Recent Advances in Stochastic Flow Clustering

Srinivasan Parthasarathy

Data Mining Research Laboratory
Dept. of Computer Science and Engineering
The Ohio State University

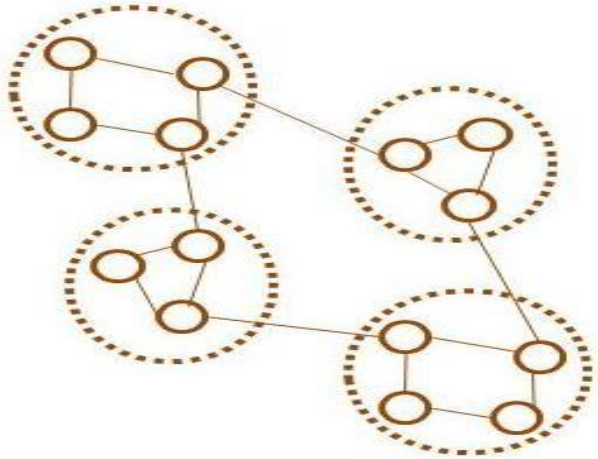
http://www.cse.ohio-state.edu/~srini

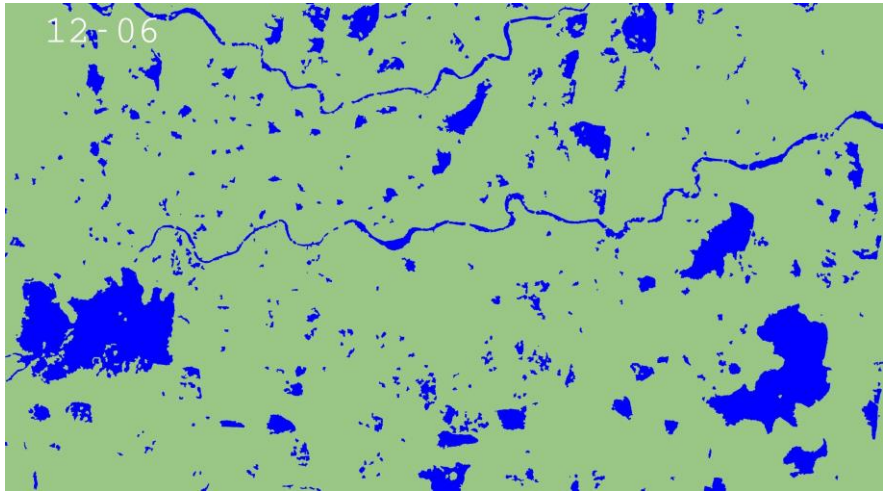
My background

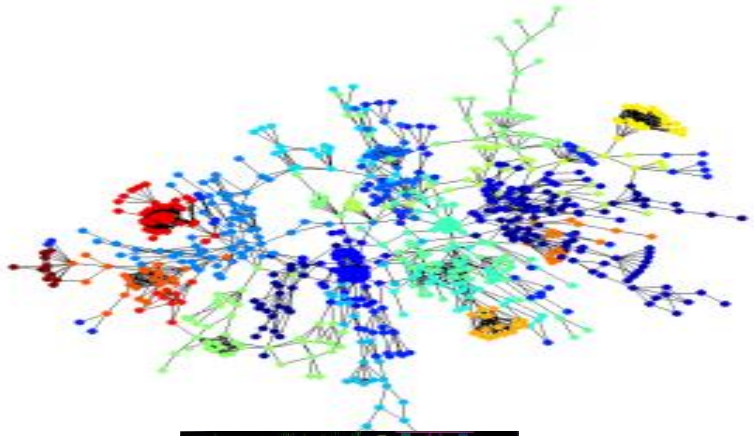
- Analytics and Machine Learning
- Database Systems
- High Performance Computing

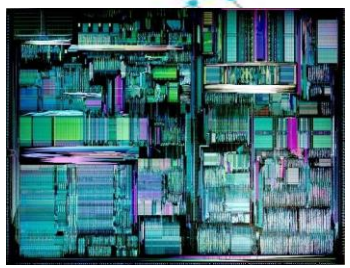
Graph Clustering: A Fundamental Problem

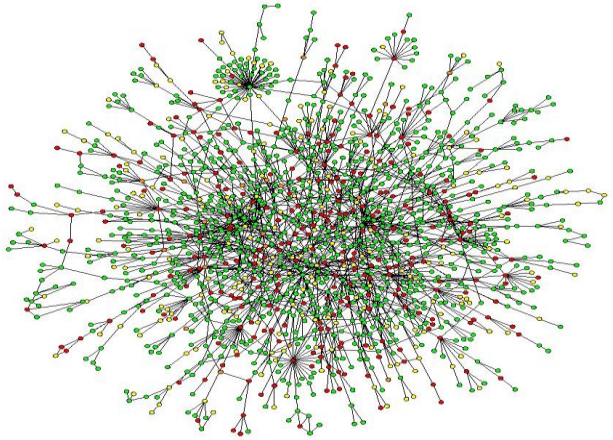
Given a graph, discover groups of nodes that are strongly connected to one another but weakly connected to the rest of the graph.







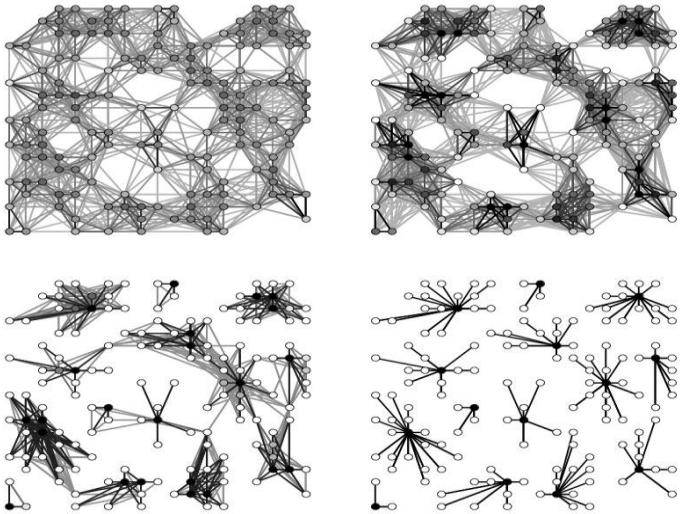




Markov Clustering (MCL)

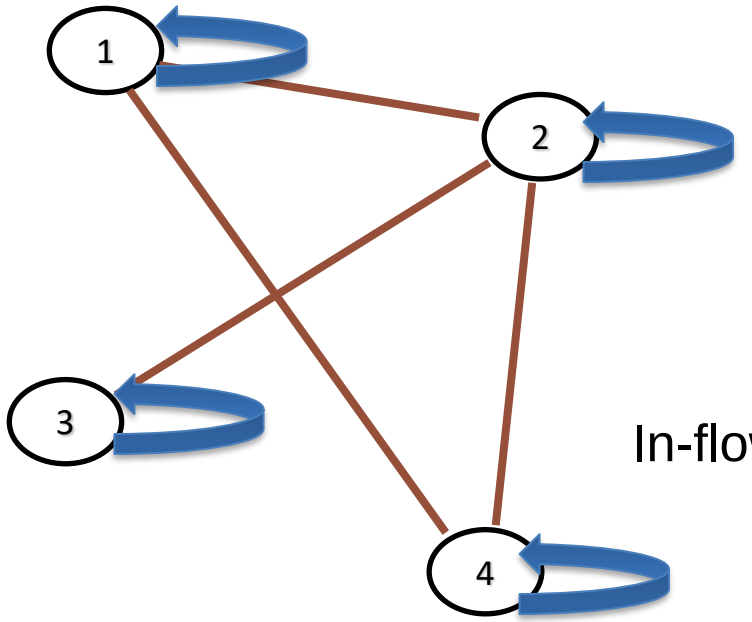
Stijn van Dongen, 2000

The original Stochastic flow clustering algorithm



Column Stochastic Matrix: A matrix where each column sums to 1.

Stochastic Flow: An entry in a column stochastic matrix, interpreted as the "flow" or "transition probability".

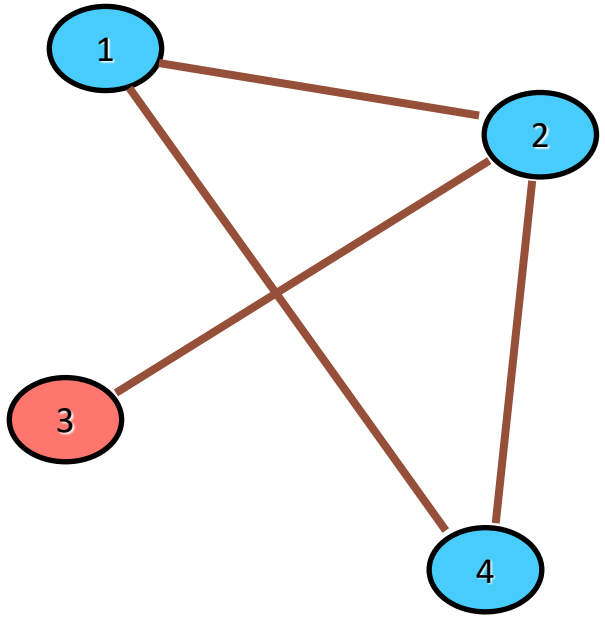


In-flows of 2

	1	2	3	4
1	0.33	0.25		0.33
2	0.33	0.25	0.5	0.33
3		0.25	0.5	
4	0.33	0.25		0.33

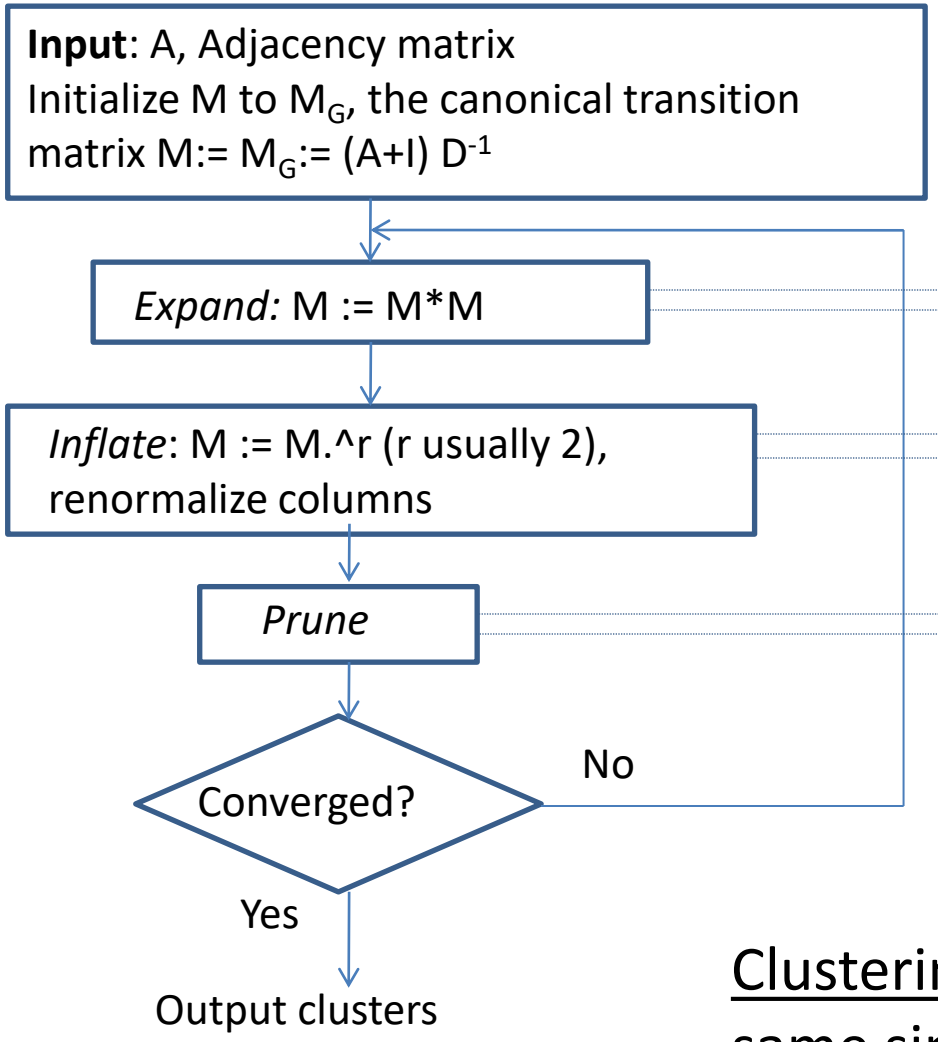
Out-flows of 2

Repeatedly apply certain operations to the flow matrix until the matrix converges and can be interpreted as a clustering.



	1	2	3	4
1				
2	1.0	1.0		1.0
3			1.0	
4				

The MCL algorithm

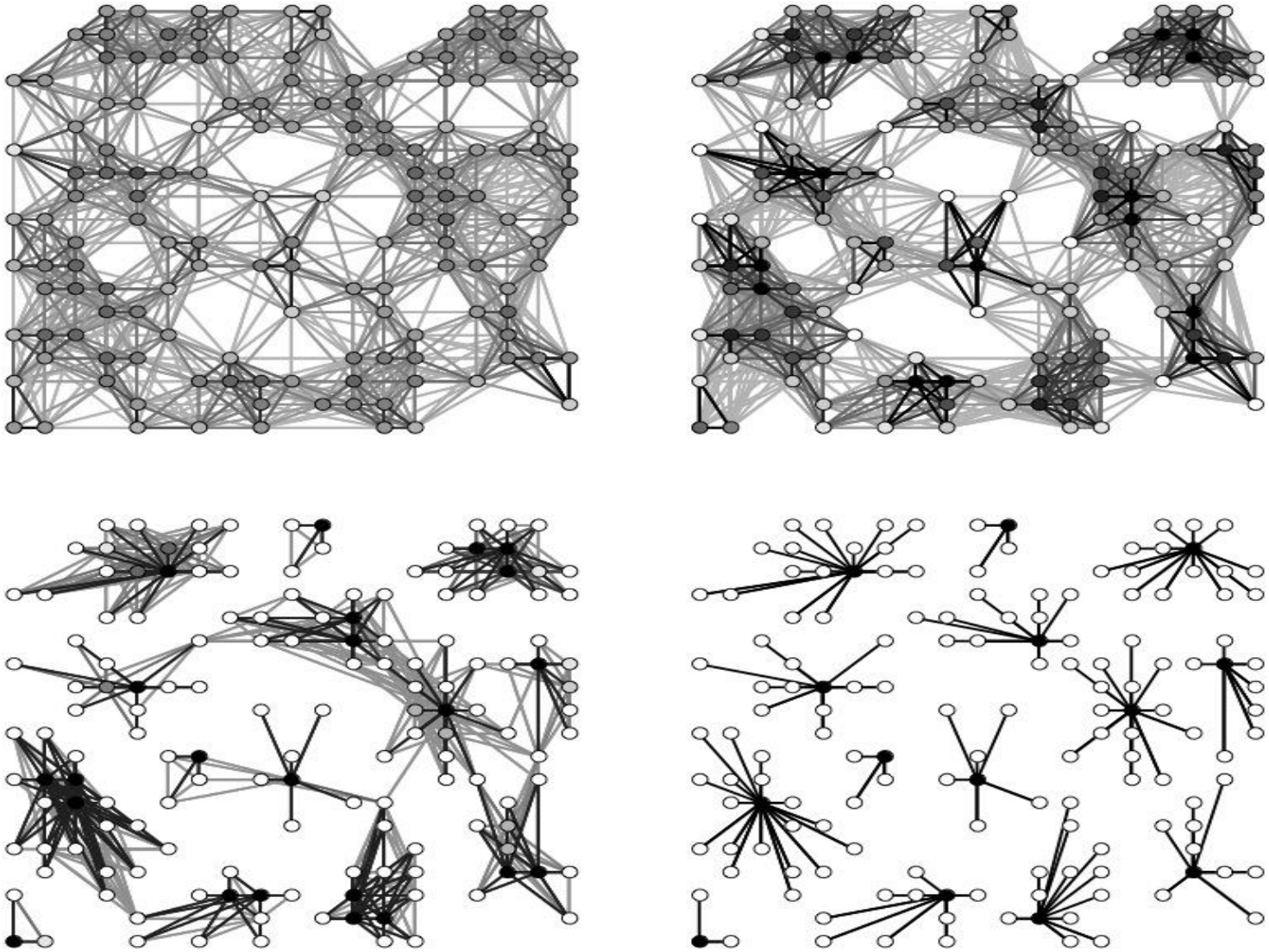


Enhances flow to well-connected nodes (i.e. nodes within a community).

Increases inequality in each column. "Rich get richer, poor get poorer."
(reduces flow across communities)

Saves memory by removing entries close to zero. Enables faster convergence

<u>Clustering Interpretation</u>: Nodes flowing into the same sink node are assigned same cluster labels



[van Dongen ’00]

MCL Strengths

1. Theoretically well founded [Von Dongen'00]
2. Simple, linear algebraic operations
3. Noise Tolerant. [Brohee'06, Vlasblom'09]

[Chakrabarti and Faloutsos '06]

MCL Limitations

1. Outputs many small clusters. [Satuluri, Parthasarathy'09]

2. Does not scale well. [Chakrabarti, Faloutsos'06]

[Chakrabarti and Faloutsos '06]

MCL Flaws

1. Outputs many small clusters.

   Fix I: Regularized MCL

2. Does not scale well.

   Fix II: Multi-Level Regularized MCL

Key Idea I: The *Regularize* operator

Why does MCL output many clusters?
Due to overfitting; it does not penalize divergence of flows between neighbors.

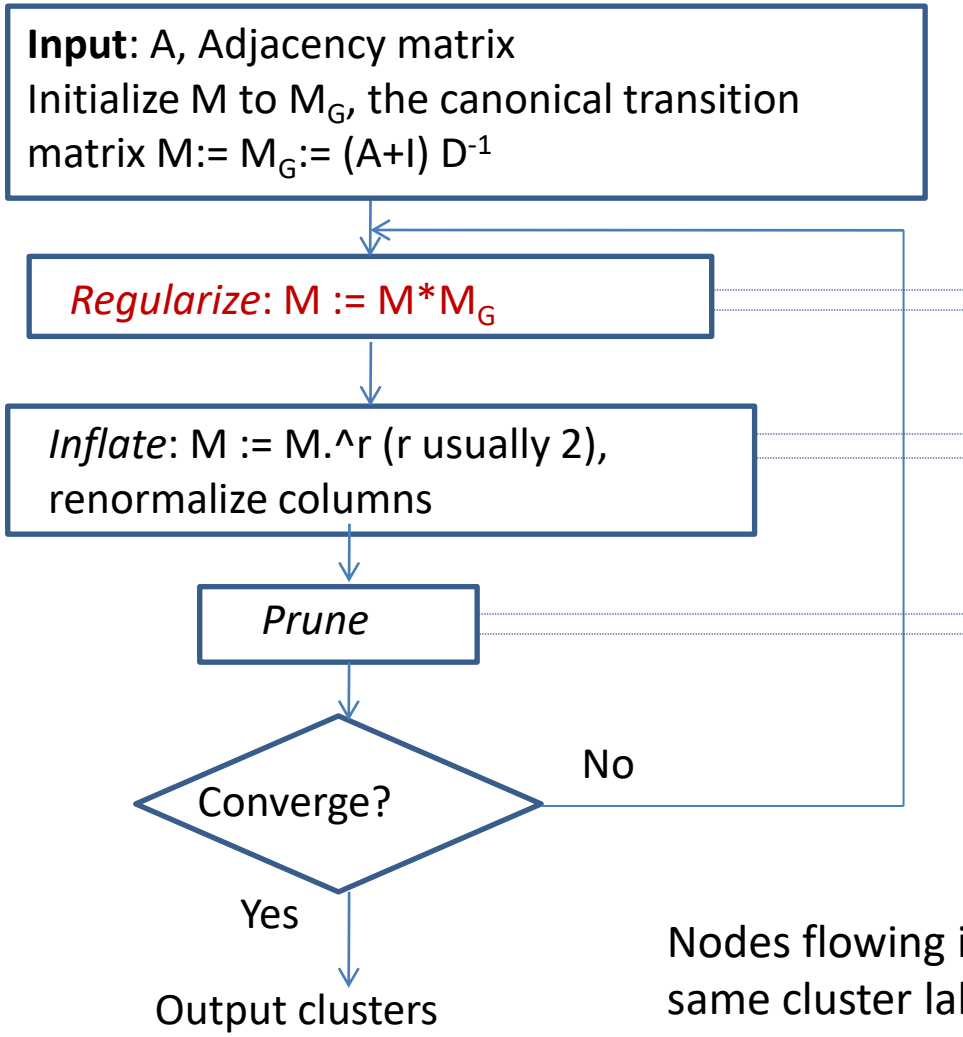
Remedy: Penalize divergence in flows between neighbors. Use KL Divergence (a well known measure for comparing probability distributions).

Turns out to have a nice closed form solution:

$$Regularize(M) := M*(A+I)D^{-1} = M*M_G$$

THE OHIO STATE UNIVERSITY

The Regularized-MCL algorithm



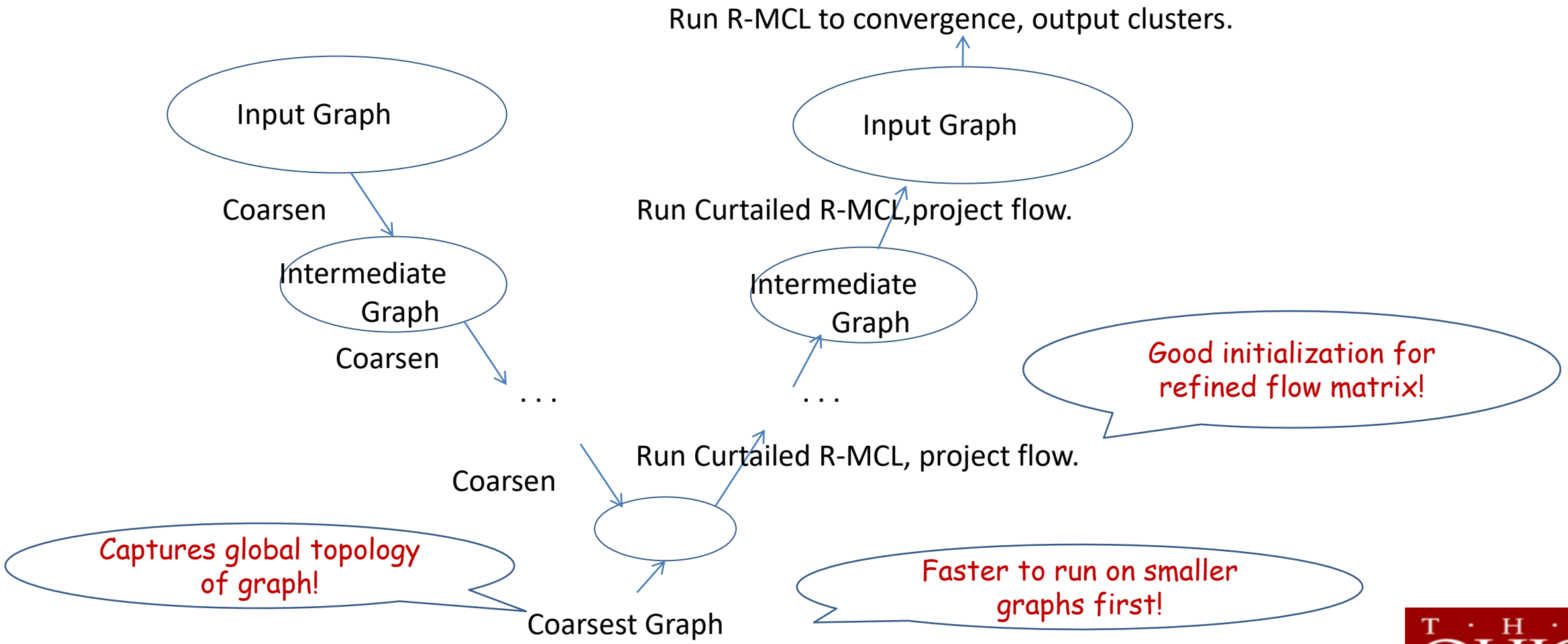
Takes into account flows of the neighbors.

Increases inequality in each column. "Rich get richer, poor get poorer."
[*Hadamard power + rescaling*]

Saves memory by removing entries close to zero. Enables faster convergence

Nodes flowing into the same sink node are assigned same cluster labels

Key Idea II: Multi-level Regularized MCL



Additional Optimizations for Scalability

Key Idea 3: Graph Sparsification

- Reduces clustering time of billion node graphs from ***hours*** down to ***minutes***. [SIGMOD'11, WWW'13]
- Theoretical rationale [SoCG'17]

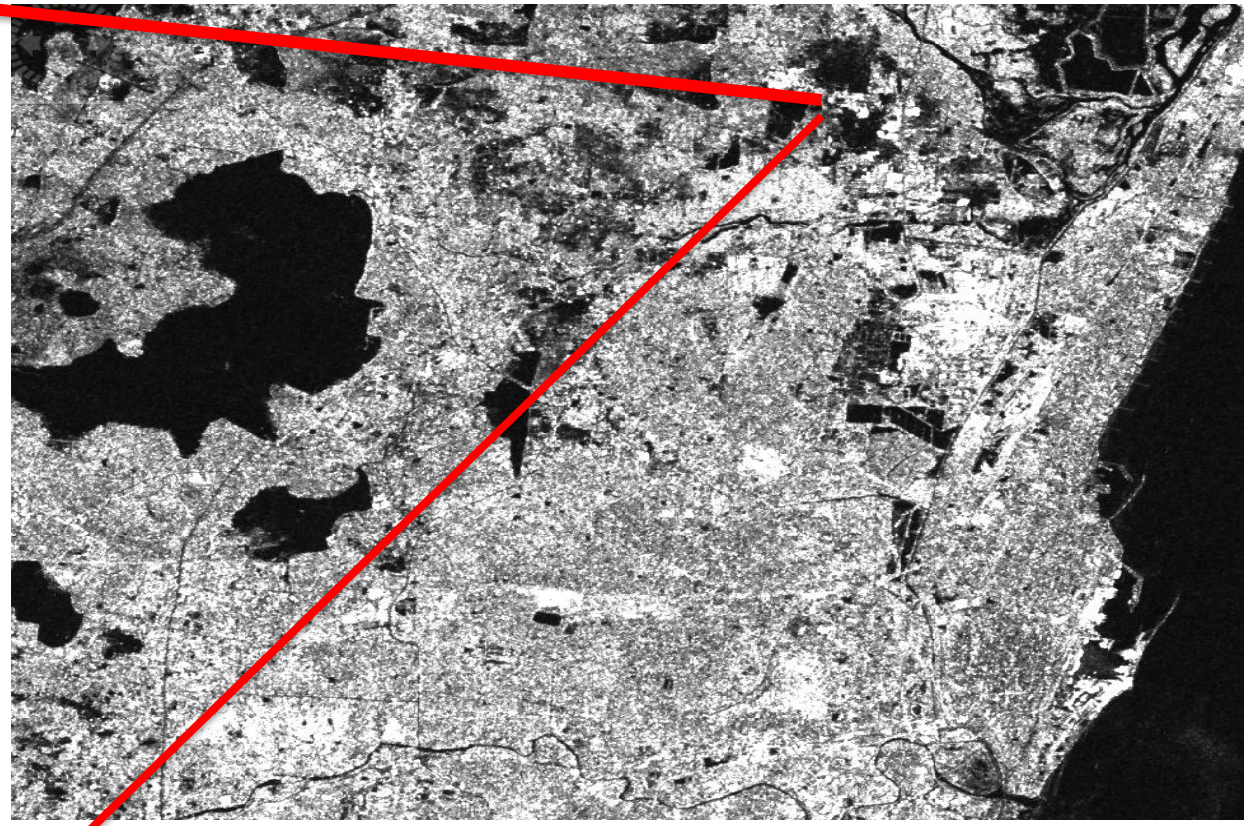
Key Idea 4: GPU acceleration [HiPC14, HPDC19]

MLR-MCL for Flood Mapping and Image Segmentation

Flood Mapping: Motivation and Challenges

- **Flood mapping**: *the process of distinguishing flooded areas from non-flooded areas* [Martinis *et al.* 2009]
- Flood mapping is very useful:
  - Guiding first response resources in a disaster situation
  - Assessing flood risk in future disaster scenarios
- Many water areas: **different sizes** and **arbitrary shapes**.
  - Fully automated methods work up to a point – not sufficient
  - Domain expert guidance can redress but:
    1. time consuming
    2. not always available
- High-resolution satellite images: require the method to be **scalable** on (10-50M+ pixels).

Enhanced Flood Mapping: Drilling Down and Rolling Up



r 3rd Depression

Image Segmentation via Graph Clustering

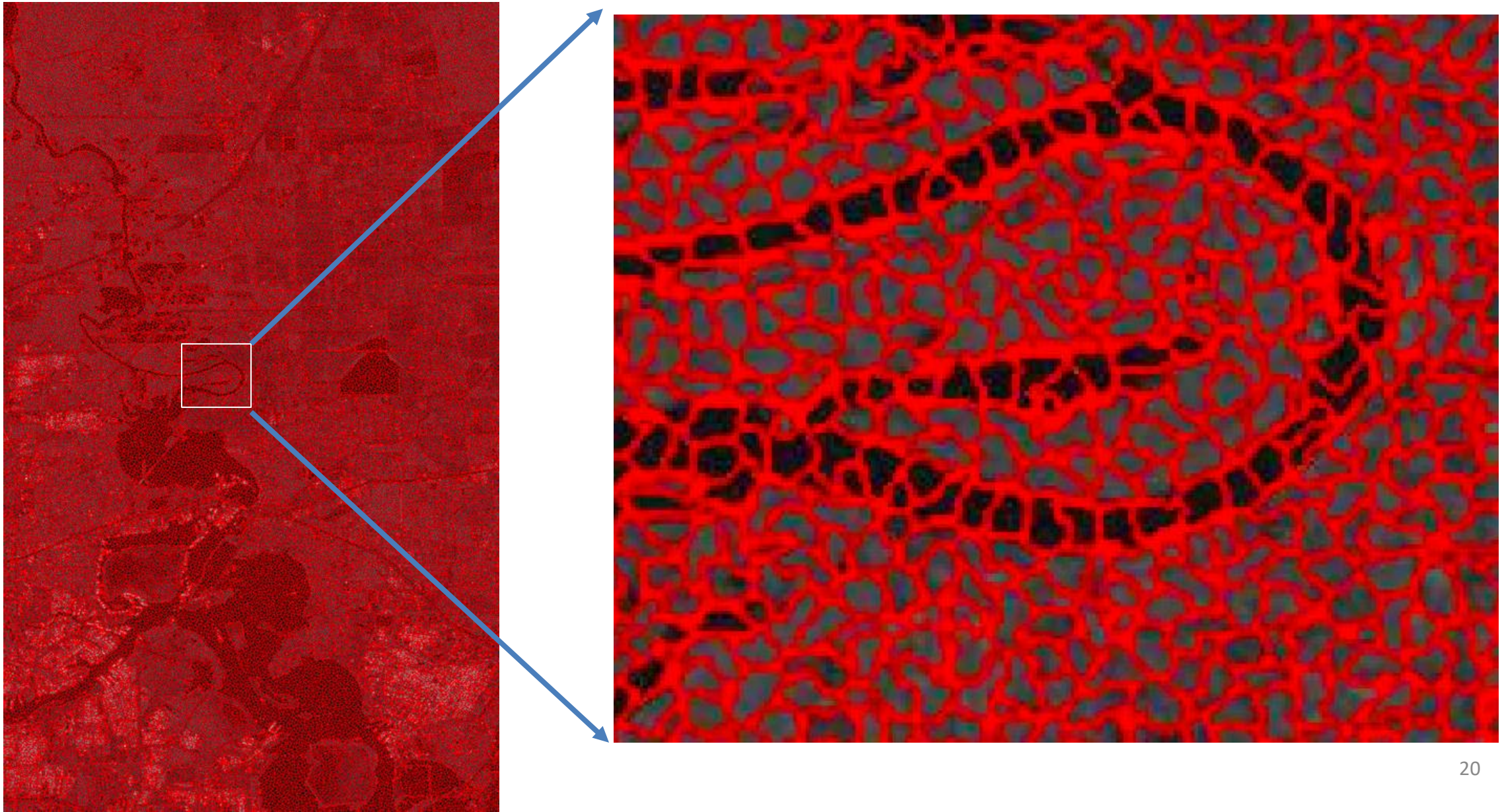
- Formulate the image as a **graph** [Cour *et al.* 2005] and use graph-based clustering approach to generate patches.

- Each pixel in the image is a node. For two nodes i and j, the weight of edge(i, j) is defined as follows:

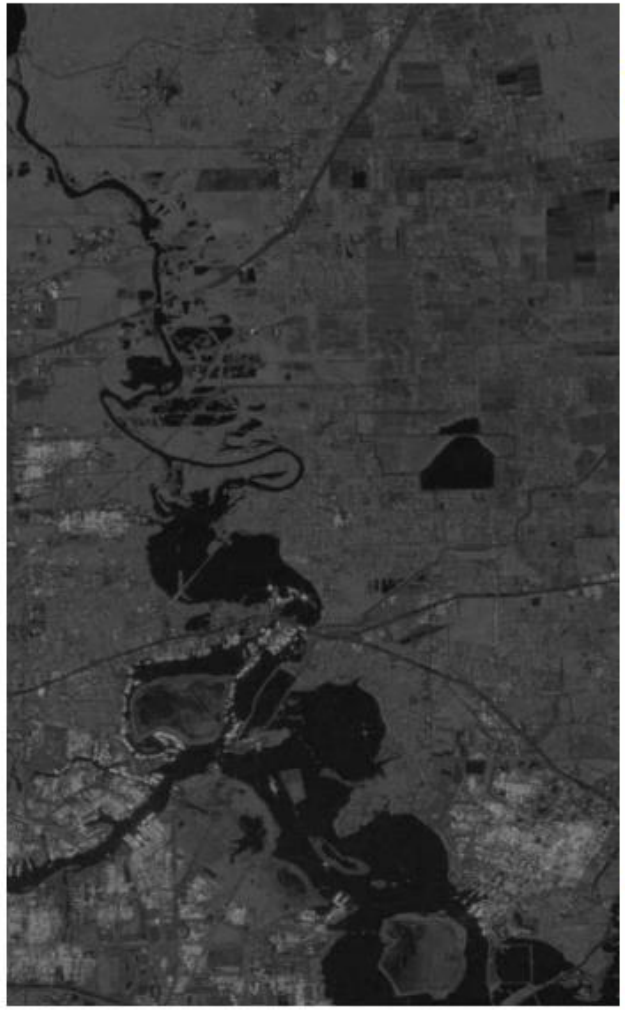
$$w_{ij} = \begin{cases} \exp\left(-\frac{\mathrm{dist}(i,j)^2}{\sigma_x^2} - \frac{\mathrm{diff}(i,j)^2}{\sigma_y^2}\right) & \text{if } \mathrm{dist}(i,j) < d_{max} \\ 0 & \text{otherwise.} \end{cases}$$

  - $\mathrm{dist}(i,j)$: geo-distance between node i and j.
  - $\mathrm{diff}(i,j)$: feature diff between node i and j (e.g., pixel intensity).
  - σ_x, σ_y and d_{max} are hyper-parameters [Cour *et al.* 2005].

MLR-MCL in action – Patch Stitching



Quantitative Results on Houston Dataset



(a) Original data

(b) HUG-FM

Method	Accuracy	F1 Score
HUG-FM (using MLR-MCL)	**0.9552**	**0.8681**
SVM	0.9451	0.8382
Planetoid	0.9445	0.8414
NORM-THR	0.9538	0.8371
Watershed algorithm	0.8904	0.6796
Otsu's thresholding	0.8977	0.7394

Quantitative evaluation on Houston dataset.

Efficiency Comparison

Method	Time
HUG-FM (using MLR-MCL)	0.057s
Otsu Thresholding	0.077s
Watershed Algorithm	0.225s
Spectral Method	538.6s
Spectral Method with postprocessing	558.2s

MLR-MCL is on average **two to three orders of magnitude** faster than vanilla MCL **while being more accurate!**

Take Home: Recent Advances in Stochastic Flow Clustering

Key Idea 1: Regularization

- Avoids fragmenting community structure

Key Idea 2: Multi-level Regularization

- Improves scalability on a single node

Key Ideas 3 &4 (not discussed) : Sparsification & GPU acceleration

Parallel processing that leverages sparse-structure

Use Case Scenario: Image Segmentation;

:Produces high quality scalable mappings with minimal human guidance.

Uses MLR-MCL as a key step. Deployed in recent hurricane/cyclone induced flood mapping.

Can we leverage similar ideas in the pathology context?

Thank you for listening