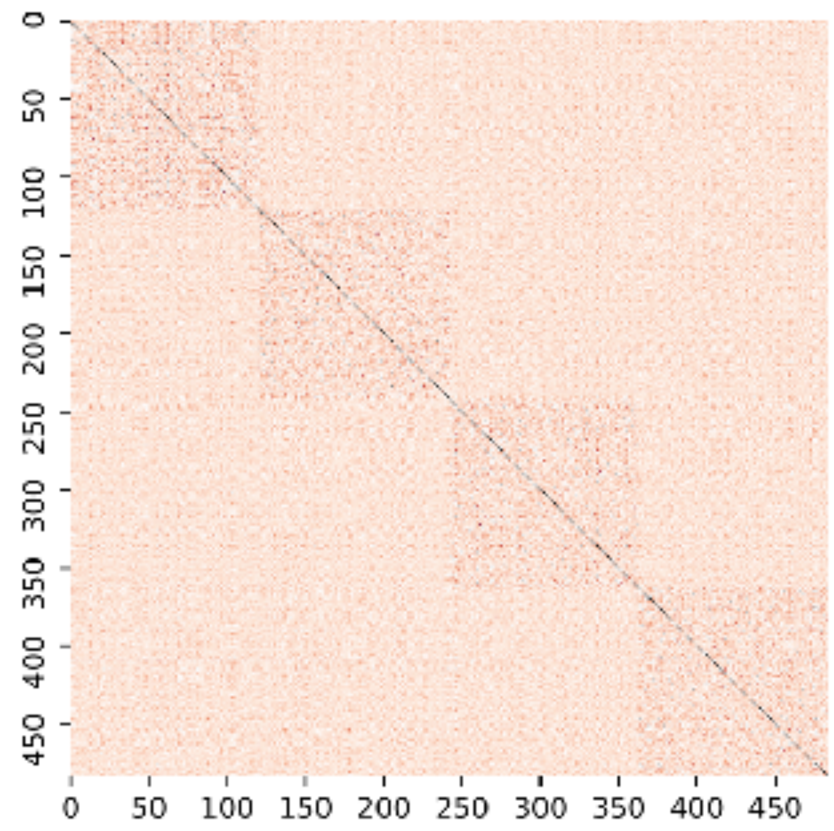
Finding groups of nodes that are more similar within the group than with those in the others.



Matrix representation



Adjacency matrix

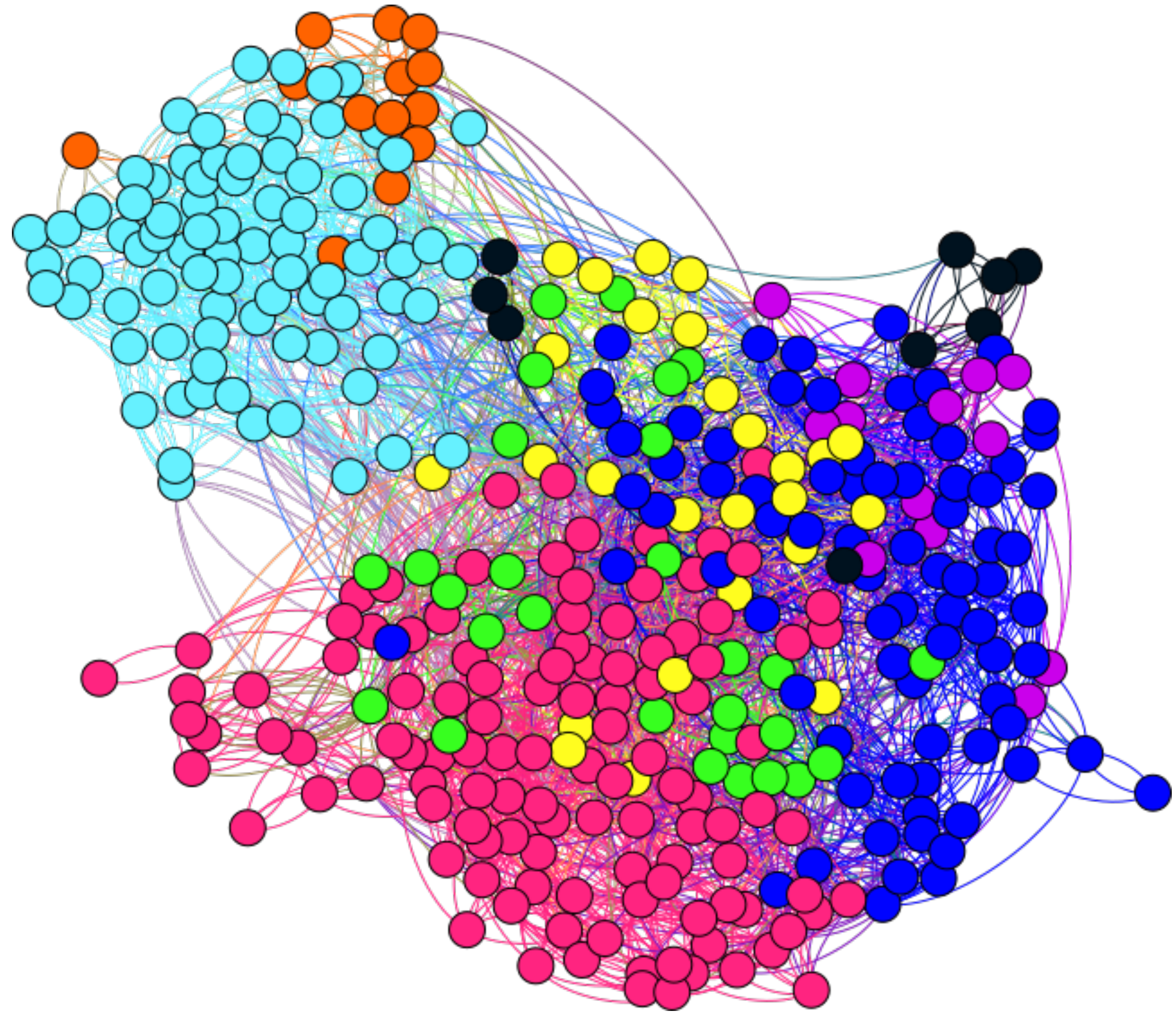


Adjacency matrix reordered by community membership

Applications

- Social support networks

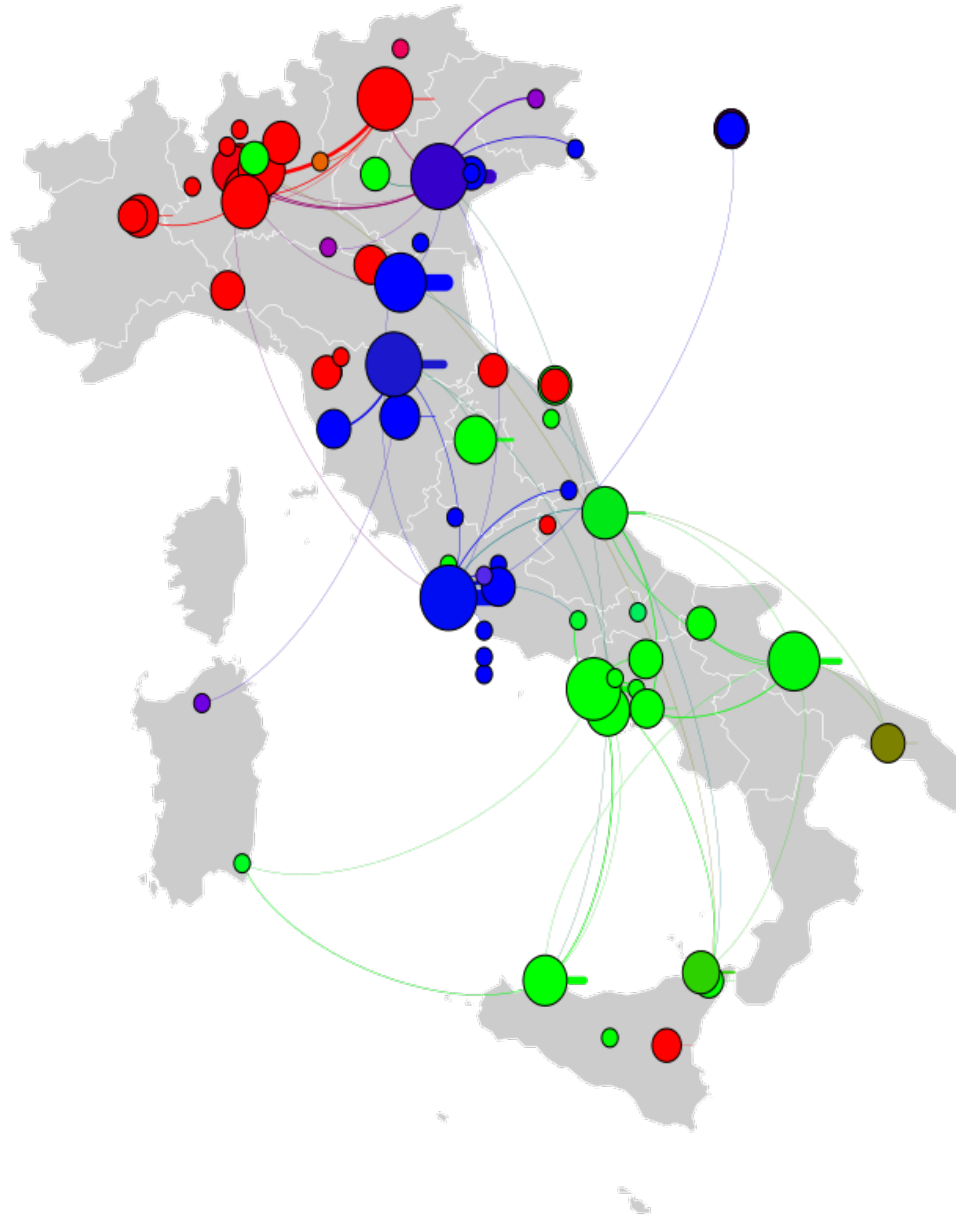
- Directed network
- Nodes are people
- Edges are types of social interactions (exchanging money, talk, worship, etc…)
    - 362 nodes, 6053 edges
    - 420 nodes, 7768 edges
- People belong to castes and follow different religions
- In village 2 there are two separated hamlets



E Power. Social Support Networks and Religiosity in Rural South India. Nature Human Behaviour. 1:0057 (2017).
C De Bacco, E Power, D B Larremore , C Moore. Phys. Rev. E 95, 1981–10 (2017).

Applications

- Faculty hiring: investigate the existence and structure of institutional hiring networks



The problem's outcomes

Membership vectors

U_i

```
0 0 0 0.0790622 0
3 0.0533999 0 0.139296 0
76 0 0 0.0805629 0.10241
127 0 0 0 0.207446
177 0 0.0738358 0.158358 0.024311
1 0 0 0.185704 0.0796013
6 0.0270201 0 0 0.0864358
17 0 0 0.192816 0.0215193
34 0 0 0.211593 0.0223843
```

V_i

```
0 0 0.00483982 0.00224297 0
3 0 0.01271 0.00688526 0
76 0 0.00115951 0.0137771 0.00347393
127 0 0.00605217 0.0168567 0
177 0.00267315 0 0.00549152 0.0149187
1 0 0.017335 0.00262135 0.0033609
6 0.00186055 0 0 0.00820299
17 0 0.0184664 0 0
34 0 0.0183341 0 0.00153079
```

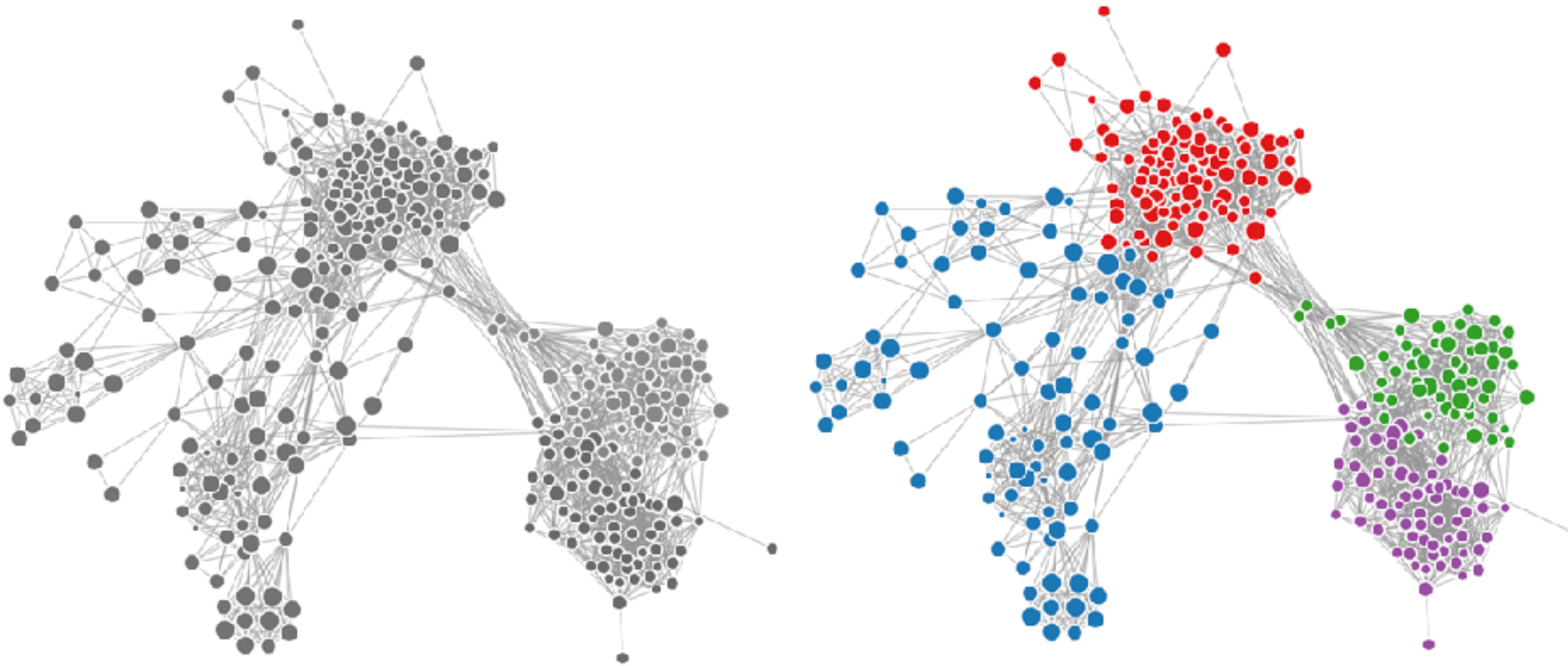
Affinity matrix

W^a

```
a= 0
0 0 0 2.66716
0 3.79189 0 0
0 0 2.75449 0
2.0515 0 0 0

a= 1
0 0 0 1.30754
0.031726 1.07808 0.0602251 0
0.271431 0.256695 1.07306 0
0.750644 0 0 0

a= 2
0 0 0 1.98783
0 2.65475 0 0
0 0 1.82214 0
1.27939 0 0 0

a= 3
0 0 0 2.74153
0 1.96755 0.0878885 0
0 0 3.99113 0
1.93712 0 0 0
```

The problem's outcomes:

Communities (i.e. labels)

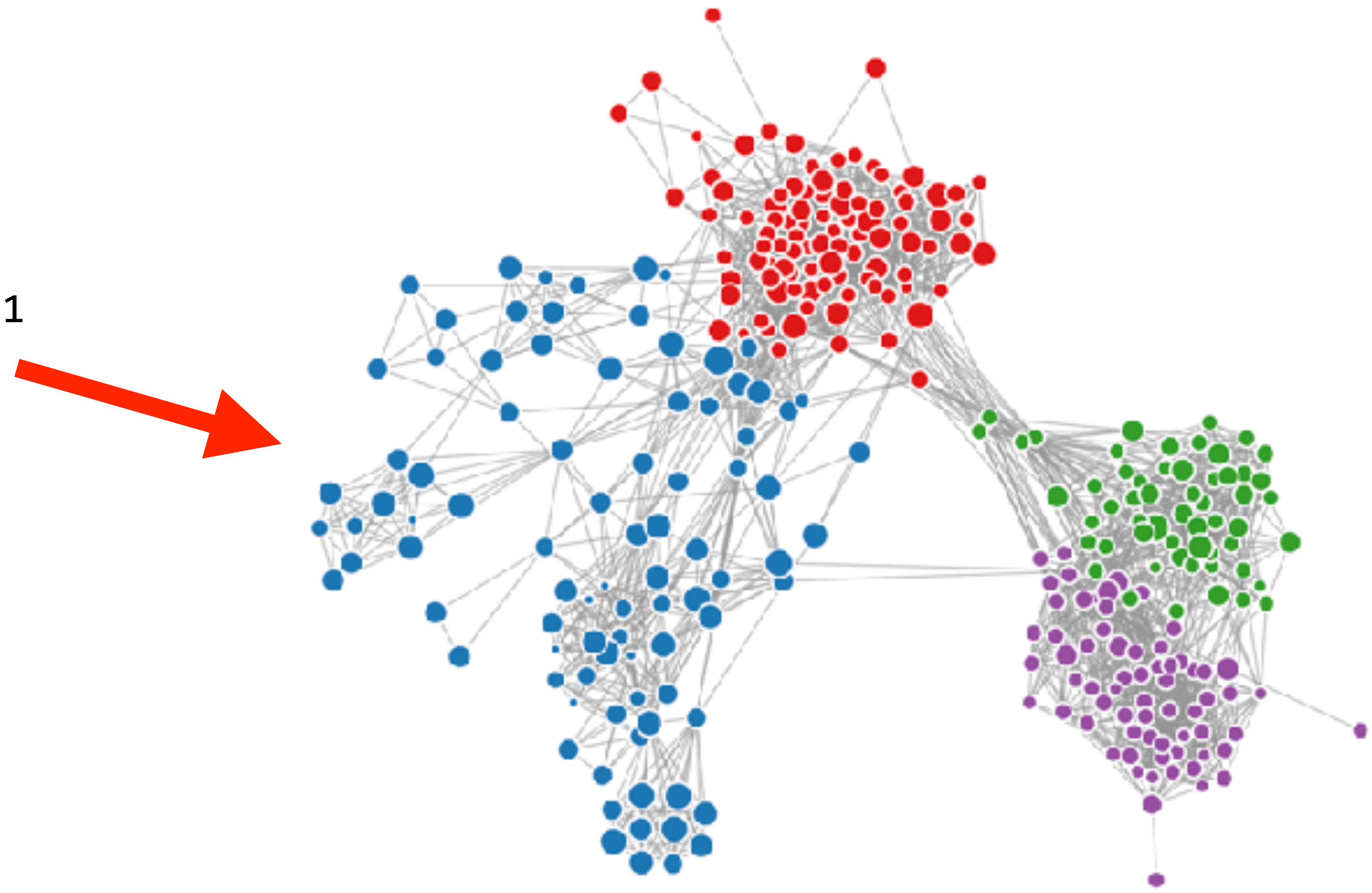


The problem's outcomes:

Communities (i.e. labels)

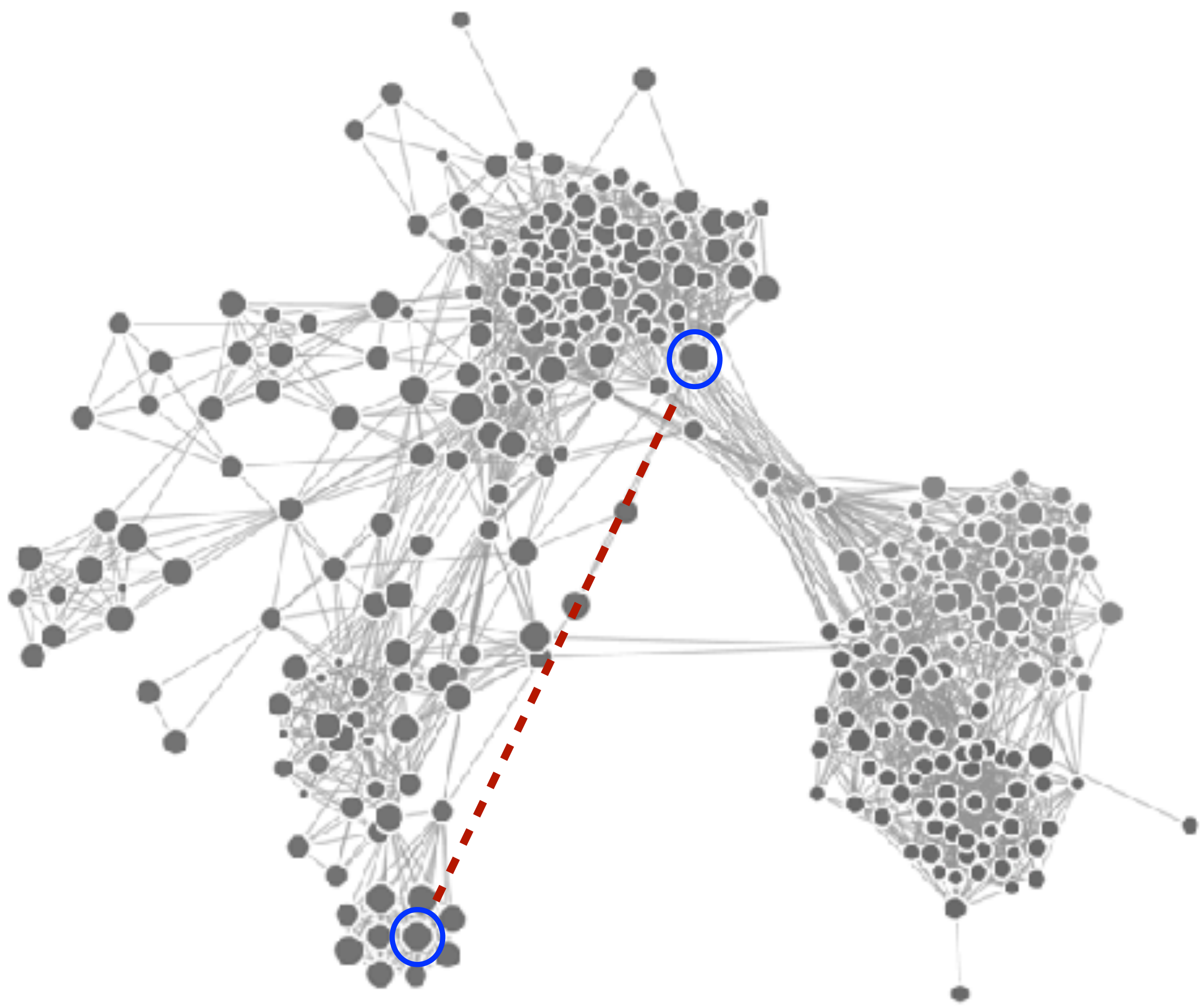
U_i

```
0 0 0 0.0790622 0
3 0.0533999 0 0.139296 0
76 0 0 0.0805629 0.10241
127 0 0 0 0.207446
177 0 0.0738358 0.158358 0.024311
1 0 0 0.185704 0.0796013
6 0.0270201 0 0 0.0864358
17 0 0 0.192816 0.0215193
34 0 0 0.211593 0.0223843
```



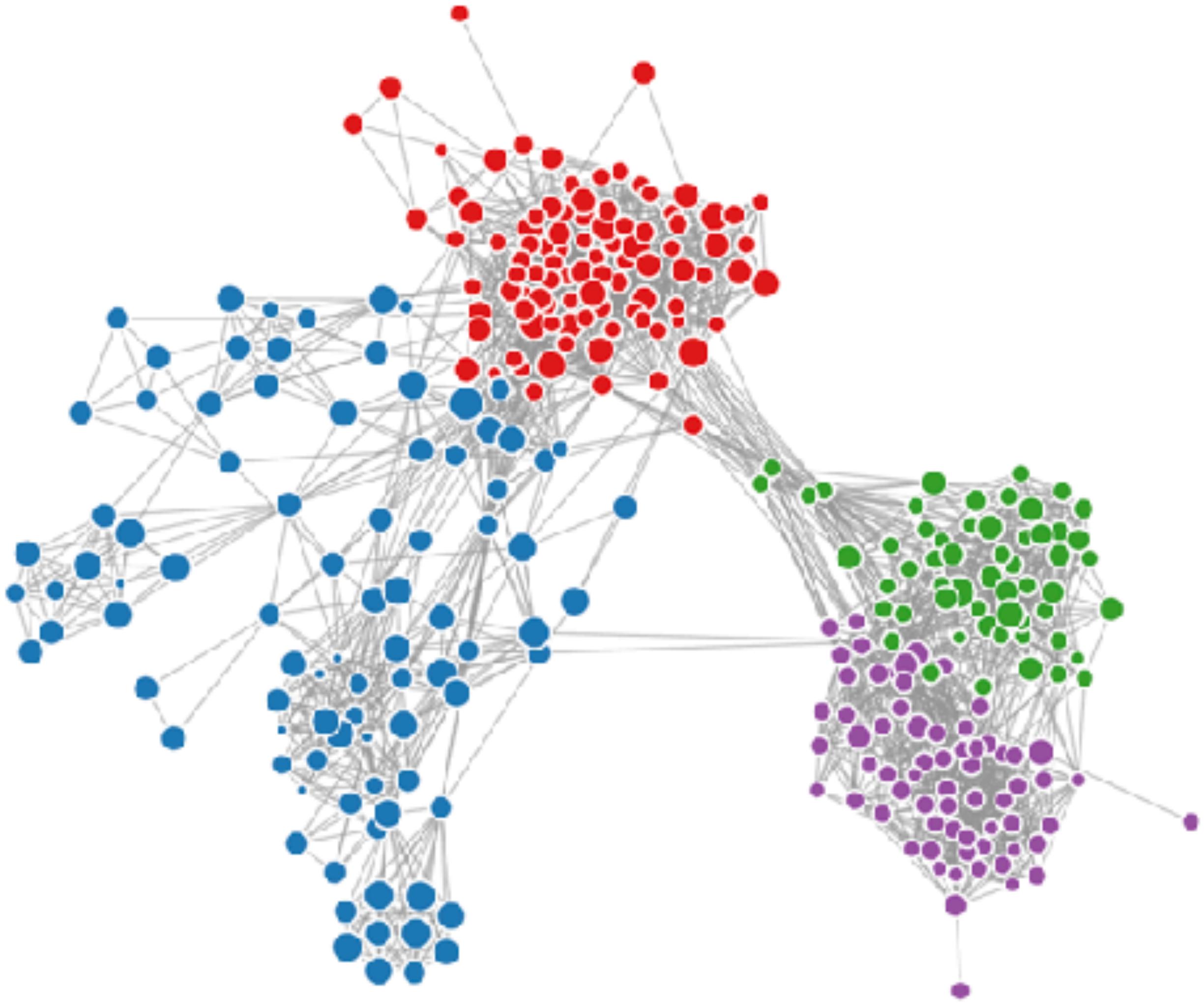
The problem's outcomes:

Probability of an interaction between two nodes



The problem's outcomes:

Data compression



- Input:

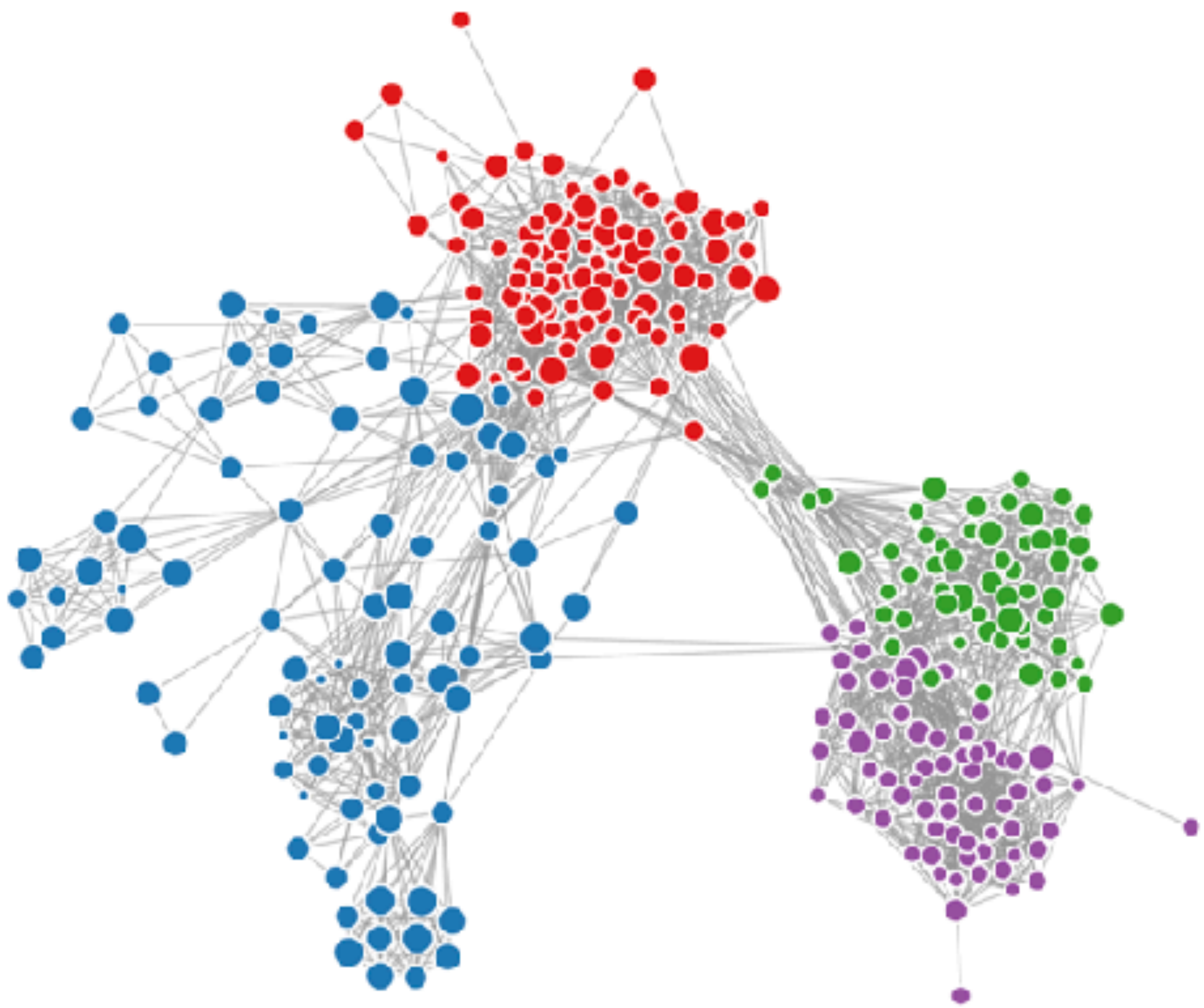
  $$A_{ij} : N \times N = O(N^2)$$

- Output:

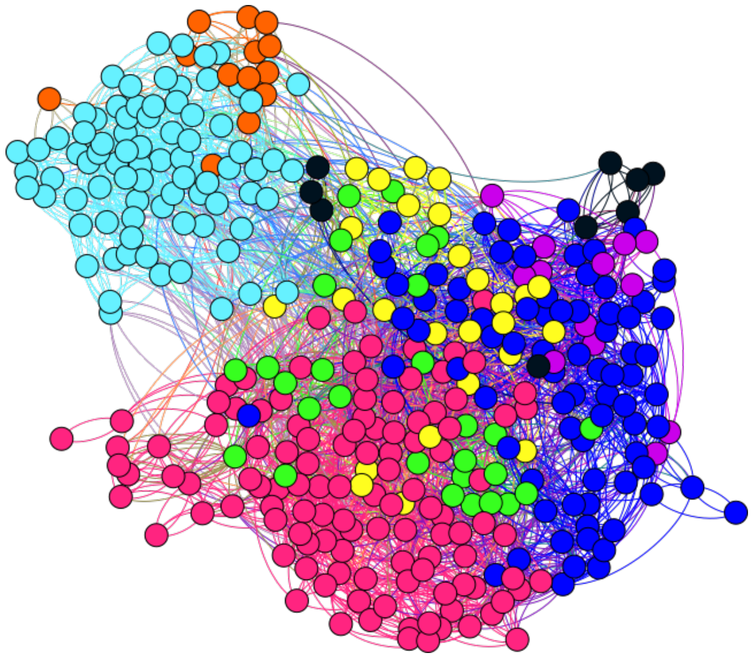
  $$u_i : N = O(N)$$

The problem's outcomes:

Network distribution

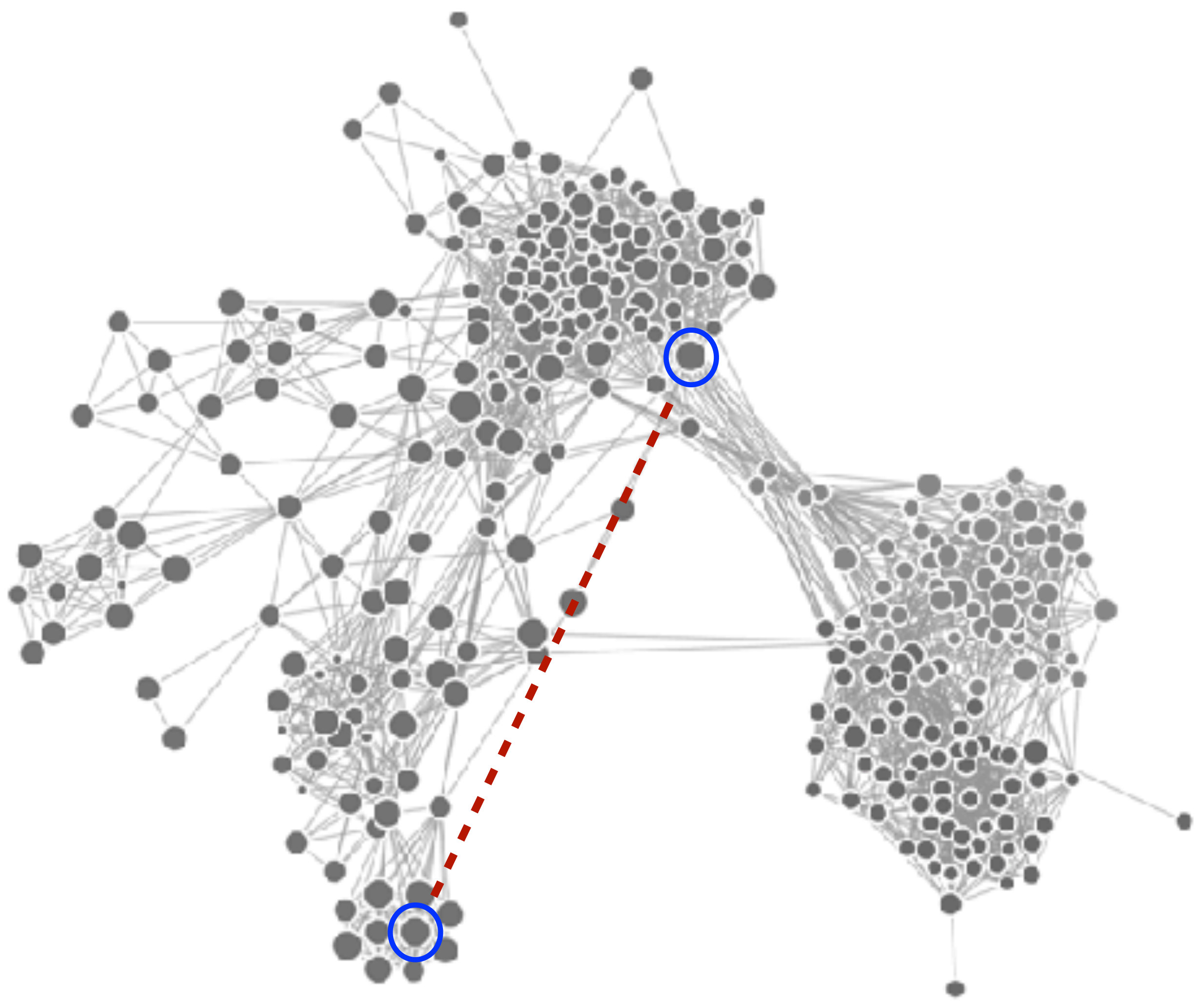


- Degree distribution
- Diameter
- Triangles
- Reciprocated edges
- …



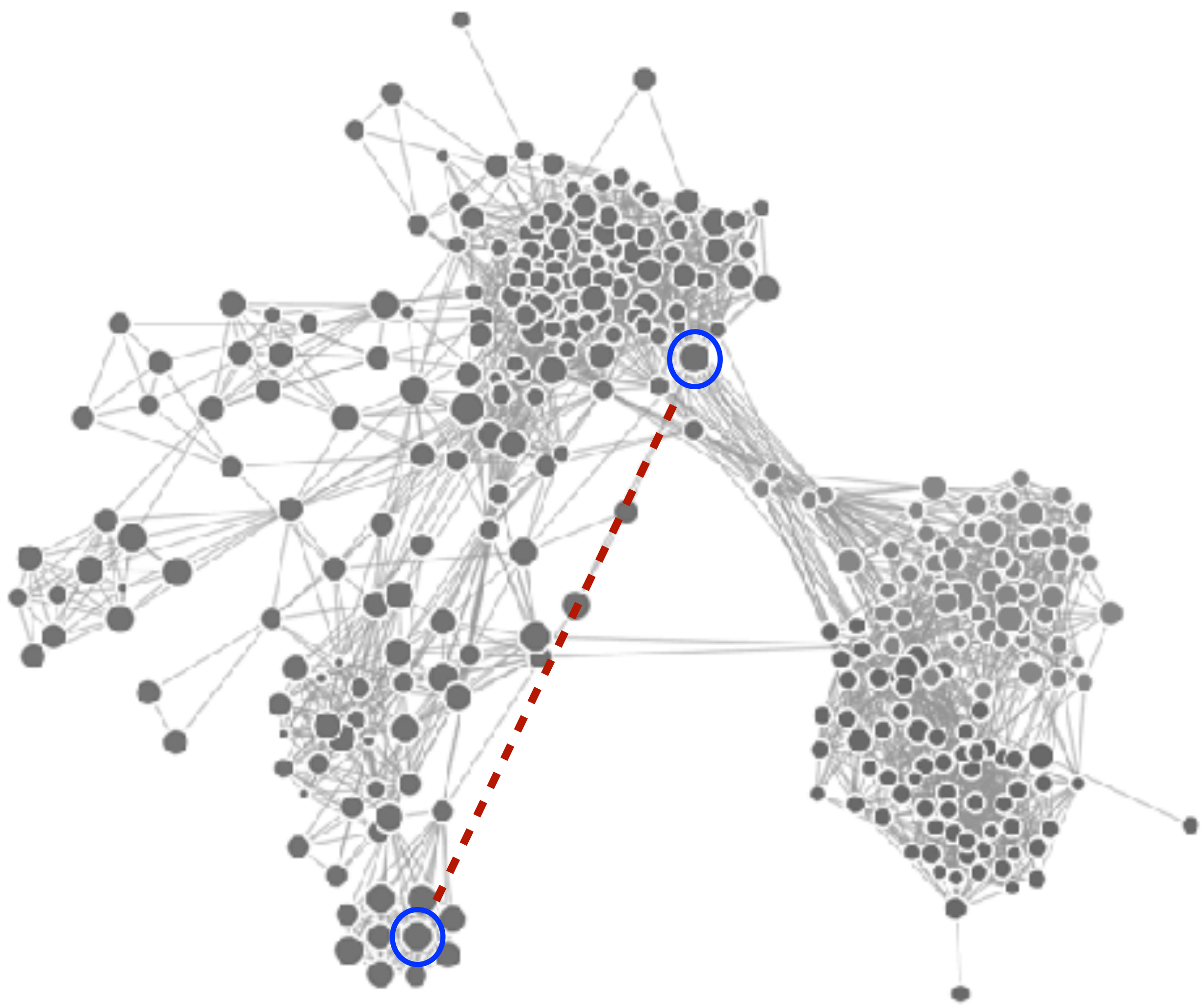
1. **The problem (10mins)**
   - Definition and motivation
   - Example application
2. The approach (15 mins)
   - Generative models
3. Advanced topics: Multilayer networks (20mins)
   - Mixed-membership factor models
   - Layer interdependence (if time allows)

Community detection on network: *the approach*
Generative models and hidden variables



What is the probability that the two nodes are connected?

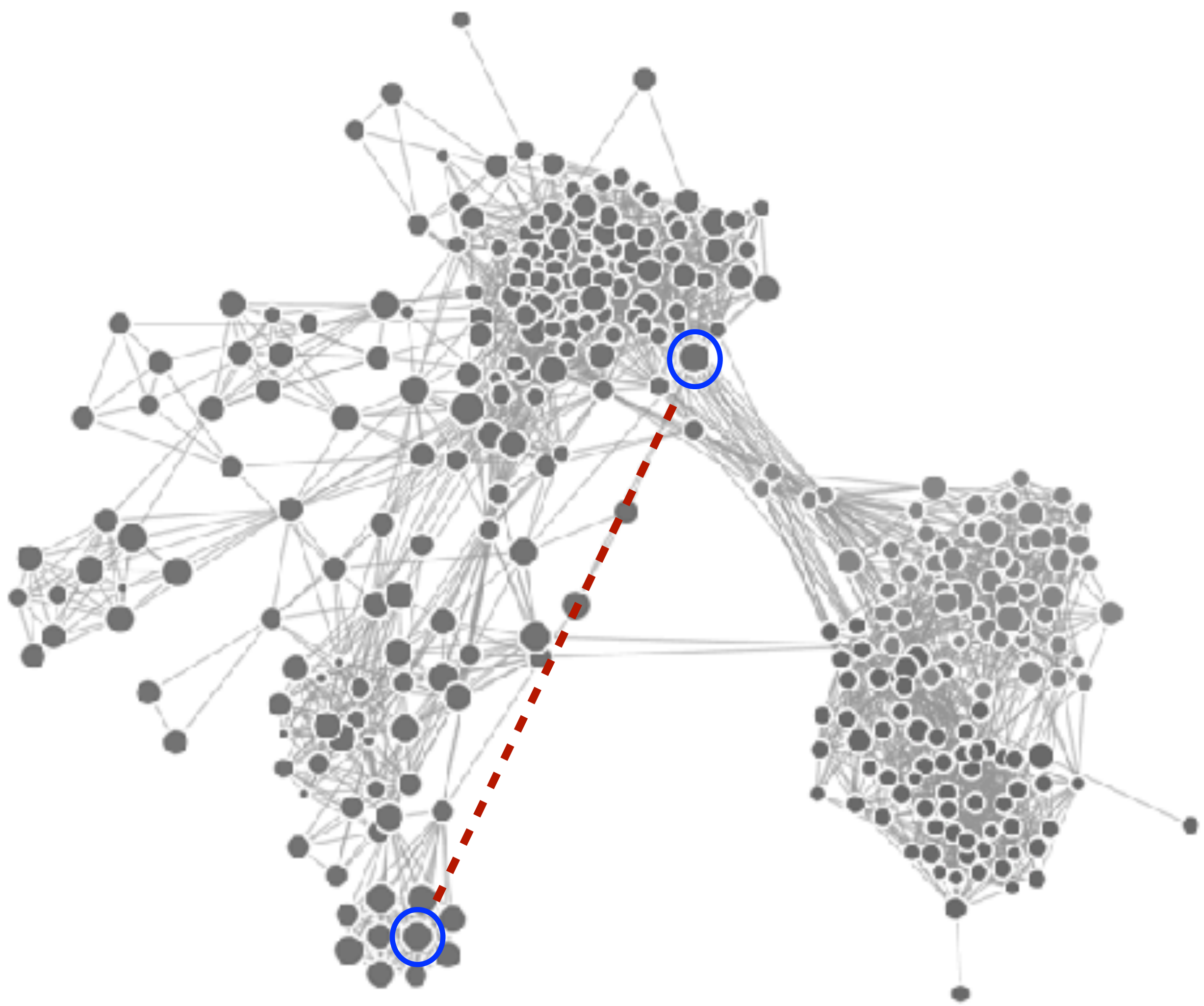
Community detection on network*: the approach*
Generative models and hidden variables



What is the probability that the two nodes are connected?

Hard to tell by just looking at the nodes (edges are unknown)… 50% ?

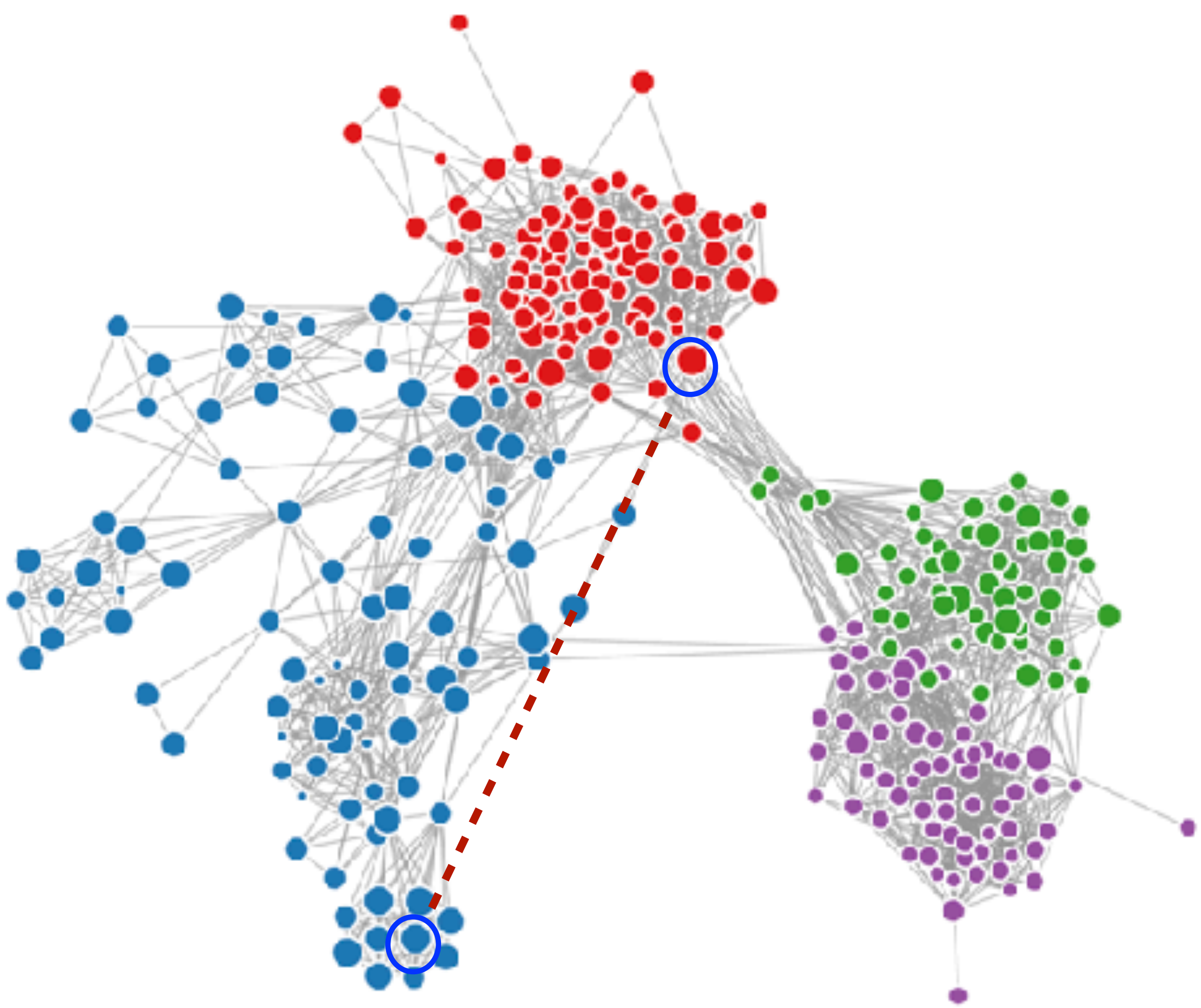
Community detection on network*: the approach*
Generative models and hidden variables



What is the probability that the two nodes are connected?

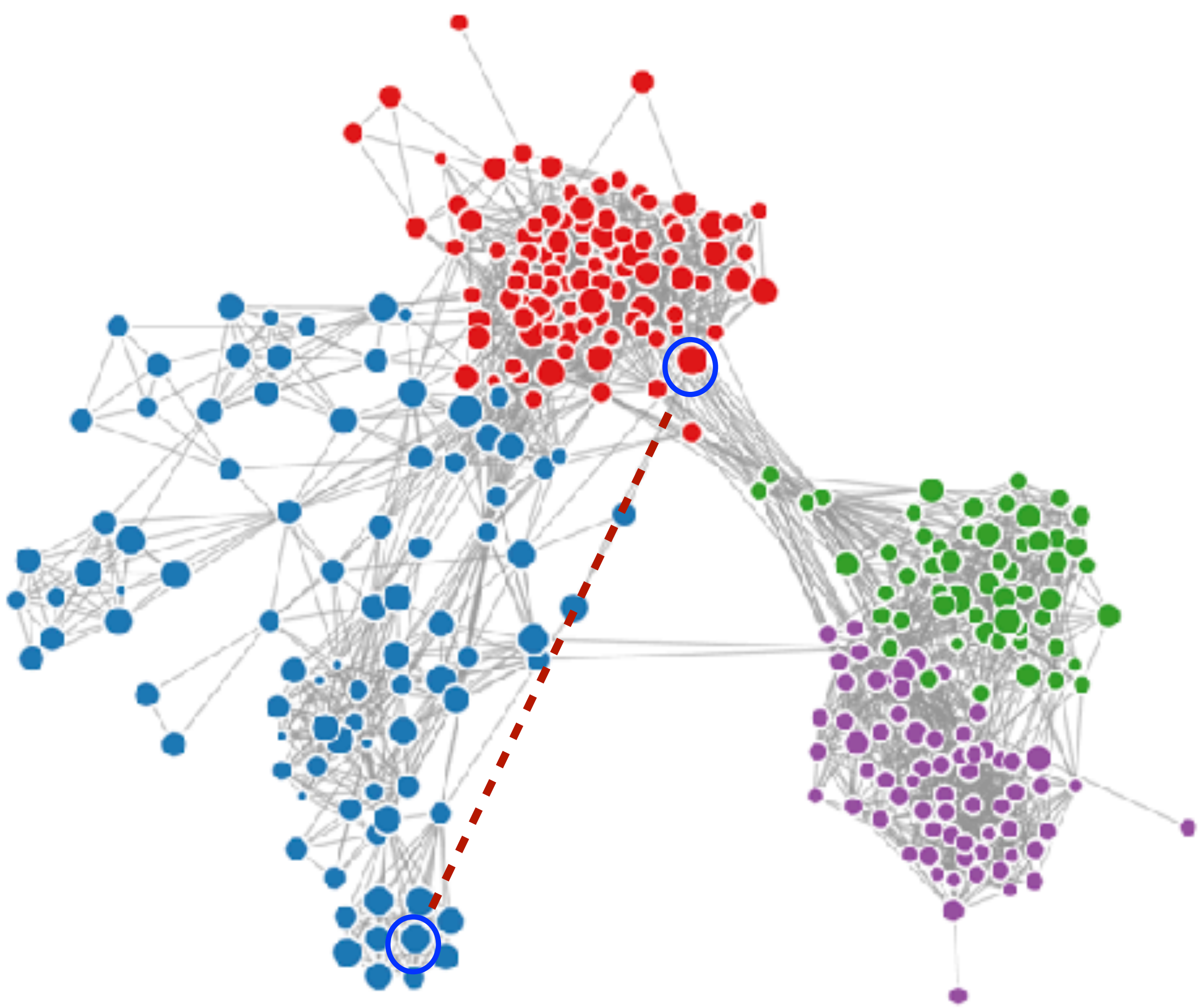
Hard to tell by just looking at the nodes (edges are unknown)… 50% ?
Or, if I know E=# of edges, then E/N(N-1) ?!?

Community detection on network*: the approach*
Generative models and hidden variables



What if I told you that one node is red and the other is blue?

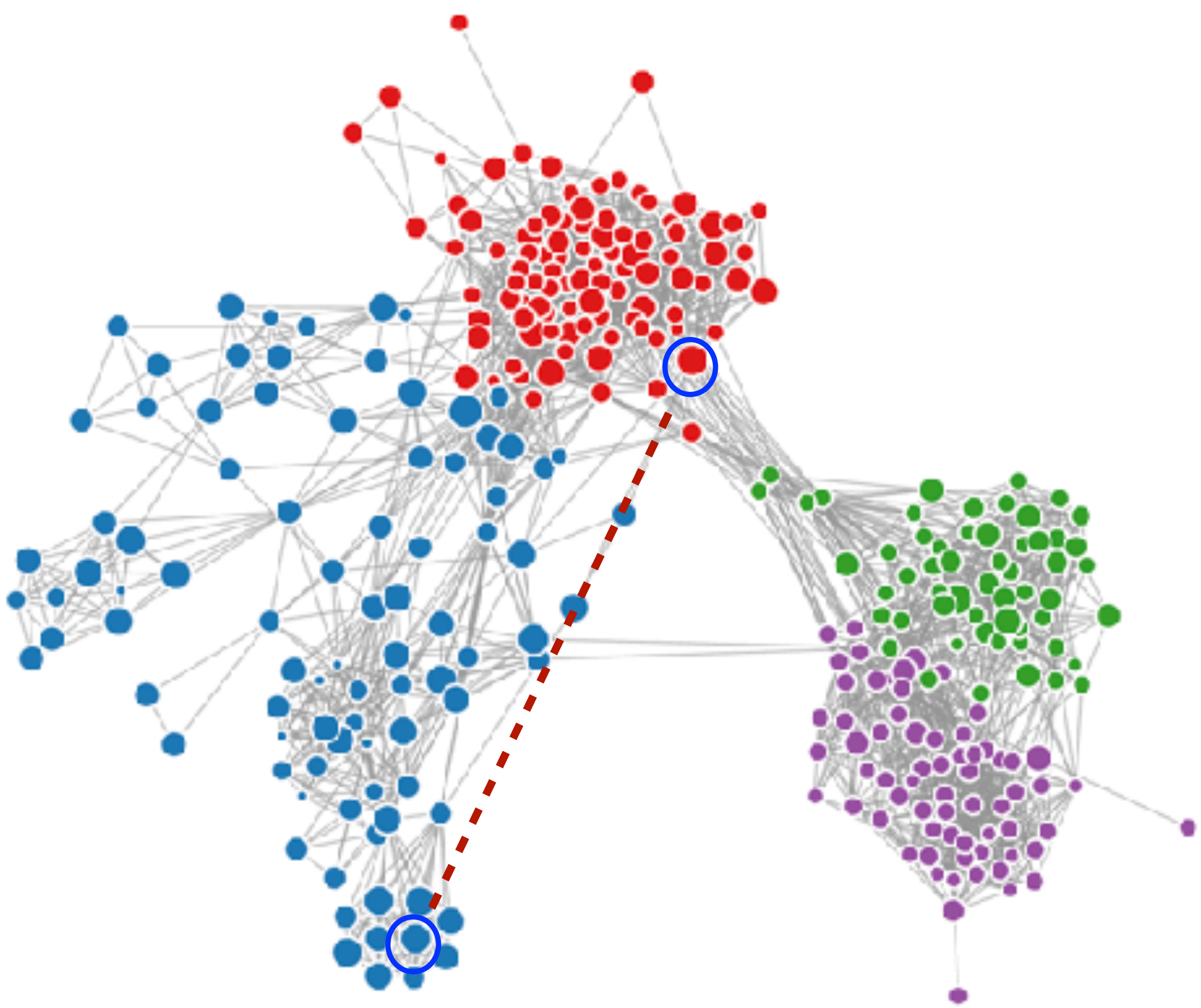
Community detection on network: *the approach*
Generative models and hidden variables



What if I told you that one node is red and the other is blue?

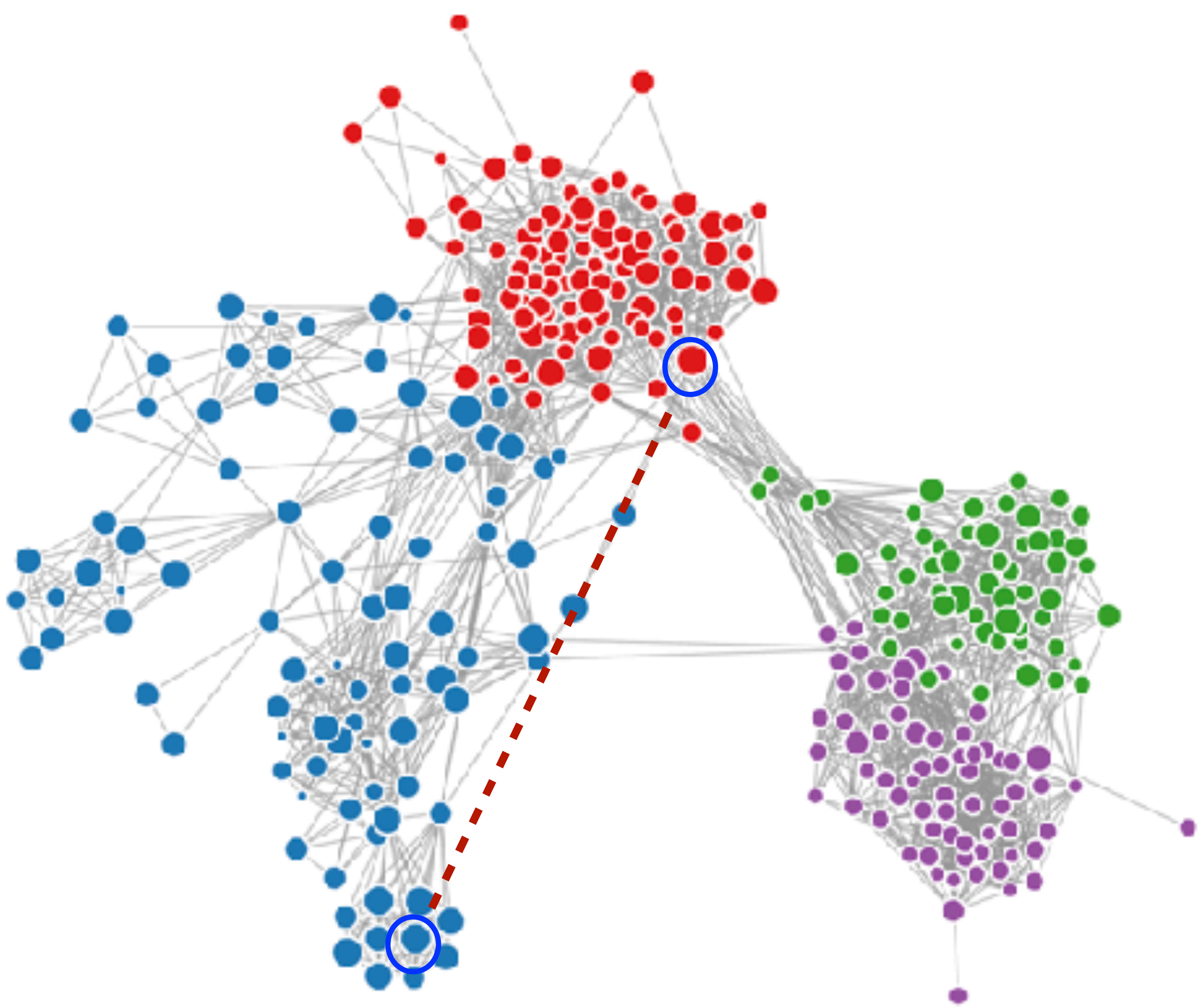
Now that's easier!
2% ?

Community detection on network*: the approach*
Generative models and hidden variables



Knowing what group the nodes belong to, made that easier…

Community detection on network*: the approach*
Generative models and hidden variables

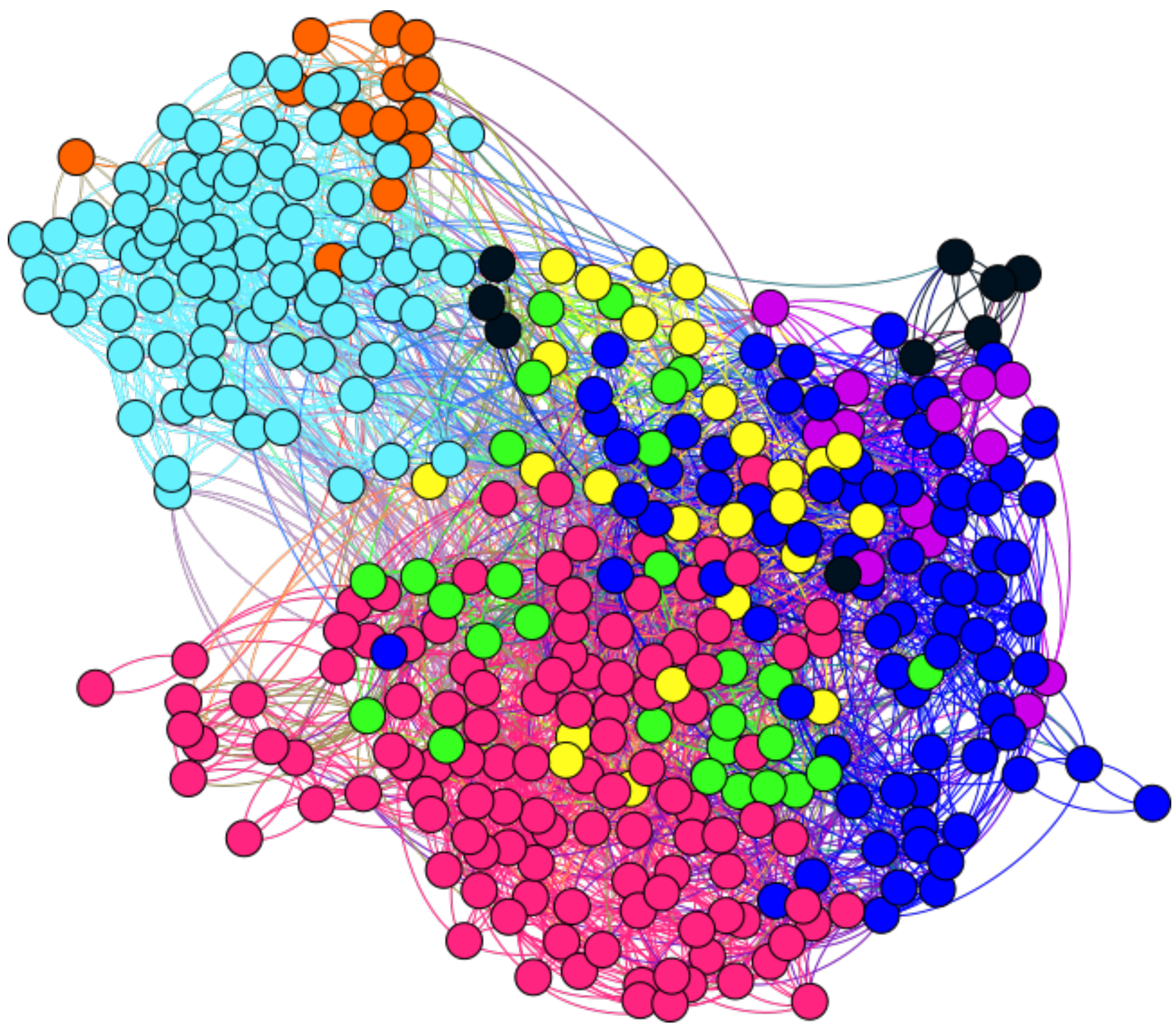


Knowing what community the nodes belong to, made that easier…

Because we are assuming that the edge we observe depend on what communities nodes belong to!

—-> Generative Model

Community detection on network*: the approach*
Generative models and latent variables



Assume that nodes have a latent variable

u_i

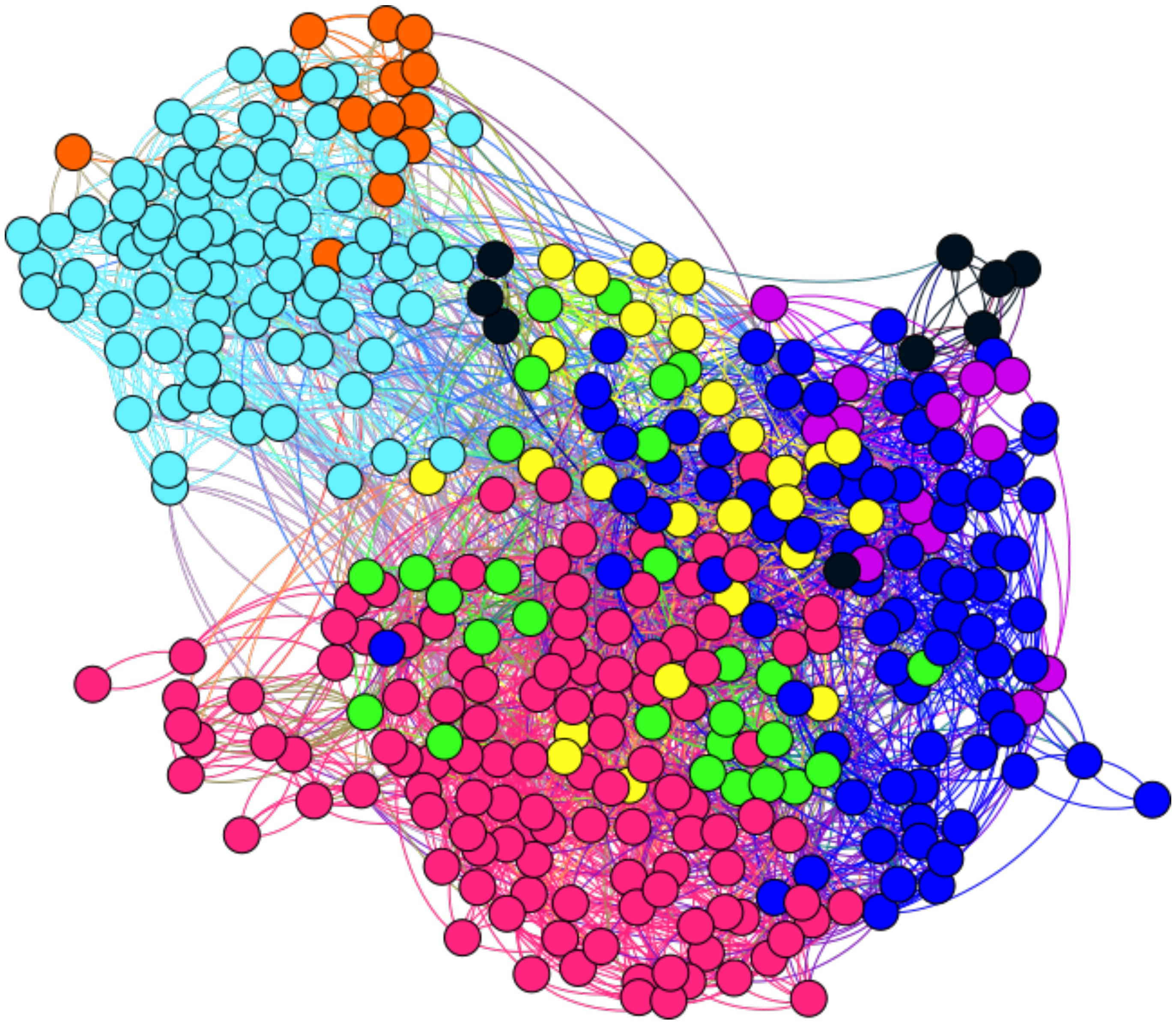
that controls how they interact with each other

Example: caste, religion, etc…

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix}$$

Mixed-membership

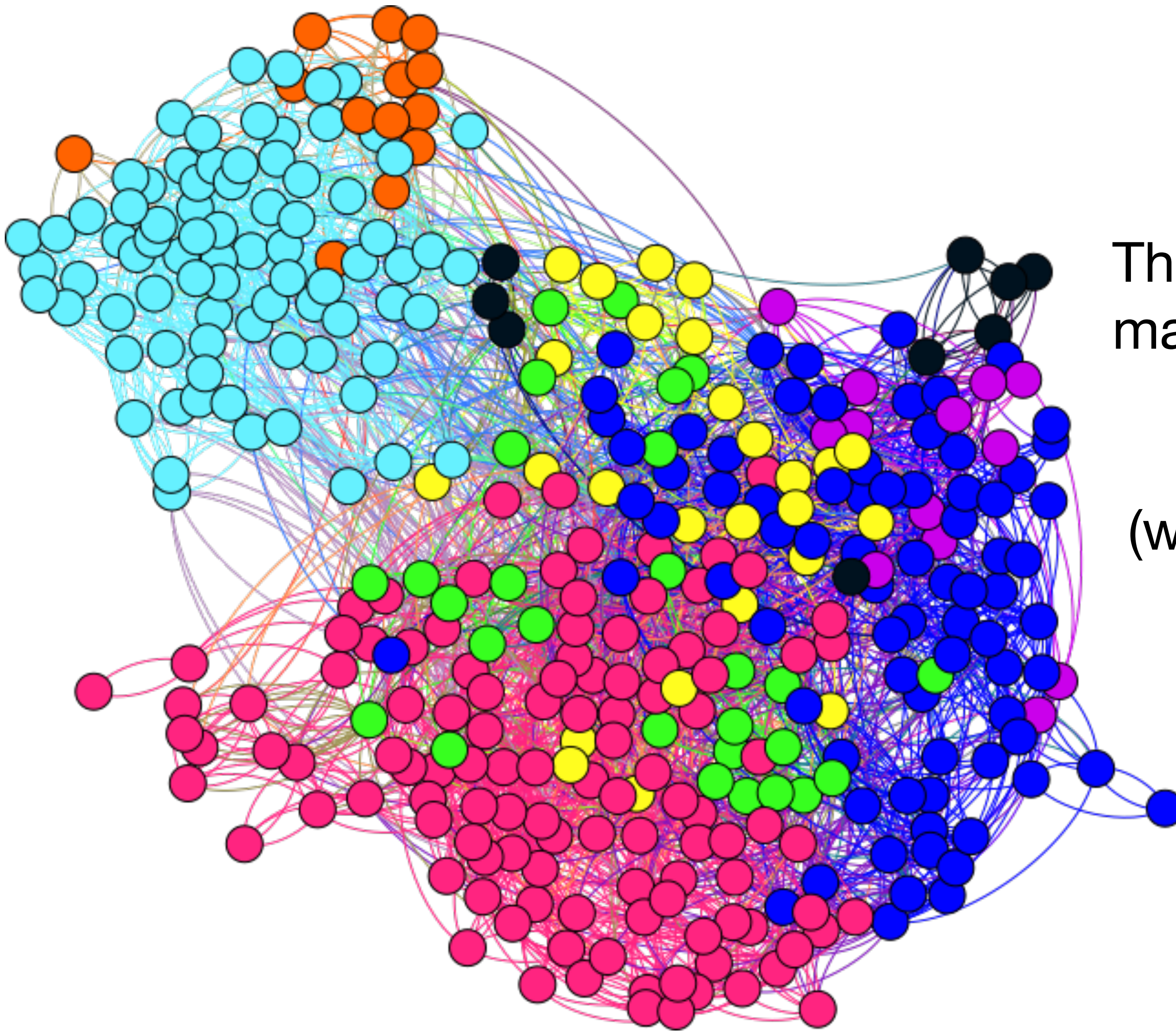
Community detection on network*: the approach*
Generative models and latent variables



Given latent variables u_i and u_j:

$$P_{ij} = f(u_i, u_j)$$

Community detection on network*: the approach*
Generative models and latent variables



Given latent variables u_i and u_j:

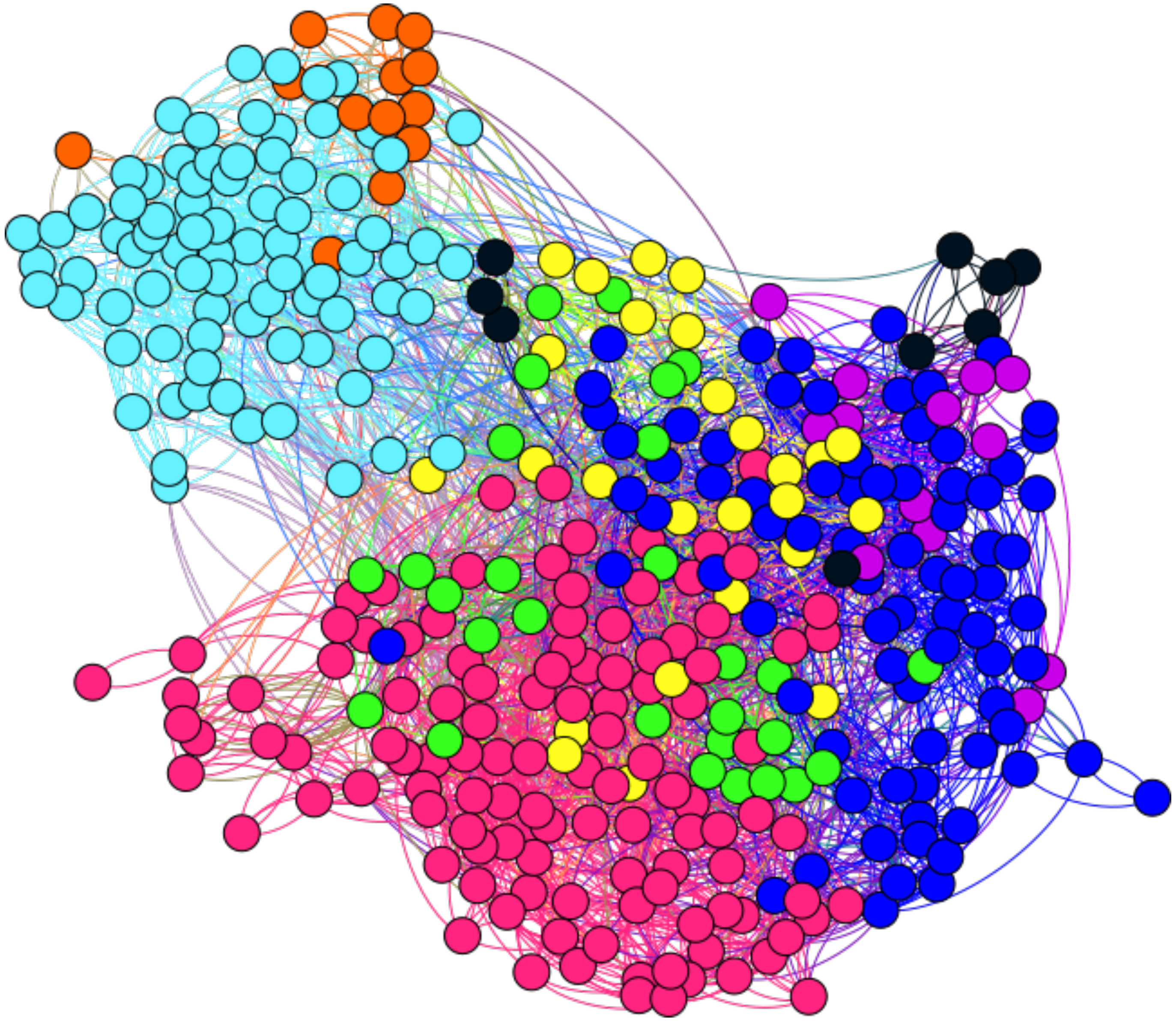
$$P_{ij} = f(u_i, u_j)$$

The observed data is the adjacency matrix entry:

A_{ij}

(we do not observe P_{ij})

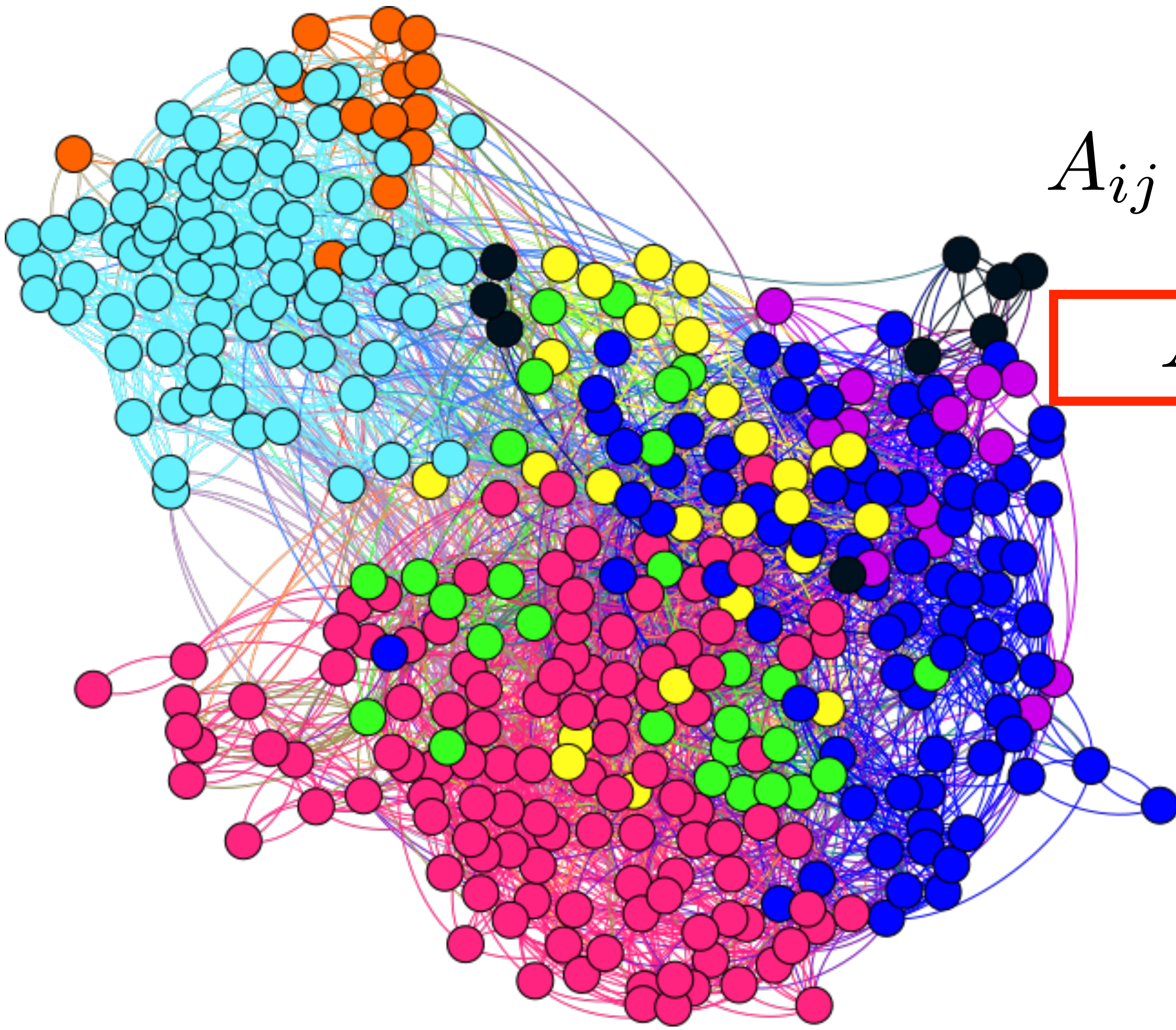
Community detection on network*: the approach*
Generative models and latent variables



$$A_{ij} \sim pdf(a_{ij}; u_i, u_j)$$

Community detection on network*: the approach*

Generative models and latent variables

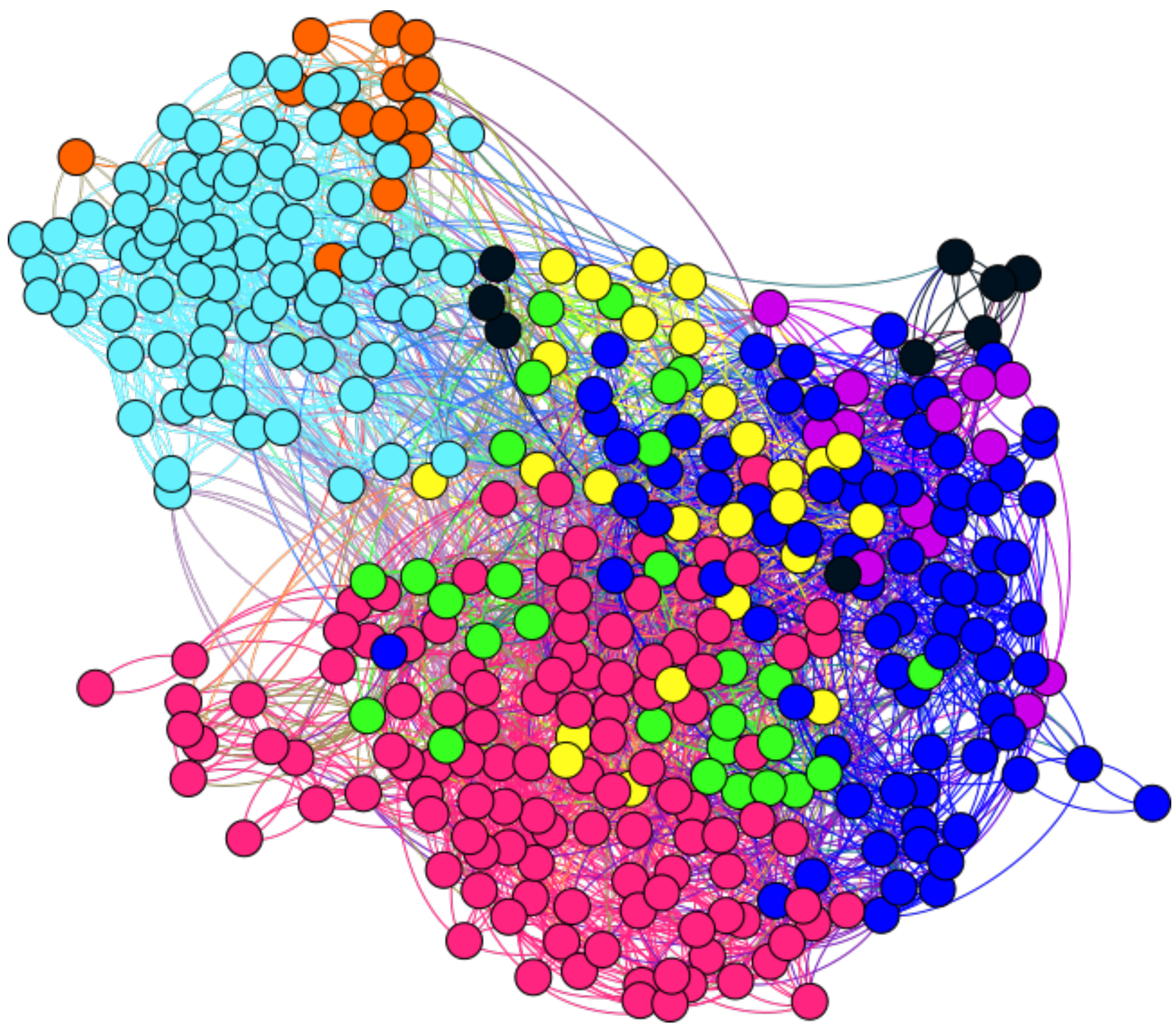


$$A_{ij} \sim pdf(a_{ij}; u_i, u_j)$$

$$A_{ij} \sim Bern(p_{ij} = \sigma(u_i \cdot u_j))$$

$$A_{ij} \sim Poi(\lambda_{ij} = u_i \cdot u_j)$$

Community detection on network*: the approach*
Mixed-membership



$$A_{ij} \sim Poi(\lambda_{ij} = u_i \cdot u_j)$$

The dot product of factor defines a distance (metric) in a latent space:

if u_i and u_j are close in this space, then higher probability to interact

$$u_i \cdot u_j = \sum_k u_{ik}\, u_{jk}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix}$$

k_1: (yoga and running)
k_2: (vegetarian)
...

Community detection on network*: the approach*
Directed

Two types of membership:
- *out-membership*: preferences
- *in-membership*: attributes

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \quad v_i = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

$$u_i \cdot v_j = \sum_k u_{ik} \, v_{jk}$$

k_1: (action and set in the mountain)
k_2: (comedy)
k_3: (duration less than 3 hours)
...

User i likes action movies and short ones:

$$u_i = \begin{bmatrix} 0.8 \\ 0.0 \\ 0.5 \\ 0.0 \end{bmatrix}$$

Community detection on network: *the approach Directed*

Two types of membership:

- *out-membership*: preferences
- *in-membership*: attributes

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \quad v_i = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

$$u_i \cdot v_j = \sum_k u_{ik} \, v_{jk}$$

k_1: (action and set in the mountain)
k_2: (comedy)
k_3: (duration less than 3 hours)
...

User i likes action movies and short ones:

$$u_i = \begin{bmatrix} 0.8 \\ 0.0 \\ 0.5 \\ 0.0 \end{bmatrix}$$

Movie j is an action movie, not too short with a hint of comedy:

$$v_j = \begin{bmatrix} 0.6 \\ 0.2 \\ 0.1 \\ 0.0 \end{bmatrix}$$

$$u_i \cdot v_j = 0.8 \cdot 0.6 + 0.5 \cdot 0.1 = 0.53$$

Community detection on network*: the approach*

In matrix notation

$$A \approx UV^T$$

K

$$U = \begin{bmatrix} \cdots \\ u_{i1} \quad \cdots \quad u_{iK} \\ \cdots \end{bmatrix}$$

N

N

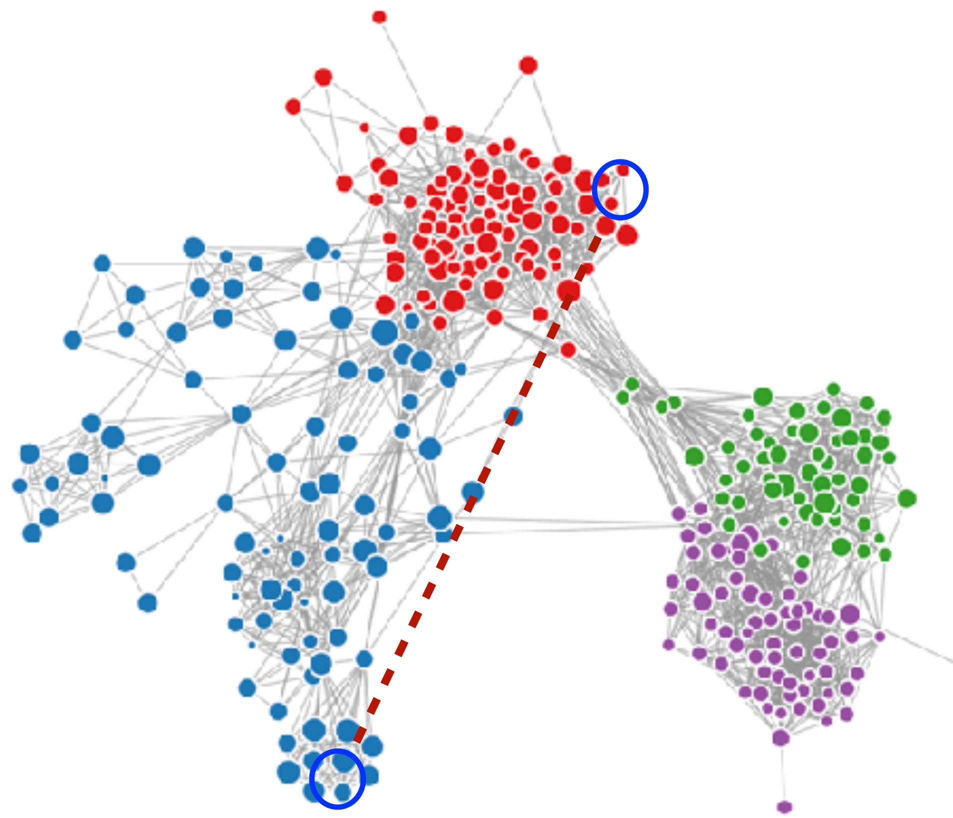
$$V^T = \begin{bmatrix} v_{i1} \\ \cdots \\ v_{iK} \end{bmatrix}$$

K

- **Latent variables** behind each node, that control how this **interacts** with others, i.e. how the network is generated

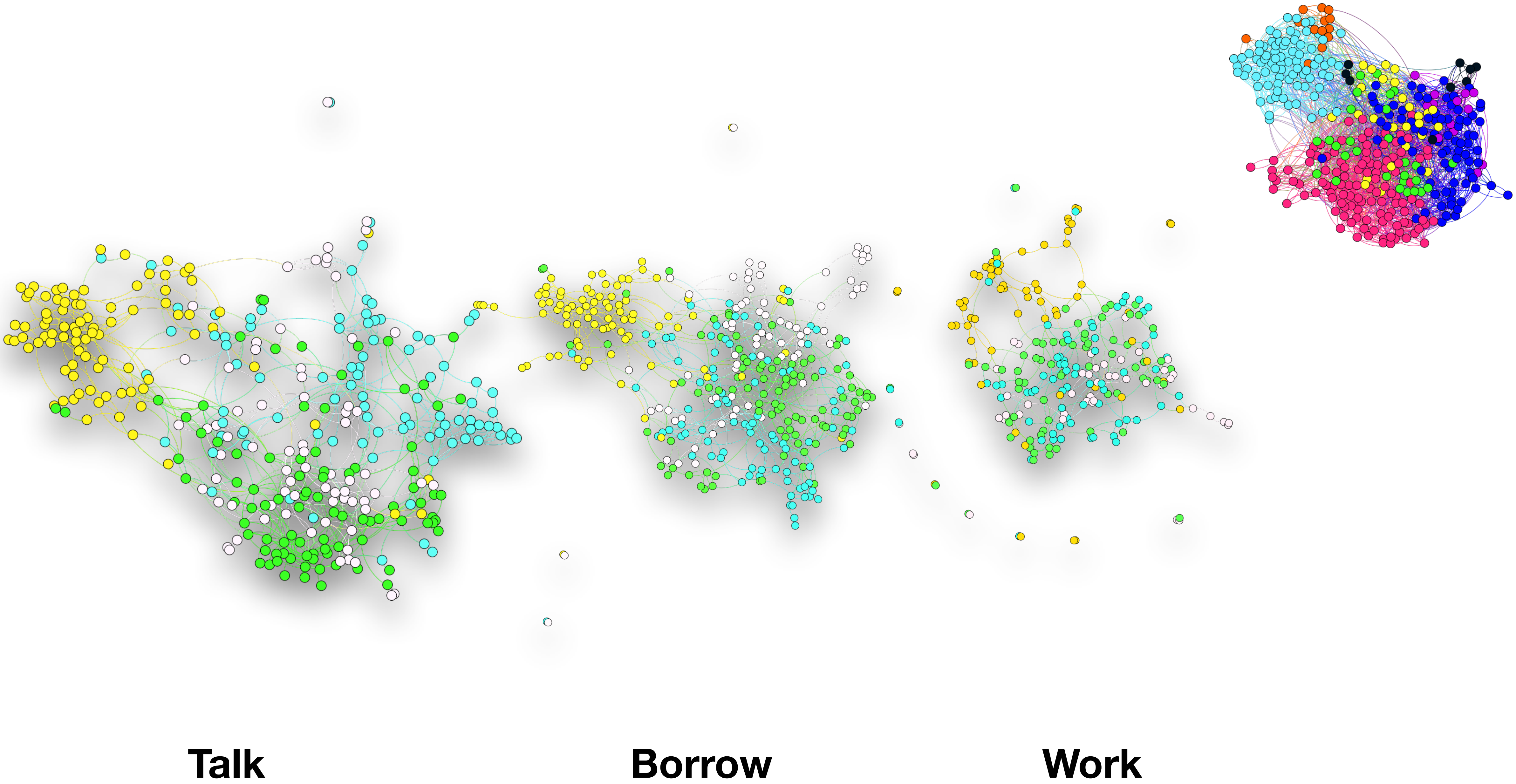
- Depending on the specific application: change/inform with domain knowledge the **details** of the model (e.g. dot product)

$$A_{ij} \sim Poi(\lambda_{ij} = u_i \cdot u_j)$$



1. The problem (10mins)
    - Definition and motivation
    - Example application
2. **The approach (15 mins)**
    - Generative models
3. Advanced topics: Multilayer networks (20mins)
    - Mixed-membership factor models
    - Layer interdependence (if time allows)

Advanced topic: *community detection in multilayer networks*

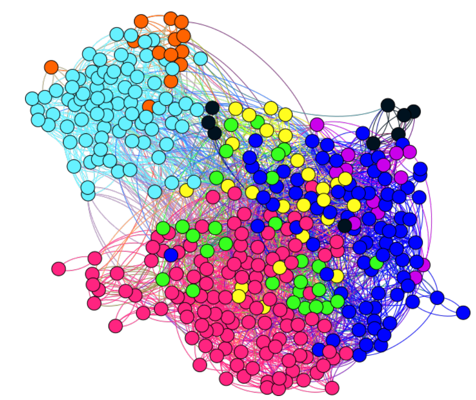


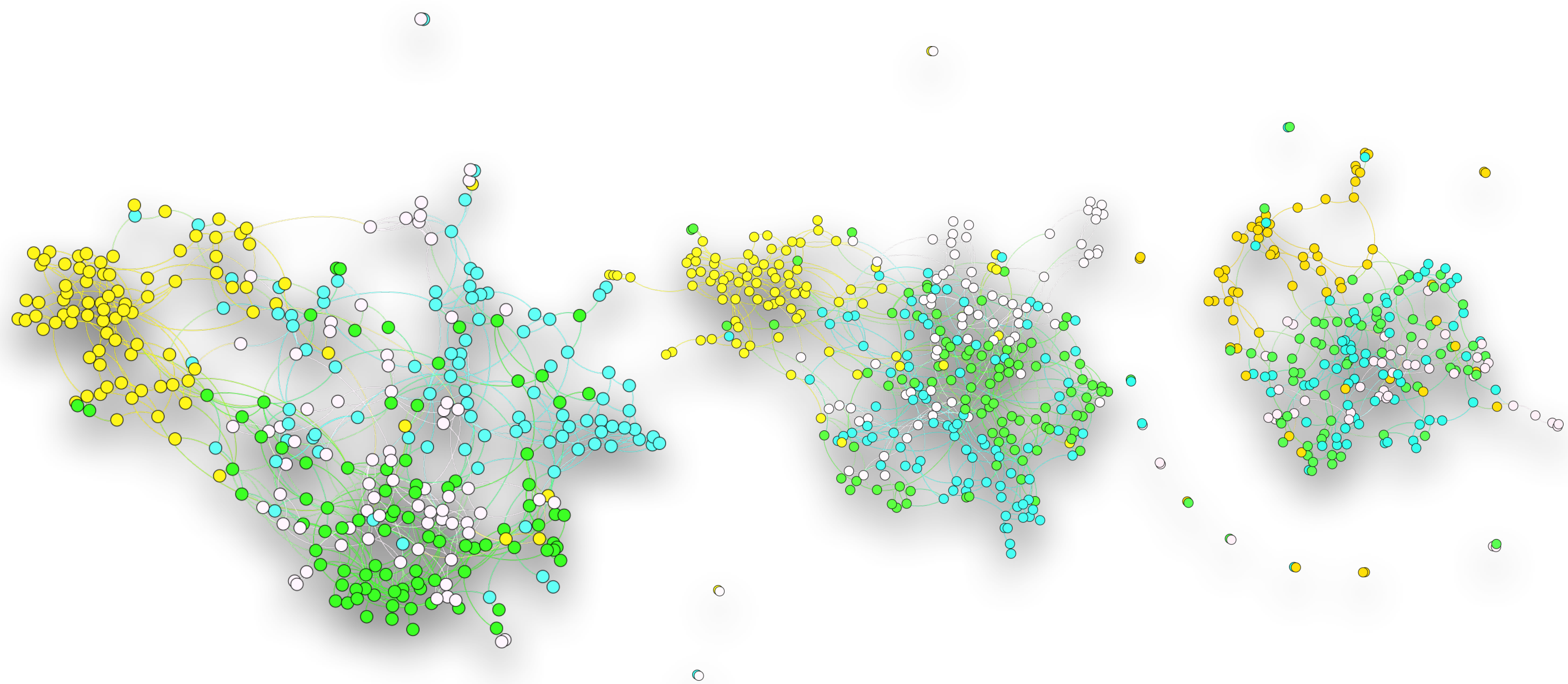
Advanced topic: *community detection in multilayer networks*

$$A^{\alpha}, \quad \alpha = 1, \ldots, L$$

L = # of layers

Nodes are the same in all layers
Edges change





Talk **Borrow** **Work**

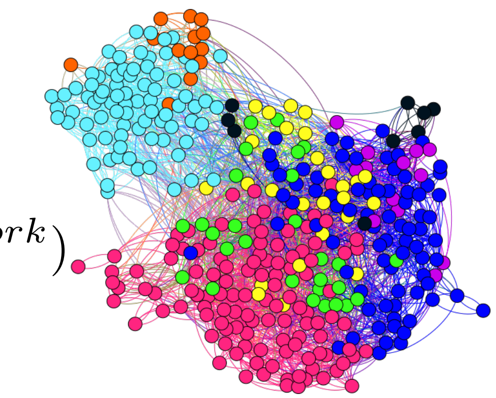
Advanced topic*: community detection in multilayer networks*

Naive approach 1: solve L separate problems

$$A_{ij}^{\alpha} \sim Poi(u_i^{\alpha} \cdot v_j^{\alpha})$$

$A_{ij}^{talk} \sim Poi(u_i^{talk} \cdot v_j^{talk})$

$A_{ij}^{work} \sim Poi(u_i^{work} \cdot v_j^{work})$



Talk **Borrow** **Work**

Advanced topic*: community detection in multilayer networks*

Naive approach 2: aggregate the layers into one

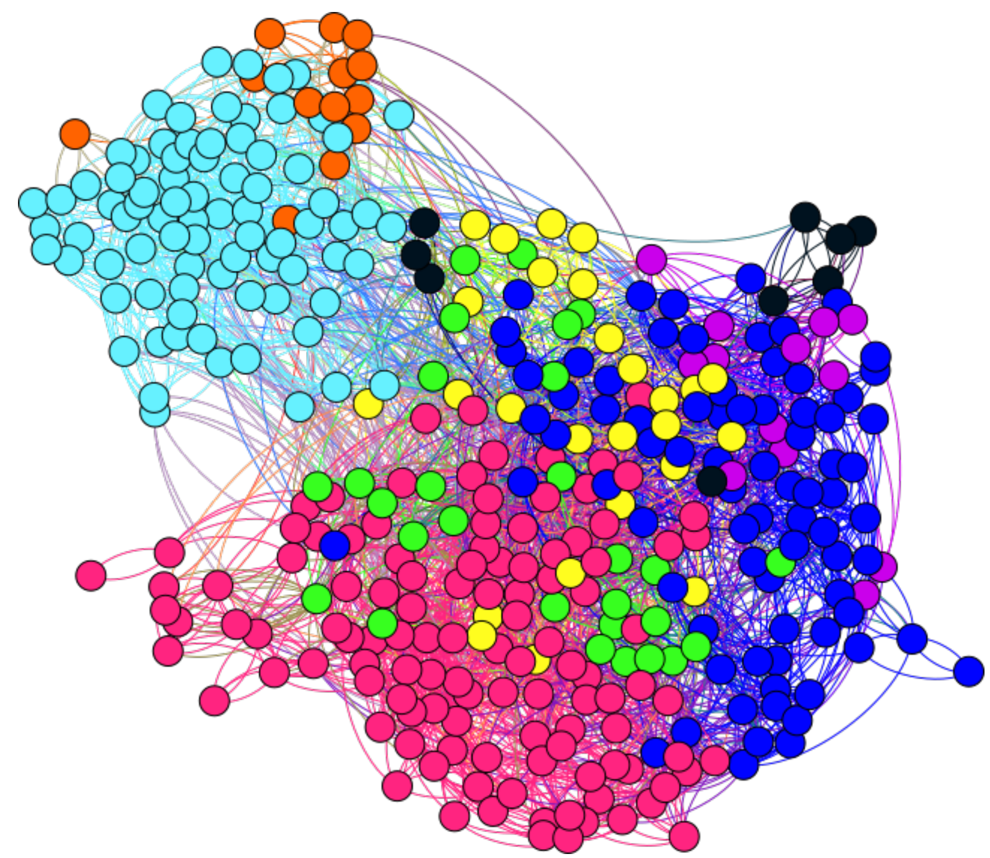
$$A_{ij}^{tot} = f(A_{ij}^1, \ldots, A_{ij}^L)$$

$$A_{ij}^{tot} \sim Poi(u_i \cdot v_j)$$

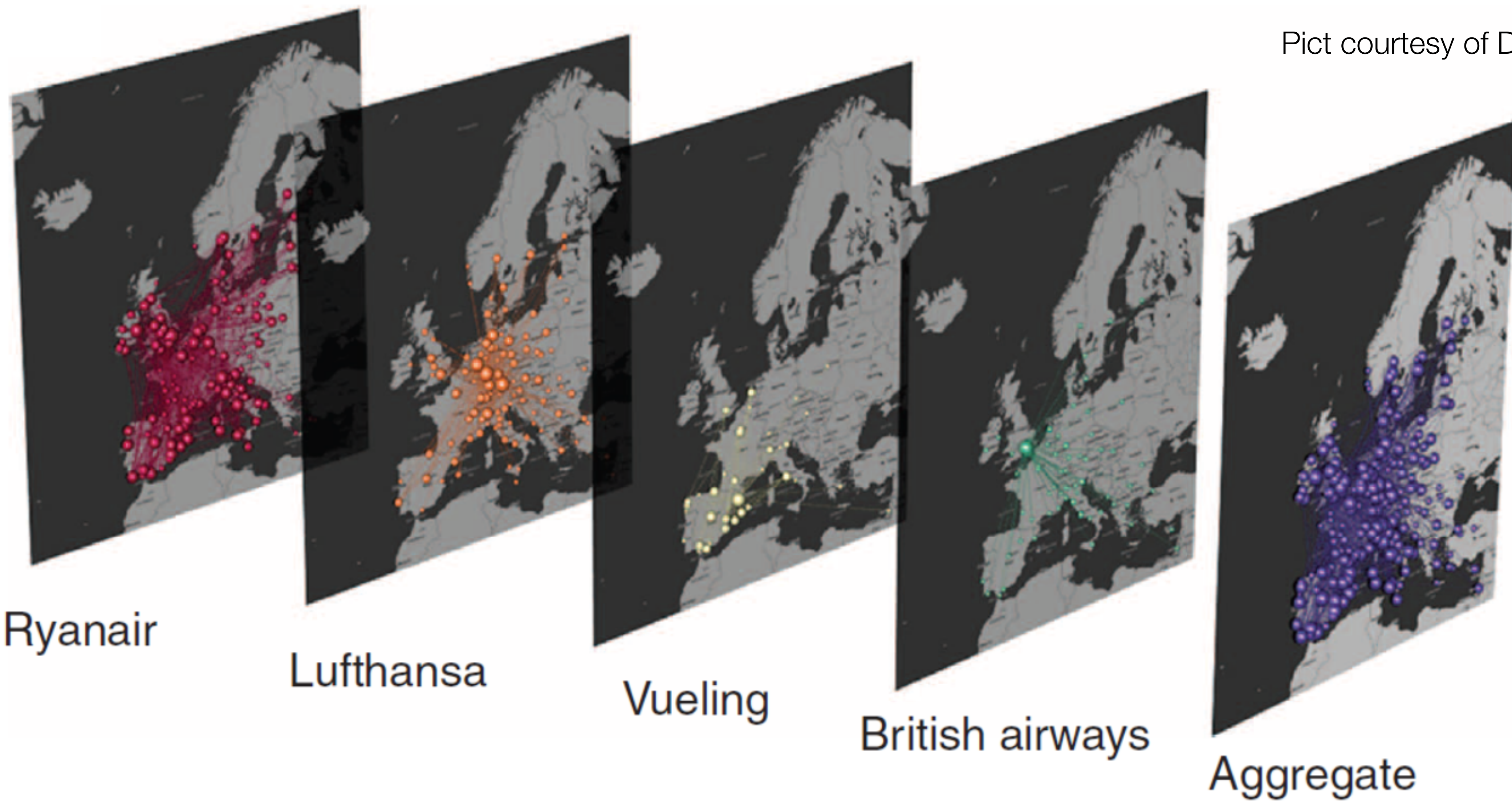
Examples of aggregation:

$$f(A_{ij}^1, \ldots, A_{ij}^L) = \sum_{\alpha=1}^{L} A_{ij}^{\alpha}$$

$$f(A_{ij}^1, \ldots, A_{ij}^L) = \prod_{\alpha=1}^{L} A_{ij}^{\alpha}$$



Advanced topic: *community detection in multilayer networks*
Example: Air travel



Pict courtesy of Dan Larremore

Naive 1: booking with airline

Naive 2: booking with kayak, expedia,…

See Didier et al. 2015, DeFord 2017, Taylor D 2017, Valles-Catala 2016 etc… for an empirical comparison of aggregation vs multilayer approaches

Community detection in multilayer networks: *approaches*

1. **Non-generative**: modularity maximization

   Mucha et al *Science* 2010
   Bazzi M et al. SIAM 2016

2. **Generative**:

   Peixoto, T. P. Phys. Rev. E 92, 042807–15 (2015) (both collapsed network and layered one)
   De Bacco, Power, Larremore , Moore. Phys. Rev. E 95, 1981–10 (2017).
   Schein A et al. ACM (2015) (bayesian)
   Stanley, Natalie, et al. IEEE transactions on network science and engineering 3.2 (2016): 95-105.(strata)

3. Spectral:

   Mercado P et al., arXiv:1803.00491 (2018) (Laplacian)
   De Domenico et al. , PRX (2015) (Infomap)
   DeFord and Pauls , arXiv:1703.05355 (2017)

Community detection in multilayer networks: *approaches*

1. **Non-generative**: modularity maximization
   Mucha et al *Science* 2010
   Bazzi M et al. SIAM 2016

2. **Generative**:
   Peixoto, T. P. Phys. Rev. E 92, 042807–15 (2015) (both collapsed network and layered one)
   De Bacco, Power, Larremore , Moore. Phys. Rev. E 95, 1981–10 (2017).
   Schein A et al. ACM (2015) (bayesian)
   Stanley, Natalie, et al. IEEE transactions on network science and engineering 3.2 (2016): 95-105.(strata)

3. Spectral:
   Mercado P et al., arXiv:1803.00491 (2018) (Laplacian)
   De Domenico et al. , PRX (2015) (Infomap)
   DeFord and Pauls , arXiv:1703.05355 (2017)

Community detection in multilayer networks: *preserving the multilayer structure: factor model approach*

$$u_i \cdot v_j = \sum_k u_{ik}\, v_{jk}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \qquad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

Layer 1 is the ‘evening’ layer.
User i prefers listening to
rock music in the evening

Layer 2 is the ‘morning’ layer.
User i prefers listening to jazz
in the morning

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \qquad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

In-Group 2 (attribute): rock
In-Group 1: jazz
Out-Group 2(preference):
rock
Out-Group K: jazz

Community detection in multilayer networks: *preserving the multilayer structure*

$$u_i \cdot v_j = \sum_k u_{ik}\, v_{jk}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \quad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

Users' preferences and songs' attributes are the same

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \quad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

The way they interact changes with layer

Community detection in multilayer networks: *preserving the multilayer structure*

$$u_i \cdot v_j = \sum_k u_{ik} \, v_{jk}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \quad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

Users' preferences and songs' attributes are the same

The way they interact changes with layer

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \quad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

Community detection in multilayer networks: *preserving the multilayer structure: collection of matrices —> tensor factorization*

$$u_i \cdot v_j = \sum_k u_{ik} \, v_{jk}$$

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \qquad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \qquad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

Community detection in multilayer networks: *preserving the multilayer structure: collection of matrices —> tensor factorization*

$$u_i \cdot v_j = \sum_k u_{ik}\, v_{jk}$$

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \qquad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

$$W^1 = \begin{bmatrix} 0.9 & \dots & 0.01 \\ & \dots & \\ 0.0 & \dots & 0.73 \end{bmatrix}$$

$$u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{ik} \end{bmatrix} \qquad \mathbf{v_j} = \begin{bmatrix} v_{i1} \\ v_{i2} \\ \vdots \\ v_{ik} \end{bmatrix}$$

$$W^2 = \begin{bmatrix} 0.02 & 0.8 \dots & 0.01 \\ & \dots & \\ 0.95 & \dots & 0.0 \end{bmatrix}$$

Mixed-membership generative model

$$A_{ij}^{\alpha} \sim Poi(M_{ij}^{\alpha})$$

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

Couple the layers

$$A_{ij}^{\alpha}, M_{ij}^{\alpha} \in \mathbb{M}_{N \times N}$$

u_{ik}, v_{ik} are the in/out membership vectors (overlapping communities)

ω^{α}_{kq} is related to the probability that, in layer α, there exists an edge from a node belonging to group k towards a node belonging to group q

De Bacco, Power, Larremore , Moore. Phys. Rev. E 95, 1981–10 (2017)

Community detection in multilayer networks: *preserving the multilayer structure*

$$A_{ij}^{\alpha} \sim Poi(M_{ij}^{\alpha})$$

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

Layer specific

Assortative-Disassortative structures

Couple the layers

$$A_{ij}^{\alpha}, M_{ij}^{\alpha} \in \mathbb{M}_{N \times N}$$

u_{ik}, v_{ik} are the in/out membership vectors (overlapping communities)

ω^{α}_{kq} is related to the probability that, in layer α, there exists an edge from a node belonging to group k towards a node belonging to group q

De Bacco, Power, Larremore , Moore. Phys. Rev. E 95, 1981–10 (2017)

Community detection in multilayer networks: *preserving the multilayer structure*

One of the possible hypothesis you can make, evaluate based on the application!

Durante et al. Jasa 2017
Ghasemian et al. PRX 2016
etc…

$$A_{ij}^{\alpha} \sim Poi(M_{ij}^{\alpha})$$

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

Layer specific

Assortative-Disassortative structures

Couple the layers

$$A_{ij}^{\alpha}, M_{ij}^{\alpha} \in \mathbb{M}_{N \times N}$$

u_{ik}, v_{ik} are the in/out membership vectors (overlapping communities)

ω^{α}_{kq} is related to the probability that, in layer α, there exists an edge from a node belonging to group k towards a node belonging to group q

De Bacco, Power, Larremore , Moore. Phys. Rev. E 95, 1981–10 (2017)

Community detection on network: *the approach*
In tensor notation

$$A^{\alpha} \approx U W^{\alpha} V^{T}$$

$$A \approx U V^{T}$$

K

$$U = \begin{bmatrix} & \cdots & \\ u_{i1} & \cdots & u_{iK} \\ & \cdots & \end{bmatrix}$$

N

N

$$V^{T} = \begin{bmatrix} v_{i1} \\ \cdots \\ v_{iK} \end{bmatrix}$$

K

K

$$W^{\alpha} = \begin{bmatrix} w^{\alpha}_{11} & \cdots & w^{\alpha}_{1K} \\ & w^{\alpha}_{kq} & \\ w^{\alpha}_{K1} & \cdots & w^{\alpha}_{KK} \end{bmatrix}$$

K

Mixed-membership generative model: *inference*

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

GM + parameters

stochastic draw →

$$A_{ij}^{\alpha} \sim Poi(M_{ij}^{\alpha})$$

Data

$$U = \begin{bmatrix} & \cdots & \\ u_{i1} & \cdots & u_{iK} \\ & & \\ & \cdots & \end{bmatrix}$$

$$V^T = \begin{bmatrix} v_{i1} \\ \cdots \\ v_{iK} \end{bmatrix}$$

$$W^{\alpha} = \begin{bmatrix} w_{11}^{\alpha} & \cdots & w_{1K}^{\alpha} \\ & w_{kq}^{\alpha} & \\ w_{K1}^{\alpha} & \cdots & w_{KK}^{\alpha} \end{bmatrix}$$

→

Mixed-membership generative model: *inference*

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

GM + parameters

stochastic draw →

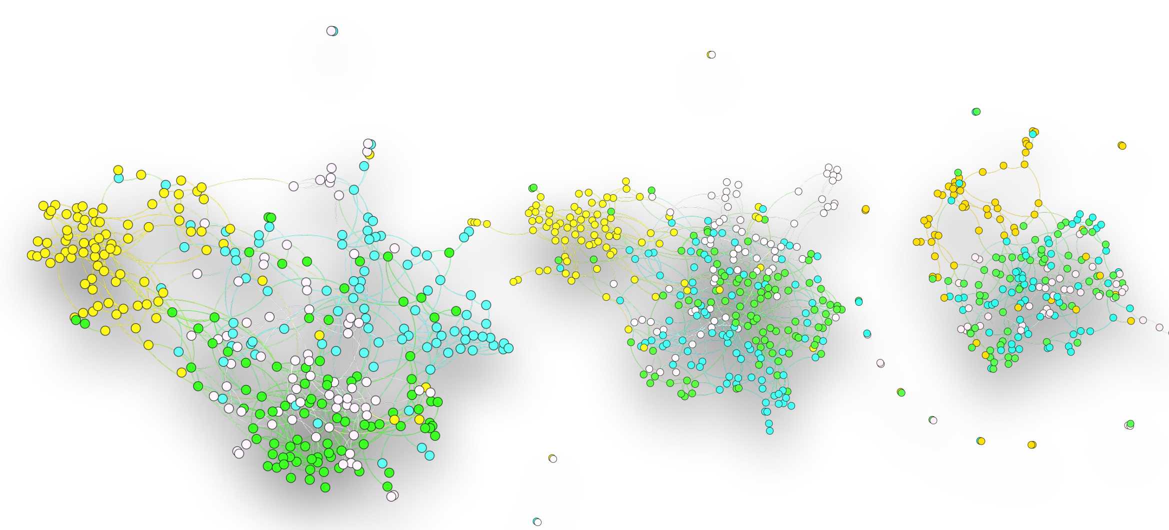
$$A_{ij}^{\alpha} \sim Poi(M_{ij}^{\alpha})$$

Data

$$U = \begin{bmatrix} & \cdots & \\ u_{i1} & \cdots & u_{iK} \\ & & \\ & \cdots & \end{bmatrix}$$

$$V^T = \begin{bmatrix} v_{i1} \\ \cdots \\ v_{iK} \end{bmatrix}$$

$$W^{\alpha} = \begin{bmatrix} w_{11}^{\alpha} & \cdots & w_{1K}^{\alpha} \\ & w_{kq}^{\alpha} & \\ w_{K1}^{\alpha} & \cdots & w_{KK}^{\alpha} \end{bmatrix}$$



Mixed-membership generative model: *inference*

$$M_{ij}^{\alpha} = \sum_{k,q} u_{ik} v_{jk} w_{kq}^{\alpha}$$

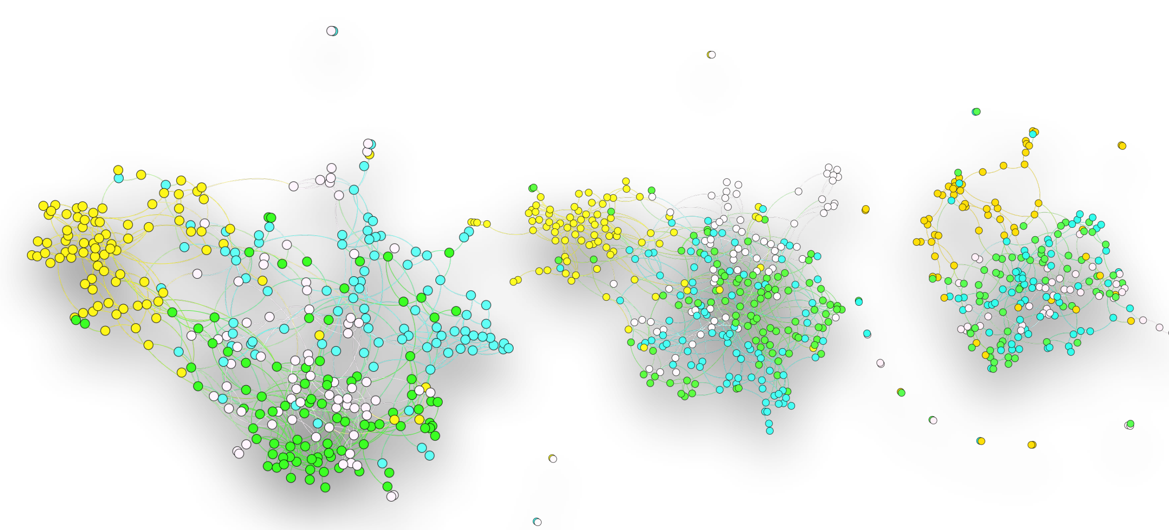
$$A_{ij}^{\alpha} \sim Poi(M_{ij}^{\alpha})$$

GM + parameters ← inference — Data

?

?

?



Mixed-membership generative model: *inference*

$$P(\Theta|A) \propto P(A|\Theta)P(\Theta)$$

Bayes' rule

Mixed-membership generative model: *inference*

$$P(\Theta|A) \propto P(A|\Theta)P(\Theta)$$

Bayes' rule

$$P(A|\Theta) = \prod_{\alpha=1}^{L} \prod_{i,j=1}^{N} \frac{e^{-M_{ij}^{\alpha}} \, {M_{ij}^{\alpha}}^{A_{ij}^{\alpha}}}{A_{ij}^{\alpha}!}$$

Likelihood

$$P(\Theta)$$

Uniform (MLE)
Gamma (conjugate with Poisson) (MAP)
…

Prior

Mixed-membership generative model: *inference*

$$P(\Theta|A) \propto P(A|\Theta)P(\Theta)$$ Bayes' rule

$$P(A|\Theta) = \prod_{\alpha=1}^{L} \prod_{i,j=1}^{N} \frac{e^{-M_{ij}^{\alpha}} \; {M_{ij}^{\alpha}}^{A_{ij}^{\alpha}}}{A_{ij}^{\alpha}!}$$ Likelihood

$$P(\Theta)$$

Uniform (MLE)
Gamma (conjugate with Poisson) (MAP)
…

Prior

- Gibbs sampling
- Expectation Maximization
- Variational inference
- Gradient descent

- In multilayer networks, latent variables can be **shared across layers**

- Depending on the specific application: change/inform with domain knowledge the **details** of the model (e.g. how layers interact, what variable is shared)

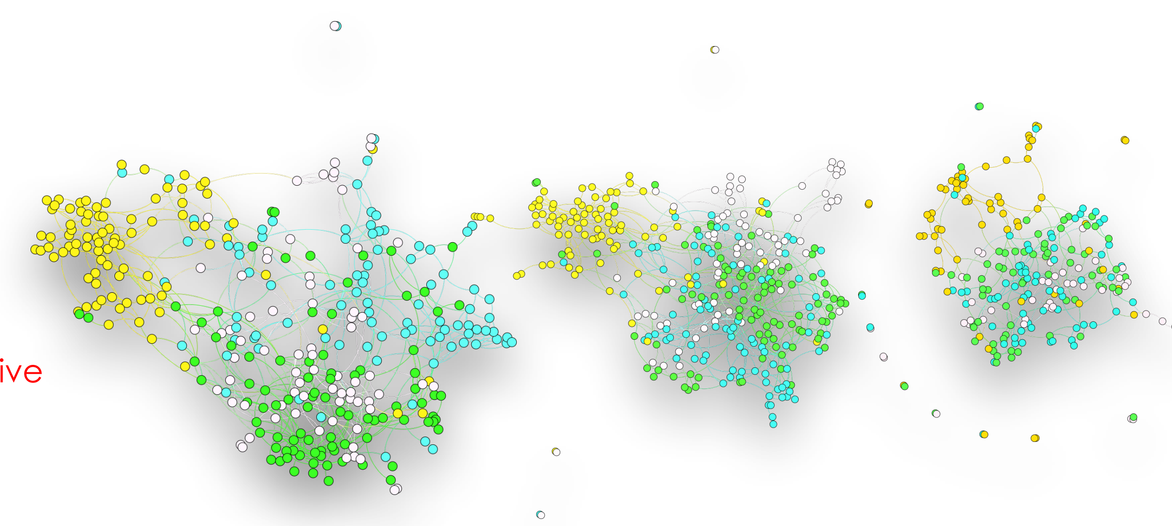
$$A^{\alpha}_{ij} \sim Poi(M^{\alpha}_{ij})$$

$$M^{\alpha}_{ij} = \sum_{k,q} u_{ik} v_{jk} w^{\alpha}_{kq}$$

Layer specific

Assortative-Disassortative structures

Couple the layers

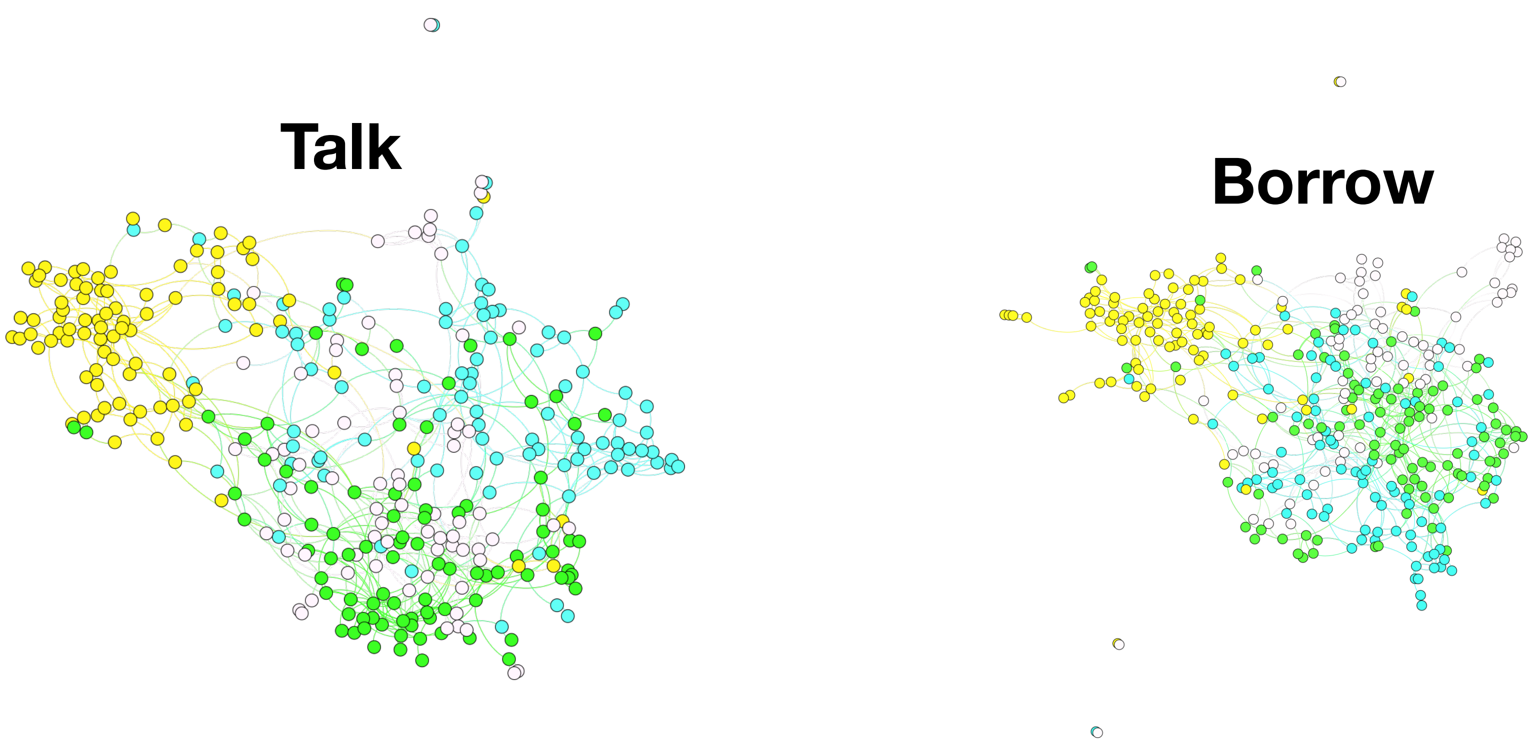


1. The problem (10mins)
    - Definition and motivation
    - Example application
2. The approach (15 mins)
    - Generative models
3. **Advanced topics: Multilayer networks (20mins)**
    - Mixed-membership factor models
    - Layer interdependence (if time allows)

The layer interdependence problem

Hypothesis: explained by common community structure…

…even if the layers seem very different

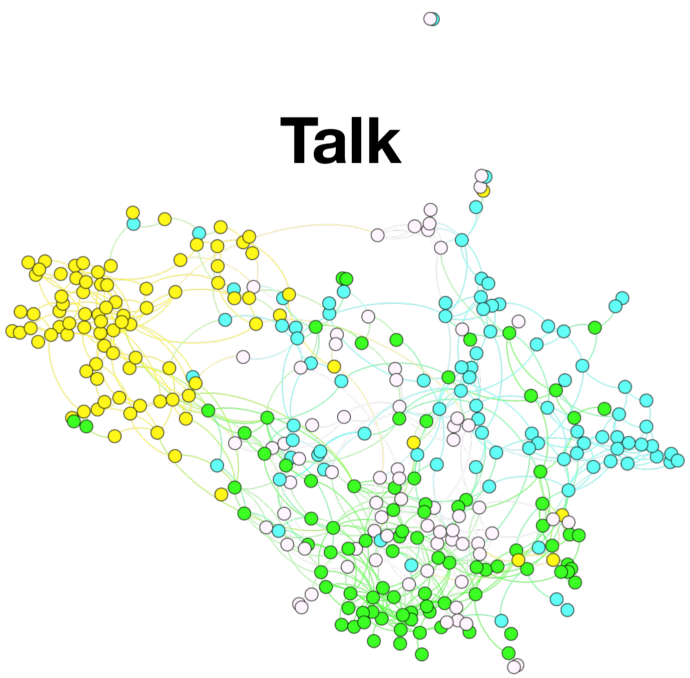


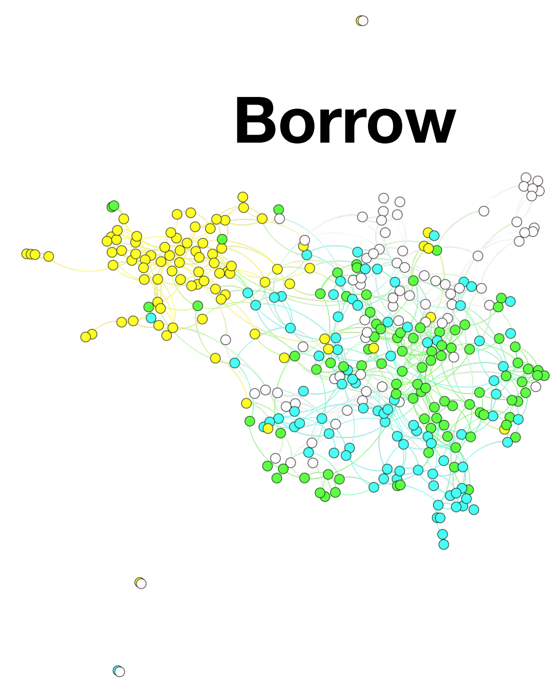
Don't ask whether two layers are correlated: ask whether knowing one helps predict the other

C De Bacco, E Power, D B Larremore , C Moore. Phys. Rev. E 95, 1981–10 (2017).

The layer interdependence problem

Don't ask whether two layers are correlated: ask whether knowing one helps predict the other



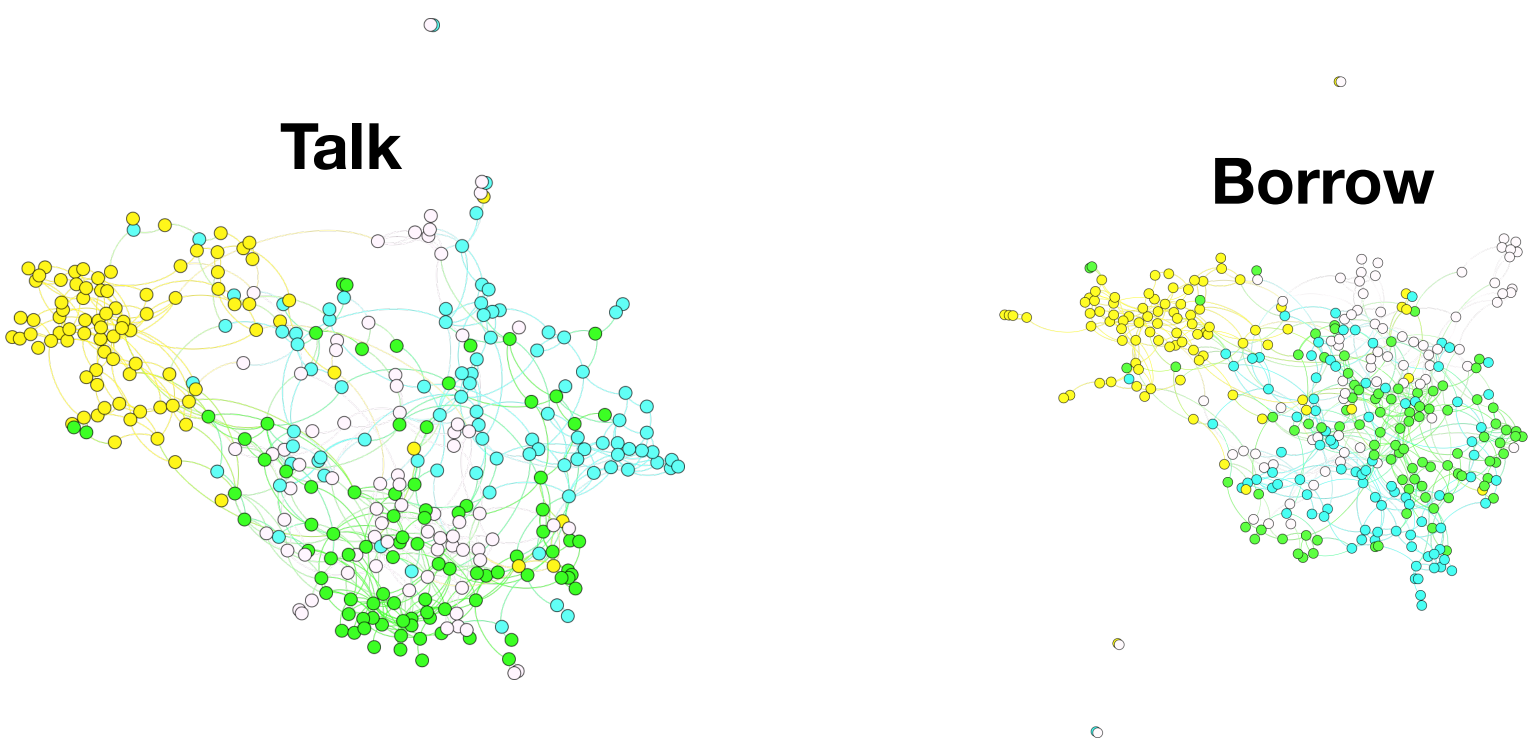


Layers are redundant if they reveal same latent features of the nodes: don't need to ask about both

But knowing one layer may make it harder to predict another, if their structures are inconsistent

The layer interdependence problem

Don’t ask whether two layers are correlated: ask whether knowing one helps predict the other

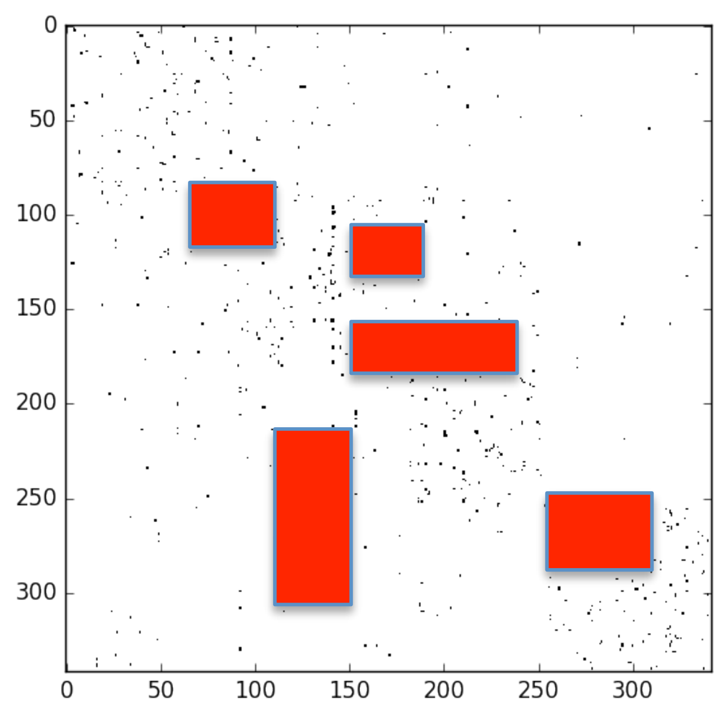


Layers are redundant if they reveal same latent features of the nodes: don’t need to ask about both

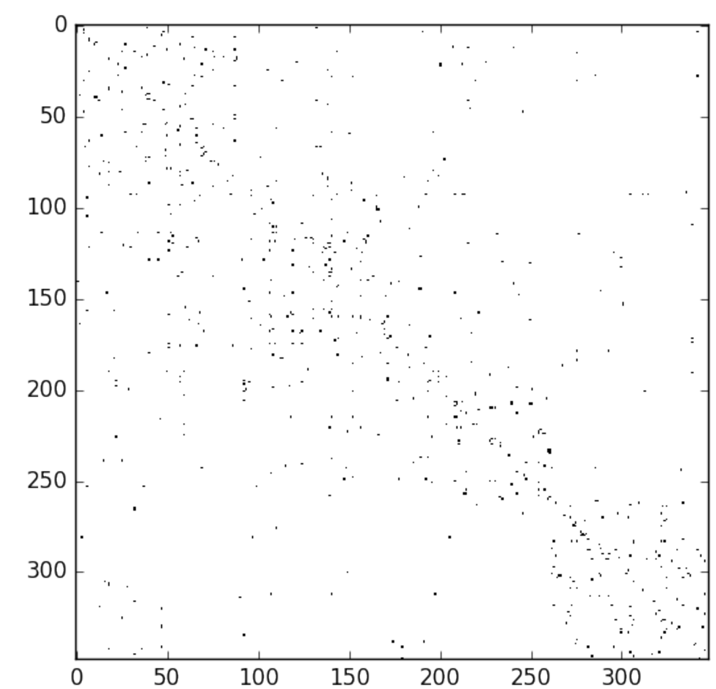
But knowing one layer may make it harder to predict another, if their structures are inconsistent

The layer interdependence problem: *in matrix form*

Does knowing matrix ‘Borrow’ help me fill the red hidden entries in ‘Talk’?



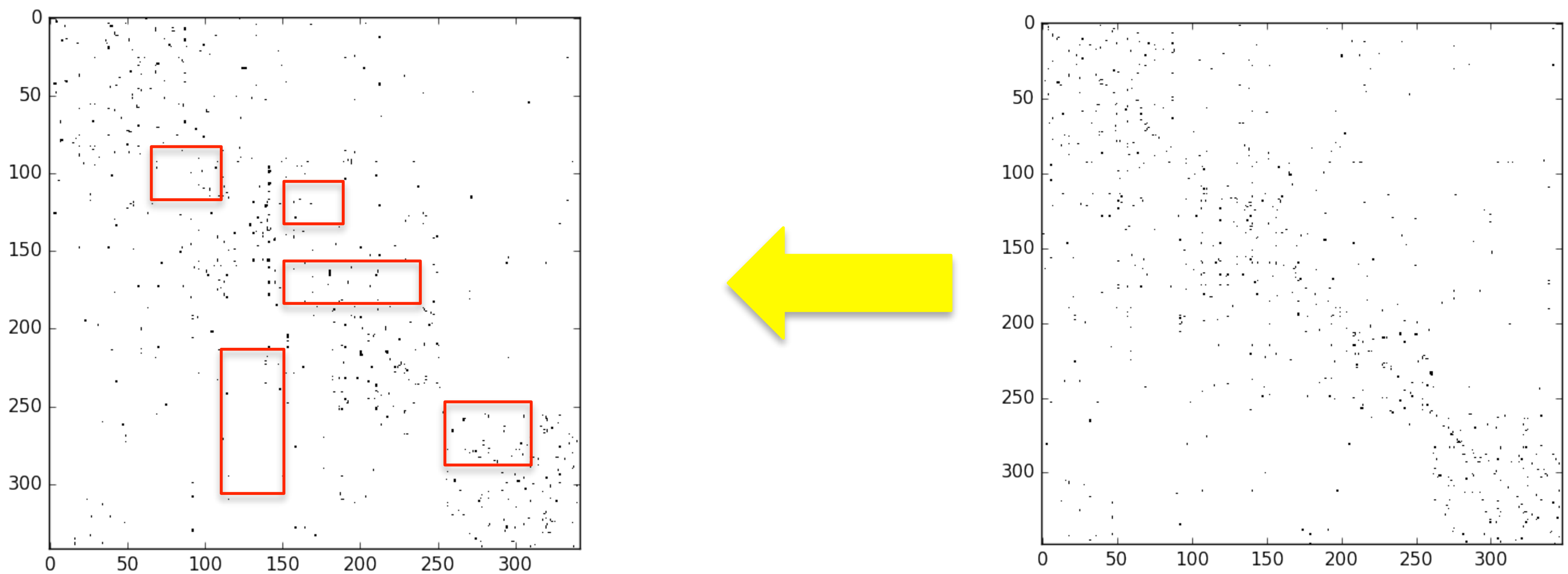
Talk



Borrow

The layer interdependence problem: *in matrix form*

Does knowing matrix ‘Borrow’ help me fill the red hidden entries in ‘Talk’?

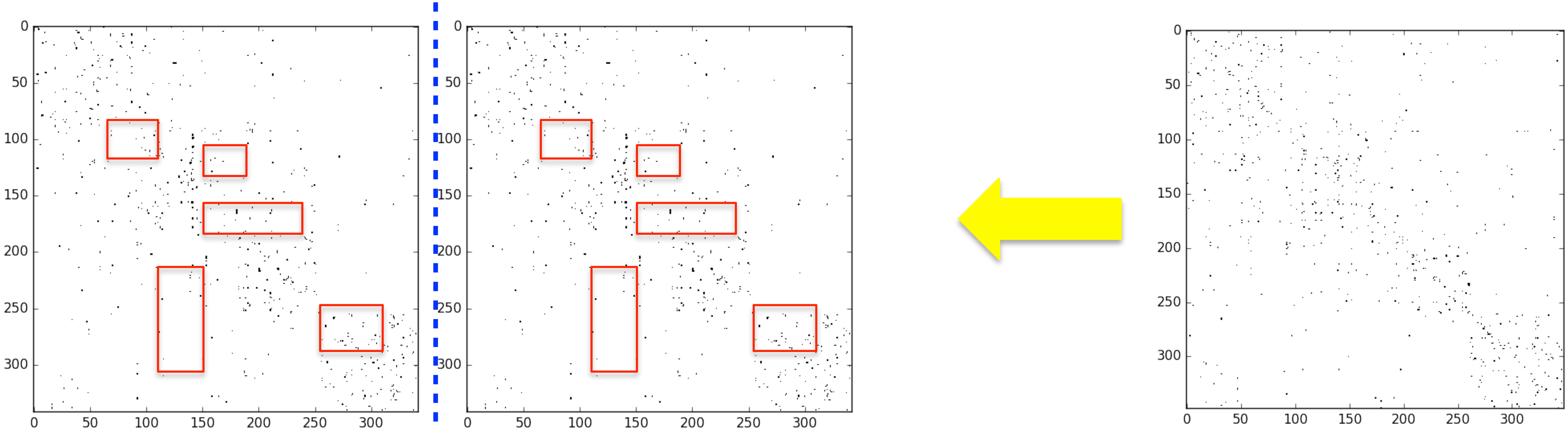


Talk

Borrow

The layer interdependence problem: *in matrix form*

Does knowing matrix ‘Borrow’ help me fill the red hidden entries in ‘Talk’?



True values

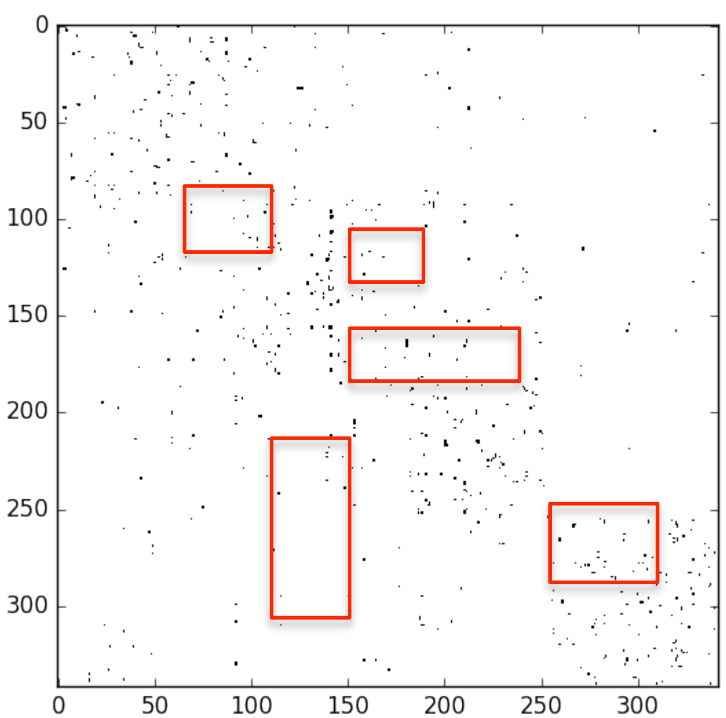
Talk

Borrow

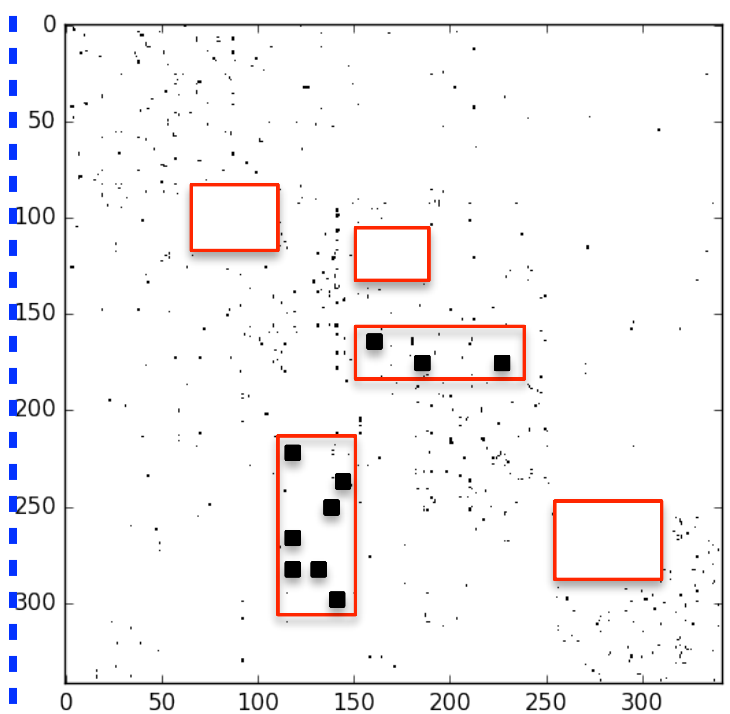
Borrow HELPS predicting Talk

The layer interdependence problem: *in matrix form*

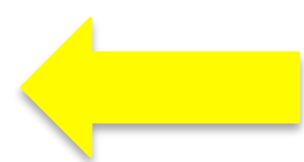
Does knowing matrix ‘Borrow’ help me fill the red hidden entries in ‘Talk’?

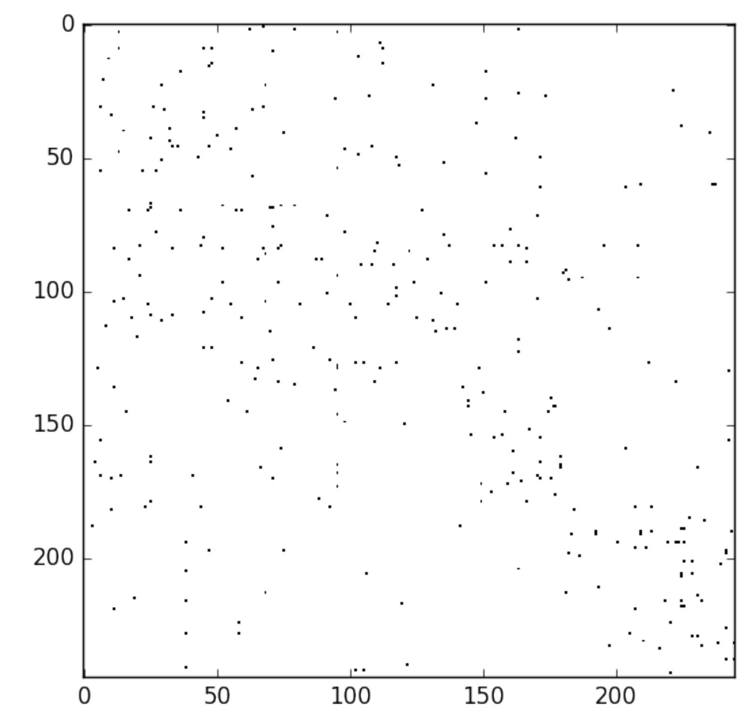


True values



Talk





Borrow

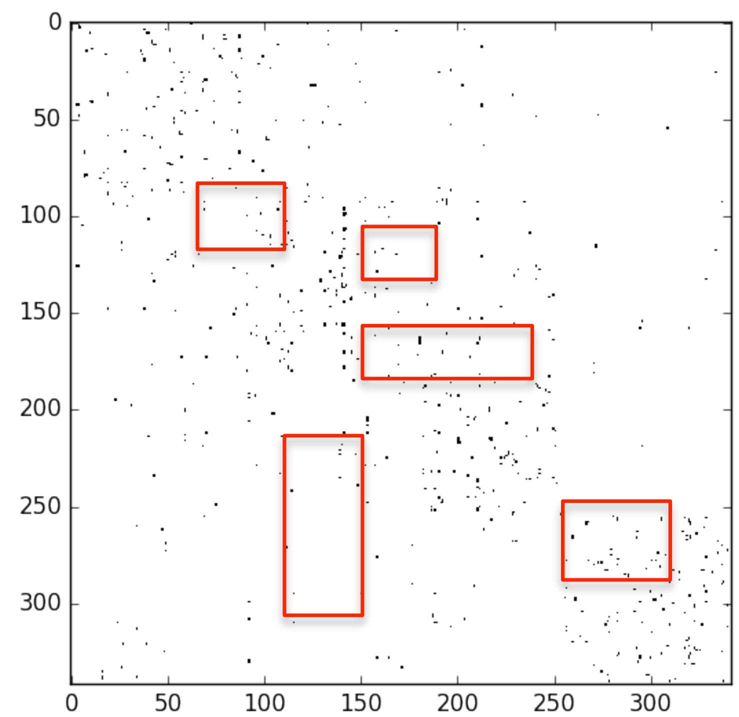
Borrow HURTS predicting Talk

The layer interdependence problem: *algorithm*

Idea

- Hide 20% of the adjacency matrix's entries from 1 test layer;
- Fit the model on the remaining 80% entries on the test layer
- Calculate AUC (measure of prediction performance): **L1**

No information from other layers is used for now

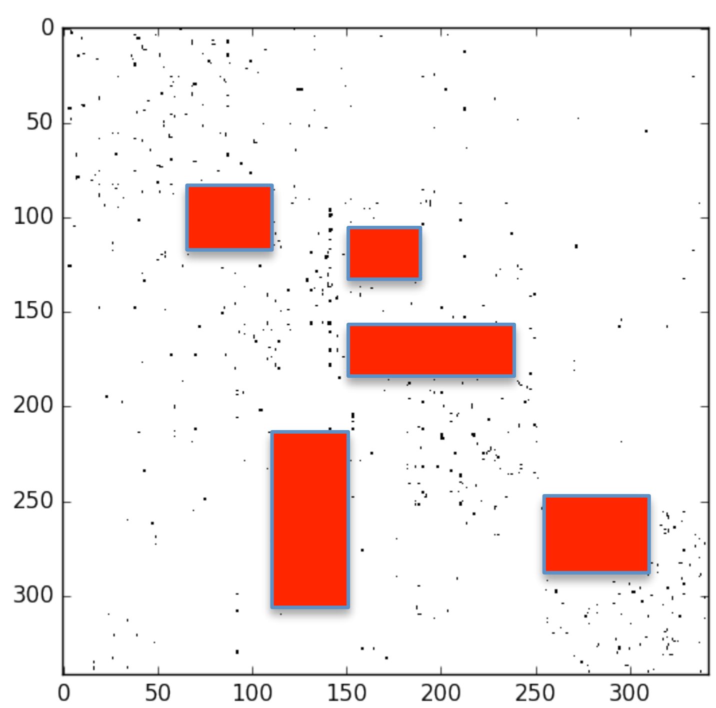


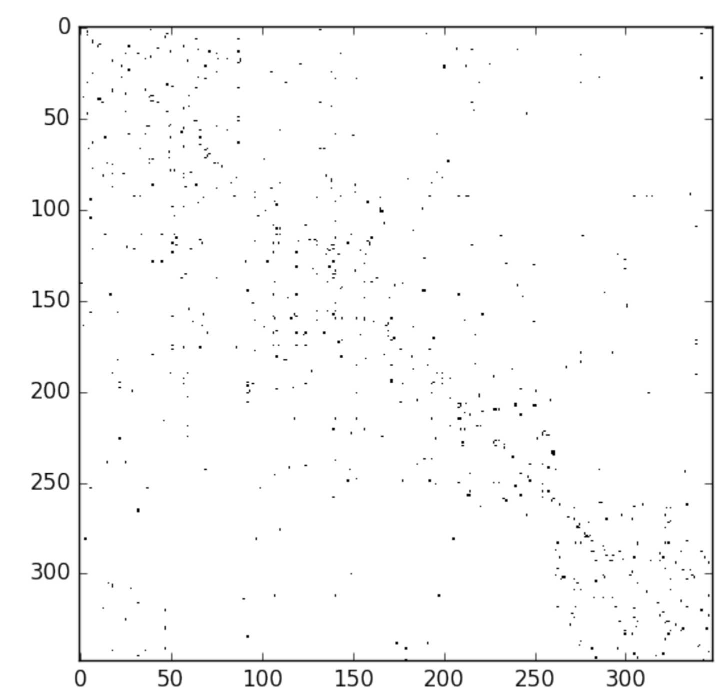
The layer interdependence problem: *algorithm*

Idea

- Hide 20% of the adjacency matrix's entries from 1 test layer;
- Fit the model on the remaining 80% entries on the test layer + 100% of the entries from 1 other layer
- Calculate AUC (measure of prediction performance)
- Select the 2nd layer that helps the most (higher AUC): **L2**

Information from 1 other layers is used, i.e. we use 2 layers in total (but only 80% from the test one)



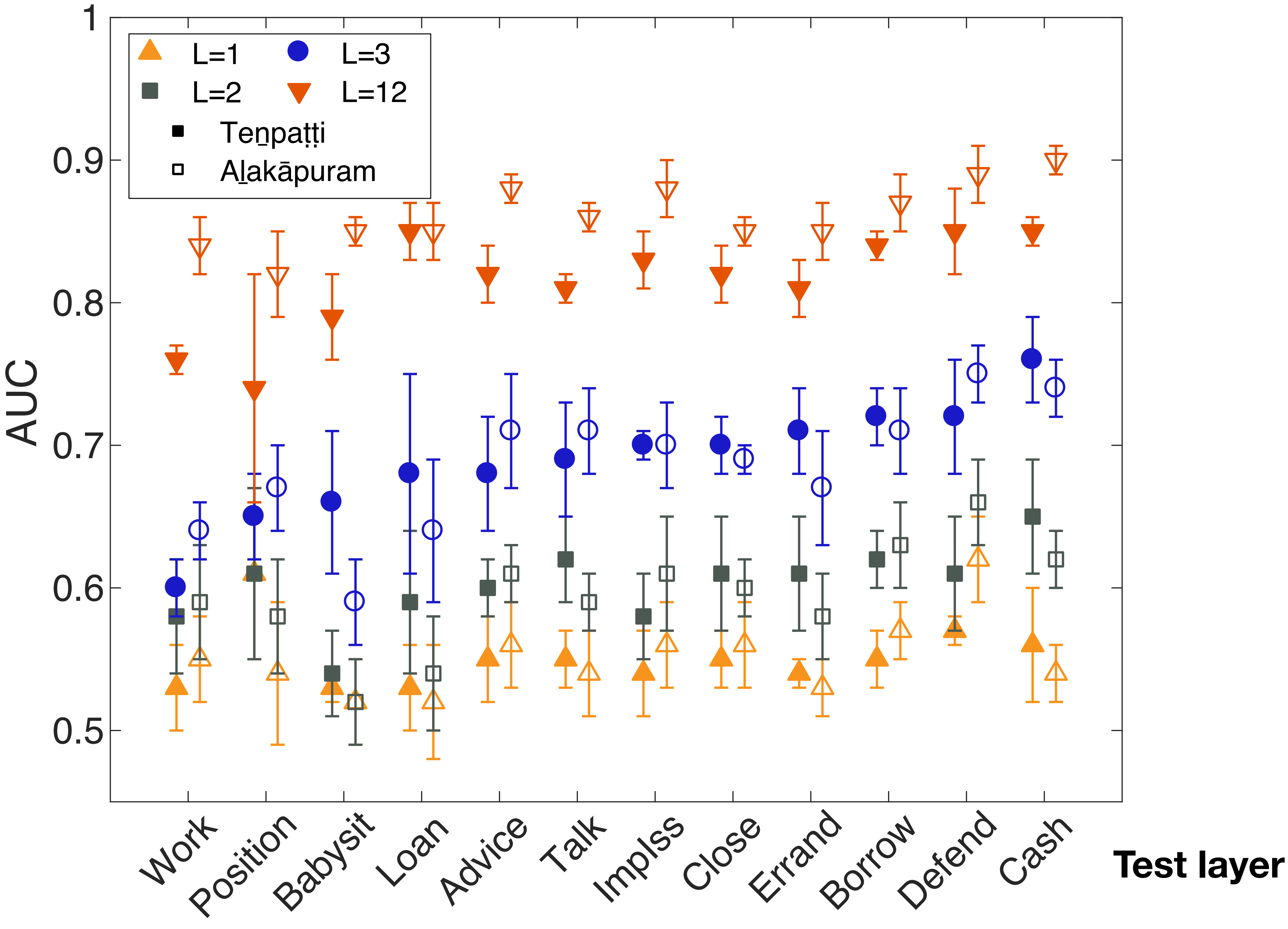


The layer interdependence problem: *algorithm*

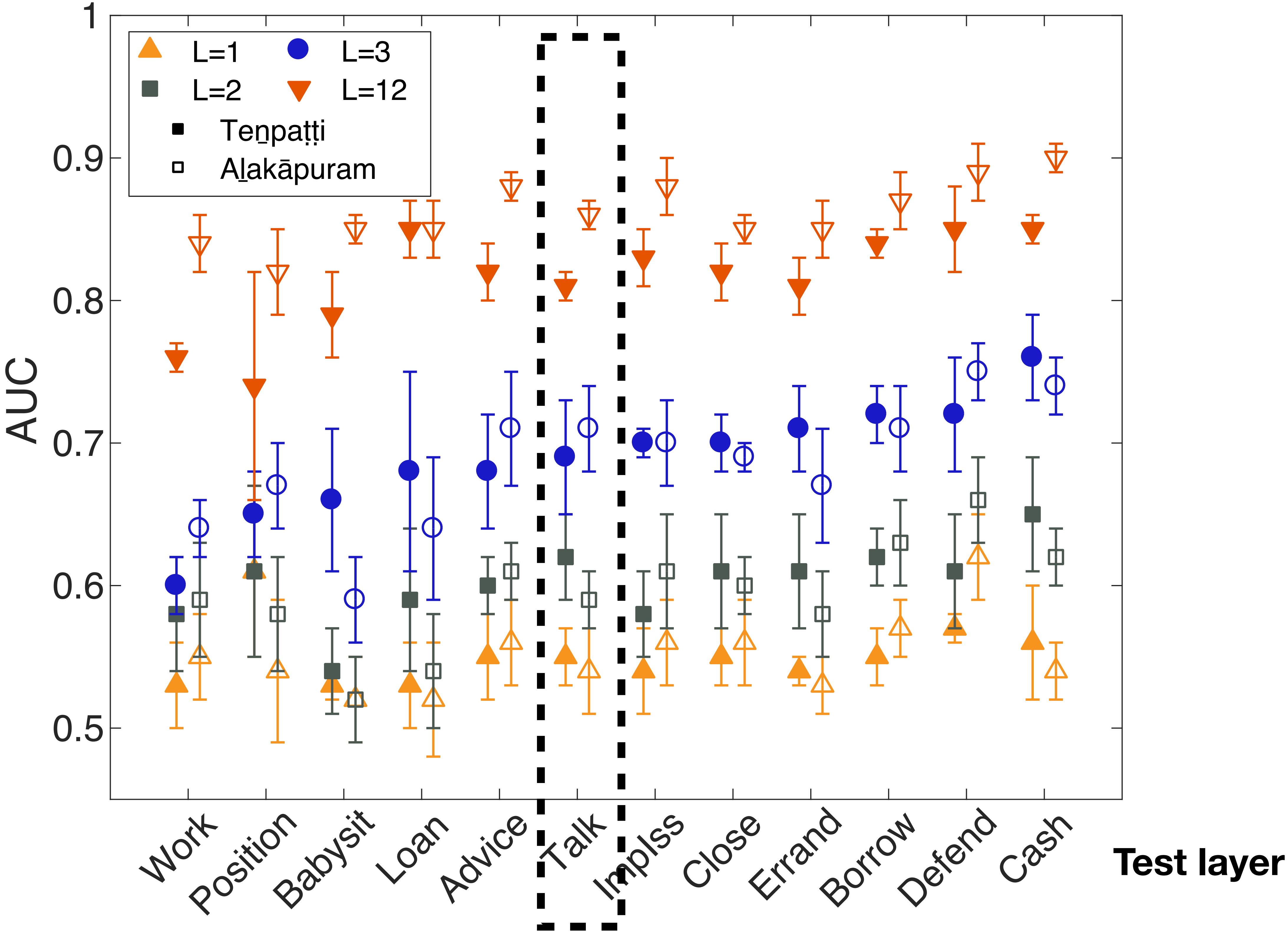
Idea

- Repeat up to using all other 11 layers… **L12**

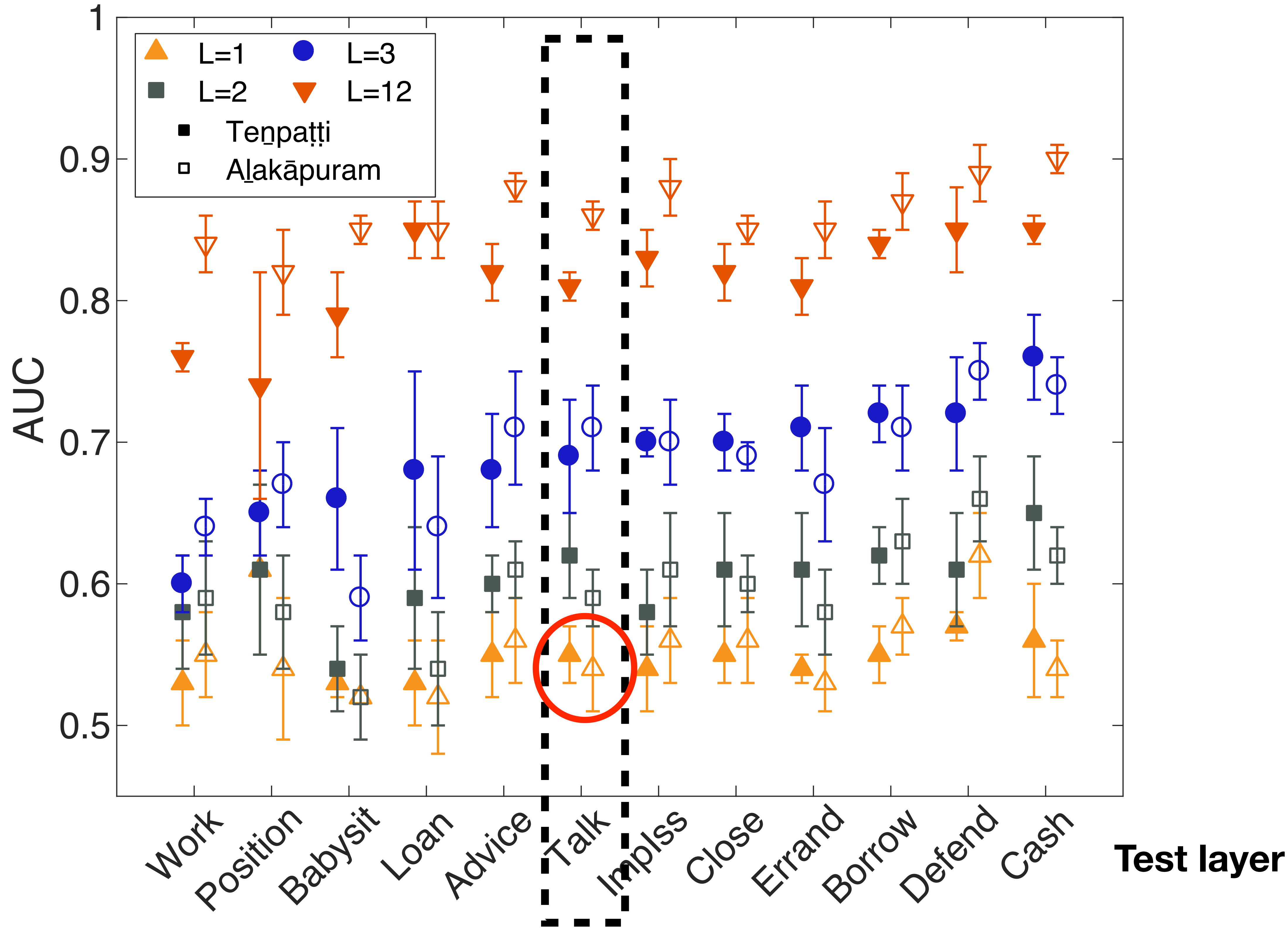
The layer interdependence problem: *results*



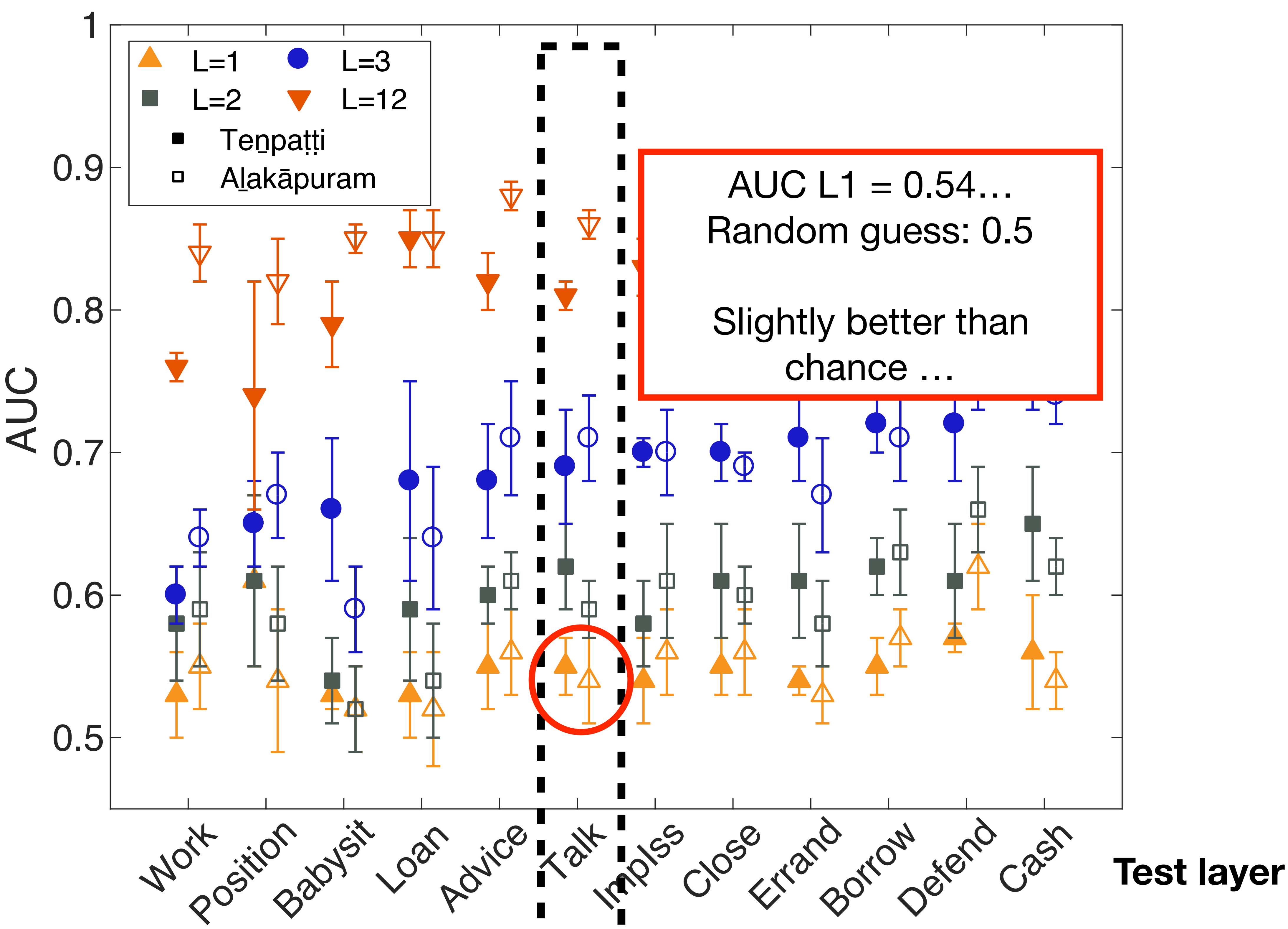
The layer interdependence problem: *results*



The layer interdependence problem: *results*



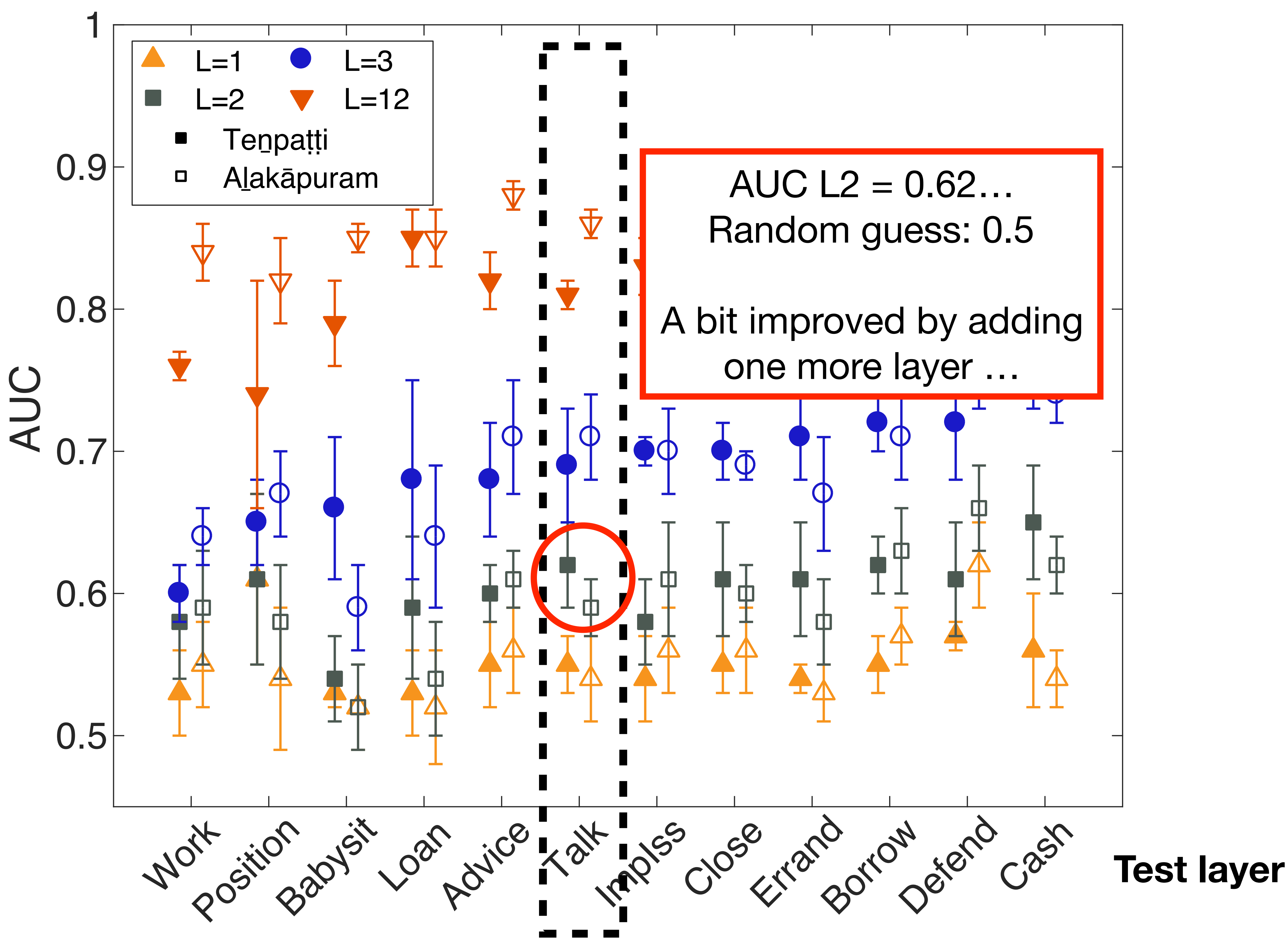
The layer interdependence problem: *results*



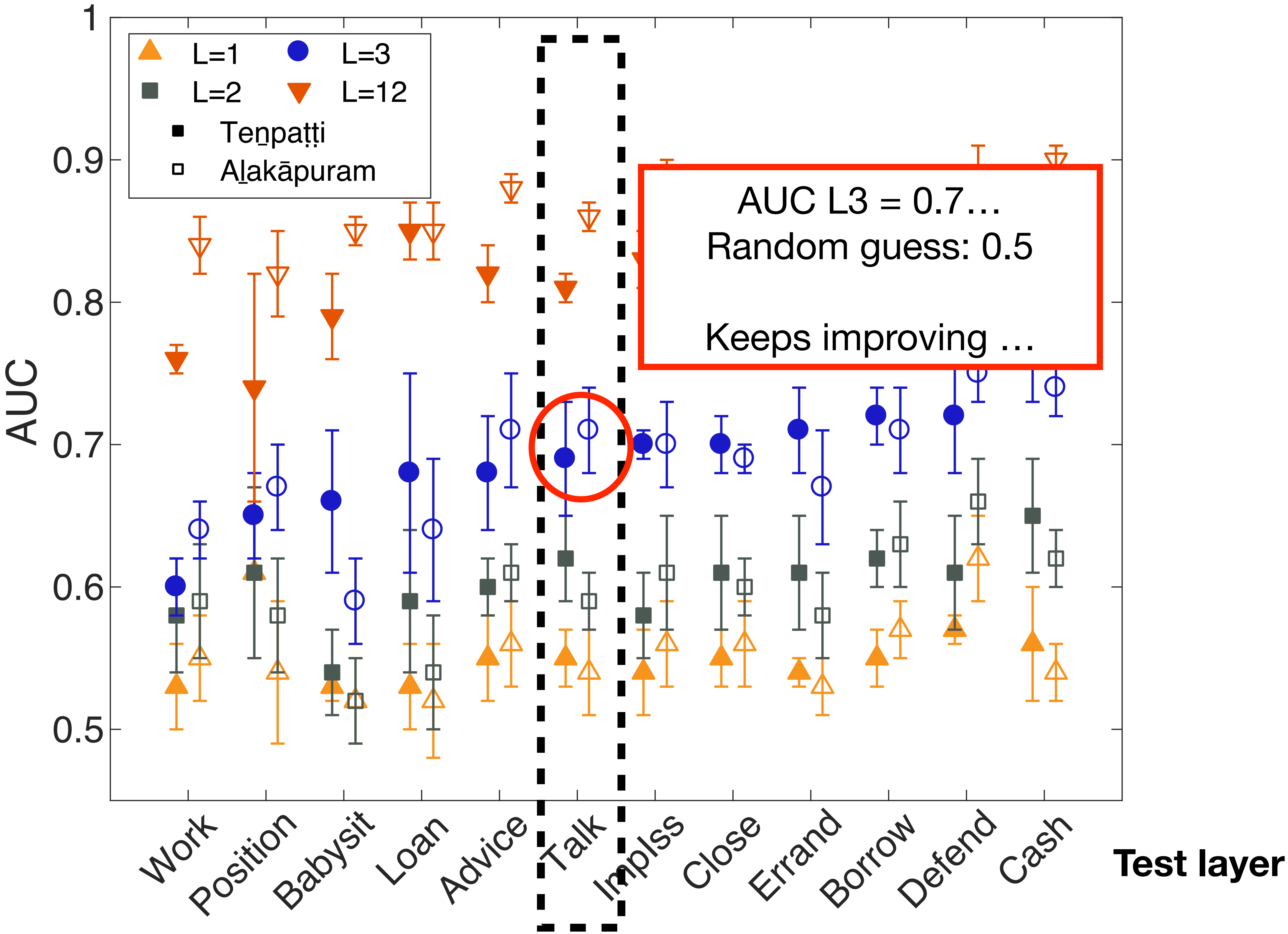
AUC L1 = 0.54…
Random guess: 0.5

Slightly better than chance …

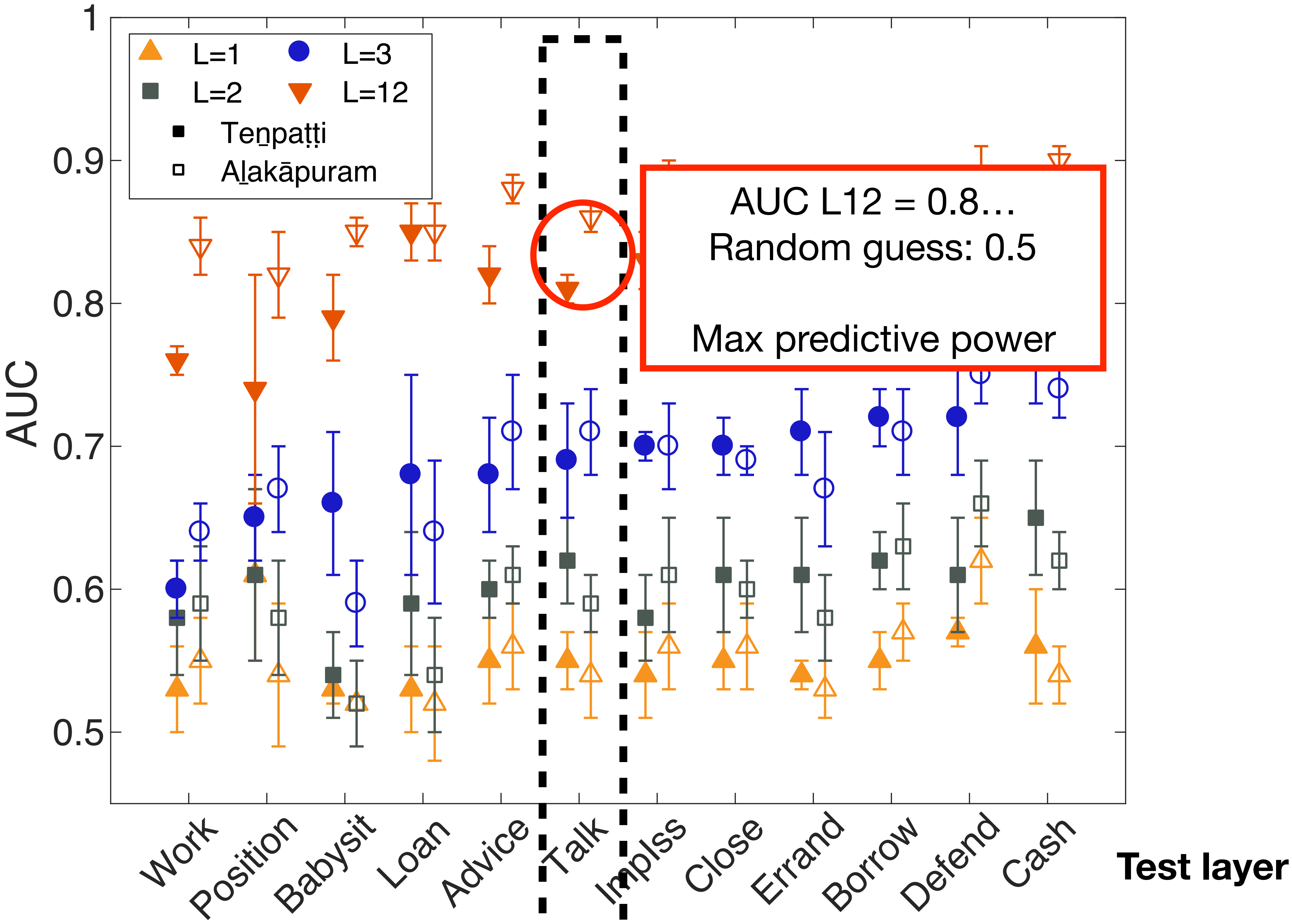
The layer interdependence problem: *results*



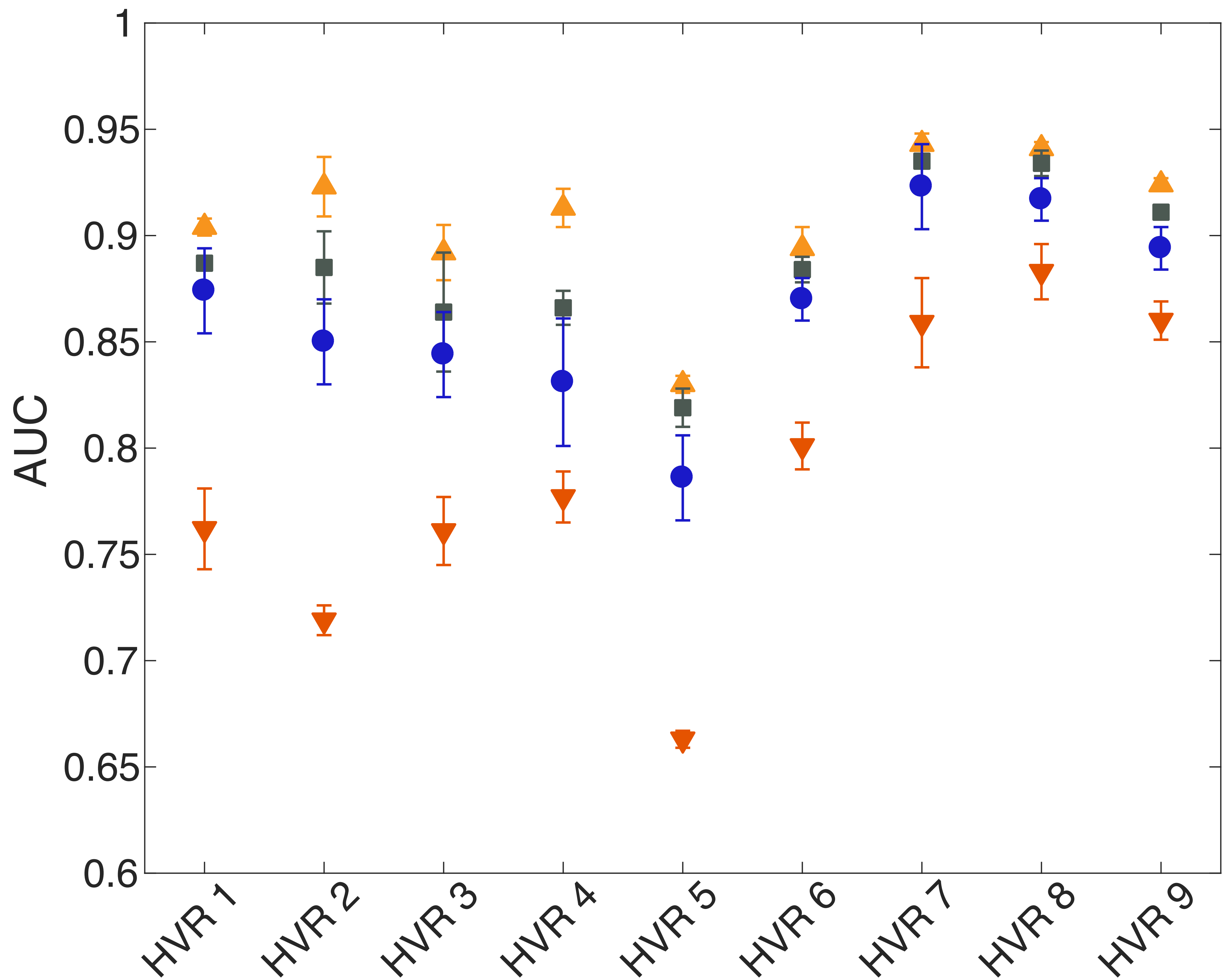
The layer interdependence problem: *results*



The layer interdependence problem: *results*

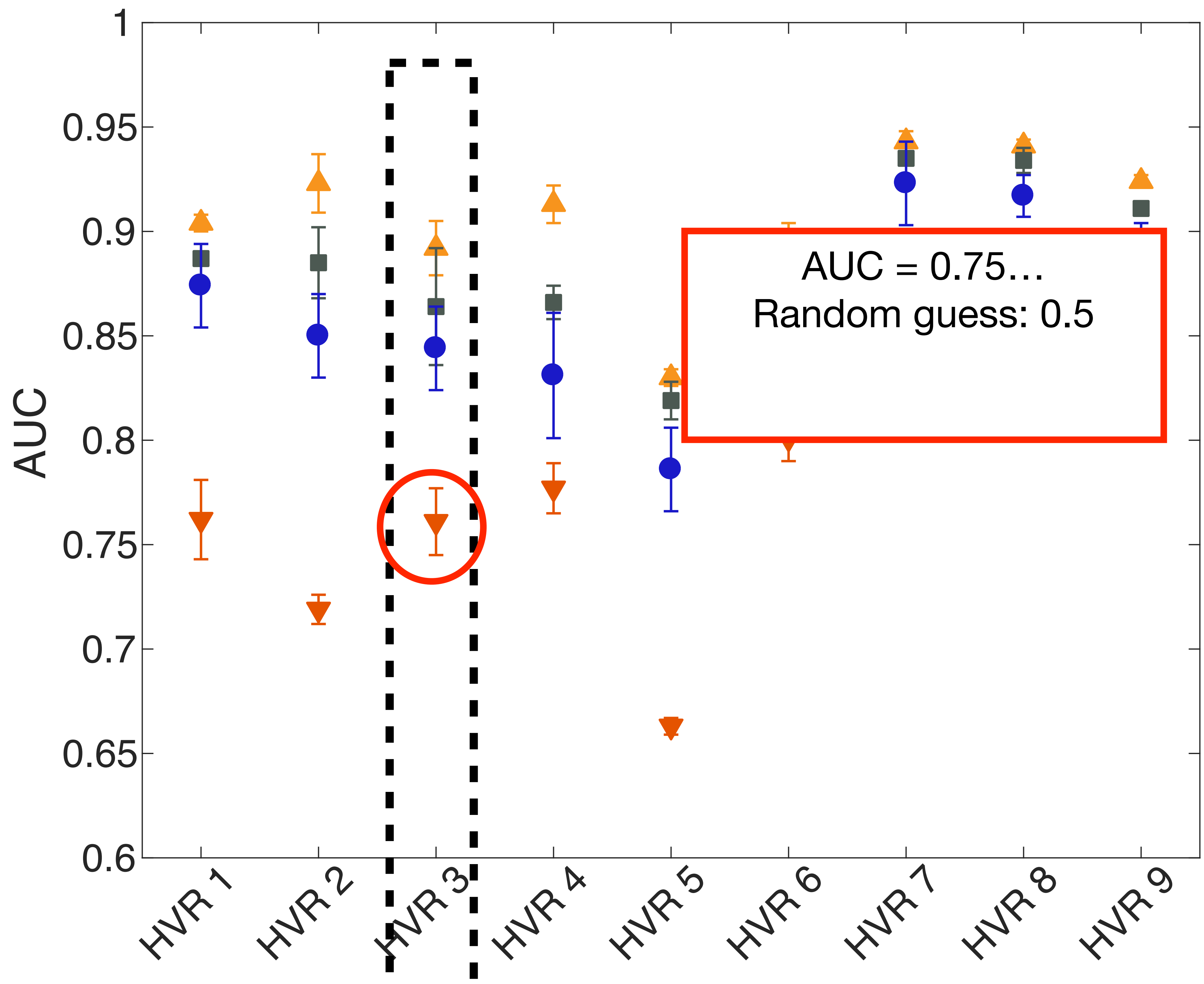


The layer interdependence problem: *malaria genetic network*



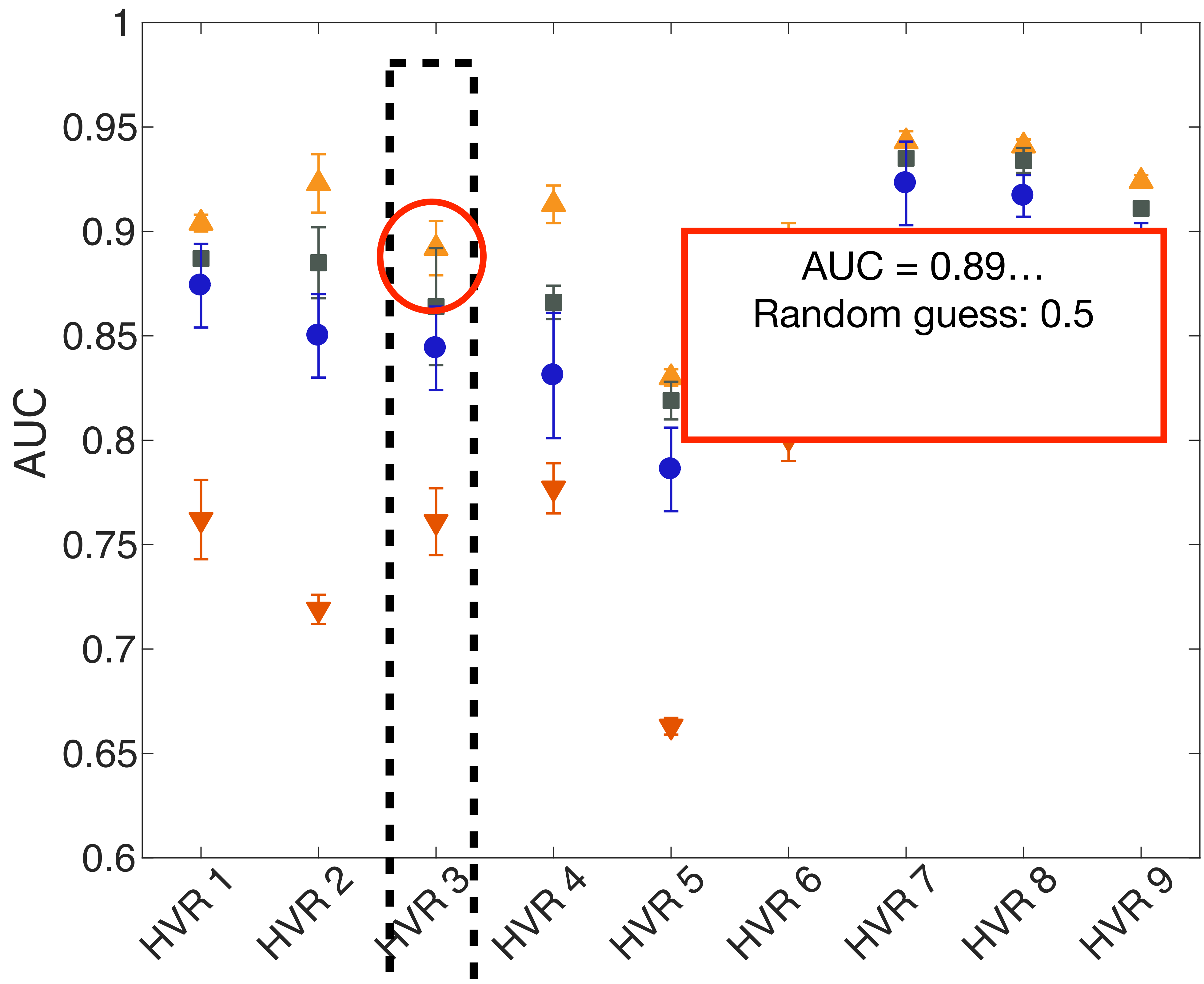
For explanation about dataset and network construction: D B Larremore et al, “A network approach to analyzing highly recombinant malaria parasite genes”, PLoS Computational Biology 9(10), e1003268 (2013)

The layer interdependence problem: *malaria genetic network*



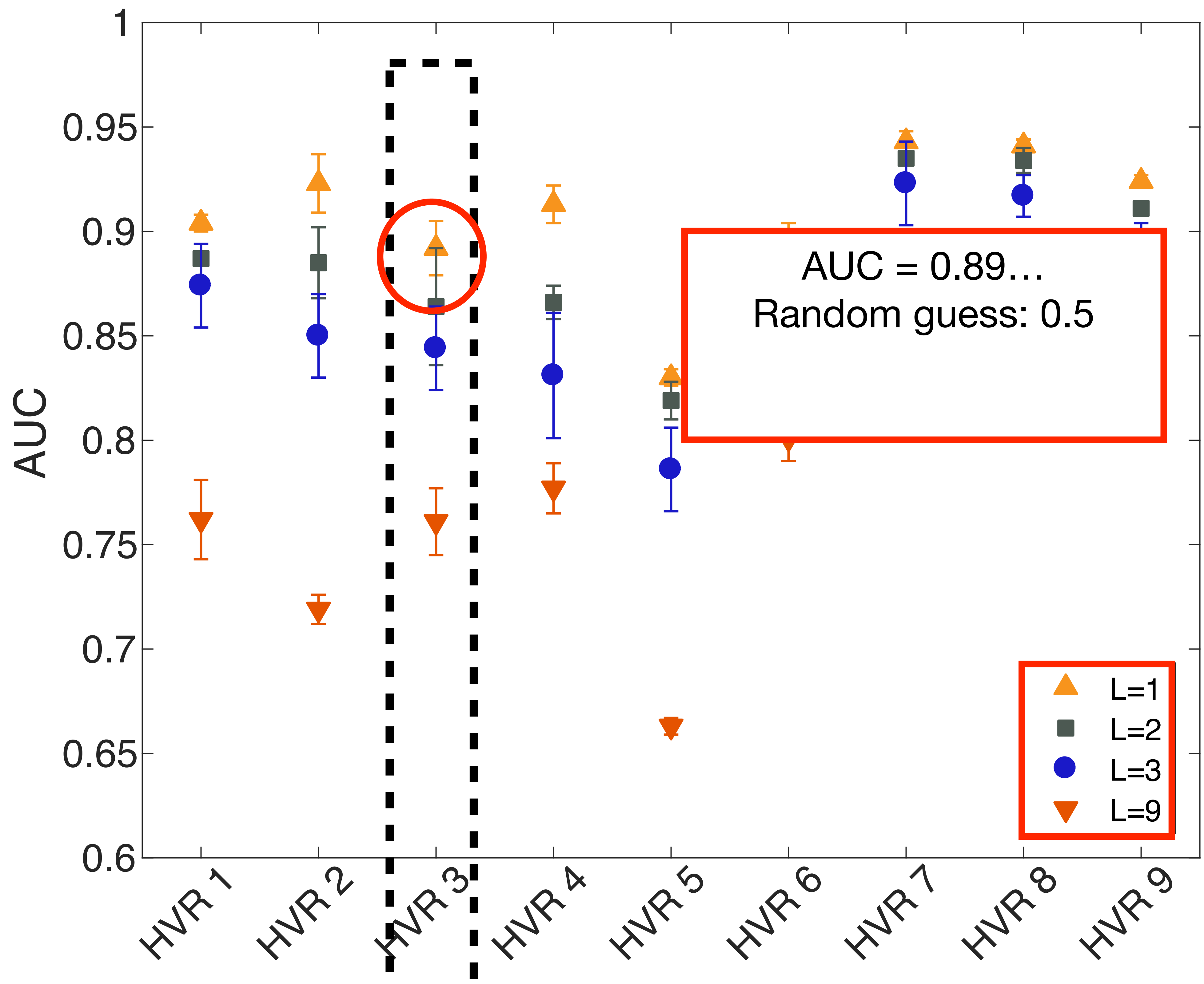
For explanation about dataset and network construction: D B Larremore et al, "A network approach to analyzing highly recombinant malaria parasite genes", PLoS Computational Biology 9(10), e1003268 (2013)

The layer interdependence problem: *malaria genetic network*



For explanation about dataset and network construction: D B Larremore et al, “A network approach to analyzing highly recombinant malaria parasite genes”, PLoS Computational Biology 9(10), e1003268 (2013)

The layer interdependence problem: *malaria genetic network*



For explanation about dataset and network construction: D B Larremore et al, "A network approach to analyzing highly recombinant malaria parasite genes", PLoS Computational Biology 9(10), e1003268 (2013)

The layer interdependence problem

- Look for **shared** information across layers
- Tensor factorization allows to model that
- Look for predictive power instead of correlations
- Many different questions that one can ask. E.g. do the reverse: *remove* hurting layers instead of *adding* helpful ones

Lecture 1: *Conclusions*

- The problem and example applications
- Generative models and latent variable formalism
- Advanced topic: Multilayer networks
- A hint about inference
- *Layer interdependence problem*

caterina.debacco@gmail.com https://github.com/cdebacco

Positions opening coming soon

- Postdoc or PhD
- Max Planck Institute for Intelligent Systems, Tubingen, Germany
- Starting date flexible, from Jan 2019 onwards
- Contact me if you are interested in working on inference and optimization problems with statistical physics, probabilistic modeling and interdisciplinary applications
- Website with more details coming soon …

- Positions openings in other groups at https://cyber-valley.de/en

caterina.debacco@gmail.com https://github.com/cdebacco