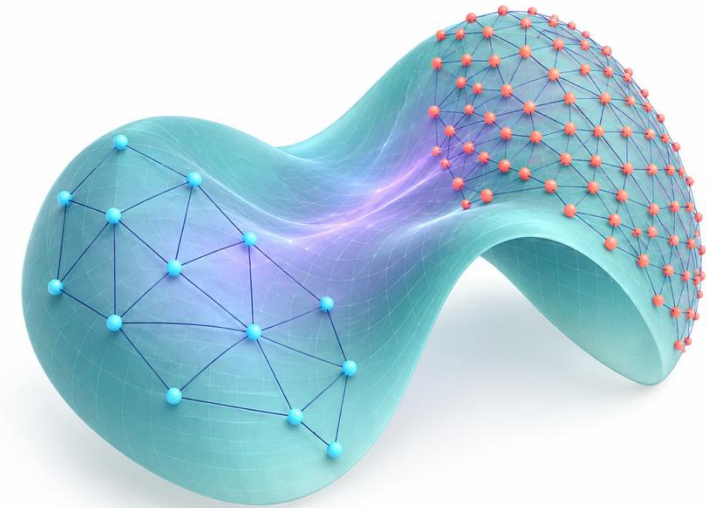


Size transferability of GNNs

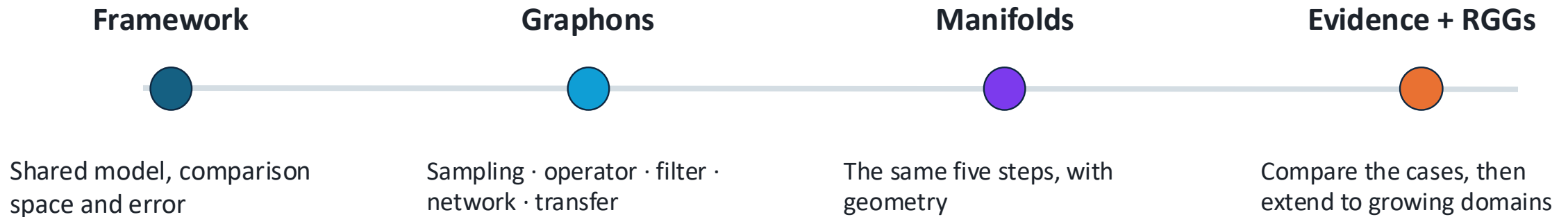
A common framework, two parallel limit models,
and a sparse geometric extension

- 01 Define a common transfer framework
- 02 Instantiate it for graphons and manifolds
- 03 Extend the argument to sparse RGGs



shared model and parameters, different graph sizes

One framework, two parallel case studies



Focus on data and model, without explicitly consider task functions.

A common model and controlled approximation support size transfer.

General framework: finite models approximate one operator

$$y_n = \Phi(S_n, x_n; H), \quad Y = \Phi(T, X; H)$$

$$e_n = \|I_n y_n - Y\|_2 \leq C_\Phi \delta_n \rightarrow 0$$

Shared ingredients

A generating model and construction rule

A limit operator T

Fixed learned parameters H

Comparison and assumptions

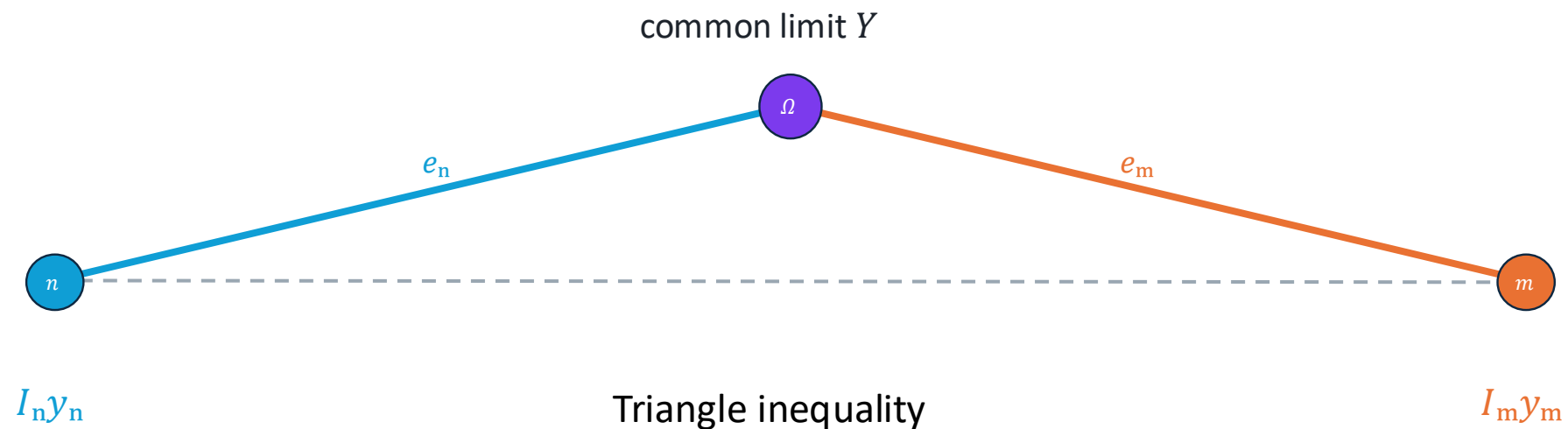
I_n lifts outputs to a common space.

δ_n records discretization errors.

C_Φ reflects network sensitivity.

This is a conditional proof template. Each setting must justify its approximation bound.

General framework: compare through the common limit



$$\|I_n y_n - I_m y_m\|_2 \leq e_n + e_m$$

If both lifted outputs approximate the same limit, they approximate each other.

The same five steps define both case studies

Framework step	Graphon	Manifold
1 Sampling	Sample one connectivity kernel W	Sample one geometric domain $\mathcal{M}$
2 Operator	Normalized adjacency $\rightarrow$ integral operator	Scaled Laplacian $\rightarrow$ differential $\mathcal{L}$
3 Filter	Fixed polynomial of bounded operator	Diffusion or controlled spectral filter
4 Network	Compose filters and activations	Compose filters and activations
5 Transfer	Lift to $L^2([0,1])$, then compare	Lift to $L^2(\mathcal{M})$, then compare

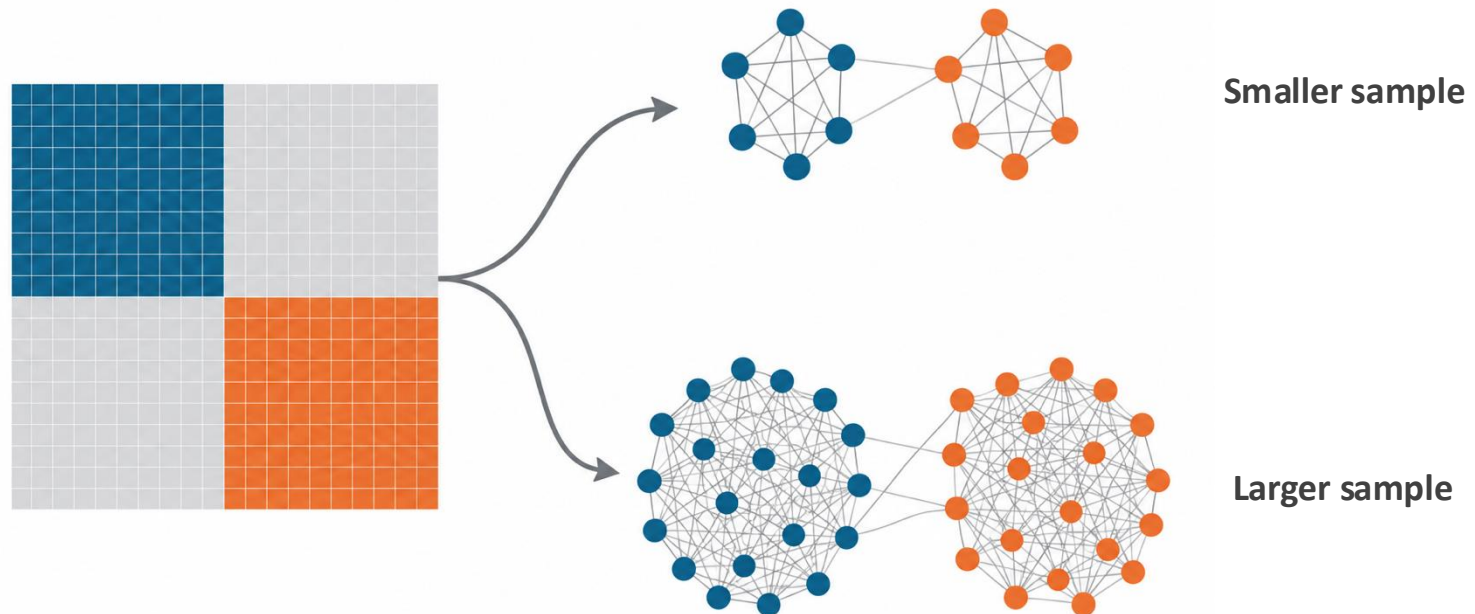
The proof structure is shared. The operator and approximation assumptions differ.

Graphon Step 1: Sampling a shared connectivity rule

ONE CONNECTION RULE

Sample latent positions.

Use W to draw edges independently.



Use SBM as an example --> shared graphon function

Keep the connection probabilities fixed. Increase the number of sampled nodes.

Graphon Step 2: Operator and spectral convergence

OPERATOR CONVERGENCE

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

$$S_n = \frac{A_n}{n}, \quad \|T_{W_n} - T_W\|_{op} \rightarrow 0$$

Lift S_n to a step-kernel operator and align the latent coordinates.

Divide adjacency by n to approximate an integral.

SPECTRAL CONVERGENCE

$$\lambda_k(S_n) = \frac{\lambda_k(A_n)}{n} \rightarrow \lambda_k(T_W)$$

$$\min_{s=\pm 1} \|s I_n v_{k,n} - \varphi_k\|_2 \rightarrow 0$$

Match nonzero eigenvalues with multiplicity.

For a simple isolated eigenvalue, align the eigenfunction sign.

Later filter analysis needs eigenvalue convergence and control of the corresponding spectral subspaces.

Graphon Step 3: Transferring the same filter across sizes

ONE FILTER: KEEP THE COEFFICIENTS FIXED

$$H(S) = \sum_{k=0}^K h_k S^k$$

$$H(T_W) = \sum_{k=0}^K h_k T_W^k$$

Represent the graph and input on $[0,1]$:

$$W_n(u, v) = (A_n)_{ij}, \quad X_n = I_n x_n$$

THE INDUCED FILTER EXACTLY REPRESENTS THE GRAPH FILTER

$$Y_n = I_n H(S_n) x_n = H(T_{W_n}) X_n$$

FILTER OUTPUTS CONVERGE TO THE SAME LIMIT

$$\|H(T_{W_n}) X_n - H(T_W) X\|_2 \rightarrow 0$$

Assume aligned operator-norm convergence, input convergence in L^2 , and fixed filter degree.

The graph size changes. The filter coefficients and limiting computation stay the same.

Graphon Step 4: Composing the network

$$X_q^\ell = \sigma \left(\sum_{p=1}^{F_{\ell-1}} H_{qp}^\ell(T_W) X_p^{\ell-1} \right)$$

Same architecture at every size

Share channel counts, filter coefficients and pointwise activations.

Replace the integral operator by its discrete approximation.

Controlled layer composition

Bound filter gains and activation Lipschitz constants.

Track how filter smoothness, depth and width amplify approximation errors.

Parameter sharing makes reuse possible. Approximation control makes it reliable.

Graphon Step 5: Comparing induced-network outputs

INDUCE THE GRAPHON AND INPUT BEFORE COMPUTATION

$$Y_n = \Phi(T_{W_n}, X_n; H), \quad Y_m = \Phi(T_{W_m}, X_m; H)$$

EXACT IDENTITY WITH THE FINITE GRAPH OUTPUT

$$Y_n = \Phi(T_{W_n}, I_n x_n; H) = I_n \Phi(S_n, x_n; H) = I_n y_n$$

COMPARE BOTH INDUCED OUTPUTS WITH THE SAME LIMIT

$$\|Y_n - Y_m\|_2 \leq \|Y_n - Y\|_2 + \|Y_m - Y\|_2 = e_n + e_m$$

$$Y = \Phi(T_W, X; H)$$

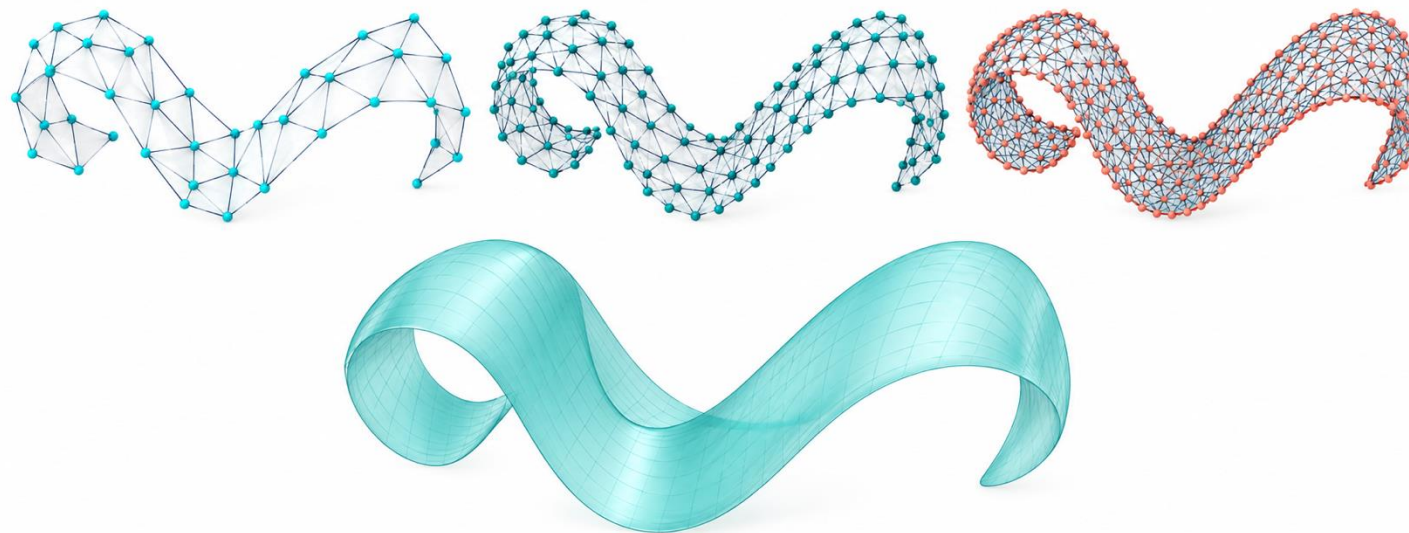
Consistent latent alignment and shared parameters.

$I_n y_n$ is the exact step-function representation of the finite output.

Manifold Step 1: Sampling a shared geometric domain

Samples lie on one smooth manifold.

Edges reflect local distance, and neighborhoods shrink with resolution.



$$w_{ij} = K\left(\frac{\|x_i - x_j\|^2}{\varepsilon_n^2}\right), \quad \varepsilon_n \rightarrow 0$$

Neighborhoods must still contain enough samples for the chosen convergence result.

Manifold Step 2: Operator and spectral convergence

OPERATOR CONVERGENCE

$$\mathcal{L} = -\text{div}_{\mathcal{M}} \nabla_{\mathcal{M}}$$

$$L_n P_n f \approx P_n \mathcal{L} f$$

Scale the graph Laplacian and correct for sampling density when required.

Keep the geometry and boundary conditions fixed.

SPECTRAL CONVERGENCE

$$\lambda_k(L_n) \rightarrow \lambda_k(\mathcal{L})$$

$$\|S I_n v_{k,n} - \varphi_k\|_2 \rightarrow 0$$

For each fixed k , under spectral-convergence assumptions. Align signs for simple eigenvalues.

Assumptions required for spectrum convergence.

Manifold Step 3: Transferring the filter

USE A BOUNDED RESPONSE OF THE GEOMETRIC LAPLACIAN

$$H(\mathcal{L}) = e^{-t\mathcal{L}}, \quad H(L_n) = e^{-tL_n}, \quad t > 0$$

$$\|H(L_n)P_nf - P_nH(\mathcal{L})f\|_2 \rightarrow 0$$

Same filtering role

Fix the diffusion time t across sizes. Nearby samples approximate the same smoothing process.

Different operator assumptions

$\mathcal{L}$ is unbounded. Prove filter convergence using geometry and sampling assumptions.

The Laplacian is unbounded, but diffusion filters are bounded.

Manifold Step 4: Composing the network

$$f_q^\ell = \sigma \left(\sum_{p=1}^{F_{\ell-1}} H_{qp}^\ell(\mathcal{L}) f_p^{\ell-1} \right)$$

Same architecture at every size

Share channels, filter responses and pointwise activations.

Replace $\mathcal{L}$ by its graph approximation.

Controlled layer composition

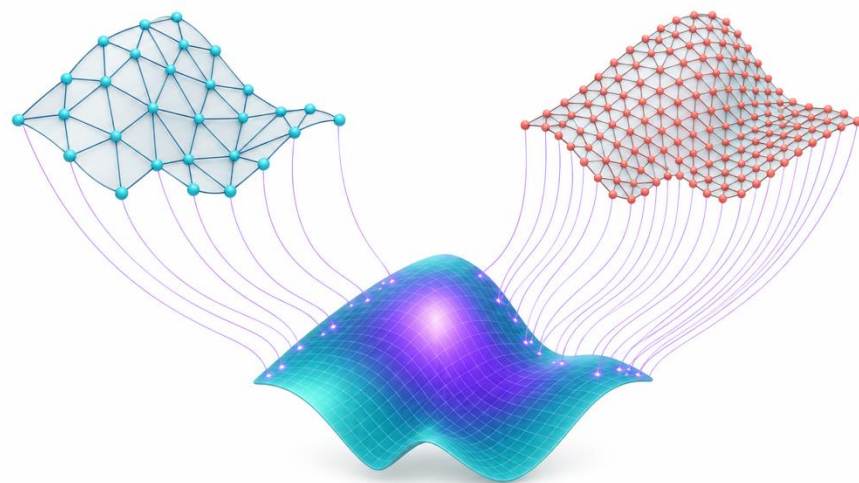
Bound gains and activation sensitivity.

Require approximation for the signals produced at each layer.

The network argument matches the graphon case once the geometric filter errors are controlled.

Manifold Step 5: Comparing outputs in a common space

INTERPOLATE NODE OUTPUTS ONTO $\mathcal{M}$



$$Y_n = I_n \Phi(L_n, P_n f; H)$$

$$\|Y_n - Y_m\|_2 \leq e_n + e_m$$

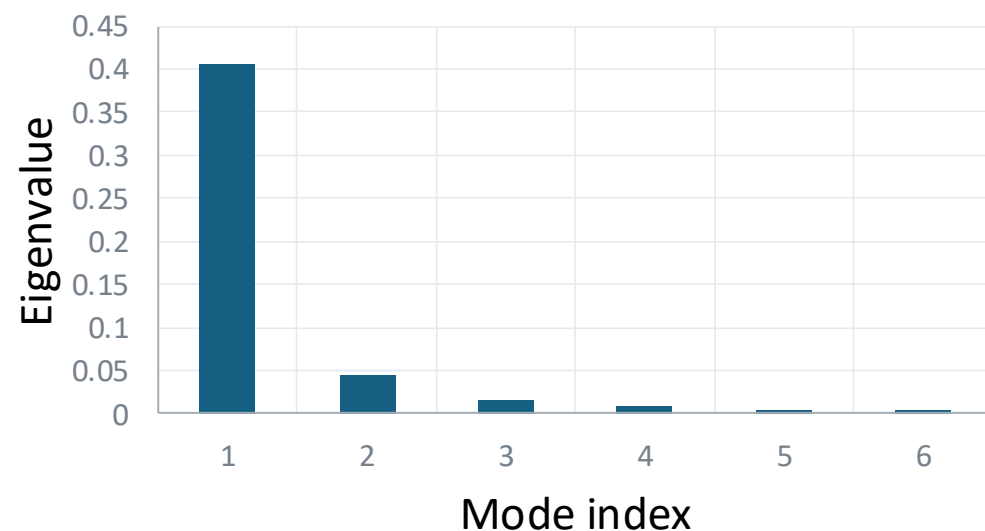
Use consistent interpolation and volume normalization.

Control geometric error by designing filter functions.

With geometry, signal and network approximation controlled, the same weights transfer.

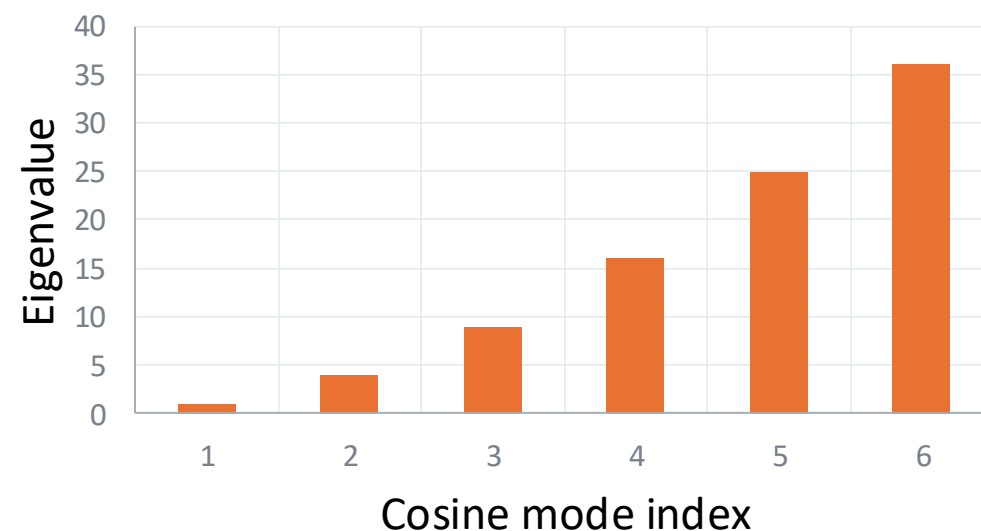
Graphon and manifold spectra have different limits

GRAPHON: KERNEL $W(u, v) = \min(u, v)$



Nonzero eigenvalues accumulate at zero.

MANIFOLD: UNIT CIRCLE COSINE MODES

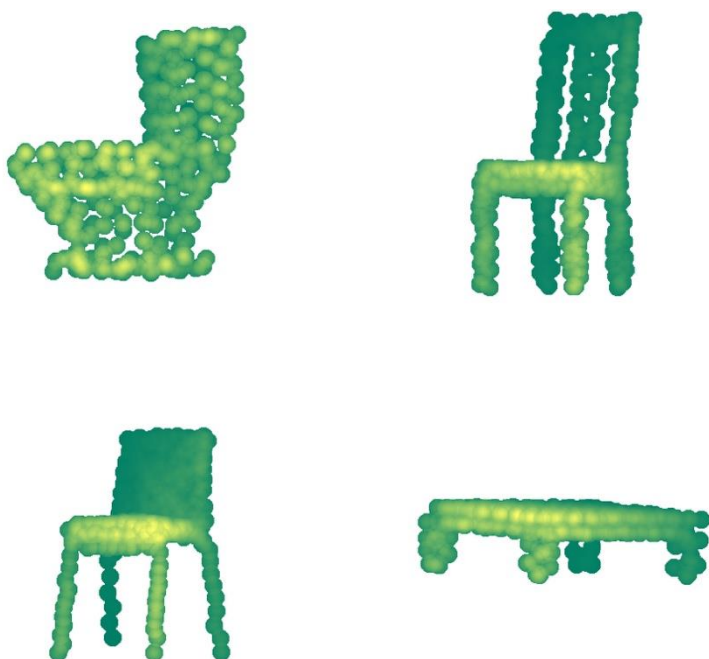


Laplacian eigenvalues grow without bound.

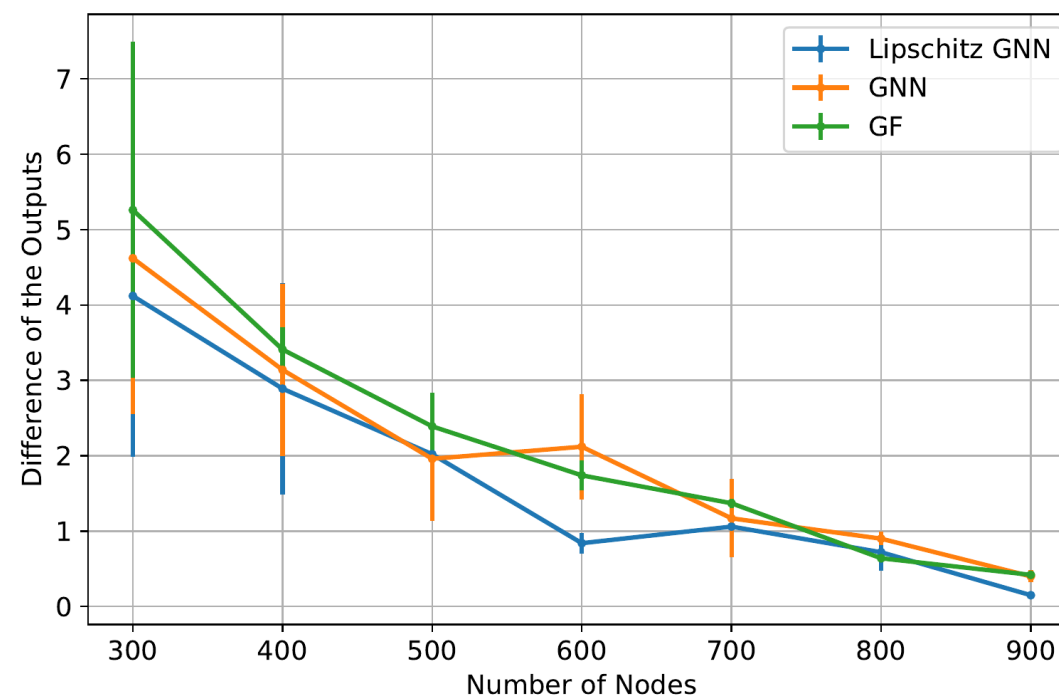
Near zero has different meanings for these two operators.

CAD models: compare small and large point clouds

ModelNet10 CAD point clouds



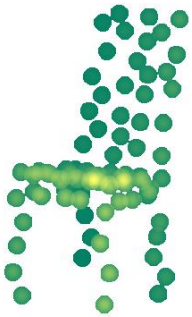
Output difference to a 1,000-point reference



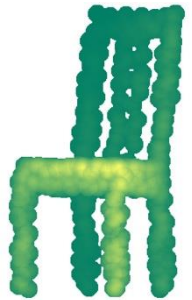
The reported output differences decrease as the training resolution increases.

CAD classification after transfer to 1,000 points

Coarse input



Denser input



Classification error (%) with reported variability; all models tested at $n = 1,000$.

Train n	Baseline GNN	Graph filter	GNN	Lipschitz GNN
300	16.95 ± 5.42	21.97 ± 4.17	10.10 ± 1.40	8.60 ± 2.95
500	13.11 ± 4.97	19.83 ± 5.94	7.74 ± 4.05	7.68 ± 3.75
700	10.02 ± 3.87	16.62 ± 2.38	7.92 ± 3.14	8.02 ± 2.77
900	6.83 ± 3.96	13.85 ± 3.81	7.45 ± 4.03	7.44 ± 3.30

At training $n = 300$, the Lipschitz GNN has the lowest reported error in this table.

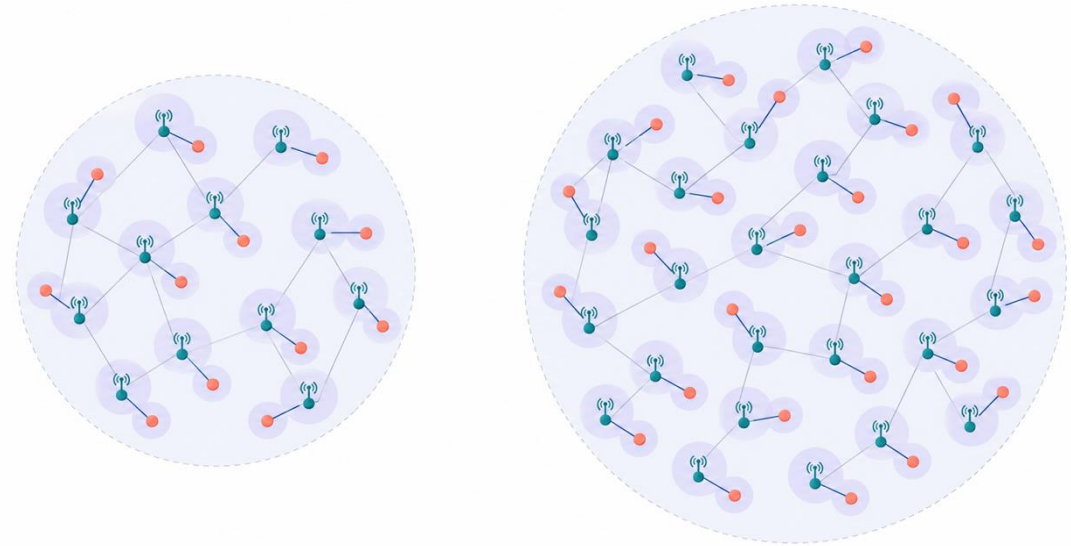
Sparse geometric growth is a third size setting

Larger physical region

Keep node density and connection radius fixed.

Add nodes by increasing area.

Expected local degree stays roughly constant away from boundaries.

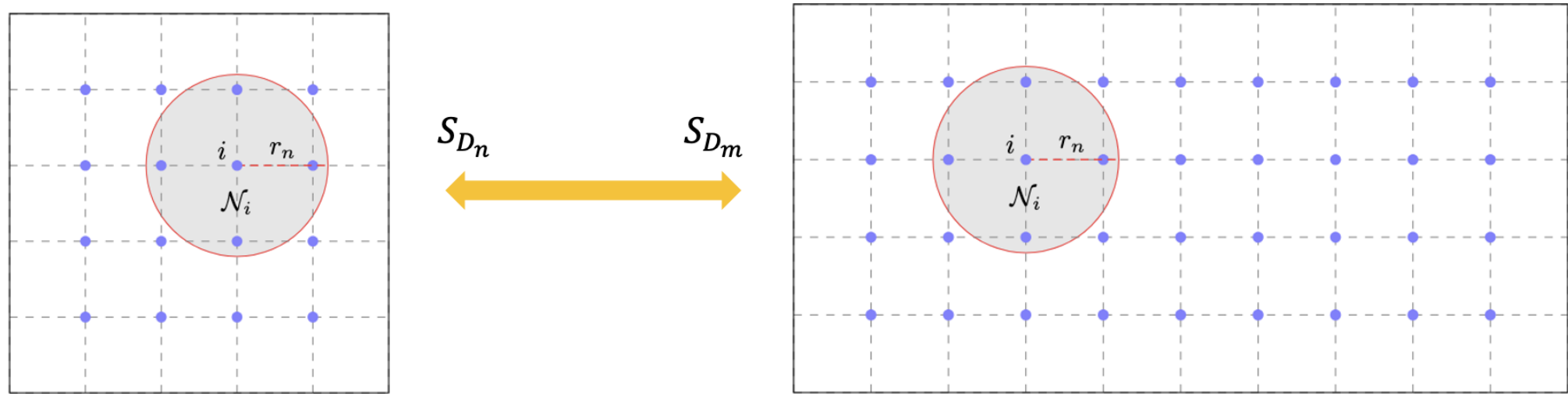


This is growth, rather than refinement of one fixed compact surface.

$$\tilde{\Delta}(S_m^1, S_m^2, H_m) := \frac{1}{m} \|\Phi(x_m, S_m^1; H_m) - \Phi(x_m, S_m^2; H_m)\|^2 \quad \Delta(S_m, S_n, H_m) := \frac{1}{m} \|\Pi_m [\Phi(x_n, S_n; H_m)] - \Phi(x_m, S_m; H_m)\|^2$$

Use a deterministic grid as an intermediate model for sparse random geometric graphs.

A grid GNN applies the same local stencil everywhere



Assumptions: jointly stationary, bounded input and output signals; a shared local policy.

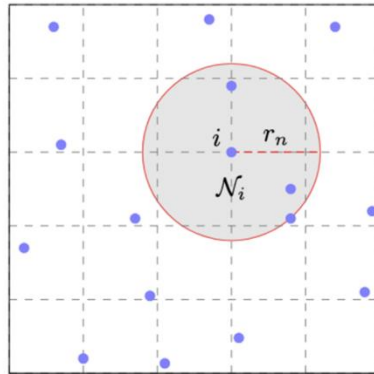
$$y_B = \sum_{k=0}^K h_k (L *)^k x_B$$

$$\Delta(G_m^D, G_n^D, H_m) \leq \frac{H_{L,K}^2}{m} (2\sqrt{m}LK + L^2K^2)x_{\max}^2,$$

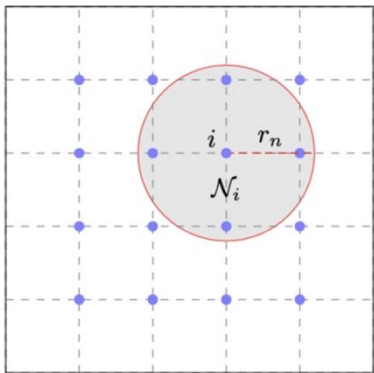
On a translation-invariant grid, graph filtering is a spatial convolution.

Stationarity connects local statistics across differently sized grid domains.

Compare a random graph with a matched grid



S_n
 S_{D_n}



Same size, aligned operators

$$\|R_m - G_m\| \leq \varepsilon$$

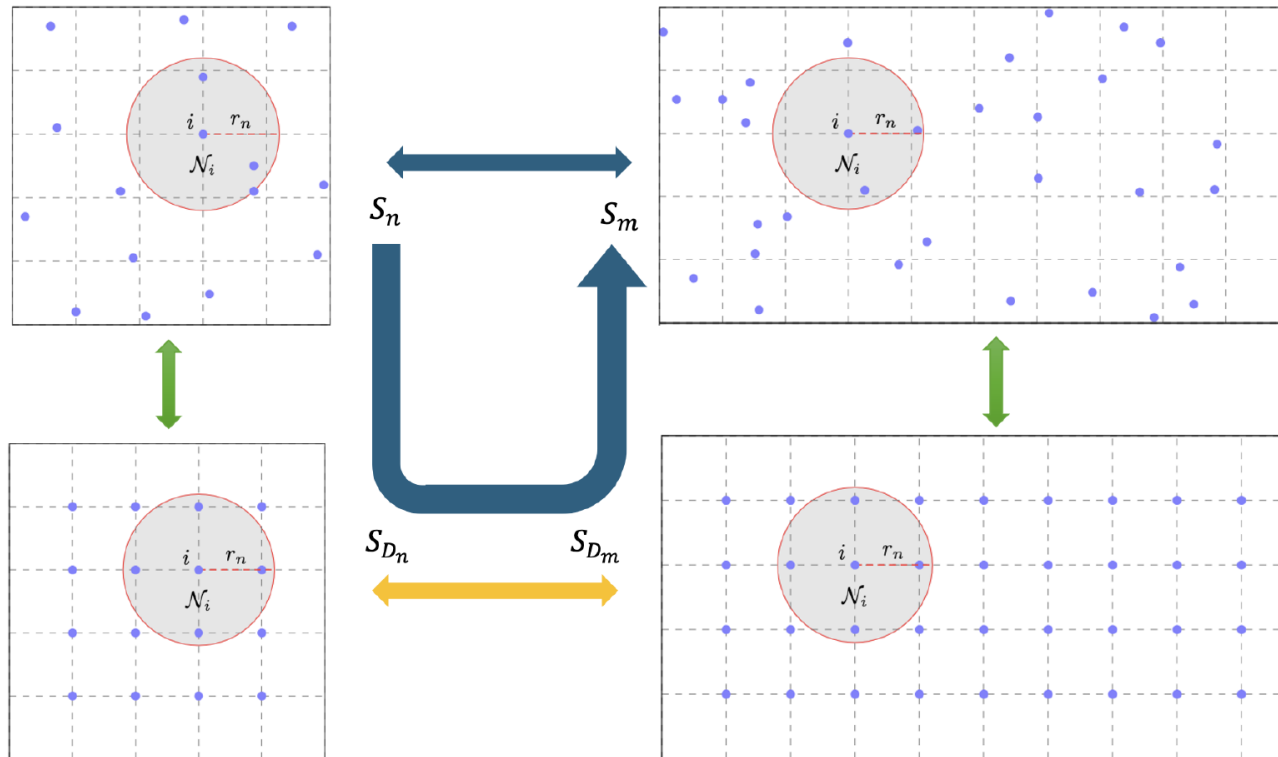
$$\tilde{\Delta}(R_m, G_m, H_m) \leq 4L^2 \varepsilon C x_{\max}^2$$

integral-Lipschitz filters.

The operator approximation is an assumption.

Fixed density alone does not establish the required random-to-grid approximation.

A grid bridge connects two random graph sizes



1 Random graph to grid

2 Small grid to large grid

3 Grid to random graph

$$\Delta(R_m, R_n, H_m) \leq 24L^2 \varepsilon C x_{\max}^2 + \frac{3H_{K,L}^2}{m} (2\sqrt{m}LK + L^2K^2)x_{\max}^2$$

Two graph-to-grid errors and one grid-size comparison control the total difference.

A controlled study states what stays fixed

Setting	Hold fixed	Check during evaluation
Graphon	Kernel, latent law and features	Alignment and adjacency normalization
Manifold	Geometry, metric and signal rule	Bandwidth and interpolation
Sparse RGG	Density, radius and local statistics	Boundary effects and grid approximation

Freeze learned parameters. Report transfer gap, task performance, sampling variation and computational cost.

Increasing n is meaningful only after the data-generating change has been specified.

Transferability and runtime answer different questions

Setting	How size changes	Typical edge scaling
Dense graphon	More samples of W	Quadratic in n
Manifold refinement	More points on a fixed domain	n times the neighborhood size
Fixed-density RGG	More area at fixed local density	Expected edges proportional to n

Parameter count can remain fixed in every setting. Runtime and memory still depend on graph construction and aggregation.

A transfer theorem does not by itself guarantee fast computation.

Synthesis: one proof structure, different approximations

Shared step	What each setting must supply
Common model	A stable kernel or geometric domain and compatible signals
Operator and filters	The correct normalization and a valid filter convergence result
Network composition	Fixed parameters and controlled amplification of errors
Output comparison	A common function space, alignment and consistent norms
Transfer conclusion	Two finite-to-limit errors bound the difference across sizes

The RGG extension keeps the comparison strategy but uses a grid bridge across growing domains.

TAKEAWAYS

One framework. Two parallel limit models.

- 01** **MODEL** Instantiate the same five steps for graphons and manifolds.
- 02** **SHARE** Reuse parameters with compatible operator normalization.
- 03** **CONTROL** Control finite-to-limit errors, then compare graph sizes.
- 04** **SCALE** Measure task performance and computation separately.

