# Minimum Cost Flows and the Graph Connection Laplacian

Sawyer Robertson[1,2], Dhruv Kohli[2], Alexander Cloninger[2,3], Gal Mishne[3]

July 10, 2024

Network Science Beyond Graphs

2024 SIAM Conference on Discrete Mathematics

[1]Presenting author

[2]Department of Mathematics, UCSD

[3]Halıcıoğlu Data Science Institute, UCSD

# Background

## Key Themes

1. As we have seen many times, a common framework for understanding **large** or **complex** datasets is through graph representations obtained through various means and pipelines.

# Key Themes

1. As we have seen many times, a common framework for understanding **large** or **complex** datasets is through graph representations obtained through various means and pipelines.

2. Sometimes it's **just not enough** to encode adjacency (and thereby losing features or coordinates); or the adjacency structure isn't super useful for what we're trying to learn (e.g., webpage datasets, "low homophily").

# Key Themes

1. As we have seen many times, a common framework for understanding **large** or **complex** datasets is through graph representations obtained through various means and pipelines.

2. Sometimes it's **just not enough** to encode adjacency (and thereby losing features or coordinates); or the adjacency structure isn't super useful for what we're trying to learn (e.g., webpage datasets, "low homophily").

3. The **Graph Connection Laplacian** is a sort of manifold-aware Laplacian matrix which allows one to encode local information about deviations of features and/or embedding coordinates directly into the graph structure.

## Key Themes

1. As we have seen many times, a common framework for understanding **large** or **complex** datasets is through graph representations obtained through various means and pipelines.

2. Sometimes it's **just not enough** to encode adjacency (and thereby losing features or coordinates); or the adjacency structure isn't super useful for what we're trying to learn (e.g., webpage datasets, "low homophily").

3. The **Graph Connection Laplacian** is a sort of manifold-aware Laplacian matrix which allows one to encode local information about deviations of features and/or embedding coordinates directly into the graph structure.

4. Lately we have been particularly interested in some new and interesting **foundational theory** for this setting, motivated simultaneously by its deep roots in algebraic graph theory and some intriguing new directions that have opened up in the graph neural network literature.

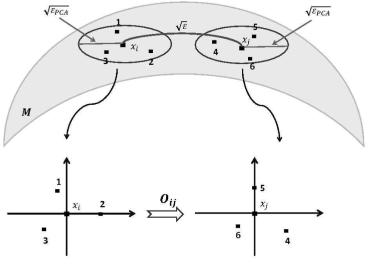
# Connection Graphs: Two Ways

- Given some data $X \in \mathbb{R}^{n \times p}$ where typically $p \gg n$,
  1. For each $x_i$, $x_j$, add $\{i, j\}$ to the graph when $\|x_i - x_j\|$ is small,
  2. For each $x_i \in \mathbb{R}^p$, apply PCA to the features in the neighborhood of $x_i$ to get a "**local view**," and thereby reduce the features to a common dimension $d$ for all nodes,
  3. for each $\{i, j\}$ try to best align the reduced features via a Procrustes problem and obtain a rotation matrix $O_{ij} \in \mathrm{O}(d)$.

[4] Amit Singer and H-T Wu. "Vector diffusion maps and the connection Laplacian". In: *Communications on pure and applied mathematics* 65.8 (2012).

# Connection Graphs: Two Ways

- Given some data $X \in \mathbb{R}^{n \times p}$ where typically $p \gg n$,
  1. For each $x_i$, $x_j$, add $\{i, j\}$ to the graph when $\|x_i - x_j\|$ is small,
  2. For each $x_i \in \mathbb{R}^p$, apply PCA to the features in the neighborhood of $x_i$ to get a "**local view**," and thereby reduce the features to a common dimension $d$ for all nodes,
  3. for each $\{i, j\}$ try to best align the reduced features via a Procrustes problem and obtain a rotation matrix $O_{ij} \in \mathrm{O}(d)$.
- Thus obtain both a **proximity-based graph representation** of the data $G = (V, E, w)$, **as well as a map** $\sigma : E \to \mathrm{O}(d)$.[4]




[4] Singer and Wu, "Vector diffusion maps and the connection Laplacian".

# Connection Graphs: Two Ways

- Another perspective is a bit more layered.
- **Signed graphs** associate a $\pm 1$ value to each edge in a given graph; these data back to the 1950s[5], and arise in, e.g., social network models;

[5] Dorwin Cartwright and Frank Harary. "Structural balance: a generalization of Heider's theory.". In: *Psychological review* 63.5 (1956).

[6] Elliott H Lieb and Michael Loss. "Fluxes, Laplacians, and Kasteleyn's theorem". In: *Statistical Mechanics: Selecta of Elliott H. Lieb*. Springer, 1993.

[7] Singer and Wu, "Vector diffusion maps and the connection Laplacian".

## Connection Graphs: Two Ways

- Another perspective is a bit more layered.
- **Signed graphs** associate a $\pm 1$ value to each edge in a given graph; these data back to the 1950s[5], and arise in, e.g., social network models;
- **Magnetic Graphs** associate a $U(1)$ value to each edge in a given graph; these appeared in the early 1990s[6] and have found use in **lots** of GNN papers;

[5] Cartwright and Harary, "Structural balance: a generalization of Heider's theory."
[6] Lieb and Loss, "Fluxes, Laplacians, and Kasteleyn's theorem".
[7] Singer and Wu, "Vector diffusion maps and the connection Laplacian".

## Connection Graphs: Two Ways

- Another perspective is a bit more layered.
- **Signed graphs** associate a $\pm 1$ value to each edge in a given graph; these data back to the 1950s[5], and arise in, e.g., social network models;
- **Magnetic Graphs** associate a $U(1)$ value to each edge in a given graph; these appeared in the early 1990s[6] and have found use in **lots** of GNN papers;
- **Connection Graphs**, as we have seen, associate a $\mathrm{O}(d)$ value to each edge in a given graph; these originated in the early 2010s[7] and have been used in the Cryo EM problem, as well as Sheaf neural networks
- All of these are instances of voltage graphs (which consider the umbrella case of a general group), and there is some interest from algebraic graph theorists here

[5] Cartwright and Harary, "Structural balance: a generalization of Heider's theory."
[6] Lieb and Loss, "Fluxes, Laplacians, and Kasteleyn's theorem".
[7] Singer and Wu, "Vector diffusion maps and the connection Laplacian".

# Some important matrices

- Given $G$ a connected, finite, undirected graph; A **connection** is any map $\sigma : E \to \mathrm{O}(d)$, and a pair $(G, \sigma)$ is called a connection graph.

# Some important matrices

- Given $G$ a connected, finite, undirected graph; A **connection** is any map $\sigma : E \to \mathrm{O}(d)$, and a pair $(G, \sigma)$ is called a connection graph.

## Connection Incidence Matrix

The **connection graph incidence matrix** is given by the $nd \times md$ block matrix:

$$B = (B_{ie} \in \mathbb{R}^{d \times d})_{i \in V, e \in E}, B_{ie} = \begin{cases} I_d & \text{if } e = (i, \cdot) \\ -\sigma_e^T & \text{if } e = (\cdot, i) \\ 0_d & \text{otherwise} \end{cases}$$

# Some important matrices

- Given $G$ a connected, finite, undirected graph; A **connection** is any map $\sigma : E \to \mathrm{O}(d)$, and a pair $(G, \sigma)$ is called a connection graph.

## Connection Incidence Matrix

The **connection graph incidence matrix** is given by the $nd \times md$ block matrix:

$$B = (B_{ie} \in \mathbb{R}^{d \times d})_{i \in V, e \in E}, B_{ie} = \begin{cases} I_d & \text{if } e = (i, \cdot) \\ -\sigma_e^T & \text{if } e = (\cdot, i) \\ 0_d & \text{otherwise} \end{cases}$$

## Connection Laplacian Matrix

If $(G = (V, E), \sigma)$ is a connection graph, the **connection Laplacian matrix** is the $nd \times nd$ block matrix:

$$L = (L_{ij} \in \mathbb{R}^{d \times d})_{i,j=1}^n, \quad L_{ij} = \begin{cases} d_i I_d & \text{if } i = j \\ -w_{ij}\sigma_{ij} & \text{if } i \sim j \\ 0_d & \text{otherwise} \end{cases}.$$

($L = BWB^T$ where $W$ is a block-diagonal matrix of edge weights).

- $L$ is positive semidefinite, symmetric, and again has a full spectral decomposition.

# A sampling of results I

- The **synchronization problem** can be realized as an optimization problem for vector fields on connection graphs.

$$\inf_{f:V\to\mathbb{S}^{d-1}} \sum_{(i,j)\in E} \|f(i) - \sigma_{ij} f(j)\|_2^2$$

$$\inf_{F:V\to O(d)} \sum_{(i,j)\in E} \|F(i) - \sigma_{ij} F(j)\|_F^2$$

- Think: how best can we **globally align** our **local views**? Either with vectors (the $\mathbb{S}^{d-1}$ case), or entire frames (the $O(d)$ case).

[8] Afonso S Bandeira, Amit Singer, and Daniel A Spielman. "A Cheeger inequality for the graph connection Laplacian". In: *SIAM Journal on Matrix Analysis and Applications* 34.4 (2013).

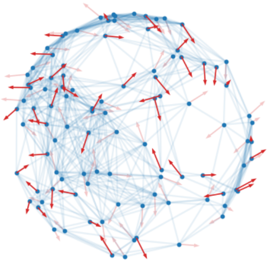
# A sampling of results I

- The **synchronization problem** can be realized as an optimization problem for vector fields on connection graphs.

$$\inf_{f:V\to\mathbb{S}^{d-1}} \sum_{(i,j)\in E} \|f(i) - \sigma_{ij} f(j)\|_2^2$$

$$\inf_{F:V\to \mathrm{O}(d)} \sum_{(i,j)\in E} \|F(i) - \sigma_{ij} F(j)\|_F^2$$

- Think: how best can we **globally align** our **local views**? Either with vectors (the $\mathbb{S}^{d-1}$ case), or entire frames (the $\mathrm{O}(d)$ case).
- Bandeira, Singer, and Spielman[8] obtain Cheeger-type inequalities which relate the optimal synchronization values to the spectral gap of $L$, and provide spectral approximations via linearly relaxed versions



[8] Bandeira, Singer, and Spielman, "A Cheeger inequality for the graph connection Laplacian".

# A sampling of results II

- Effective resistance on connection graphs has also received a bit of attention.
- Recall that for nodes $i, j \in V$, the effective resistance $r_{ij} = (\delta_i - \delta_j)^T L^\dagger (\delta_i - \delta_j)$ defines a metric on the nodes which can be used for sparsification, nodal embeddings, and beyond.

[9] Fan Chung, Wenbo Zhao, and Mark Kempton. "Ranking and sparsifying a connection graph". In: *Internet Mathematics* 10.1-2 (2014).

[10] Alex Cloninger et al. "Random Walks, Conductance, and Resistance for the Connection Graph Laplacian". In: *SIAM Journal on Matrix Analysis and Applications, To Appear.* (2023).

# A sampling of results II

- Effective resistance on connection graphs has also received a bit of attention.
- Recall that for nodes $i, j \in V$, the effective resistance $r_{ij} = (\delta_i - \delta_j)^T L^\dagger (\delta_i - \delta_j)$ defines a metric on the nodes which can be used for sparsification, nodal embeddings, and beyond.
- Chung, Kempton, and Zhao[9] looked at this in the setting of connection graphs and edge ranking.
- Cloninger et al.[10] revisited effective resistance with a new approach related to random walk-based mean rotations.

[9] Chung, Zhao, and Kempton, "Ranking and sparsifying a connection graph".
[10] Cloninger et al., "Random Walks, Conductance, and Resistance for the Connection Graph Laplacian".

# Minimum Cost Flows

# OT on Graphs

- Optimal transport is a mathematical framework for finding the most efficient way to transport one distribution of mass to another, minimizing a cost function that quantifies the expense of moving each unit of goods.[11]
- Loosely speaking one distribution is an initial location of the mass, and the second is a prescribed location for where it is to be deposited; and Wasserstein distance is the optimal cost to transport one to the other with respect to some ground metric.

[11] Gabriel Peyré, Marco Cuturi, et al. "Computational optimal transport". In: *Center for Research in Economics and Statistics Working Papers* 1.2017-86 (2017).

# OT on Graphs

- Optimal transport is a mathematical framework for finding the most efficient way to transport one distribution of mass to another, minimizing a cost function that quantifies the expense of moving each unit of goods.[11]
- Loosely speaking one distribution is an initial location of the mass, and the second is a prescribed location for where it is to be deposited; and Wasserstein distance is the optimal cost to transport one to the other with respect to some ground metric.
- Let $\mathcal{P}(G) = \left\{\alpha \in \mathbb{R}^n : \alpha_i \geq 0, \sum_{i=1}^n \alpha_i = 1\right\}$ be the simplex of probability densities on $G$.

## Earth-mover's distance

Let $\alpha, \beta \in \mathcal{P}(G)$. Then the 1-Wasserstein, or Earth-mover's distance between $\alpha, \beta$ is given by the following LP:

$$\mathcal{W}_1(\alpha, \beta) = \inf \left\{ \sum_{i,j \in V} d_{ij}\pi_{ij} : \pi \in \mathbb{R}^{n\times n}, \pi \geq 0, \pi \mathbf{1}_n = \alpha, \mathbf{1}_n^T \pi = \beta^T \right\}, \tag{2.1}$$

where $d_{ij}$ is the shortest-path distance between two nodes $i, j \in V$.

[11] Peyré, Cuturi, et al., "Computational optimal transport".

# Minimum Cost Flows

- On graphs (or rather, with shortest-path metric) the previous problem can be realized as a min cost flow problem. A proof of equivalence can be found in, e.g.,[12].

## Beckmann Problem on Graphs

$$\mathcal{W}_1(\alpha, \beta) = \inf \left\{ \sum_{e \in E'} w(e)|J(e)| : J \in \mathbb{R}^m, BJ = \alpha - \beta \right\}.$$

[12] Peyré, Cuturi, et al., "Computational optimal transport".

[13] James B Orlin, Serge A Plotkin, and Éva Tardos. "Polynomial dual network simplex algorithms". In: *Mathematical programming* 60.1 (1993).

# Minimum Cost Flows

- On graphs (or rather, with shortest-path metric) the previous problem can be realized as a min cost flow problem. A proof of equivalence can be found in, e.g.,[12].

## Beckmann Problem on Graphs

$$\mathcal{W}_1(\alpha,\beta) = \inf\left\{\sum_{e\in E'} w(e)|J(e)| : J \in \mathbb{R}^m, BJ = \alpha - \beta\right\}.$$

- The objective being nonsmooth and minimizers being nonunique, solving this is nontrivial. Time complexity of exact solutions are roughly on the order of $O(n^3\log(n))$[13]; there are many methods for approximate and/or regularized solutions with varying time complexities and error rates, classical methods are roughly $O(n^3)$.
- A solution is to quadratically regularize the problem and use duality. More on this momentarily.

[12] Peyré, Cuturi, et al., "Computational optimal transport".
[13] Orlin, Plotkin, and Tardos, "Polynomial dual network simplex algorithms".

## Some nuance

- $\mathcal{W}_1$ on graphs is pretty versatile; can be used in unsupervised learning problems, graph Ricci curvature (a can of worms in and of itself), image processing, ...

## Some nuance

- $\mathcal{W}_1$ on graphs is pretty versatile; can be used in unsupervised learning problems, graph Ricci curvature (a can of worms in and of itself), image processing, ...
- What about flows on **connection graphs**?

## Some nuance

- $\mathcal{W}_1$ on graphs is pretty versatile; can be used in unsupervised learning problems, graph Ricci curvature (a can of worms in and of itself), image processing, ...
- What about flows on **connection graphs**?
- A subtle detail from before becomes very important: on classical graphs, the linear system $BJ = \alpha - \beta$ is **feasible** if and only if $\alpha, \beta$ have the same total mass.
- On connection graphs, if $\alpha, \beta : V \to \mathbb{R}^d$ are **vector fields** representing supply and demand, $BJ = \alpha - \beta$ is not always feasible.

# A simple counterexample

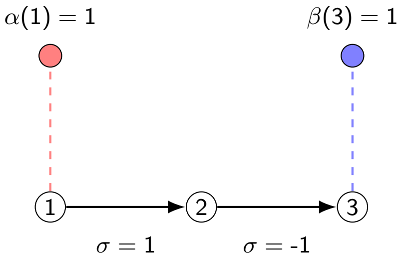



Figure: Case where $\alpha, \beta$ have no feasible flow and equal mass. Here, for simplicity, $d = 1$ and our connection is just a $\pm 1$ signature. As $\alpha$ is "pushed" in the direction of $\beta$, the sign is flipped, which is therefore not compatible. $\beta(3) = -1$ is feasible.

# More food for thought

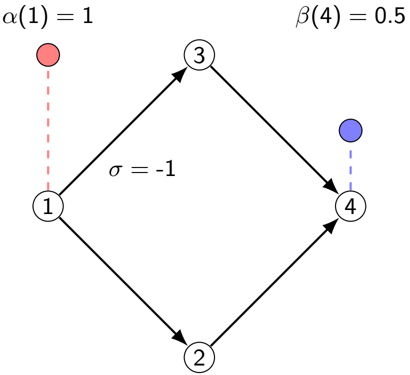



Figure: Case where $\alpha, \beta$ do not have equal mass but the problem is feasible. Take $J = 0.25$ on the upper path and $J = 0.75$ on the lower path, and $BJ = \alpha - \beta$.

# The Crux of the Problem

- We have illuminated a fundamental concept; that min cost flows on connection graphs are a bit weird: lack of feasible solutions on occasion, ability to have destructive interference from a flow.

## Beckmann Problem for Connection Graphs

Let $\alpha, \beta \in \mathbb{R}^{nd}$ be any vector-valued supply and demand functions on $V$. We define

$$\mathcal{W}_1^{\sigma}(\alpha, \beta) = \inf \left\{ \sum_{e \in E'} w(e)\|J(e)\|_2 : J \in \mathbb{R}^{md}, BJ = \alpha - \beta \right\}, \tag{2.2}$$

where $\mathcal{W}_1^{\sigma}(\alpha, \beta) = \infty$ if $BJ = \alpha - \beta$ has no solution.

- When is this feasible? How can we solve it?

- Define $\mathcal{P}_d(G) = \left\{\alpha \in \mathbb{R}^{n\times d} : \alpha_{ij} \geq 0, \sum_{i=1}^{n} \alpha_{i,:} = \mathbf{1}_d\right\}$.

- A connection $\sigma$ is said to be **consistent** whenever the product of signatures along any cycle in the graph is Id.

- A connection $\sigma$ is said to be **inconsistent** if it is not consistent, and **absolutely inconsistent** if the the subgroup of $O(d)$ generated by product of signatures along all cycles in the graph has no nontrivial invariant vector in $\mathbb{R}^d$ (and, among other things, the kernel of $L$ is trivial).

- Define $\mathcal{P}_d(G) = \left\{\alpha \in \mathbb{R}^{n\times d} : \alpha_{ij} \geq 0, \sum_{i=1}^n \alpha_{i,:} = \mathbf{1}_d\right\}$.
- A connection $\sigma$ is said to be **consistent** whenever the product of signatures along any cycle in the graph is Id.
- A connection $\sigma$ is said to be **inconsistent** if it is not consistent, and **absolutely inconsistent** if the the subgroup of $O(d)$ generated by product of signatures along all cycles in the graph has no nontrivial invariant vector in $\mathbb{R}^d$ (and, among other things, the kernel of $L$ is trivial).

**Theorem (SR, DK, GM, AC 2023)**

*If $(G, \sigma)$ is connected and absolutely inconsistent then $\mathcal{W}_1^{\sigma}(\alpha, \beta)$ is always feasible. For any non-absolutely inconsistent connection Laplacian $L$ there is a block diagonal orthogonal matrix $U = \mathrm{diag}(u_1, \ldots, u_n) \in O(nd)$, where $u_i \in O(d)$, such that for the modified connection Laplacian $U^T LU$, the problem $\mathcal{W}_1^{\sigma'}(\alpha, \beta)$ is always feasible when $\alpha, \beta \in \mathcal{P}_d(G)$.*

# Regularization

## On classical graphs

- Adding a bit of regularization to an $\ell_1$ problem has lots of advantages: solutions are often unique owing to strict convexity (depending on the regularizer), and primal/dual methods often afford otherwise inaccessible solution methods

# On classical graphs

- Adding a bit of regularization to an $\ell_1$ problem has lots of advantages: solutions are often unique owing to strict convexity (depending on the regularizer), and primal/dual methods often afford otherwise inaccessible solution methods

## Regularized Beckmann Problem on Graphs; Essid, Solomon 2018

Let $\lambda > 0$ be fixed, and define

$$\mathcal{W}_{1,\lambda}(\alpha, \beta) = \inf_{J \in \mathbb{R}^m} \left\{ \sum_{e \in E'} w(e)|J(e)| + \frac{\lambda}{2} \|J(e)\|_2^2 : BJ = \alpha - \beta \right\}.$$

This problem admits a convex dual and the optimal values coincide:

$$\mathcal{W}_{1,\lambda}(\alpha, \beta) = \sup_{\phi \in \mathbb{R}^n} \left[ \phi^T(\alpha - \beta) - \frac{1}{2\lambda} \|(B^T\phi(e) - w_e)_+\|_2^2 \right]$$

where for $x \in \mathbb{R}^m$, $(x)_+ = x\mathbf{1}_{x \geq 0}$. Moreover, duality correspondence holds: we can write down an optimal solution for the primal if an optimal solution for the dual is found.[a]

[a] Essid and Solomon, "Quadratically regularized optimal transport on graphs".

## Adapting for Connection Graphs

### Theorem (SR, DK, GM, AC 2023)

*Let $(G,\sigma)$ be a connected connection graph. Let $\alpha, \beta \in \mathcal{P}_d(G)$. Then strong duality holds for the following problems,*

$$\mathcal{W}_1^{\sigma,\lambda}(\alpha,\beta) = \inf_{J \in \ell_2(E';\mathbb{R}^d)} \left\{ \sum_{e \in E'} w(e)\|J(e)\|_2 + \frac{\lambda}{2}\|J(e)\|_2^2 : \text{div}(J) = \alpha - \beta \right\} \tag{3.1}$$

$$= \sup_{\phi \in \ell_2(V;\mathbb{R}^d)} \left\{ \phi^T(\alpha-\beta) - \frac{1}{2\lambda}\sum_{e\in E'} \chi_e(\phi)\left(\left\|(B^T\phi)(e)\right\|_2 - w(e)\right)^2 \right\} \tag{3.2}$$

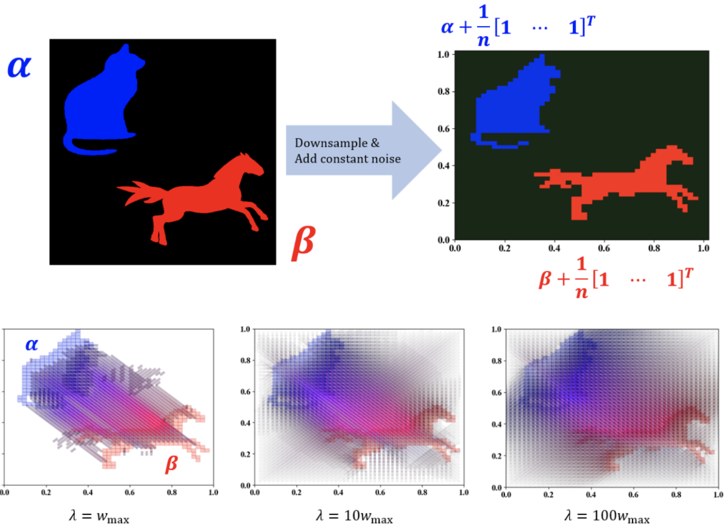
*where*

$$\chi_e(\phi) := \begin{cases} 1 & \text{if } \left\|(B^T\phi)(e)\right\|_2 > w(e) \\ 0 & \text{otherwise} \end{cases}. \tag{3.3}$$

Moreover, if the primal is feasible and $\phi$ maximizes the dual then the optimal primal $J(\phi)$ is given by

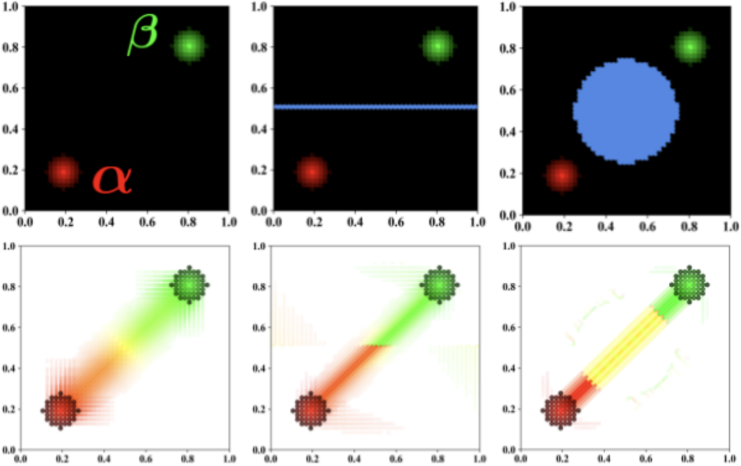
$$[J(\phi)](e) = -\chi_e(\phi)\left(\frac{\left\|(B^T\phi)(e)\right\|_2 - w(e)}{\lambda}\right)\frac{(B^T\phi)(e)}{\|(B^T\phi)(e)\|_2}. \tag{3.4}$$

# Example 1



$\alpha$
$\beta$
Downsample &
Add constant noise
$\alpha + \frac{1}{n}[1 \ \cdots \ 1]^T$
$\beta + \frac{1}{n}[1 \ \cdots \ 1]^T$
0.0 0.2 0.4 0.6 0.8 1.0
$\lambda = w_{\max}$
$\lambda = 10w_{\max}$
$\lambda = 100w_{\max}$

# Example 2



# Acknowledgements

Dhruv Kohli

Alex Cloninger

Gal Mishne


- SR wishes to acknowledge financial support from the Halıcıoğlu Data Science Institute through their Graduate Prize Fellowship
- AC was funded by NSF DMS 2012266 and a gift from Intel. GM was supported by NSF CCF 2217058.
- Thanks to the organizers of Network Science Beyond Graphs


NSF

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

intel®