Graph positional encodings: methods and properties

LECTURE 18 · CONTINUATION OF LECTURE 17

Absolute and relative representations of graph structure

Fixed constructions and learned encoding functions

Expressivity, perturbation stability, and transferability across graph sizes

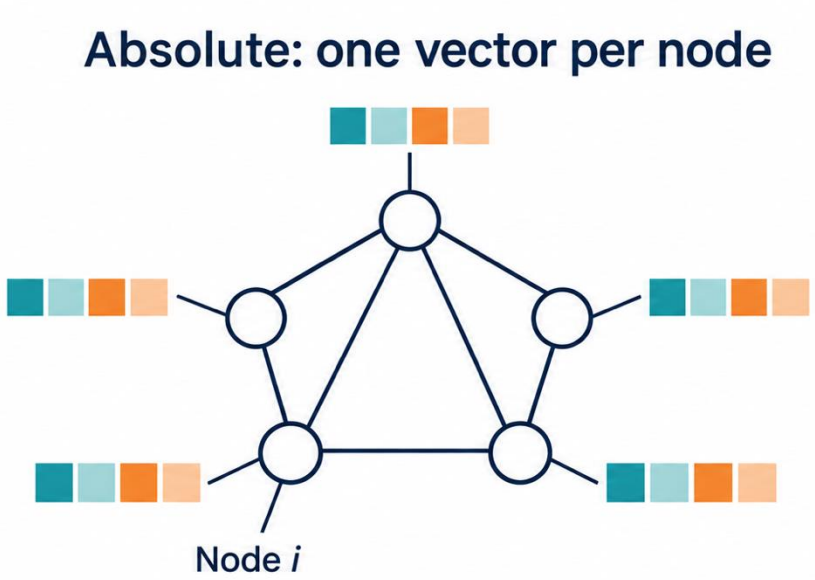
The encoding determines which structural information the model can use

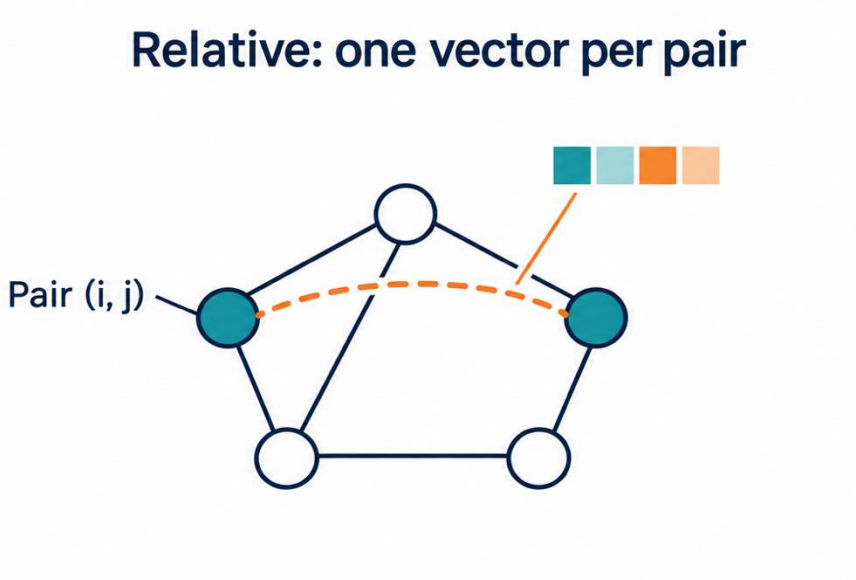
Rampášek et al. (2022). Recipe for a General, Powerful, Scalable Graph Transformer.

Two independent encoding choices

Absolute / relative describes the object encoded.

Fixed / learnable describes how it is computed.





A Laplacian coordinate is absolute.

A distance bias is relative.

Either kind can be processed by a learned function.

Rampášek et al. (2022). Recipe for a General, Powerful, Scalable Graph Transformer.

Absolute encoding: Laplacian coordinates for nodes

Undirected, nonnegative weights; assume positive degrees for the displayed normalization.

$$L_{sym} = I - D^{-1/2} A D^{-1/2}, \qquad L_{sym} u_k = \lambda_k u_k$$

$$p_i = (u_1(i), \ldots, u_K(i))$$

Select nontrivial modes and state the treatment of disconnected components.

K controls the spectral information retained and the preprocessing cost.

Low-frequency coordinates give global context without using node IDs

Dwivedi & Bresson (2020). A Generalization of Transformer Networks to Graphs.
Kreuzer et al. (2021). Rethinking Graph Transformers with Spectral Attention.

Absolute encoding: node-wise random-walk signatures

Random-Walk Structural Encoding (RWSE) assigns a return-probability vector to each node.

$$p_i = ((T^1)_{ii}, \ldots, (T^K)_{ii}), \qquad \text{with } T = D^{-1}A,$$

Node vector p_i: probabilities of returning to i after 1 through K random walk steps.

Nodes with the same return profile receive the same structural signature.

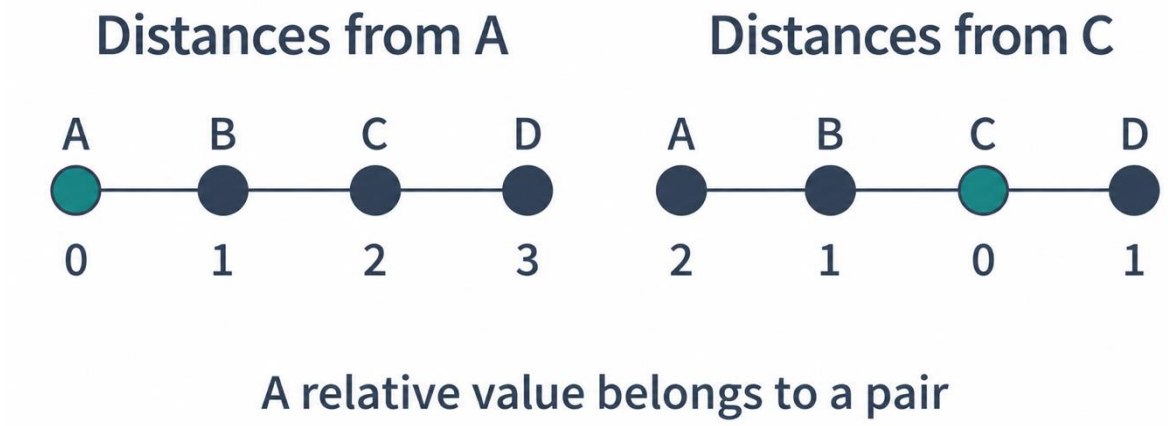
Full matrices T^k preserve destinations and lead to relative pair encodings.

Node-attached structural signatures need not be unique node addresses

Rampášek et al. (2022). Recipe for a General, Powerful, Scalable Graph Transformer.
Ma et al. (2023). Graph Inductive Biases in Transformers without Message Passing.

Relative encoding: shortest-path distance in attention

The pair is the object represented, and the distance is computed in the observed graph.



Choose a source node, then describe each target relative to it.

A–D has distance 3. C–D has distance 1.

A learned distance bias can modify attention.

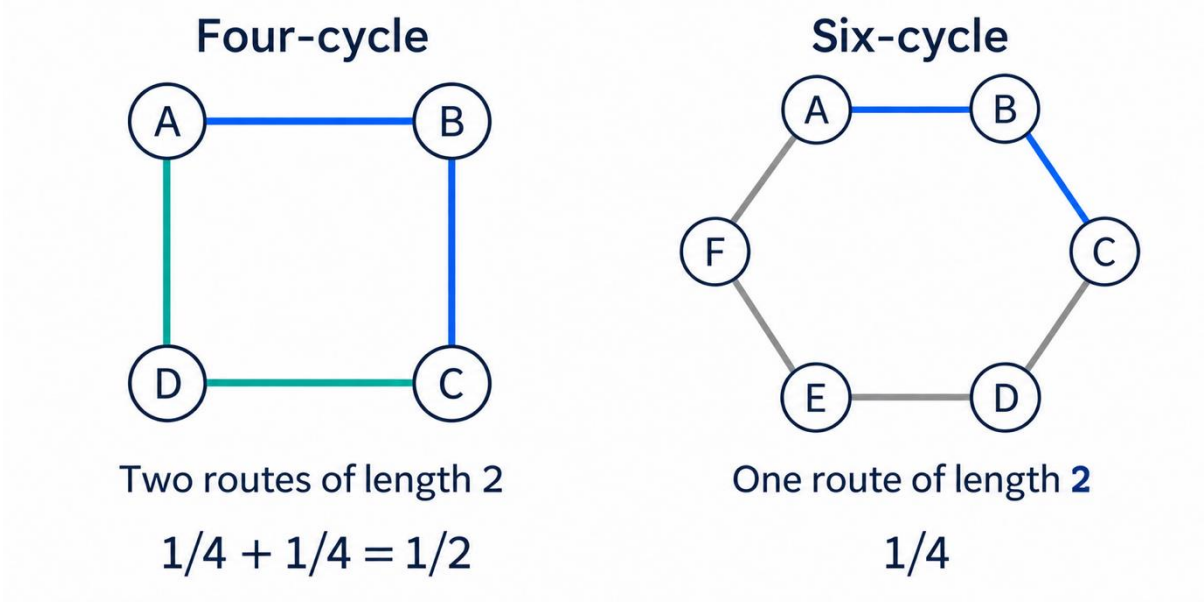
$$r_{ij} = \mathrm{dist}_G(i, j), \qquad b_{ij} = \mathrm{Embed}_\theta(r_{ij})$$

Equal shortest distances need not imply equivalent pair relationships

Ying et al. (2021). Do Transformers Really Perform Bad for Graph Representation?

Relative encoding: probabilities of walks between nodes

With positive degrees, $T = D^{-1}A$. Include step zero when an identity channel is useful.



Both highlighted pairs have shortest distance 2.

Each two-step route has probability 1/4.

The four-cycle has two such routes.

$$r_{ij} = ((T^0)_{ij}, (T^1)_{ij}, \ldots, (T^K)_{ij})$$

Walk profiles retain information that the shortest distance discards

Relative encoding: spectral kernels for node pairs

For a symmetric Laplacian, weight modes by a function of their eigenvalues.

$$K_g = U g(\Lambda) U^T = \sum_k g(\lambda_k) u_k u_k^T$$

Equal eigenvalues receive equal weights, so a complete eigenspace is basis invariant.

$K_g(i,j)$ is a relative descriptor; $K_g(i,i)$ gives a node-level descriptor.

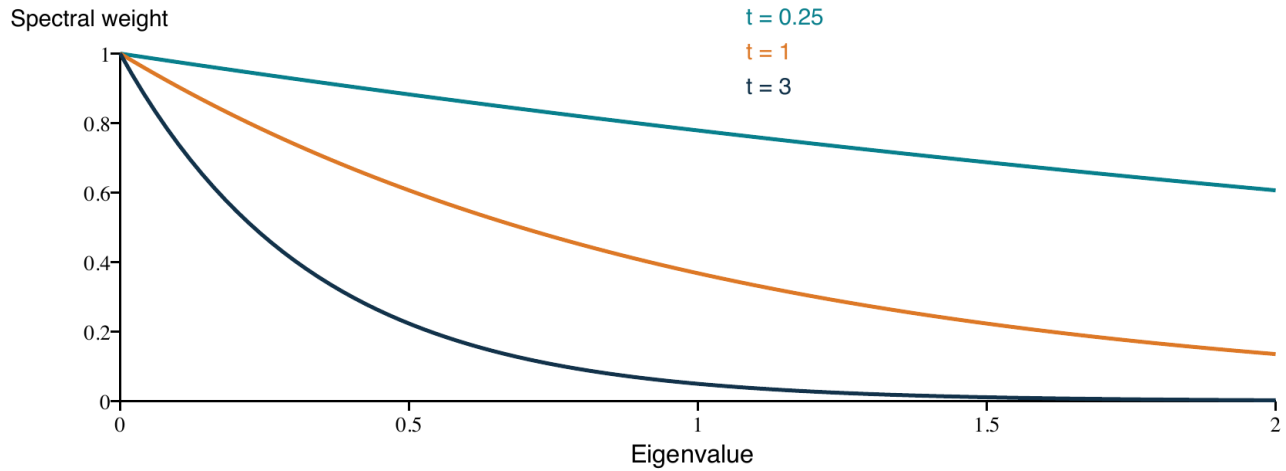
Smooth spectral weighting avoids selecting one arbitrary direction.

One spectral construction can supply both absolute and relative information

Huang et al. (2024). On the Stability of Expressive Positional Encodings for Graphs.

Relative encoding: diffusion time sets the spatial scale

For symmetric $L \succeq 0$, diffusion time t determines which spectral scales remain.



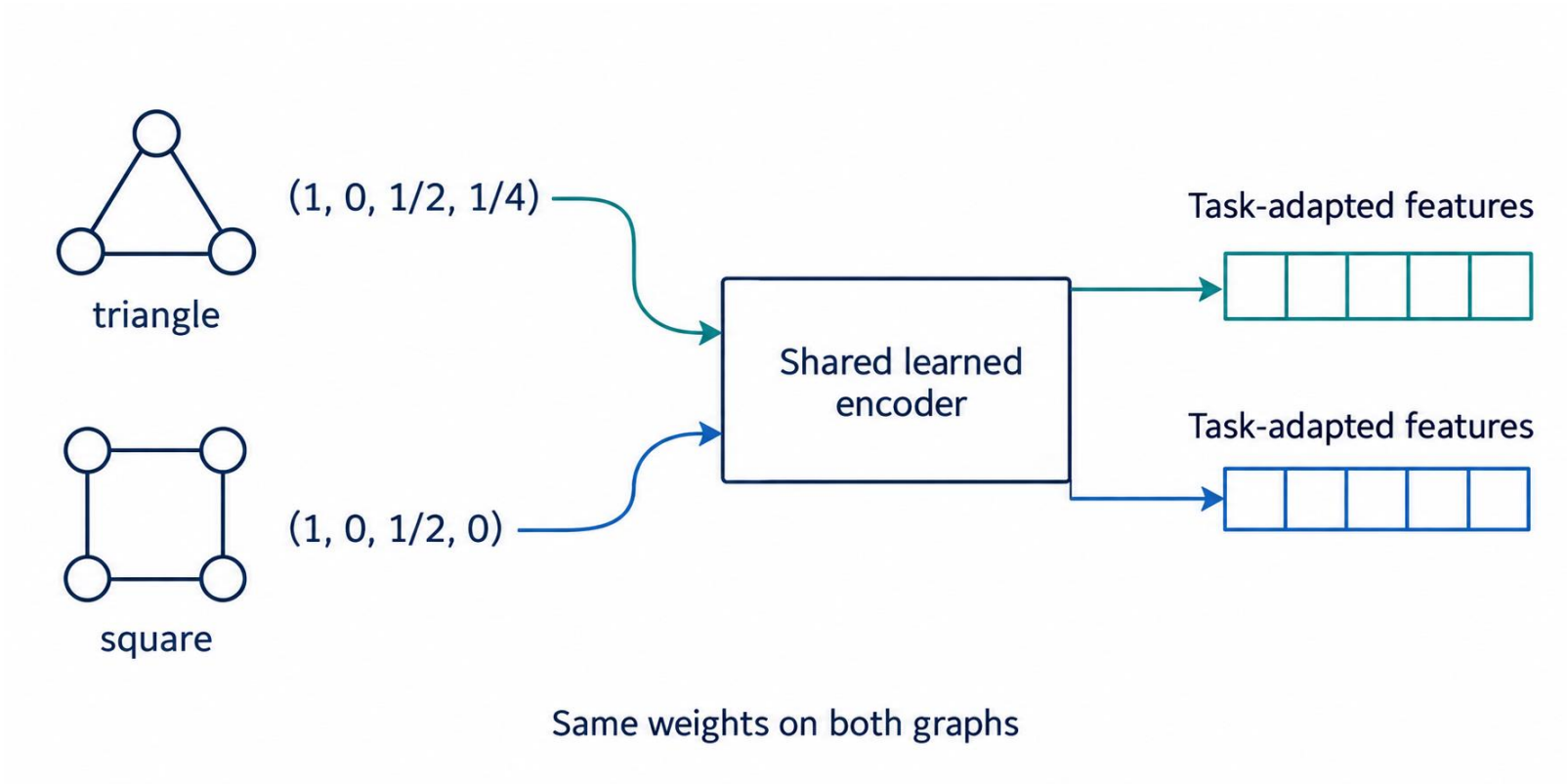
$$K_t = e^{-tL} = \sum_k e^{-t\lambda_k} u_k u_k^T$$

Small t preserves local detail. Larger t emphasizes slow, broad variations.
For connected combinatorial L, K_t tends to $\mathbf{1}\mathbf{1}^T/n$ as t grows.
K_t is not row-stochastic for every Laplacian convention.

Use several scales when one diffusion time loses useful distinctions

Learned encoding: shared maps of structural features

A shared neural encoder can learn a useful transformation of graph-derived statistics.



A graph-conditioned encoder differs from a table indexed by arbitrary node IDs

Dwivedi et al. (2022). Graph Neural Networks with Learnable Structural and Positional Representations.
Ma et al. (2023). Graph Inductive Biases in Transformers without Message Passing.

Learned encoding: LSPE updates a positional state

Conceptual update notation; the published modules specify the actual aggregation functions.

$$H^{\ell+1} = F_\theta^\ell(H^\ell, P^\ell, E, A)$$

$$P^{\ell+1} = G_\theta^\ell(H^\ell, P^\ell, E, A)$$

Initialize P from graph-derived information, then refine it with shared learned updates.

Keep feature and positional channels distinct so that their roles remain explicit.

Learning an encoding does not automatically remove its initialization ambiguities.

Learnable Structural and Positional Encoding (LSPE) adapts positional representations to the task

Dwivedi et al. (2022). Graph Neural Networks with Learnable Structural and Positional Representations.

Learned encoding: SAT summarizes rooted neighborhoods

A continuation of the short lecture: relate the extractor to the encoding viewpoint.

$$s_i = \varphi_\theta(i, G, X), \qquad q_i = W_Q s_i, \qquad k_i = W_K s_i$$

The extractor describes an attributed neighborhood, rather than a node ID.

Using a GNN to construct queries and keys can implement this SAT mechanism.

A standard message-passing extractor still inherits its own expressivity limits.

The important choice is what the encoder captures and where its output is used

Chen, O'Bray & Borgwardt (2022). Structure-Aware Transformer for Graph Representation Learning.

Learned encoding: GRIT updates node-pair features

In the Graph Inductive Bias Transformer (GRIT), Relative Random Walk Probabilities (RRWP) initialize pair states; the architecture then updates node and pair features.

$$R_{ij}^{0} = f_{\theta}(r_{ij})$$

The learned encoding is attached to ordered pairs, not only to individual nodes.

The full architecture also uses a specialized attention mechanism and degree information.

Its expressivity statements require the stated walk horizon and architectural conditions.

A learned pair encoder can use more information than a distance lookup

Ma et al. (2023). Graph Inductive Biases in Transformers without Message Passing.

Learned encoding: SignNet and BasisNet handle ambiguity

Illustrative sign symmetrization, followed by the broader basis-invariant idea.

$$\psi(u) = f_\theta(u) + f_\theta(-u)$$

$$P_\lambda = U_\lambda U_\lambda^T$$

The first construction is exactly sign invariant. Random sign augmentation is not a guarantee.

BasisNet processes eigenspace information without depending on its chosen basis.

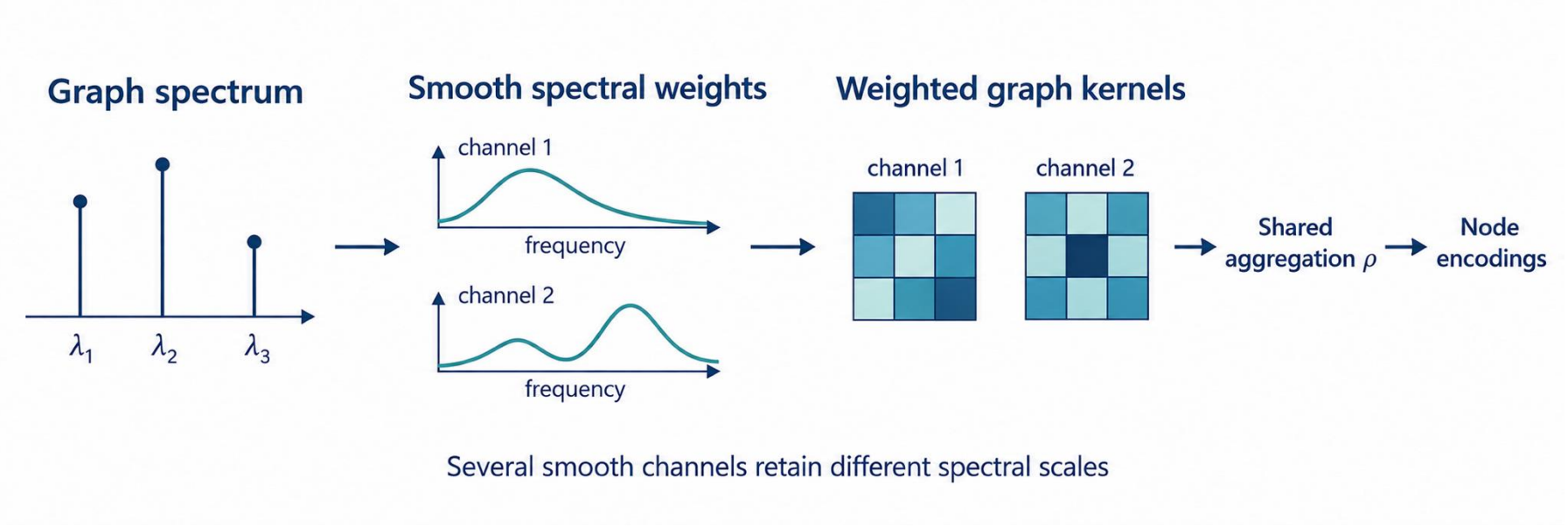
Coordinate invariance alone does not guarantee continuity when eigenvalues move.

SignNet and BasisNet address symmetries of the same eigendecomposition

Lim et al. (2023). Sign and Basis Invariant Networks for Spectral Graph Representation Learning.

Learned encoding: smooth spectral kernels in SPE

In Stable and Expressive Positional Encodings (SPE), learned spectral weights preserve spectral information while handling eigenvector symmetries.



$$\mathrm{SPE}(U,\lambda) = \rho(U\mathrm{diag}(\varphi_1(\lambda))U^T, \ldots, U\mathrm{diag}(\varphi_m(\lambda))U^T)$$

Smooth weighting and symmetry-aware aggregation work together

Huang et al. (2024). On the Stability of Expressive Positional Encodings for Graphs.

Symmetry: node relabeling requires equivariance

Use Π for node permutations.

$$p(\Pi A \Pi^T, \Pi X) = \Pi p(A, X)$$

$$R(\Pi A \Pi^T, \Pi X) = \Pi R(A, X) \Pi^T$$

Absolute encodings follow rows. Relative encodings follow both node indices.

Raw eigenvectors satisfy this only up to sign or basis changes.

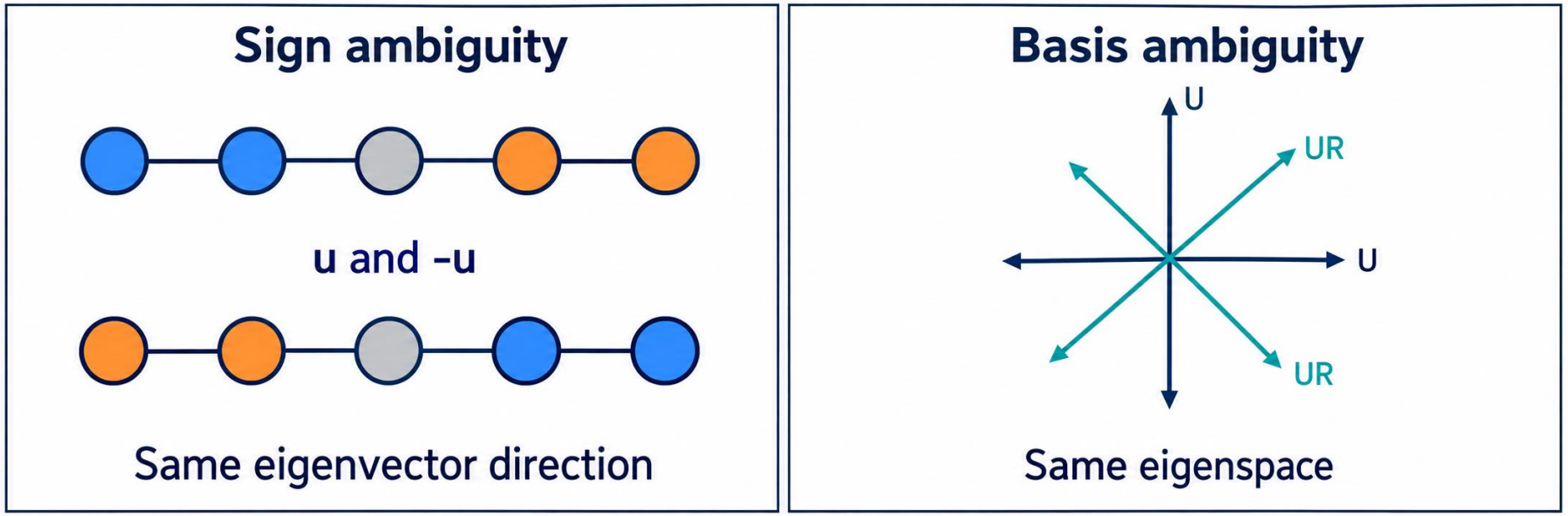
An arbitrary learned node-ID table has no canonical correspondence on a new graph.

Node names must not change the graph-level prediction

Wang et al. (2022). Equivariant and Stable Positional Encoding for More Powerful Graph Neural Networks.

Symmetry: eigenvector signs and bases are nonunique

Both transformations describe exactly the same operator.



Sign handling is sufficient only for isolated, simple eigenvalues.
Repeated eigenvalues require invariance to rotations of their entire eigenspace.

Symmetry: projectors remove eigenspace basis choices

U holds a full orthonormal basis of one eigenspace.
The graph fixes the eigenspace, the eigensolver may return U or any rotation UR.

$$LU = \lambda U, \qquad R^T R = I$$

$$(UR)(UR)^T = UU^T$$

Fix: UU^{T} is the same for every R. Its entry keeps the geometry and drops the sign.

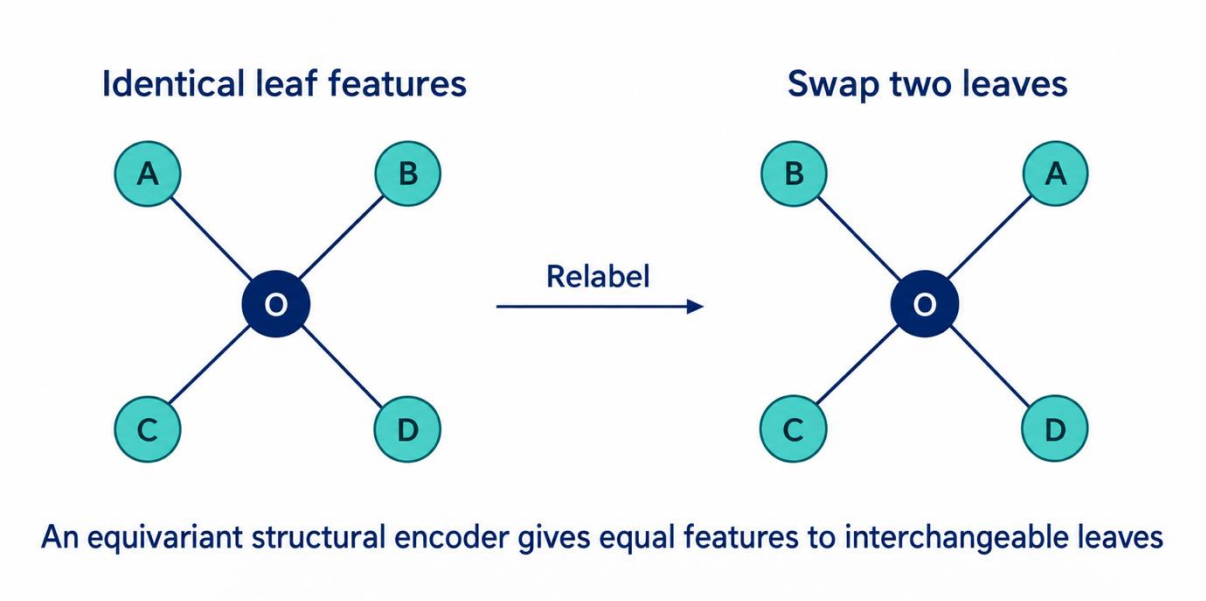
Use: $UU^{\mathrm{T}}(i,j)$ as a pairwise attention bias. Relabeling gives $\Pi UU^{\mathrm{T}}\Pi^{\mathrm{T}}$. Needs the whole eigenspace.

The projector is the part of the eigendecomposition the graph actually determines

Lim et al. (2023). Sign and Basis Invariant Networks for Spectral Graph Representation Learning.

Expressivity: retained information and graph symmetry

A star graph shows the limit imposed by symmetry. The full eigensystem illustrates retained information.



The four leaves play identical structural roles.

Swapping leaves preserves the attributed graph.

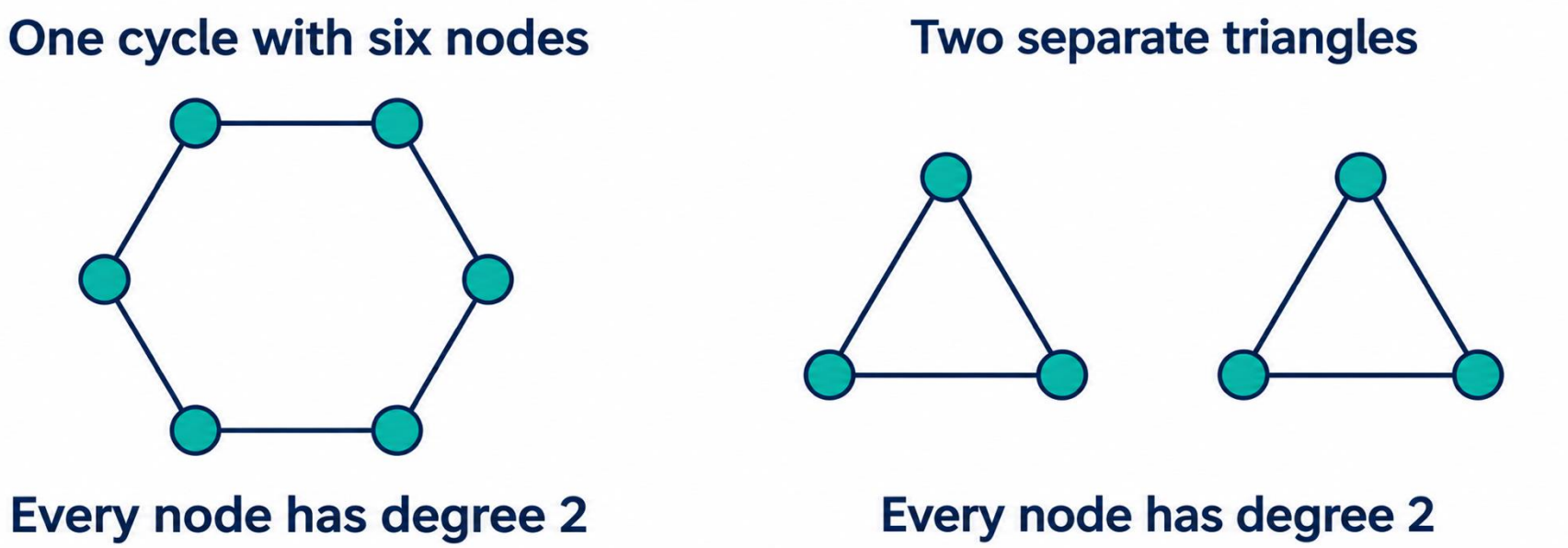
Extra coordinates or anchors supply extra information.

$$L = U\Lambda U^T$$

More expressive is a statement under assumptions, not an accuracy guarantee

Expressivity example: a cycle versus two triangles

Unweighted simple random walks: one six-cycle versus two disjoint triangles.



Every node has degree 2, and identical features defeat ordinary 1-WL message passing.
After three steps: return probability is 0 on the six-cycle and 1/4 on each triangle.

A structural encoding can separate graphs that a basic message-passing model cannot

Stability: small eigengaps can amplify perturbations

A different issue: the graph changes, rather than the solver choosing a different basis.

$$\|\sin\Theta(U, \tilde{U})\|_2 \lesssim \frac{\|L - \tilde{L}\|_2}{\delta}$$

Schematic subspace bound: δ separates the retained cluster from the rest.

Small δ permits large changes under a small operator perturbation.

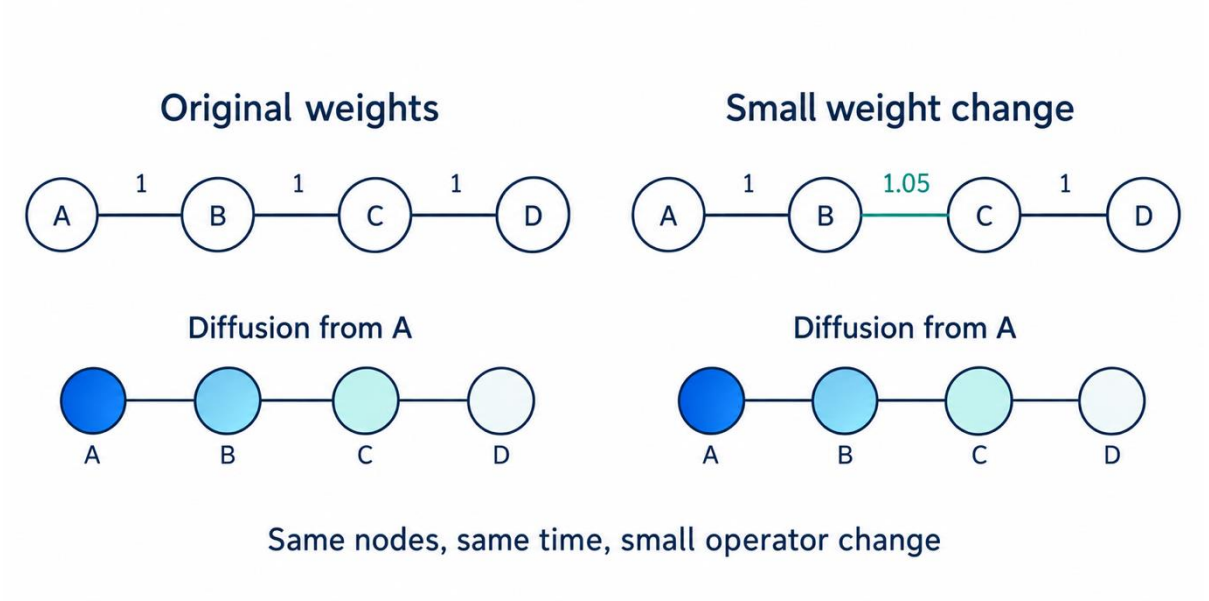
Sign alignment cannot repair a rotation that mixes nearby eigenspaces.

Invariance to coordinates does not by itself imply perturbation stability

Huang et al. (2024). On the Stability of Expressive Positional Encodings for Graphs.
Wang et al. (2022). Equivariant and Stable Positional Encoding for More Powerful Graph Neural Networks.

Stability: a norm bound for diffusion kernels

Same node set and alignment; symmetric positive-semidefinite L and $\tilde{L}$.



Change one edge weight from 1 to 1.05.

Compare diffusion on the same aligned nodes.

The colors illustrate similar heat profiles qualitatively.

$$\|e^{-tL} - e^{-t\tilde{L}}\|_2 \leq t\|L - \tilde{L}\|_2$$

A stable kernel can be analyzed without tracking individual eigenvectors

Stability: walk length and degree normalization matter

For row-stochastic T and $\tilde{T}$ on aligned nodes, use the induced infinity norm.

$$T^k - \tilde{T}^k = \sum_{r=0}^{k-1} T^r (T - \tilde{T}) \tilde{T}^{k-1-r}$$

$$\|T^k - \tilde{T}^k\|_\infty \leq k \|T - \tilde{T}\|_\infty$$

The error bound grows with walk length. It is an upper bound, not a growth prediction.

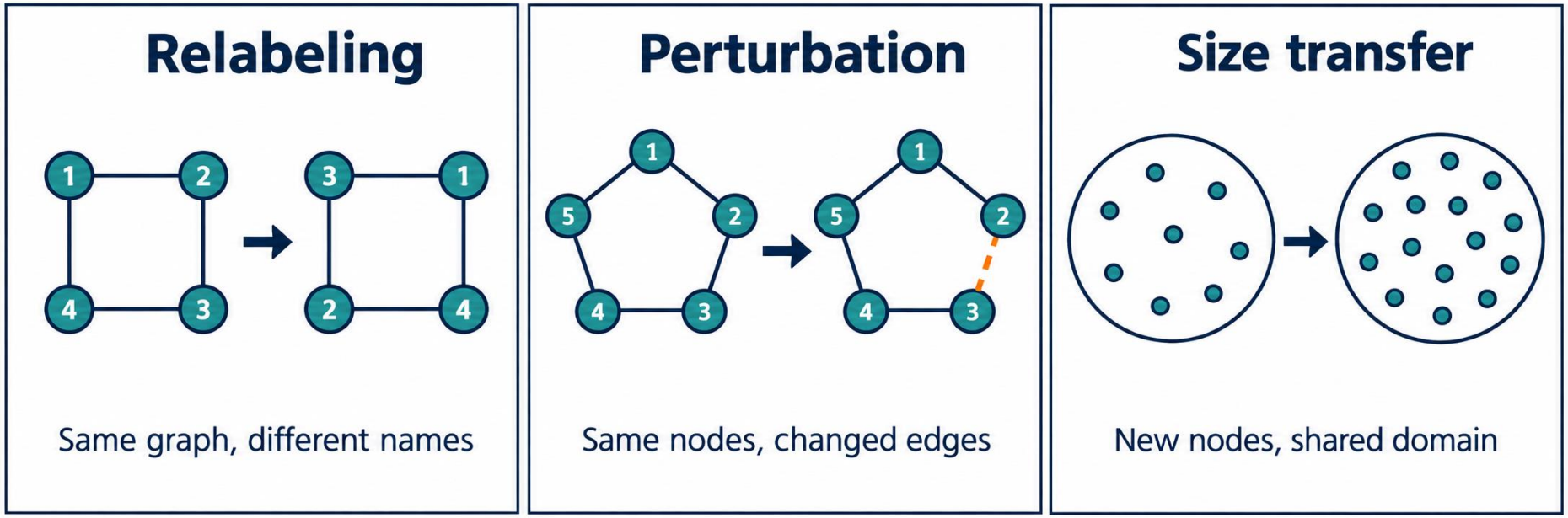
Small changes in A need not mean small changes in T when degrees are tiny.

A Lipschitz learned encoder preserves a bound, multiplied by its Lipschitz constant.

Specify the operator, norm, and normalization before claiming stability

Transferability: relabeling, perturbation, and size shifts

These comparisons require different mathematical and experimental setups.

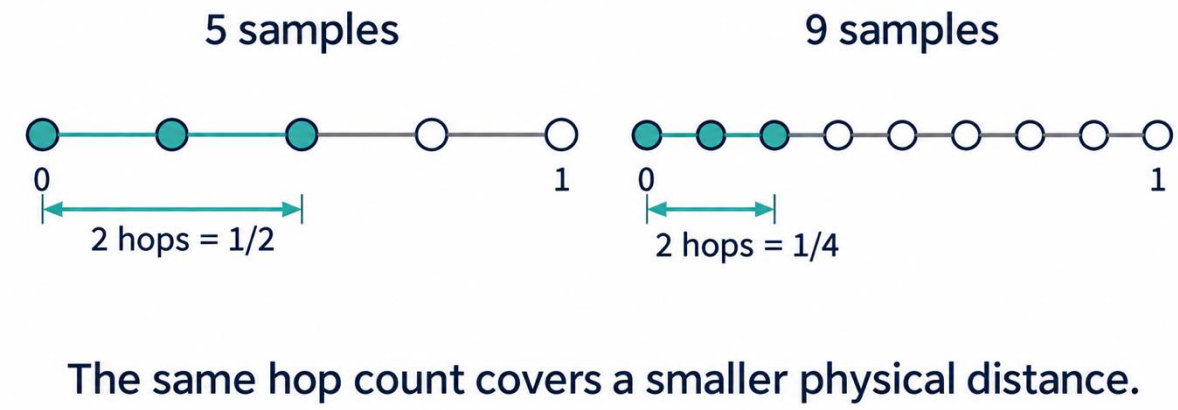


A permutation relates equal-size relabelings. It cannot match graphs with different node counts.
New graph sizes need sampling, correspondence, or a common-domain comparison.

Perturbation stability does not by itself prove size transferability

Transferability: coordinate normalization and spatial scale

An orthonormal discrete eigenvector uses a counting-measure normalization.



The physical interval stays fixed while sampling becomes denser.

Two hops cover 1/2 on the coarse path and 1/4 on the fine path.

Transfer needs compatible amplitude and spatial scales.

$$\sum_{i=1}^{n}(u_i^{(n)})^2 = 1, \qquad \frac{1}{n}\sum_{i=1}^{n}(\sqrt{n}\,u_i^{(n)})^2 = 1, \qquad \sqrt{n}\,u_i^{(n)} \approx f(x_i)$$

A shared parameter tensor is necessary for reuse, but insufficient for accurate transfer

Transferability: graphon and manifold operator limits

Connection to the earlier transferability lectures, rather than a new universal PE theorem.

$$S_n = A_n/n \quad \text{(dense graphon setting)}$$

$$\mathcal{L}_n \longrightarrow \Delta_{\mathcal{M}} \quad \text{(properly scaled geometric setting)}$$

Graphons: normalize dense aggregation and compare lifted operators on a common domain.

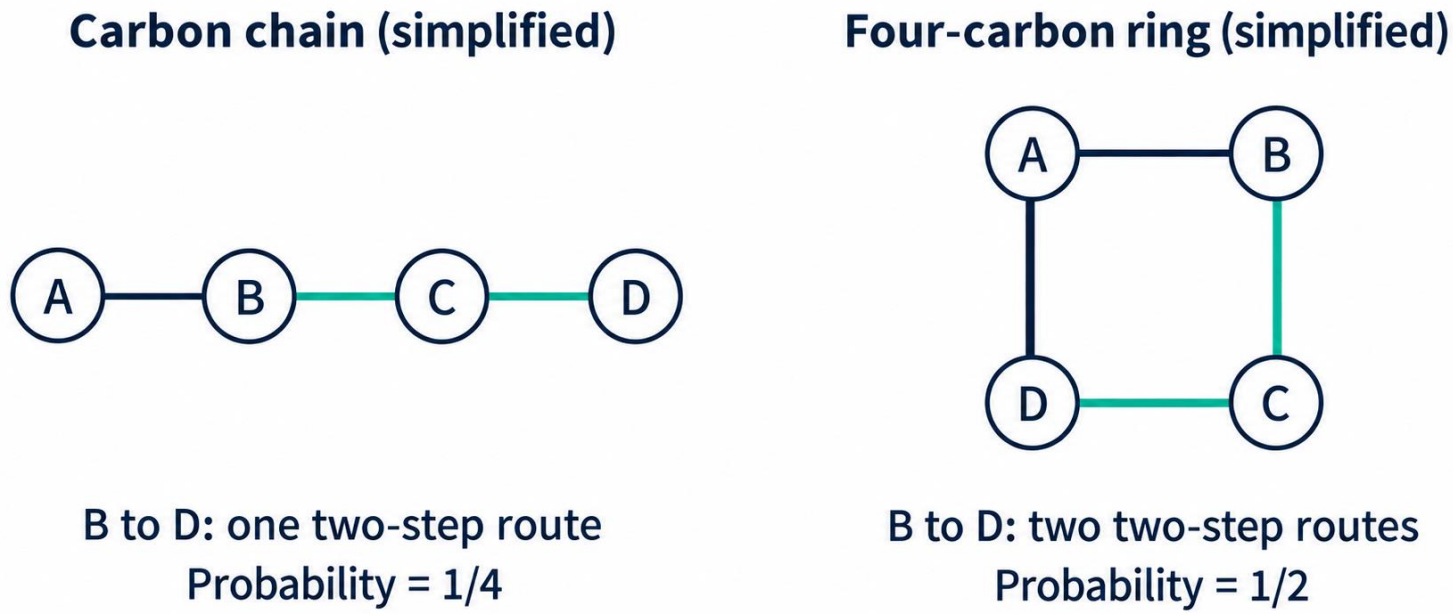
Manifolds: kernel bandwidth, sampling density, and Laplacian scaling must be specified.

Spectral-subspace alignment and filter regularity matter in both settings.

An encoding transfers only under the assumptions of its graph sequence and operator

Application: molecular chains, rings, and walk encodings

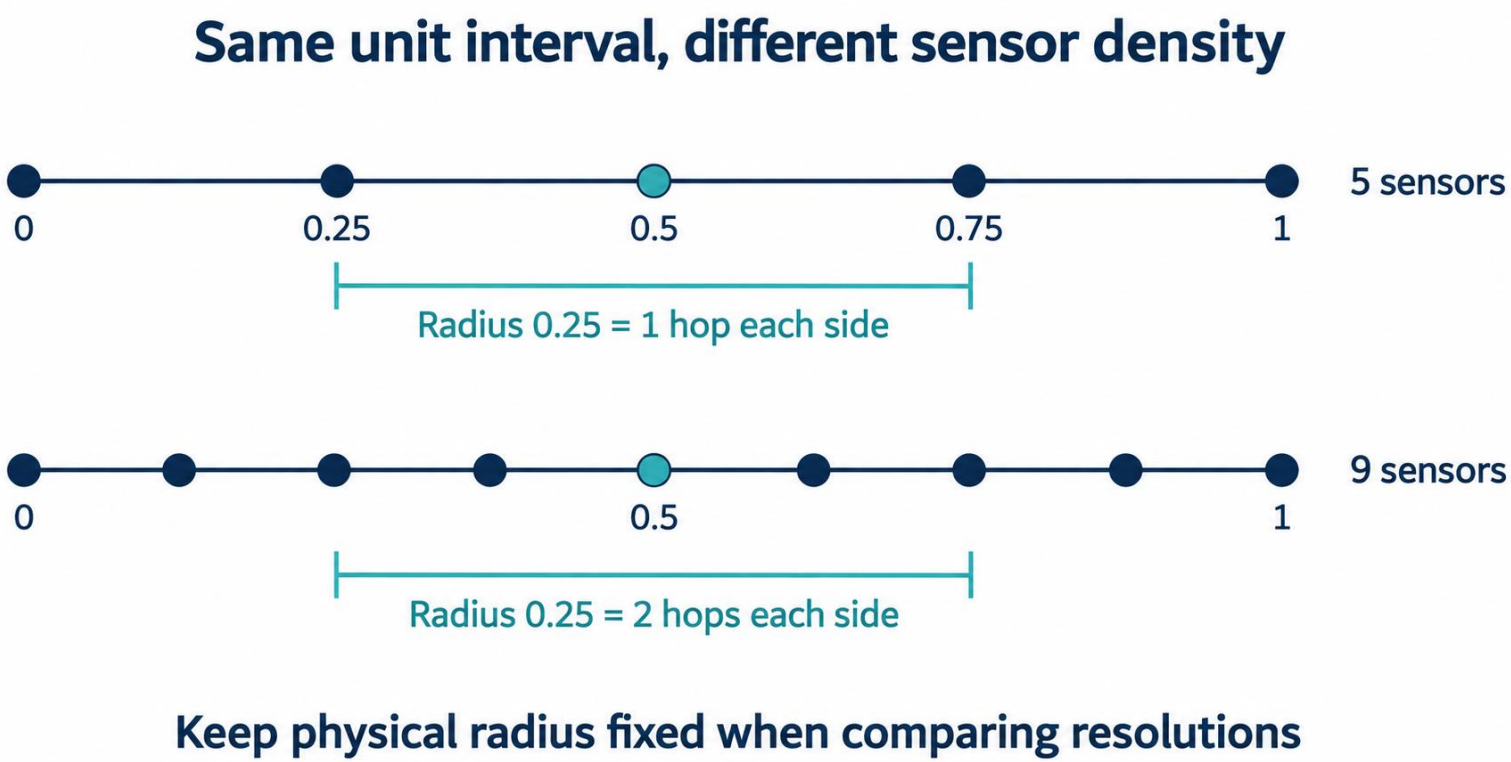
A topology-only calculation. Real molecular prediction also needs atom and bond attributes.



Equal shortest distance can hide different routes and different walk probabilities

Application: sensor density and a fixed physical radius

Estimate temperature at a sensor using nearby measurements on the same physical interval.



Recompute geometry on the new graph and compare neighborhoods in physical units

Computation: spectral and pair-encoding costs

Preprocessing and storage can dominate the attention layer itself.

Selected eigenvectors require an eigensolver; full decomposition is generally expensive.

Dense RRWP or spectral kernels store pair information at $O(Kn^2)$ or $O(n^2)$.

Short walk features, approximations, and sparse pair sets trade information for cost.

Report encoding construction, memory, and reuse across epochs separately

Rampášek et al. (2022). Recipe for a General, Powerful, Scalable Graph Transformer.
Ma et al. (2023). Graph Inductive Biases in Transformers without Message Passing.

Evaluation: separate expressivity, stability, and transfer

A controlled experiment isolates the encoding while keeping the backbone and budget fixed.

Expressivity probe: controlled graph pairs with matched size and node features.

Stability probe: edge-weight perturbations with measured operator differences.

Transfer probe: train on one size range, test on larger graphs from a stated generator.

Report accuracy and cost, with structural and size-distribution shifts made explicit

Summary: encoding choices and their guarantees

An encoding is useful only when its information and assumptions match the task.

Expressivity: the structures and relationships the model can distinguish.

Stability: sensitivity to a specified change in the graph or its operator.

Transferability: retained meaning and predictive accuracy across graph families or sizes.

References: absolute, relative, and learned encodings

Full titles and source links are included in the recording notes.

Dwivedi & Bresson (2020). A Generalization of Transformer Networks to Graphs.

Kreuzer et al. (2021). Rethinking Graph Transformers with Spectral Attention.

Ying et al. (2021). Do Transformers Really Perform Bad for Graph Representation?

Dwivedi et al. (2022). Graph Neural Networks with Learnable Structural and Positional Representations.

Chen, O’Bray & Borgwardt (2022). Structure-Aware Transformer for Graph Representation Learning.

Ma et al. (2023). Graph Inductive Biases in Transformers without Message Passing.

Read the mathematical definition before comparing architectures

References: symmetry, stability, and generalization

These papers support specific constructions and assumptions, rather than a universal ranking.

Lim et al. (2023). Sign and Basis Invariant Networks for Spectral Graph Representation Learning.

Wang et al. (2022). Equivariant and Stable Positional Encoding for More Powerful Graph Neural Networks.

Huang et al. (2024). On the Stability of Expressive Positional Encodings for Graphs.

Rampášek et al. (2022). Recipe for a General, Powerful, Scalable Graph Transformer.

Check the retained spectrum, norm, graph family, and experimental shift