Day 4: Transferability of Graph Neural Networks

Charilaos I. Kanatsoulis, Navid NaderiAlizadeh,

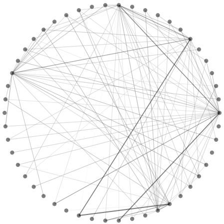
Alejandro Parada-Mayorga, Alejandro Ribeiro, and Luana Ruiz

gnn.seas.upenn.edu

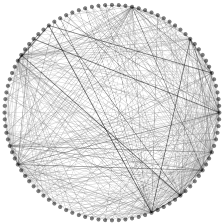
2023 International Conference on Acoustics, Speech, and Signal Processing

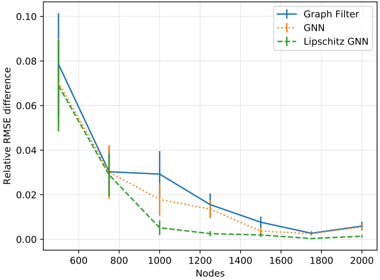
Rhodes, Greece – June 9, 2023

- Transferability of graph neural networks is ready to verify in practice $\Rightarrow$ recommendation system



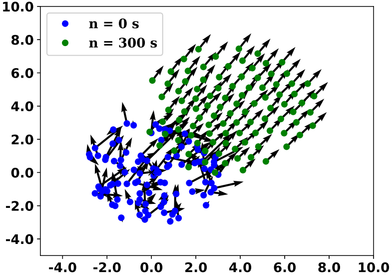
$\rightarrow$

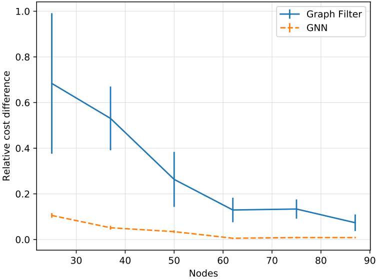




- Performance difference on training and target graphs decreases as size of training graph grows
- GNNs appear to be more transferable than graph convolutional filters $\Rightarrow$ better ML model

- Transferability of graph neural networks is ready to verify in practice $\Rightarrow$ decentralized robot control





- Performance difference on training and target graphs decreases as size of training graph grows
- GNNs appear to be more transferable than graph convolutional filters $\Rightarrow$ better ML model

Q1: We have empirically observed that GNNs transfer at scale. Why do they?

Q2: Can success of GNNs on moderate-size graphs be used to create success at large-scale?

- To answer these questions, turn to CNNs $\Rightarrow$ known to scale well for images and time sequences

- Discrete time/image signals converge to continuous time/image signals $\Rightarrow \downarrow$ intrinsic dimension

143 × 95 →

205 × 136 →

294 × 195 →

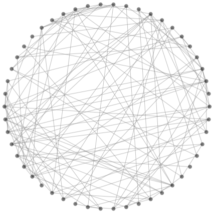
600 × 399

$\Rightarrow$ From SP theory, CNNs have well-defined limits on the limits of images and time signals

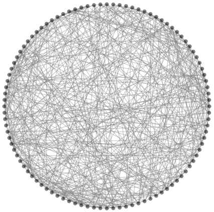
- A1: Intrinsic dimensionality of the problem is less than the size of the image
- A2: Training with small images is sufficient $\Rightarrow$ CIFAR 10 images are 32×32

Graphons

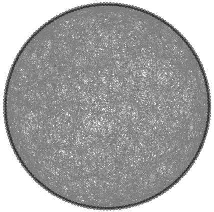
- Graphs also have limit objects that effectively limit their dimensionality $\Rightarrow$ one is the graphon



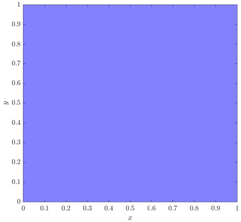
$n = 50$ nodes $\rightarrow$



$n = 100$ nodes $\rightarrow$



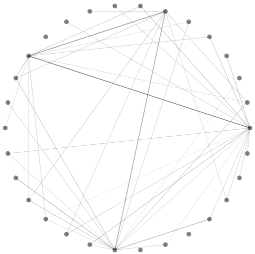
$n = 200$ nodes $\rightarrow$



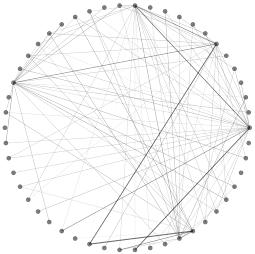
Graphon $W(u, v) = p$

- A graphon can be thought of as a graph with an uncountable number of nodes

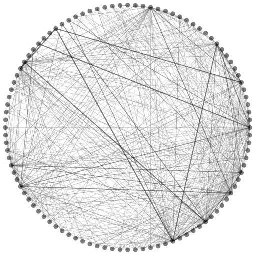
- Graphs however do not have the Euclidean structure time and image signals have in the limit



$n = 30$ products



$n = 50$ products



$n = 100$ products

- So do graph convolutions and graph neural networks converge to limits on the graphon?

Q1: We have empirically observed that GNNs scale. Why do they scale?

- A1: Because graph convolutions and GNNs have well-defined limits on graphons

L. Ruiz et al, *Graphon Signal Processing*, TSP 2021, https://arxiv.org/abs/2003.05030

L. Ruiz et al, *Transferability Properties of Graph Neural Networks*, https://arxiv.org/abs/2112.04629

Q2: Can success of GNNs on moderate-size graphs be used to create success at large-scale?

- A2: Yes, as GNNs are transferable $\Rightarrow$ can be trained on moderate-size and executed on large-scale

J. Cerviño et al, *Learning by Transference: Training Graph Neural Networks on Growing Graphs.*, https://arxiv.org/abs/2106.03693

Graphons

- We introduce graphons to study graph filters and GNNs in the limit of large number of nodes

Definition (Graphon)

A graphon is a bounded symmetric measurable function $\Rightarrow W : [0, 1]^2 \to [0, 1]$

- Can think of a graphon as a weighted symmetric graph with uncountable nodes

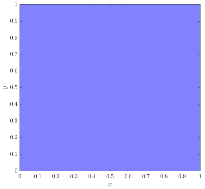
$\Rightarrow$ The labels are the graphon arguments $\Rightarrow u \in [0, 1]$.

$\Rightarrow$ The weights are the graphon values $\Rightarrow W(u, v) = W(v, u)$

Definition (Graphon)

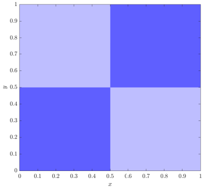
A graphon is a bounded symmetric measurable function $\Rightarrow$ $W : [0,1]^2 \rightarrow [0,1]$

Uniform (Erdős-Rényi)



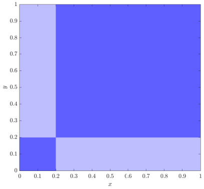
$W(u,v) = p$

Balanced stochastic block model (SBM)



$W(u,v) = p \gg W(u,v) = q$

Unbalanced (SBM)



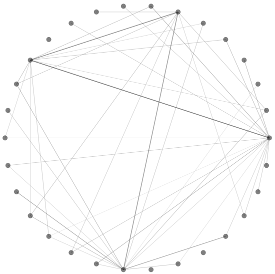
$W(u,v) = p \gg W(u,v) = q$

Definition (Graphon)

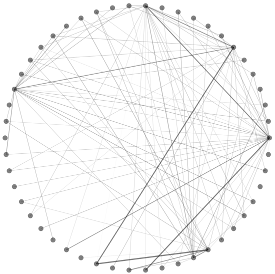
A graphon is a bounded symmetric measurable function $\Rightarrow W : [0,1]^2 \rightarrow [0,1]$

- Practice $\Rightarrow$ Graph sets where graphs in the set have large number of nodes and similar structure
- Theory $\Rightarrow$ A generative model of graph families via deterministic or stochastic edge sampling
- Theory $\Rightarrow$ A limit object for a sequence of graphs

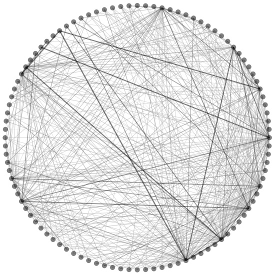
- Product similarity graphs, even with different number of nodes, "look like each other"
- Abstract similarities between graphs into a limit object $\Rightarrow$ The product similarity "graphon"



$n = 30$ products



$n = 50$ products

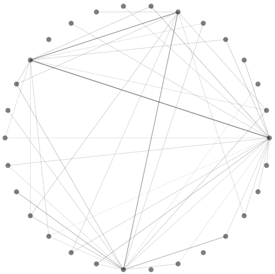


$n = 100$ products

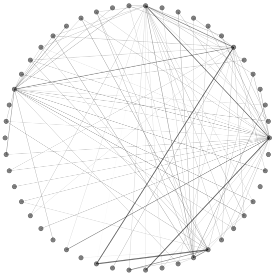
The Product Similarity "Graphon"

- We never compute the product similarity "graphon"

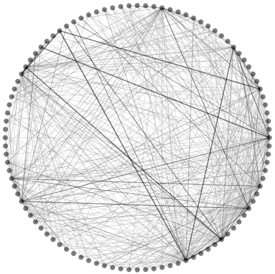
$\Rightarrow$ Use abstract idea of graphon to work with all of these graphs as if they were the same object



$n = 30$ products



$n = 50$ products



$n = 100$ products

- Vertices: For an n-node graph, sample n points $\{u_1, u_2, \ldots, u_n\}$ from the unit interval $[0, 1]$

  $\Rightarrow$ Points can be sampled on a grid, uniformly at random, etc.

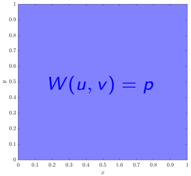
  $\Rightarrow$ Each sample u_i corresponds to a node $i \in \{1, 2, 3, \ldots, n\}$ of the graph

- Edges: Evaluate $\mathrm{W}(u_i, u_j)$ for edge (i, j)

  $\Rightarrow$ Stochastic: Connect i and j with an unweighted undirected edge with probability $\mathrm{W}(u_i, u_j)$

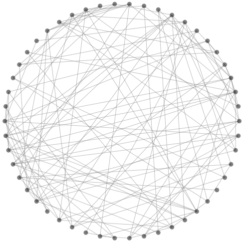
  $\Rightarrow$ Weighted: Connect i and j with weighted undirected edge with weight $\mathrm{W}(u_i, u_j)$

- Use uniform Graphon

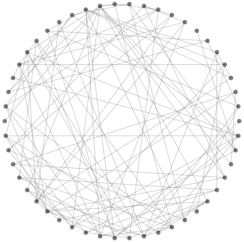


To generate random graphs with the same

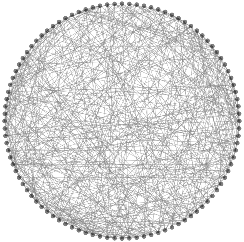
Or different number of nodes



$n = 50$ nodes



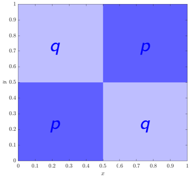
$n = 50$ nodes



$n = 100$ nodes

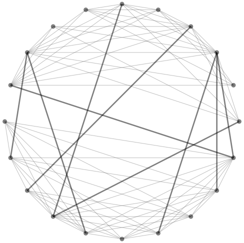
- Use balanced SBM

  Graphon

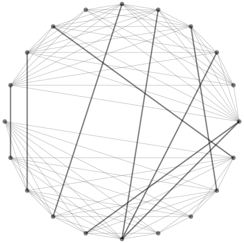


To generate balanced SBM graphs with the same

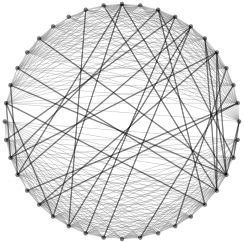
Or different number of nodes



$n = 20$ nodes

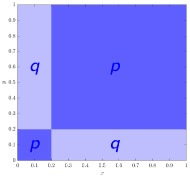


$n = 20$ nodes

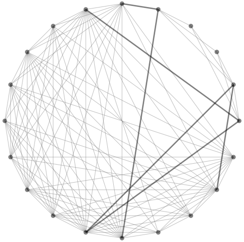


$n = 40$ nodes

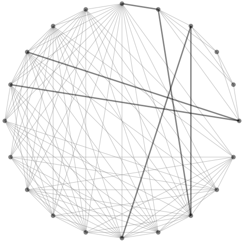
- Use Unbalanced SBM Graphon



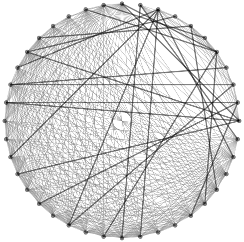
To generate unbalanced SBM graphs with the same Or different number of nodes



$n = 20$ nodes



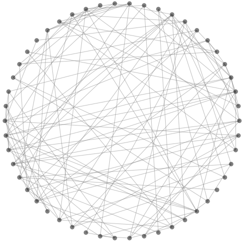
$n = 20$ nodes



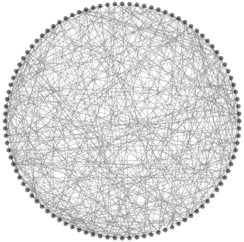
$n = 40$ nodes

- As we consider random graphs with larger numbers of nodes the graphs approach a limit

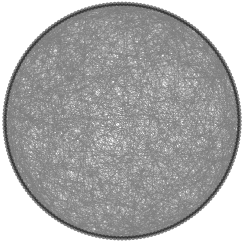
$\Rightarrow$ It is unclear what that limit is. The graphon is the limit. As we will see



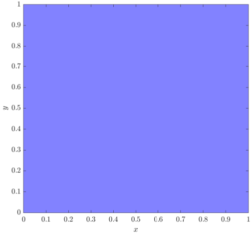
$n = 50$ nodes $\rightarrow$



$n = 100$ nodes $\rightarrow$



$n = 200$ nodes $\rightarrow$



Graphon $W(u, v) = p$

Convergence of Graph Sequences

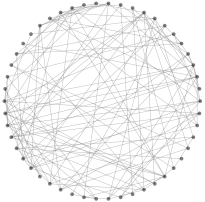
- A graphon is the limit of a sequence of graphs that converges in terms of homomorphism densities

Convergent Graph Sequences

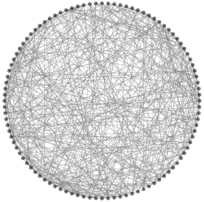
- Sequence of graphs with growing number of nodes n $\Rightarrow \left\{ G_n = (V_n, E_n, S_n) \right\}_{n=1}^{\infty}$.

- The graph sequence $\{G_n\}_{n=1}^{\infty}$ converges to a graphon W $\Rightarrow$ In what sense?

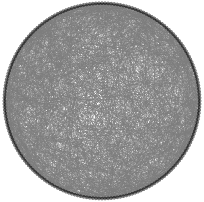
$\Rightarrow$ We need to introduce three concepts: Motifs, homomorphisms, and homomorphism densities



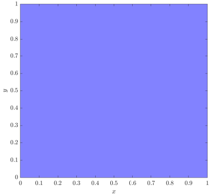
$n = 50$ nodes $\rightarrow$



$n = 100$ nodes $\rightarrow$

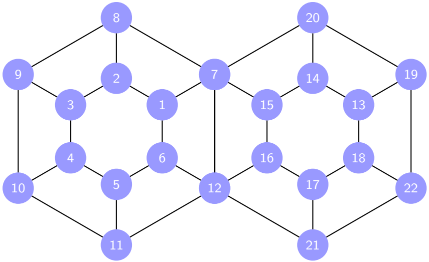


$n = 200$ nodes $\rightarrow$



Graphon $W(u, v) = p$

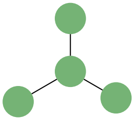
- A motif F is a graph. But think of it as a small graph that we embed in another larger graph

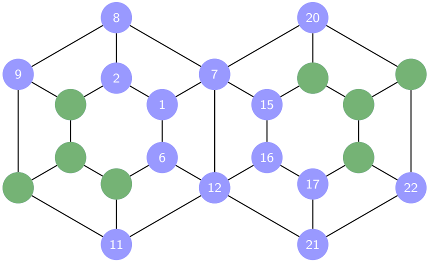


- Homomorphisms are adjacency preserving maps from motif $F = (V', E')$ into graph $G = (V, E)$

$$\beta : V' \to V \text{ such that } \Big(i, j\Big) \in E' \text{ implies } \Big(\beta(i), \beta(j)\Big) \in E$$

- A motif F is a graph. But think of it as a small graph that we embed in another larger graph

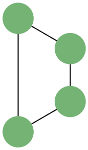


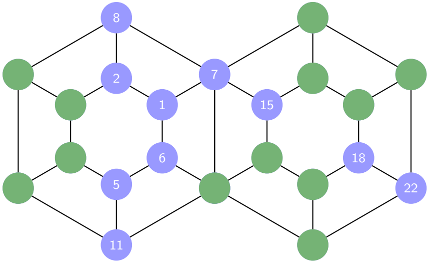


- Homomorphisms are adjacency preserving maps from motif $F = (V', E')$ into graph $G = (V, E)$

$$\beta \ : \ V' \ \rightarrow \ V \ \text{ such that } \ \Big(i, j \Big) \ \in \ E' \ \text{ implies } \ \Big(\beta(i), \beta(j) \Big) \ \in \ E$$

- A motif F is a graph. But think of it as a small graph that we embed in another larger graph

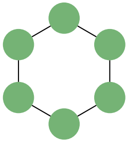


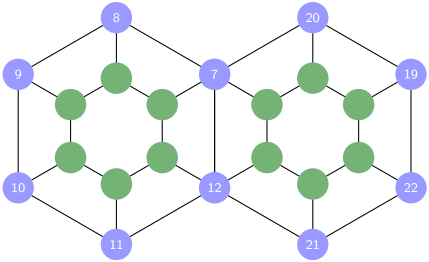


- Homomorphisms are adjacency preserving maps from motif $F = (V', E')$ into graph $G = (V, E)$

$$\beta \; : \; V' \rightarrow V \;\text{ such that }\; \Big(i, j\Big) \in E' \;\text{ implies }\; \Big(\beta(i), \beta(j)\Big) \in E$$

- A motif F is a graph. But think of it as a small graph that we embed in another larger graph

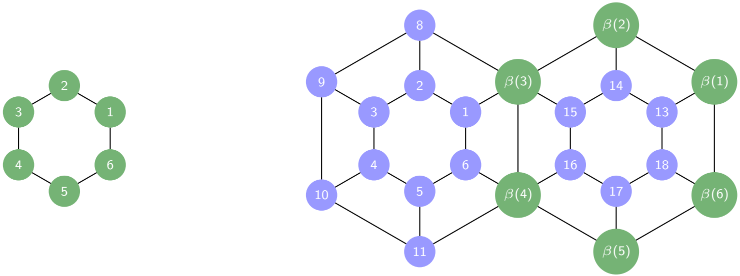




- Homomorphisms are adjacency preserving maps from motif $F = (V', E')$ into graph $G = (V, E)$

$$\beta : V' \rightarrow V \text{ such that } \Big(i, j\Big) \in E' \text{ implies } \Big(\beta(i), \beta(j)\Big) \in E$$

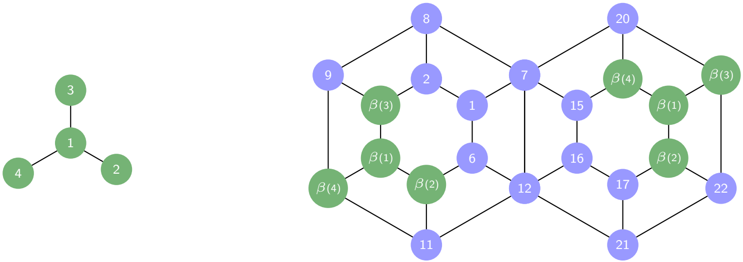
- A motif F is a graph. But think of it as a small graph that we embed in another larger graph



- Homomorphisms are adjacency preserving maps from motif $F = (V', E')$ into graph $G = (V, E)$

$$\beta : V' \to V \text{ such that } \Big(i, j\Big) \in E' \text{ implies } \Big(\beta(i), \beta(j)\Big) \in E$$

- Given motif F and graph G, there are multiple homomorphism functions β



- We define $\text{hom}(F, G)$ to represent the number of homomorphisms between motif F and graph G

- If the graph G has n nodes and the motif F has n' nodes, there are $n^{n'}$ different maps from F to G

- Homomorphism density of motif F in graph G is the fraction of maps that are homomorphisms

$$t(F, G) \;=\; \frac{\text{hom}(F, G)}{n^{n'}}$$

- Density $t(F, G)$ is a relative measure of the number of ways in in which F can be mapped into G

- Consider weighted graph $G = (V, E, S)$ with adjacency matrix S

- Homomorphism density of motif F in weighted graph G with the adjacency matrix S is

$$t(F, G) \; = \; = \frac{\sum_{\beta} \prod_{(i,j)\in\mathcal{E}'} [\mathbf{S}]_{\beta(i)\beta(j)}}{n^{n'}}$$

- Weight each motif embedding by the product of the edge weights in the homomorphism image.

Homomorphism Density for Graphons

- The Homomorphism density of a motif F into a given graphon W is defined as

$$t(F, W) = \int_{[0,1]^{n'}} \prod_{(i,j)\in\mathcal{E}'} W(u_i, u_j) \prod_{i\in\mathcal{V}'} du_i$$

- The homomorphism density is the probability of drawing the motif F from the graphon W

Definition (Convergent graph sequence)

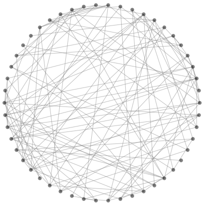
A sequence of undirected graphs G_n converges to the graphon W if and only if for all motifs F

$$\lim_{n\to\infty} t(F, G_n) = t(F, W)$$

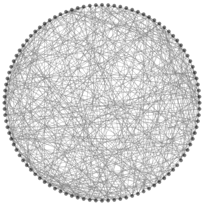
- We say that the sequence G_n converges to W in the homomorphism density sense
- It can be proven that every graphon is the limit object of a sequence of convergent graphs
- It can be proven that every convergent graph sequence converges to a graphon

Example of Convergent Graph Sequence

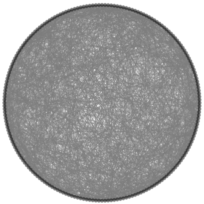
- Consider a sequence of random graphs $\{G_n\}$ sampled from the graphon W. Graphs G_n have
  - $\Rightarrow$ Labels $u_i \sim U[0,1]$ drawn uniformly at random from the interval $[0,1]$
  - $\Rightarrow$ Edge sets such that $(u_i, u_j) \in \mathcal{E}$ with probability $W(u_i, u_j)$
- We have $\lim_{n\to\infty} t(F, G_n) = t(F, W)$ in the homomorphism density sense almost surely



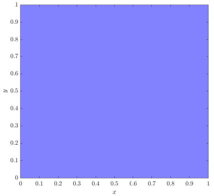
$n = 50$ nodes $\rightarrow$



$n = 100$ nodes $\rightarrow$

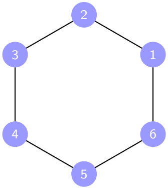


$n = 200$ nodes $\rightarrow$



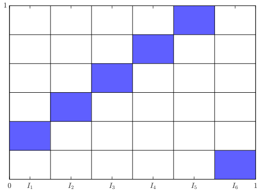
Graphon $W(u, v) = p$

- Every undirected graph admits a graphon representation which we call its induced graphon
- Consider a graph $G = \{\mathcal{V}, \mathcal{E}, \mathbf{S}\}$ with $|\mathcal{V}| = n$ and normalized graph shift operator $\mathbf{S}$
- Regular partition of the unit interval with n subintervals $\Rightarrow \mathrm{I}_i = \left[(i-1)/n,\ i/n\right)$
- We define the induced graphon $W_G \Rightarrow W_G(u,v) = [\mathbf{S}]_{ij}\ \mathbb{I}(u \in \mathrm{I}_i)\ \mathbb{I}(v \in \mathrm{I}_j)$



Cycle graph G with $n = 6$ nodes

$\rightarrow$



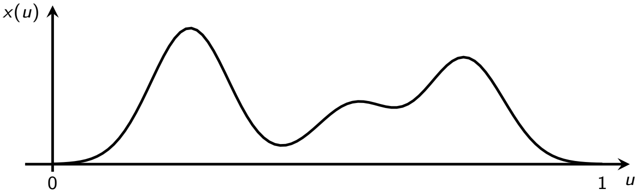
Graphon W_G induced by the graph G

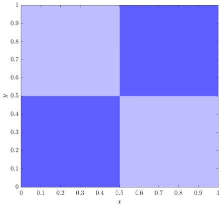
Graphon Signals

- Graph signals are signals supported on graphons. They are limit objects of graph signals

- Graphon signals are pairs (W, X) where W is a graphon and $X : [0,1] \to \mathbb{R}$ is a function

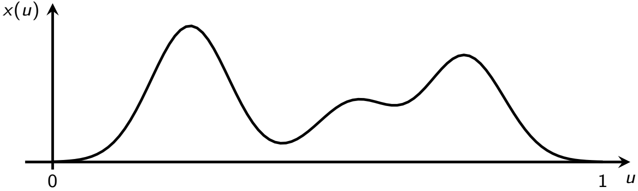
- Function $X(u) \in L^2([0,1])$ has finite energy $\Rightarrow \int_0^1 |X(u)|^2 du < \infty$.

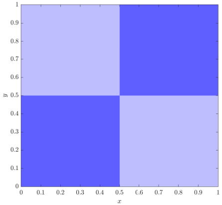




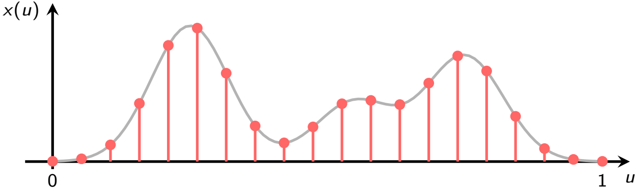
- Generative models of graph signals. And limits of convergent sequences of graph signals

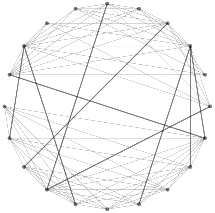
- We generate graph signals $(\mathbf{S}_n, \mathbf{x}_n)$ by taking n samples of the graphon signal (W, X)
- Sample the graphon at node labels u_i. Sample the function X at node labels $u_i \Rightarrow x_i = X(u_i)$
- Graph signal sampled from the unit interval in the same coordinates u_i where graphon is sampled





- We generate graph signals $(\mathbf{S}_n, \mathbf{x}_n)$ by taking n samples of the graphon signal (W, X)
- Sample the graphon at node labels u_i. Sample the function X at node labels $u_i \Rightarrow x_i = X(u_i)$
- Graph signal sampled from the unit interval in the same coordinates u_i where graphon is sampled

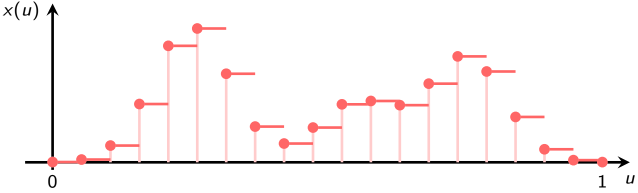


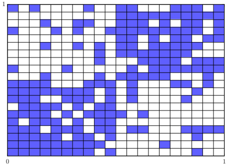


- Every graph signal x supported on graph G induces a graphon signal (W_G, X_G)
- Regular partition of unit interval with n subintervals $\mathrm{I}_i = \left[(i-1)/n,\ i/n\right)$

$\Rightarrow$ Induced signal $X_G(u) = x_i\, \mathbb{I}(u \in I_i)$

$\Rightarrow$ W_G is the graphon induced by the graph G $\Rightarrow$ $W_G(u,v) = [\mathrm{S}]_{ij}\ \mathbb{I}(u \in \mathrm{I}_i)\ \mathbb{I}(v \in \mathrm{I}_j)$





Definition (Convergent sequences of graph signals)

A sequence of graph signals $(G_n, \mathbf{x}_n)$ is said to converge to the graphon signal (W, X), if there exists a sequence of permutations π_n such that for all motifs F we have

$$t(F, G_n) \rightarrow t(F, W), \qquad \text{and} \left\| X_{\pi_n(G_n)} - X \right\|_{L^2} \rightarrow 0$$

We say (W, X) is the limit of the graph signal sequence and write $(G_n, x_n) \rightarrow (W, X)$

- The permutation is used here to make the convergence definition independent of labels
- To enable comparison of the vector $\mathbf{x}_n$ and the function X we use the induced signal in the L_2 norm

- The Graphon W can be used to define an integral linear operator $\Rightarrow T_W : L^2([0,1]) \to L^2([0,1])$

- When applied to the graphon signal X, the operator T_W produces the signal $T_W X$ with values

$$(T_W X)(v) = \int_0^1 W(u,v) X(u)\, du$$

- This is a Hilbert-Schmidt operator because W is bounded and compact. It's a matrix multiplication

- We say that the linear operator T_W is the graphon shift operator (WSO) of the graphon W

$\Rightarrow$ Applying the WSO T_W to the graphon signal X diffuses X over the graphon W

Graphon Fourier Transform

- We define a graphon Fourier transform to enable spectral representation of graphon signals.

- The WSO is a self adjoint Hilbert-Schmidt operator $\Rightarrow (T_{\mathrm{W}}X)(v) = \int_0^1 \mathrm{W}(u,v)X(u)\,du$

- The function $\varphi : [0,1] \to \mathbb{R}$ is an eigenfunction of T_W with associated eigenvalue λ if

$$(T_{\mathrm{W}}\varphi)(v) \; = \; \int_0^1 \mathrm{W}(u,v)\varphi(u)\,du \; = \; \lambda\varphi(v)$$

- T_{W} has a countable number of eigenvalue-eigenfunction pairs $\Rightarrow \left\{ (\lambda_i, \varphi_i) \right\}_{i=1}^{\infty}$

- We assume eigenfunctions are normalized to unit energy $\Rightarrow \|\varphi_i\|^2 = \int_0^1 \varphi(u)du = 1$

- The (countable number of) eigenfunctions of the operator T_w are an orthonormal basis of $L^2([0,1])$

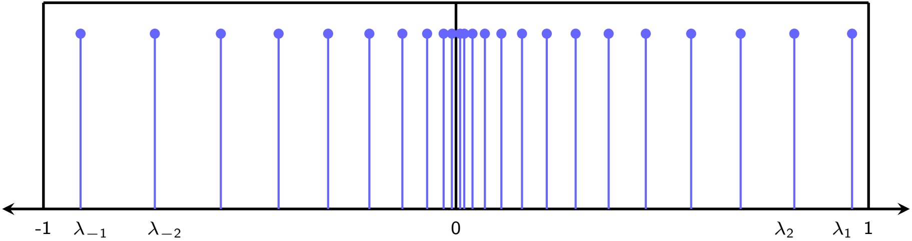
- We can thus decompose the graphon W in the basis $\{\varphi_i\}_{i=1}^{\infty}$ of eigenfunctions of the operator T_W

$$W(u, v) = \sum_{i=0}^{\infty} \lambda_i \, \varphi_i(u) \, \varphi_i(v)$$

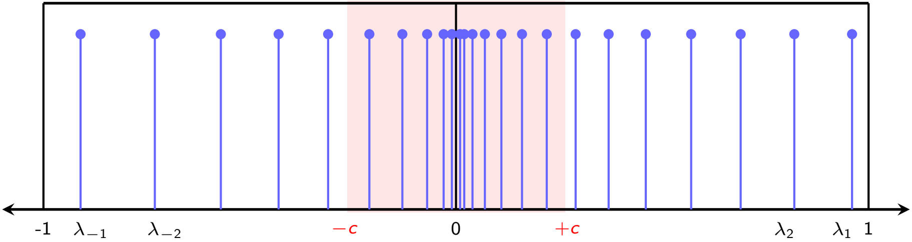
- More or less the same as the eigenvector decomposition $\Rightarrow S = V \Lambda V^H = \sum_{i=0}^{\infty} \lambda_i \, v_i \, v_i^T$

- T_W is self adjoint and $0 \leq W(x,y) \leq 1$ $\Rightarrow$ Eigenvalues are real and lie in the interval $[-1,1]$

- Order them as $\Rightarrow -1 \leq \lambda_{-1} \leq \lambda_{-2} \leq \ldots \leq 0 \leq \ldots \leq \lambda_2 \leq \lambda_1 \leq 1$



- Graphon eigenvalues accumulate at $\lambda = 0$ $\Rightarrow \lim_{i\to\infty} \lambda_i = \lim_{i\to\infty} \lambda_{-i} = 0$. And only at $\lambda = 0$
- For any $c > 0$, the number of eigenvalues with $|\lambda_i| \geq c$ is finite $\Rightarrow \#\left\{\lambda_i : |\lambda_i| \geq c\right\} = n_c < \infty$
- All eigenvalues that are not $\lambda_j = 0$ have finite multiplicity



Theorem (Eigenvalue Convergence of a Graph Sequence)

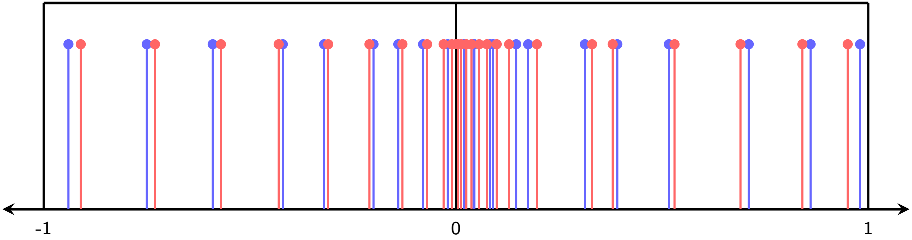
If a graph sequence $\{G_n\}$ converges to a graphon W in the homomorphism density sense , then

$$\lim_{n\to\infty} \frac{\lambda_j(S_n)}{n} = \lambda_j(T_W) = \lim_{n\to\infty} \lambda_j(T_{W_n}) \text{ for all } j$$

- For any convergent graph sequence, the eigenvalues of the graph converge to those of the graphon

Borgs-Chayes-Lovász-Sós-Vesztergombi, *Convergent Sequences of Dense Graphs II. Multiway Cuts and Statistical Physics,*

- For a convergent graph sequence, eigenvalues of the graph converge to those of the limit graphon



- Convergence holds in the sense that $\Rightarrow \exists\, n_0$ s.t. for all $n > n_0$, $\left| \frac{\lambda_j(\mathbf{S}_n)}{n} - \lambda_j(T_{\mathbf{W}}) \right| < \epsilon\,, \epsilon > 0$
- But n_0 will be different for each j. Eigenvalue convergence is not uniform

The Graphon Shift Operator Induces a Transform

- The graphon shift operator can be rewritten as

$$(T_{\mathrm{W}}\phi)(v) = \sum_{j=0}^{\infty} \lambda_j \varphi_j(v) \int_0^1 \varphi_j(u) X(u) du$$

- Integral terms correspond to inner products $\langle X, \varphi_j \rangle$ between the signal and the eigenfunctions
- Moreover, the eigenfunctions form a complete orthonormal basis of $L^2([0,1])$
- Thus, the inner products can provide a complete representation of the signal on the graphon basis
- That change of basis is called the graphon Fourier Transform

Definition (Graphon Fourier transform)

The graphon Fourier transform (WFT) of a graphon signal X is defined as a functional $\hat{X} = \text{WFT}(X)$ with continuous input X and discrete output

$$\hat{X}_j = \hat{X}(\lambda_j) = \int_0^1 X(u)\varphi_j(u)du$$

with $\{\lambda_j\}_{j\in\mathbb{Z}/\{0\}}$ the eigenvalues and $\{\varphi_j\}_{j\in\mathbb{Z}/\{0\}}$ the eigenfunctions of T_W

- The eigenvalues λ_j are countable $\Rightarrow$ The graphon Fourier transform $\hat{X}$ can always be defined

Definition (Inverse graphon Fourier transform)

The inverse graphon Fourier transform (iWFT) of a graphon Fourier transform $\hat{X}$ is defined as

$$\text{iWFT}(\hat{X}) = \sum_{j \in \mathbb{Z}/\{0\}} \hat{X}(\lambda_j)\varphi_j = X$$

with $\{\lambda_j\}_{j \in \mathbb{Z}/\{0\}}$ the eigenvalues and $\{\varphi_j\}_{j \in \mathbb{Z}/\{0\}}$ the eigenfunctions of T_{W}

- Eigenfunctions $\{\varphi_j\}_{j \in \mathbb{Z}/\{0\}}$ are orthonormal. The iWFT is a proper inverse of the WFT

The GFT converges to the WFT

- We discuss the convergence of the GFT to the WFT for graph sequences that converge to graphons.

- This need us to review convergence of eigenvectors and eigenvalues of graph sequences

- Graphon FT, $\text{WFT}(W, X)$ is the eigenspace projection $\Rightarrow \hat{X}_j = \hat{X}(\lambda_j) = \int_0^1 X(u)\, \varphi_j(u)\, du$

- Graph FTs, $\text{GFT}(G_n, \mathbf{x}_n)$ are the eigenspace projections $\Rightarrow \hat{x}_n(j) = \hat{x}_n(\lambda_{nj}) = \sum_{i=1}^{n} x_n(i)\, \mathbf{v}_{nj}(i)$

- Graph signal sequence $(G_n, \mathbf{x}_n)$ converges to graphon signal (W, X) $\Rightarrow$ Conjecture GFT convergence

$$\text{GFT}(G_n, \mathbf{x}_n) \rightarrow \text{WFT}(W, X)$$

- Eigenvalue convergence holds $\Rightarrow \lambda_{nj} \rightarrow \lambda_j$. Conjecture is reasonable GFT convergence should hold

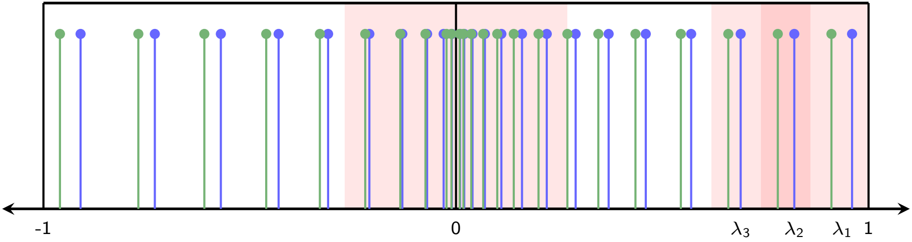
The Graphon Fourier Transform and the Graph Fourier Transform

- Graphon FT, WFT(W, X) is the eigenspace projection $\Rightarrow \hat{X}_j \;=\; \hat{X}(\lambda_j) \;=\; \int_0^1 X(u)\,\varphi_j(u)\,du$
- Graph FTs, GFT(G_n, x_n) are the eigenspace projections $\Rightarrow \hat{x}_n(j) \;=\; \hat{x}_n(\lambda_{nj}) \;=\; \sum_{i=1}^{n} \mathrm{x}_n(i)\,\mathrm{v}_{nj}(i)$
- Alas, this conjecture is wrong $\Rightarrow$ GFT convergence to the WFT does not hold in general

$$\mathrm{GFT}(G_n, \mathrm{x}_n) \;\not\to\; \mathrm{WFT}(W, X)$$

- GFT and WFT are projections on eigenvectors and eigenfunctions. Not eigenvalues

- Convergence of two eigenvectors depends on how close the eigenvalues of other eigenvectors are
- Eigenvalues accumulate around $\lambda = 0$. They all converge. But different eigenvalues are close
- It makes the eigenvectors slow to converge $\Rightarrow$ They all converge but convergence is not uniform



Eigenvalue Margin for Linear Operators

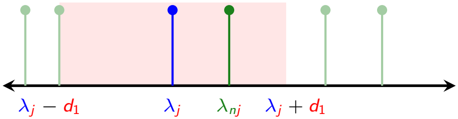
- Consider eigenvalues λ_j of graphon W and λ_{nj} of graph G_n with the same index j

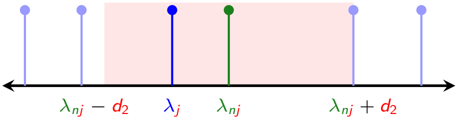
  $\Rightarrow$ Compare graphon eigenvalue λ_j to the closest graph eigenvalue other than λ_{nj}

  $\Rightarrow$ Compare graph eigenvalue λ_{ni} to the closest graphon eigenvalue other than λ_j

$$d(\lambda_j, \lambda_{nj}) \;=\; \min\left(\, d_1 \;=\; \min_{i \neq j} \left| \lambda_j - \lambda_{ni} \right|,\; d_2 \;=\; \min_{i \neq j} \left| \lambda_{nj} - \lambda_i \right| \,\right)$$

  $\Rightarrow$ The minimum of these two is the eigenvalue margin $d(\lambda_j, \lambda_{nj})$ for the eigenvalue pair $(\lambda_j, \lambda_{nj})$





Theorem (Davis-Kahan)

Given graphon W and graphon W_{G_n} induced by graph G_n we consider graphon eigenvalue λ_j and graph eigenvalue λ_{nj}. The distance between the associated eigenfunctions is bounded by

$$\|\varphi_j - \varphi_{nj}\| \leq \frac{\pi}{2} \frac{\|W - W_{G_n}\|}{d(\lambda_j, \lambda_{nj})}$$

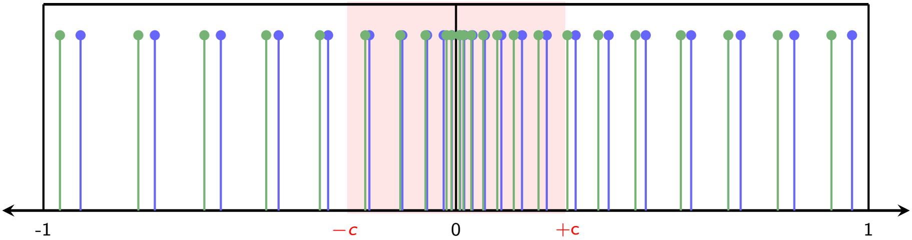
where $d(\lambda_j, \lambda_{nj})$ is the eigenvalue margin for the eigenvalue pair $(\lambda_j, \lambda_{nj})$

- Graph eigenvectors converge to graphon eigenfunctions if graph sequence converges to graphon
- When the distance to other eigenvalues decreases, the distance between eigenvectors increases

- For eigenvalues close to 0 the margin $d(\lambda_j, \lambda_{nj})$ vanishes $\Rightarrow$ There are infinite eigenvalues in $[-c, c]$

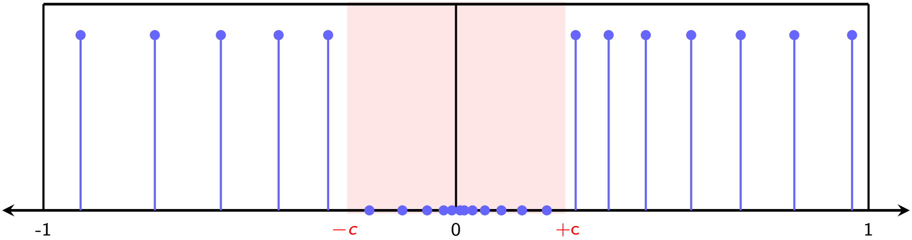
- Thus for any n and $\epsilon > 0$ we have some j for which $\Rightarrow \frac{\pi}{2} \frac{\|W - G_n\|}{d(\lambda_j, \lambda_{nj})} > \epsilon$

- Opposite of a convergence claim. $\Rightarrow$ For any $\epsilon > 0$, all $n > n_0$, and j $\Rightarrow \frac{\pi}{2} \frac{\|W - G_n\|}{d(\lambda_j, \lambda_{nj})} \leq \epsilon$



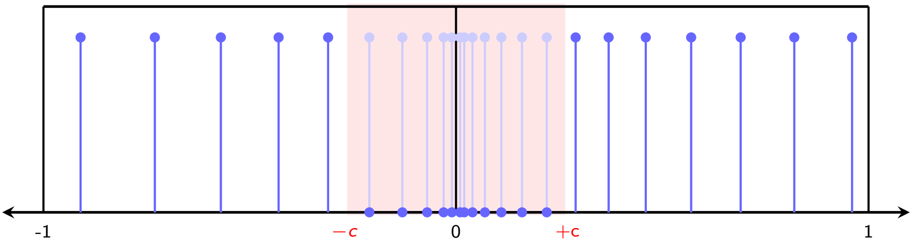
Definition (Graphon bandlimited signals)

A graphon signal (W, X) is c-bandlimited, with bandwith $c \in (0, 1]$, if $\hat{X}(\lambda_j) = 0$ for all $|\lambda_j| < c$.



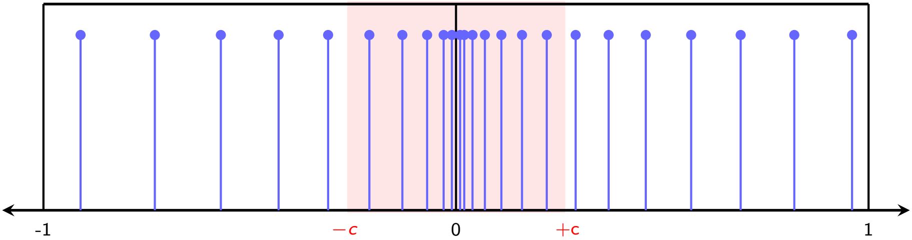
Bandlimited and Not-Bandlimited Graphon Signals

- Just to emphasize the simplicity of this definition consider a graphon signal that is Not-Bandlimited
- To make it bandlimited it suffices for us to nullify all of the WFT components in the interval $(-c, c)$



- Just to emphasize the simplicity of this definition consider a graphon signal that is Not-Bandlimited

- To make it bandlimited it suffices for us to nullify all of the WFT components in the interval $(-c, c)$



Theorem (GFT convergence for graphon bandlimited signals)

Let $(G_n, \mathbf{x}_n)$ be a sequence of graph signals converging to the c-bandlimited graphon signal (W, X).

There exists a sequence of permutations π_n such that

$$\text{GFT}\Big(\pi_n(G_n), \pi_n(\mathbf{x}_n)\Big) \to \text{WFT}\Big(W, X\Big)$$

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-9/ ■

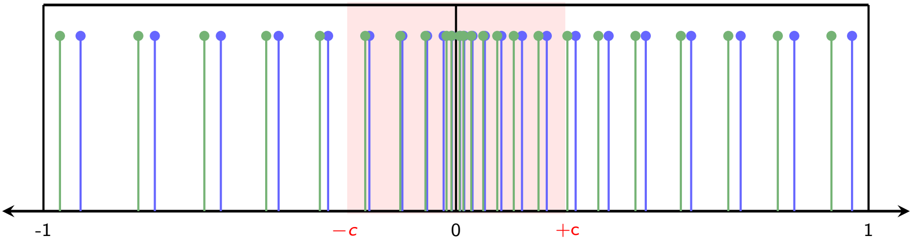
Theorem (iGFT convergence for graphon bandlimited signals)

Let $(G_n, \hat{\mathbf{x}}_n)$ be a sequence of GFTs converging to the WFT (W, X). The WFT is associated to a c-bandlimited graphon signal. There exists a sequence of permutations $\{\pi_n\}$ such that

$$\pi_n\Big(\text{iGFT}(\hat{\mathbf{x}}_n)\Big) \rightarrow \text{iWFT}\Big(\hat{X}\Big).$$

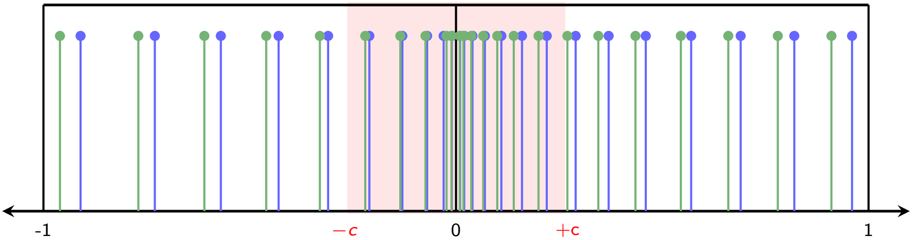
Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-9/ ■

- Convergence of GFT depends on convergence of graph eigenvalues to graphon eigenvalues
- As the number of nodes n grows, the eigenvalues of G_n converge to the eigenvalues of W.



- However, for large $|j|$ the graph and graphon eigenvalues become difficult to tell apart

- Therefore, the GFT only converges to the WFT for graphon bandlimited signals



Graphon Filters

- We define graphon filters and prove their frequency response, which is independent of the graphon.

- Apply the Graphon shift operator recursively to create the graphon diffusion sequence

$$\left(T_{\mathrm{W}}^{(k)} X \right)(v) \ = \ \int_0^1 \mathrm{W}(u,v) \left(T_{\mathrm{W}}^{(k-1)} X \right)(u)\, du \qquad T_{\mathrm{W}}^{(0)} X = X$$

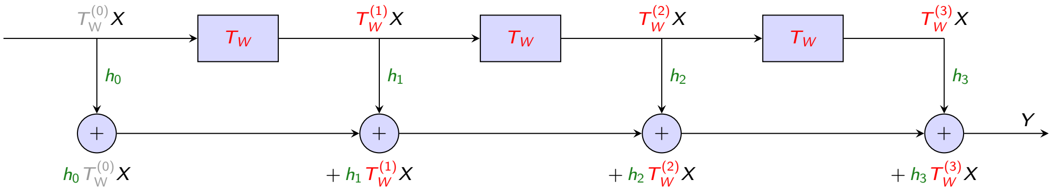
- A graphon filter of order K is defined by the filter coefficients h_k and produces outputs as per

$$Y(v) \ = \ \sum_{k=1}^{K} h_k \left(T_{\mathrm{W}}^{(k)} X \right)(v) \ = \ (T_H X)(v)$$

- A linear combination of the elements of the diffusion sequence modulated by coefficients h_k

- A graphon filter has the same algebraic structure of a graph filter $\Rightarrow Y(v) = \sum_{k=1}^{K} h_k \left(T_W^{(k)} X \right)(v)$

- Only difference is a change of shift operator $\Rightarrow T_W X \ : \ (T_W)X(v) \ = \ \int_0^1 W(u,v)\, X(u)\, du$



⇒ WFTs of input signal ⇒ $\hat{X}_j = \int_0^1 X(u)\varphi_j(u)du$ ⇒ WFT of output ⇒ $\hat{Y}_j = \int_0^1 Y(u)\varphi_j(u)du$

Theorem (Graph frequency representation of graphon filters)

Given a graphon filter T_{H} with coefficients h_k, the components of the graphon Fourier transforms of the input and output signals are related by

$$\hat{Y}_j = \sum_{k=0}^{K} h_k \lambda_j^k \hat{X}_j$$

- The same polynomial that defines the filter but with the eigenvalue λ_i as a variable

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-9/ ∎

- Graphon filters are pointwise in the WFT domain $\Rightarrow \hat{Y}_j = \sum_{k=0}^{K} h_k \lambda_j^k \hat{X}_j = h(\lambda_j)\, \hat{X}_j$

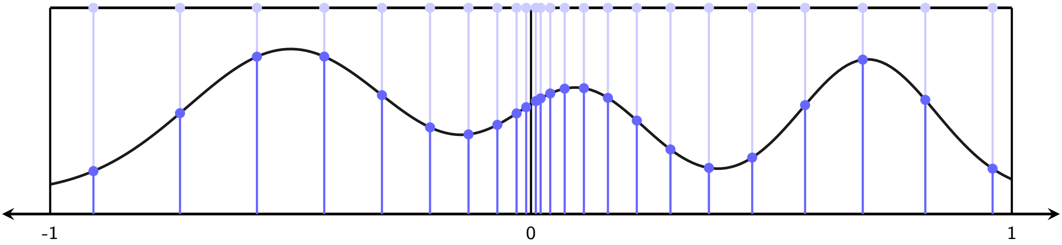
Definition (Frequency response of a graphon filter)

Given a graphon filter with coefficients $\mathrm{h} = \{h_k\}_{k=1}^{\infty}$ the frequency response is the polynomial

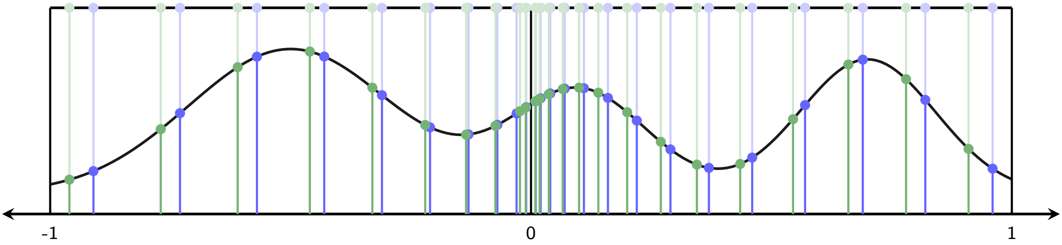
$$h(\lambda) = \sum_{k=0}^{\infty} h_k \lambda^k$$

- This is also the exact same definition of the frequency response of a graph filter with coefficients h_k

- The frequency response of a graphon filter and a graph filter with the same coefficients are the same
- Graphon filter instantiates graphon eigenvalues. Graph filter instantiates graph eigenvalues
- If graph sequence converges to a graphon eigenvalues converge $\Rightarrow$ The filter transfers



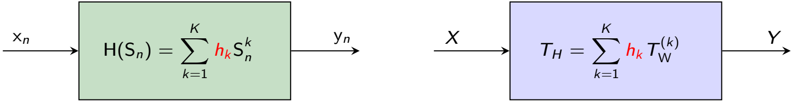
- The frequency response of a graphon filter and a graph filter with the same coefficients are the same
- Graphon filter instantiates graphon eigenvalues. Graph filter instantiates graph eigenvalues
- If graph sequence converges to a graphon eigenvalues converge $\Rightarrow$ The filter transfers



Convergence of Graph Filters in the Spectral Domain

- Convergence of graph filter sequences towards graphon filters for convergent graph signal sequences

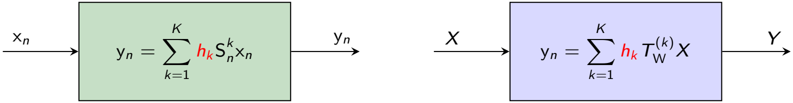
- Given coefficients h_k consider a graph filter sequence and a graphon filter with the same coefficients



- Does the graph filter sequence converge to the graphon filter? $\Rightarrow$ Not the most pertinent question

$\Rightarrow$ Filter convergence is important inasmuch as it implies convergence of filter outputs

- Given coefficients h_k consider a graph filter sequence and a graphon filter with the same coefficients



- Consider a convergent sequence of graph signals $(G_n, x_n) \to (W, X)$

$\Rightarrow$ Input graph signal x_n to graph filter $H(S_n)$ to produce output graph signal y_n

$\Rightarrow$ Input graphon signal X to graphon filter T_H to produce output graphon signal Y

- The graph signal sequence (G_n, y_n) converges to the graphon signal (W, Y) under some conditions

- Given filter coefficients h_k we have five polynomials which are the same except for their variables

- Two polynomials are representations in the node domain

$\Rightarrow$ The graph filter sequence defined on variable $\mathbf{S}_n$ $\Rightarrow \mathbf{H}(\mathbf{S}_n) = \sum_{k=1}^{K} h_k \mathbf{S}_n^k$

$\Rightarrow$ The graphon filter defined on variable $T_{\mathbf{W}}$ $\Rightarrow T_H = \sum_{k=1}^{K} h_k T_{\mathbf{W}}^{(k)}$

- Given filter coefficients h_k we have five polynomials which are the same except for their variables

- Three polynomials are representations in the spectral domain

$\Rightarrow$ The frequency response of the graph and graphon filters with variable λ $\Rightarrow \tilde{h}(\lambda) = \sum_{k=1}^{K} h_k \lambda^{(k)}$

$\Rightarrow$ The frequency representation of the graph filters with variable λ_{nj} $\Rightarrow \tilde{h}(\lambda_{nj}) = \sum_{k=1}^{K} h_k \lambda_{nj}^{(k)}$

$\Rightarrow$ The frequency representation of the graphon filter with variable λ_j $\Rightarrow \tilde{h}(\lambda_j) = \sum_{k=1}^{K} h_k \lambda_j^{(k)}$

⇒ Frequency representation of graph filters ⇒ $\tilde{h}(\lambda_{nj}) = \sum_{k=1}^{K} h_k \lambda_{nj}^k$

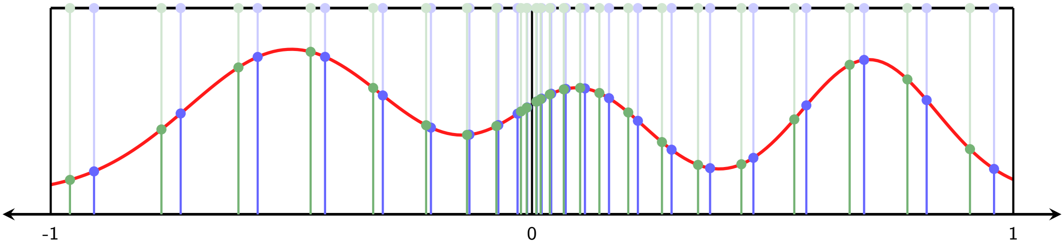
⇒ Frequency representation of graphon filter ⇒ $\tilde{h}(\lambda_j) = \sum_{k=1}^{K} h_k \lambda_j^k$

Theorem (Convergence of graph filter sequences in the frequency domain)

Consider filter coefficients h_k generating a sequence of graph filters $\mathbf{H}(\mathbf{S}_n)$ supported on the graph sequence G_n and a graphon filter T_H supported on the graphon W. If $G_n \to W$

$$\lim_{n\to\infty} \tilde{h}(\lambda_{nj}) = \tilde{h}(\lambda_j)$$

- Graph filter GFT representations converge to graphon filter WFT representation $\Rightarrow \lim_{n\to\infty} \tilde{h}(\lambda_{nj}) = \tilde{h}(\lambda_j)$
- This is true because eigenvalues converge and the frequency responses are the same
- This is not much to say $\Rightarrow$ GFT and WFT are representations. $\Rightarrow$ Filters operate in the node domain

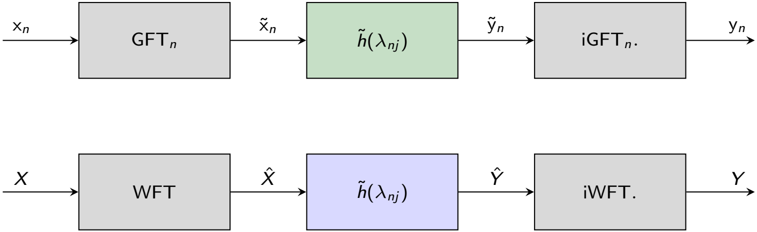


Convergence of Graph Filters in the Node Domain

- We leverage spectral domain convergence to prove convergence of graph filters in the node domain

$\Rightarrow$ Provides a first approach to the study of transferability of graph filters

- To prove convergence in the node domain we can go to the frequency domain and back



- Frequency representation of graph filters converge to frequency representation of graphon filter

  $\Rightarrow$ But the GFT and the iGFT do not converge $\Rightarrow$ Unless the signals are graphon bandlimited

- Input graph signal sequence (G_n, x_n) $\Rightarrow$ Generates output sequence (G_n, y_n) with $\mathrm{y}_n = \mathrm{H}(\mathrm{S}_n)\mathrm{x}_n$
- Input graphon signal (W, X) $\Rightarrow$ Generates output signal (W, Y) with $Y = T_H X$

Theorem (Graph filter convergence for bandlimited inputs)

Given convergent graph signal sequence $(G_n, \mathrm{x}_n) \rightarrow (W, X)$ and filters $\mathrm{H}(\mathrm{S}_n)$ and T_H generated by the same coefficients h_k. If the input signals are c-bandlimited

$$(\mathrm{G}_n, \mathrm{y}_n) \rightarrow (\mathrm{W}, Y)$$

The sequence of output graph signals converges to the output graphon signal

Lipschitz Graphon Filters

- Convergence for bandlimited input is easy. Also weak. Therefore cheap. A stronger result is possible

- Lipschitz graphon filters are filters with frequency responses that are Lipschitz in $[-1, 1]$

$$\Big| h(\lambda_1) - h(\lambda_2) \Big| \leq L \Big| \lambda_1 - \lambda_2 \Big|, \quad \text{for all } \lambda_1, \lambda_2 \in [0, 1]$$

- Claim convergence of graph filter sequence, despite lack of convergence of the GFT and the iGFT

Theorem (Graph filter convergence for Lipschitz continuous filters)

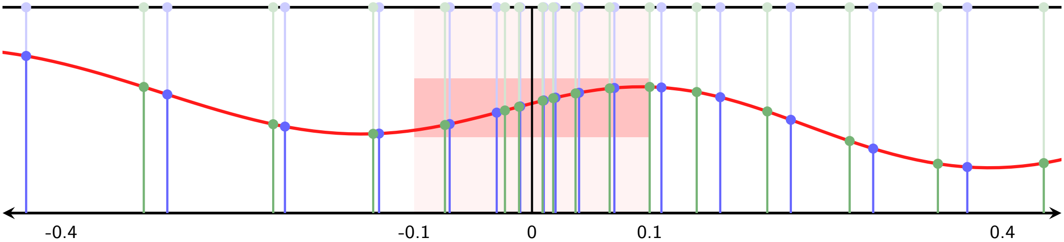
Given convergent graph signal sequence $(G_n, \mathrm{x}_n) \to (W, X)$ and filters $\mathrm{H}(\mathrm{S}_n)$ and T_H generated by the same coefficients h_k. If the frequency response $\tilde{h}(\lambda)$ is Lipschitz

$$(\mathrm{G}_n, \mathrm{y}_n) \to (\mathrm{W}, Y)$$

The sequence of output graph signals converges to the output graphon signal

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-10/ ■

- The challenge of filter convergence comes from the accumulation of eigenvalues around $\lambda = 0$
- Which causes complications with eigenvector convergence.
- Lipschitz continuity renders the effect void. All components are multiplied by similar numbers

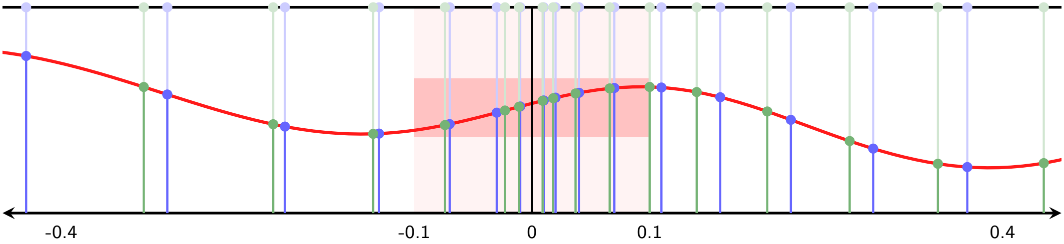


Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-10/ ■

- We identify a fundamental issue $\Rightarrow$ Transferability is counter to discriminability

  $\Rightarrow$ If the filter converges, it can't separate eigenvectors associated to eigenvalues close to $\lambda = 0$

- Characterization is just a limit $\Rightarrow$ Work on a finite-n transference bounding

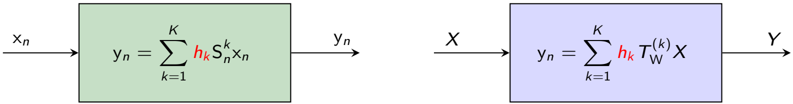


Graphon Filters are Generative Models for Graph Filters

- Graph filters can approximate graphon filters under certain conditions. We discuss them now.

- For a converging graph sequence, graph filters converge asymptotically to graphon filters

- Thus, as n grows, the graph filters become more similar to the graphon filter

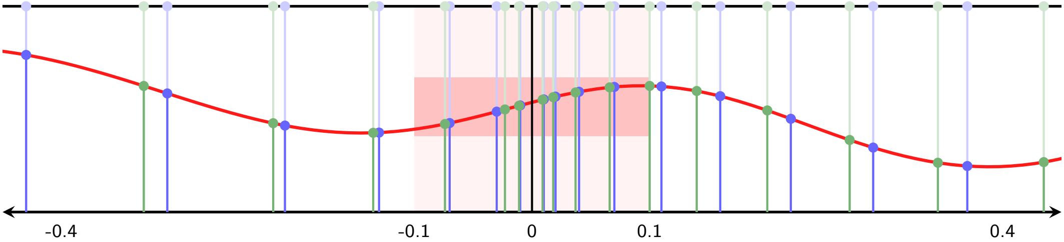


- And we can then use a graph filter as a surrogate for the graphon filter

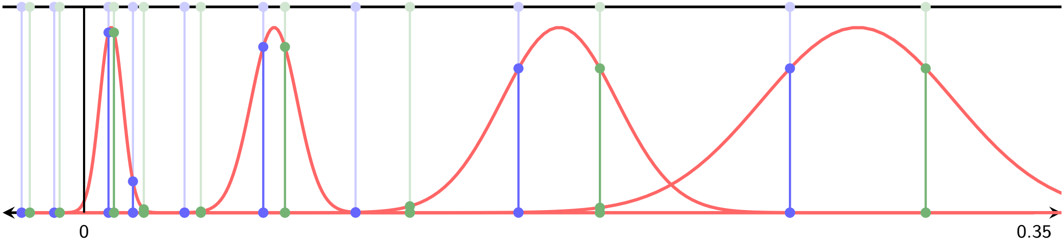
- We now want to quantify the quality of that approximation for different values of n

Small Eigenvalues are Hard to Discriminate

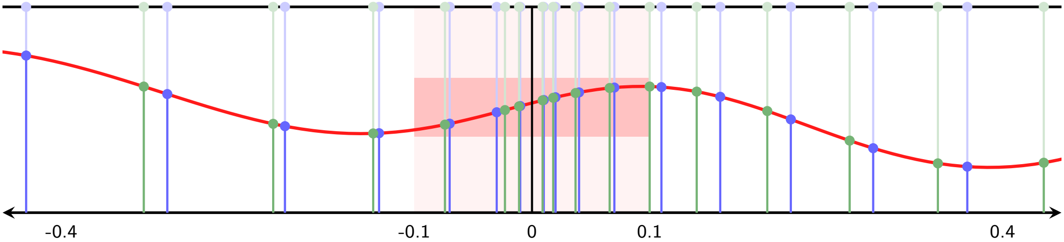
- Graphon eigenvalues accumulate at $\lambda = 0$
- Making it hard to match graph eigenvalues to the corresponding graphon eigenvalues if λ is small



- Which in turn makes it hard to discriminate consecutive eigenvalues in that range
- If the filter changes rapidly near zero, it may modify the graph and graphon eigenvalues differently
- To obtain good approximations, we must then assume filters do not change much around $\lambda = 0$

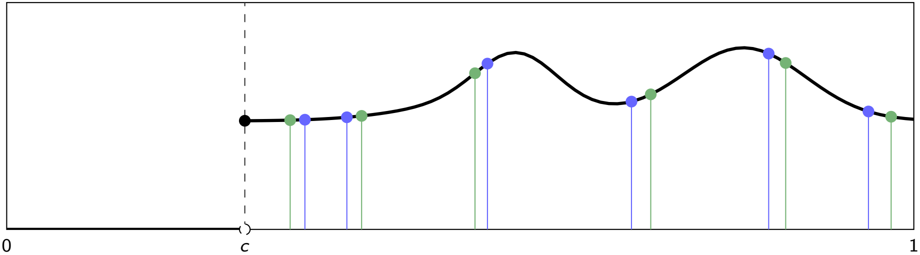


- Which in turn makes it hard to discriminate consecutive eigenvalues in that range
- If the filter changes rapidly near zero, it may modify the graph and graphon eigenvalues differently
- To obtain good approximations, we must then assume filters do not change much around $\lambda = 0$



- Graphon eigenvalues tend to zero as the index i grows $\Rightarrow \lim_{i\to\infty} \lambda_i = \lim_{i\to\infty} \lambda_{-i} = 0$

- Low-pass graphon filters must thus be zero for $\lambda < c$. Constant c determines the filter's band.



- The filter removes high frequency components. But low-frequency components are not affected.

(A1) The graphon W is L_1-Lipschitz $\Rightarrow$ For all arguments (u_1, v_1) and (u_2, v_2), it holds

$$\Big| W(u_2, v_2) - W(u_1, v_1) \Big| \leq L_1 \Big(\big| u_2 - u_1 \big| + \big| v_2 - v_1 \big| \Big)$$

(A2) The filter's response is L_2-Lipschitz and normalized $\Rightarrow$ For all λ_1, λ_2 and λ we have

$$\big| \tilde{h}(\lambda_2) - \tilde{h}(\lambda_1) \big| \leq L_2 \big| \lambda_2 - \lambda_1 \big| \quad \text{and} \quad \big| h(\lambda) \big| \leq 1$$

(A3) The graphon signal X is L_3-Lipschitz $\Rightarrow$ For all u_1 and u_2

$$\big| X(u_2) - X(u_1) \big| \leq L_3 \big| u_2 - u_1 \big|$$

- We fix a bandwidth $c > 0$ to separate eigenvalues not close to $\lambda = 0$ and define

(D1) The c-band cardinality of G_n is the number of eigenvalues with absolute value larger than c

$$B_{nc} = \#\left\{ \lambda_{ni} \ : \ |\lambda_{ni}| > c \right\}$$

(D2) The c-eigenvalue margin of graph G_n is the

$$\delta_{nc} = \min_{i, j \neq i} \left\{ |\lambda_{ni} - \lambda_j| \ : \ |\lambda_{ni}| > c \right\}$$

- Where λ_{ni} are eigenvalues of the shift operator S_n and λ_j are eigenvalues of graphon W

Theorem (Graphon filter approximation by graph filter for low-pass filters)

Consider a graphon filter $Y = \Phi(X; h, W)$ and a graph filter $y_n = \Phi(x_n; h, S_n)$ instantiated from Y. With Definitions **(D1)** - **(D2)**, Assumptions **(A1)** - **(A3)**, and

(A4) $h(\lambda)$ is zero for $|\lambda| < c$

The difference between Y and $Y_n = \Phi(X_n; h, W_n)$ (graph filter induced by y_n) is bounded by

$$\|Y - Y_n\|_{L_2} \leq \sqrt{L_1}\left(L_2 + \frac{\pi n_c}{\delta_{nc}}\right) n^{-\frac{1}{2}} \|X\|_{L_2} + \frac{L_3}{\sqrt{3}} n^{-\frac{1}{2}}$$

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-10/ ■

- High-pass filters have null frequency response for $|\lambda| > c$, removing low-frequency components

- Moreover, we consider filters that have low variability around $\lambda = 0$

- This makes it easier to match graph eigenvalues to graphon eigenvalues around $\lambda = 0$

Theorem (Graphon filter approximation by graph filter for high-pass filters)

Consider a graphon filter $Y = \Phi(X; h, W)$ and a graph filter $y_n = \Phi(x_n; h, S_n)$ instantiated from Y. With Definitions **(D1)** - **(D2)**, Assumptions **(A1)** - **(A3)**, and

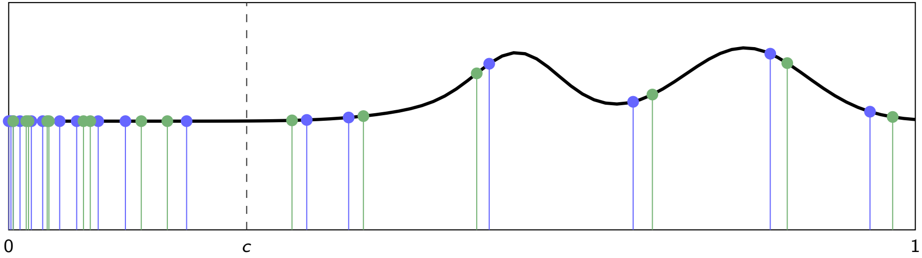
(A4) $h(\lambda)$ is zero for $|\lambda| > c$

The difference between Y and $Y_n = \Phi(X_n; h, W_n)$ (graph filter induced by y_n) is bounded by

$$\|Y - Y_n\|_{L_2} \leq L_2 c \|X\|$$

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-10/ ■

- Filter response has low variability for $|\lambda| < c$. Where the eigenvalues of the graphon accumulate
- For $|\lambda| > c$, graphon eigenvalues are countable. And easier to match to those of the graph



- A Lipschitz filter with variable band is the composition of a low-pass filter and a high-pass one

Theorem (Graphon filter approximation by graph filter)

Consider a graphon filter $Y = \Phi(X; \mathrm{h}, \mathrm{W})$ and a graph filter $\mathrm{y}_n = \Phi(\mathrm{x}_n; \mathrm{h}, \mathrm{S}_n)$ instantiated from Y. With Definitions **(D1)** - **(D2)**, Assumptions **(A1)** - **(A3)**, and

(A4) $h(\lambda)$ has low variability for $|\lambda| < c$

The difference between Y and $Y_n = \Phi(X_n; \mathrm{h}, \mathrm{W}_n)$ (graph filter induced by y_n) is bounded by

$$\|Y - Y_n\|_{L_2} \leq \sqrt{L_1}\left(L_2 + \frac{\pi n_c}{\delta_{nc}}\right) n^{-\frac{1}{2}} \|X\|_{L_2} + \frac{L_3}{\sqrt{3}} n^{-\frac{1}{2}} + L_2 c \|X\|$$

- Filter with variable band is the sum of an L_2-Lipschitz filter $h_1(\lambda)$ with $h_1(\lambda) = 0$ for $|\lambda| < c$
- And a high-pass filter $h_2(\lambda)$ with $h_2(\lambda)$ showing low variability for $|\lambda| < c$ and 0 otherwise
- Thus, by the triangle inequality

$$\|Y - Y_n\|_{L_2} = \|T_{\mathrm{H}}X - T_{\mathrm{H}_n}\|_{L_2} \leq \|T_{\mathrm{H}_1}X - T_{\mathrm{H}_{1_n}}X_n\|_{L_2} + \|T_{\mathrm{H}_2}X - T_{\mathrm{H}_{2_n}}X_n\|_{L_2}$$

- We know the first-term on the right-hand side. It's the bound for low-pass filters
- And the second-term on the right-hand side is the bound for constant filters
- Summing up the two bounds, we then prove our result for Lipschitz filters with variable band

Parse the Bound

Theorem (Graphon filter approximation by graph filter)

The difference between Y and $Y_n = \Phi(X_n; \mathrm{h}, \mathrm{W}_n)$ (graph filter induced by y_n) is bounded by

$$\|Y - Y_n\|_{L_2} \leq \sqrt{L_1}\left(L_2 + \frac{\pi n_c}{\delta_{nc}}\right) n^{-\frac{1}{2}} \|X\|_{L_2} + \frac{L_3}{\sqrt{3}} n^{-\frac{1}{2}} + L_2 c \|X\|$$

- Bound depends on the filter transferability constant and on the difference between X and X_n
- Transferability constant depends on the graphon via L_1 which also affects the graphon variability
- As n grows, the transferability constant dominates the bound

Theorem (Graphon filter approximation by graph filter)

The difference between Y and $Y_n = \Phi(X_n; \mathrm{h}, \mathrm{W}_n)$ (graph filter induced by y_n) is bounded by

$$\|Y - Y_n\|_{L_2} \leq \sqrt{L_1}\left(L_2 + \frac{\pi n_c}{\delta_{nc}}\right) n^{-\frac{1}{2}} \|X\|_{L_2} + \frac{L_3}{\sqrt{3}} n^{-\frac{1}{2}} + L_2 c \|X\|$$

- Transferability constant depends on the filter parameters L_2, n_c and δ_{nc}
- Filter's Lipschitz constant L_2 and filter's band $[c, 1]$ determine variability of the spectral response
- Number of eigenvalues in the passing band has to be limited: $n_c < \sqrt{n}$
- This ensures eigenvalues of W_n converge to those of W. And thus so does the filter approximation

- We identify a fundamental issue $\Rightarrow$ Good approximations are counter to discriminability
  - $\Rightarrow$ Tight approximation bounds require filters with low variability around $\lambda = 0$
  - $\Rightarrow$ But then the filter can't discriminate components associated to eigenvalues close to $\lambda = 0$
- That is less of an issue for larger graphs. Filter approximation requires $n_c < \sqrt{n}$
  - $\Rightarrow$ As n grows, we can afford a larger number of eigenvalues n_c in the passing band
  - $\Rightarrow$ Improving discriminability without penalizing the approximation bound

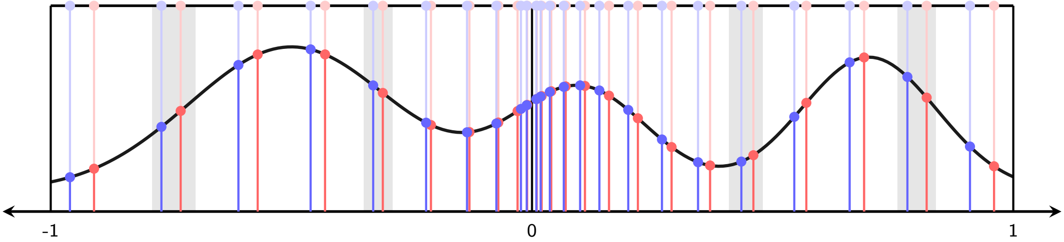
Transferability of Graph Filters: Theorem

- We show that graph filters are transferable across graphs that are drawn from a common graphon

- Have not proven transferability $\Rightarrow$ Have proven that graph filters are close to graphon filters

$\Rightarrow$ Graph G_n with n nodes sampled from graphon W

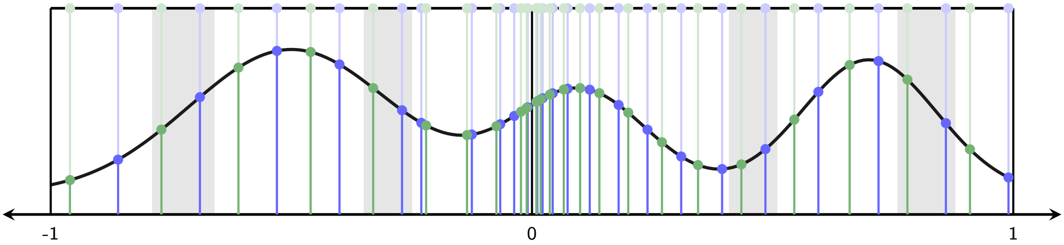
$\Rightarrow$ Have shown that graph filter $\mathrm{H}(\mathrm{S}_n)$ running on G_n is close to the graphon filter T_H



- Transferability means that two different graphs with different number of nodes are close

$\Rightarrow$ Graph G_n and graph G_m with $n \neq m$ nodes. Both sampled from graphon W

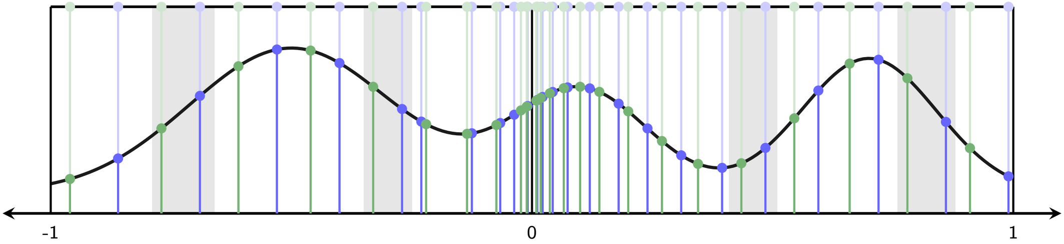
$\Rightarrow$ Want to show that graph filter $\text{H}(\text{S}_n)$ and graph filter $\text{H}(\text{S}_m)$ are close



- But graph filters are close because they are both close to the graphon filter

  ⇒ Graph filter $\mathbf{H}(\mathbf{S}_n)$ close to graphon filter T_H. Graph filter $\mathbf{H}(\mathbf{S}_m)$ close to graphon filter T_H

  ⇒ Graph filter $\mathbf{H}(\mathbf{S}_n)$ is close to graph filter $\mathbf{H}(\mathbf{S}_m)$ ⇒ This is just the triangle inequality



- Consider graph signals $(\mathbf{S}_n, \mathbf{x}_n)$ and $(\mathbf{S}_m, \mathbf{x}_m)$ sampled from the graphon signal (W, X)

- Given filter coefficients h_k we process signals on their respective graphs

  $\Rightarrow$ Run filter with coefficients h_k on graph S_n to process x_n $\Rightarrow \mathbf{y}_n = \mathbf{H}(\mathbf{S}_n)\mathbf{x}_n = \sum_{k=1}^{K} h_k \mathbf{S}_n^k \mathbf{x}_n$

  $\Rightarrow$ Run filter with coefficients h_k on graph S_m to process x_m $\Rightarrow \mathbf{y}_m = \mathbf{H}(\mathbf{S}_m)\mathbf{x}_m = \sum_{k=1}^{K} h_k \mathbf{S}_m^k \mathbf{x}_n$

- Since they have different number of components we compare induced graphon signals Y_n and Y_m

(A1) The graphon W is L_1-Lipschitz $\Rightarrow$ For all arguments (u_1, v_1) and (u_2, v_2), it holds

$$\Big| W(u_2, v_2) - W(u_1, v_1) \Big| \; \leq \; L_1 \Big(\, \big| \, u_2 - u_1 \, \big| \; + \; \big| \, v_2 - v_1 \, \big| \, \Big)$$

(A2) The filter's response is L_2-Lipschitz and normalized $\Rightarrow$ For all λ_1, λ_2 and λ we have

$$\big| \, \tilde{h}(\lambda_2) - \tilde{h}(\lambda_1) \, \big| \; \leq \; L_2 \big| \, \lambda_2 - \lambda_1 \, \big| \qquad \text{and} \qquad \big| \, h(\lambda) \, \big| \; \leq \; 1$$

(A3) The graphon signal X is L_3-Lipschitz $\Rightarrow$ For all u_1 and u_2

$$\big| \, X(u_2) - X(u_1) \, \big| \; \leq \; L_3 \big| u_2 - u_1 \big|$$

- We fix a bandwidth $c > 0$ to separate eigenvalues not close to $\lambda = 0$ and define

(D1) The c-band cardinality of G_n is the number of eigenvalues with absolute value larger than c

$$B_{nc} = \#\Big\{ \lambda_{ni} \,:\, |\lambda_{ni}| > c \Big\}$$

(D2) The c-eigenvalue margin of of graph G_n is the

$$\delta_{nc} = \min_{i, j \neq i} \Big\{ |\lambda_{ni} - \lambda_j| \,:\, |\lambda_{ni}| > c \Big\}$$

- Where λ_{ni} are eigenvalues of the shift operator S_n and λ_j are eigenvalues of graphon W

Theorem (Graph filter transferability)

Consider graph signals $(\mathbf{S}_n, \mathbf{x}_n)$ and $(\mathbf{S}_m, \mathbf{x}_m)$ sampled from graphon signal (W, X) along with filter outputs $\mathbf{y}_n = \mathbf{H}(\mathbf{S}_n)\mathbf{x}_n$ and $\mathbf{y}_m = \mathbf{H}(\mathbf{S}_m)\mathbf{x}_m$. With Assumptions (A1)-(A3) and Definitions (D1)-(D2) the difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq \sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{2L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + L_2 c\|X\|$$

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-10/ ■

Transferability of Graph Filters: Remarks

- We present remarks on the transferability theorem of graph filters sampled from a graphon filter

Theorem (Graph filter transferability)

The difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq \sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{2L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + L_2 c\|X\|$$

Thing 1: A term that comes from the discretization of the graphon signal $\Rightarrow$ Not very important

Thing 2: A term coming from filter variability at eigenvalues $|\lambda| > c$ $\Rightarrow$ The easy components

Thing 3: A term coming from filter variability at eigenvalues $|\lambda| \leq c$ $\Rightarrow$ The difficult components

Theorem (Graph filter transferability)

The difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq \sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{2L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + L_2 c\|X\|$$

- As $(n, m) \to \infty$ most of the transferability error decreases with the square root of the graph sizes
- We can also afford smaller bandwidth limit c $\Rightarrow$ Transfer filters closer to $\lambda = 0$
- Sharper filter responses (larger Lipschitz constant L_2) $\Rightarrow$ Transfer more discriminative filters

Theorem (Graph filter transferability)

The difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq \sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{2L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + L_2 c\|X\|$$

- Graph signals and graphons with rapid variability make filter transference more difficult
- This is because of sampling approximation error $\Rightarrow$ Not fundamental
- The constants can be sharpened with modulo-permutation Lipschitz constants

Theorem (Graph filter transferability)

The difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq \sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{2L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + L_2 c\|X\|$$

- Filters that are more discriminative are more difficult to transfer

  $\Rightarrow$ True in the part of the bound related to easy components associated with eigenvalues $|\lambda| > c$

  $\Rightarrow$ True in the part of the bound related to difficult components associated with eigenvalues $|\lambda| \leq c$

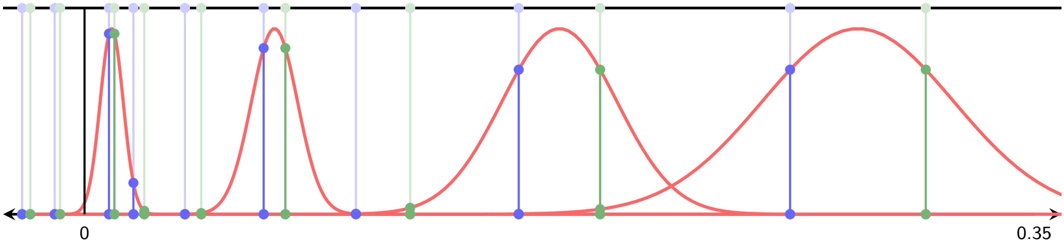
Theorem (Graph filter transferability)

The difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq \sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{2L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + L_2 c\|X\|$$

- Bound is parametric on the bandwidth c $\Rightarrow$ Different c result in different values for the bound
- Increase c-band cardinality or decrease c-eigenvalue margin $\Rightarrow$ More challenging transferability
- A property of the graphon $\Rightarrow$ Since eigenvalues converge B_{nc} and δ_{nc} converge

- If we fix n and m we observe emergence of a transferability vs discriminability non-tradeoff
- Discriminating around $\lambda = 0$ needs large Lipschitz constant L_2 $\Rightarrow$ Useless transferability bound
- To make transferability and discriminability compatible $\Rightarrow$ Graph Neural Networks



Transferability of GNNs

- We define graphon neural networks and discuss their interpretation as generative models for GNNs
- We show that graph neural networks inherit the transferability properties of graph filters

- Graph filters are transferable $\Rightarrow$ we can expect GNNs to inherit transferability from graph filters

- To analyze GNN transferability, we we first define Graphon Neural Networks (WNNs)

- The lth layer of a WNN composes a graphon convolution with parameters h and a nonlinearity σ

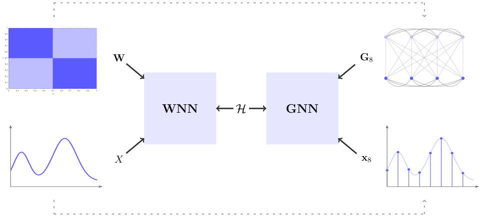
$$X_l^f = \sigma\left(\sum_{g=1}^{F_{l-1}} h_{kl}^{fg} T_W^{(k)} X_{l-1}^g\right)$$

L layers, $1 \leq f \leq F_l$ output features per layer. WNN input is $X_0 = X$. Output is $Y = X_L$

- Can be represented as $Y = \Phi(\mathcal{H}; W; X)$ with coefficients $\mathcal{H} = \{\mathrm{h}_{kl}^{fg}\}_{k,l,f,g}$. Just like the GNN

WNNs as Generative Models for GNNs

- As in the GNN map $\Phi(\mathcal{H}; S; x)$, in the WNN $\Phi(\mathcal{H}; W; X)$, the set $\mathcal{H}$ doesn't depend on the graphon
- Therefore, we can use WNNs to instantiate GNNs $\Rightarrow$ the WNN is a generative model for GNNs



- We will consider GNNs $\Phi(\mathcal{H}; S_n; x_n)$ instantiated from $\Phi(\mathcal{H}; W; X)$ on weighted graphs G_n

$$[S_n]_{ij} = W(u_i, u_j) \qquad [x_n]_i = X(u_i)$$

- Consider a graph signal $(\mathbf{S}_n, \mathbf{x}_n)$ sampled from the graphon signal (W, X)

- Given WNN coefficients $\mathcal{H}$ for L layers, width $F_l = F$ for $1 \leq l < L$, and $F_0 = F_L = 1$

  $\Rightarrow$ Run WNN with coefficients $\mathcal{H}$ on graphon W to process X $\Rightarrow Y = \Phi(\mathcal{H}; W, X)$

  $\Rightarrow$ Run GNN with coefficients $\mathcal{H}$ on graph S_n to process x_n $\Rightarrow \mathbf{y}_n = \Phi(\mathcal{H}; S_n, x_n)$

- Since one is a vector and the other a function we consider the induced graphon signal Y_n

(A1) The graphon W is L_1-Lipschitz $\Rightarrow$ For all arguments (u_1, v_1) and (u_2, v_2), it holds

$$\left| W(u_2, v_2) - W(u_1, v_1) \right| \leq L_1 \Big(\left| u_2 - u_1 \right| + \left| v_2 - v_1 \right| \Big)$$

(A2) The filter's response is L_2-Lipschitz and normalized $\Rightarrow$ For all λ_1, λ_2 and λ we have

$$\left| \tilde{h}(\lambda_2) - \tilde{h}(\lambda_1) \right| \leq L_2 \left| \lambda_2 - \lambda_1 \right| \qquad \text{and} \qquad \left| h(\lambda) \right| \leq 1$$

(A3) The graphon signal X is L_3-Lipschitz $\Rightarrow$ For all u_1 and u_2

$$\left| X(u_2) - X(u_1) \right| \leq L_3 \left| u_2 - u_1 \right|$$

(A4) The nonlinearities σ are normalized Lipschitz and $\sigma(0) = 0$ $\Rightarrow$ For all x and y

$$\left| \sigma(x) - \sigma(y) \right| \leq |x - y|$$

- We fix a bandwidth $c > 0$ to separate eigenvalues not close to $\lambda = 0$ and define

(D1) The c-band cardinality of G_n is the number of eigenvalues with absolute value larger than c

$$B_{nc} = \#\left\{ \lambda_{ni} \ : \ |\lambda_{ni}| > c \right\}$$

(D2) The c-eigenvalue margin of of graph G_n is the

$$\delta_{nc} = \min_{i, j \neq i} \left\{ |\lambda_{ni} - \lambda_j| \ : \ |\lambda_{ni}| > c \right\}$$

- Where λ_{ni} are eigenvalues of the shift operator S_n and λ_j are eigenvalues of graphon W

Theorem (GNN-WNN approximation)

Consider the graph signal (S_n, x_n) sampled from the graphon signal (W, X) along with the GNN output $y_n = \Phi(\mathcal{H}; S_n, x_n)$ and WNN output $Y = \Phi(\mathcal{H}; W, X)$. With Assumptions (A1)-(A4) and Definitions (D1)-(D2) the norm difference $\|Y_n - Y\|$ is bounded by

$$\|Y - Y_n\| \leq LF^{L-1}\sqrt{L_1}\left(L_2 + \pi\frac{B_{nc}}{\delta_{nc}}\right)\left(\frac{1}{\sqrt{n}}\right)\|X\| + \frac{L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}}\right) + LF^{L-1}L_2 c\|X\|$$

Proof: See course webpage https://gnn.seas.upenn.edu/lectures/lecture-10/ ■

- The error incurred when using a GNN to approximate a WNN can be upper bounded
- Same comments as for graph and graphon filters apply. With additional dependence on L and F
- Distances between GNNs and WNN can be combined to calculate distance between GNNs
- GNNs $Y_n = \Phi(\mathcal{H}; W_n, x_n)$ and $Y_m = \Phi(\mathcal{H}; W_m, x_m)$ instantiated from WNN $Y = \Phi(\mathcal{H}; W, X)$

$$\|Y_n - Y_m\| = \|Y_n - Y + Y - Y_m\| \leq \|Y_n - Y\| + \|Y - Y_m\|$$

- The inequality follows from the triangle inequality. By which we have proved GNN transferability

- Consider graph signals $(\mathbf{S}_n, \mathbf{x}_n)$ and $(\mathbf{S}_m, \mathbf{x}_m)$ sampled from the graphon signal (W, X)

- Given GNN coefficients $\mathcal{H}$ for L layers, width $F_l = F$ for $1 \leq l < L$, and $F_0 = F_L = 1$

  $\Rightarrow$ Run GNN with coefficients $\mathcal{H}$ on graph S_n to process x_n $\Rightarrow y_n = \Phi(\mathcal{H}; S_n, x_n)$

  $\Rightarrow$ Run filter with coefficients $\mathcal{H}$ on graph S_m to process x_m $\Rightarrow y_m = \Phi(\mathcal{H}; S_m, x_n)$

- Since they have different number of components we compare induced graphon signals Y_n and Y_m

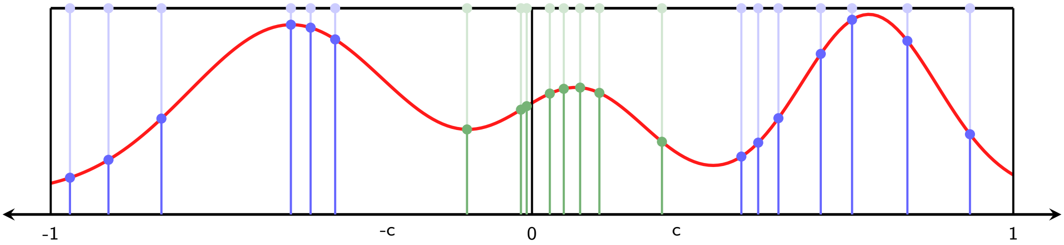
Theorem (GNN transferability)

Consider graph signals $(\mathbf{S}_n, \mathbf{x}_n)$ and $(\mathbf{S}_m, \mathbf{x}_m)$ sampled from graphon signal (W, X) along with GNN outputs $\mathbf{y}_n = \Phi(\mathcal{H}; S_n, \mathbf{x}_n)$ and $\mathbf{y}_m = \Phi(\mathcal{H}; S_m, \mathbf{x}_m)$. With Assumptions (A1)-(A4) and Definitions (D1)-(D2) the difference norm of the respective graphon induced signals is bounded by

$$\|Y_n - Y_m\| \leq LF^{L-1}\sqrt{L_1}\left(L_2 + \pi\frac{\max(B_{nc}, B_{mc})}{\min(\delta_{nc}, \delta_{mc})}\right)\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)\|X\| + \frac{L_3}{\sqrt{3}}\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right) + LF^{L-1}L_2 c\|X\|$$

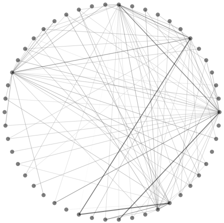
- Same comments as in the case of graph filter transferability. With additional dependence on L, F

- The transferability-discriminability trade-off looks the same. But it is helped by the nonlinearities
- At each layer of the GNN, the nonlinearities σ scatter eigenvalues from $|\lambda| \leq c$ to $|\lambda| > c$

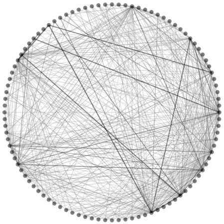


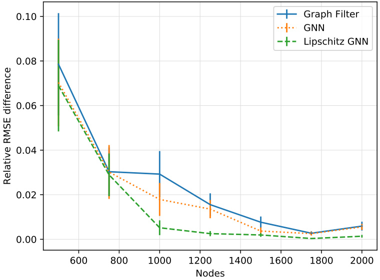
- Nonlinearities allows $\downarrow c$ and $\uparrow L_2$ $\Rightarrow$ increasing discriminability while retaining transferability
- For the same level of discriminability, GNNs are more transferable than graph filters

- Transferability of graph neural networks is ready to verify in practice $\Rightarrow$ recommendation system



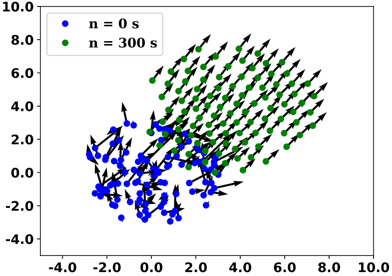
$\rightarrow$

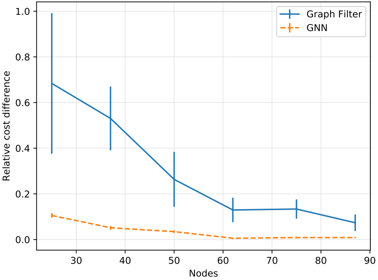




- Performance difference on training and target graphs decreases as size of training graph grows
- GNNs appear to be more transferable than graph convolutional filters $\Rightarrow$ better ML model

- Transferability of graph neural networks is ready to verify in practice $\Rightarrow$ decentralized robot control





- Performance difference on training and target graphs decreases as size of training graph grows

- GNNs appear to be more transferable than graph convolutional filters $\Rightarrow$ better ML model

Takeaways

GNNs are more transferable than graph convolutional filters

GNNs are more transferable because of their mixing properties

- Empirical and theoretical evidence support using GNNs for large-scale graph machine learning

Limitations and Extensions

Limitation 1: When Transferability is Not Enough

- Using the transferability property to train GNNs for large graphs $\mathbf{G}_N$ might not be sufficient
- The difference between the outputs of the same GNN decreases with the training graph size

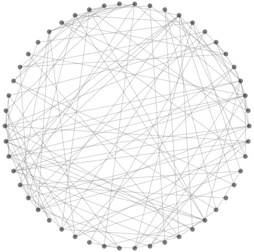
  $\Rightarrow$ But no guarantee that the learned GNN will actually perform well on the large graph

- In safety-critical applications (e.g. multi-agent systems), the error allowance is small

  $\Rightarrow$ The minimum training graph size n in this case is likely still too large $\Rightarrow \mathcal{O}(N)$

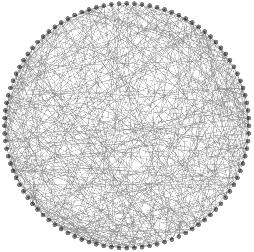
Solution: leverage convergence/transferability in the training algorithm of the large-scale GNN

- We train GNNs on sequences of growing graphs $\Rightarrow$ trade-off between costs and performance



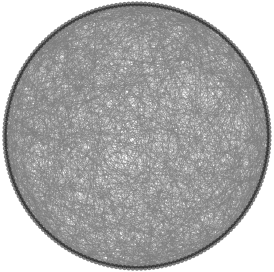
10^2 nodes

$\Rightarrow$



10^3 nodes

$\Rightarrow$



10^4 nodes

- Leverage transferability to increase the graph as we improve the GNN $\Rightarrow$ Learning By Transference

- Obtain the NN coefficients $\mathcal{H}$ that minimize a loss ℓ over an unknown distribution
  - $\Rightarrow$ Large Scale Graph Model: predict graphon label Y given graphon signal X
  - $\Rightarrow$ Graphs: predict graph signal label y given graph signal x

Learning Problem on graphon

$$\underset{\mathcal{H}}{\text{minimize}} \quad \mathbb{E}\Big[\ell(Y, \Phi(X; \mathcal{H}, \mathrm{W}))\Big]$$

Learning Problem on graph

$$\underset{\mathcal{H}}{\text{minimize}} \quad \mathbb{E}\Big[\ell(\mathrm{y}, \Phi(\mathrm{x}; \mathcal{H}, \mathrm{S}))\Big]$$

- Given the regularity in the graphon W the two problem are close $\Rightarrow$ the number of nodes in graph n

- We want to obtain the filters $\mathcal{H}$ that obtain the best performance on the very large graph

- We show that these two gradients are close and that the distance depends on the number of nodes
- By successively increasing the number of nodes, we can follow the learning direction on the graphon

Learning by Transference Convergence Theorem

Under smoothness assumptions, if the norm of the WNN gradient is larger than the difference between the gradients then,

$$\mathbb{E}[\|\nabla_{\mathcal{H}}\ell(Y,\Phi(X;\mathcal{H}_{k^*},\mathrm{W}))\|] \leq \alpha + \epsilon \quad \text{taking } k^* = \mathcal{O}(1/\epsilon^2) \text{ steps of Learning by Transference}$$

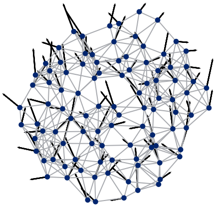
where α is a constant that depends on the parameter of the problem.

- The optimal WNN can be obtained by taking learning steps on growing GNNs $\Rightarrow$ more efficient

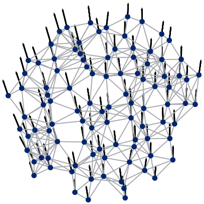
Cerviño-Ruiz-Ribeiro, *Learning by Transference: Training Neural Networks on Growing Graphs, TSP 2023*, arxiv.org/abs/2106.03693

Distributed Control Benchmark: Flocking Setup

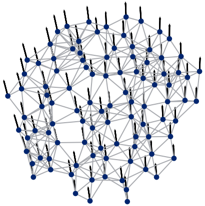
- Control a multi agent decentralized setup that aims to coordinate velocities and avoid collisions
- We construct the communication graph S_n using the proximity between agents
- Each agents controls their own acceleration $a = \Phi(x_n; \mathcal{H}, S_n)$ $\Rightarrow$ imitate a centralized controller y_n



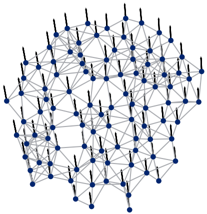
Initial Setup - $t = 0s$



t=0.5s

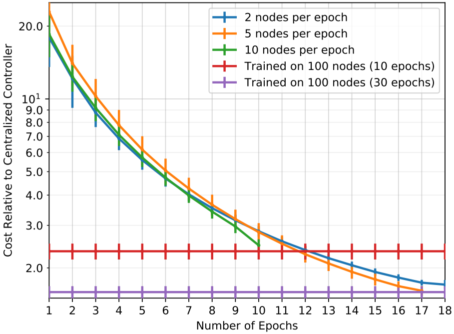


t=0.5s

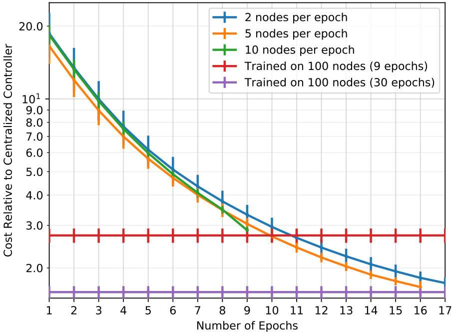


Final Setup - t=2s

- Showcase learning by transference with different number of initial nodes and nodes added per epoch
- We compare the control cost to the one we would have obtained training in the large scale graph



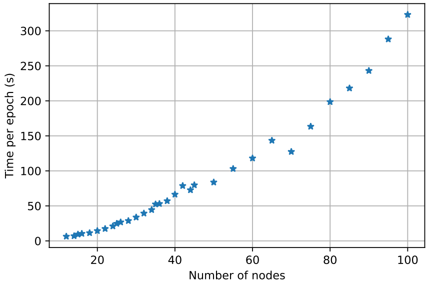
Start at 10 nodes



Start at 20 nodes

- We obtain a comparable control cost to the large scales graph by training on growing graphs

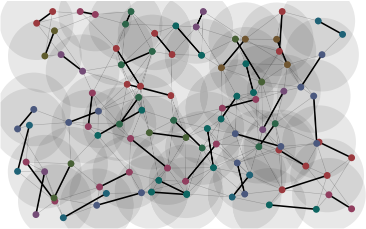
- We look at the time required to compute an epoch as a function of the number of nodes

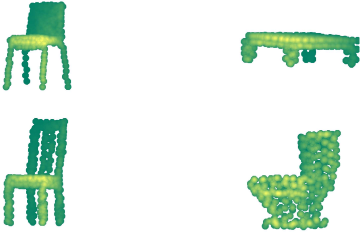


- Start with 10 nodes and adding 2 per epoch
  $\Rightarrow$ 523s $\sim$ 9 minutes
- Normal train 30 epochs with 100 nodes
  $\Rightarrow$ 9690s $\sim$ 2.7 hours
- Normal train 9 epochs with 100 nodes
  $\Rightarrow$ 2907s $\sim$ 49 minutes

- Learning by Transference reduces the training times by up to $\approx$ 20 times without compromising accuracy

- Graphon is good model for limit of dense graphs, but not as suitable for real-world, sparser graphs
- Signals on geometric graphs appear in several application domains
  - ⇒ Wireless communication networks, 3D point clouds, climate data





- We develop a limit theory of signal processing (SP) on geometric graphs
  - ⇒ Geometric graphs converge (or are sampled from) Manifolds
  - ⇒ Convergence. Stability. Wireless Networks. Vector Fields

- Manifold $\mathcal{M} \subset \mathbb{R}^{\mathrm{N}}$ is d-dimensional with Laplace-Beltrami (LB) operator $\mathcal{L}$
- A Manifold filter with coefficients $\tilde{h}$ is defined by the input-output relationship

$$g(x) \;=\; \int_0^\infty \tilde{h}(t)\, e^{-t\mathcal{L}}\, f(x)\mathrm{d}t \;=\; \mathrm{h}(\mathcal{L})\, f(x)\,.$$

- Discretizing a manifold filter yields a graph filter with shift operator $e^{-T_s \mathrm{L}_n}$

$$\mathrm{g} \;=\; \sum_{k=0}^{K_t-1} \tilde{h}(kT_s)\, e^{-kT_s\mathrm{L}_n}\, \mathrm{f} \;\approx\; \sum_{k=0}^{K_t-1} \tilde{h}(kT_s)\, (\mathrm{I} - T_s\mathrm{L}_n)^k\, \mathrm{f}$$

- Recover standard convolutions if we make the particular choice $\mathcal{L} = \mathrm{d}/\mathrm{d}x$

$$g(x) \;=\; \int_0^\infty \tilde{h}(t)\, e^{-t\mathrm{d}/\mathrm{d}x} f(x)\, \mathrm{d}t \;=\; \int_0^\infty \tilde{h}(t)\, f(x-t)\, \mathrm{d}t$$

- Manifold convolutions generalize standard (time) and graph convolutions

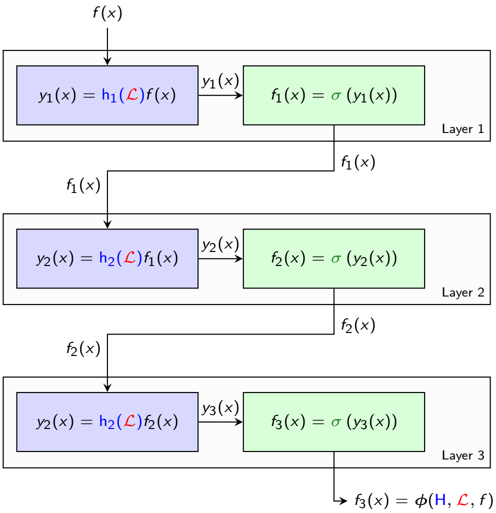
- LB operator admits discrete spectral decomposition $\Rightarrow \mathcal{L}f = \sum_{i=1}^{\infty} \lambda_i \langle f, \phi_i \rangle \phi_i$
- Manifold Fourier Transform of f is the set of projections $\Rightarrow [f]_i = \langle f, \phi_i \rangle$
- Frequency response of filter h is $\Rightarrow \hat{h}(\lambda) = \int_0^{\infty} \tilde{h}(t) e^{-t\lambda} \mathrm{d}t$

Theorem (Manifold Filters in the Manifold Spectral Domain)

Manifold filters are pointwise in the spectral domain $\Rightarrow [g]_i = h(\lambda_i)[f]_i$

- Manifold filters are easy to study in the manifold frequency (spectral) domain

- A MNN is a cascade of L layers
- Each of the layers is composed of
  - $\Rightarrow$ Manifold convolutions $\mathbf{h}(\mathcal{L})$
  - $\Rightarrow$ Pointwise nonlinearities σ
- Group learnable coefficients in $\mathcal{H}$
- Write MNN as map $y = \Phi(\mathcal{H}, \mathcal{L}, f)$



- Geometric graph filters and GNNs converge to their manifold counterparts

  $\Rightarrow$ Enables transferability of geometric GNNs from small to large graphs

- Sample the manifold at $\{x_i\}_{i=1}^n$. Construct graph Laplacian of G_n with edges

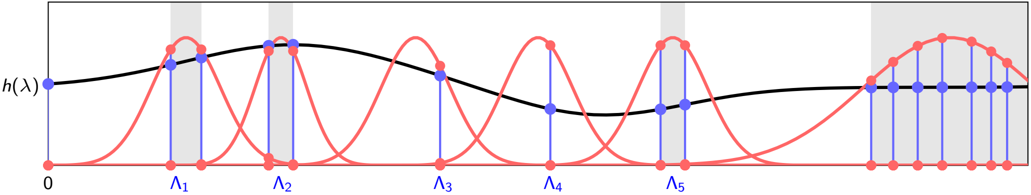
$$w_{ij} = K_\xi\left(\frac{\|x_i - x_j\|^2}{\xi}\right)$$

- Geometric graph filter is defined by replacing with graph Laplacians L_n

$$\mathrm{g} = \int_0^\infty \tilde{h}(t) e^{-t\mathrm{L}_n} \mathrm{d}t\mathrm{f} = \mathrm{h}(\mathrm{L}_n)\mathrm{f}, \qquad [\mathrm{f}]_i = f(x_i)$$

- Geometric graph neural networks on G_n $\Rightarrow \Phi(\mathrm{H}, \mathrm{L}_n, \mathrm{f})$

- A filter is A_h-Lipschitz if its frequency response $\hat{h}(\lambda)$ is A_h-Lipschitz
- Partition spectrum such that λ_i and λ_j are in different partitions if $|\lambda_i - \lambda_j| \geq \alpha$
- A filter is α-FDT if $|\hat{h}(\lambda_i) - \hat{h}(\lambda_j)| \leq \delta_D$ for all λ_i, λ_j in the same partition



- Does not discriminate frequency components associated to close eigenvalues

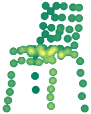
Theorem (Convergence of Geometric GNNs)

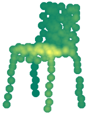
If an L-layer MNN $\Phi(\mathrm{H}, \mathcal{L}, \cdot)$ on $\mathcal{M}$ and GNN $\Phi(\mathrm{H}, \mathrm{L}_n, \cdot)$ on G_n have normalized Lipschitz nonlinearities, it holds in high probability that

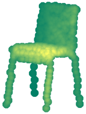
$$\left\| \Phi(\mathrm{H}, \mathrm{L}_n^{\epsilon}, \mathrm{P}_n f) - \mathrm{P}_n \Phi(\mathrm{H}, \mathcal{L}, f) \right\|_{L^2(\mathrm{G}_n)} \leq O\left[\left(\frac{N}{\alpha} + A_h\right)\sqrt{\xi}\right] + O\left(\frac{\log(n)}{n}\right)$$

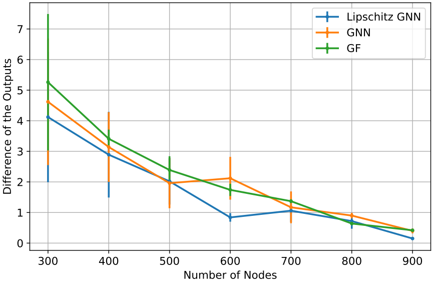
with filters that are α-FDT with $\delta_D \leq O(\sqrt{\xi}/\alpha)$ and A_h-Lipschitz continuous.

- The properties of large GNNs can be analyzed via MNN as their limit
- The error bounds show trade-off between discriminability and approximation









	Graph Filters	GNN	Lipschitz GNN
$n = 300$	$21.15\% \pm 3.48\%$	$9.35\% \pm 2.46\%$	$7.63\% \pm 3.36\%$
$n = 500$	$18.09\% \pm 6.28\%$	$7.80\% \pm 3.50\%$	$7.54\% \pm 4.01\%$
$n = 700$	$17.31\% \pm 6.59\%$	$8.16\% \pm 2.95\%$	$7.97\% \pm 2.45\%$
$n = 900$	$15.58\% \pm 4.54\%$	$7.20\% \pm 3.77\%$	$6.68\% \pm 3.94\%$

Wang-Ruiz-Ribeiro, *Geometric Graph Filters and Neural Networks: Limit Properties and Discriminability Trade-offs.* arxiv.org/abs/2305.18467,

Manifold Deformations as Operator Perturbations

- Stability to deformations is a distinguishable property of CNNs $\Rightarrow$ generalizable to GNNs and CNNs
- Consider manifold signal f and a deformation $\tau(x)$ over the manifold

$$p(x) = \mathcal{L}' f(x) = \mathcal{L} g(x) = \mathcal{L} f(\tau(x))$$

Theorem (Manifold deformations)

Let the deformation $\tau(x) : \mathcal{M} \to \mathcal{M}$ satisfy $\text{dist}(x, \tau(x)) = \epsilon$ and $J(\tau_*) = I + \Delta$ with $\|\Delta\|_F = \epsilon$.

If the gradient field is smooth, it holds that

$$\mathcal{L} - \mathcal{L}' = \text{E}\mathcal{L} + \mathcal{A},$$

where E and $\mathcal{A}$ satisfy $\|\text{E}\| = O(\epsilon)$ and $\|\mathcal{A}\|_{op} = O(\epsilon)$.

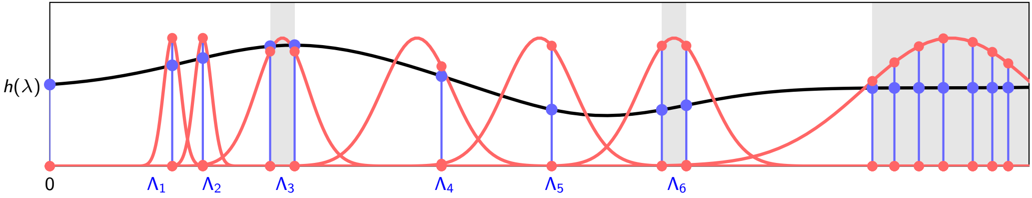
- Translate manifold signal perturbations as LB operator perturbations

- A filter is B_h-Integral Lipschitz if its frequency response satisfies

$$|\hat{h}(a) - \hat{h}(b)| \leq \frac{B_h|a-b|}{(a+b)/2}, \quad \text{for all } a, b \in (0, \infty)$$

- Partition spectrum such that λ_i and λ_j are in different partitions if $\left|\frac{\lambda_i}{\lambda_j} - 1\right| \geq \gamma$

- A filter is γ-FRT if $|\hat{h}(\lambda_i) - \hat{h}(\lambda_j)| \leq \delta_R$ for all λ_i, λ_j in the same partition



- Discriminate frequency components that are relatively far from each other

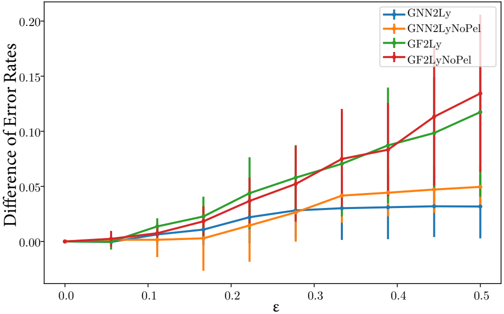
Theorem (Stability of MNNs to deformations)

An L-layer MNN $\Phi(\mathbf{H}, \mathcal{L}, f)$ have normalized Lipschitz continuous nonlinearities. Let $\mathcal{L}'$ be the deformed LB operator with $\max\{\alpha, 2, |\gamma/1-\gamma|\} \gg \epsilon$, then

$$\left\| \Phi(\mathbf{H}, \mathcal{L}, f) - \Phi(\mathbf{H}, \mathcal{L}', f) \right\|_{L^2(\mathcal{M})} \leq O\left[\left(\frac{N}{\alpha} + A_h + \frac{M}{\gamma} + B_h\right)\epsilon\right] \|f\|_{L^2(\mathcal{M})}$$

if the manifold filters are α-FDT with $\delta_D \leq O(\epsilon/\alpha)$, γ-FRT with $\delta_R \leq O(\epsilon/\gamma)$, A_h-Lipschitz continuous and B_h-integral Lipschitz continuous.

- The difference bound shows a trade-off between stability and discriminability

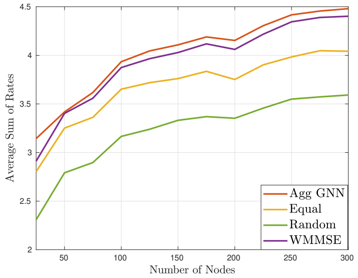


Architecture	$\epsilon = 0.2$	0.4
GNN2Ly	7.37% $\pm$ 1.43%	7.71% $\pm$ 3.96%
GF2Ly	13.76% $\pm$ 6.82%	13.54% $\pm$ 7.16%
Architecture	$\epsilon = 0.6$	0.8
GNN2Ly	8.04% $\pm$ 2.83%	11.01% $\pm$ 6.33%
GF2Ly	14.76% $\pm$ 5.67%	16.04% $\pm$ 6.34%

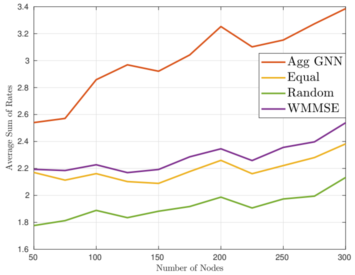
Wang-Ruiz-Ribeiro, *Stability to Deformations of Manifold Filters and Manifold Neural Networks.* arxiv.org/abs/2106.03725,

- We test the trained GNN in other networks of increasing size and fixed density
  - ⇒ The GNN transfers to larger ad-hoc networks with no need of retraining

Ad-hoc network with 25 pairs



Ad-hoc network with 50 pairs



Wang-Eisen-Ribeiro, *Learning decentralized wireless resource allocations with graph neural networks. TSP 2022.* arxiv.org/abs/2107.01489,