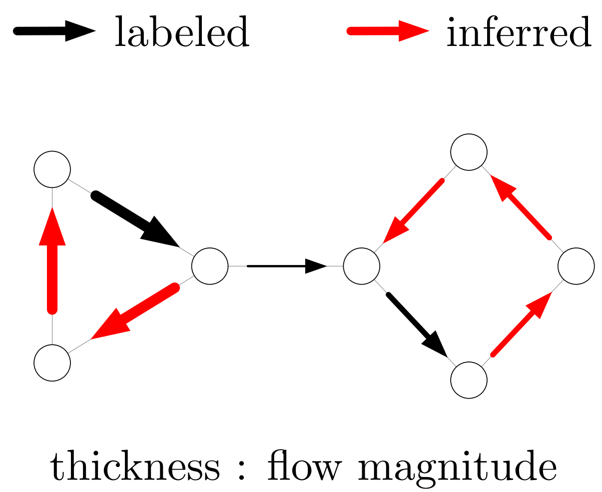
# Graph-based semi-supervised learning for edge flows

Austin R. Benson · Cornell University

NetSci HONS

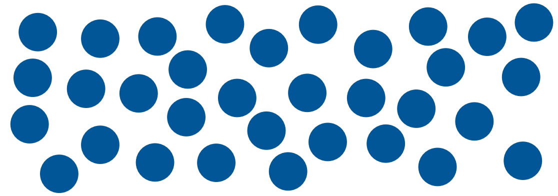
May 28, 2019

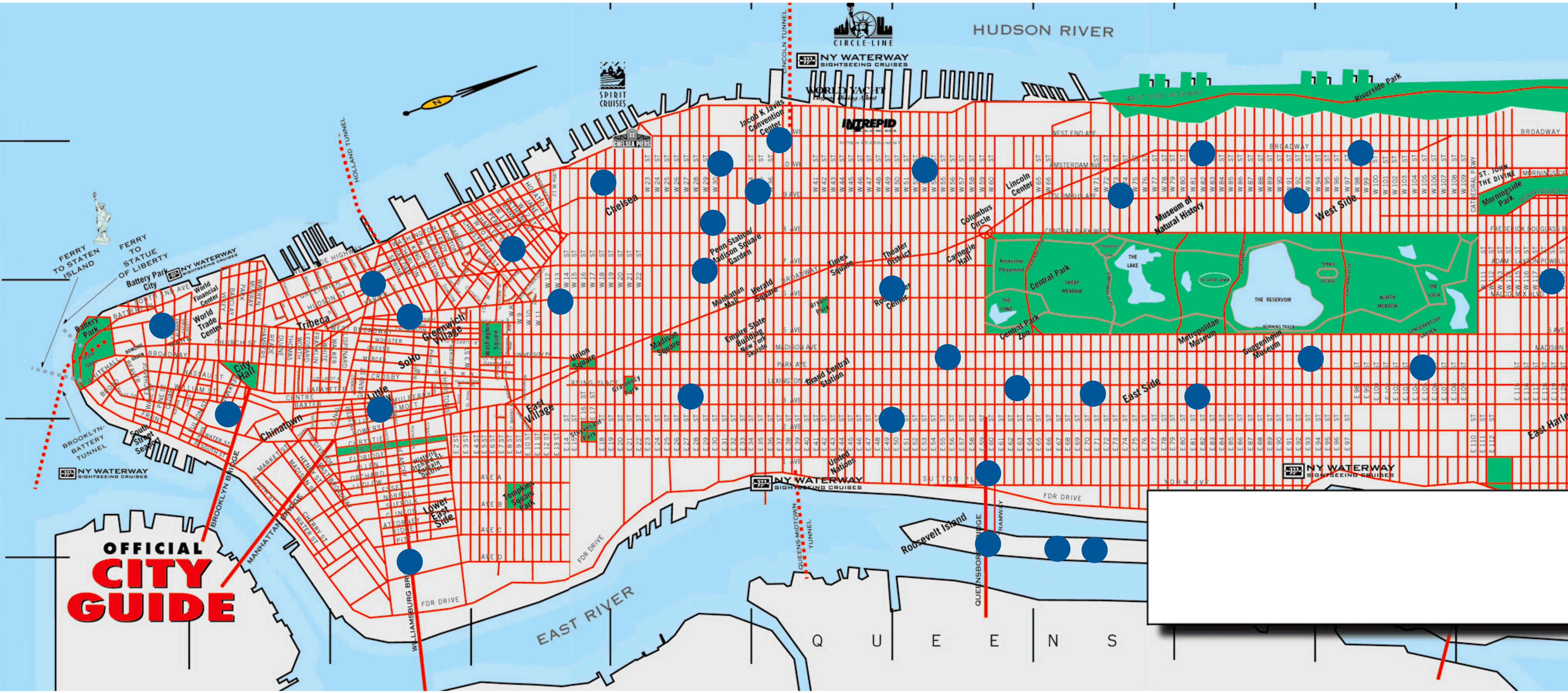
**Slides.** `bit.ly/arb-HONS-19`




Joint work with
Junteng Jia (Cornell),
Michael T. Schaub (MIT), &
Santiago Segarra (Rice)

Traffic flow sensors





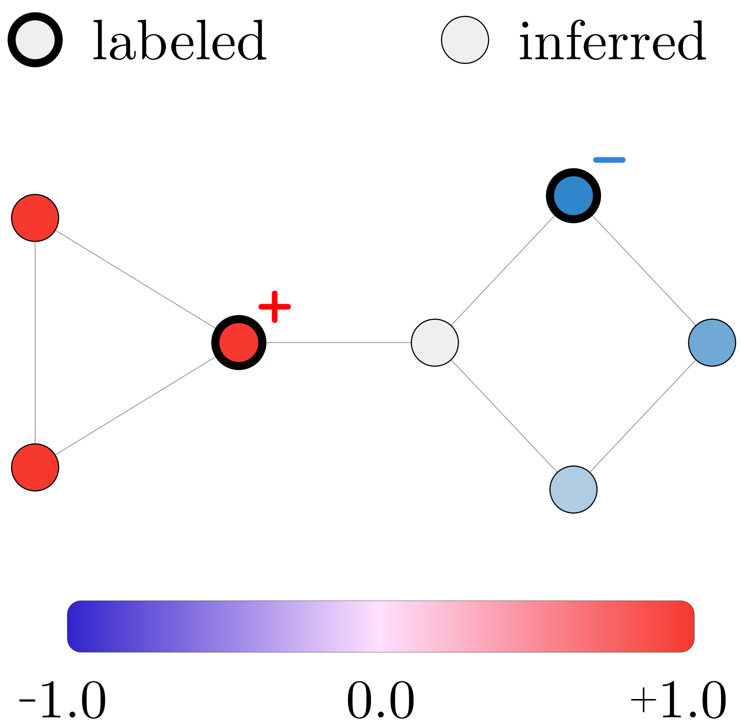

http://www.vidiani.com/road-map-of-manhattan/

# Two major questions in semi-supervised learning.

1. **Interpolation.** Given the measurements, how do I interpolate to locations where I don't know have data.

2. **Active learning.** Where are the best locations to make my measurements, knowing step 1?

# **Background.** Classical graph-based semi-supervised learning interpolates from labels on a few vertices.

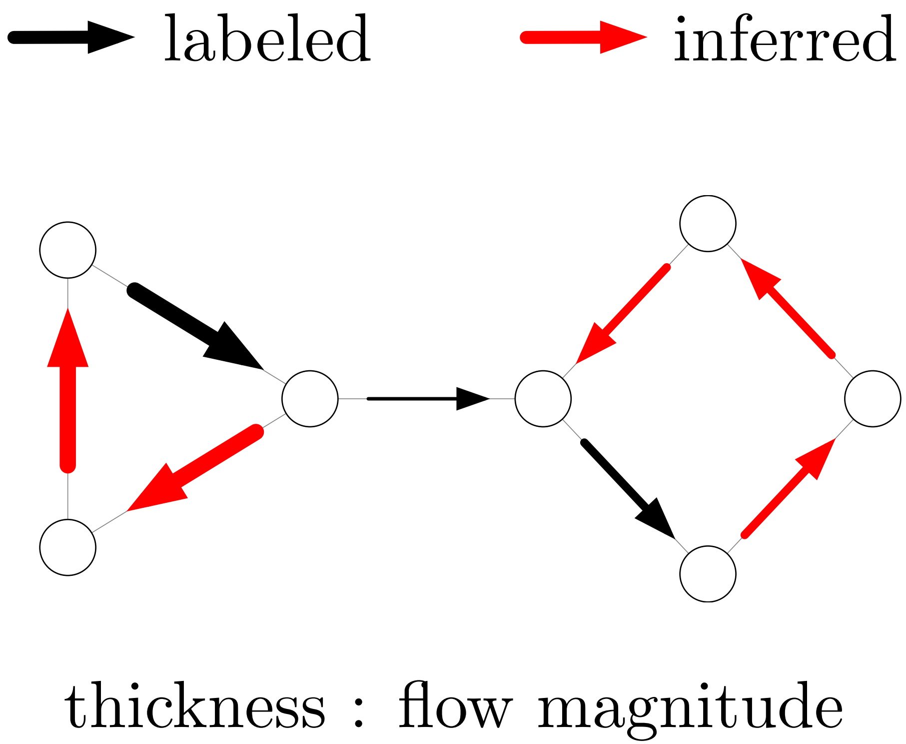



**Key idea.** [Zhu+ 03]
My label is similar to the labels of my connections.

$$\underset{\text{labels } \mathbf{x}}{\text{minimize}} \quad \sum_{(i,j)\in E} (x_i - x_j)^2$$

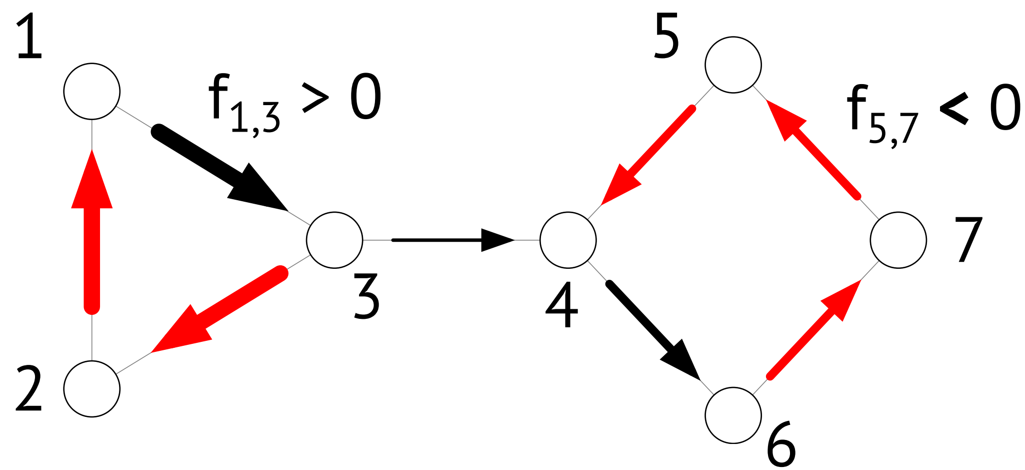
$$\text{subject to} \quad \mathbf{x} \text{ matches given labels}$$

# In the higher-order case of edge flows, we have a different type of objective.



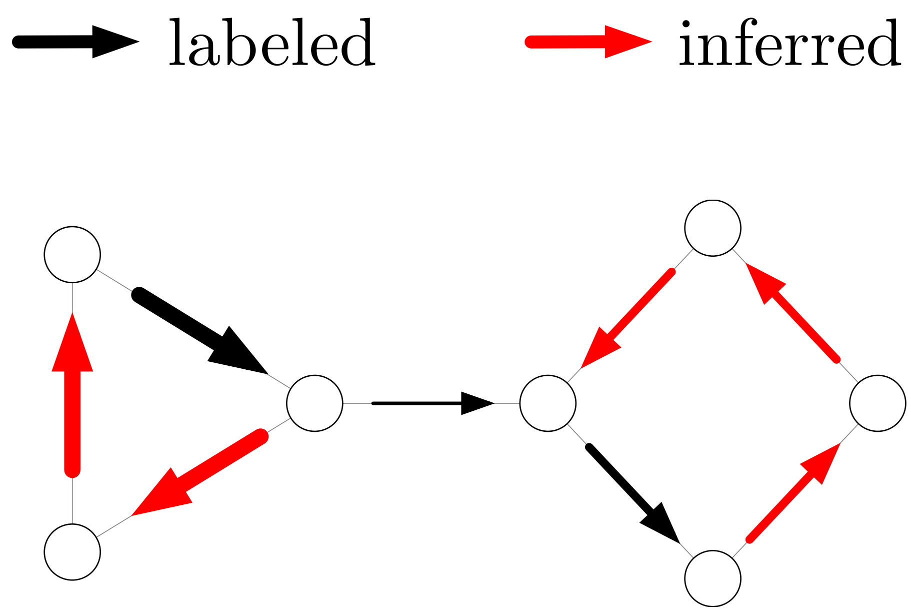

**Key idea ("divergence-free").**
Net flow into a node should be similar to net flow out of a node.

# An edge flow represents net flow along an edge.




- As an alternating function: F(i, j) = -F(j, i)
- For the linear algebra, first orient each edge $i \to j$ if $i < j$. Then vector **f** gives flows on these oriented edges.
- If $f_{i,j} > 0$, if net flow aligns with orientation
- If $f_{i,j} < 0$, net flow is opposite of orientation.

# In the higher-order case of edge flows, we have a different type of objective.



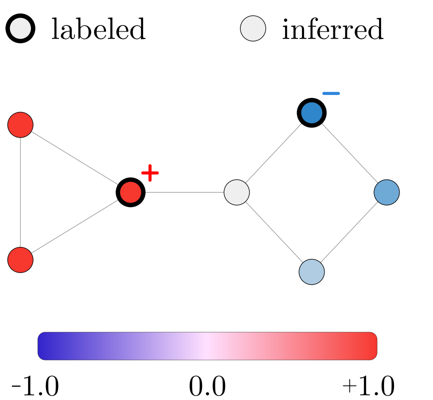

thickness : flow magnitude

**Key idea ("divergence-free").**
Net flow into a node should be similar to net flow out of a node.

$$\underset{\text{flows } \mathbf{f}}{\text{minimize}} \quad \sum_i \left[ \sum_{j>i,(i,j)\in E} f_{ij} - \sum_{k<i,(k,i)\in E} f_{ki} \right]^2$$

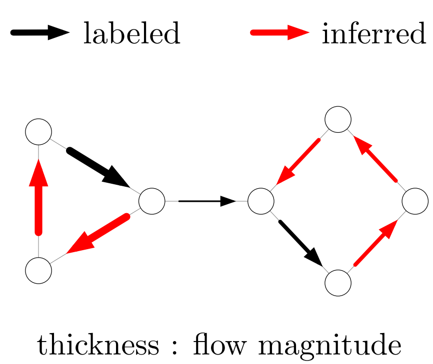
subject to **f** matches labels

# There is a close relationship between node-based SSL and edge-based SSL objective functions.




$$\sum_{(i,j)\in E} (x_i - x_j)^2 = \mathbf{x}^T \boldsymbol{L} \mathbf{x} = \mathbf{x}^T \boldsymbol{B}\boldsymbol{B}^T \mathbf{x} = \|\boldsymbol{B}^T \mathbf{x}\|_2^2$$

$$B_{k,(i,j)} = \begin{cases} 1 & k = i, i < j \\ -1 & k = j, i < j \\ 0 & \text{otherwise} \end{cases}$$

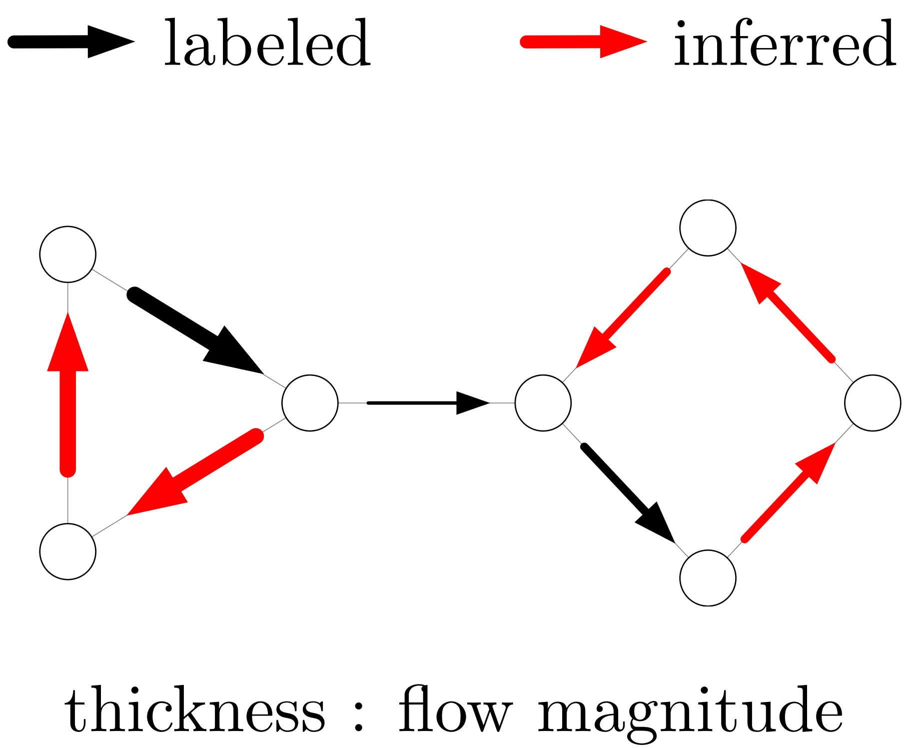



$$\sum_i \left[ \sum_{j>i,(i,j)\in E} f_{ij} - \sum_{k<i,(k,i)\in E} f_{ki} \right]^2 = \mathbf{f}^T \boldsymbol{B}^T \boldsymbol{B}\mathbf{f} = \|\boldsymbol{B}\mathbf{f}\|_2$$

**One catch.**

- Having labels in node case gives unique answer.
- Having labels in edge case is under-constrained.

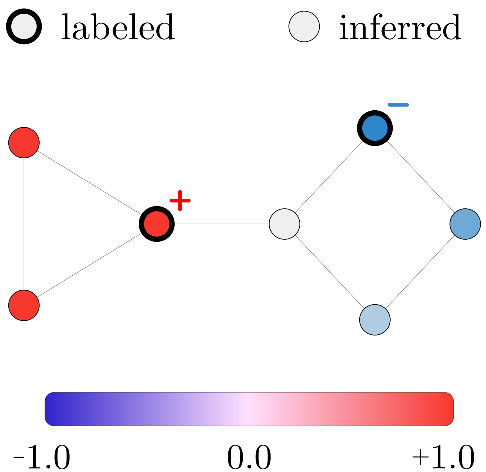
# We add regularization to get a nice sparse linear least squares problem.




$$\underset{\text{flows } \mathbf{f}}{\text{minimize}} \quad \|\boldsymbol{B}\mathbf{f}\|_2^2 + \lambda\|\mathbf{f}\|_2^2$$

subject to $\mathbf{f}$ matches labels

- We use iterative solvers LSQR or LSMR to compute the solution efficiently.



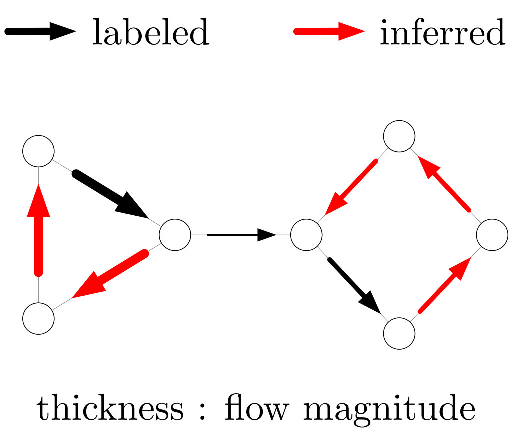

## Key Idea

My label is similar to the labels of my connections.

## Objective

$$\underset{\text{vertex values } \mathbf{x}}{\text{minimize}} \quad \|\boldsymbol{B}^T\mathbf{x}\|_2^2$$

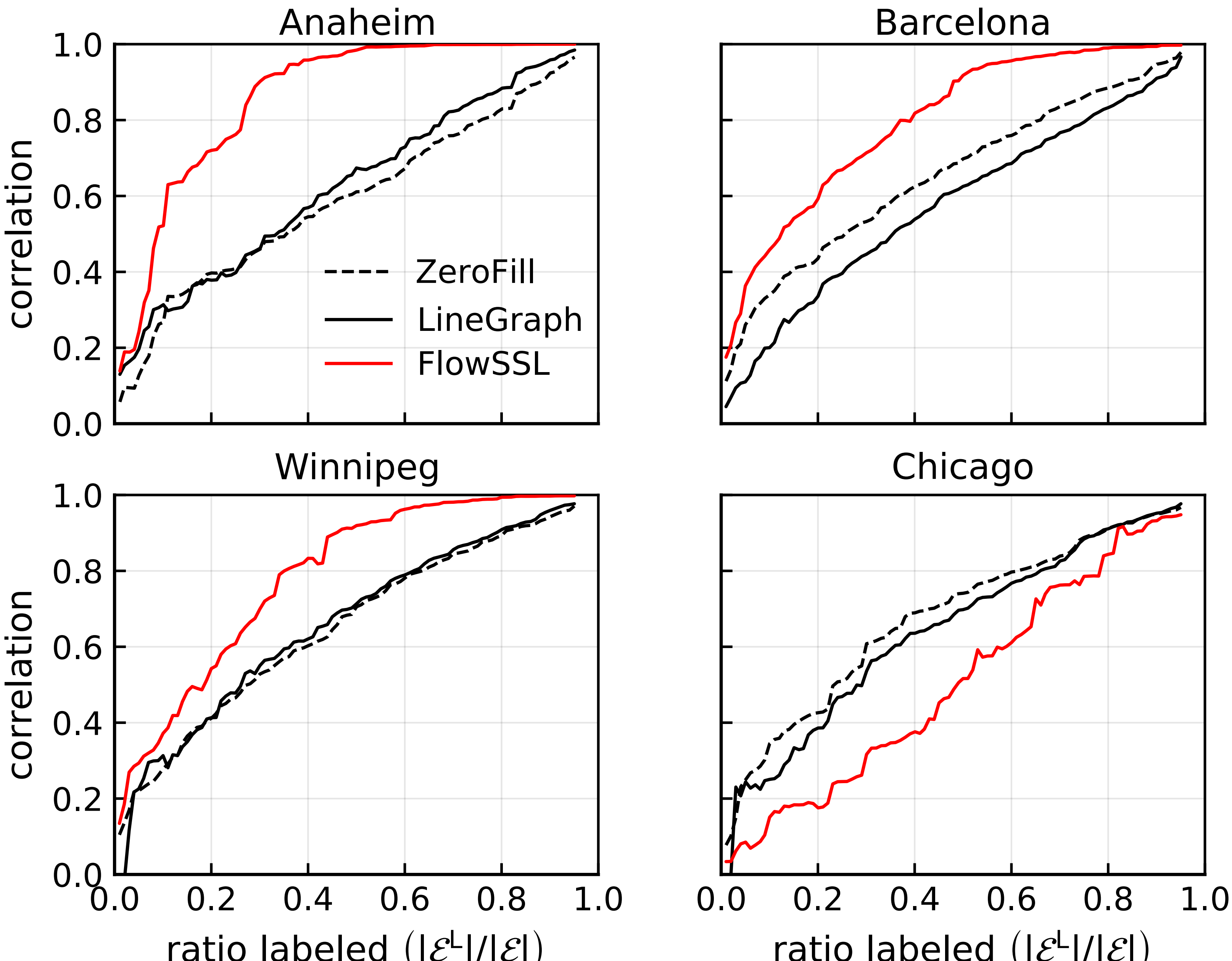
subject to $\mathbf{x}$ matches labels



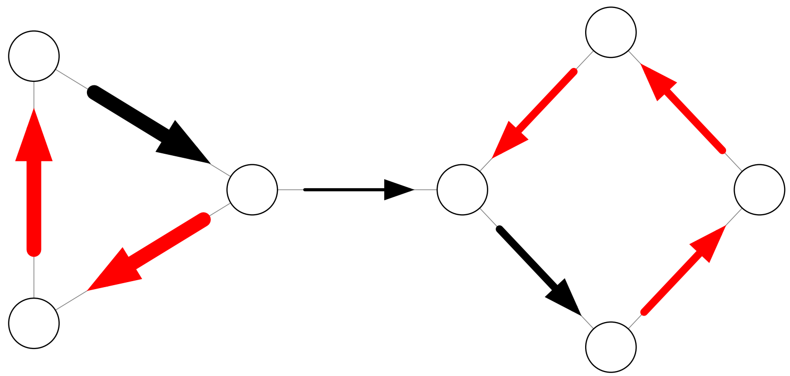

Net in flow = net out flow at all nodes.

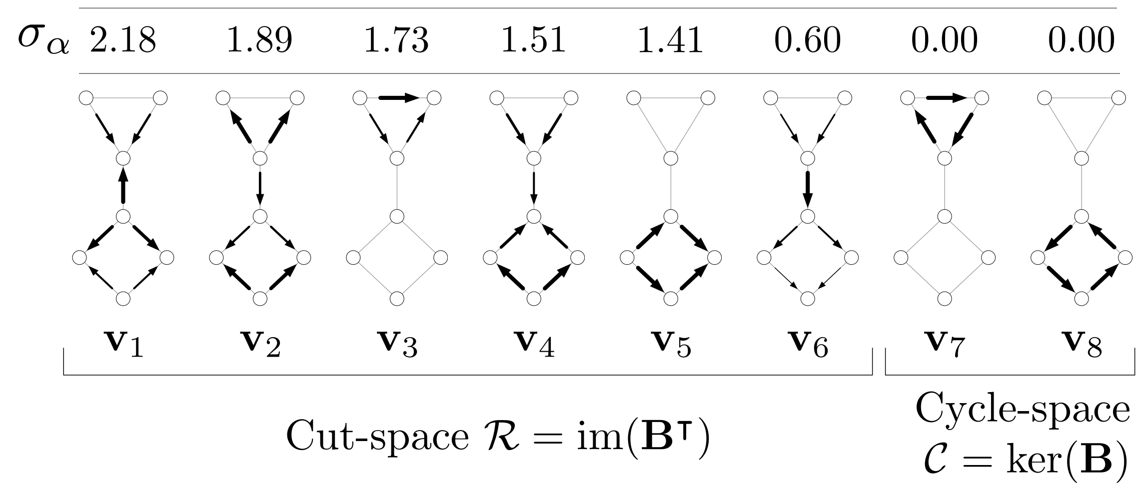
$$\underset{\text{edge flows } \mathbf{f}}{\text{minimize}} \quad \|\boldsymbol{B}\mathbf{f}\|_2^2 + \lambda\|\mathbf{f}\|_2^2$$

subject to $\mathbf{f}$ matches labels


Anaheim
Barcelona
Winnipeg
Chicago
correlation
0.0
0.2
0.4
0.6
0.8
1.0
ZeroFill
LineGraph
FlowSSL
ratio labeled $(|\mathcal{E}^{\mathrm{L}}|/|\mathcal{E}|)$

# Why do we do so poorly on Chicago?

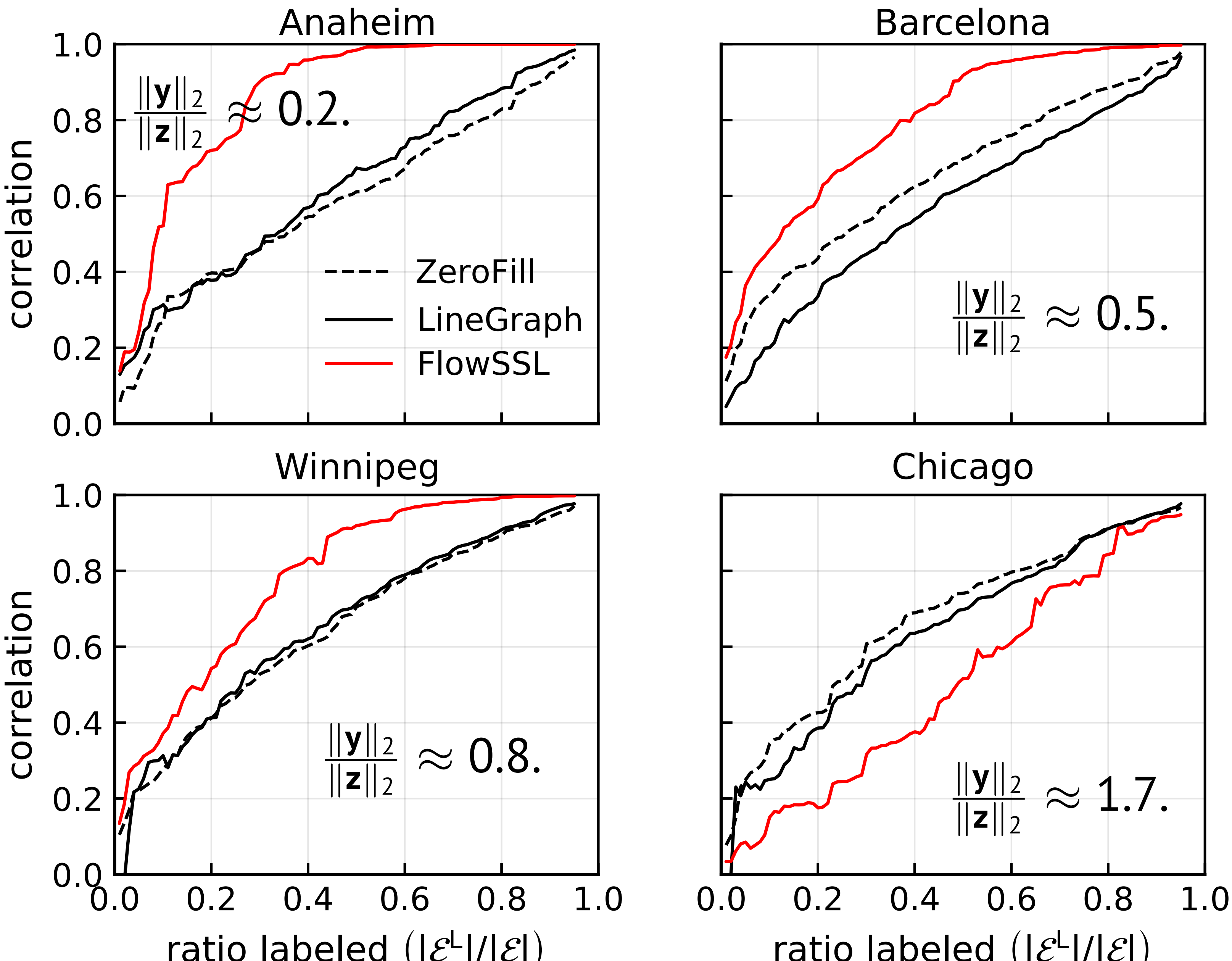





$\mathbf{f}_{\text{truth}} = \mathbf{y} \oplus \mathbf{z}, \ \mathbf{y} \in \mathcal{R}, \ \mathbf{z} \in \mathcal{C}$

Our divergence-free assumption says that $\mathbf{f}_{\text{truth}} \approx \mathbf{z}$.

Does this actually hold in our data?



Anaheim
Barcelona
Winnipeg
Chicago
correlation
ratio labeled $(|\mathcal{E}^L|/|\mathcal{E}|)$
ZeroFill
LineGraph
FlowSSL
$\frac{\|\mathbf{y}\|_2}{\|\mathbf{z}\|_2} \approx 0.2.$
$\frac{\|\mathbf{y}\|_2}{\|\mathbf{z}\|_2} \approx 0.5.$
$\frac{\|\mathbf{y}\|_2}{\|\mathbf{z}\|_2} \approx 0.8.$
$\frac{\|\mathbf{y}\|_2}{\|\mathbf{z}\|_2} \approx 1.7.$

# We have some theoretical guarantees if the true flow is indeed nearly divergence-free.

$$\underset{\text{edge flows } \mathbf{f}}{\text{minimize}} \quad \|\boldsymbol{B}\mathbf{f}\|_2^2 + \lambda\|\mathbf{f}\|_2^2$$

$$\text{subject to} \quad \mathbf{f} \text{ matches labels}$$

**Theorem.**

- Let $\boldsymbol{V_C}$ be a basis for the divergence-free space $\ker(\boldsymbol{B})$.
- Suppose the true flow is a divergence-free flow perturbed by $\mathbf{d}$.
- For any $m - n + 1$ labels corresponding to linearly independent rows of $\boldsymbol{V_C}$, denoted $\boldsymbol{V}_C^L$, the relative reconstruction error is bounded by

$$\left[\sigma_{\min}^{-1}(\boldsymbol{V}_C^L) + 1\right] \cdot \|\mathbf{d}\|_2$$

# Two major questions in semi-supervised learning.

1. **Interpolation.** Given the measurements, how do I interpolate to locations where I don't know have data.

2. **Active learning.** Where are the best locations to make my measurements, knowing step 1?

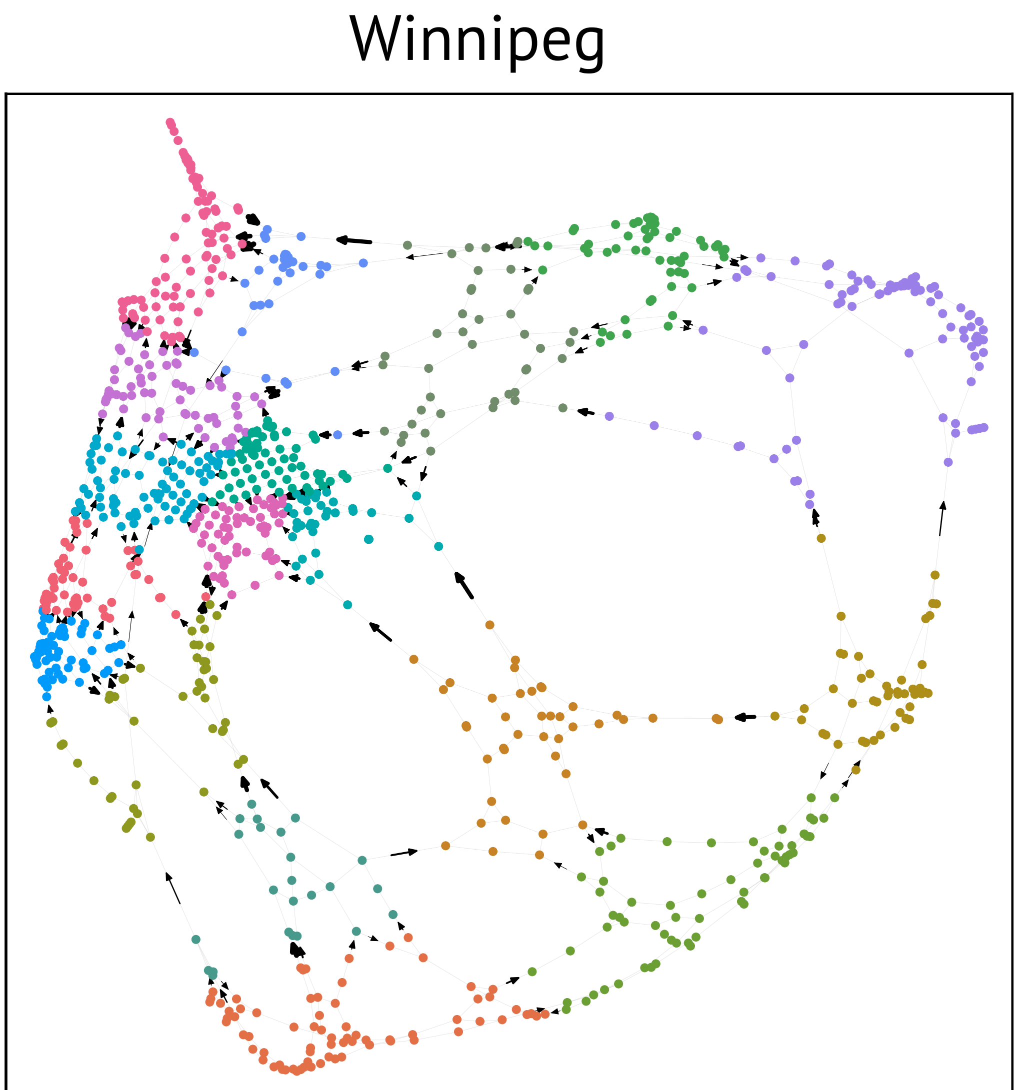
# Active learning strategies in vertex-based and edge-based SSL are similar.

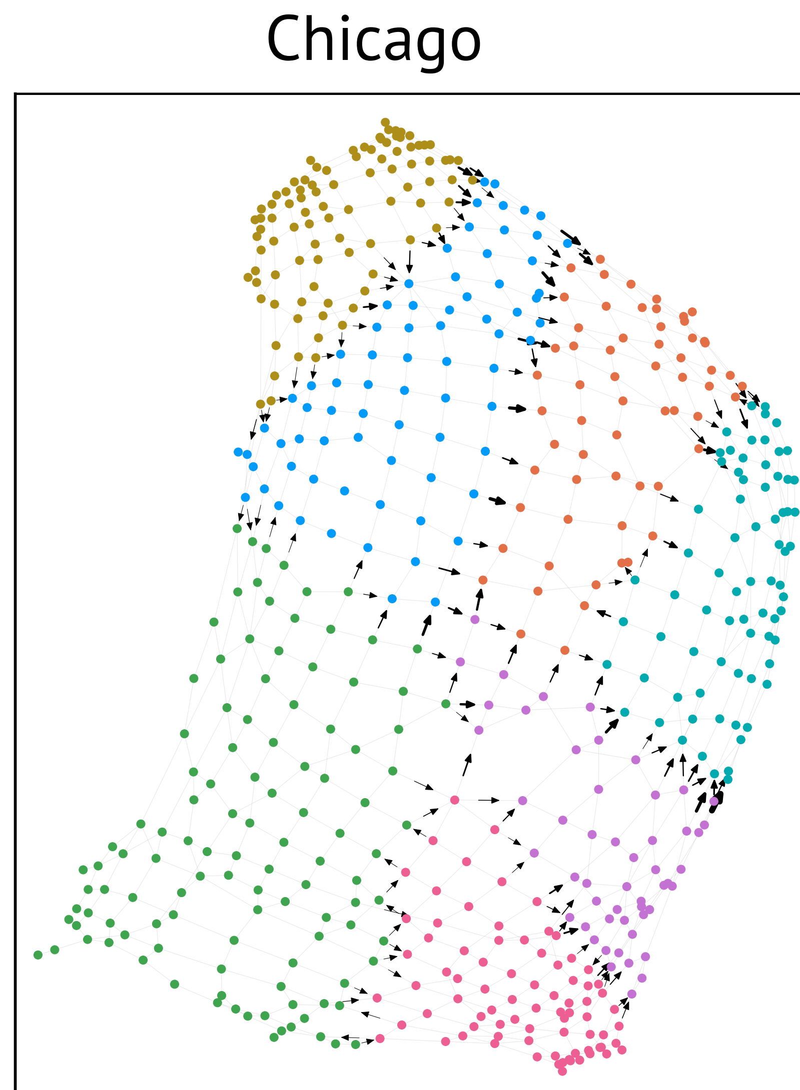
**Active vertex-based SSL** [Guillory-Bilmes 17]

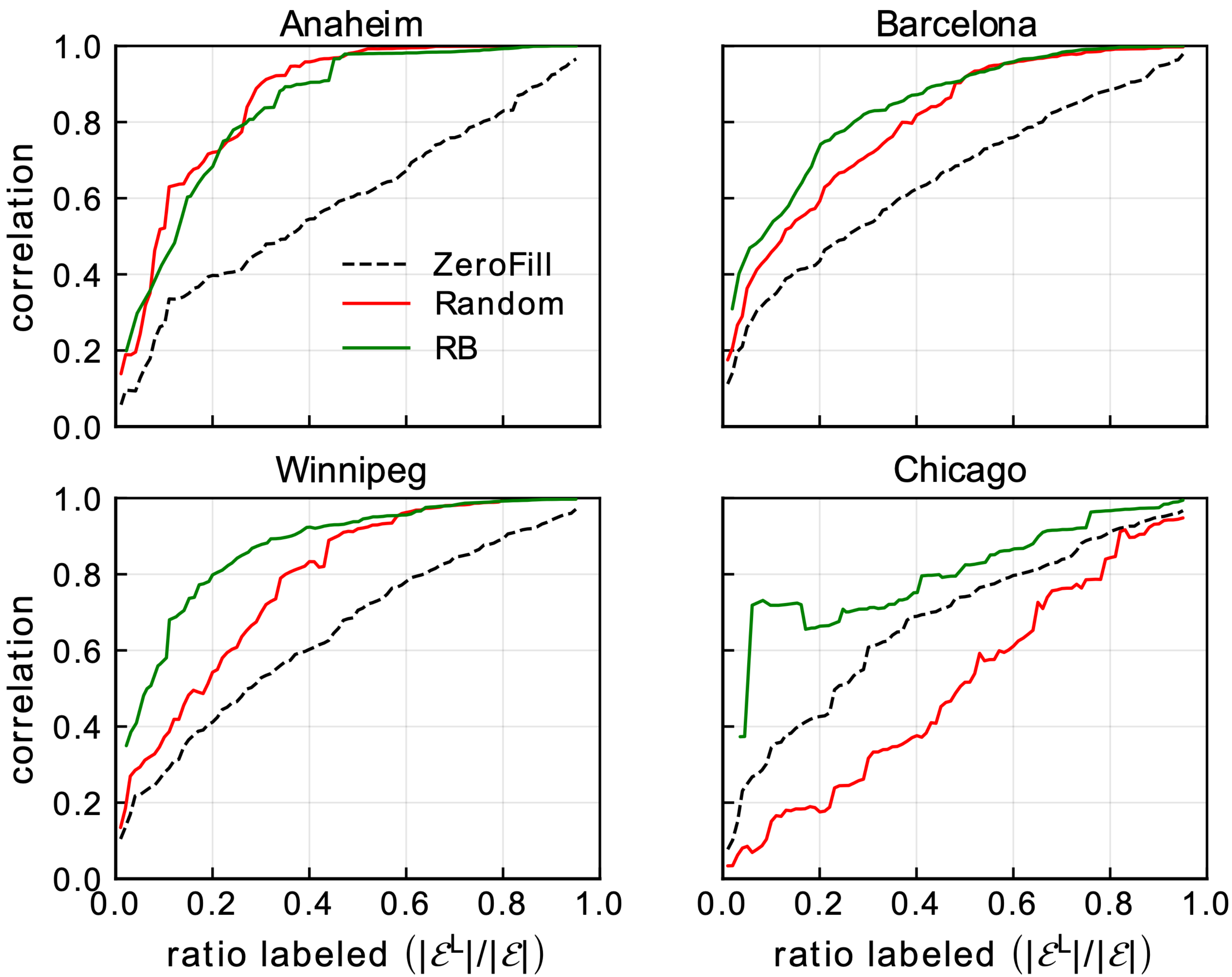
1. Cluster the graph (e.g., using spectral clustering).
2. Pick points from each cluster.

**Active edge-based SSL**

1. Cluster the graph (e.g., using spectral clustering).
2. Pick edges that cross cluster boundaries.


Winnipeg

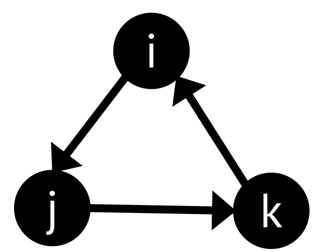

Chicago


Anaheim
Barcelona
Winnipeg
Chicago
correlation
ratio labeled $(|\mathcal{E}^L|/|\mathcal{E}|)$
ZeroFill
Random
RB
0.0
0.2
0.4
0.6
0.8
1.0

# We can extend our framework beyond looking for divergence-free flows.

Hodge decomposition [Lim 15, others]

$$\underbrace{\mathbf{f}}_{\text{edge flow}} = \overbrace{\mathbf{B}^{T}\mathbf{y}}^{\text{gradient flow}} \oplus \overbrace{\underbrace{\mathbf{Cw}}_{\text{curl flow}} \oplus \underbrace{\mathbf{h}}_{\text{harmonic flow}}}^{\text{divergence-free flow}}$$




$f_{ij} + f_{jk} + f_{ki}$

- So far, we have penalized gradient flow via $\|\boldsymbol{B}\mathbf{f}\|_2^2$.
- Could penalize other types of flows.

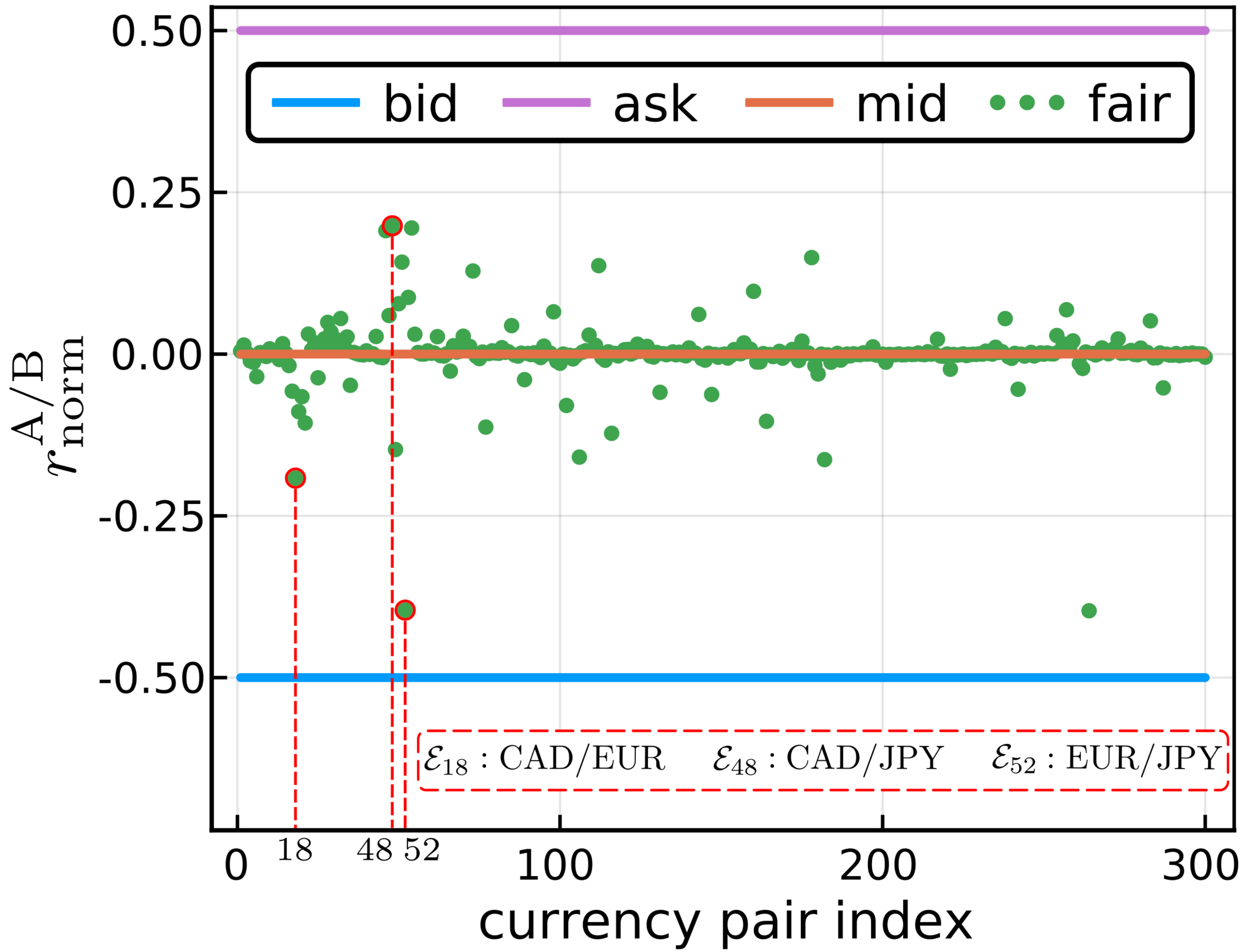
# We can extend our framework beyond looking for divergence-free flows.

- Data is currency exchange rates (fully connected graphs).
- Buyers willing to buy at "bid" price.
- Sellers willing to sell at "ask" price.
- Settle on some price in the middle (usually mid point).
- Want prices that have no cyclic flow (arbitrage).

$$\mathbf{f}^* = \underset{\text{flows } \mathbf{f}}{\text{argmin}} \quad \|\mathbf{C}^\mathsf{T}\mathbf{f}\|^2 + \lambda^2 \cdot \|\mathbf{f} - \mathbf{f}^{\text{mid}}\|^2$$

$$\text{subject to} \quad \mathbf{f}^{\text{bid}} \leq \mathbf{f} \leq \mathbf{f}^{\text{ask}}$$

# Optimal flows eliminate arbitrage opportunities.



# Graph-based semi-supervised learning for edge flows.

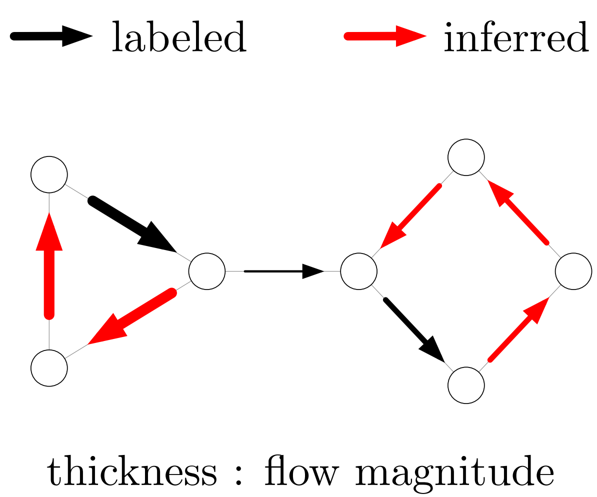
- We have a framework for semi-supervised learning in the *edge space* with a natural connection to classical vertex-based SSL.

- We also have a practical and efficient active learning method.

- Can extend the SSL framework to other types of edge flows through results in combinatorial Hodge theory.

# Graph-based semi-supervised learning for edge flows

To appear in KDD 2019.




**THANKS!** Austin R. Benson
**Slides.** bit.ly/arb-HONS-19
http://cs.cornell.edu/~arb
@austinbenson
arb@cs.cornell.edu

bit.ly/ssl-flow-code
(code, reproducibility, and data)