Graph Representation Learning & Transfer

From Graphons and Spectral Theory to GNNs and Foundation Models

Yuyao Wang

yuyaow@bu.edu

Outline

Graphs are everywhere

Graph-structured data is ubiquitous:

- *Social / communication networks* — Facebook, Twitter, email graphs.
- *Molecules and proteins* — atoms as nodes, bonds as edges; protein interaction networks.
- *Knowledge graphs* — Wikidata, Freebase; nodes as entities, edges as relations.
- *Recommender systems* — bipartite user–item graphs.
- *Physical systems* — meshes, point clouds, particle interactions.
- *Neuroscience* — brain functional/structural connectomes.

The core problem. Classical ML assumes i.i.d. vectorial inputs. Graph data is:

- *relational* — observations depend on each other,
- of *variable size and topology*,
- permutation-invariant under node relabeling.

$\Rightarrow$ we need *representations* that respect these structures.

What is graph representation learning?

Goal. Learn a map Φ from a graph (or its nodes, edges, subgraphs) into a *Euclidean* feature space:

$$\Phi : \mathcal{G} \longrightarrow \mathbb{R}^d \qquad \text{(or node-level } \Phi : \mathcal{V} \to \mathbb{R}^d\text{)}$$

such that downstream tasks (classification, regression, clustering, link prediction) become easy.

Desiderata:

- **Permutation equivariance/invariance** — the representation should not depend on node labels.
- **Locality** — nodes with similar neighborhoods should have similar embeddings.
- **Scalability** — billions of nodes in industrial graphs.
- **Transferability** — representations learned on one graph should generalize to others.

Why "transfer"?

- Labels are expensive; we want to *pre-train* on large unlabeled graphs and *fine-tune*.
- A model trained on one social network should work on another.
- A molecular GNN trained on ZINC should help on a new drug screening dataset.

Graphs, adjacency, and Laplacians

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$. Node features $X \in \mathbb{R}^{n \times p}$.

Key matrices on the graph:

- **Adjacency** $A \in \{0,1\}^{n \times n}$: $A_{ij} = 1$ iff $(i,j) \in \mathcal{E}$.
- **Degree** $D = \mathrm{diag}(A\mathbf{1})$.
- **Combinatorial Laplacian** $L = D - A$.
- **Normalized Laplacian** $\mathcal{L} = I - D^{-1/2} A D^{-1/2}$.
- **Random-walk Laplacian** $L_{\mathrm{rw}} = I - D^{-1} A$.

Proposition (Spectrum of $\mathcal{L}$)

$\mathcal{L}$ is symmetric PSD with eigenvalues $0 = \lambda_1 \le \cdots \le \lambda_n \le 2$. The multiplicity of 0 equals the number of connected components.

Graph signals. A function $x : \mathcal{V} \to \mathbb{R}$ is a vector $x \in \mathbb{R}^n$. The Laplacian quadratic form

$$x^\top L x \;=\; \tfrac{1}{2} \sum_{(i,j) \in \mathcal{E}} (x_i - x_j)^2$$

measures *smoothness* of x over the graph.

Graph Fourier transform

Diagonalize $\mathcal{L} = U\Lambda U^\top$, with eigenpairs (λ_k, u_k).

Graph Fourier transform:

$$\hat{x} = U^\top x, \qquad x = U\hat{x}.$$

- Eigenvectors play the role of *Fourier modes*; eigenvalues play the role of *frequencies*.
- Low frequencies (λ_k small) $\Leftrightarrow$ *smooth* signals on the graph.
- High frequencies $\Leftrightarrow$ *oscillatory* signals, rapidly changing across edges.

Spectral graph filters: given $g : \mathbb{R} \to \mathbb{R}$, the filter $g(\mathcal{L})$ acts on signals as

$$g(\mathcal{L})x \;=\; U\, g(\Lambda)\, U^\top x.$$

This is the foundation of *spectral GNNs* (Bruna et al. 2014, ChebNet, GCN).

Random graph models — why statisticians care

Any principled analysis of GNNs requires a data-generating process. Three foundational models:

1. **Erdős–Rényi $G(n,p)$:** $A_{ij} \overset{\text{iid}}{\sim} \mathrm{Ber}(p)$.
   Too simple: all nodes equivalent. But the starting point for perturbation analysis.
2. **Stochastic Block Model (SBM):** nodes have hidden communities $\sigma(i) \in [K]$;

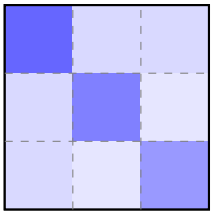
$$A_{ij} \mid \sigma \;\sim\; \mathrm{Ber}(B_{\sigma(i)\sigma(j)}), \quad B \in [0,1]^{K\times K}.$$

   Captures community structure; central in the networks literature.
3. **Graphon model:** the nonparametric limit of exchangeable random graphs. $U_i \overset{\text{iid}}{\sim} \mathrm{Unif}[0,1]$; $A_{ij} \mid U \sim \mathrm{Ber}(W(U_i,U_j))$, for $W : [0,1]^2 \to [0,1]$ symmetric measurable.

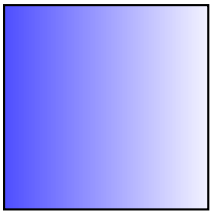
Key theorem (Aldous–Hoover, 1981; Lovász, 2012): every *exchangeable* random graph is generated by a graphon W, up to measure-preserving transformations.

$\Rightarrow$ graphons are *the* canonical nonparametric model of dense graphs.

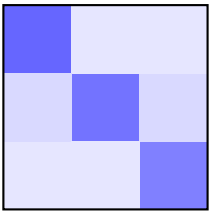
Visualizing graphons and SBMs



SBM ($K = 3$)



smooth graphon $W(u, v) = uv$



planted-community graphon

Interpretations:

- The SBM block matrix is a *step-function* graphon.
- Smooth graphons encode heterogeneous, continuous "positions" for nodes.
- Graphon distance (cut metric $\delta_\square$) gives a principled way to say two graphs are "similar."

Spectral methods and community detection

Classical statistical question: given A from an SBM, recover the community labels σ.

Spectral clustering (Von Luxburg; Rohe, Chatterjee, Yu 2011):

1. Compute top-K eigenvectors of A (or $\mathcal{L}$), stack into $U \in \mathbb{R}^{n \times K}$.
2. Run k-means on rows of U to assign labels.

Why does it work? Under the SBM, $\mathbb{E}[A]$ is low-rank with community-structured eigenvectors, and Davis–Kahan + matrix concentration ($\|A - \mathbb{E}A\|_{op} = O(\sqrt{np})$) give eigenvector stability.

Phase transitions

For two-block SBM with intra/inter probabilities $a/n, b/n$:

- *Detection threshold* (Decelle et al.; Mossel–Neeman–Sly): $(a-b)^2 > 2(a+b)$.
- Below threshold: no algorithm can beat random guessing (information-theoretic barrier).

$\Rightarrow$ community detection has a *sharp statistical theory* — a gold standard we aspire to in GNN analysis.

Shallow node embeddings

Before GNNs, the dominant approach was *shallow* embeddings: learn a matrix $Z \in \mathbb{R}^{n \times d}$ directly.

DeepWalk (Perozzi et al. 2014). Sample random walks on the graph; treat them as "sentences" and apply `word2vec` (skip-gram with negative sampling).
node2vec (Grover & Leskovec 2016). Biased random walks balancing BFS (structural) vs. DFS (community) exploration.
LINE (Tang et al. 2015). Preserve first- and second-order proximity.

Qiu et al. (2018): they are all matrix factorization

DeepWalk, node2vec, LINE, and PTE are all (approximately) factorizing a closed-form matrix:

$$\log\Big(\mathrm{vol}(G) \cdot \big(\tfrac{1}{T}\textstyle\sum_{r=1}^{T} P^r\big) D^{-1}\Big) \;-\; \log b,$$

where $P = D^{-1}A$ is the transition matrix and b the number of negative samples.

Statistical take: shallow embeddings are *transductive* (no out-of-sample generalization), do not use node features, and are equivalent to spectral methods with a specific kernel.

The message-passing framework (Gilmer et al. 2017)

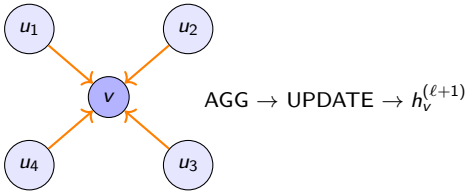
Every modern GNN fits the *message-passing neural network (MPNN)* template. At each layer ℓ:

$$h_v^{(\ell+1)} \;=\; \phi^{(\ell)}\Big(h_v^{(\ell)},\; \mathrm{AGG}\big\{\!\!\big\{\, \psi^{(\ell)}(h_v^{(\ell)}, h_u^{(\ell)}) : u \in \mathcal{N}(v) \,\big\}\!\!\big\}\Big)$$

Design choices:

- ψ — the *message function*.
- AGG — a permutation-invariant aggregator (sum, mean, max, attention).
- ϕ — the *update function* (usually an MLP).

Graph-level prediction: pool node embeddings after L layers: $h_G = \mathrm{READOUT}(\{h_v^{(L)} : v \in \mathcal{V}\})$.



Four canonical GNNs

GCN (Kipf & Welling, 2017): first-order spectral filter + self-loops:

$$H^{(\ell+1)} \;=\; \sigma\big(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}\,H^{(\ell)}\,W^{(\ell)}\big), \qquad \tilde{A} = A + I.$$

GraphSAGE (Hamilton, Ying, Leskovec, 2017): inductive, sampled neighborhoods:

$$h_v^{(\ell+1)} = \sigma\big(W^{(\ell)} \cdot [\, h_v^{(\ell)} \,\|\, \mathrm{AGG}(\{h_u^{(\ell)} : u \in \mathcal{S}(v)\})\,]\big).$$

GAT (Veličković et al. 2018): learnable attention over neighbors:

$$\alpha_{vu} = \mathrm{softmax}_u\big(\mathrm{LeakyReLU}(a^\top [Wh_v \| Wh_u])\big), \qquad h_v^{(\ell+1)} = \sigma\big(\textstyle\sum_{u\in\mathcal{N}(v)\cup\{v\}} \alpha_{vu}\, Wh_u^{(\ell)}\big).$$

GIN (Xu et al. 2019): provably maximally expressive among 1-WL GNNs:

$$h_v^{(\ell+1)} = \mathrm{MLP}^{(\ell)}\big((1+\epsilon^{(\ell)})\,h_v^{(\ell)} + \textstyle\sum_{u\in\mathcal{N}(v)} h_u^{(\ell)}\big).$$

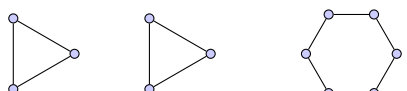
Expressive power — the WL connection

Weisfeiler–Leman (1-WL) color refinement. Iteratively re-color each node by a hash of its current color and the multiset of neighbor colors. Two graphs are *1-WL equivalent* if the resulting color histograms match.

Theorem (Morris et al.; Xu et al. 2019)

Any message-passing GNN with *countable* aggregation is *at most as powerful as 1-WL* in distinguishing non-isomorphic graphs. GIN achieves this upper bound.

Consequence: a real limitation. 1-WL cannot distinguish many natural pairs:



2 disjoint triangles vs 6-cycle: indistinguishable by 1-WL (both are 2-regular).

Higher-order remedies: k-GNNs, Provably Powerful GN, ring/cycle features, subgraph GNNs.

Spectral vs. spatial GNNs

Spectral view	**Spatial (message-passing) view**
Filter $g_\theta(\mathcal{L})$ in the eigenbasis.	Aggregate from local neighborhoods.
ChebNet: $g_\theta = \sum_k \theta_k T_k(\tilde{\mathcal{L}})$, Chebyshev polynomials.	GCN, GraphSAGE, GAT, GIN.
Naturally global; polynomial filters $\equiv$ k-hop locality.	Intrinsically local; L layers $\to$ L-hop receptive field.
Transferability via graphon filters.	Transferability via graphon message passing.

Key identity. A degree-K polynomial spectral filter

$$g_\theta(\mathcal{L}) \;=\; \sum_{k=0}^{K} \theta_k \, \mathcal{L}^k$$

is *exactly* a K-hop message-passing scheme. Spectral and spatial are two sides of the same coin.

Pathologies: over-smoothing and over-squashing

Deep GNNs suffer from two geometric pathologies that statisticians should recognise.

Over-smoothing (Li et al. 2018; Oono & Suzuki 2020). As depth $L \to \infty$, node embeddings collapse to a common subspace $\Rightarrow$ the model cannot distinguish nodes.
Formally: if σ is non-expansive and $\left\|W^{(\ell)}\right\| \leq 1$, then after L layers,

$$\left\|H^{(L)} - \mathbf{1}\bar{h}^{\top}\right\|_F \;\leq\; \lambda_*^{L} \left\|H^{(0)} - \mathbf{1}\bar{h}_0^{\top}\right\|_F,$$

where $\lambda_* < 1$ is the second eigenvalue of the propagation operator. $\Rightarrow$ exponential contraction.

Over-squashing (Alon & Yahav 2021; Topping et al. 2022). Information from distant nodes must be compressed through bottleneck edges. Formalized via *Ricci curvature* of graph edges; negatively curved edges squash information.

Remedies: residual connections, PairNorm, graph rewiring, graph transformers, positional encodings.

From message passing to transformers on graphs

Motivation. Message passing is inherently local $\Rightarrow$ long-range dependencies need many layers $\Rightarrow$ over-smoothing / over-squashing. Solution: *fully-connected attention* + *structural encodings*.

Graph Transformer (Dwivedi & Bresson; GraphGPS framework):

$$h_v^{(\ell+1)} \;=\; \mathrm{MHA}\big(h_v^{(\ell)};\, \{h_u^{(\ell)}\}_{u\in\mathcal{V}}\big) + \mathrm{MPNN}(h_v^{(\ell)}) + \mathrm{PE}(v),$$

combining (i) global attention, (ii) local message passing, and (iii) structural positional encodings.

Structural / positional encodings — critical for expressiveness:

- *Laplacian PE:* eigenvectors of $\mathcal{L}$ (analogue of sinusoidal PE in sequences).
- *Random-walk PE:* diagonals of P^k for $k = 1, \ldots, K$.
- *Shortest-path encodings* (Graphormer, Ying et al. 2021).

Recent SOTA on molecules: Graphormer (KDD Cup 2021 winner), GPS+, TokenGT. $\Rightarrow$ graph transformers consistently outperform pure MPNNs on long-range tasks.

Why self-supervised?

The labeling bottleneck.

- Labels on graphs are *very* expensive — manual annotation for molecules, curated protein functions, fraud labels.
- But *unlabeled graphs* are abundant: ZINC (250M molecules), OGB, social networks.

The SSL/pre-training recipe (copied from NLP and vision):

1. Design a *pretext task* on unlabeled graphs.
2. Pre-train Φ_θ to high capacity.
3. *Fine-tune* on the small labeled target task.

Three families of SSL on graphs:

- *Contrastive* — maximize agreement between augmented views. (GraphCL, GCC, GCA)
- *Generative / reconstructive* — mask and reconstruct structure/features. (GraphMAE, VGAE)
- *Predictive* — predict structural properties (degree, centrality, motif counts). (S^2GRL)
- *Bootstrap* — predict one view from another without negatives. (BGRL)

Contrastive graph SSL

GraphCL (You et al. 2020). Given a graph G, produce two augmented views $G^{(1)}, G^{(2)}$ (node dropping, edge perturbation, subgraph sampling, attribute masking). Train

$$\mathcal{L}_{\mathrm{NCE}} \;=\; -\mathbb{E}\left[\log \frac{\exp(\langle z^{(1)},\, z^{(2)}\rangle/\tau)}{\sum_j \exp(\langle z^{(1)},\, z^{(j)}\rangle/\tau)}\right],$$

i.e. InfoNCE pulling positive pairs together and pushing negatives apart.

Statistical interpretation (HaoChen et al. 2021 adapted to graphs). Contrastive loss approximates a *spectral decomposition of the augmentation graph*. The learned embedding space recovers the top eigenvectors of a positive-pair kernel.

Why it transfers. If augmentations preserve task-relevant invariances, pre-training aligns representations with the *latent variables* of the generative model (e.g. community labels in an SBM-like setting).

Pitfalls: augmentation design is *critical*; poor augmentations (e.g. breaking molecular chirality) can destroy task information.

Why is transfer on graphs hard?

Non-graph transfer: typically same input space $\mathcal{X} = \mathbb{R}^d$, changes in $\mathbb{P}(X)$ or $\mathbb{P}(Y|X)$.

Graph transfer has extra sources of shift:

1. **Feature shift** — node/edge attribute distributions differ.
2. **Structural shift** — degree distributions, clustering coefficients, graph size.
3. **Label/task shift** — target labels may not align.
4. **Size generalization** — trained on graphs of size n; tested on $n' \gg n$.

A fundamental asymmetry: unlike fixed-dimensional vectors, graphs live in a space of varying dimension. What does it even *mean* for two graphs to be "close"?

$\Rightarrow$ need a notion of graph similarity that is:

- permutation-invariant,
- dimension-agnostic (compares graphs of different sizes),
- compatible with statistical concentration.

Answer: graphon distance.

Graphons as the vehicle of transfer

Setup. Both source and target are sampled from the *same* graphon W, or from *close* graphons in the cut metric $\delta_\square$:

$$\delta_\square(W_1, W_2) \;=\; \inf_{\varphi} \sup_{S,T\subseteq[0,1]} \left| \int_{S\times T} (W_1 - W_2^{\varphi}) \right|,$$

where the infimum is over measure-preserving bijections φ.

Graphon transferability theorem (Ruiz, Chamon, Ribeiro 2021–2023)

Let Φ be a graphon neural network with Lipschitz spectral filters and Lipschitz nonlinearities. For graphs G_n, G_m sampled from graphon W with n, m nodes:

$$\|\Phi_{G_n}(x_n) - \Phi_{G_m}(x_m)\|_2 \;\le\; C \cdot \left(n^{-1/2} + m^{-1/2}\right) \;+\; (\text{sampling term}),$$

with high probability.

Interpretation. Once a GNN is Lipschitz in the spectral domain, *it automatically transfers*. Graphon NNs are *consistent* across graph size.

Size generalization and structural transfer

Size generalization (Yehudai et al. 2021). Even simple GNNs can fail dramatically when tested on graphs *larger* than the training graphs — degree distributions shift, aggregation statistics change.

Explanations:

- *Mean aggregation:* $\frac{1}{|\mathcal{N}(v)|}\sum_u h_u$ is more size-stable than sum.
- *Sum aggregation:* more expressive but degree-dependent $\Rightarrow$ sensitive to size shift.
- *Max aggregation:* scale-free but discards density.

Theoretical lens: under a graphon generating process, *normalized* aggregators converge to graphon operators; *un-normalized* ones diverge as n grows.

Structural transfer under domain shift:

- *StruRW (Structural Reweighting, Liu et al. 2022):* adjust edge weights to match source-to-target degree distributions.
- *EGI (Zhu et al. 2021):* information-theoretic pre-training objective tied to structural regularity.
- *Pair-align, GCOPE, SpecReg:* align spectra, covariances, or ego-graph distributions across domains.

A statistical theory of graph transfer

Domain adaptation bound (Ben-David et al. 2010, adapted to graphs).
For target risk $\varepsilon_T(h)$ and source risk $\varepsilon_S(h)$:

$$\varepsilon_T(h) \;\le\; \varepsilon_S(h) \;+\; \tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \;+\; \lambda^*,$$

where λ^* is the best joint risk, and $d_{\mathcal{H}\Delta\mathcal{H}}$ a hypothesis-class divergence.

For graphs, $\mathcal{D}_S, \mathcal{D}_T$ are distributions over graphs. Options to instantiate the bound:

- Compare *graphon representations* via cut distance.
- Compare *empirical spectral distributions* of $\mathcal{L}_S, \mathcal{L}_T$.
- Compare *ego-graph distributions* under a fixed-depth GNN.

PAC-Bayes for GNNs (Liao et al.; Garg et al. 2020):

$$\mathcal{R}(h) \;\le\; \hat{\mathcal{R}}(h) \;+\; O\left(\sqrt{\frac{\text{capacity}(h) \cdot \text{max-degree}^L}{n}}\right).$$

Generalization *degrades* with maximum degree and depth — matching empirical observations.

Towards graph foundation models

Analogy to LLMs. In NLP:

$$\text{pre-train on huge corpus} \rightarrow \text{fine-tune / prompt for any task.}$$

Can we do this on *graphs*?

Fundamental obstacle. Graphs from different domains don't share a "vocabulary":

- Molecules vs. social networks vs. knowledge graphs all have different node/edge semantics.
- No universal token set; feature dimensions differ.

Current approaches to graph foundation models (2023–2026):

- *One For All (OFA, Liu et al. 2024):* unify node/edge features into text embeddings via an LLM encoder.
- *GraphGPT / LLaGA:* convert graphs into token sequences ingested by LLMs.
- *GFT, GraphAny:* learn *task-agnostic* graph vocabularies.
- *Prompt tuning on graphs:* GPPT, GraphPrompt — apply NLP-style prompting.

Open question: is there a *scaling law* for graph data analogous to LLMs? Current evidence is mixed (Liu et al. 2024 find sub-linear scaling).

LLMs meet graphs

Three paradigms for combining LLMs and graphs:

Paradigm	Idea	Example
LLM as enhancer	Use LLM to generate/refine node features, then feed into a GNN.	TAPE, OFA
LLM as predictor	Serialize the graph into text; let the LLM do everything.	NLGraph, GraphGPT
LLM + GNN aligned	Joint embedding via contrastive or projection layer.	LLaGA, GraphLLM

What works, what doesn't (empirical lessons):

- LLMs are surprisingly good at *text-attributed graph* (TAG) tasks when structure is shallow.
- LLMs are *poor* at tasks requiring deep multi-hop reasoning on graph structure alone.
- Hybrid GNN+LLM systems currently lead on heterogeneous, text-rich benchmarks (OGB-arxiv, ogbn-products-like).

Frontier: diffusion, equivariance, causality

Graph diffusion models. DiGress, EDM, GeoLDM — generate molecules and graphs via discrete/continuous diffusion. SOTA on drug design and material science.

Equivariant / geometric GNNs. For data with spatial structure (proteins, crystals, point clouds): $E(3)$-equivariant message passing (EGNN, SchNet, DimeNet, MACE, Equiformer). Core in AlphaFold 2/3.

Causal graph learning. GNNs for treatment-effect estimation on networked data (interference problem, Ogburn & VanderWeele). Identifiability under structural causal models with graph-valued covariates.

OOD generalization and invariance. GOOD benchmark, EERM, GSAT — learn invariant subgraphs across environments.

Provably expressive architectures. k-IGNs, PPGN, Graphormer-GD — architectures bounded by k-WL rather than 1-WL.

Open problems

1. **Minimax rates for GNN estimation.** What is the optimal rate for estimating graph functionals (e.g. regression $y = f(G)$) under graphon or SBM models? Only partial answers so far (Maskey et al. 2022; Ma & Xu 2023).
2. **Sample complexity of transfer.** Precise bounds on *how many target-domain labels* are needed after graphon-pretraining.
3. **Identifiability of foundation models.** Can we characterize, statistically, *what* a pre-trained GNN learns?
4. **Beyond 1-WL in a principled way.** Develop architectures whose expressivity matches their statistical efficiency.
5. **Sparse graphs.** Graphon theory is *dense* (edge density $\Theta(1)$). Real-world graphs are sparse — we need graphex / sparse graph limits (Veitch & Roy; Caron & Fox).
6. **Uncertainty quantification.** Conformal prediction on graphs (Huang et al. 2023); calibrated posterior for network data.
7. **Privacy and fairness on graphs.** Differential privacy under relational data is much more delicate than in the i.i.d. setting.

Synthesis: the statistical view

What statistics contributes to graph representation learning
Graphons as the universal limit of exchangeable graphs (Aldous–Hoover; Lovász).
Spectral concentration of random graphs (Davis–Kahan, matrix Bernstein).
Minimax rates for SBM estimation and community detection.
Phase transitions and information-theoretic thresholds (Mossel–Neeman–Sly).
Domain adaptation theory (Ben-David et al. bounds, adapted to graph distributions).
PAC-Bayes generalization bounds for GNNs (Liao et al.; Garg et al.).
Inference tools: bootstrap on networks, conformal prediction on graphs, network-valued U-statistics.

The emerging statistical picture:

- Graphs are samples from a *latent object* (graphon, exchangeable array, graphex for sparse regime).
- GNNs are *estimators* of graphon functionals — their error decomposes into bias (graphon approximation) and variance (finite sample).
- *Transfer* is the statement that the latent object is shared, even when surface graphs differ in size or labeling.
- *Foundation models* are an attempt to nonparametrically learn a very rich latent object from vast unlabeled graph data.

Essential references

Books and surveys:

- L. Lovász, *Large Networks and Graph Limits* (2012) — graphon theory bible.
- W. Hamilton, *Graph Representation Learning Book* (2020, open access).
- Y. Ma & J. Tang, *Deep Learning on Graphs* (2021).
- A. Bronstein et al., *Geometric Deep Learning* book and ICLR 2021 course.

Landmark papers:

- Kipf & Welling (2017) — GCN.
- Veličković et al. (2018) — GAT.
- Xu et al. (2019) — GIN and the WL connection.
- Rohe, Chatterjee, Yu (2011) — spectral clustering for SBM.
- Ruiz, Chamon, Ribeiro (2021–2023) — graphon NNs and transferability.
- You et al. (2020) — GraphCL; Hou et al. (2022) — GraphMAE.
- Ying et al. (2021) — Graphormer; Rampasek et al. (2022) — GraphGPS.
- Liu et al. (2024) — One For All and graph foundation models.