Mesoscale reconstruction of images and networks using tensor decomposition

Hanbaek Lyu

Department of Mathematics

University of Wisconsin – Madison

Partially supported by NSF DMS #2206296 and #2010035

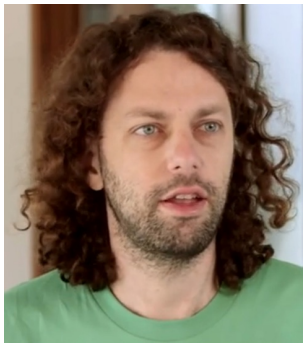
Multi-Modal Imaging with Deep Learning and Modeling @ IPAM

Nov. 28, 2022

Collaborators

Deanna Needell

Laura Balzano



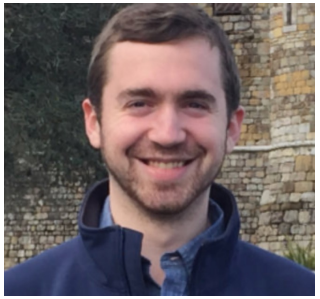
Mason A. Porter

Liza Rebrova

Alona Kryshchenko

Joshua Vendrow

Yacoub Kureh



Christopher Strohmeier

Ahmet Alacaoglu

Lara Kassab

Denali Molitor

Introduction

Key concept:

Low-rank Mesoscale Reconstruction

For Images/Networks

Image Reconstruction Problem

- Given an image $\boldsymbol{A}$, a reference image $\boldsymbol{B}$, and a scale parameter k:
- Can we reconstruct a best approximation $\widehat{\boldsymbol{A}}$ of $\boldsymbol{A}$ that resembles $\boldsymbol{B}$ at $k \times k$ scale?

Observed image $\boldsymbol{A}$

Reference image $\boldsymbol{B}$

Image Reconstruction Problem

- Given an image $\boldsymbol{A}$, a reference image $\boldsymbol{B}$, and a scale parameter k:
- Can we reconstruct a best approximation $\widehat{\boldsymbol{A}}$ of $\boldsymbol{A}$ that resembles $\boldsymbol{B}$ at $k \times k$ scale?

Observed image $\boldsymbol{A}$

Reference image $\boldsymbol{B}$

Reconstructed image $\widehat{\boldsymbol{A}}$

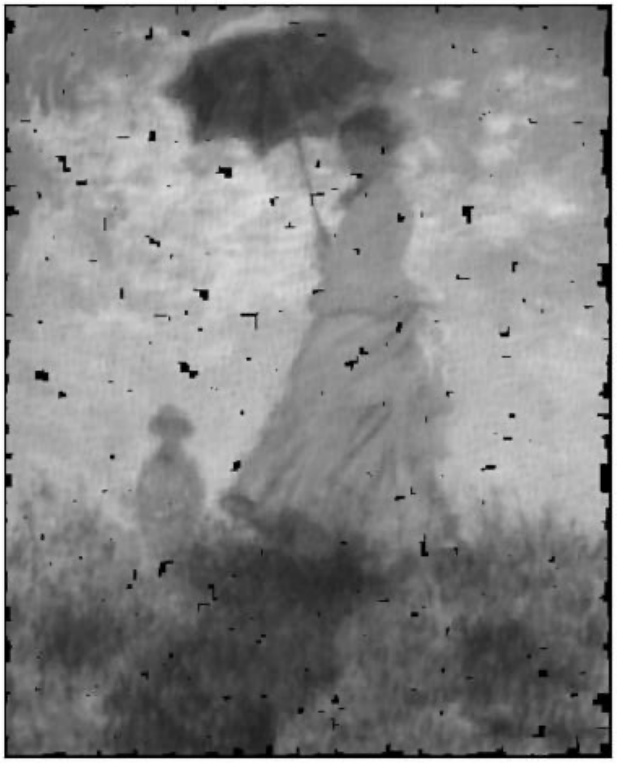
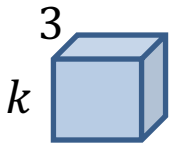
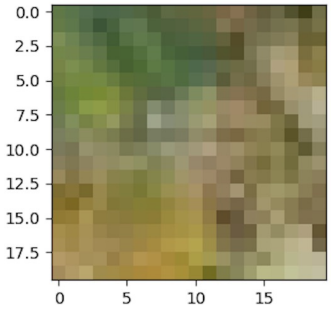


Image Reconstruction at mesoscale



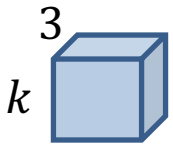
Observed



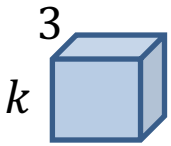
$\approx$

$A_{\mathbf{x}}$: sampled sub-img

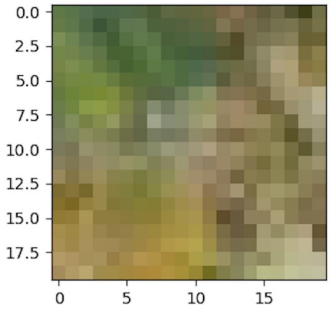
Image Reconstruction at mesoscale



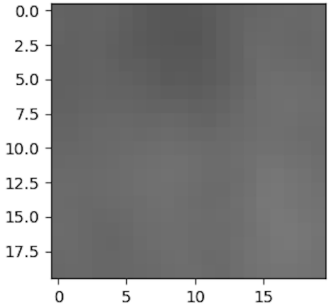
Observed



Reconstructed



$\approx$

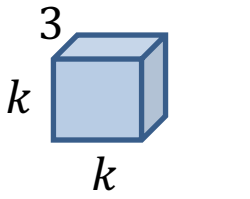


$\mathbf{A_x}$: sampled sub-img

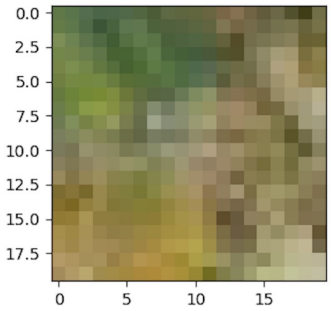
$\hat{A}_{\mathbf{x}}$: reconstructed sub-img

Image Reconstruction at mesoscale

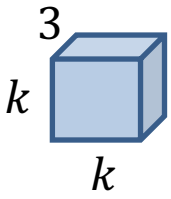
Reference image $\boldsymbol{B}$



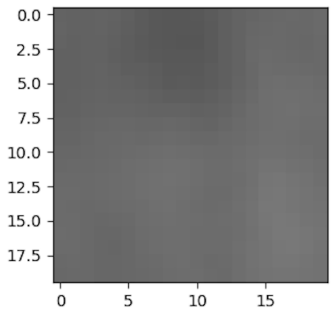
Observed



$\approx$



Reconstructed



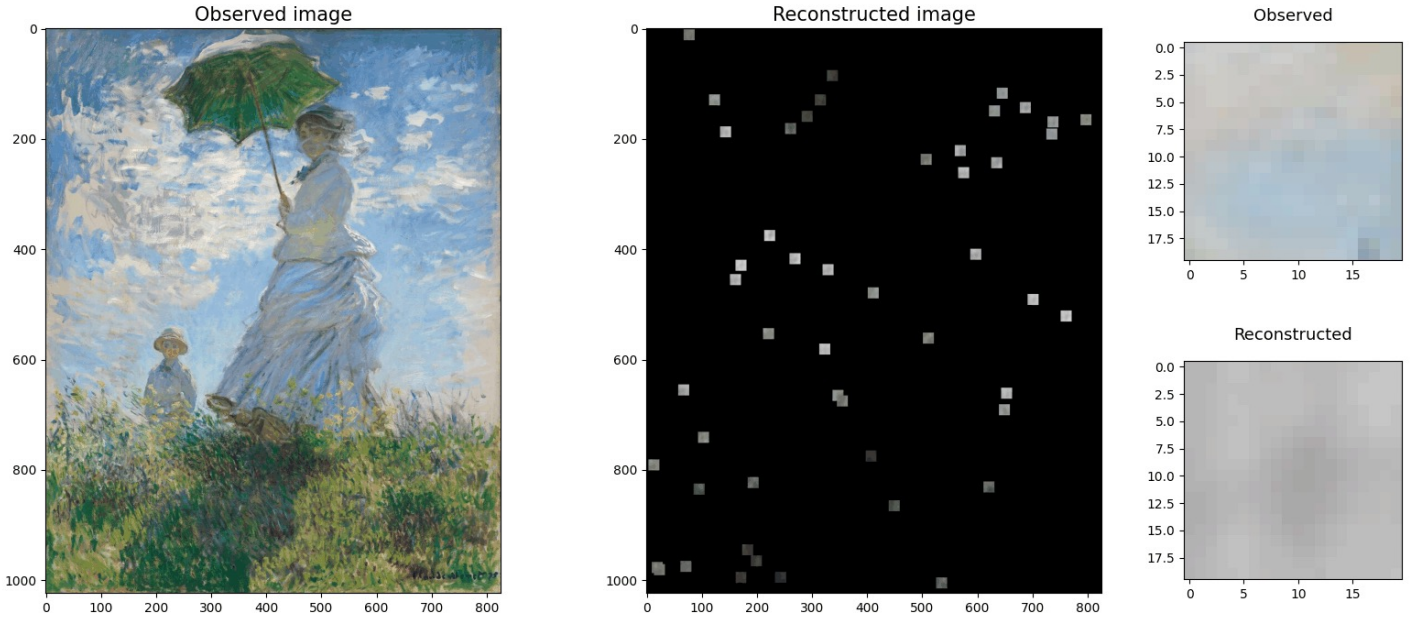
Basis images ($k\times k\times 3$) learned from B

$$= a_1 L_1 + \cdots + a_r L_r$$

$\mathbf{A_x}$: sampled sub-img

$\hat{A}_{\mathbf{x}}$: reconstructed sub-img

Image Reconstruction Algorithm



Input: Observed image A; A low-rank approximation oracle $\mathcal{R}$ for $k\times k\times 3$ matrices

Do: $\hat{A} \leftarrow \mathrm{np.zeros}(\mathrm{shape} = A.\mathrm{shape})$

Repeat:

A) Sample a $k\times k$ window $\mathbf{x}$ in A uniformly at random;

B) $A_{\mathbf{x}} \leftarrow$ $k \times k \times 3$ sub-image of induced on the window $\mathbf{x}$

C) $\hat{A}_{\mathbf{x}} \leftarrow \mathcal{R}(A)$: "Low-rank approximation" of $A_{\mathbf{x}}$

D) Add in the weights in $\hat{A}_{\mathbf{x}}$ at the corresponding pixels in $\hat{A}$

(edge weights are normalized at the end or recursively)

Questions in Image Reconstruction

- Reconstruction Error Bound:
  - $d(\boldsymbol{A}, \widehat{\boldsymbol{A}}) \leq F(\text{scale } k, \text{ avg. approximation error at } k \times k \text{ scale})$?

Observed image $\boldsymbol{A}$

Reference image $\boldsymbol{B}$

Reconstructed image $\widehat{\boldsymbol{A}}$

Questions in Image Reconstruction

- Reconstruction Error Bound:
  - $d(\boldsymbol{A}, \widehat{\boldsymbol{A}}) \leq F(\text{scale } k, \text{ avg. approximation error at } k \times k \text{ scale})$?
- Dictionary Learning:
  - How do we learn the `best' basis images $L_1, \cdots, L_r$?
  - Can we leverage **tensor structure**?

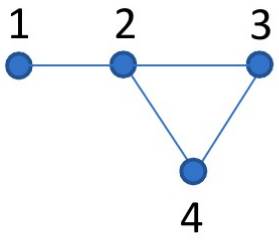
Observed image $\boldsymbol{A}$

Reference image $\boldsymbol{B}$

Reconstructed image $\widehat{\boldsymbol{A}}$

Networks: Basic language describing complex systems

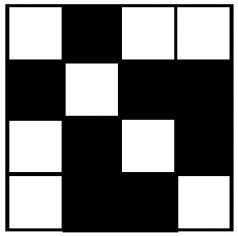
- In this talk: Simple networks (symmetric 0-1 matrices with 0's on diagonal)



Graph

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Matrix

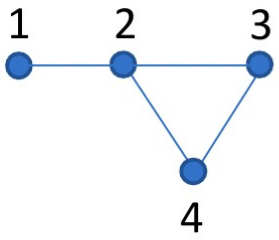


Pixel picture

- In pixel picture: Cross shape $\leftrightarrow$ hub node (node 2);
  Block shape $\leftrightarrow$ community (nodes 2,3,4)

Networks: Basic language describing complex systems

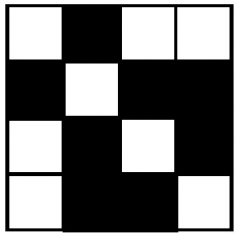
- In this talk: Simple networks (symmetric 0-1 matrices with 0's on diagonal)



Graph

	1	2	3	4
1	0	1	0	0
2	1	0	1	1
3	0	1	0	1
4	0	1	1	0

Matrix



Pixel picture

- In pixel picture: Cross shape $\leftrightarrow$ hub node (node 2);
  Block shape $\leftrightarrow$ community (nodes 2,3,4)

- Huge amount of information is being encoded into networks in various domains (e.g., Social networks, biological networks, brain networks, genetic networks, citation networks, ecology networks, economic networks, electric power networks, road networks)

- Developing proper theory and algorithm for network data analysis is becoming more important

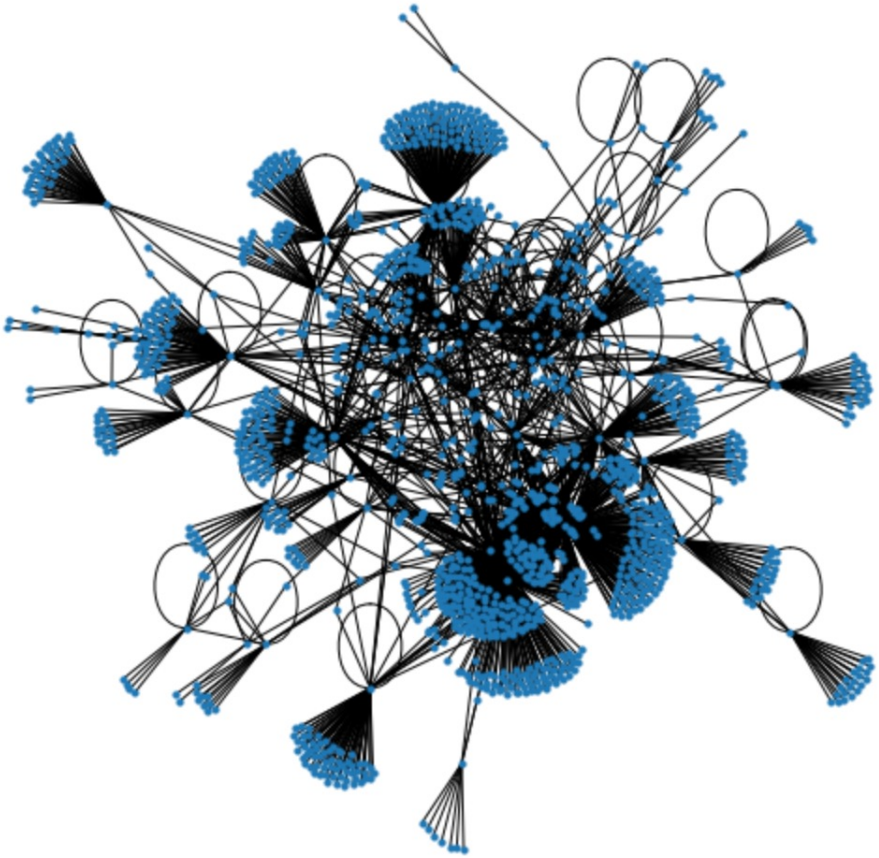


Figure. Coronavirous PPI network

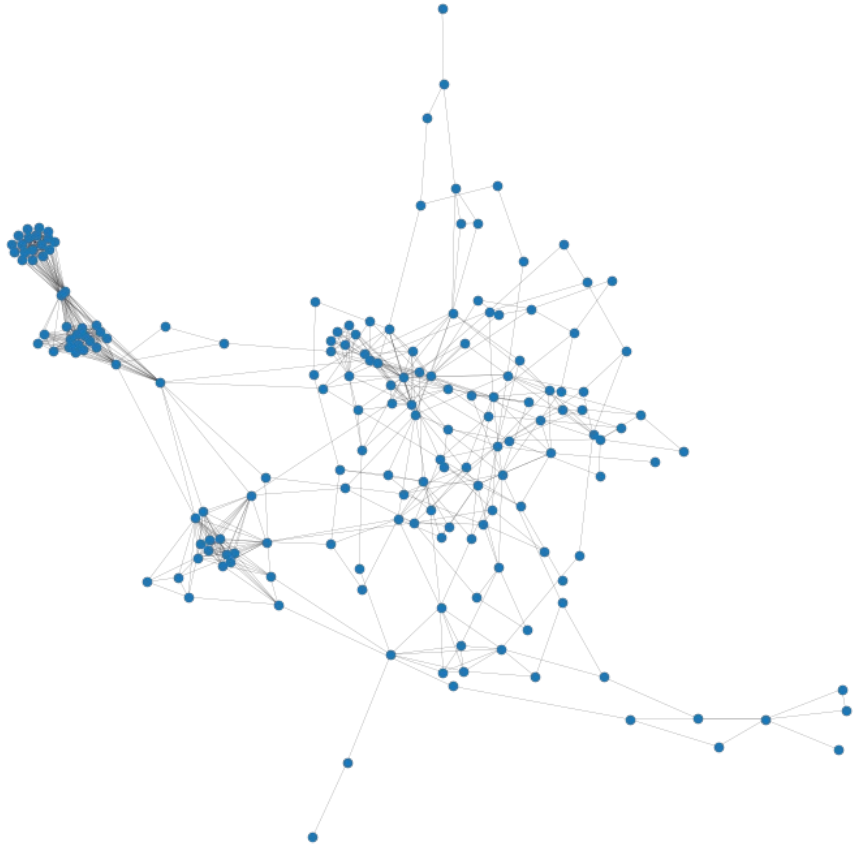


Figure. A 200-node subgraph from arXiv collaboration network

What do we mean by “Network analysis”?

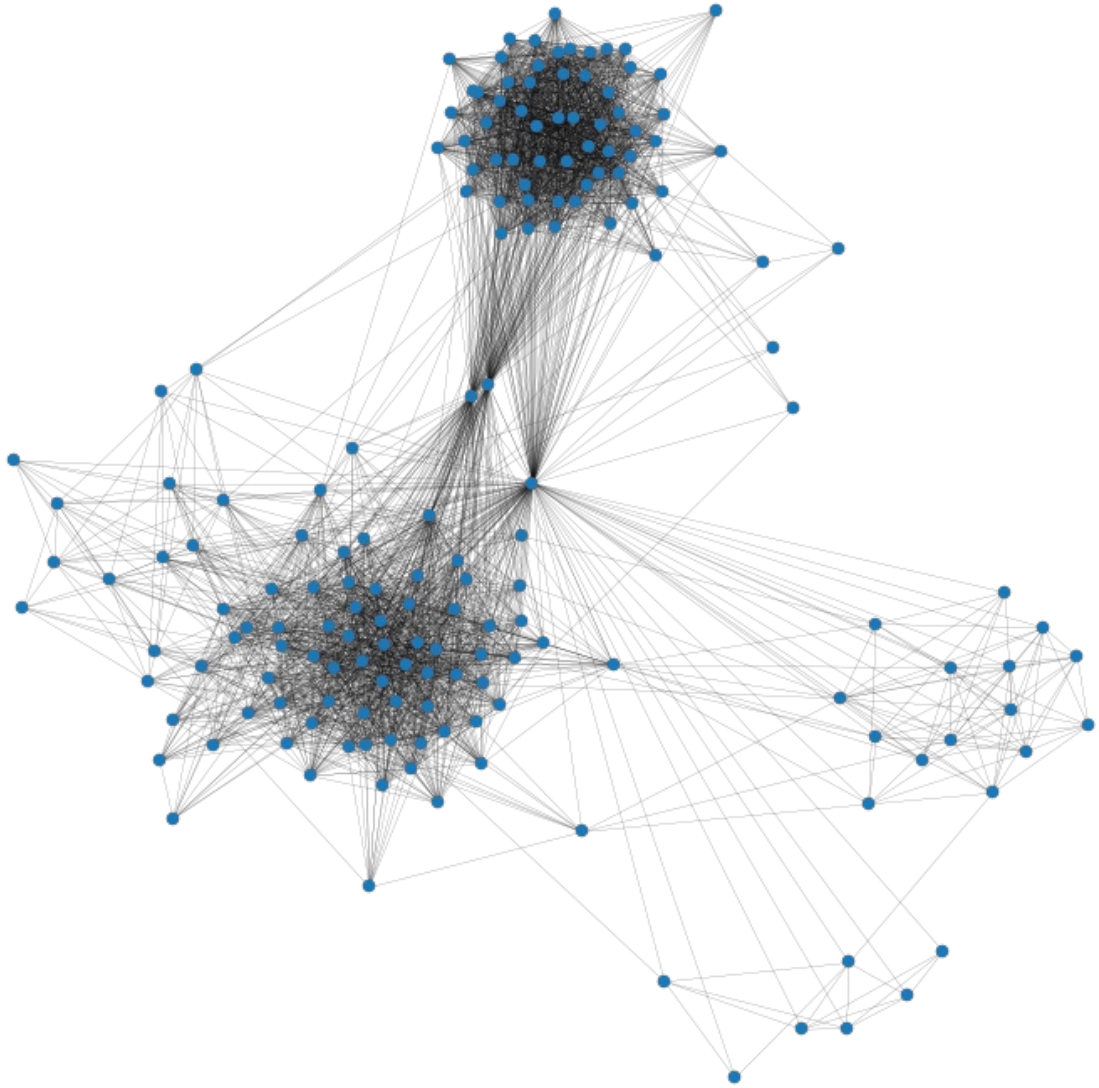


Figure. A 200-node subgraph from Facebook social network

What do we mean by “Network analysis”?

Communities

Subset of nodes better connected with themselves than to the others

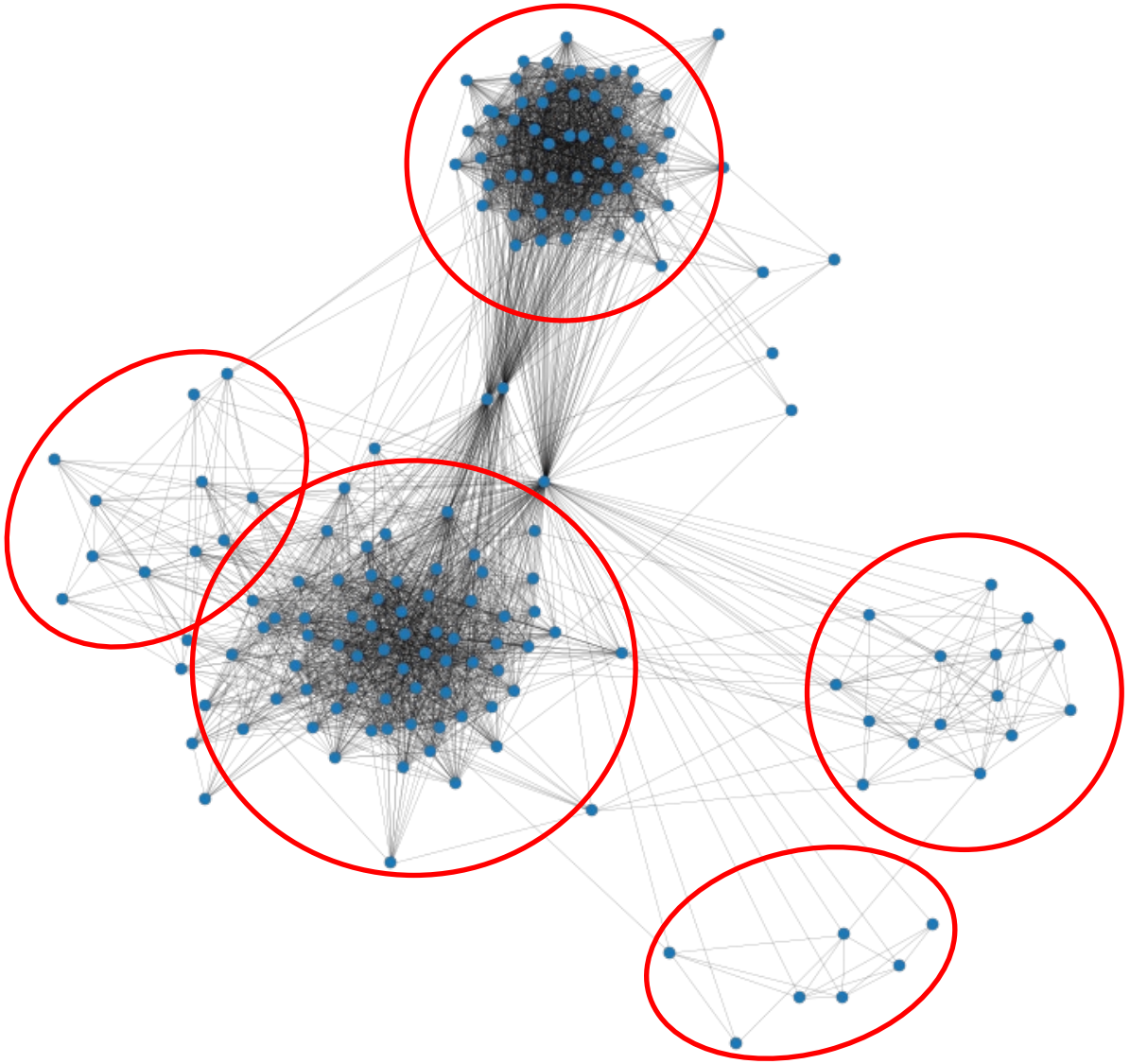


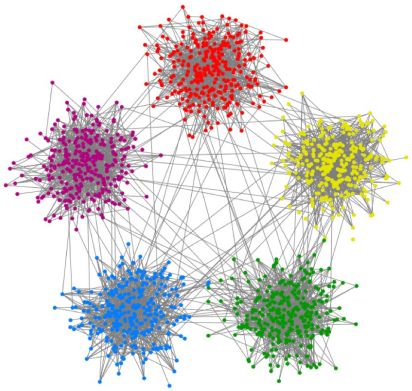
Figure. A 200-node subgraph from Facebook social network

What do we mean by "Network analysis"?

Communities

Subset of nodes better connected with themselves than to the others

Stochastic Block Model



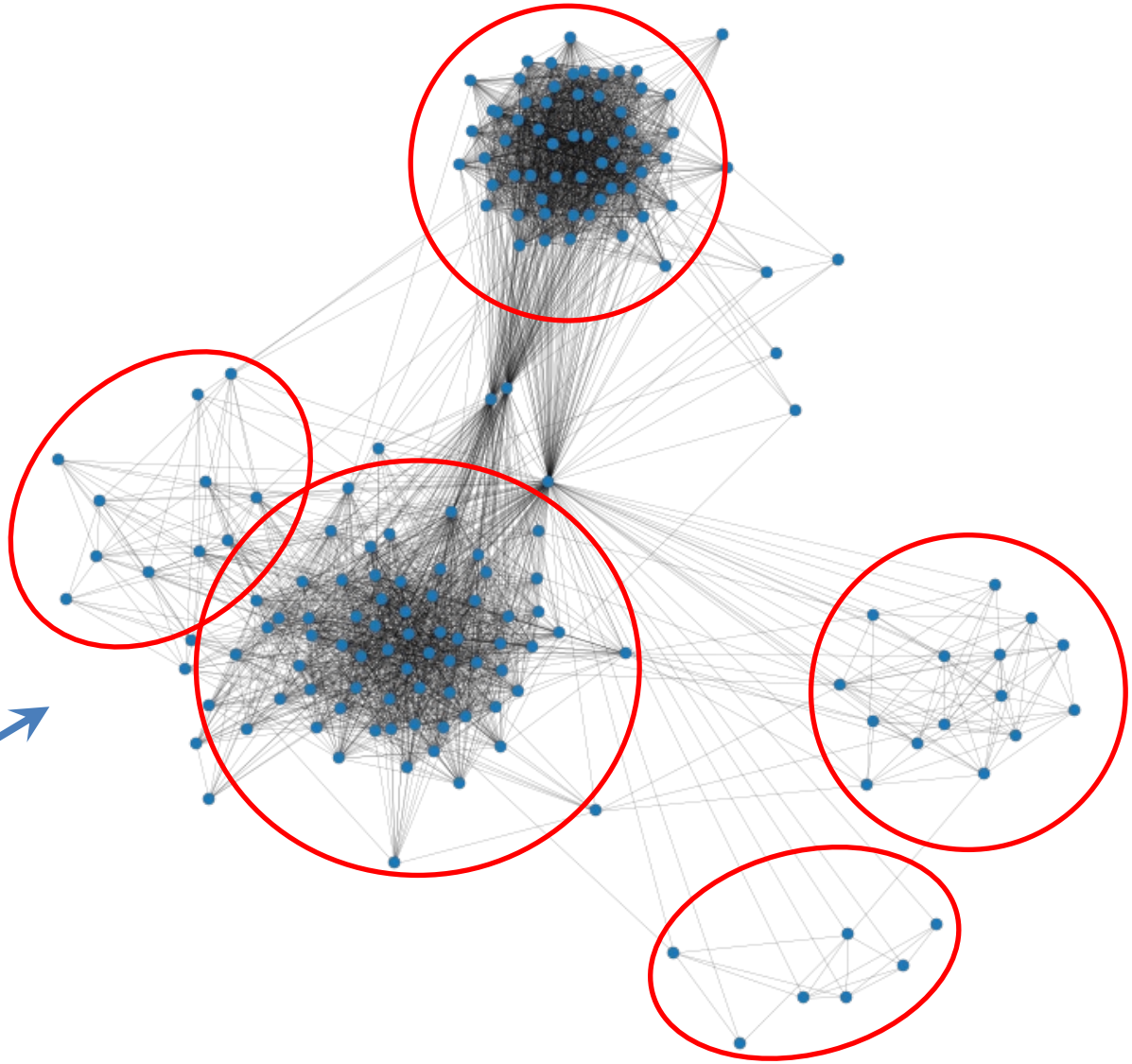


Figure. A 200-node subgraph from Facebook social network

What do we mean by “Network analysis”?

Hub nodes

Nodes that have exceptionally large degree

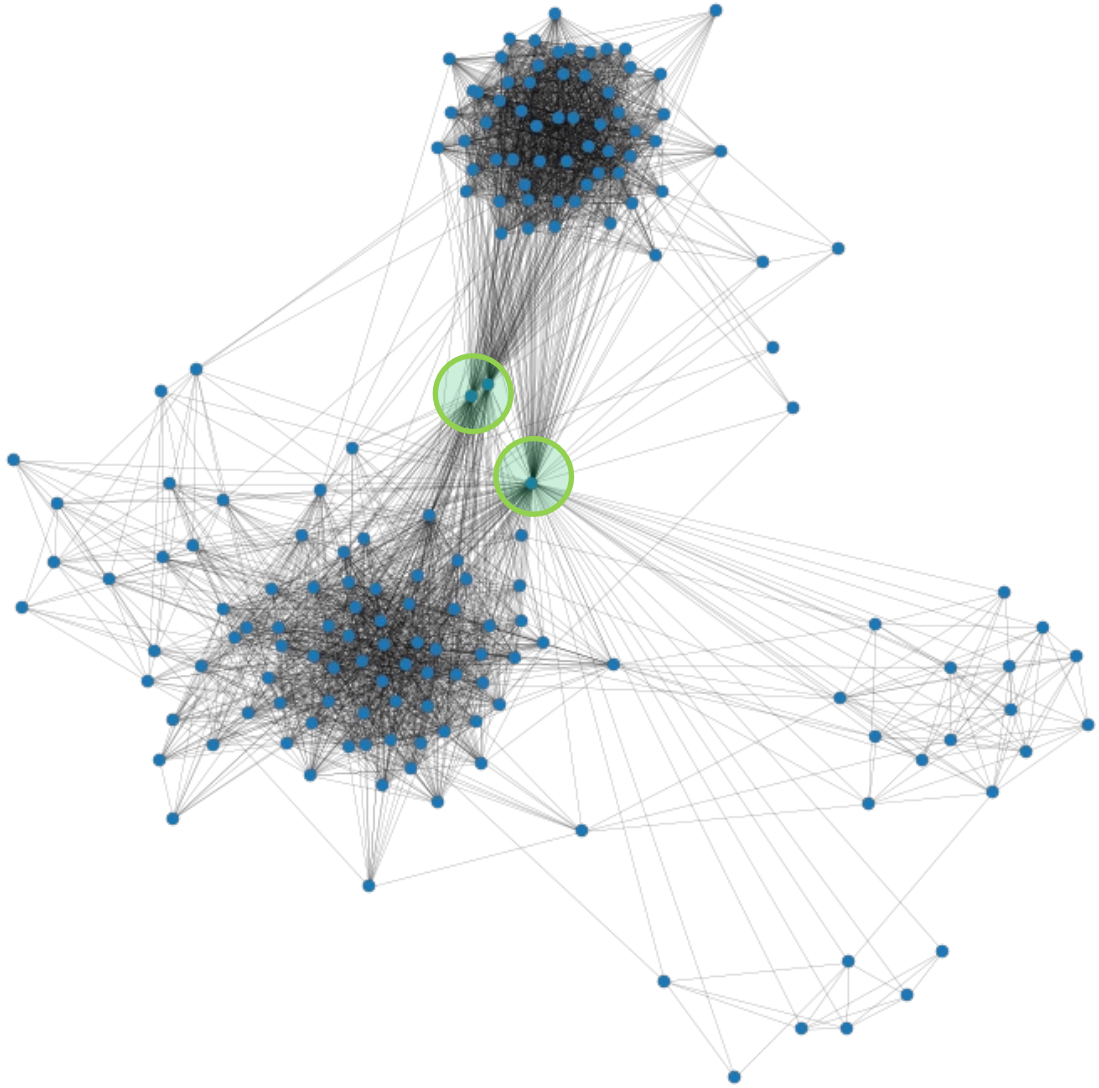


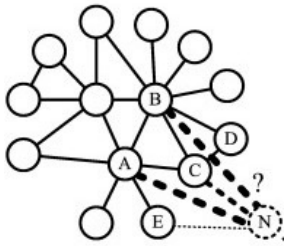
Figure. A 200-node subgraph from Facebook social network

What do we mean by "Network analysis"?

Hub nodes

Nodes that have exceptionally large degree

Preferential Attachment model



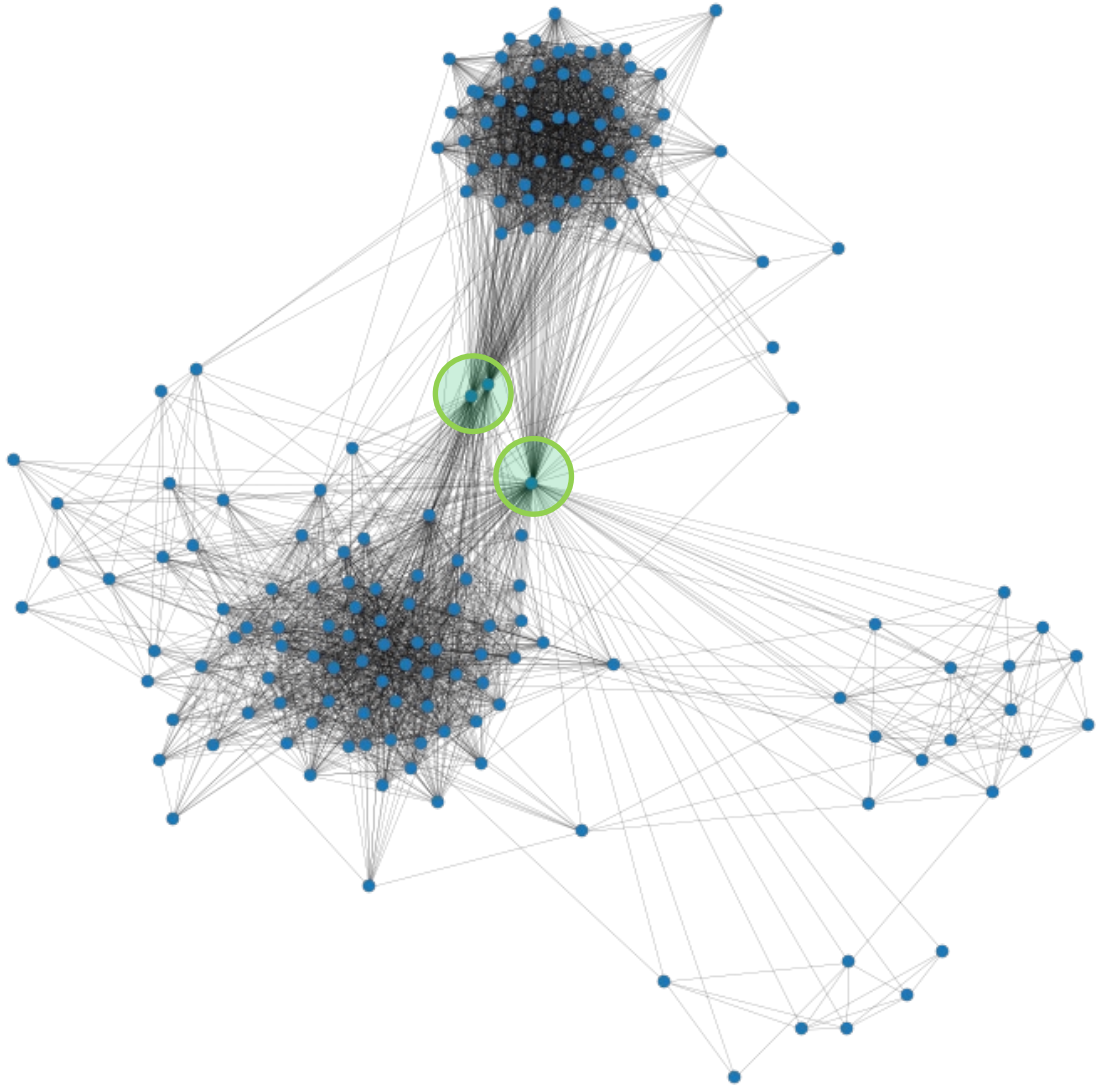
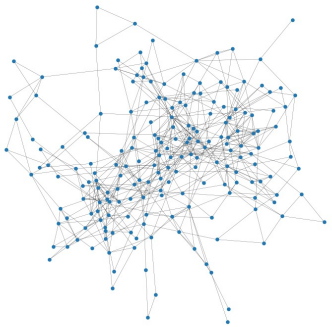
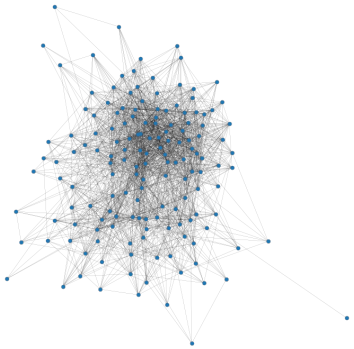


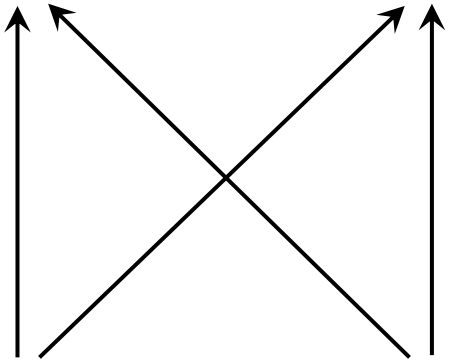
Figure. A 200-node subgraph from Facebook social network

UCLA Facebook Network

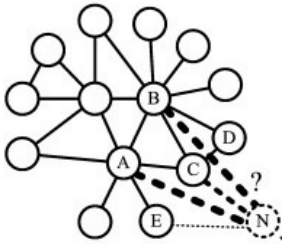
CALTECH Facebook Network



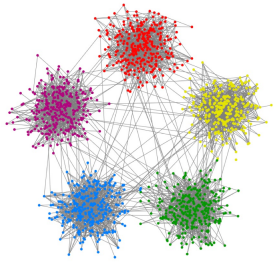




Preferential Attachment



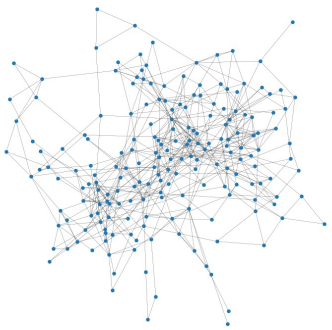
Stochastic Block Model

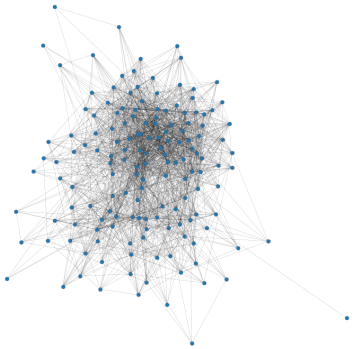


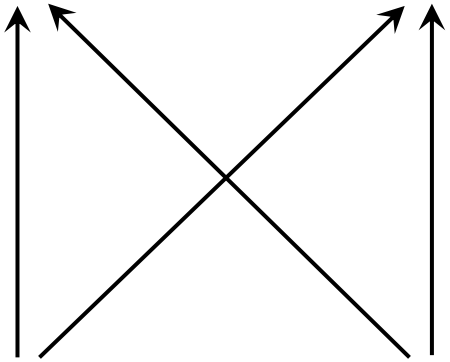
Local-level
Random Network Models

UCLA Facebook Network

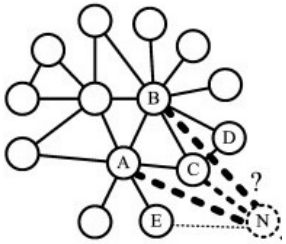
CALTECH Facebook Network



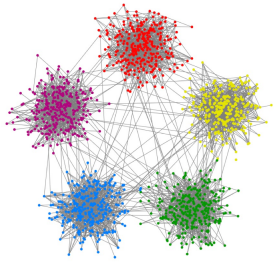




Preferential Attachment



Stochastic Block Model

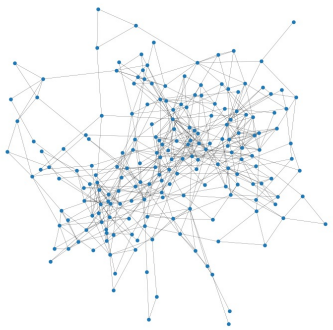


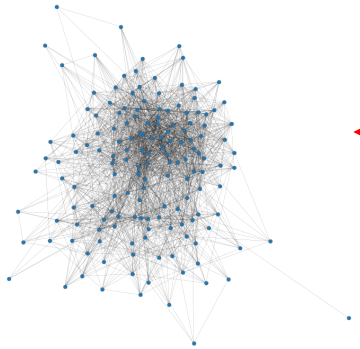
Model fitting
Parameter estimation

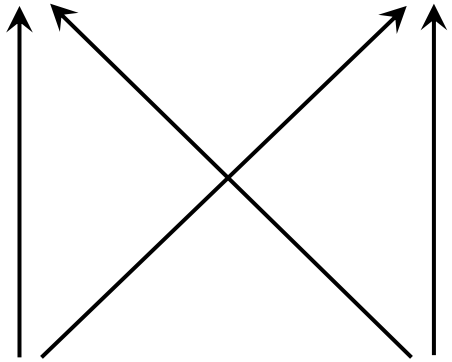
Local-level
Random Network Models

UCLA Facebook Network

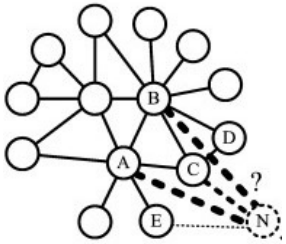
CALTECH Facebook Network



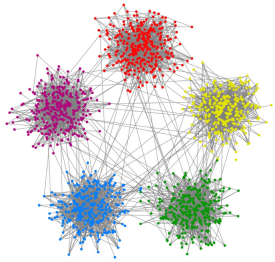




Preferential Attachment



Stochastic Block Model

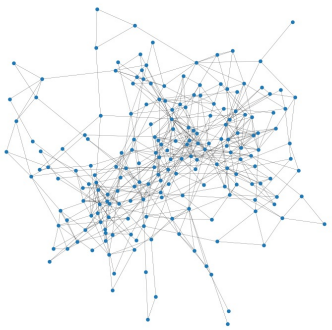


Power-law degree distribution (← PA)
Community detection (← SBM)

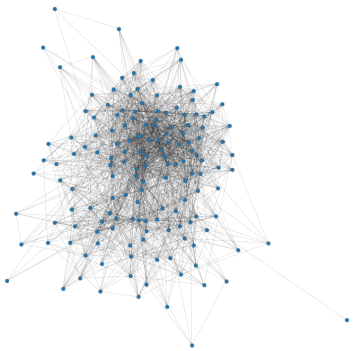
Model fitting
Parameter estimation

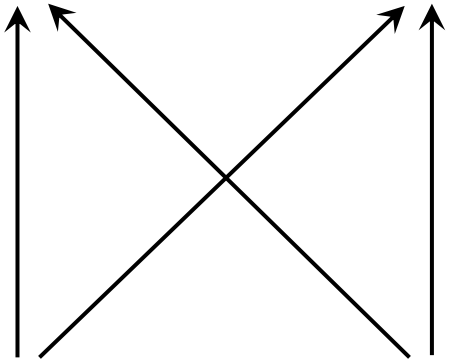
Local-level
Random Network Models

UCLA Facebook Network

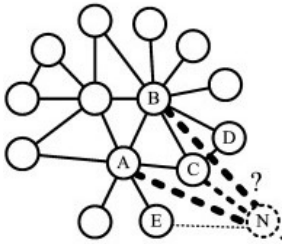


CALTECH Facebook Network

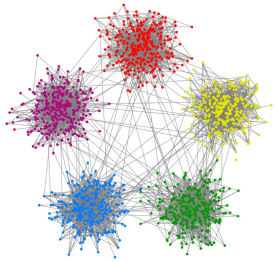




Preferential Attachment



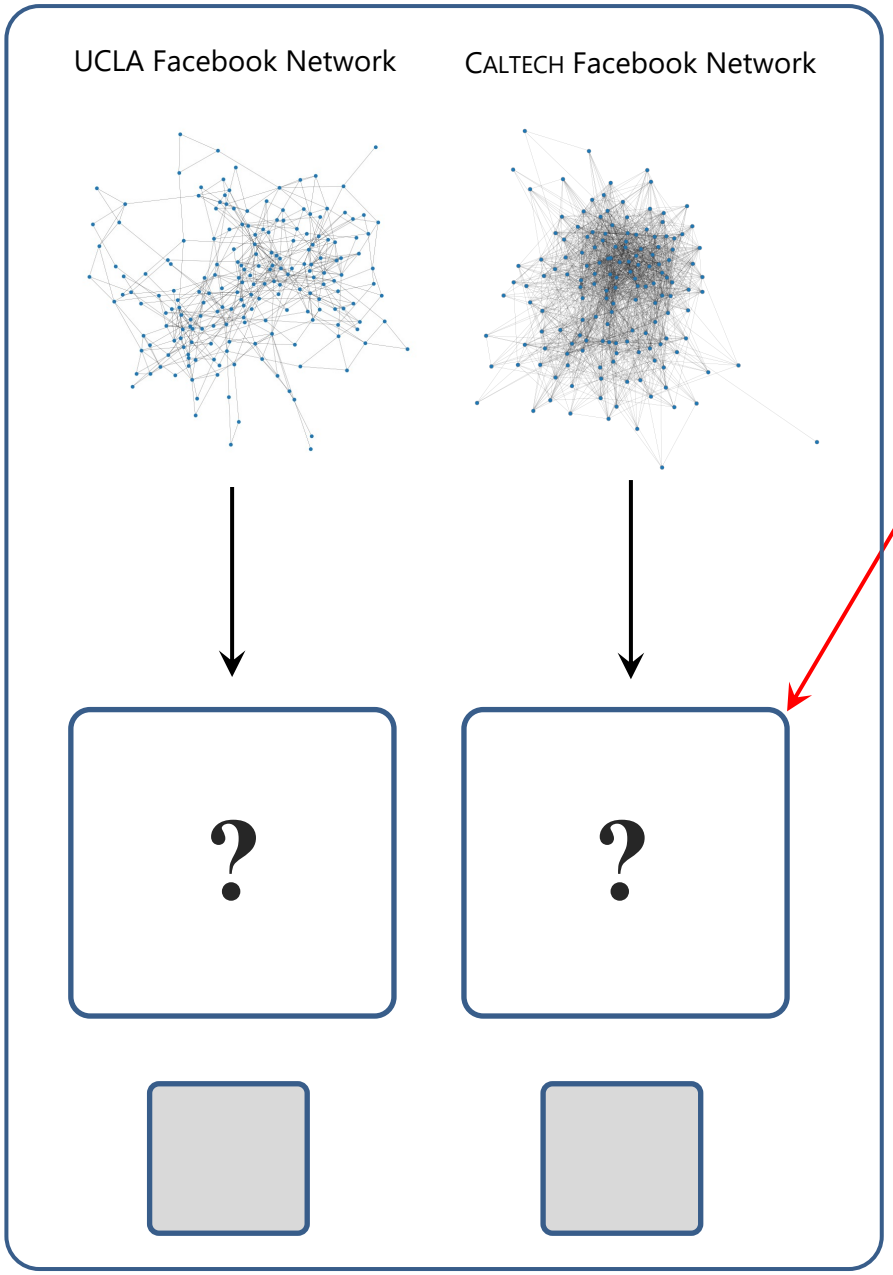
Stochastic Block Model



Limitations of model-based approach

- How to choose the right model?
- What if there is no right model?
- What if model fitting is too expensive?
- Develop new models to explain new structure?
- Develop new models to explain new networks?

Local-level Random Network Models

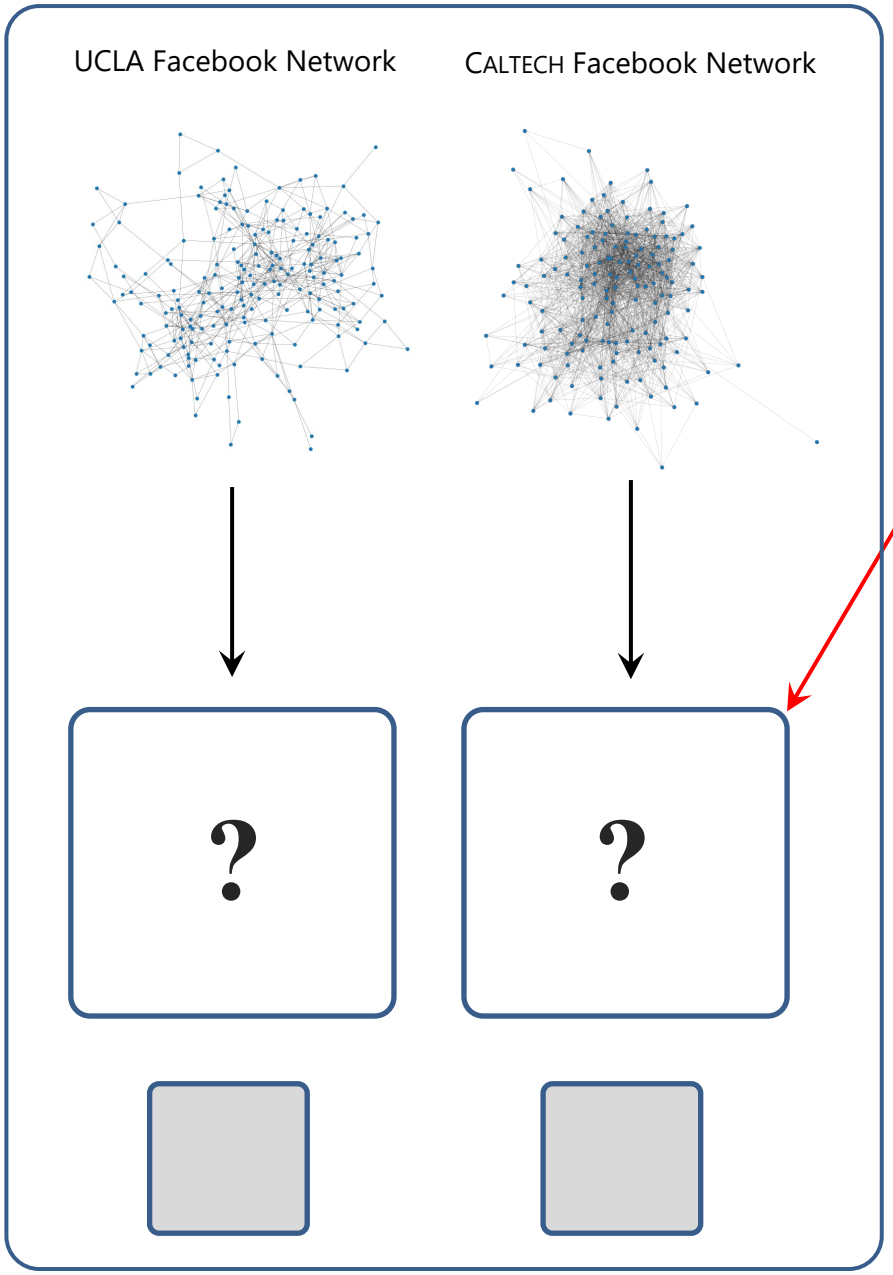


- **Mesoscale** (intermediate-scale):

  Large enough to have rich structures

  Small enough to do statistics

We can compute/learn them from networks



- **Mesoscale** (intermediate-scale):

  Large enough to have rich structures

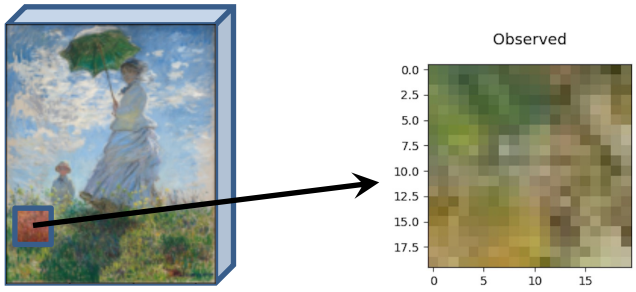
  Small enough to do statistics

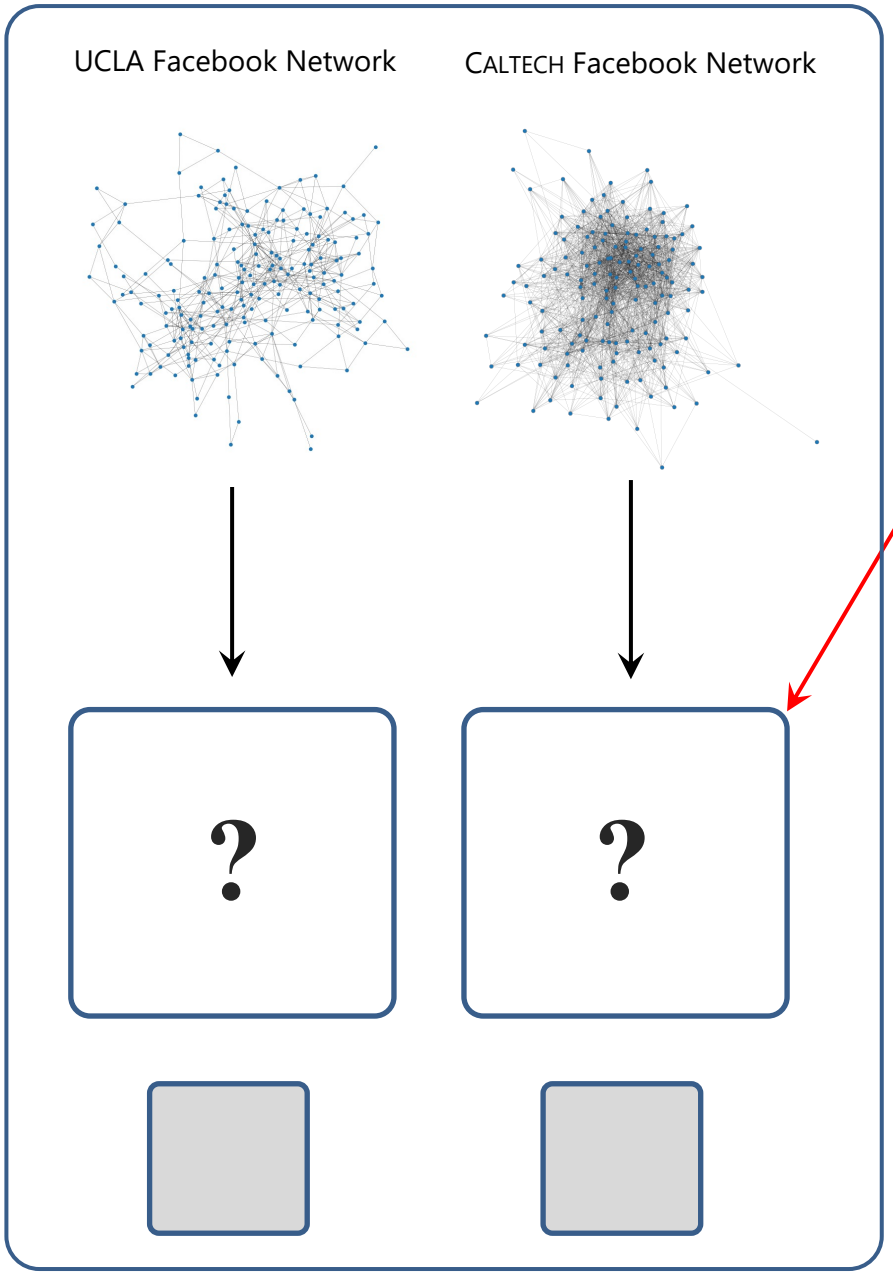
We can compute/learn them from networks

Mesoscale structure of Images

≈ Structure involving

$k \times k$ sub-images





- **Mesoscale** (intermediate-scale):

  Large enough to have rich structures

  Small enough to do statistics

We can compute/learn them from networks

- In this talk:

Mesoscale structure of networks

≈ Network structure involving k-node subgraphs

What do we mean by "Mesoscale Network analysis"?

Are there some "**important subgraphs (network motifs)**"?

Statistically significant
(Surprisingly frequent)

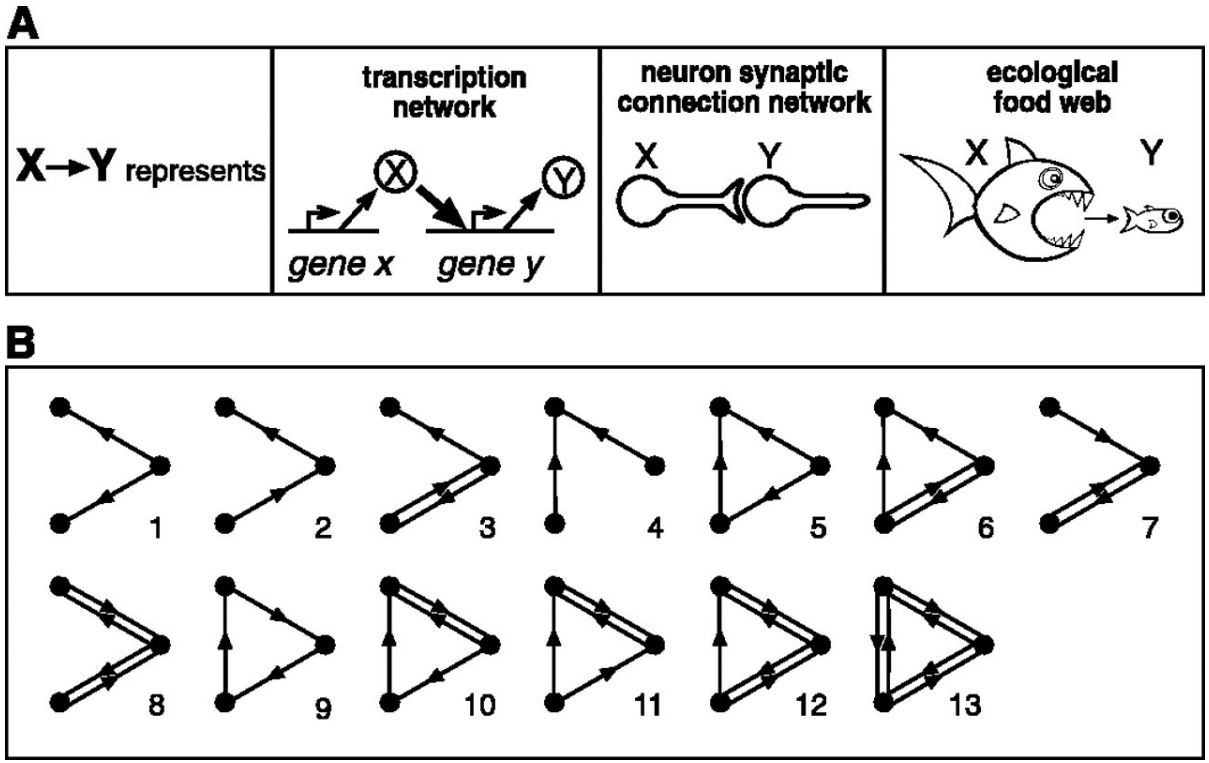
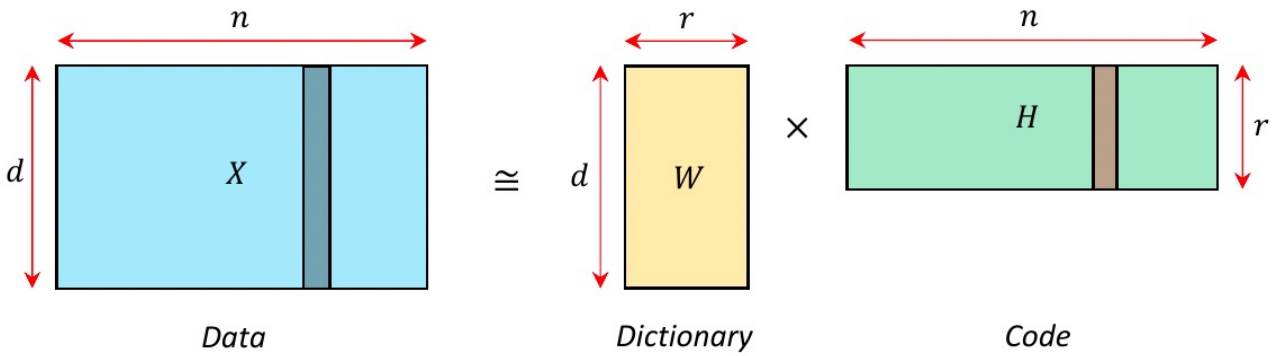


Figure. Three node biological Network Motifs (excerpted from [1])

[1] Milo, Ron, et al. "Network motifs: simple building blocks of complex networks." Science 298.5594 (2002): 824-827.

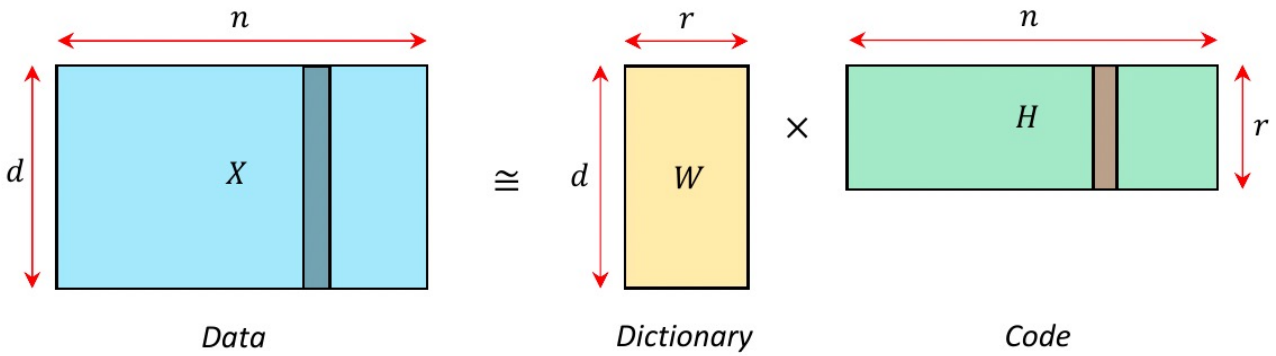
Low-rank structure of data matrices

∃ a few latent basis vectors



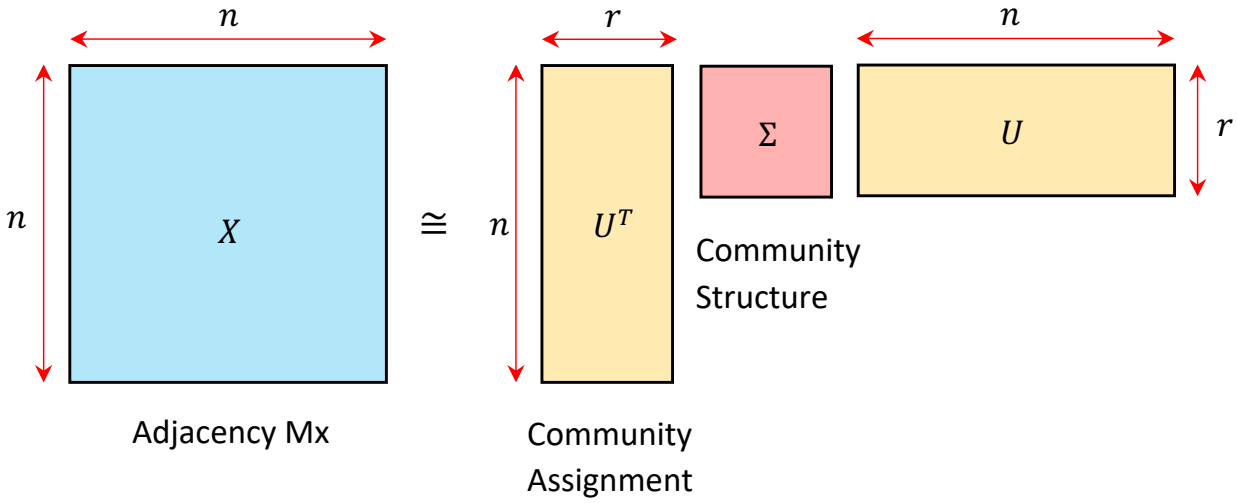
Low-rank structure of data matrices

∃ a few latent basis vectors



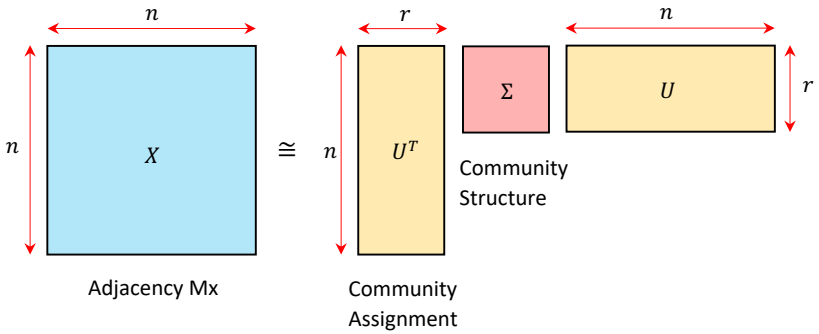
Low-rank structure of networks

∃ global community structure/Network kernel



Do networks have low-rank structure?

∃ global community structure/Network kernel



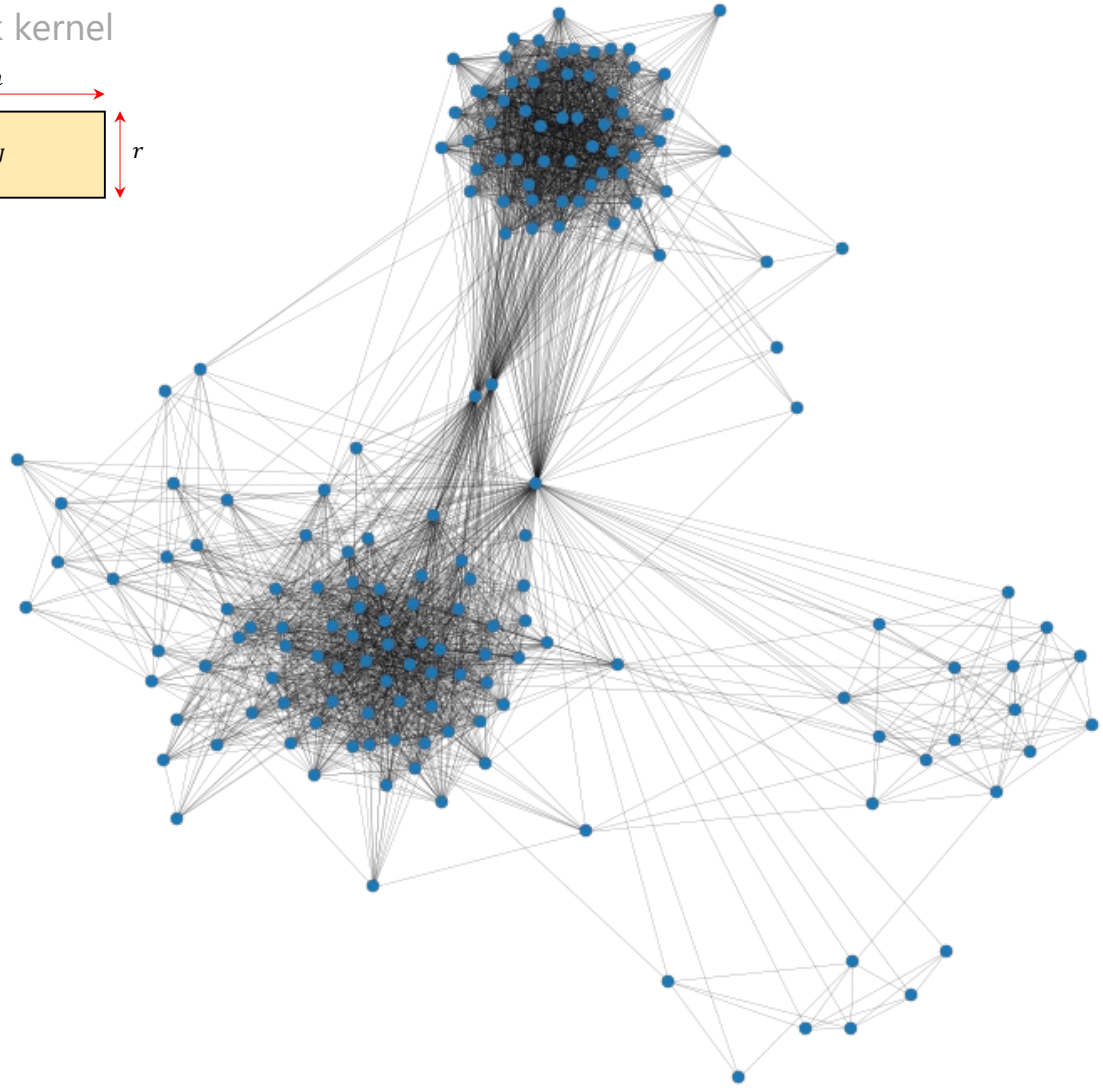
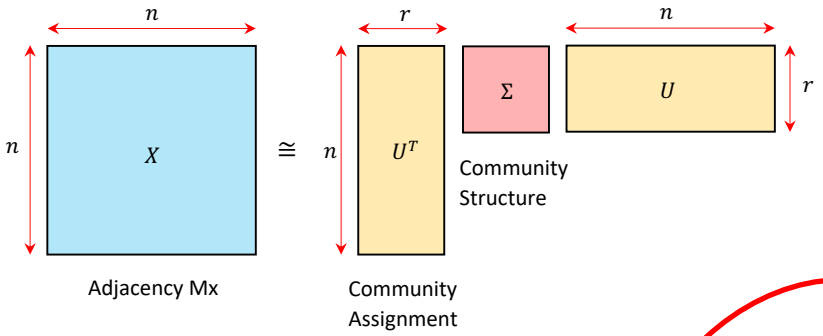


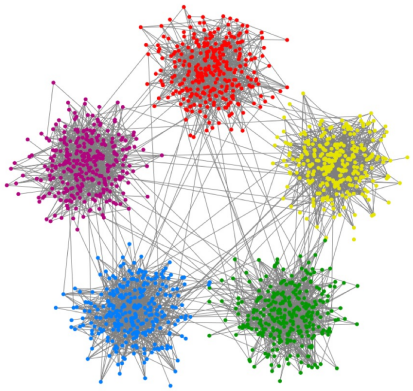
Figure. A 200-node subgraph from Facebook social network

Do networks have low-rank structure?

∃ global community structure/Network kernel



Stochastic Block Model



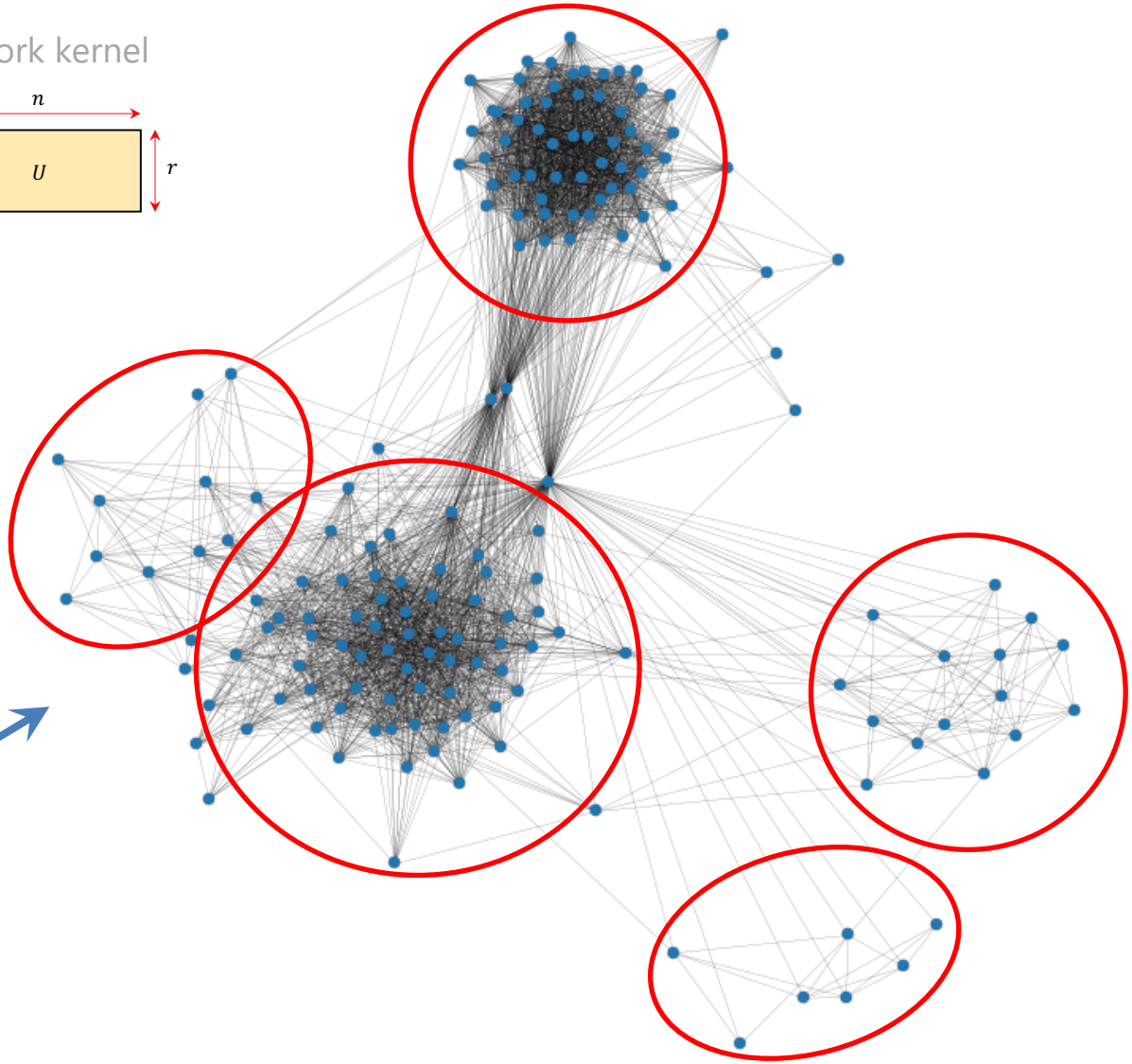
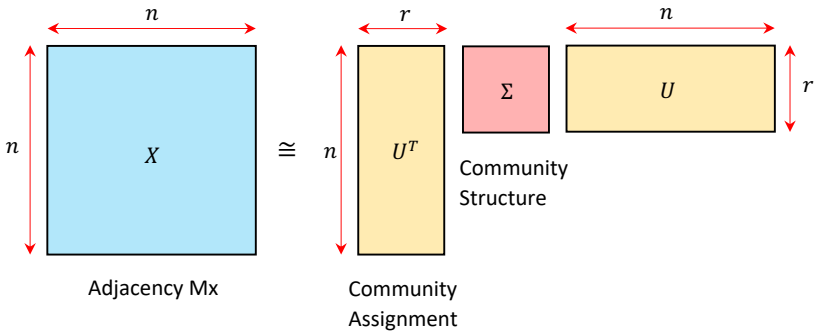


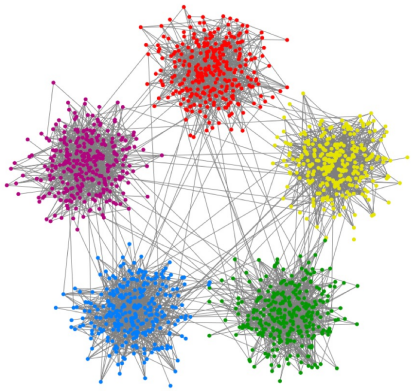
Figure. A 200-node subgraph from Facebook social network

Do networks have low-rank structure?

∃ global community structure/Network kernel



Stochastic Block Model



?

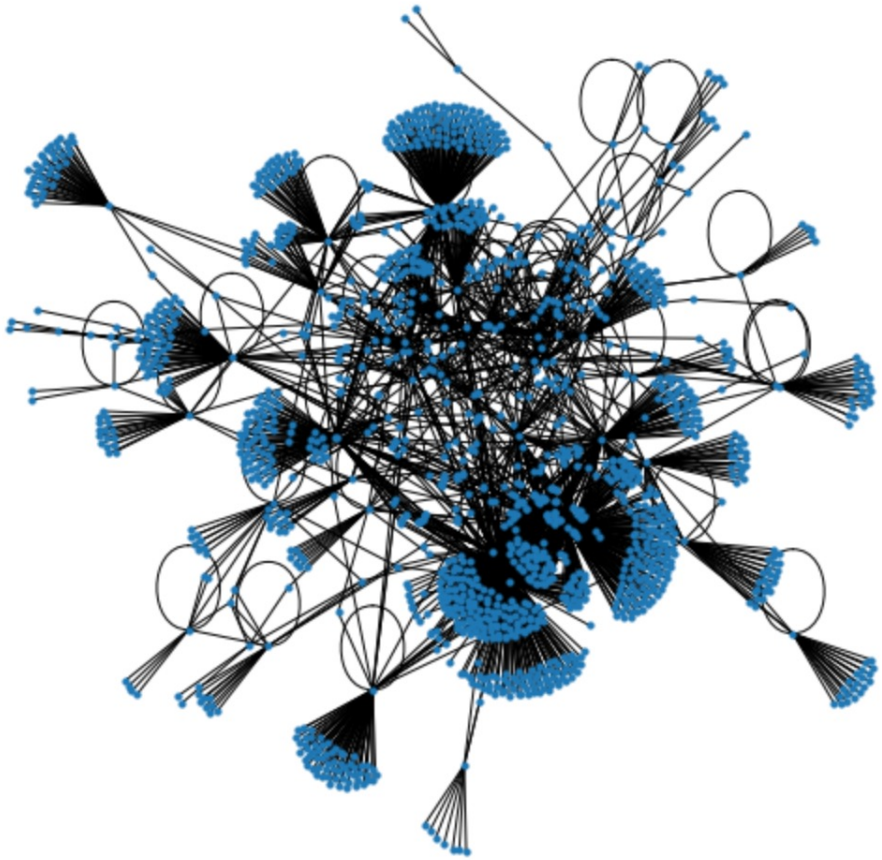
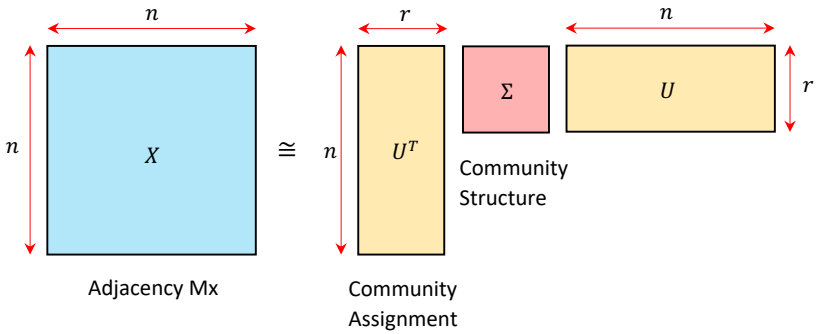


Figure. Coronavirous PPI network

Do networks have low-rank structure?

∃ global community structure/Network kernel



Impossible to have low-rank approximation of networks with many triangles among low-degree nodes

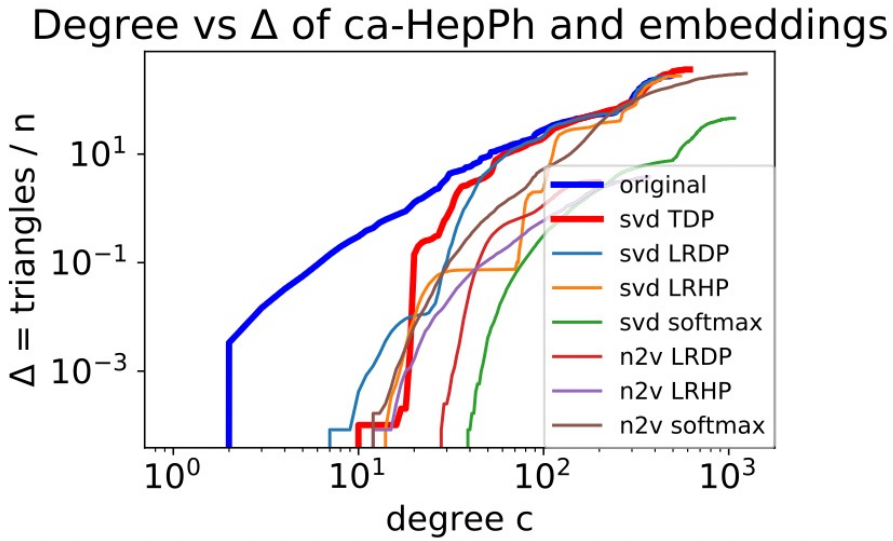
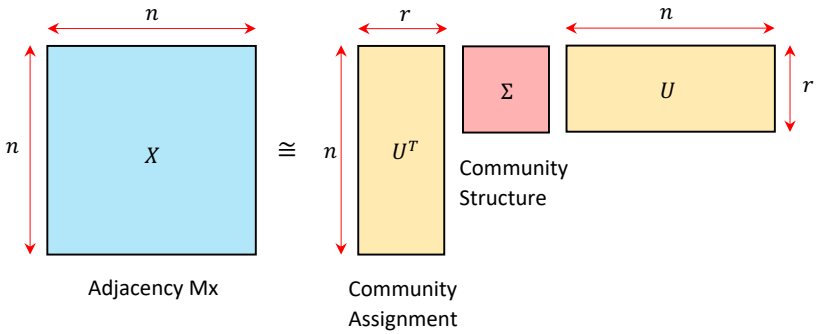


Figure from [3]

[3] Seshadhri, C., et al. "The impossibility of low-rank representations for triangle-rich complex networks." *Proceedings of the National Academy of Sciences* 117.11 (2020): 5631-5637.

Do networks have low-rank structure?

∃ global community structure/Network kernel



Do networks have low-rank structure
at Mesoscale?

Impossible to have low-rank approximation of networks with many triangles among low-degree nodes

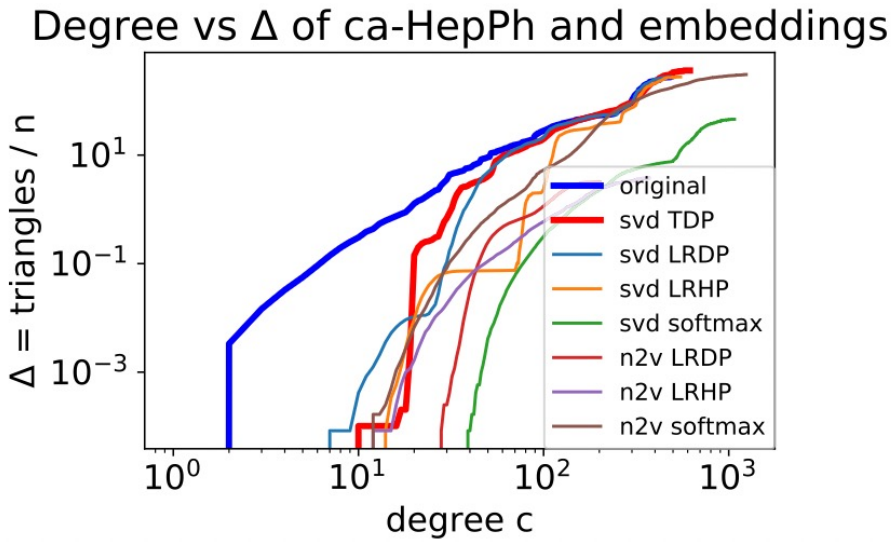
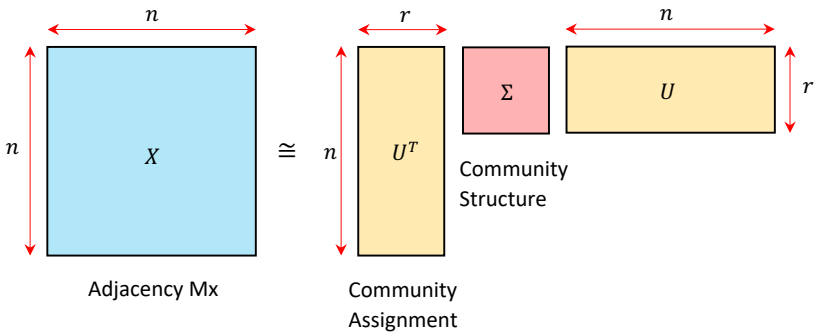


Figure from [3]

[3] Seshadhri, C., et al. "The impossibility of low-rank representations for triangle-rich complex networks." *Proceedings of the National Academy of Sciences* 117.11 (2020): 5631-5637.

Do networks have low-rank structure?

∃ global community structure/Network kernel



Do networks have low-rank structure
at Mesoscale?

⟷ Can we reconstruct networks using low-rank approx. at mesoscale?

Impossible to have low-rank approximation of networks with many triangles among low-degree nodes

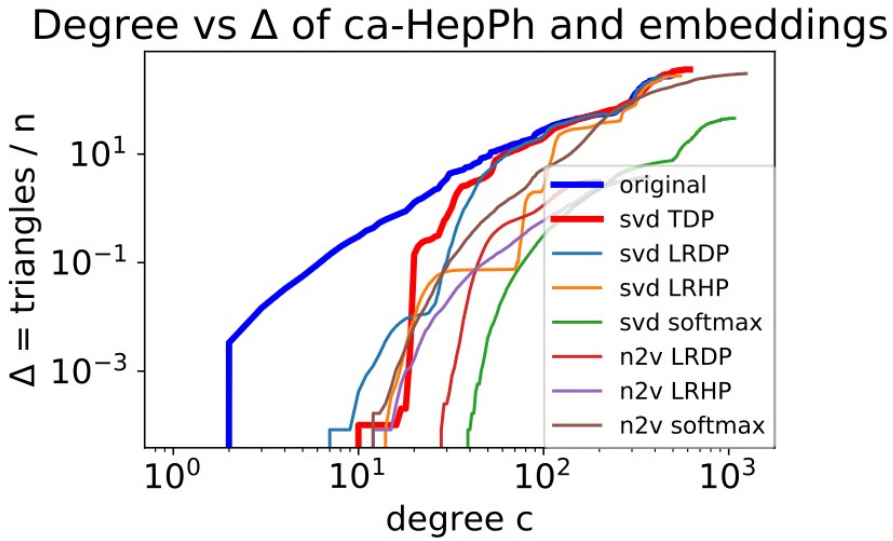
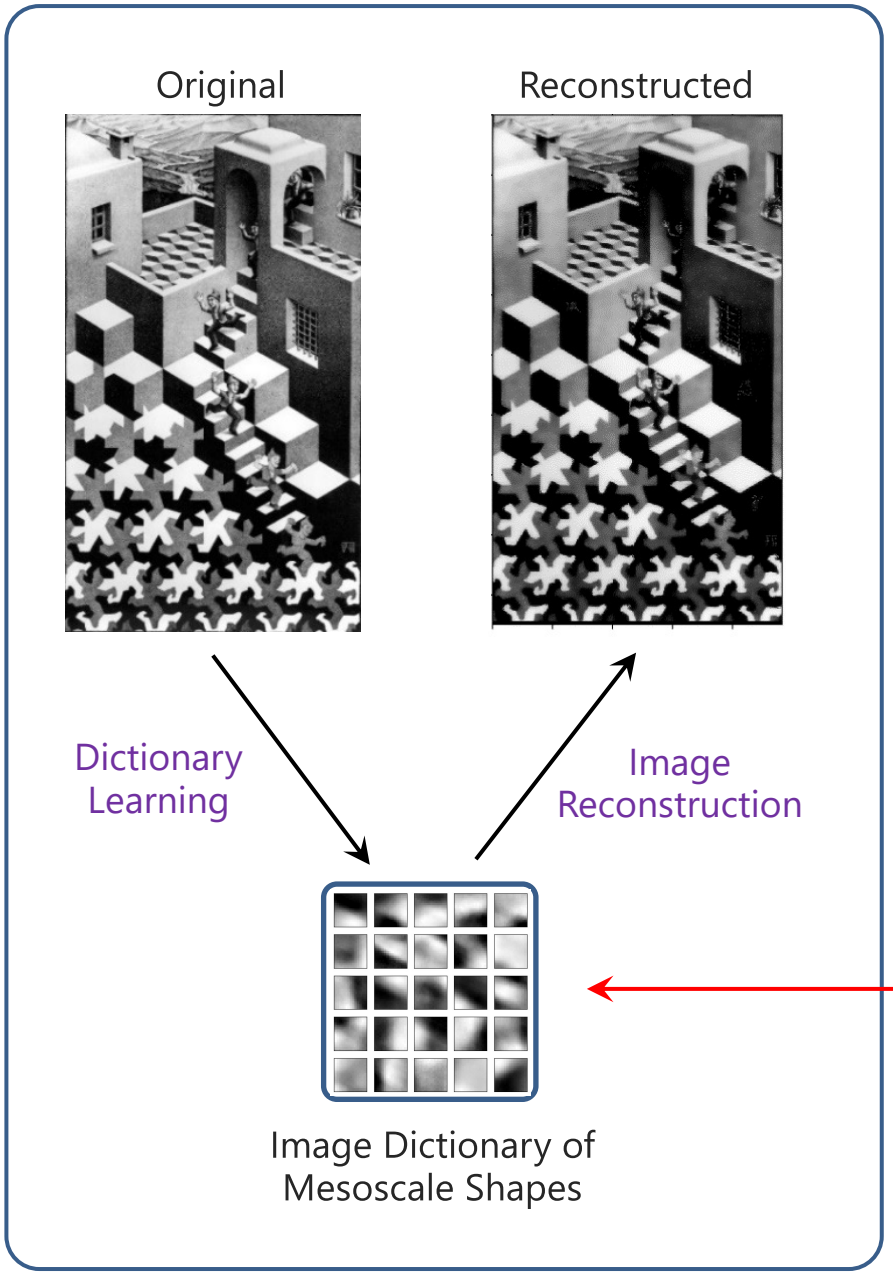


Figure from [3]

[3] Seshadhri, C., et al. "The impossibility of low-rank representations for triangle-rich complex networks." *Proceedings of the National Academy of Sciences* 117.11 (2020): 5631-5637.

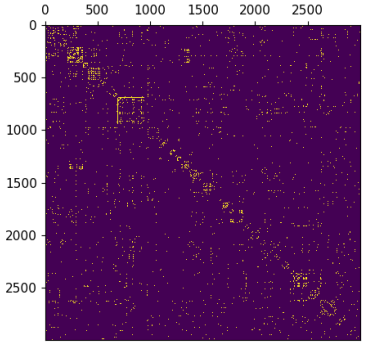


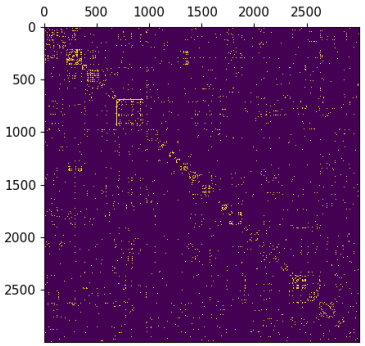
- Patch-based image processing ('06, '08, '09, '10)
- Dictionary learning
  - Matrix/Tensor Factorization
  - Sparse coding
  - Nonconvex constrained optimization
  - Stochastic Optimization for i.i.d. data

Original

Reconstructed

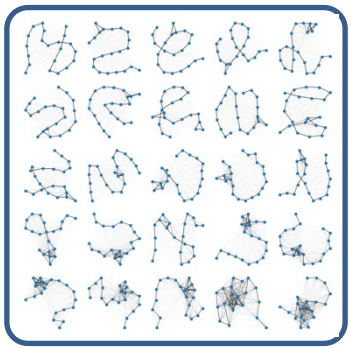
UCLA Facebook Network





Network Dictionary Learning

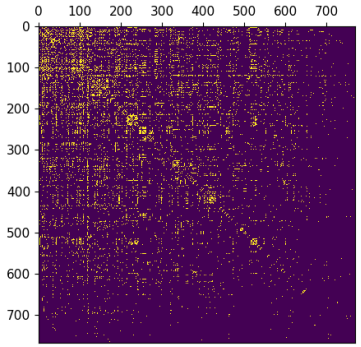
Network Reconstruction

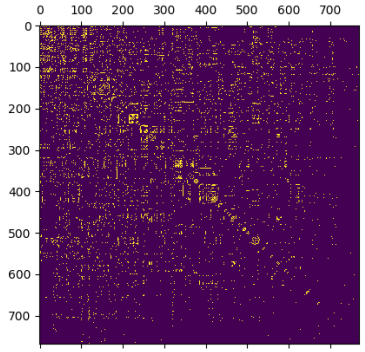


Original

Reconstructed

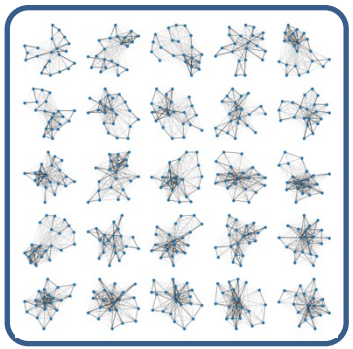
CALTECH Facebook Network





Network Dictionary Learning

Network Reconstruction

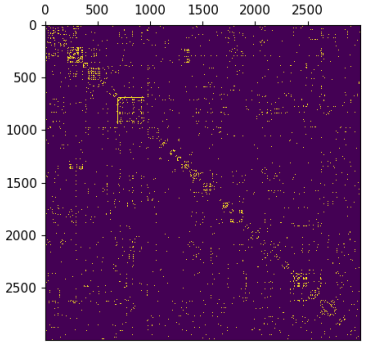


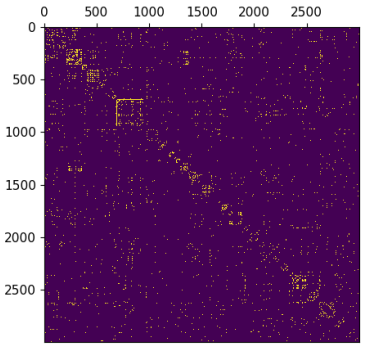
Network Dictionary of Mesoscale Subgraphs

Original

Reconstructed

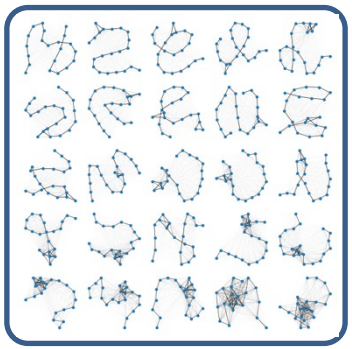
UCLA Facebook Network



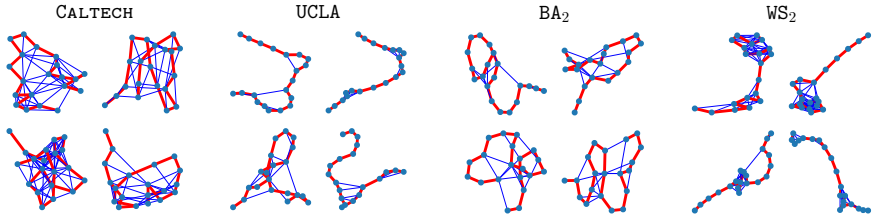


Network Dictionary Learning

Network Reconstruction



MCMC subgraph sampling
(w/ Sivakoff, Memoli, in revision for JMLR)



Network Reconstruction Theory
(w/ Kureh, Vendrow, Porter, 2022+
in revision for Nature Comms.)

Online Matrix Factorization for Markovian Data
(w/ Needell, Balzano, JMLR '20)

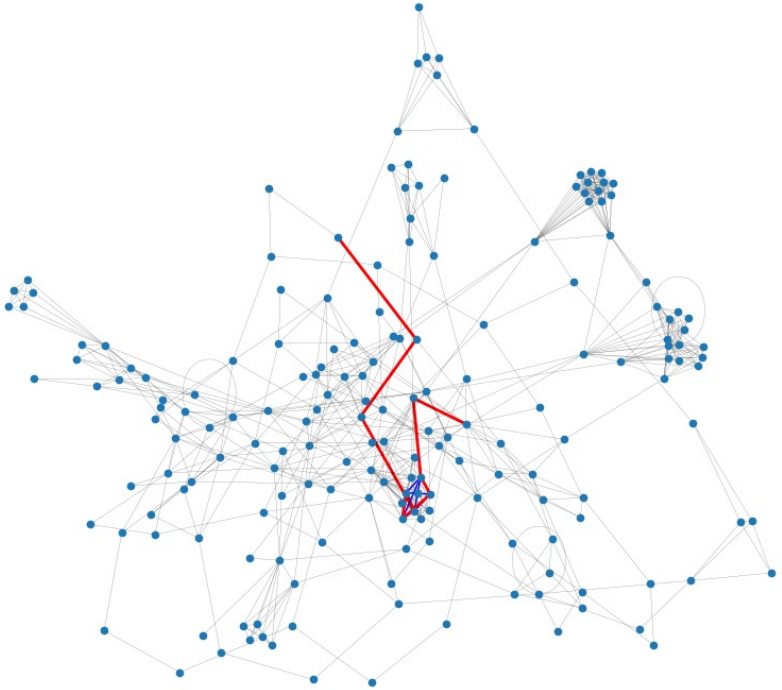
Online Tensor Factorization for Markovian Data
(w/ Strohmeier, Needell, JMLR '22)

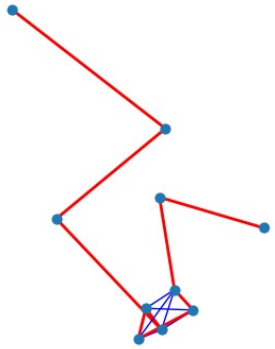
Proximal SGD for Markovian Data
(w/ Alacaoglu. '22+)

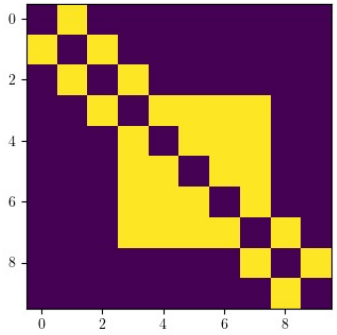
First complexity results for
Block Coordinate Descent ('20+)
Online Dictionary Learning ('22+)
Supervised Dictionary Learning ('22+)

MCMC Subgraph sampling

How do we sample subgraphs from sparse networks?





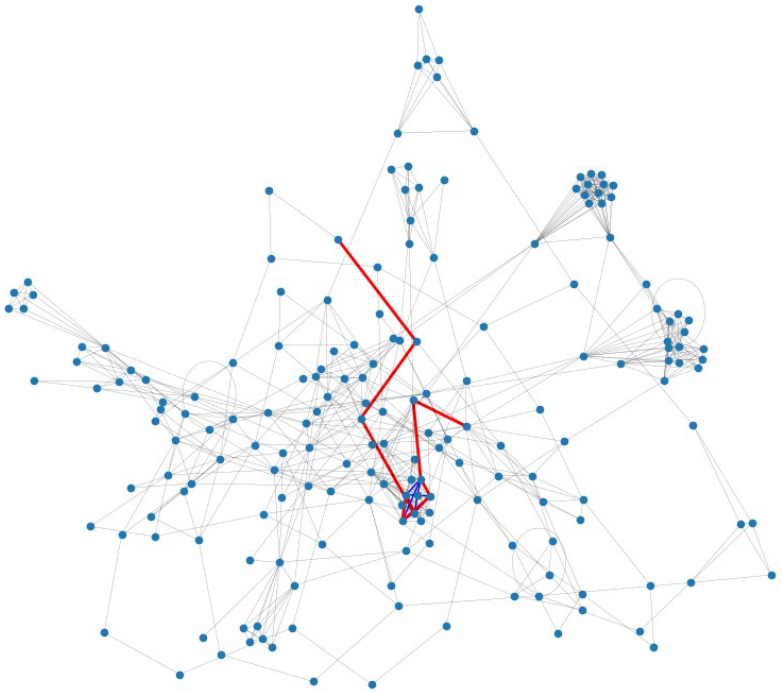


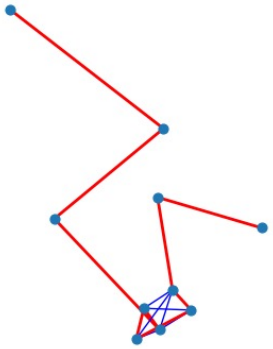
How do we sample subgraphs from sparse networks?

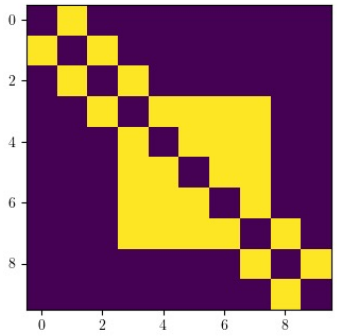
$\mathbf{x} = (x_1, x_2, \ldots, x_k), \quad x_i \sim x_{i+1}, \quad x_i's$ distinct

1. Sample a k-path $\mathbf{x} \subseteq G$ *uniformly at random*

   Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$



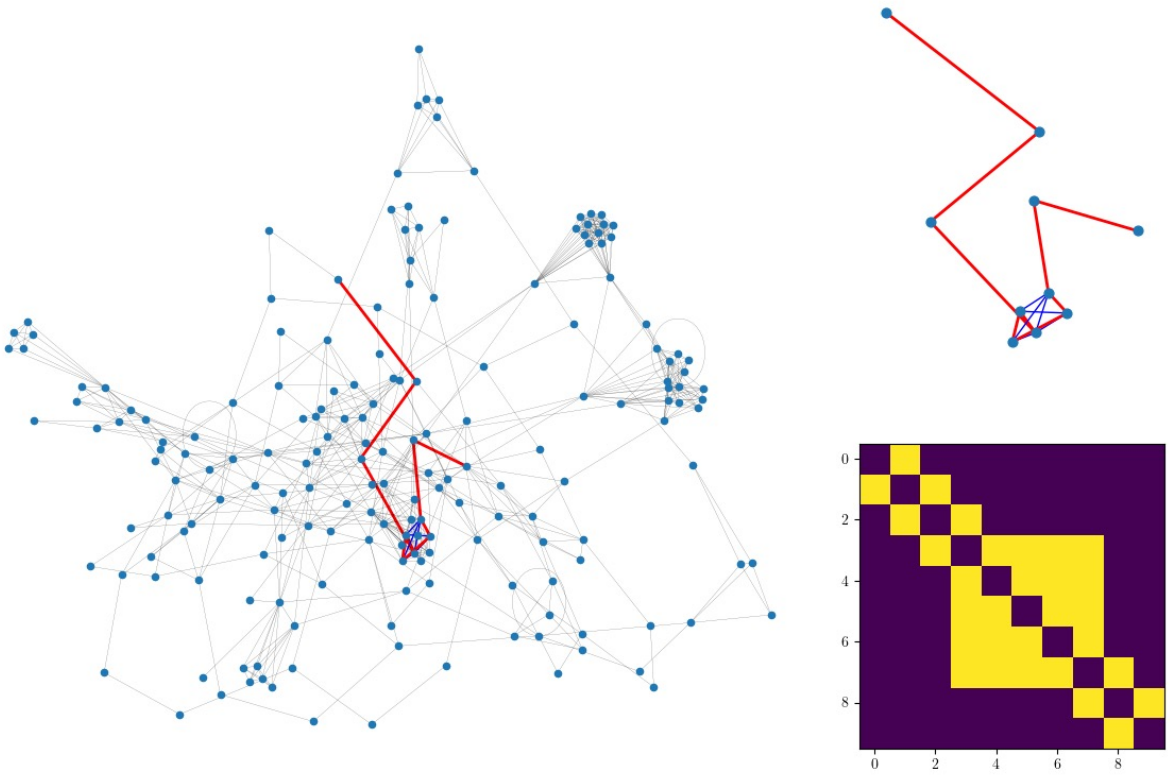




Random graph homomorphism sampling

$\mathbf{x} = (x_1, x_2, \ldots, x_k), \quad x_i \sim x_{i+1}, \quad x_i's$ distinct

1. Sample a k-path $\mathbf{x} \subseteq G$ *uniformly at random*
   Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$
2. Take the induced subgraph H on $\mathbf{x}$

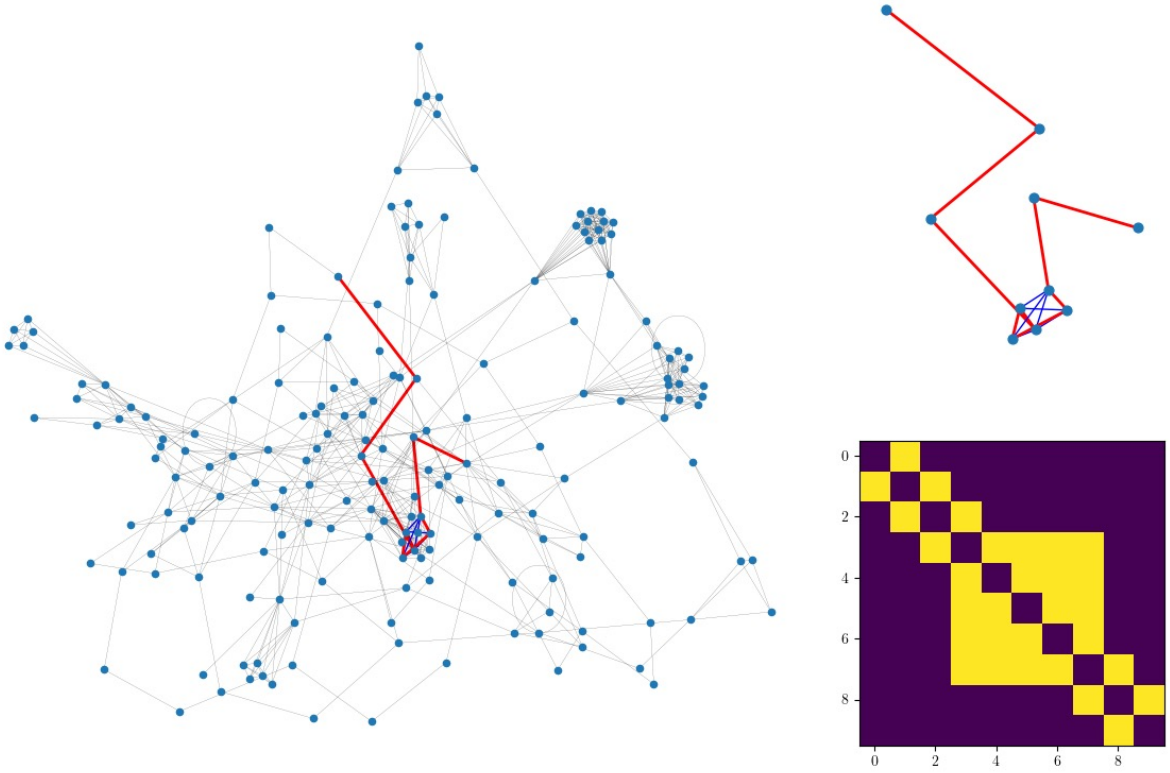


Random graph homomorphism sampling

$\mathbf{x} = (x_1, x_2, \ldots, x_k), \quad x_i \sim x_{i+1}, \quad x_i's$ distinct

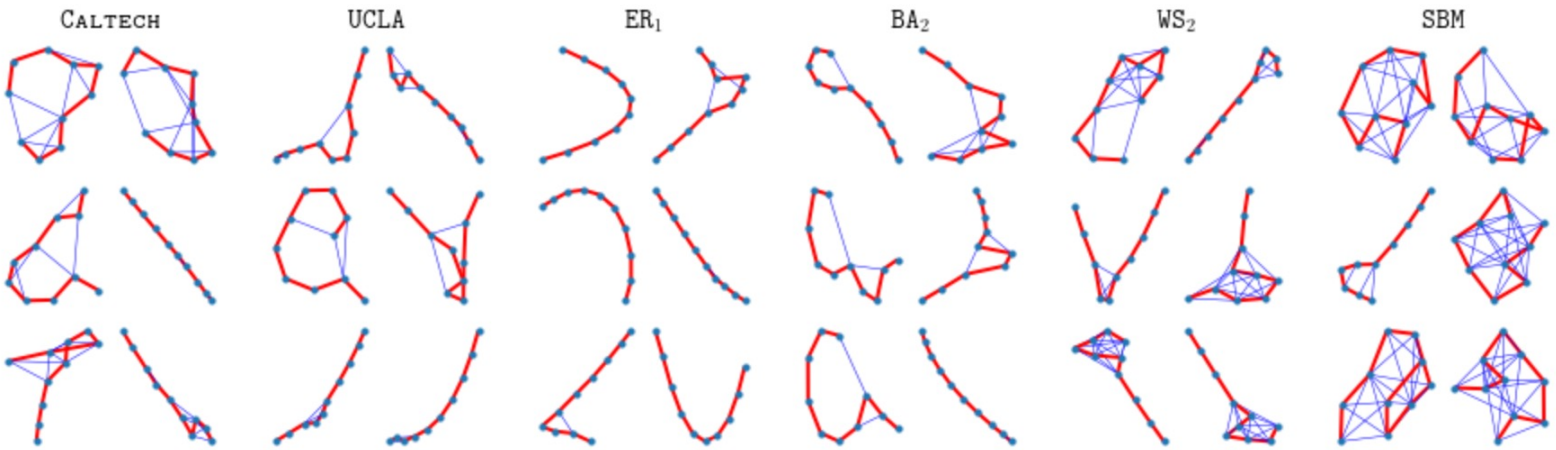
1. Sample a k-path $\mathbf{x} \subseteq G$ *uniformly at random*
   Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$
2. Take the induced subgraph H on $\mathbf{x}$

3. Returns a subgraph H with $\mathbf{x}$ its Hamiltonian path

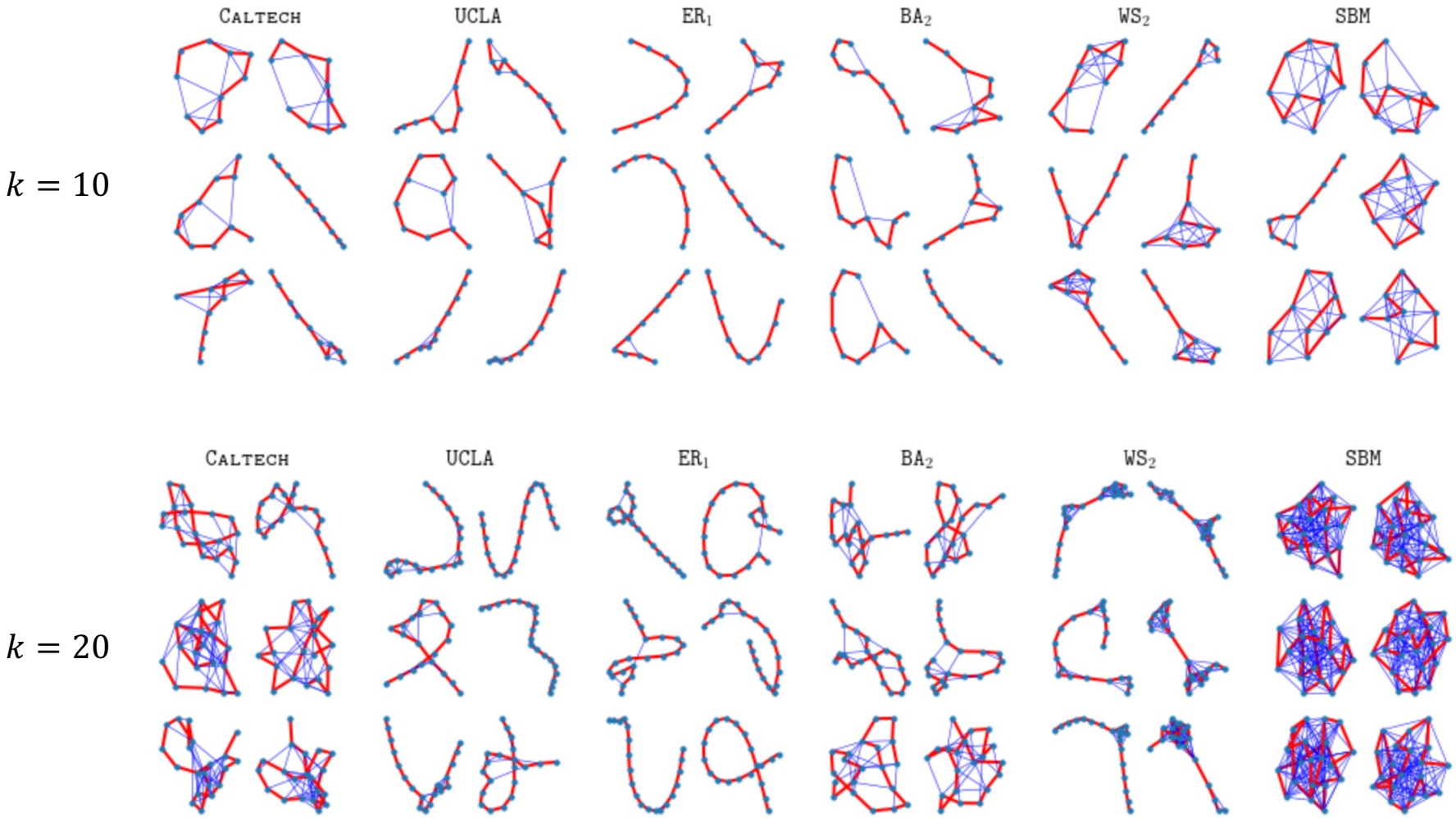


Random graph homomorphism sampling

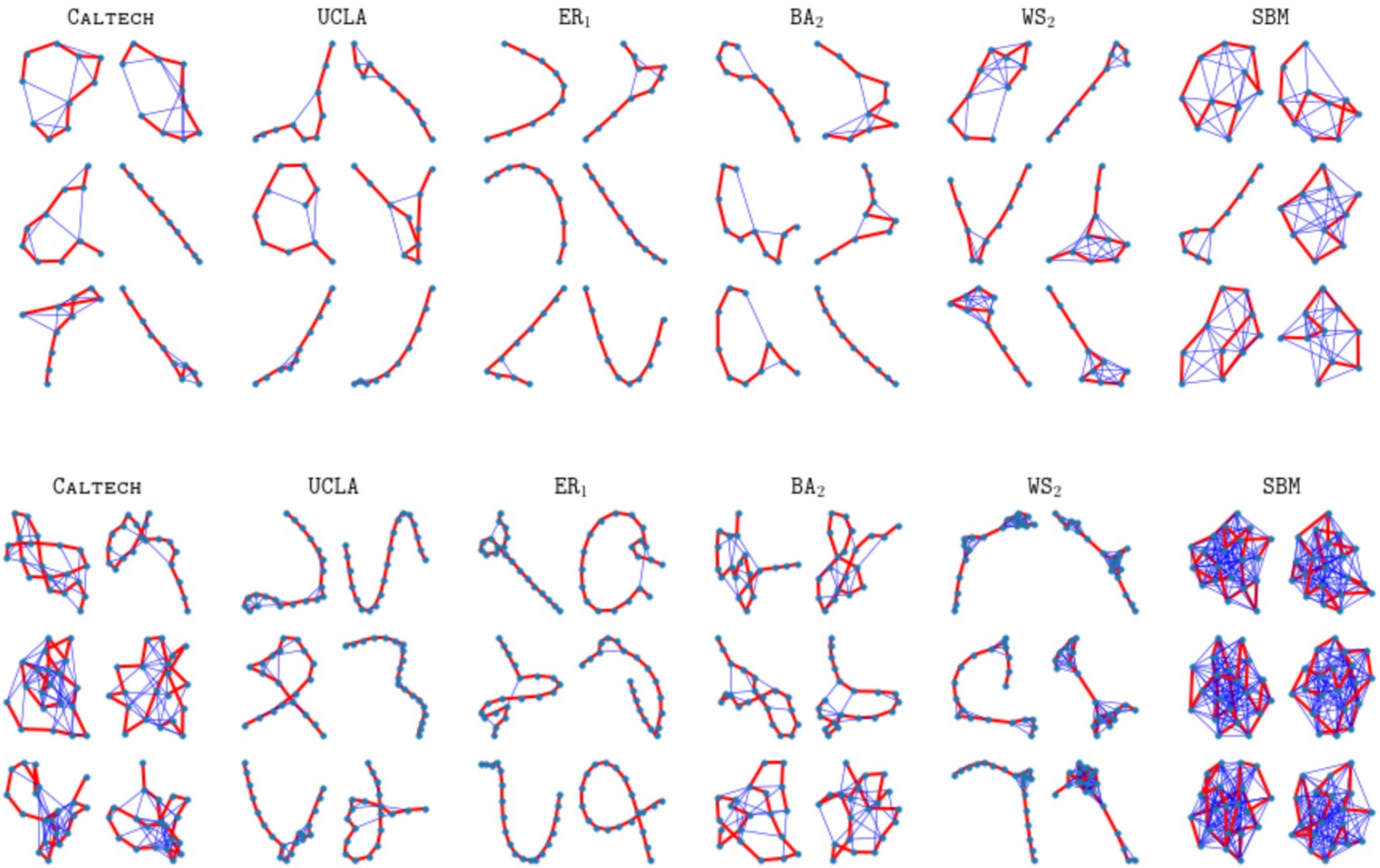
$k = 10$



Random graph homomorphism sampling



Random graph homomorphism sampling



Random graph homomorphism sampling

1. Sample a k-path P *uniformly at random*

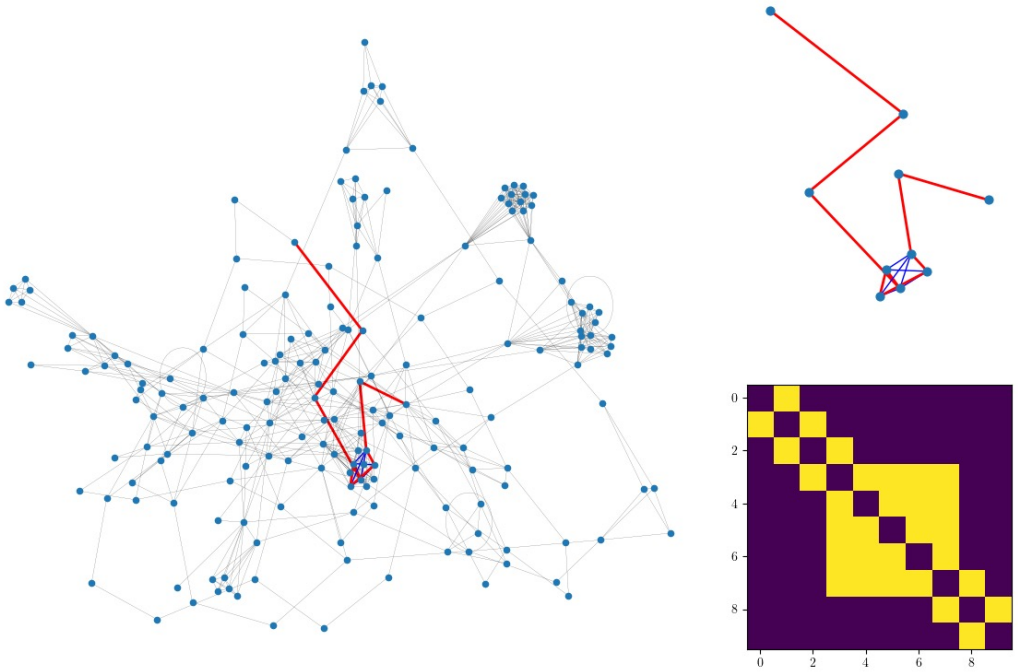
Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$

Naïve approach:

A) Sample k nodes $x_1, x_2, \ldots, x_k$ uniformly at random

B) If $x_1, x_2, \ldots, x_k$ forms a path, done

C) Otherwise, go back to A)



Random graph homomorphism sampling

1. Sample a k-path P *uniformly at random*

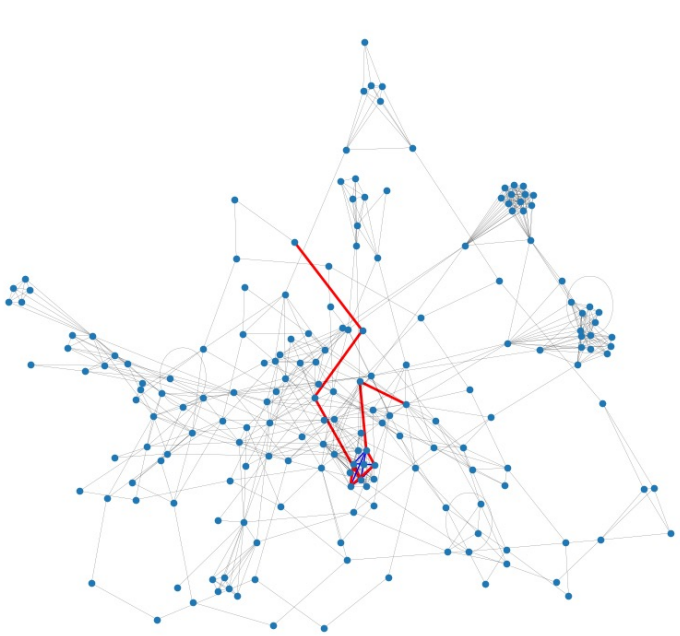
Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$

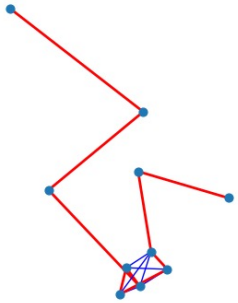
Naïve approach:

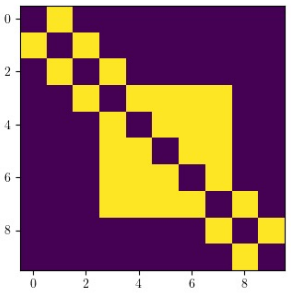
A) Sample k nodes $x_1, x_2, \ldots, x_k$ uniformly at random

B) **If $x_1, x_2, \ldots, x_k$ forms a path, done** ← For sparse networks, this occurs with probability ≈ 0

C) Otherwise, reject and go back to A)







Random graph homomorphism sampling

1. Sample a k-path P *uniformly at random*

Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$

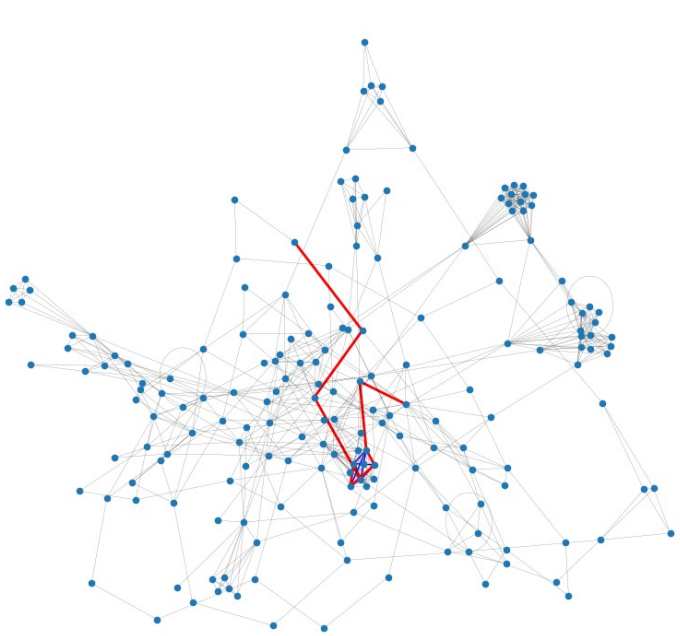
Nodes may overlap

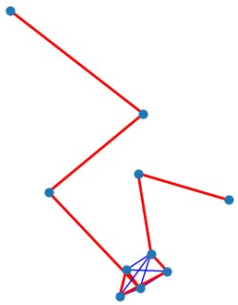
Our approach:

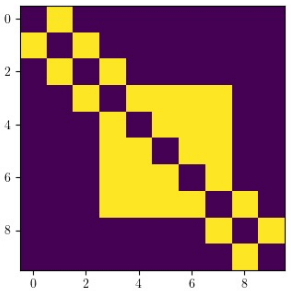
A) Sample a sequence of **k-walks** $(\mathbf{x}_t)_{t\geq 0}$ using MCMC sampling alg.

B) If $\mathbf{x}_t = (x_1, x_2, \dots, x_k)$ forms a path, done

C) Otherwise, go back to A)







Random graph homomorphism sampling

1. Sample a k-path P *uniformly at random*

Uniformly random injective graph homomorphism $\varphi : P_k \hookrightarrow G$

Nodes may overlap

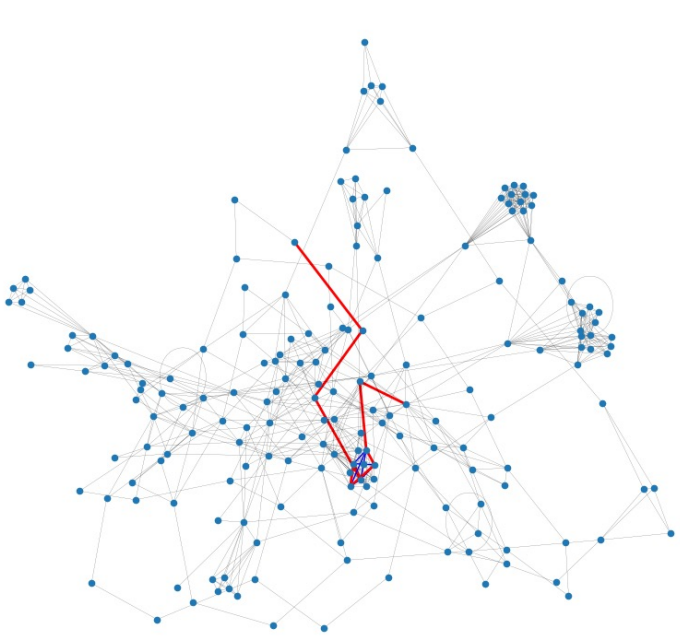
Our approach:

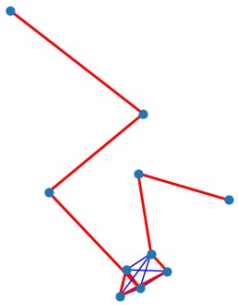
A) Sample a sequence of **k-walks** $(\mathbf{x}_t)_{t\geq 0}$ using MCMC sampling alg.

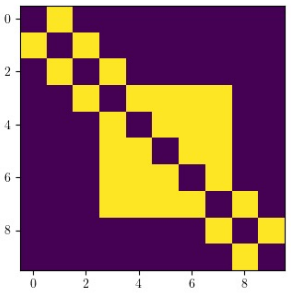
B) If $\mathbf{x}_t = (x_1, x_2, \dots, x_k)$ forms a path, done

Already forms a k-walk; additionally needs to be a k-path

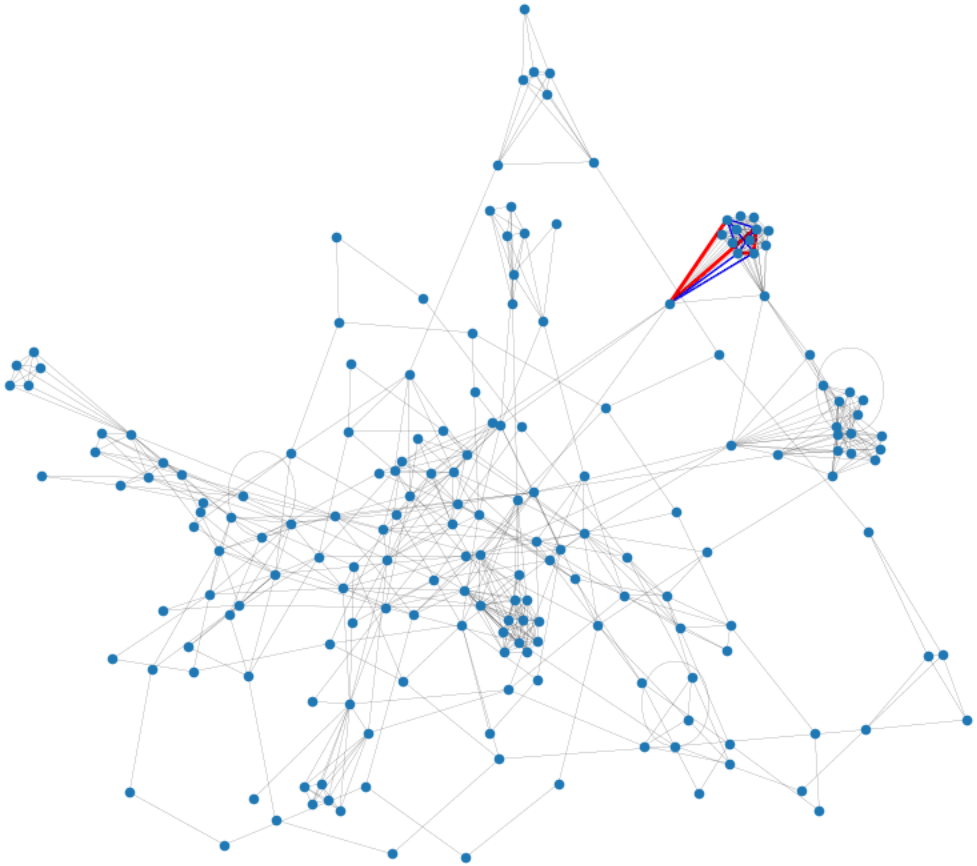
C) Otherwise, go back to A)

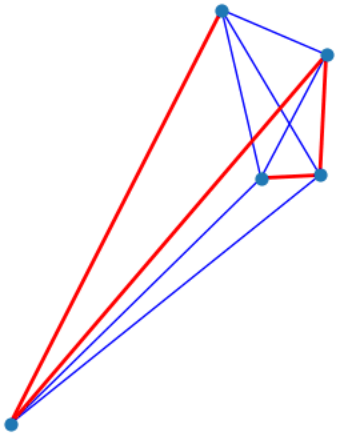


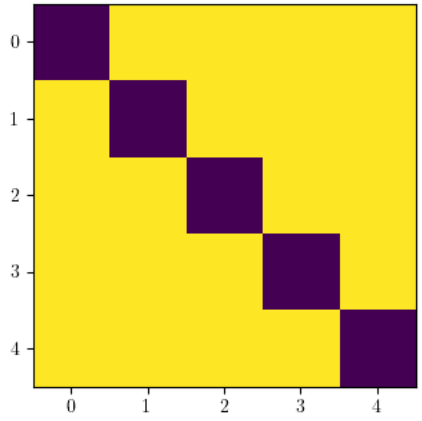




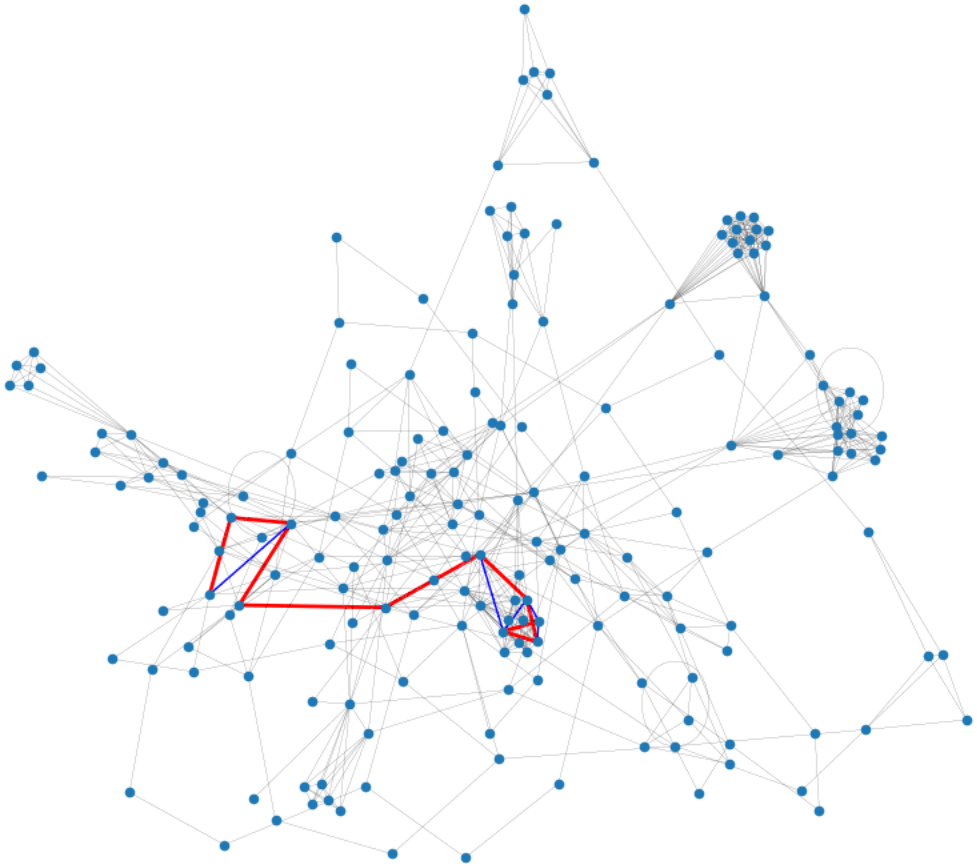
$k = 5$

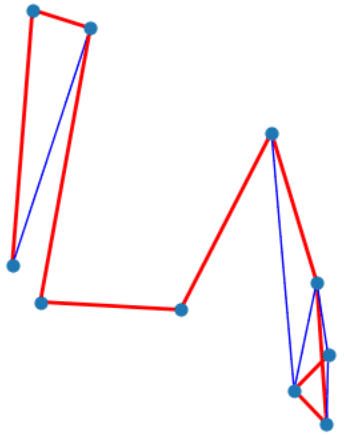


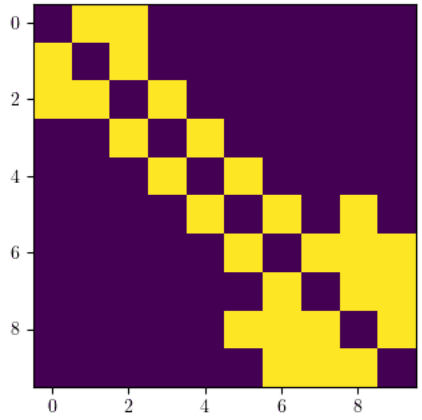




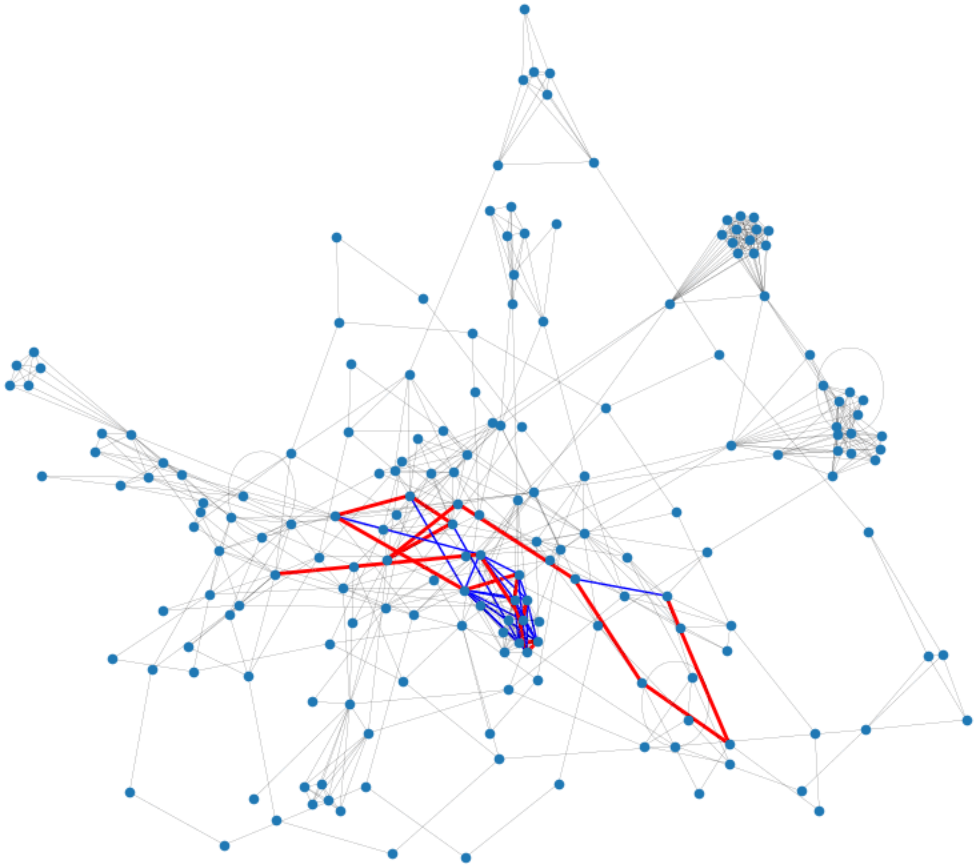
$k = 10$

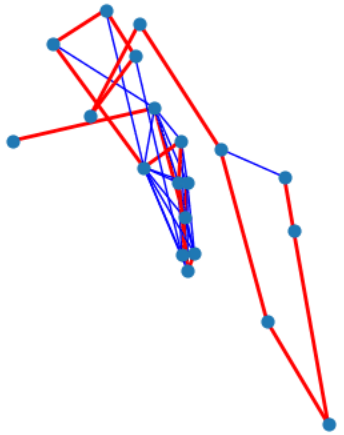


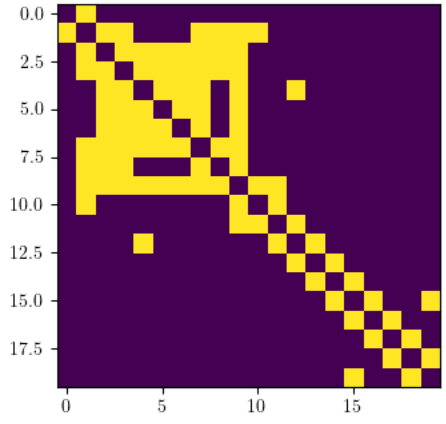




$k = 20$







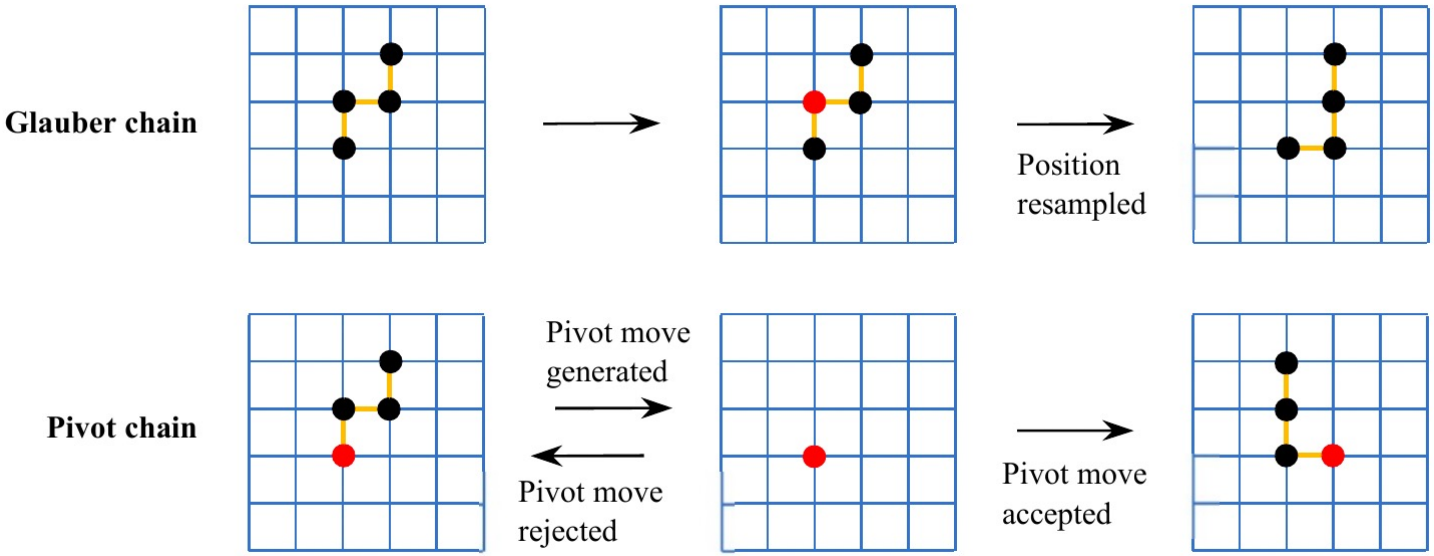
MCMC motif sampling convergence guarantee

Theorem. (Memoli, L., Sivakoff '20+)

If G is *non-bipartite*, then the MCMC motif sampling algorithm converges to the uniform distribution over all k-walks in G exponentially fast.

What exponent? – May grow in $|G|$

Additional "Mixing time" results obtained



Network Reconstruction

Network Reconstruction Algorithm

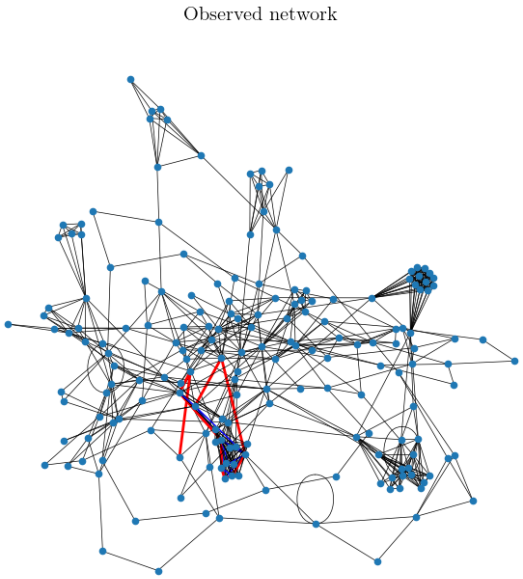
Input: Observed graph G; A low-rank approximation oracle $\mathcal{R}$ for $k \times k$ matrices

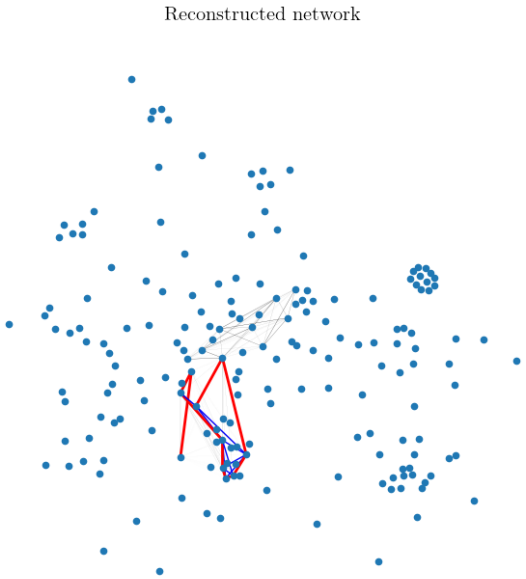
Do: $G_{\text{recons}} \leftarrow (V(G),\ \emptyset)$

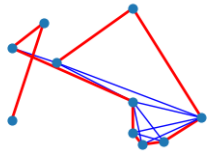
Repeat:

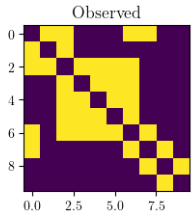
A) Sample a k-path $\mathbf{x} \subseteq G$ uniformly at random;

B) $A_{\mathbf{x}} \leftarrow$ $k \times k$ binary matrix of induced subgraph on $\mathbf{x}$









Network Reconstruction Algorithm

Input: Observed graph G; A low-rank approximation oracle $\mathcal{R}$ for $k \times k$ matrices

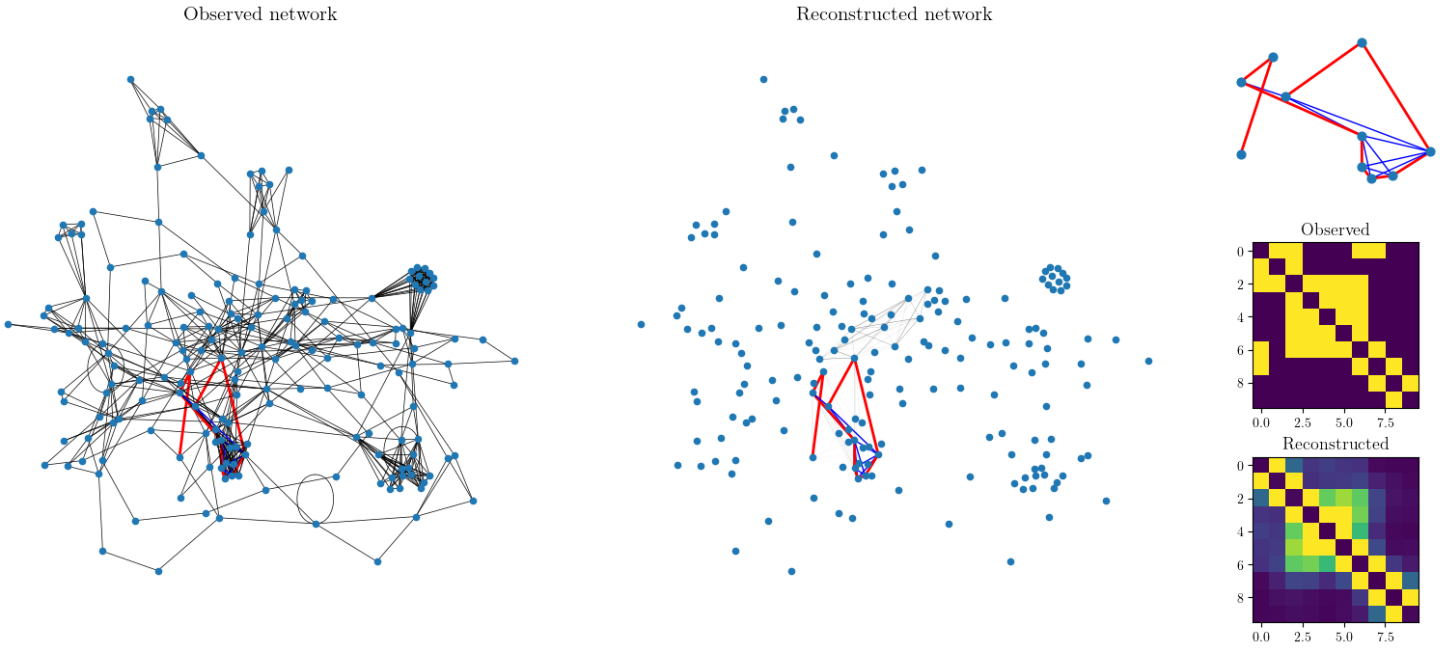
Do: $G_{\text{recons}} \leftarrow (V(G),\ \emptyset)$

Repeat:

A) Sample a k-path $\mathbf{x} \subseteq G$ uniformly at random;

B) $A_{\mathbf{x}} \leftarrow$ $k \times k$ binary matrix of induced subgraph on $\mathbf{x}$

C) $\hat{A}_{\mathbf{x}} \leftarrow \mathcal{R}(A)$: Low-rank approximation of $A_{\mathbf{x}}$



Network Reconstruction Algorithm

Input: Observed graph G; A low-rank approximation oracle $\mathcal{R}$ for $k \times k$ matrices

Do: $G_{\text{recons}} \leftarrow (V(G),\ \emptyset)$

Repeat:

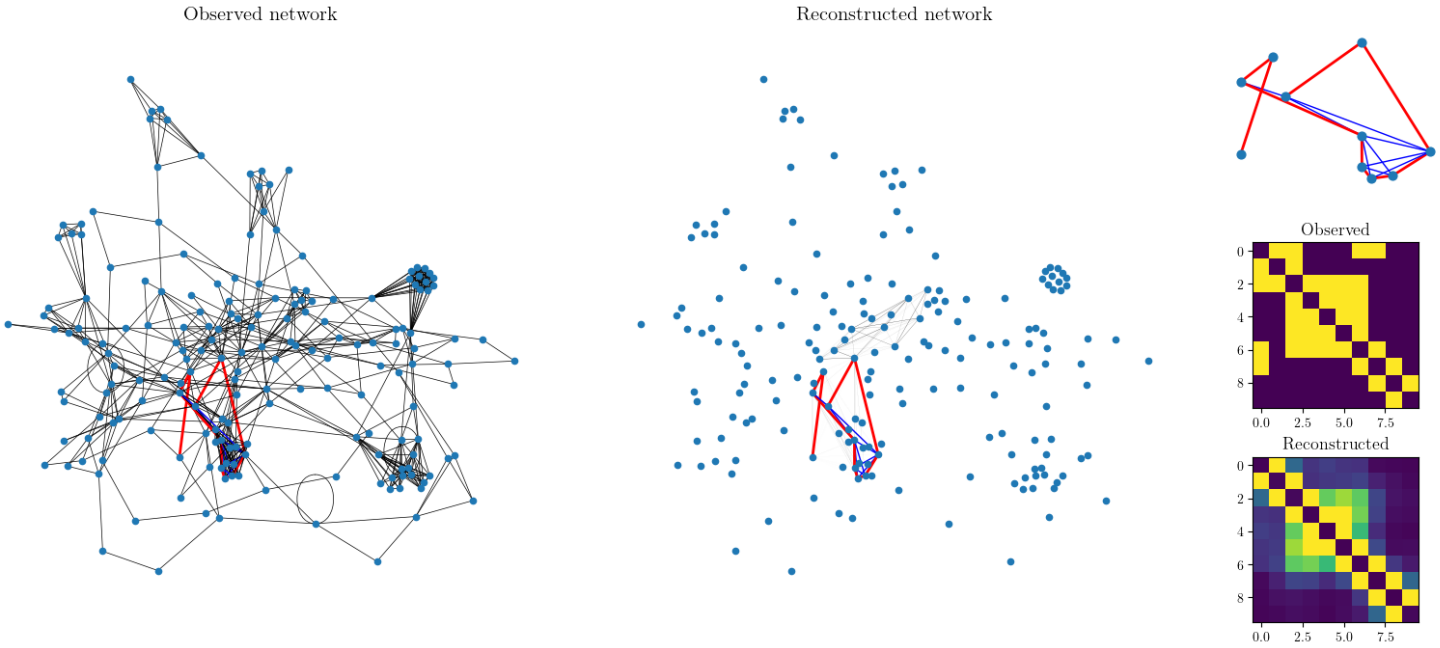
A) Sample a k-path $\mathbf{x} \subseteq G$ uniformly at random;

B) $A_{\mathbf{x}} \leftarrow$ $k \times k$ binary matrix of induced subgraph on $\mathbf{x}$

C) $\hat{A}_{\mathbf{x}} \leftarrow \mathcal{R}(A)$: Low-rank approximation of $A_{\mathbf{x}}$

D) Add in the edge weights in $\hat{A}_{\mathbf{x}}$ between nodes of $\mathbf{x}$ in G_{recons}

(edge weights are normalized at the end or recursively)



Network Reconstruction Algorithm

Input: Observed graph G; A low-rank approximation oracle $\mathcal{R}$ for $k \times k$ matrices

Do: $G_{\text{recons}} \leftarrow (V(G),\ \emptyset)$

Repeat:

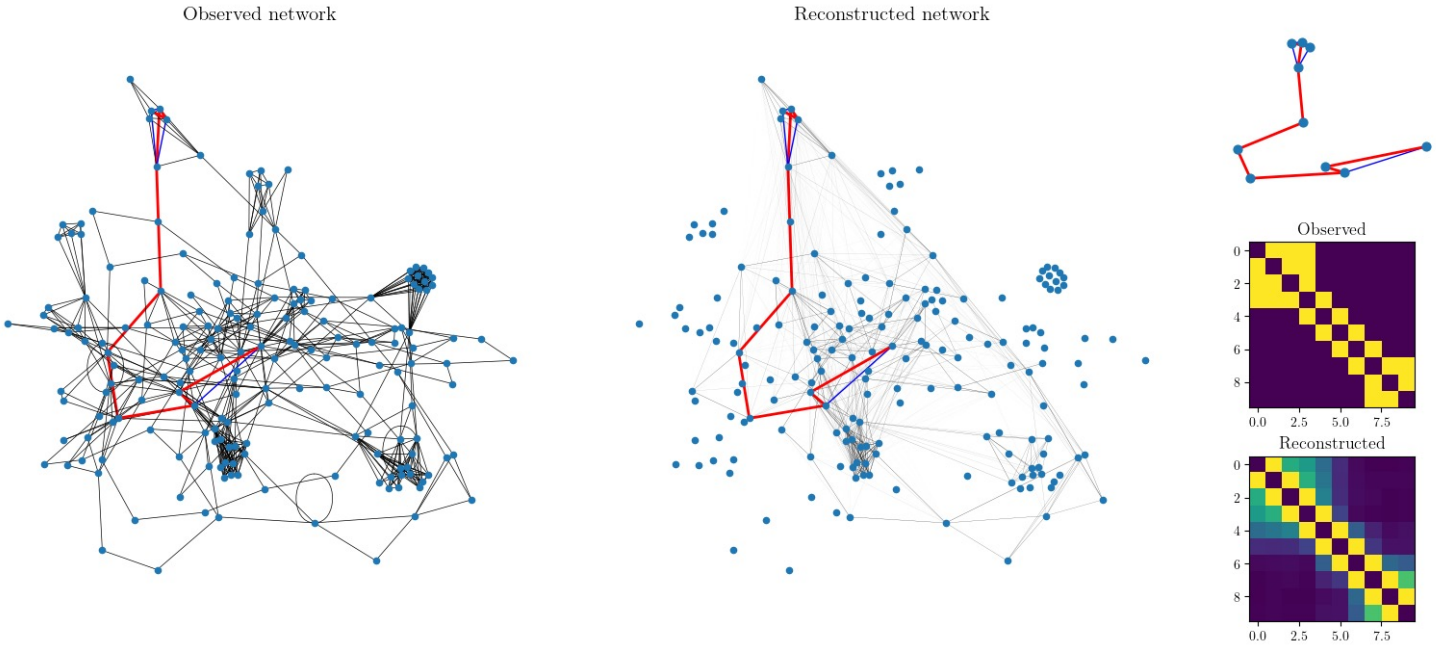
A) Sample a k-path $\mathbf{x} \subseteq G$ uniformly at random;

B) $A_{\mathbf{x}} \leftarrow$ $k \times k$ binary matrix of induced subgraph on $\mathbf{x}$

C) $\hat{A}_{\mathbf{x}} \leftarrow \mathcal{R}(A)$: Low-rank approximation of $A_{\mathbf{x}}$

D) Add in the edge weights in $\hat{A}_{\mathbf{x}}$ between nodes of $\mathbf{x}$ in G_{recons}

(edge weights are normalized at the end or recursively)



Network Reconstruction Algorithm

Input: Observed graph G; A low-rank approximation oracle $\mathcal{R}$ for $k \times k$ matrices

Do: $G_{\text{recons}} \leftarrow (V(G),\ \emptyset)$

Repeat:

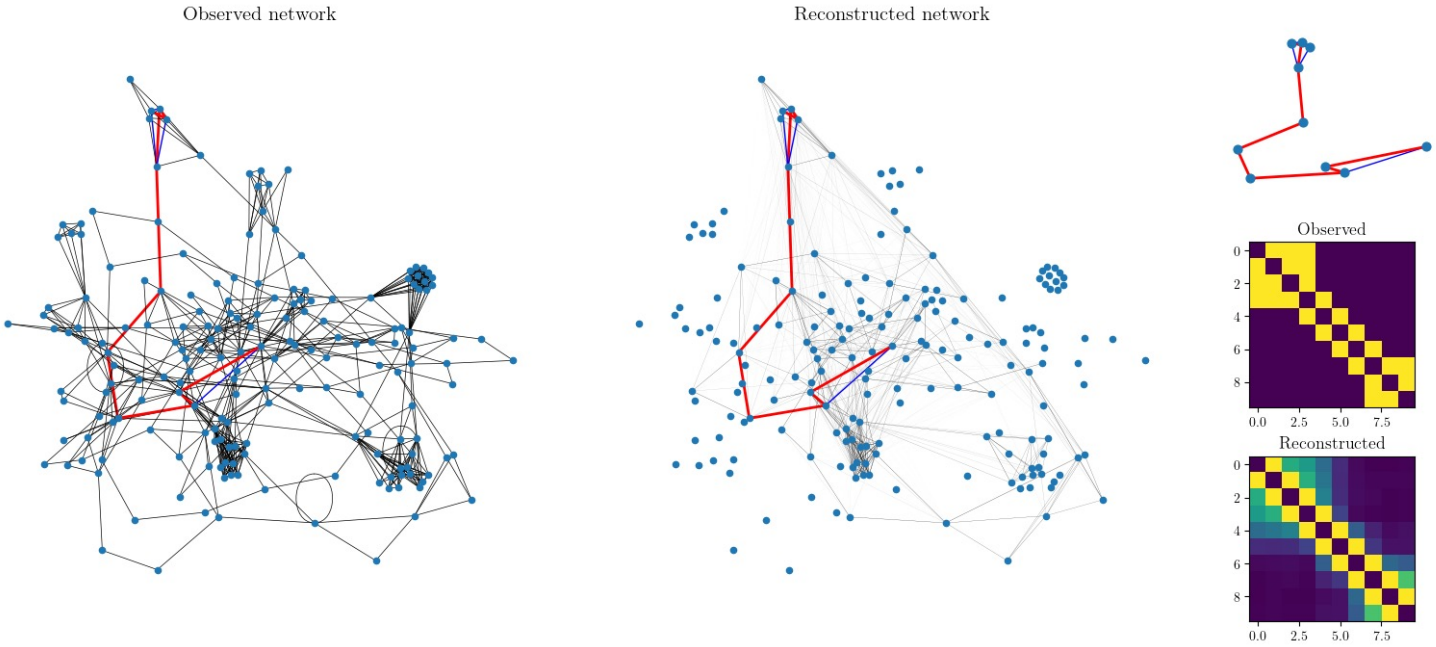
A) Sample a k-path $\mathbf{x} \subseteq G$ using the **MCMC motif-sampling alg**.

B) $A_{\mathbf{x}} \leftarrow$ $k \times k$ binary matrix of induced subgraph on $\mathbf{x}$

C) $\hat{A}_{\mathbf{x}} \leftarrow \mathcal{R}(A)$: Low-rank approximation of $A_{\mathbf{x}}$

D) Add in the edge weights in $\hat{A}_{\mathbf{x}}$ between nodes of $\mathbf{x}$ in G_{recons}

(edge weights are normalized at the end or recursively)



Network Reconstruction Algorithm

Input: Observed graph G; A low-rank approximation oracle $\mathcal{R}$ for $k \times k$ matrices

Do: $G_{\text{recons}} \leftarrow (V(G),\ \emptyset)$

Repeat:

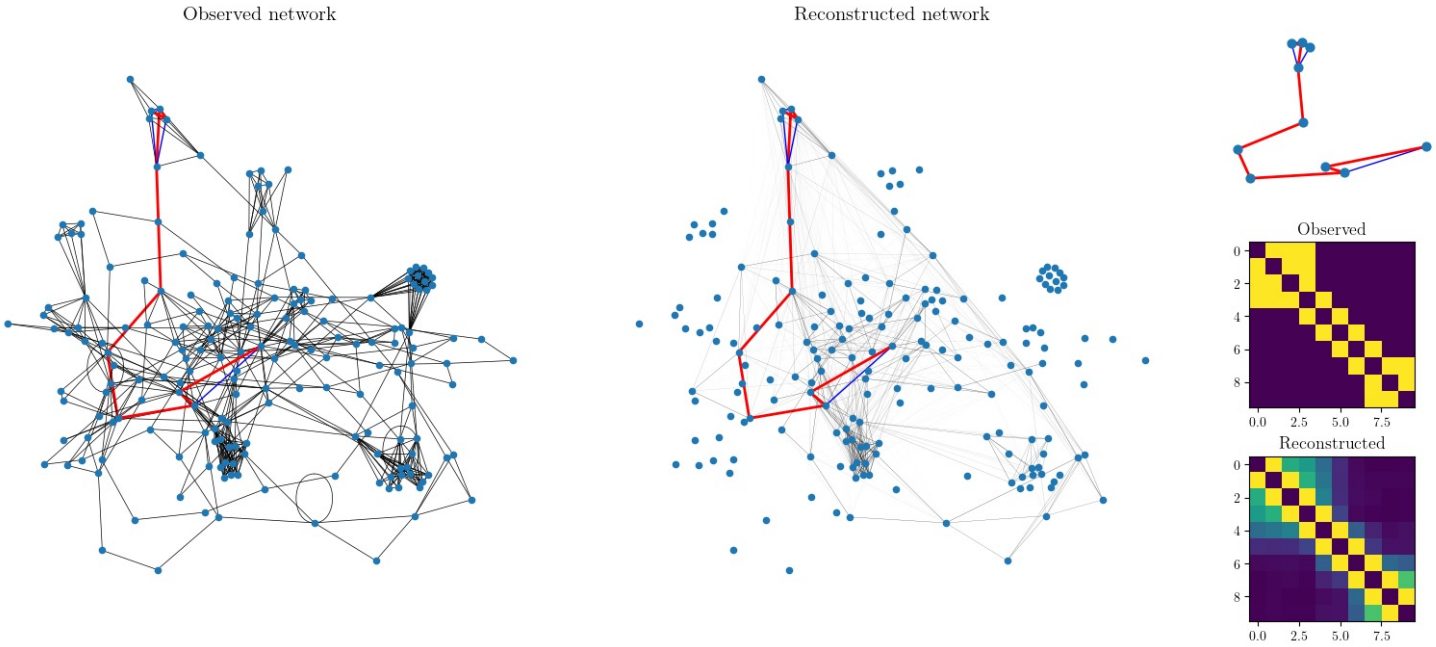
A) Sample a ***k*-walk** $\mathbf{x} \subseteq G$ using the MCMC motif-sampling alg.

B) $A_{\mathbf{x}} \leftarrow$ $k \times k$ binary matrix of induced subgraph on $\mathbf{x}$

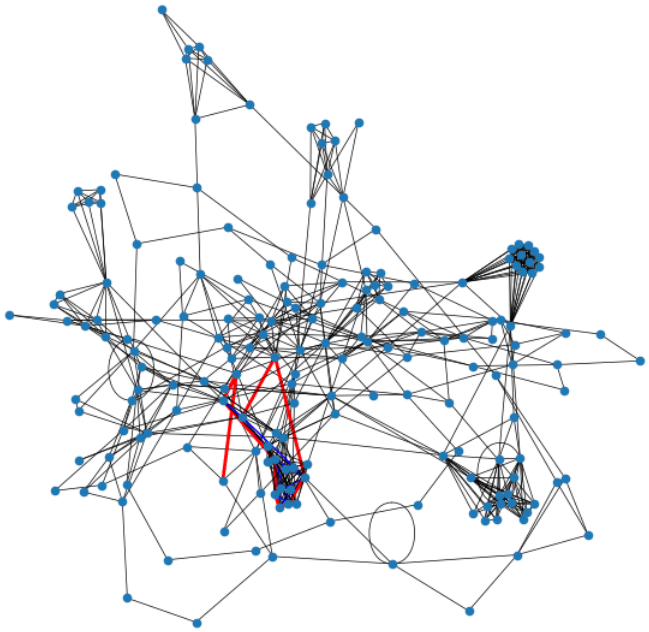
C) $\hat{A}_{\mathbf{x}} \leftarrow \mathcal{R}(A)$: Low-rank approximation of $A_{\mathbf{x}}$

D) Add in the edge weights in $\hat{A}_{\mathbf{x}}$ between nodes of $\mathbf{x}$ in G_{recons}

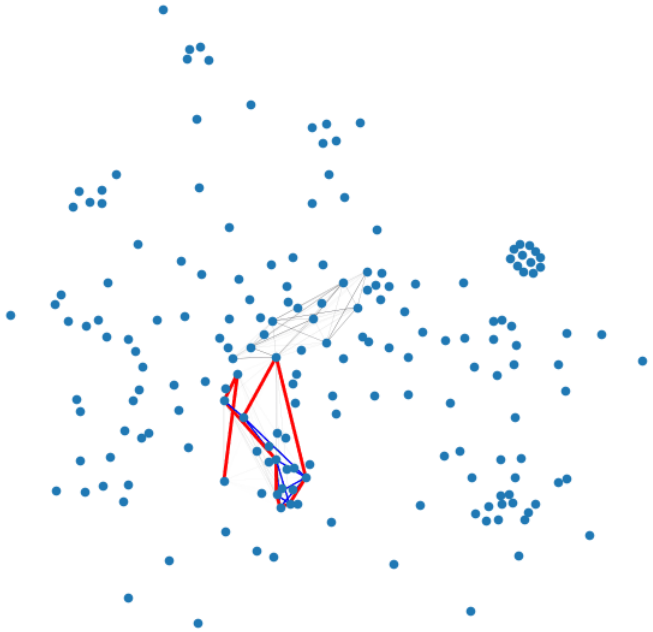
(edge weights are normalized at the end or recursively)

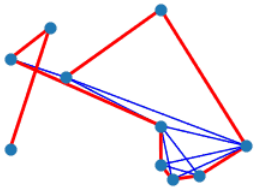


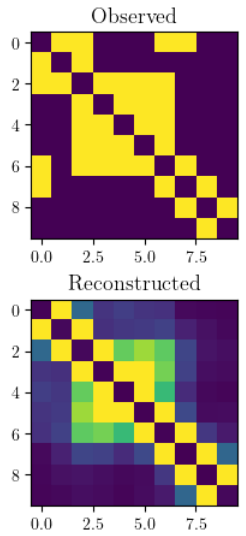
Observed network



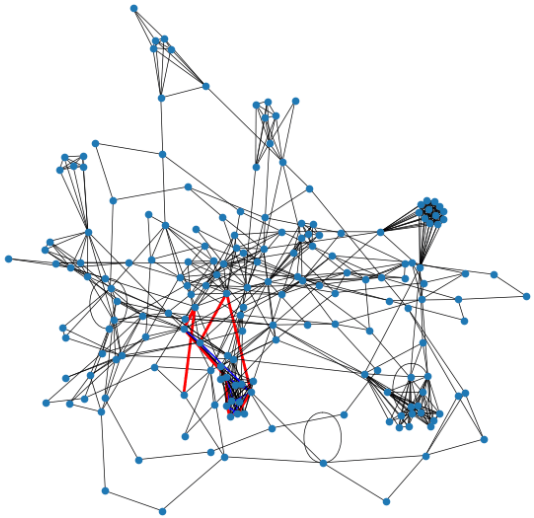
Reconstructed network



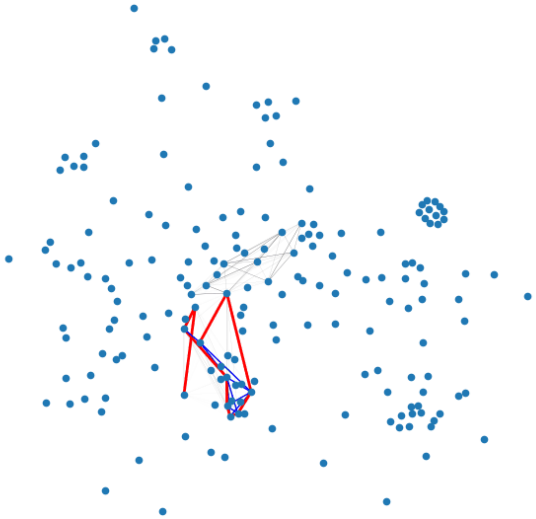


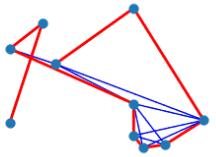


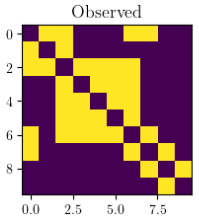
Observed network

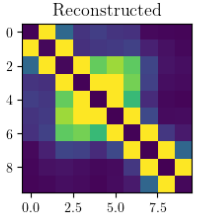


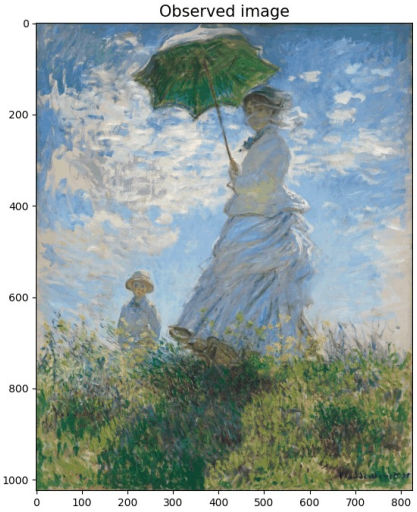
Reconstructed network

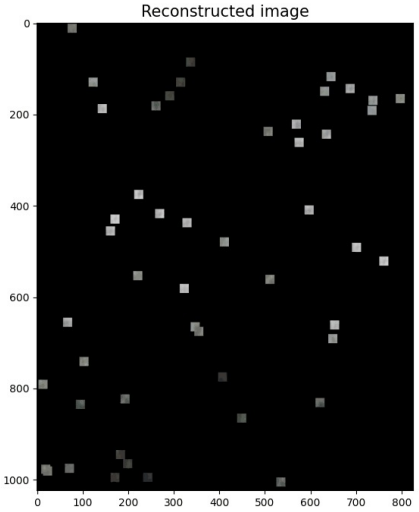


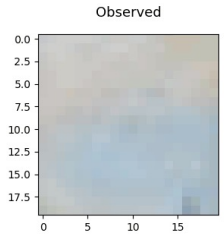


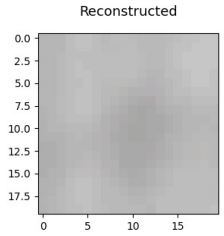












Network Reconstruction Guarantee

Theorem. (Ntwk. Recons. Err Bd) **(L., Kureh, Vendrow, Porter '22+)**

Given an observed network $G = (V, A)$, (wtd) Adj. mx

Network reconstruction alg. → limiting reconstruction network $G_{\text{recons}} = (V, \hat{A})$.

Furthermore,

$$\text{JaccardDistance}(G, G_{\text{recons}}) \leq \frac{1}{2(k-1)} \mathbb{E}_{\mathbf{x}} \left[\left\| A_{\mathbf{x}} - \hat{A}_{\mathbf{x}} \right\|_1 \right].$$

$\frac{\|A - \hat{A}\|}{\|A \vee \hat{A}\|}$

Mesoscale parameter

Low-rank mesoscale reconstruction error

Image Reconstruction Guarantee

Theorem. (Img. Recons. Err Bd) **(L. '22+)**

Given an image $\boldsymbol{A} \in \mathbb{R}^{a \times b \times 3}$

Image reconstruction alg. → limiting reconstruction image $\widehat{\boldsymbol{A}}$

Furthermore,

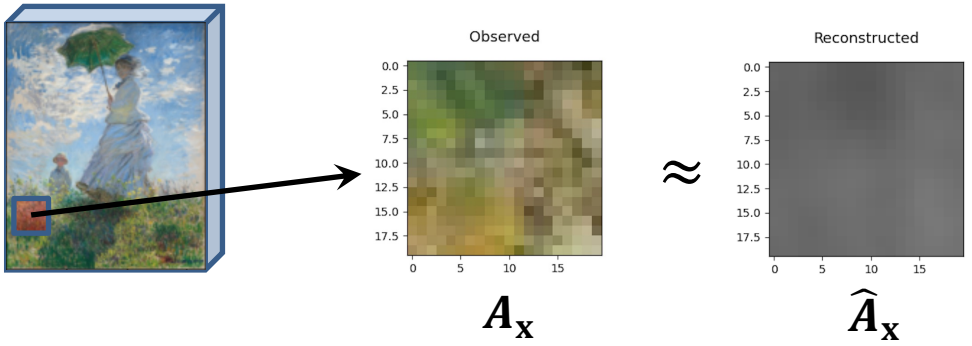
$$\text{JaccardDistance}\left(\boldsymbol{A}, \widehat{\boldsymbol{A}}\right) \leq \frac{1}{(k-1)^2} \, \mathbb{E}_{\mathbf{x}} \left[\left\| \boldsymbol{A}_{\mathbf{x}} - \widehat{\boldsymbol{A}}_{\mathbf{x}} \right\|_1 \right].$$

$\frac{\|A - \hat{A}\|}{\|A \vee \hat{A}\|}$

Mesoscale parameter

Low-rank mesoscale reconstruction error

$\mathbf{x}$: Unif. random $k \times k$ window



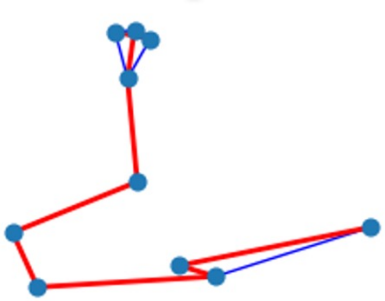
Network Dictionary Learning

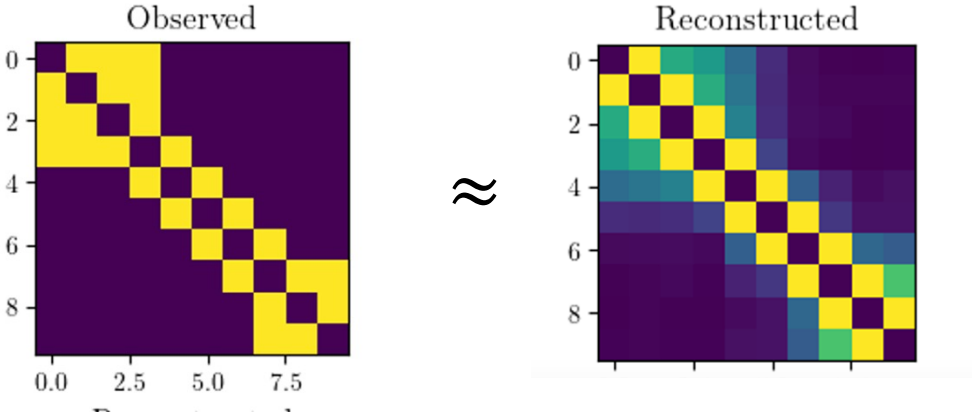
Uniformly random k-path

Theorem. $\text{JaccardDistance}(G, G_{\text{recons}}) \leq \frac{1}{2(k-1)} \mathbb{E}_{\mathbf{x}} \left[\left\| A_{\mathbf{x}} - \hat{A}_{\mathbf{x}} \right\|_1 \right]$

Sampled $k \times k$ subgraph adj. mx

Rank-r approx. of $A_{\mathbf{x}}$



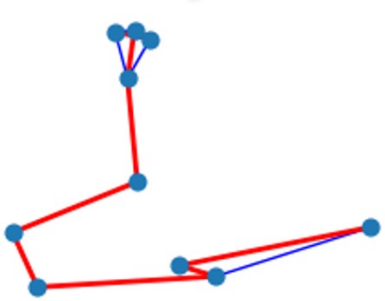


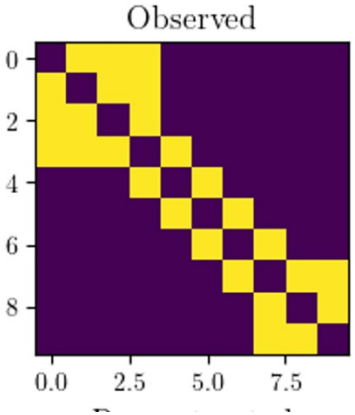
Uniformly random k-path

Theorem. $\text{JaccardDistance}(G, G_{\text{recons}}) \leq \frac{1}{2(k-1)} \mathbb{E}_{\mathbf{x}} \left[\| A_{\mathbf{x}} - \hat{A}_{\mathbf{x}} \|_1 \right]$

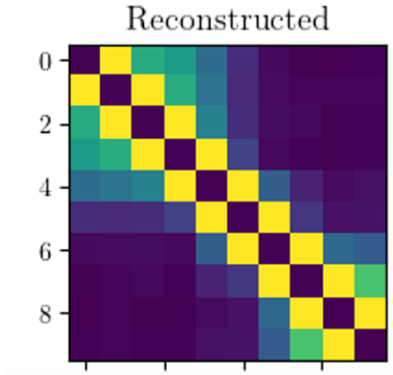
Sampled $k \times k$ subgraph adj. mx

Rank-r approx. of $A_{\mathbf{x}}$





$\approx$



Rank-r basis for k-node subgraph adj. mxs.

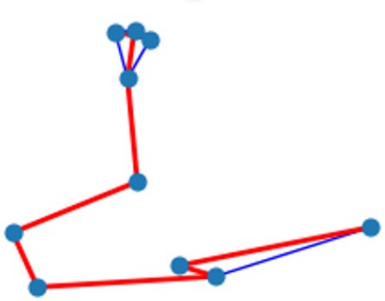
$= a_1 L_1 + \cdots + a_r L_r$

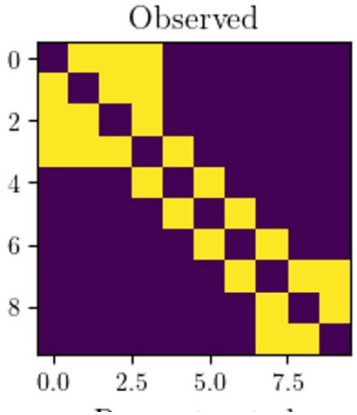
Uniformly random k-path

Theorem. $\text{JaccardDistance}(G, G_{\text{recons}}) \leq \frac{1}{2(k-1)} \mathbb{E}_{\mathbf{x}} \left[\| A_{\mathbf{x}} - \hat{A}_{\mathbf{x}} \|_1 \right]$

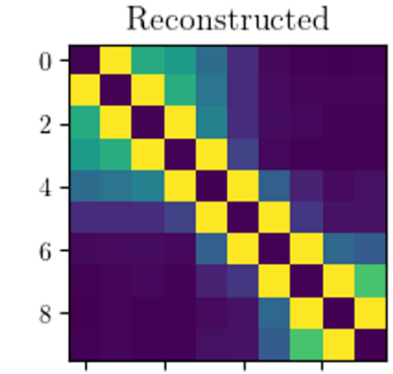
Sampled $k \times k$ subgraph adj. mx

Rank-r approx. of $A_{\mathbf{x}}$





$\approx$

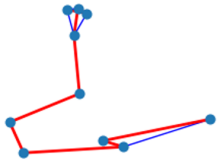


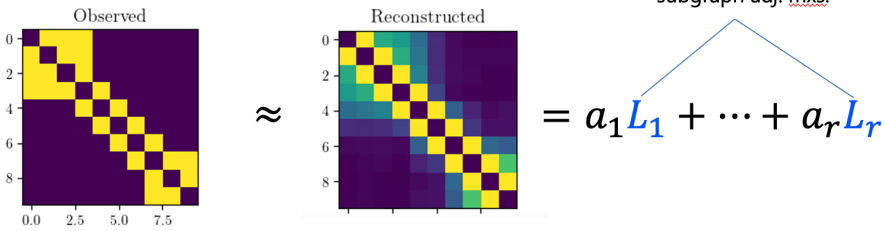
Rank-r basis for k-node subgraph adj. mxs.

$$= a_1 L_1 + \cdots + a_r L_r$$

Choose the basis subgraphs $L_1, \ldots L_r$ so that $\mathbb{E}_{\mathbf{x}} \left[\| A_{\mathbf{x}} - \hat{A}_{\mathbf{x}} \|_1 \right]$ is minimized!

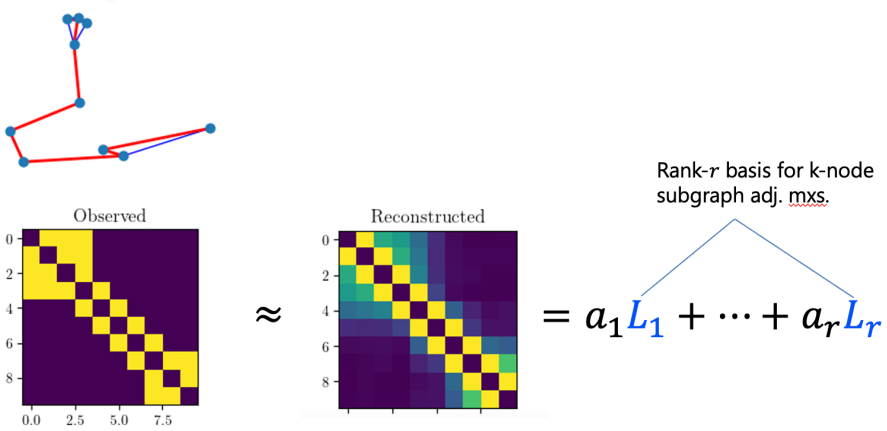
Network Dictionary Learning (NDL)





NDL Problem. $\min_{L_1,\ldots,L_r} \mathbb{E}_{\mathbf{x}}[\|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1]$

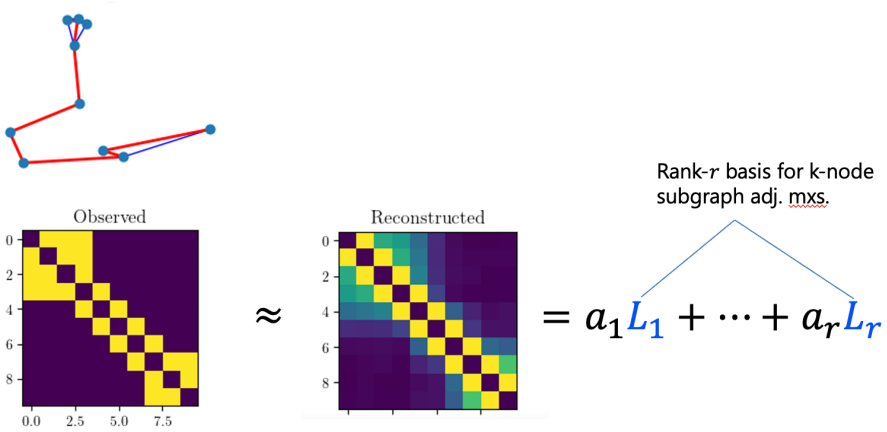
where $(a_1, \ldots, a_r) = \operatorname{argmin} \|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1$



NDL Problem. $\min_{L_1,\dots,L_r} \mathbb{E}_{\mathbf{x}}[\|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1]$

where $(a_1, \dots, a_r) = \operatorname{argmin} \|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1$

1. We want $L_1, \dots, L_r$ to be subgraph adj. mxs → Nonnegativity constraints



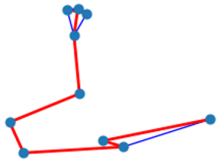
NDL Problem. $\min_{L_1,\dots,L_r} \mathbb{E}_{\mathbf{x}}[\|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1]$

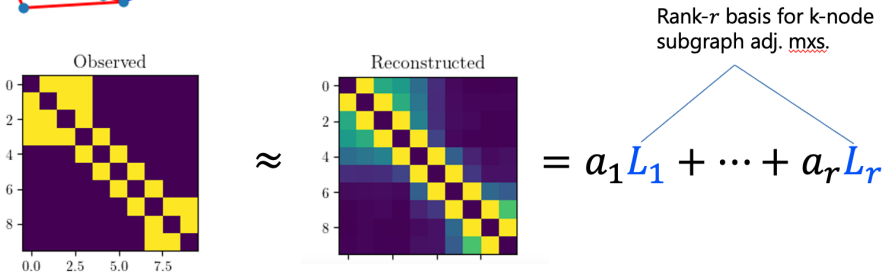
where $(a_1, \dots, a_r) = \operatorname{argmin} \|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1$

1. We want $L_1, \dots, L_r$ to be subgraph adj. mxs → Nonnegativity constraints

Stochastic nonconvex constrained problem

2. It is an online dictionary learning problem where data samples are obtained by MCMC k-path sampling algorithm on networks





NDL Problem. $\min_{L_1,\dots,L_r} \mathbb{E}_{\mathbf{x}}[\|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1]$

where $(a_1, \dots, a_r) = \operatorname{argmin} \|A_{\mathbf{x}} - (a_1 L_1 + \cdots + a_r L_r)\|_1$

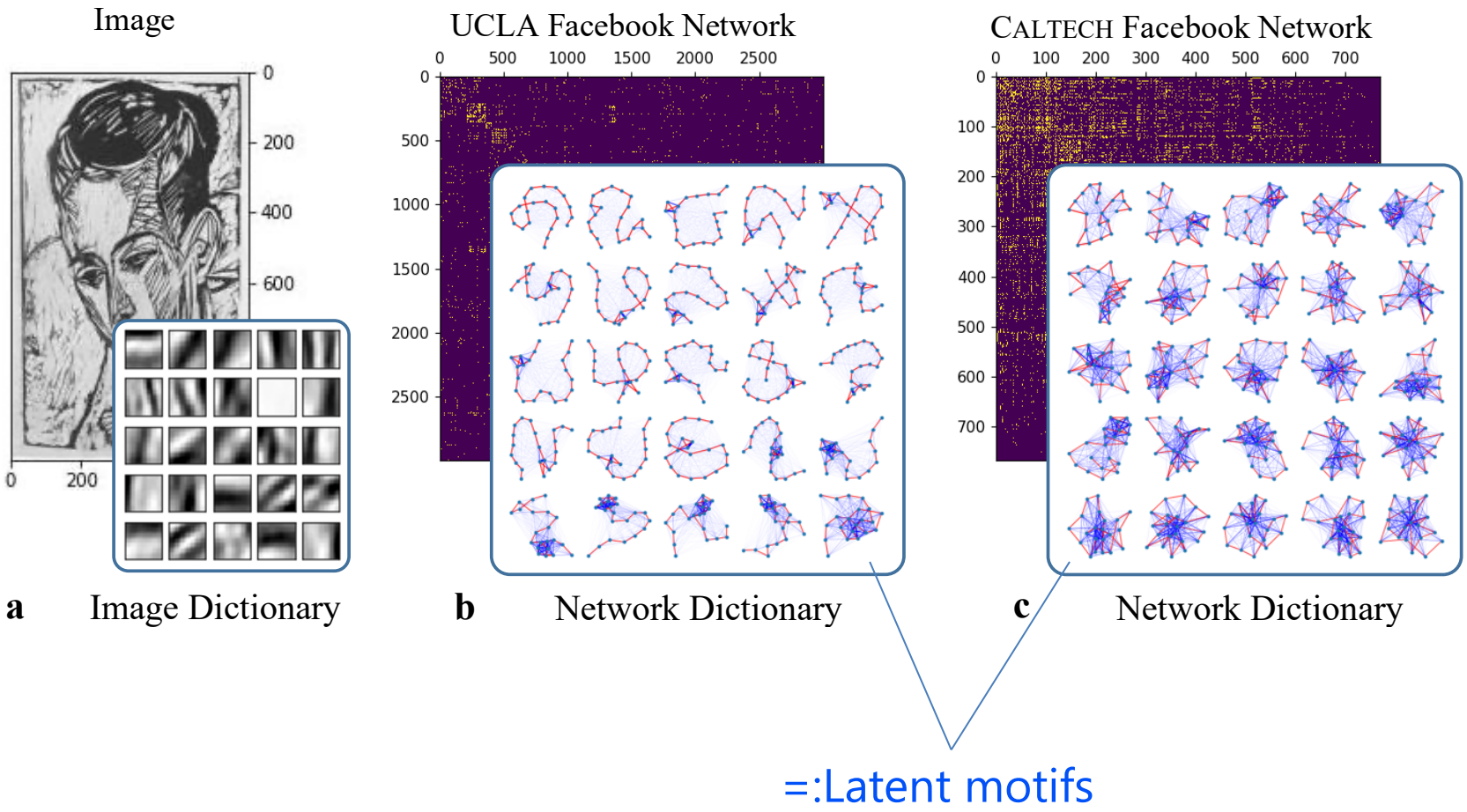
1. We want $L_1, \dots, L_r$ to be subgraph adj. mxs → Nonnegativity constraints

Stochastic nonconvex constrained problem

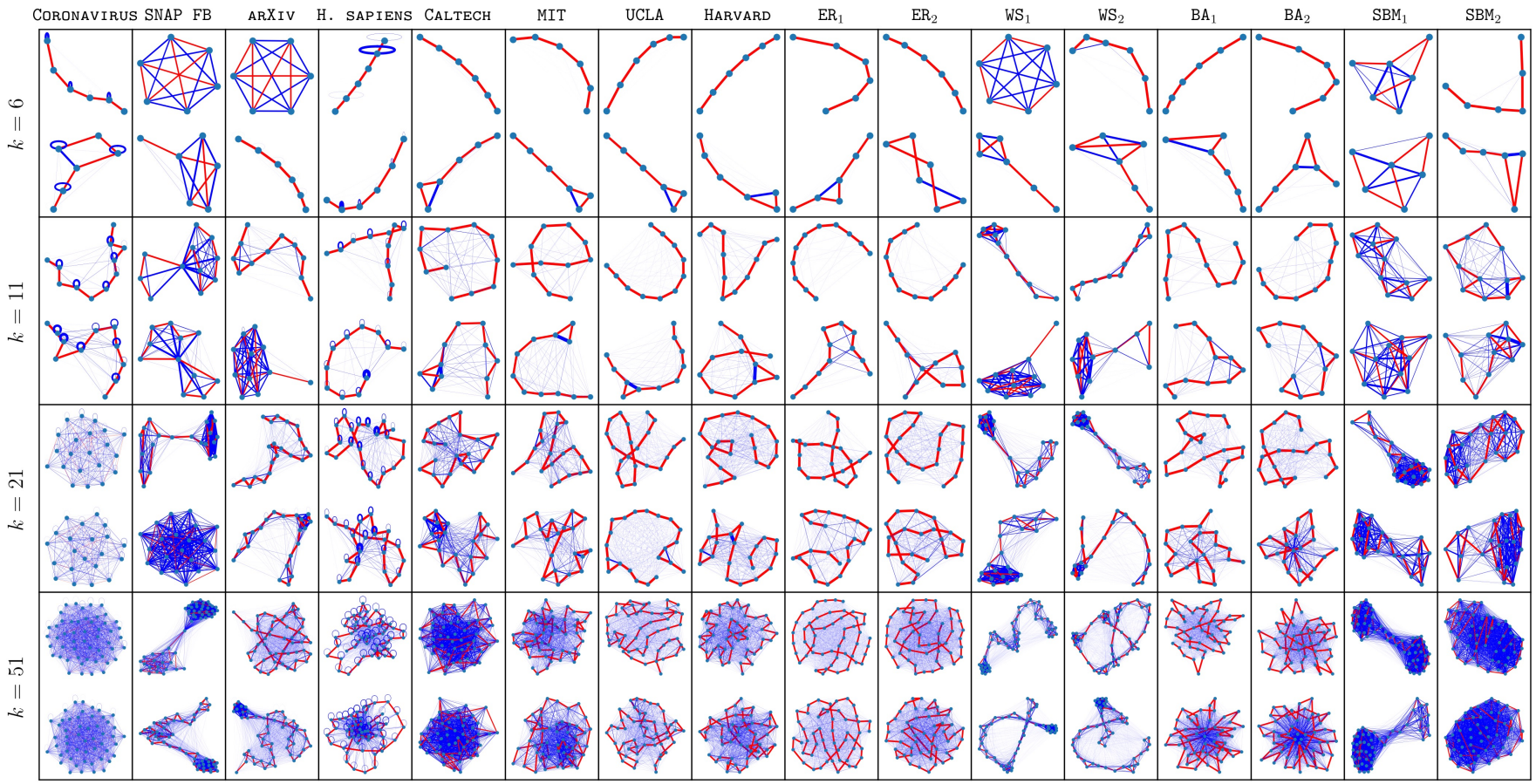
2. It is an online dictionary learning problem where data samples are obtained by MCMC k-path sampling algorithm on networks

→ Online Nonnegative Matrix Factorization with Markovian data

Network Dictionary Learning



Latent motifs at various mesoscales

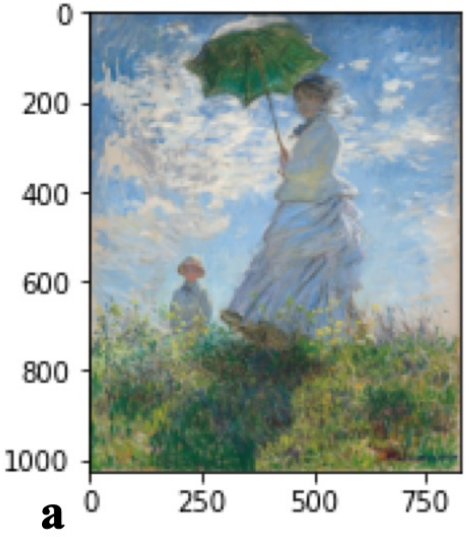


Latent motifs capture mesoscale structures at various mesoscales

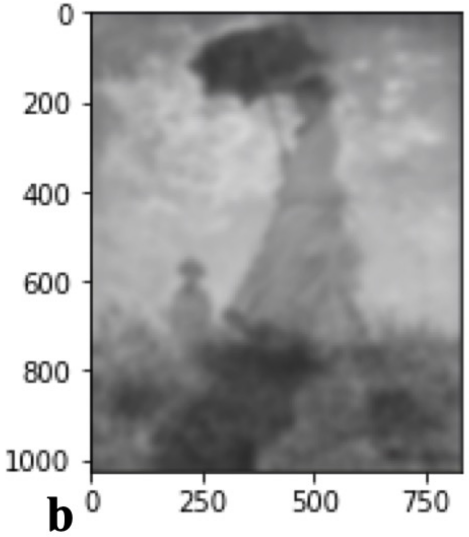
Applications of the new methods

Image cross-reconstruction examples

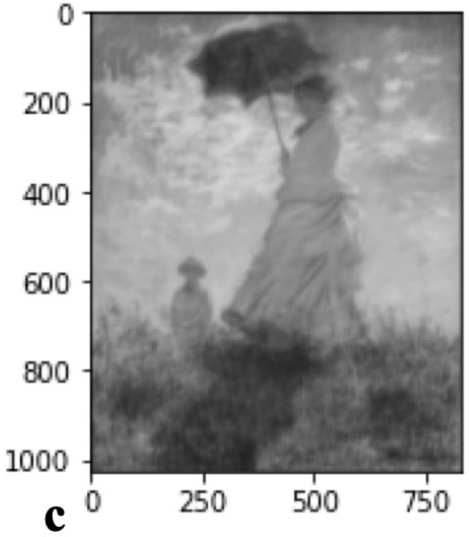
Original Img



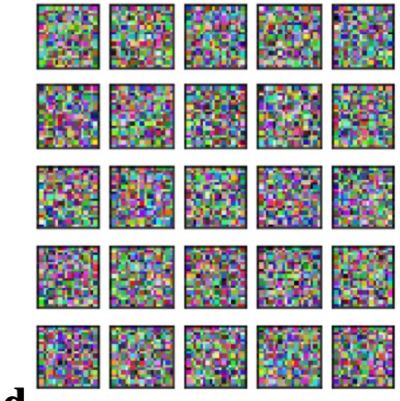
Reconstructed using **d**



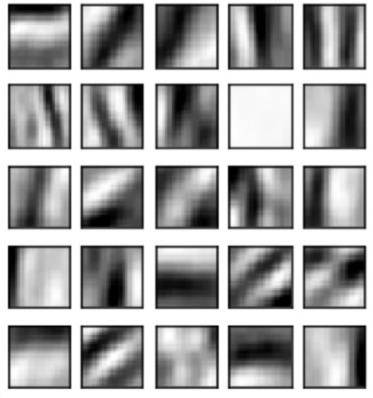
Reconstructed using **e**

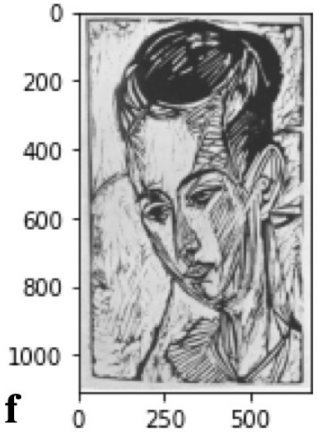


Random dictionary



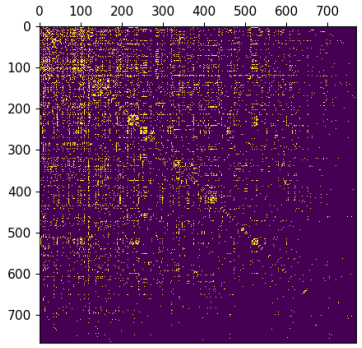
Dictionary learned from **f**



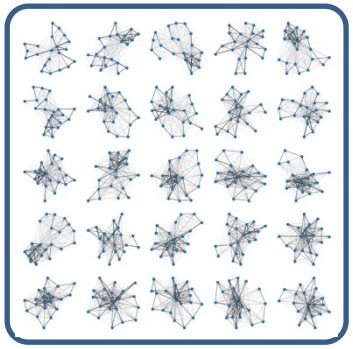


Network cross-reconstruction

Network Y

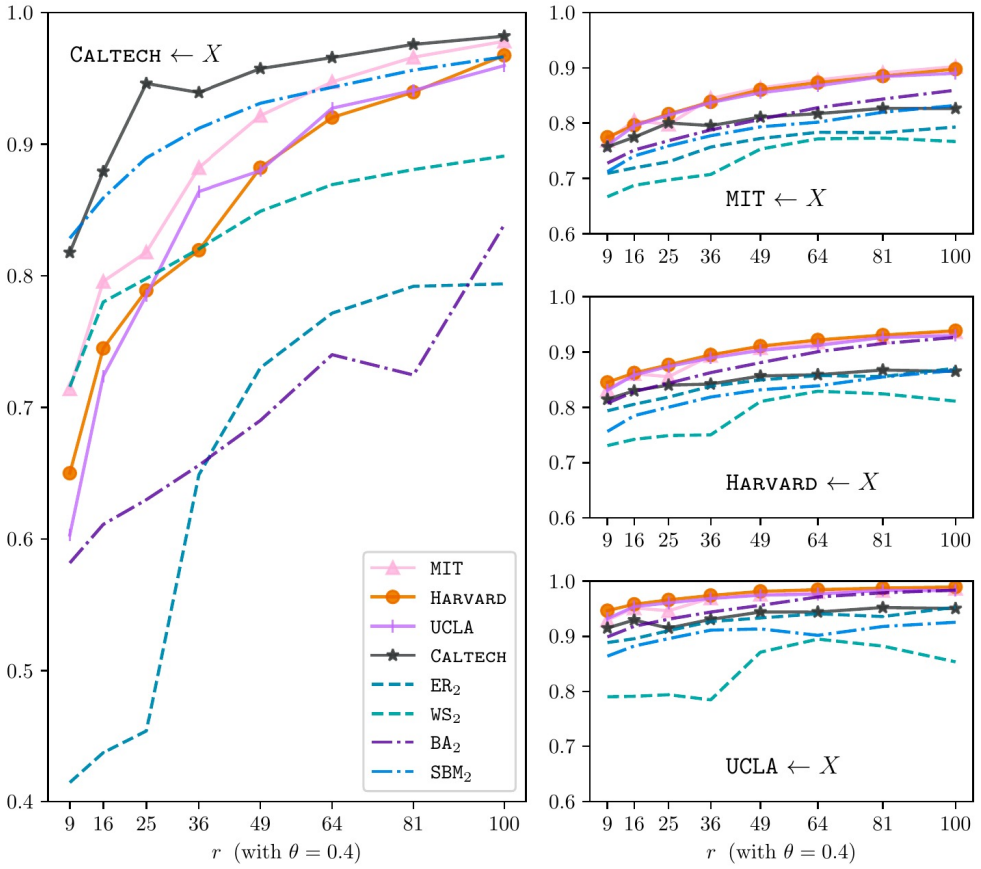


Network Reconstruction



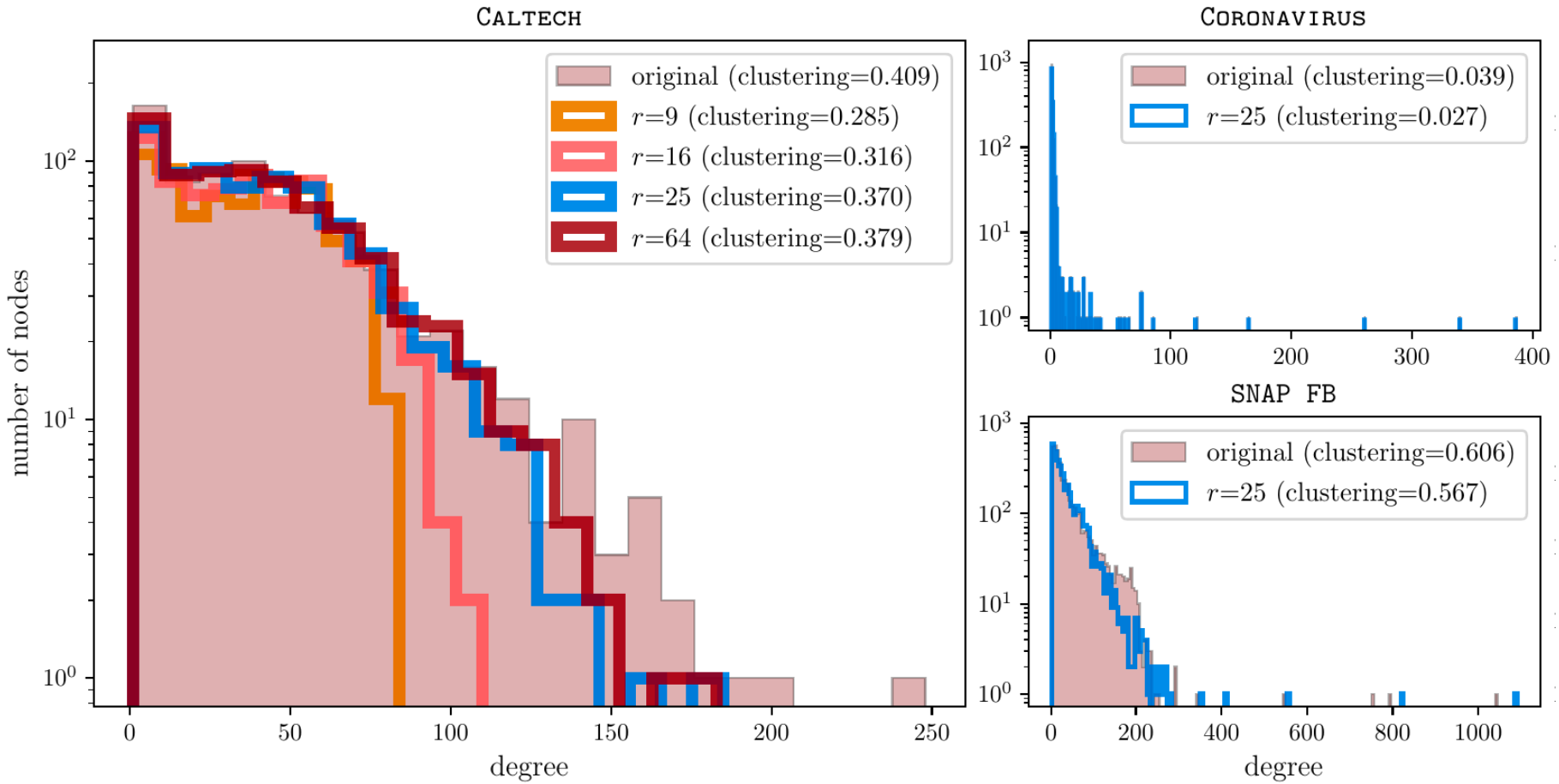
Dictionary learned from network X

Global Reconstruction Accuracy vs. Rank of mesoscale used



Social Networks can be well-reconstructed by low-rank mesoscale structures

Degree distribution in original and reconstructed networks

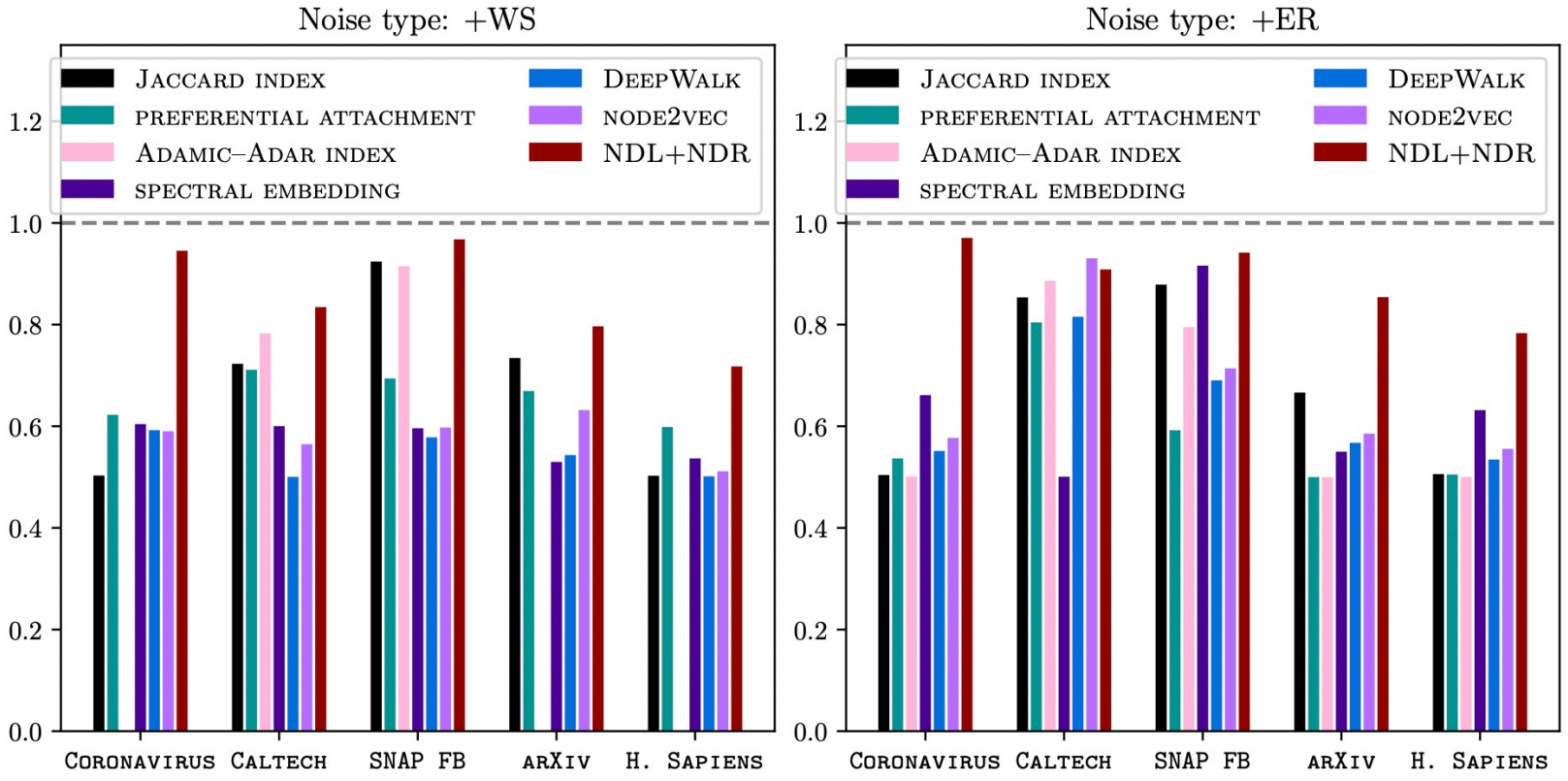


Network Reconstruction with low-rank mesoscale structure
→ Recovers a degree-truncated network

Network Denoising

Adding Watts-Strogatz edges

Adding uniformly random edges

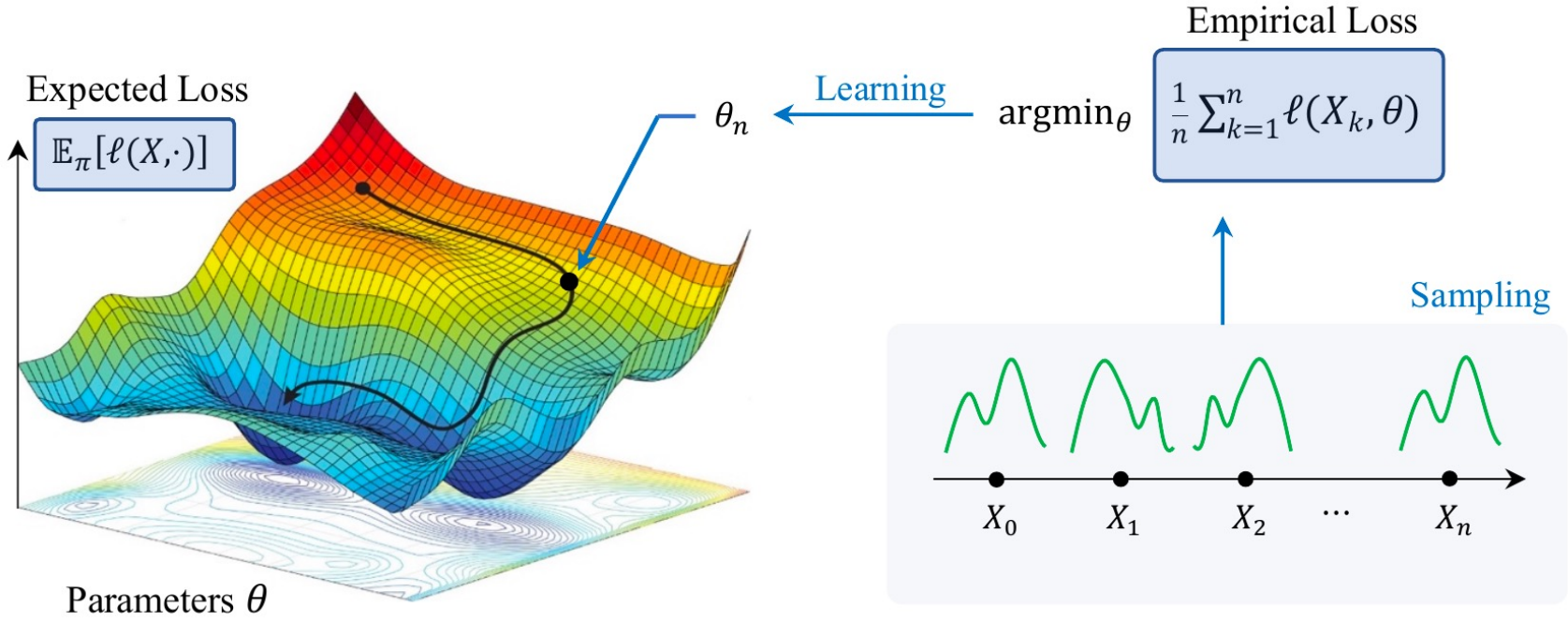


Our method (NDL+NDR) leverages mesoscale structures

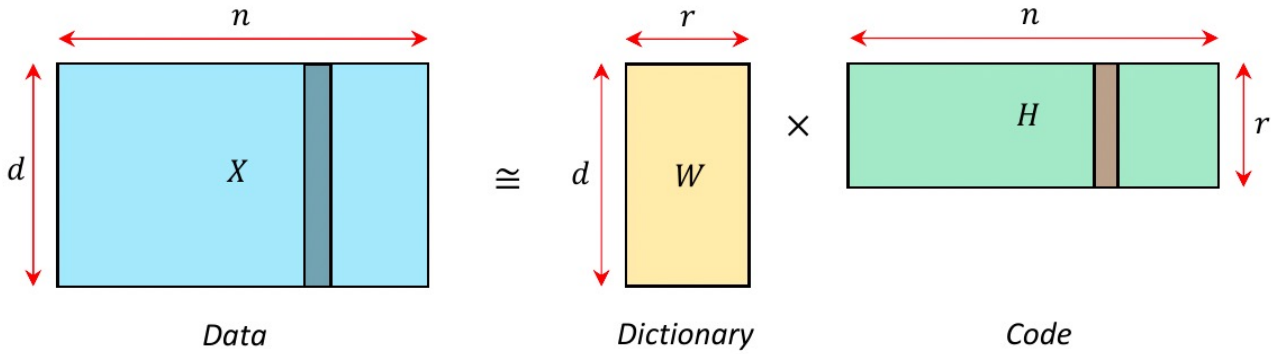
→ Performs well for denoising localized noise

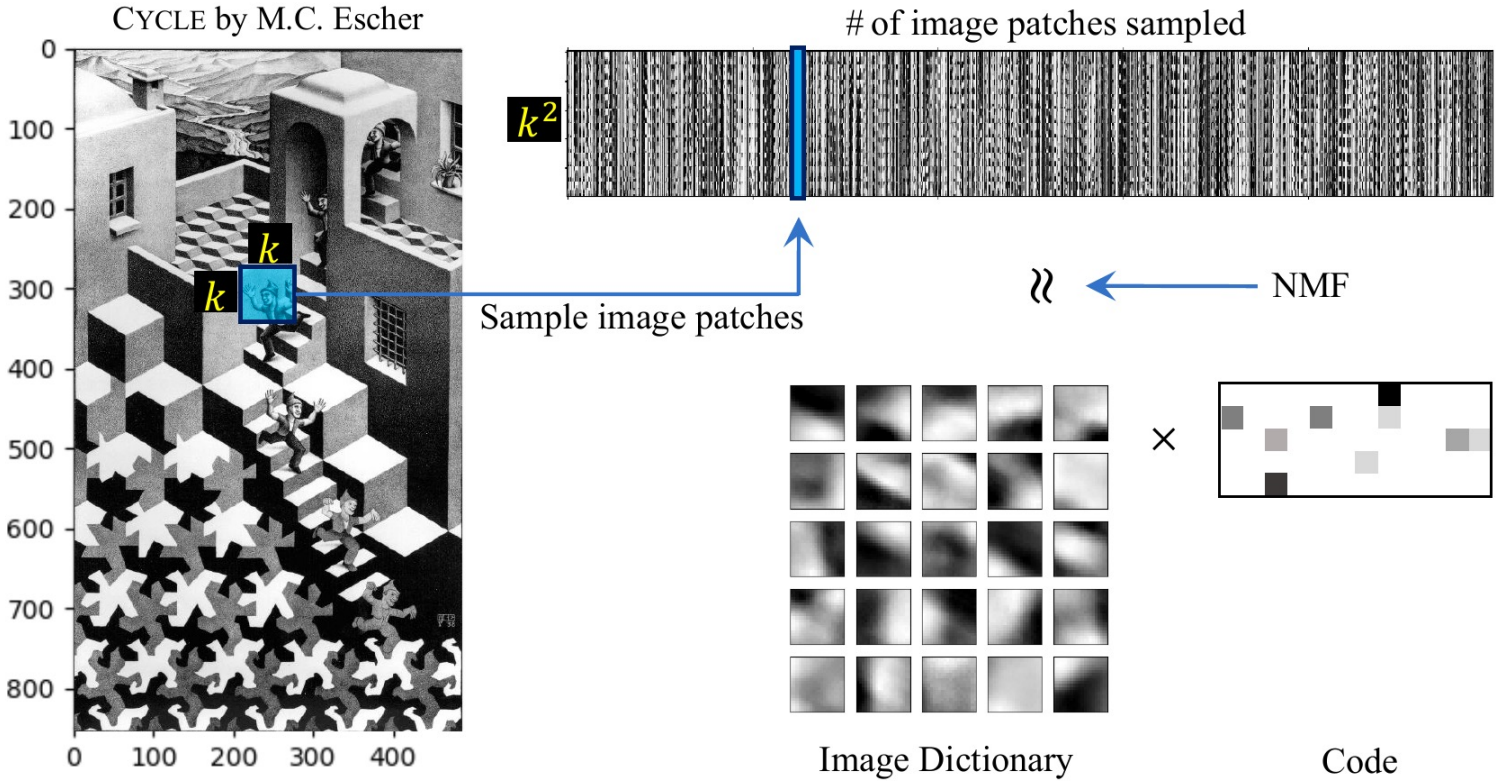
Stochastic Optimization with Markovian data

- Goal: Minimize the expected loss $\mathbb{E}_{X\sim\pi}[\ell(X,\theta)]$ given a loss function ℓ
- First attempt: *Empirical Loss Minimization*
  - Background: $\lim_{n\to\infty}$ Empirical Loss = Expected loss
  - Not practical in many cases:
    - The *empirical loss is often hard to minimize* (e.g., Matrix Factorization)

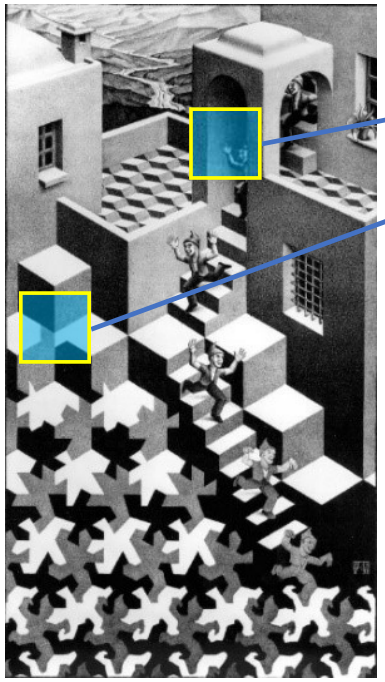


Network Dictionary Learning





CYCLE by M.C. Escher



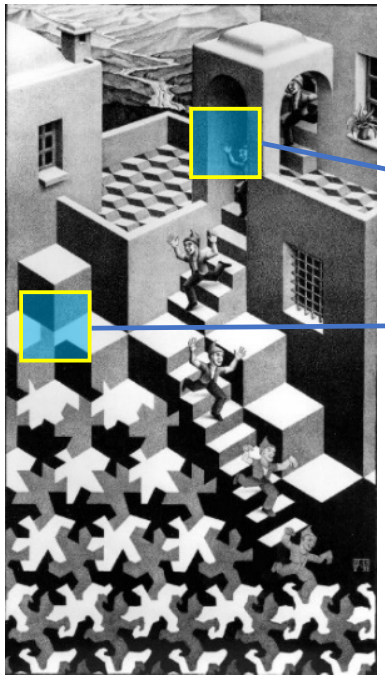
i.i.d. sampling

Minibatches of flattened image patches

Online NMF

Dictionary_1

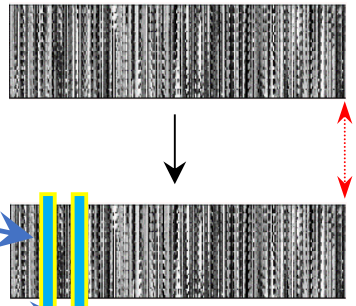
CYCLE by M.C. Escher



i.i.d. sampling

Minibatches of flattened image patches

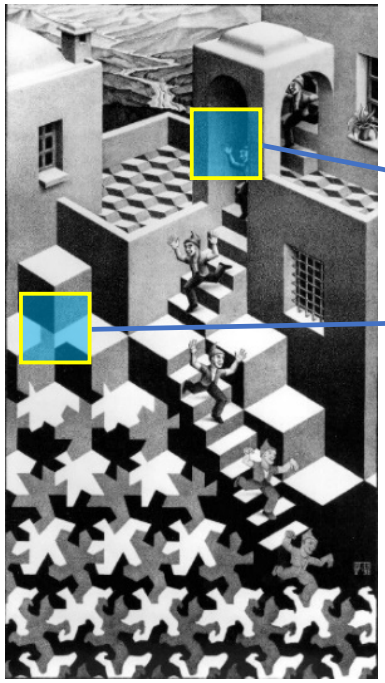
Online NMF



Dictionary_1

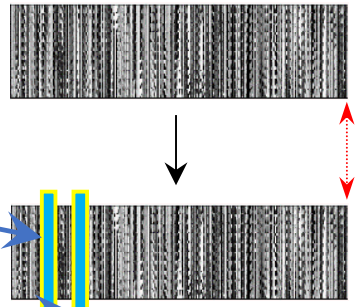
i.i.d.

CYCLE by M.C. Escher



i.i.d. sampling

Minibatches of flattened image patches



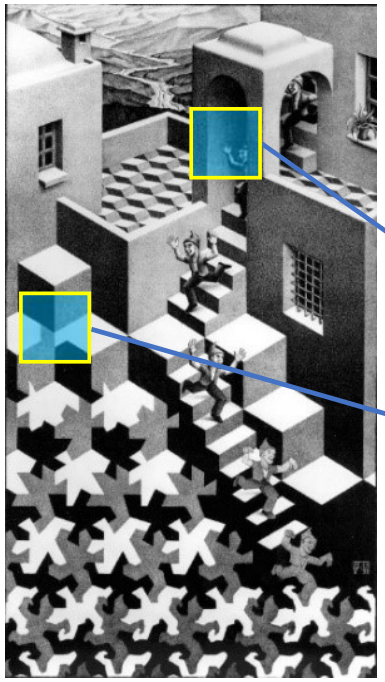
i.i.d.

Online NMF

Dictionary_1

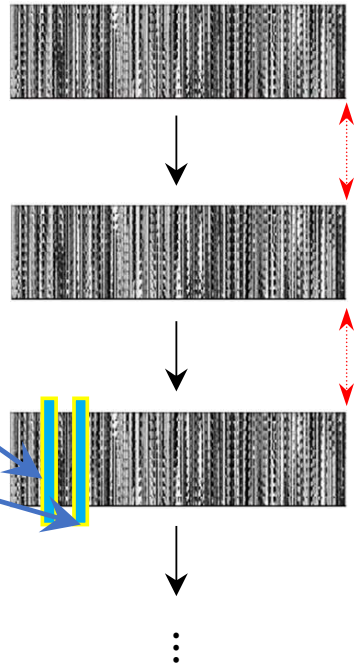
Dictionary_2

CYCLE by M.C. Escher



i.i.d. sampling

Minibatches of flattened image patches

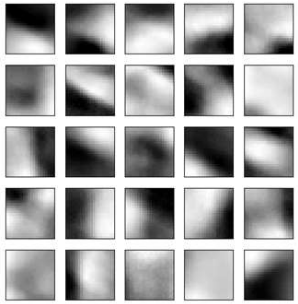


Online NMF

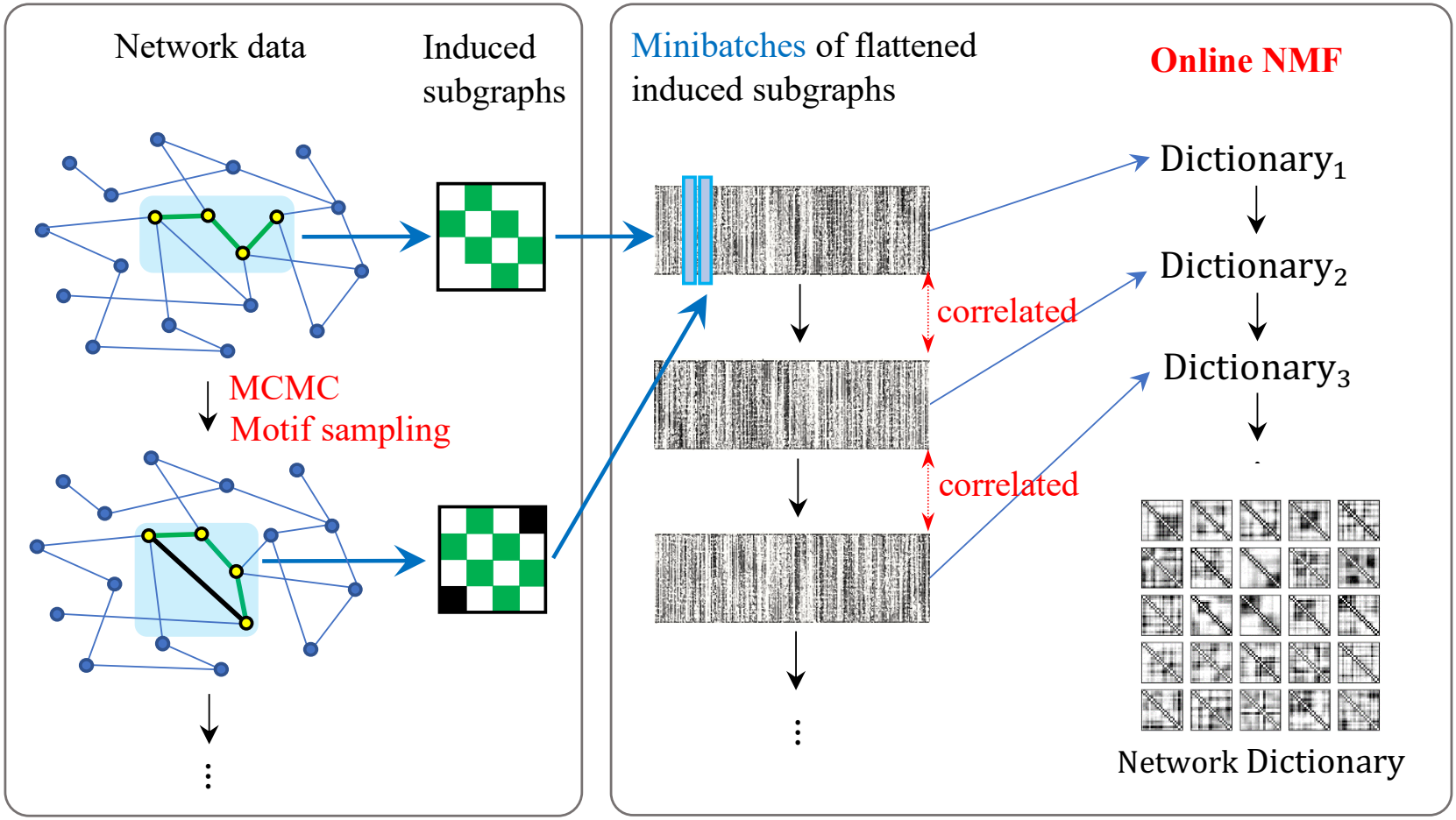
Dictionary_1

Dictionary_2

Dictionary_3



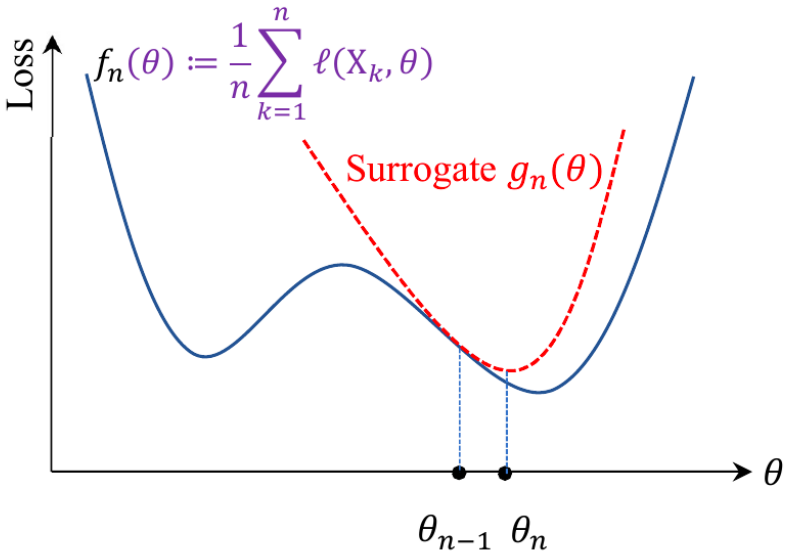
Network Dictionary Learning



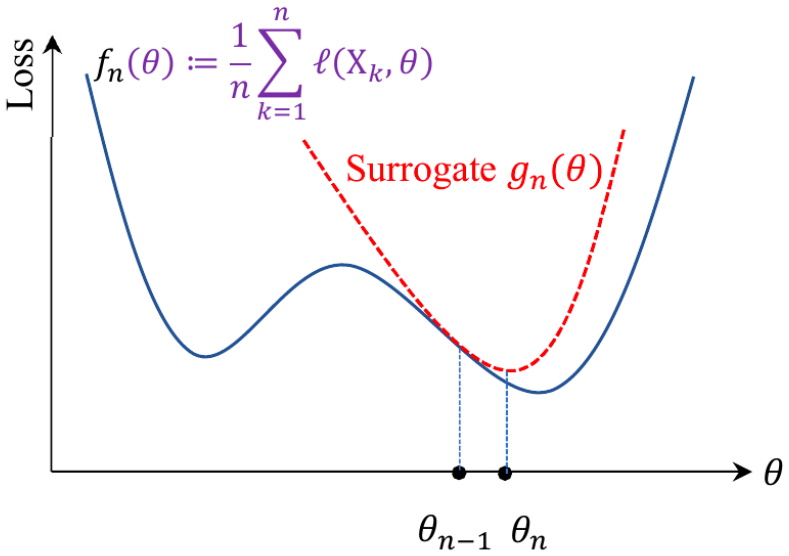
Here minibatches are non-i.i.d. --- functions of the underlying Markov chain

Do we still have convergence of online dictionary learning algoirthms? At what rate?

- Stochastic Majorization-Minimization (SMM) – Mairal [6]
  - Iteratively minimize majorizing surrogates g_n of the empirical loss f_n



- Stochastic Majorization-Minimization (SMM) – Mairal [6]
  - Iteratively minimize majorizing surrogates g_n of the empirical loss f_n



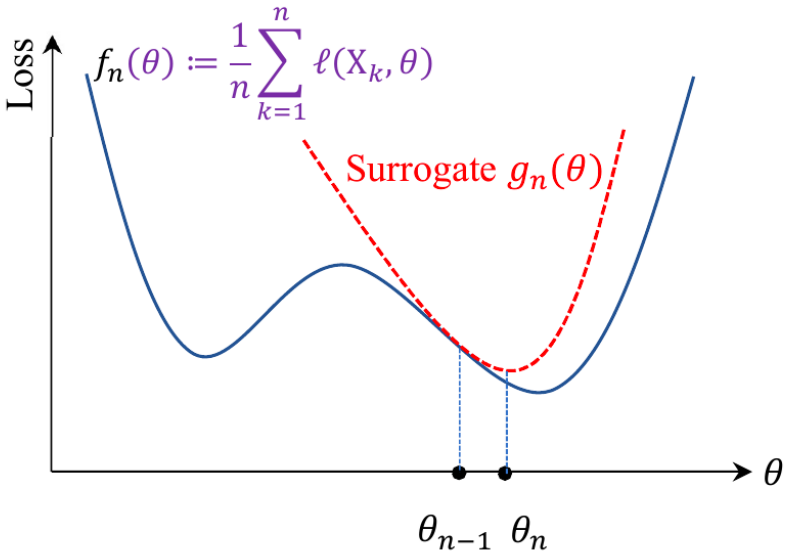
- Online Matrix Factorization in Mairal et al. [8]:

$$\text{(coding)} \quad H_n \leftarrow \operatorname*{argmin}_H \|X_n - \theta_{n-1} H\|_F^2$$

$$\text{(surrogate update)} \quad g_n(\theta) \leftarrow (1 - w_n) g_{n-1}(\theta) + w_n \cdot \|X_n - \theta H_n\|_F^2$$

$$\text{(dictionary update)} \quad \theta_n \leftarrow \operatorname*{argmin}_{\theta \in \Theta} g_n(\theta)$$

- Stochastic Majorization-Minimization (SMM) – Mairal [6]
  - Iteratively minimize majorizing surrogates g_n of the empirical loss f_n



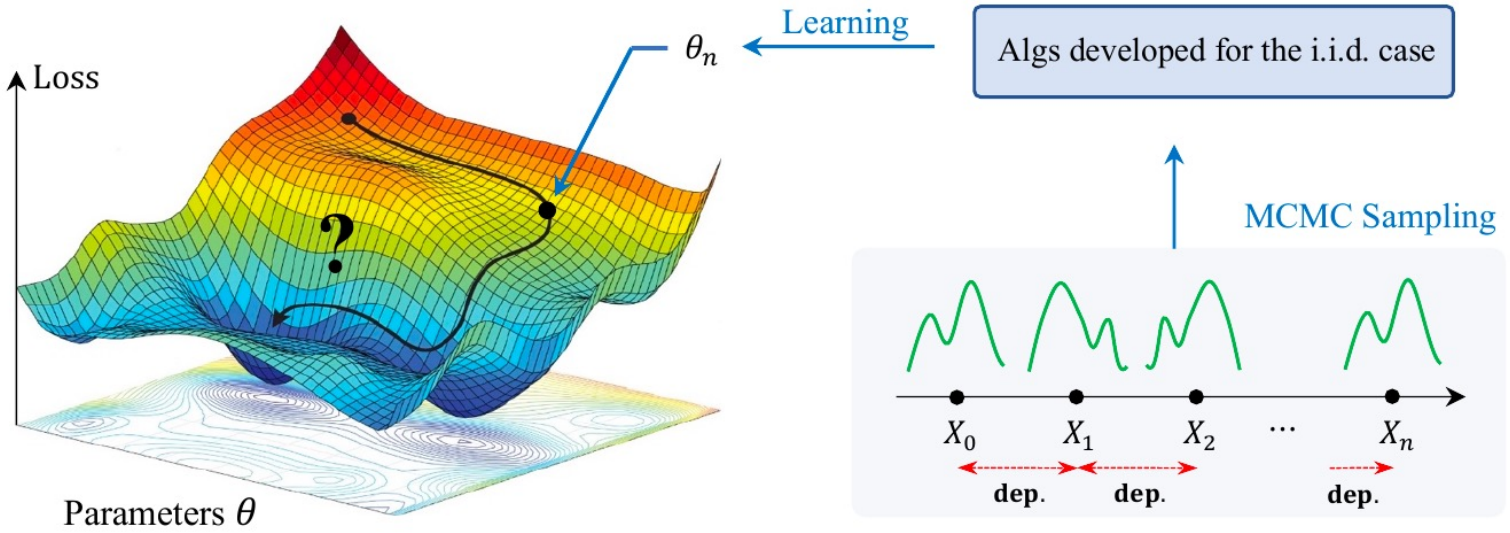
- When $\theta \mapsto \ell(X, \theta)$ is convex, $\theta_n \to$ global minimum at rate $O(\log n/\sqrt{n})$ for i.i.d. data samples X_n
- When $\theta \mapsto \ell(X, \theta)$ is non-convex, $\theta_n \to \{\text{st. pts. of expected loss}\}$ for i.i.d. data samples X_n

Convergence guarantees of stochastic optimization algorithms for non-i.i.d. data

Theorem. (L., Needell, Balzano, '20 JMLR)

Online dictionary learning algorithm for i.i.d. data

converges a.s. to the set of stationary points even for f(Markovian data).



Convergence guarantees of stochastic optimization algorithms for non-i.i.d. data

Theorem. (L., Needell, Balzano, '20 JMLR)

Online dictionary learning algorithm for i.i.d. data

converges a.s. to the set of stationary points even for f(Markovian data).

Theorem. (L., '22+)

Generalizes Online DL algs.

Nonconvex Stochastic Majorization-Minimization algorithm for i.i.d. data

converges a.s. to the set of stationary points even for f(Markovian data).

Conv. rate = $O(n^{1/4})$ for expected loss, = $O(n^{1/2})$ for empirical loss.

Convergence guarantees of stochastic optimization algorithms for non-i.i.d. data

Theorem. (L., Needell, Balzano, '20 JMLR)

Online dictionary learning algorithm for i.i.d. data

converges a.s. to the set of stationary points even for f(Markovian data).

Theorem. (L., '22+)

Generalizes Online DL algs.

Nonconvex Stochastic Majorization-Minimization algorithm for i.i.d. data

converges a.s. to the set of stationary points even for f(Markovian data).

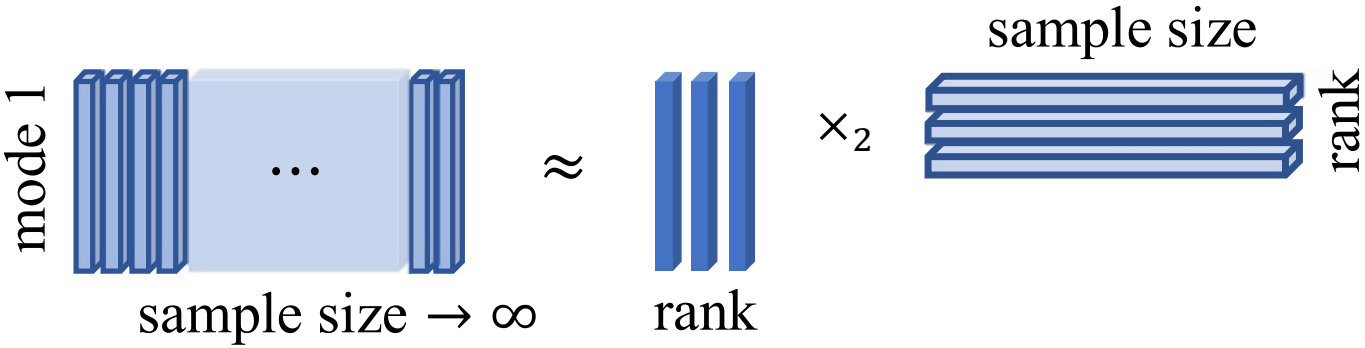
Conv. rate = $O(n^{1/4})$ for expected loss, = $O(n^{1/2})$ for empirical loss.

Theorem. (Alacaoglu, L., '22+)

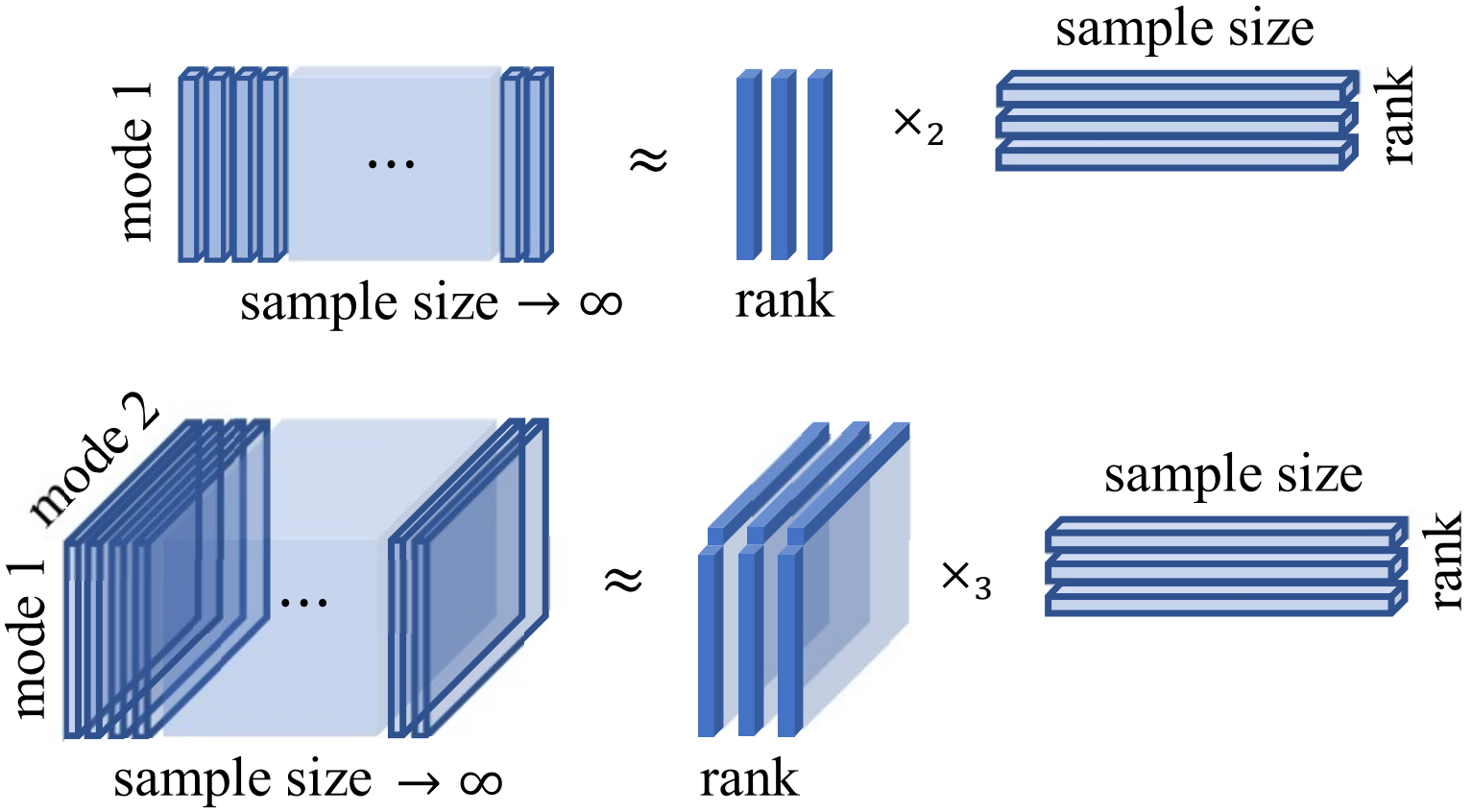
Nonconvex, nonsmooth, constrained Stochastic Proximal GD for f(Markovian data)

converges a.s. to the set of stationary points at rate $O(n^{1/4})$ for expected loss.

CP-Dictionary Learning for tensor-valued data



CP-Dictionary Learning for tensor-valued data



CP-DL for short-lasting topic detection

- $\mathbf{X}$ = words × time × docs
- $\mathbf{U}^{(1)}$ = words × topic, $\mathbf{U}^{(2)}$ = time × topic, $\mathbf{U}^{(3)}$ = docs × topic

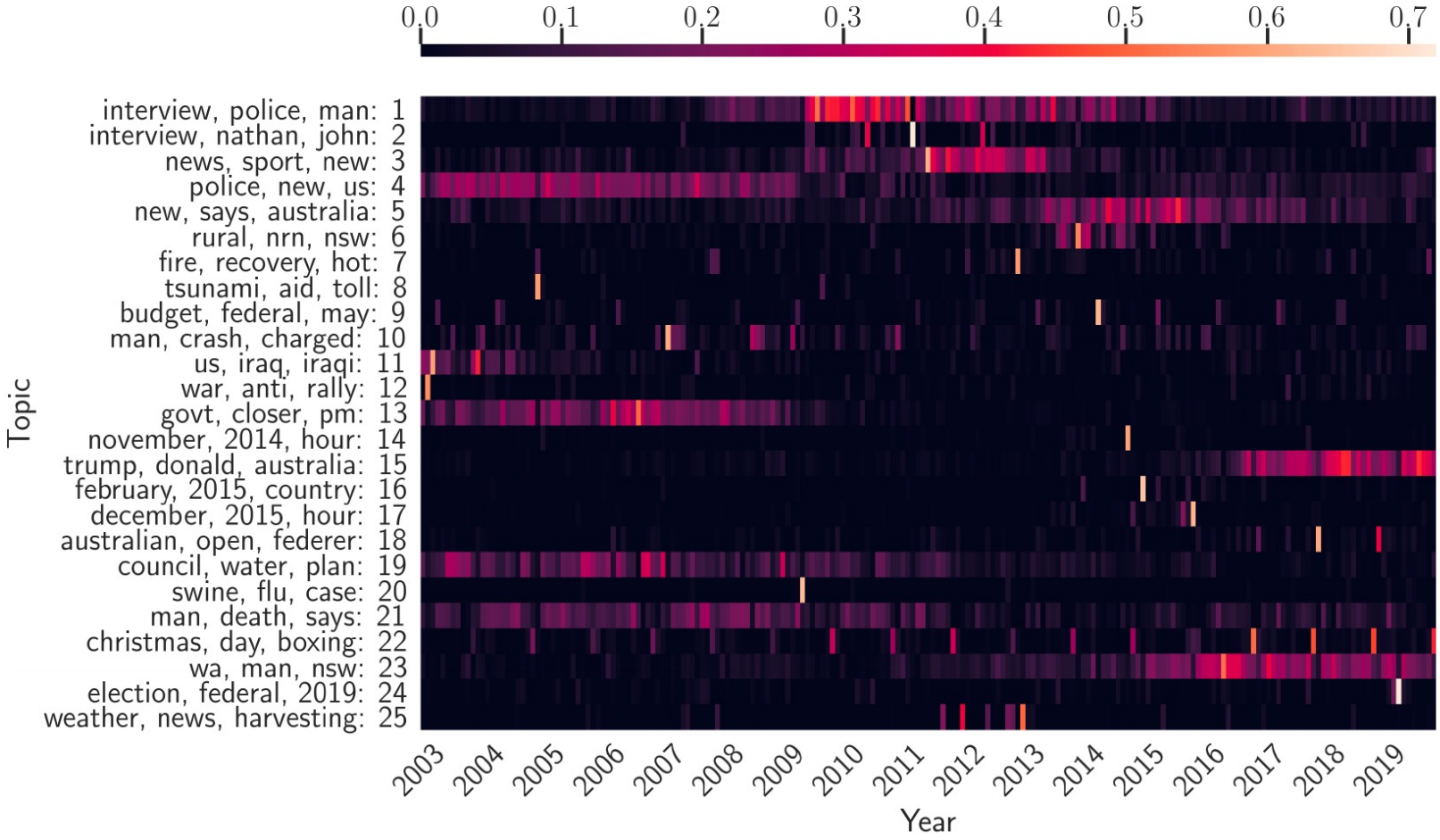


Figure: From (Kassab, Kryshchenko, L., Molitor, Needell, and Rebrova '21)

CP-DL for learning spatio-temporal brain activity patterns

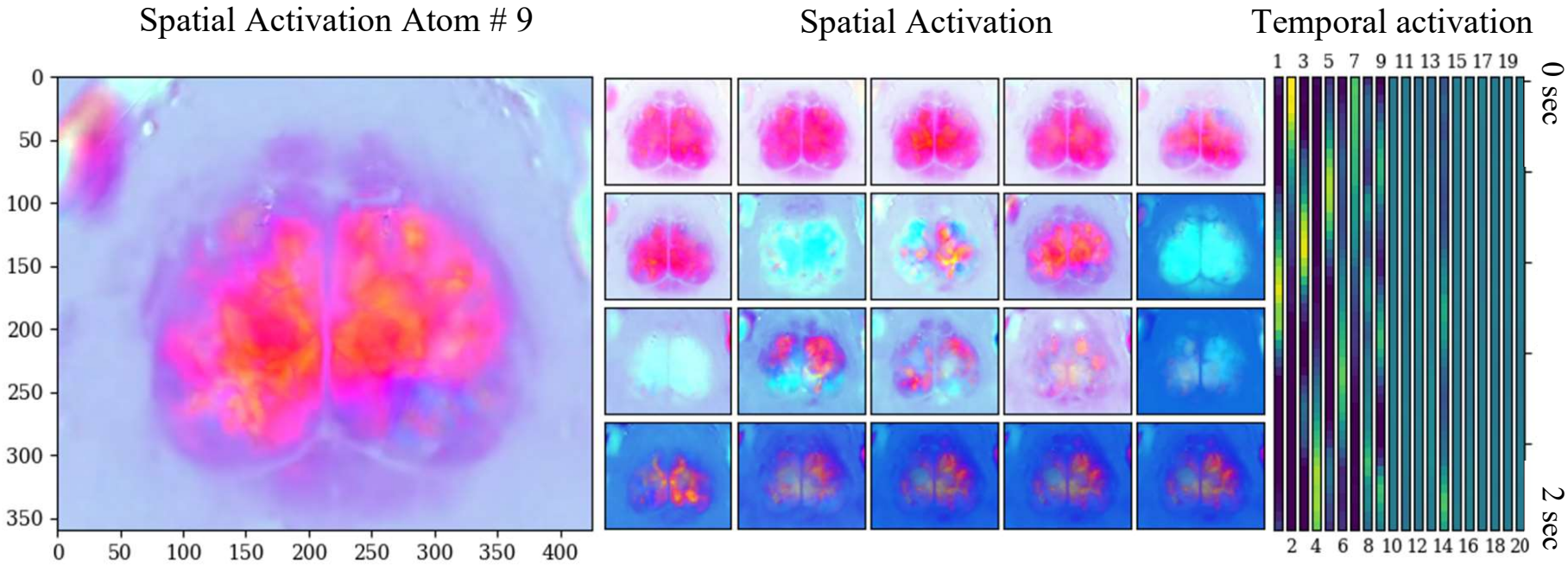
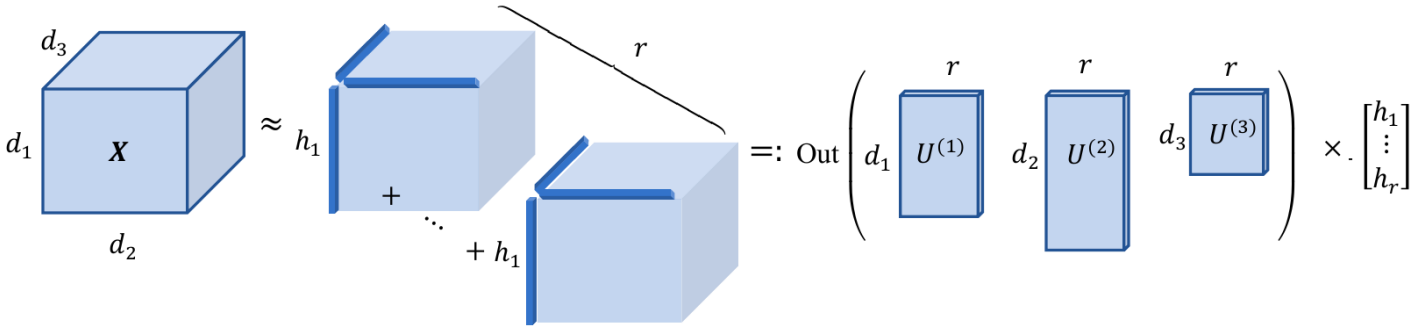


Figure. Learning 20 CP-dictionary from video frames on brain activity across the m ouse cortex.

Tensor CP-Dictionary Learning

- Online CP-dictionary Learning (L., Strohmeier, Needell '20 [5]):

(CP-recons. error) $\ell(\underbrace{\mathbf{X}}_{m\text{-tensor}}, \mathbf{U} = \underbrace{[U^{(1)},\dots,U^{(m)}]}_{\text{factor matrices}}, H) := \|\mathbf{X} - \underbrace{\mathrm{Out}(\mathbf{U})}_{\text{CP-dict.}} \times_{m+1} H\|_F^2$



- Upon arrival of $\mathbf{X}_n \in \mathbb{R}^{d_1 \times \cdots \times d_m}$:

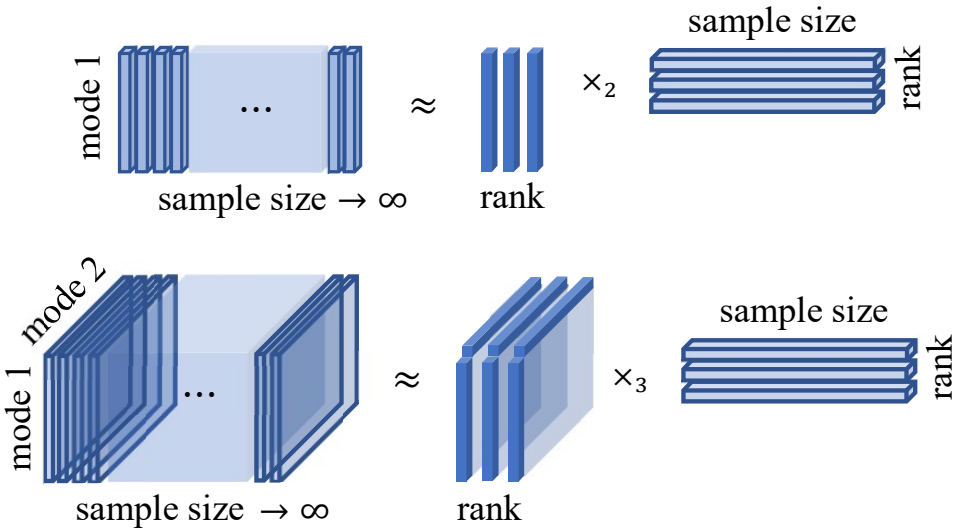
$$\begin{cases} H_n = \operatorname{argmin}_{H \in \subseteq \mathbb{R}^{r\times 1}_{\geq 0}} \ell(\mathbf{X}_n, \mathbf{U}_{n-1}, H) \\ \bar{g}_n(\mathbf{U}) = (1-w_n)\bar{g}_{n-1}(\mathbf{U}) + w_n \ell(\mathbf{X}_n, \mathbf{U}, H_n) \qquad (m\text{-block multi-convex}) \\ \text{for } i = 1,\dots,m: \\ \quad U_n^{(i)} \in \operatorname*{argmin}_{\substack{U \in \mathbb{R}^{d_i \times r}_{\geq 0} \\ \|U - U_{n-1}^{(i)}\| \leq c' w_n}} \bar{g}_n(U_n^{(1)},\dots,U_n^{(i-1)}, U, U_{n-1}^{(i+1)},\dots,U_{n-1}^{(m)}). \end{cases}$$

Convergence guarantees of stochastic optimization algorithms for non-i.i.d. data

Theorem. (L., Strohmeier, Needell, '22 JMLR)

Online CP-dictionary learning algorithm

converges a.s. to the set of stationary points even for f(Markovian data).



Convergence guarantees of stochastic optimization algorithms for non-i.i.d. data

Theorem. (L., Strohmeier, Needell, '22 JMLR)

Online CP-dictionary learning algorithm

converges a.s. to the set of stationary points even for f(Markovian data).

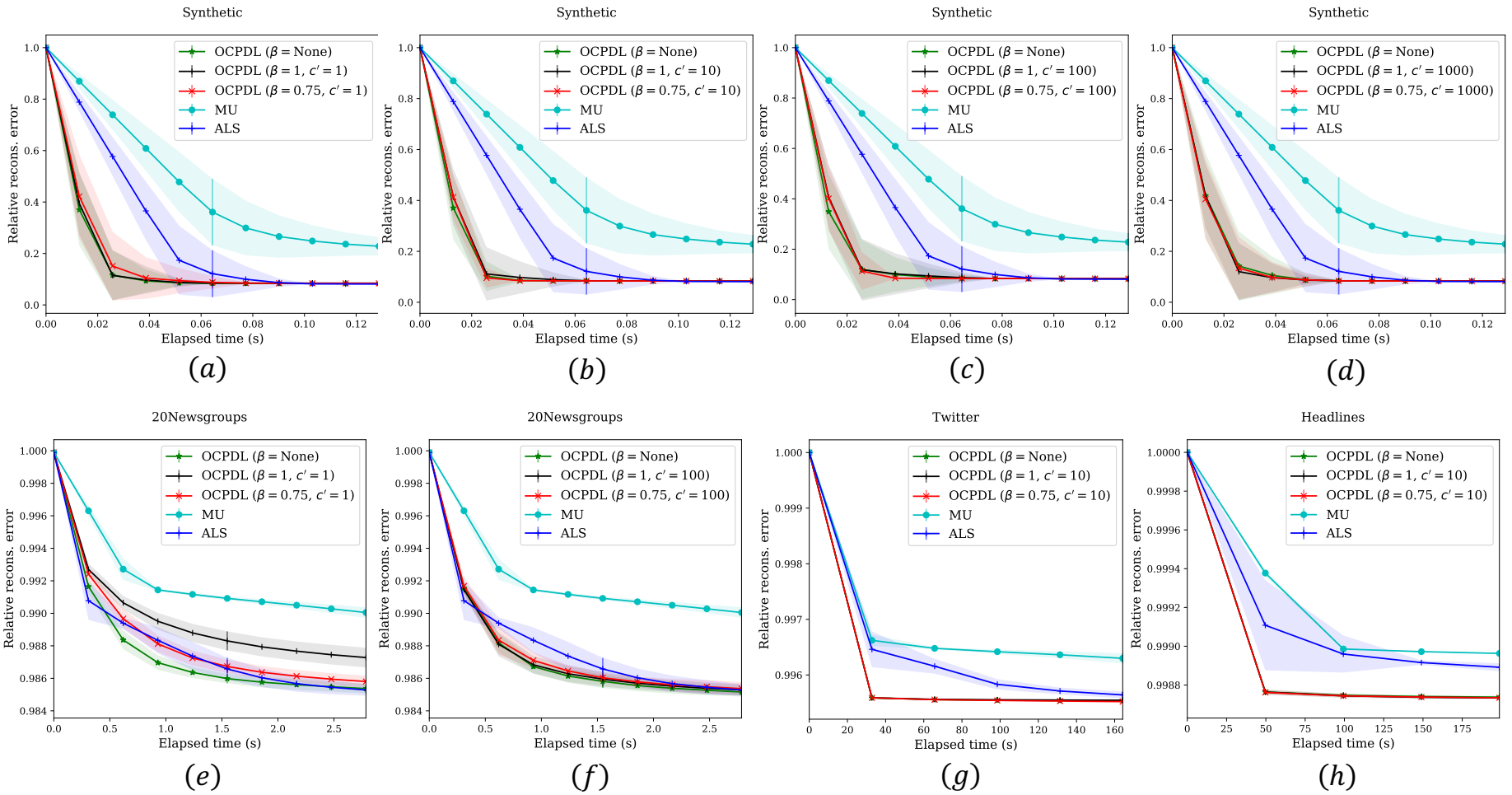
Theorem. (L., '22+)

Online CP-dictionary learning algorithm

converges a.s. to the set of stationary points even for f(Markovian data).

Conv. rate = $O(n^{1/4})$ for expected loss, = $O(n^{1/2})$ for empirical loss.

- Online CP-dictionary Learning
  - Only bounded memory to learn from infinitely many samples
  - Cheaper per-iteration cost than offline methods
  - Converges faster than offline methods (empirically)



Future/Ongoing Works

Temporal Network Dictionary Learning

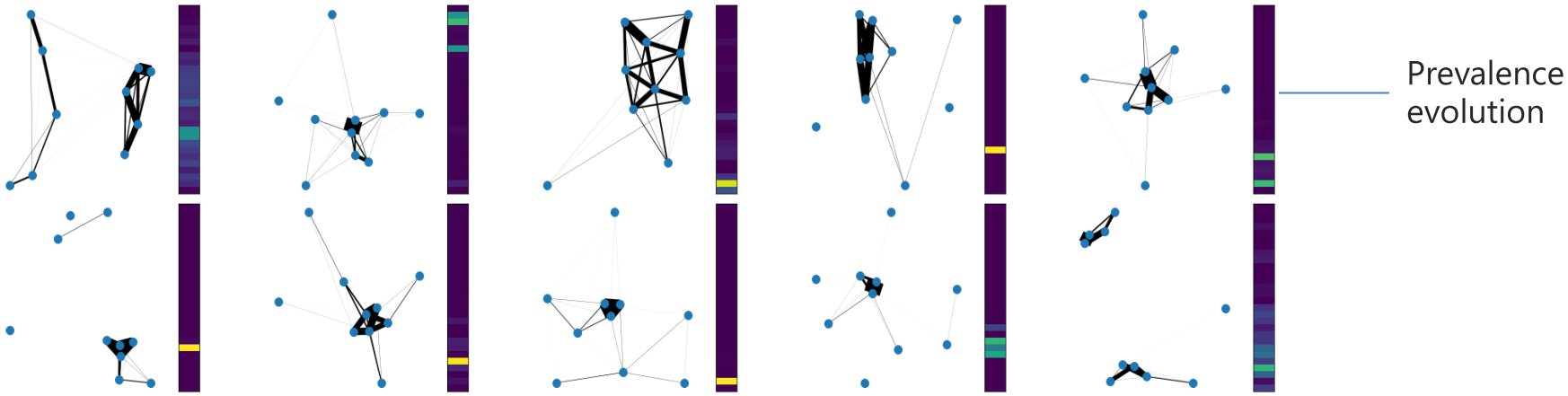
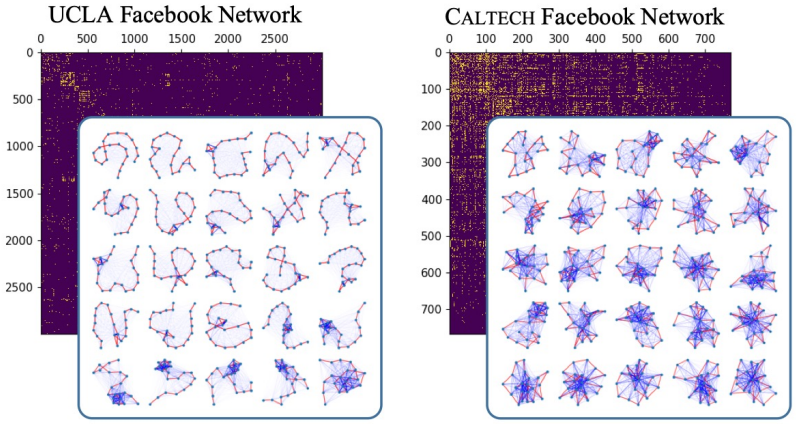
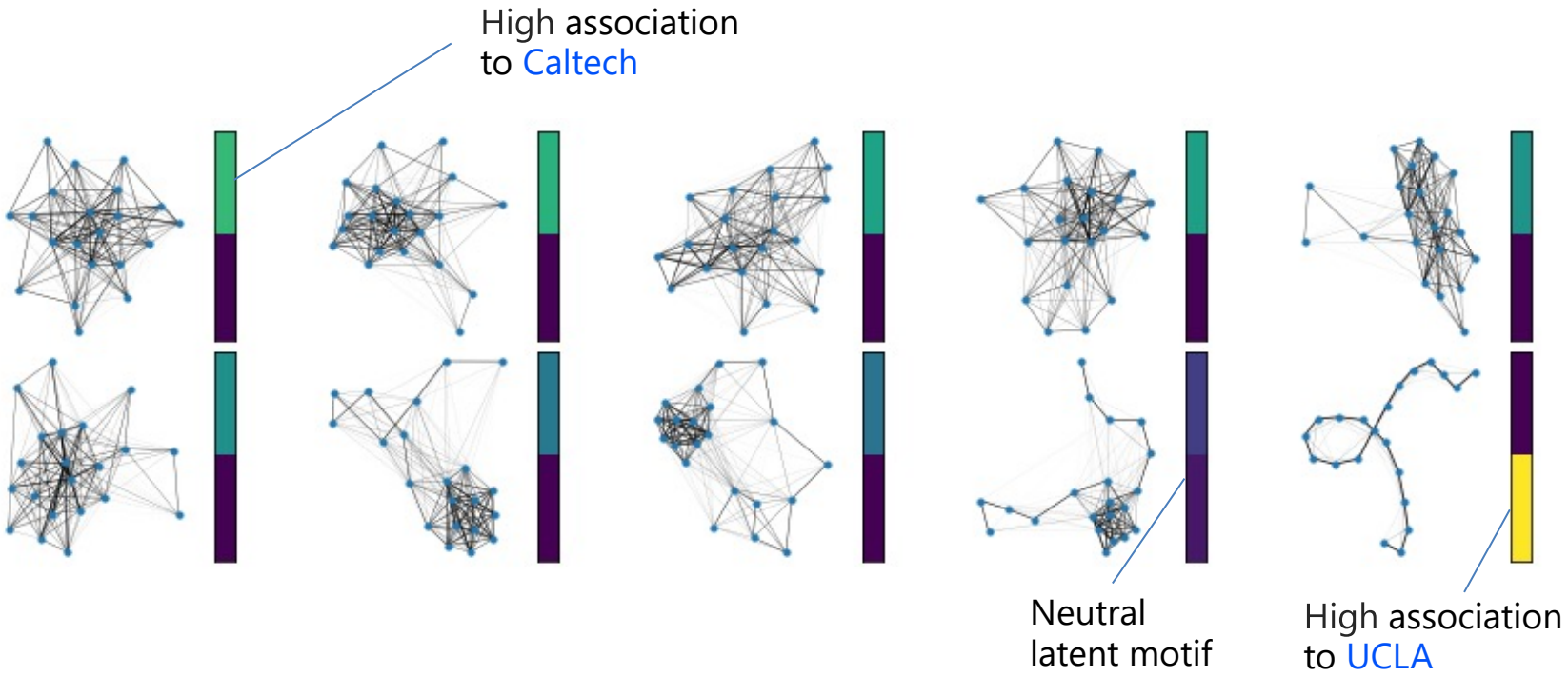


Figure. Temporal network dictionary learned from coauthorship network of DBLP fr
om year 1990 to 2018 (top to bottom).

Supervised Network Dictionary Learning



Learn not only summarizing, but discriminating latent motifs?



Thank you very much!

References

1. Hanbaek Lyu, Facundo Memoli, and David Sivakoff, "*Sampling random graph homomorphisms and applications to network data analysis.*" In revision for Journal of Machine Learning Research. [Preprint, GitHub]
2. Hanbaek Lyu, Christopher Strohmeier, and Deanna Needell, "*Online nonnegative tensor factorization and CP-Dictionary Learning for Markovian data*" Journal of Machine Learning Research 23(148):1–50, 2022 [Journal, Preprint, GitHub]
3. Hanbaek Lyu, Deanna Needell, and Laura Balzano, "*Online matrix factorization for markovian data and applications to network dictionary learning.*" Journal of Machine Learning Research. 21(251):1–49, 2020 [Journal, Preprint, ,GitHub]
4. Joowon Lee, Hanbaek Lyu, and Weixin Yao, "*Supervised Dictionary Learning with Auxiliary Covariates*" [Preprint, GitHub] (2022)
5. Ahmet Alacaoglu and Hanbaek Lyu, "*Convergence and Complexity of Stochastic Subgradient Methods with Dependent Data for Nonconvex Optimization*" Preprint (2022)
6. Hanbaek Lyu, "*Stochastic regularized block majorization-minimization with weakly convex and multi-convex surrogates*" Preprint (2022)
7. Hanbaek Lyu, Yacoub Kureh, Joshua Vendrow*, and Mason A. Porter, "*Learning low-rank mesoscale structures of networks*" (2021) [Preprint, GitHub, Python package "*ndlearn*"]
8. Hanbaek Lyu, "*Convergence and complexity of block coordinate descent with diminishing radius for nonconvex optimization*" Preprint (2020)

Sketch of proof (Handling Markovian Dependence)

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \|\underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n)\|_F^2 \end{cases}$$

Sketch of proof (Handling Markovian Dependence)

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \|\underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n)\|_F^2 \end{cases}$$

- **Lem 1**: $\sum_{n=0}^{\infty} w_n \mathbb{E}[\Delta_n] < \text{Abs. Const.} < \infty$.
- **Lem 2**: $O(\mathbb{E}[\Delta_n] - \mathbb{E}[\Delta_{n-1}]) = O(w_n)$. $\cdots$ (not today:))

Sketch of proof (Handling Markovian Dependence)

- $$\Delta_n := \begin{cases} (\text{Relaxation error})_n := \overbrace{g_n(\theta_n)}^{\text{surrogate error at time } n} - \overbrace{f_n(\theta_n)}^{\text{empirical error at time } n} \geq 0 \\ (\text{Optimality gap})_n := \|\underbrace{\nabla g(\theta_n)}_{\perp \text{ to } \partial\Theta} - \nabla f(\theta_n)\|_F^2 \end{cases}$$

- **Lem 1**: $\sum_{n=0}^{\infty} w_n \mathbb{E}[\Delta_n] < \text{Abs. Const.} < \infty$.
- **Lem 2**: $O(\mathbb{E}[\Delta_n] - \mathbb{E}[\Delta_{n-1}]) = O(w_n)$. $\cdots$ (not today:))

  - From this, one can deduce

  (1) $\Delta_n \to 0$ a.s. as $n \to \infty$,

  (2) $\min_{1 \leq k \leq n} \sup_{\text{initialization}} \Delta_n = O\left(\frac{C}{\sum_{k=0}^{n} w_k}\right)$ a.a.s.

Sketch of proof (Handling Markovian Dependence)

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

Sketch of proof (Handling Markovian Dependence)

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Standard approach for the i.i.d. case:
  - $$\mathbb{E}\left[\ell(X_{n+1},\theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1},\theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_n\right]\right]$$
  $$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X,\theta_n)] - f_n(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by uniform CLT}} \right]$$

Sketch of proof (Handling Markovian Dependence)

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Standard approach for the i.i.d. case:
  - $$\mathbb{E}\left[\ell(X_{n+1},\theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1},\theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_n\right]\right]$$
  $$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X,\theta_n)] - f_n(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by uniform CLT}} \right]$$
  - So the RHS above is $\le C\sum_{n=1}^{\infty} w_n^2\sqrt{n} < \infty$.

c.f.

- $w_n \equiv$ stepsize in SGD
- Nonconvex, unconstrained SGD convergence requires $\sum_{n=0}^{\infty} w_n^2 < \infty$
- This is where we get $O(1/n^{1/4})$ SMM convergence instead of $O(1/n^{1/2})$ in SGD

Sketch of proof (Handling Markovian Dependence)

- After some nontrivial work, one can show

$$\sum_{n=0}^{\infty} w_{n+1}\mathbb{E}[\Delta_n] \le c_1 + c_2 \sum_{n=0}^{\infty} w_{n+1} \left| \mathbb{E}\left[\underbrace{\ell(X_{n+1},\theta_n)}_{\text{random loss at time } n+1} - \underbrace{f_n(\theta_n)}_{\text{empirical loss at time } n} \right] \right|$$

- Our approach for the dependent case:
  - Condition on distant past $\mathcal{F}_{n-\sqrt{n}}$ instead of the recent history $\mathcal{F}_n$:

$$\mathbb{E}\left[\ell(X_n,\theta_n) - f_n(\theta_n)\right] = \mathbb{E}\left[\mathbb{E}\left[\ell(X_{n+1},\theta_n) - f_n(\theta_n) \,\middle|\, \mathcal{F}_{n-\sqrt{n}}\right]\right]$$

$$= \mathbb{E}\left[\underbrace{\mathbb{E}_{X\sim\pi}[\ell(X,\theta_n)] - f_{n-\sqrt{n}}(\theta_n)}_{O(w_n\sqrt{n}) \text{ uniformly by MC uniform CLT}} \right] + C \underbrace{\|\pi - \pi(\mathbf{x}_n|\mathcal{F}_{n-\sqrt{n}})\|_{TV}}_{\text{MC mixing: } O(\exp(-\sqrt{n}))}$$

  - Again, the RHS above is $\le C' \sum_{n=1}^{\infty} w_n^2\sqrt{n} < \infty$.