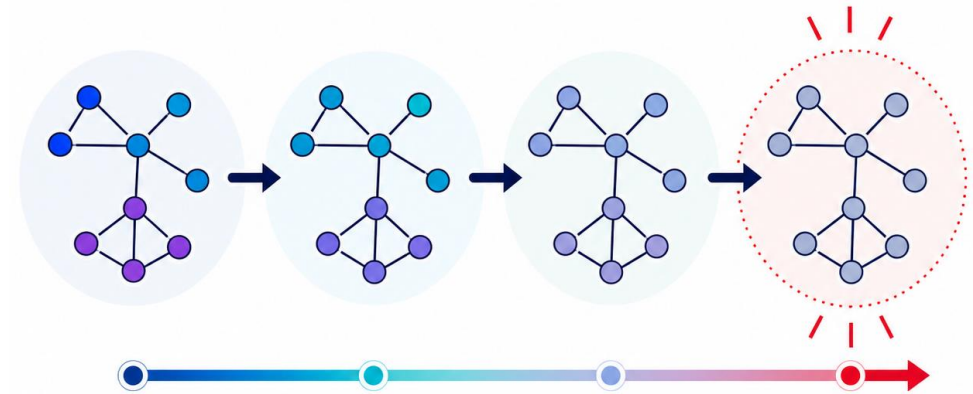


Oversmoothing in graph neural networks

Lecture 11

- Understand the stationary limit
- Measure the loss of node variation
- Choose a targeted intervention



Repeated averaging can remove the differences a prediction task needs.

Questions for today

WHAT COLLAPSES?

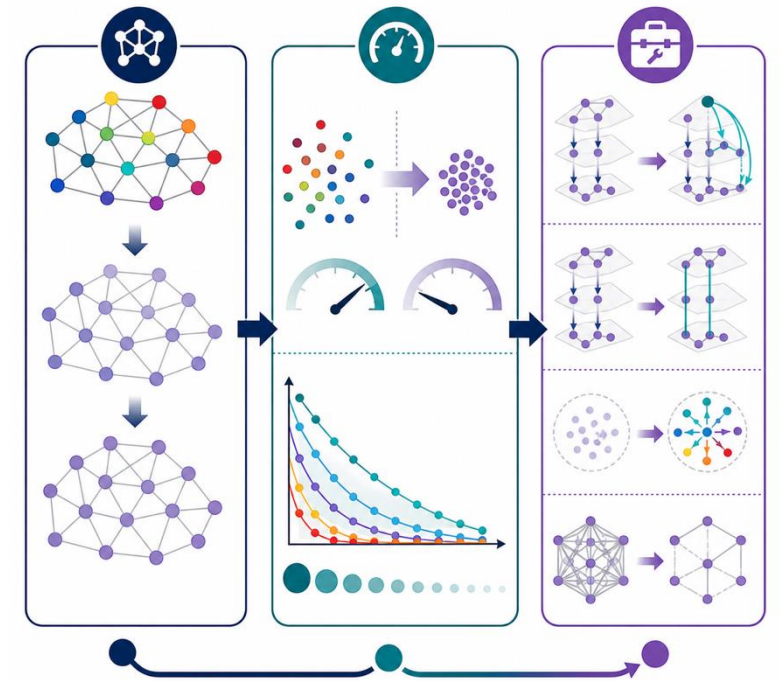
Node representations approach a small stationary subspace.

HOW CAN WE TELL?

Inspect energy, distances, feature scale, and task separation.

WHAT SHOULD WE CHANGE?

Preserve information or change how far and how strongly it mixes.

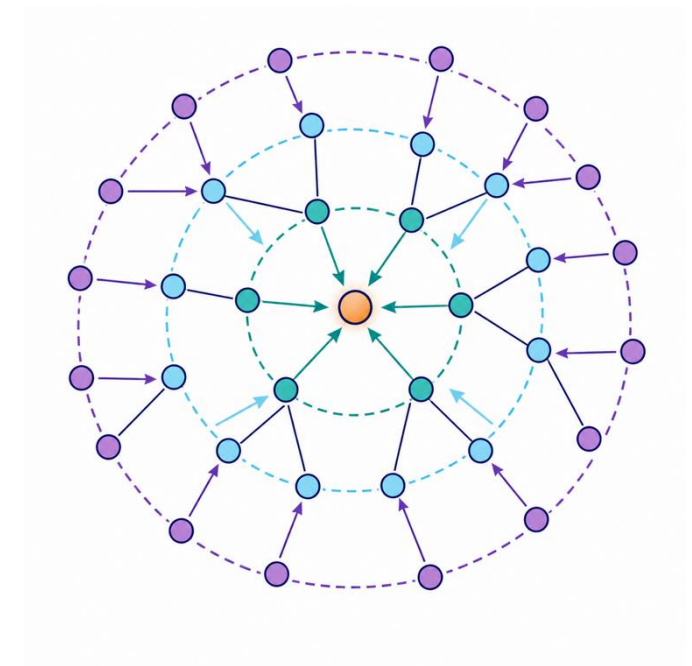


Depth and the receptive field

A one-hop layer moves information across at most one edge.

$$\text{dist}(u, v) \leq K \Rightarrow h_v^{(K)} \text{ can depend on } x_u$$

Pipeline example: a local alarm may need measurements from several upstream sensors.

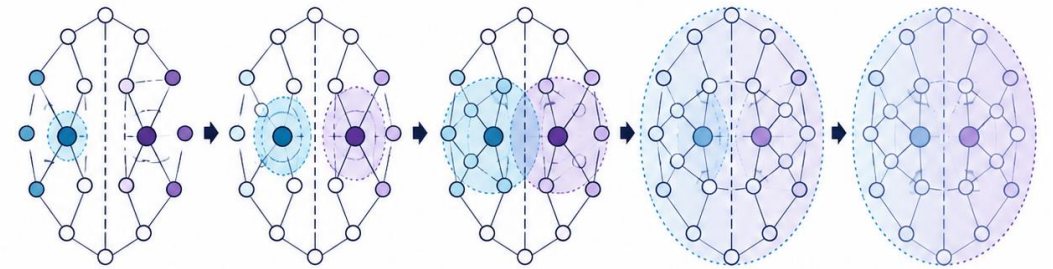


A reachable node need not have a strong influence.

Context and feature contrast

CONTEXT

More layers expose a node to information from farther away.



CONTRAST

Averaging can suppress differences that locate a hotspot or a tissue boundary.



Whether agreement helps depends on the prediction target.

What oversmoothing means

After many propagation steps, every node ends up with (almost) the same feature vector.

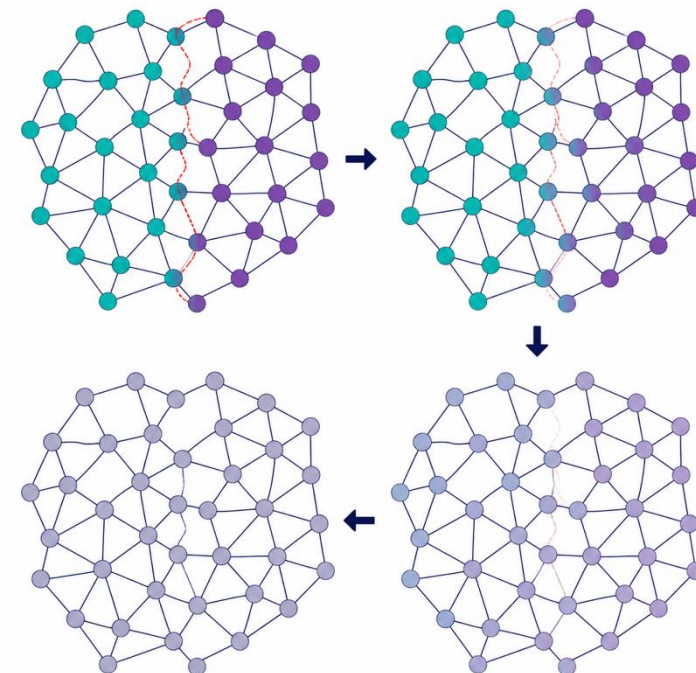
Setup: propagate on a connected graph with self-loops.

1 AVERAGE OVER NEIGHBORS (row-normalized)

$$P = D^{-1}A, \quad h_i^{(K)} \rightarrow c$$

2 GCN NORMALIZATION (symmetric)

$$S = D^{-1/2}AD^{-1/2}, \quad h_i^{(K)} \rightarrow d_i^{1/2} c$$



Node colors merge as propagation repeats

Either way the feature matrix collapses to rank one, so compare directions, not raw values.

Is oversmoothing a problem for all GNNs?

Architecture	Susceptibility	Why
GCN, SGC (Simplified GCN): fixed low-pass filter	High	Same contraction every layer; converges to dominant eigenvector
GraphSAGE, GIN, GAT	Moderate	Aggregation still smooths; self-terms and attention only slow it
GCNII, APPNP, JK-Net	Low	Residual / initial connections keep a non-trivial fixed point
PairNorm, DropEdge, GraphCON	Low	Normalization, sparsification or non-diffusive dynamics

Oversmoothing is a property of repeated averaging, not of GNNs in general.

Source: Wu et al., Simplifying Graph Convolutional Networks (SGC), ICML 2019;
Rusch et al., A Survey on Oversmoothing in GNNs, 2023; Wu et al., NeurIPS 2023.

The propagation core of a GCN

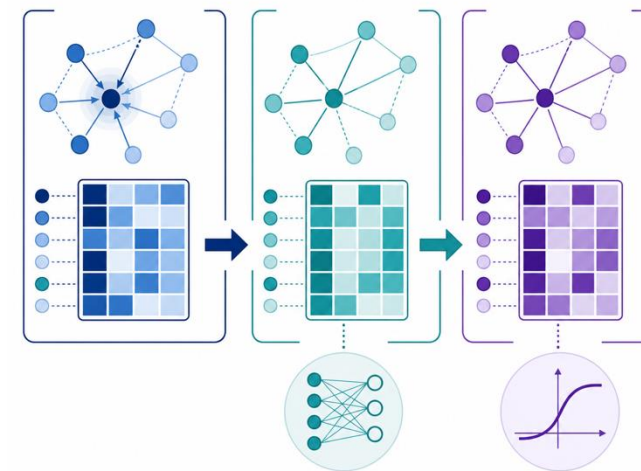
$$H^{(\ell+1)} = \sigma(\hat{S}H^{(\ell)}W^{(\ell)})$$

Left multiplication mixes nodes. Right multiplication transforms features.

$$\tilde{A} = A + I, \quad \hat{S} = \tilde{D}^{-1/2} \tilde{A} \tilde{D}^{-1/2}$$

FIXED GRAPH, NO NONLINEARITIES OR BIASES

$$H^{(K)} = \hat{S}^K X \Theta, \quad \Theta = W^{(0)} \dots W^{(K-1)}$$



The next derivation isolates pure propagation, not every trained GNN.

Why graph modes decay

$$\hat{S} = U\Lambda U^T$$

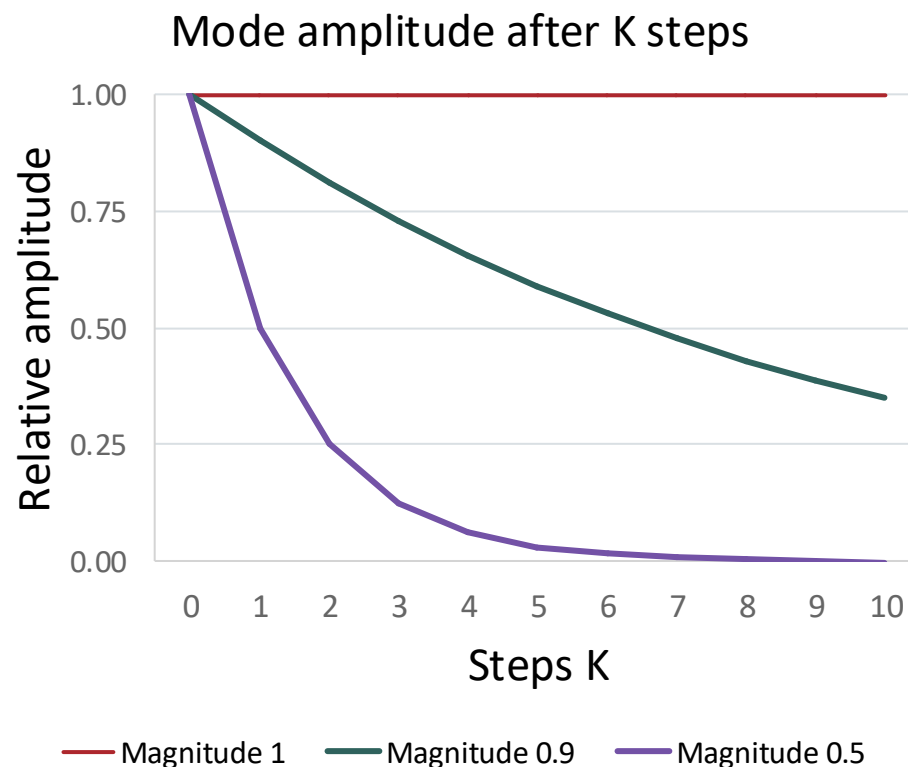
$$\hat{S}^K X = \sum_{i=1}^n \lambda_i^K u_i u_i^T X$$

Each eigenvector u_i is a pattern on the graph;

λ_i says how much of it survives one step.

After K steps the pattern is scaled by λ_i^K .

Only the smoothest pattern ($\lambda = 1$) keeps its size; every other pattern shrinks toward zero.



Propagation keeps the smoothest pattern and shrinks every other one.

The stationary limit

Keep propagating on a connected graph (with self-loops). What do the node features become?

GCN NORMALIZATION

$$\hat{S}^K X \rightarrow u_1 u_1^T X, \quad u_1 = \frac{\tilde{D}^{1/2} \mathbf{1}}{\sqrt{\sum_i \tilde{d}_i}}$$

All nodes end up pointing the same way; a node with more neighbors just gets a longer vector.

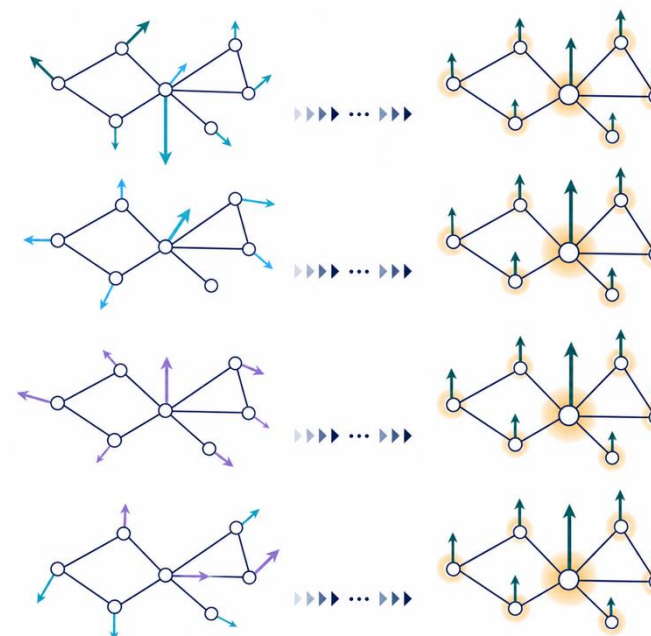
AVERAGE OVER NEIGHBORS

$$P^K X \rightarrow \mathbf{1} \pi^T X, \quad \pi_i = \frac{\tilde{d}_i}{\sum_j \tilde{d}_j}$$

All nodes end up with the same vector: an average of the inputs, weighted by degree.

After enough steps, every node carries the same information.

Source: Li, Han & Wu, Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning, AAAI 2018.



Dirichlet energy: how different are neighbors?

WHAT IT MEASURES

$$E_L(H) = \text{Tr}(H^T L H) = \frac{1}{2} \sum_{i,j} A_{ij} \|h_i - h_j\|_2^2 \quad \text{Normalized: } \bar{E}(H) = E_L(H) / \|H\|^2$$

Large: neighbors disagree.

Zero: every connected node has the same feature.

WHAT IT TELLS US ABOUT OVERSMOOTHING

Healthy network: energy stays roughly level as you add layers.

Oversmoothing: energy slides toward zero as layers stack; neighbors are becoming copies of each other.

Normalized energy falling toward zero with depth is the fingerprint of oversmoothing.

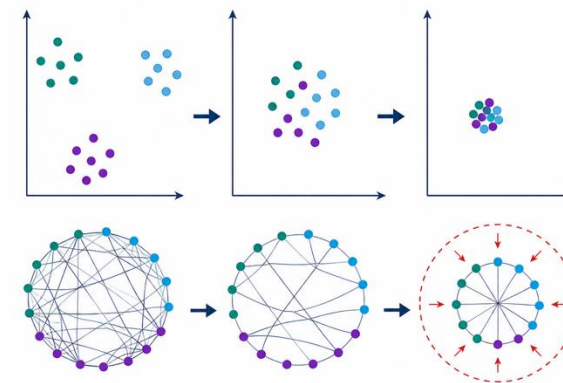
Distances and Mean Average Distance (MAD)

MAD averages cosine dissimilarity over a specified set of node comparisons.

$$d_{cos}(i, j) = 1 - \frac{h_i^T h_j}{\|h_i\|_2 \|h_j\|_2}$$

$$MADGap = MAD_{remote} - MAD_{near}$$

Report the pair mask. Handle zero vectors.
Cosine distance ignores feature magnitude.
Graph distance is not a class label.



Combine distance diagnostics with feature scale and task separation.

Spread and effective rank

STEP 1: REMOVE THE SHARED PART

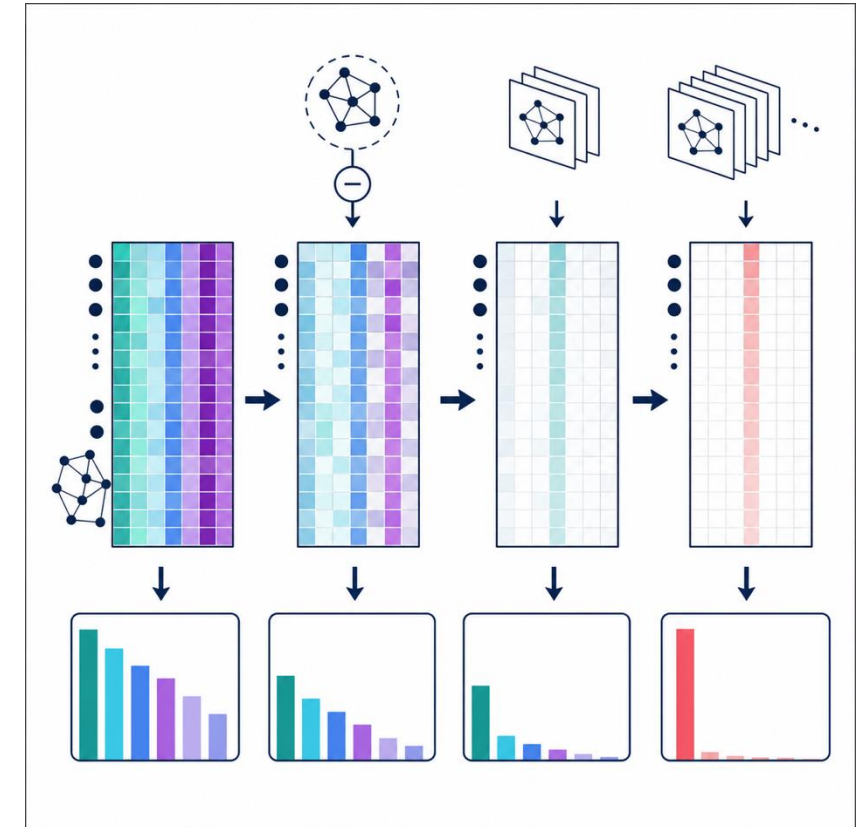
$$\mathbf{H}_c = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{n} \right) \mathbf{H}$$

$$\mathbf{H}_\perp = (\mathbf{I} - \mathbf{\Pi})\mathbf{H}, \quad \mathbf{\Pi} = \mathbf{u}_1\mathbf{u}_1^T$$

STEP 2: COUNT THE DIRECTIONS THAT REMAIN

$$\text{erank}(\mathbf{H}_c) = \exp(-\sum_i p_i \log p_i), \quad p_i = \sigma_i / \sum_j \sigma_j$$

Effective rank counts how many directions the features really use -
- soft, continuous version of matrix rank



Effective rank falling toward 1 with depth means the features have collapsed.

Accuracy alone is not enough

LOOK FOR JOINT EVIDENCE

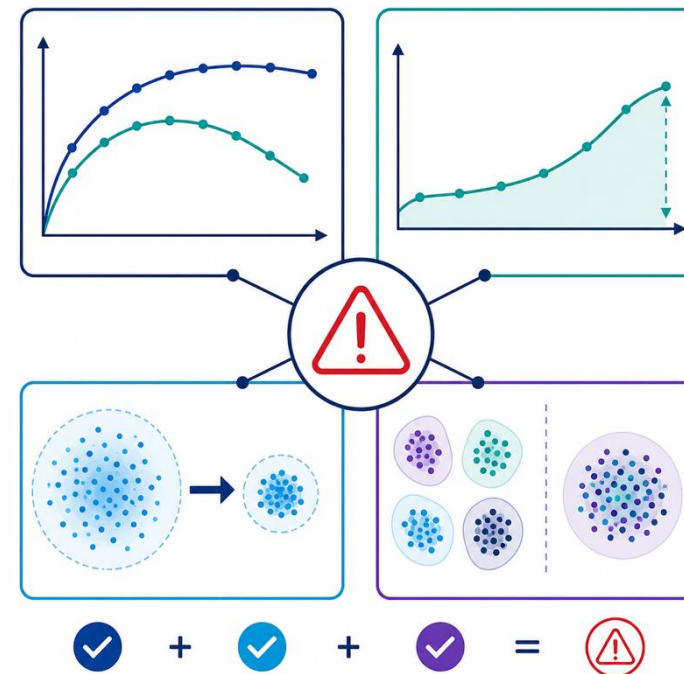
Compare training and validation behavior.

Log energy, distances, norms, and class separation.

Compare a shallow GNN with a no-message MLP.

Control training budget, capacity, and regularization.

Test whether a targeted remedy restores useful information.

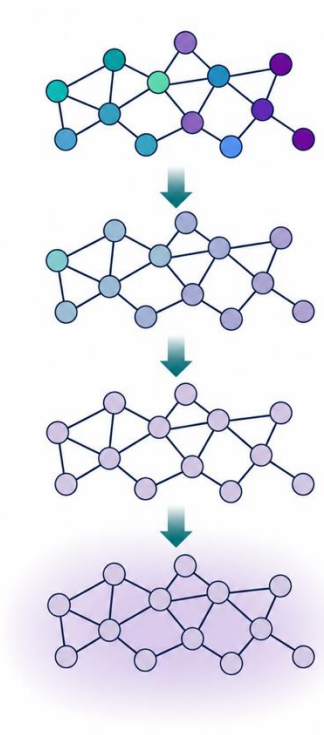


Lower accuracy at greater depth does not identify the failure mechanism.

Oversmoothing and oversquashing

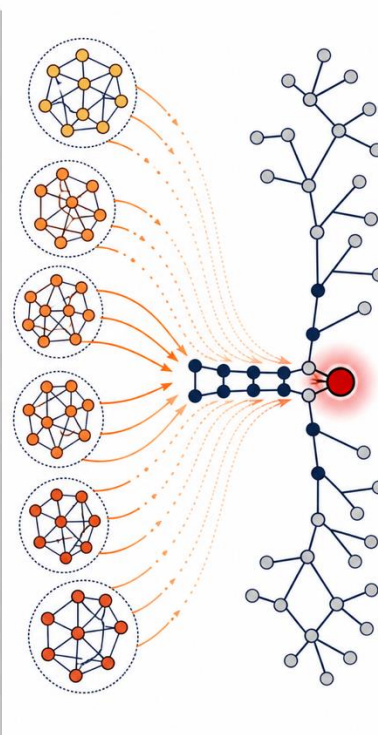
OVERSMOOTHING

Repeated mixing removes distinctions between node representations.



OVERSQUASHING

Many long-range dependencies compete for limited message-passing capacity.



Adding edges can ease a bottleneck while also accelerating mixing.

Where remedies act

PRESERVE INPUT

Initial residuals and restart

CONTROL PROPAGATION

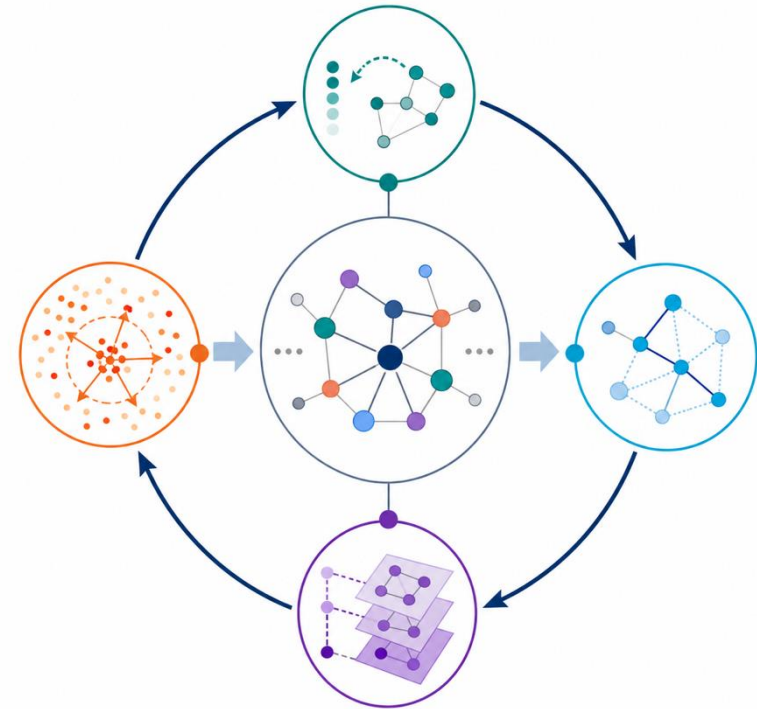
Operator choice and edge sampling

COMBINE DEPTHS

Skip readouts and Jumping Knowledge

MAINTAIN SPREAD

PairNorm and related normalization



Choose an intervention for the failure you measured.

Residual and skip connections

$$H^{(\ell+1)} = H^{(\ell)} + F_{\ell}(H^{(\ell)})$$

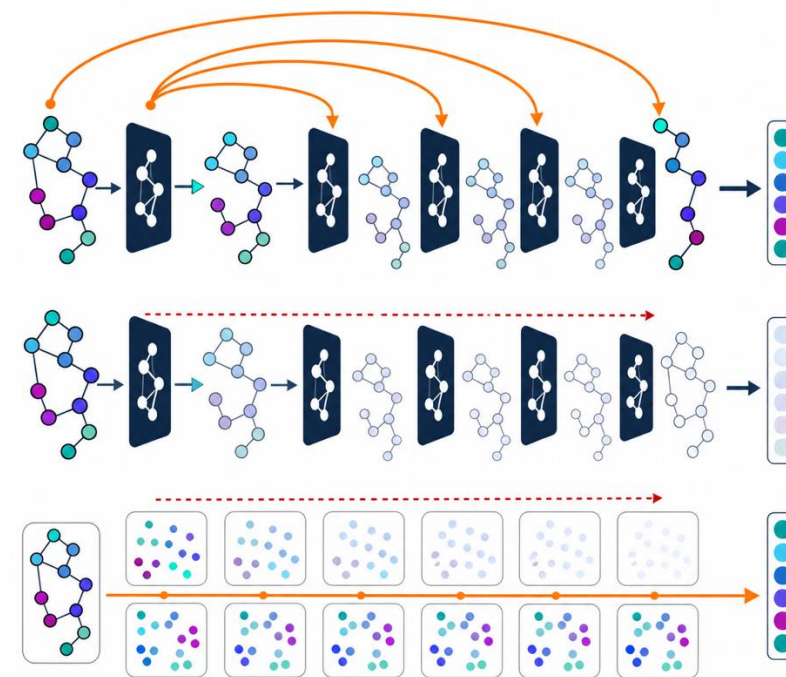
An identity route can help optimization.

A direct skip readout preserves access to early states.

SPECIAL CASE: RESIDUAL GCN (DeepGCNs)

$$H^{(\ell+1)} = H^{(\ell)} + \sigma(\hat{S} H^{(\ell)} W_{\ell})$$

The identity path keeps each node's own state, so smoothing is slowed but not stopped



A residual to the previous layer delays smoothing; a skip to the input removes the collapse.

Source: He et al., Deep Residual Learning for Image Recognition, CVPR 2016;

Li et al., DeepGCNs: Can GCNs Go as Deep as CNNs?, ICCV 2019.

GCNII and repeated input injection

Graph Convolutional Network via **Initial residual** and **Identity mapping**

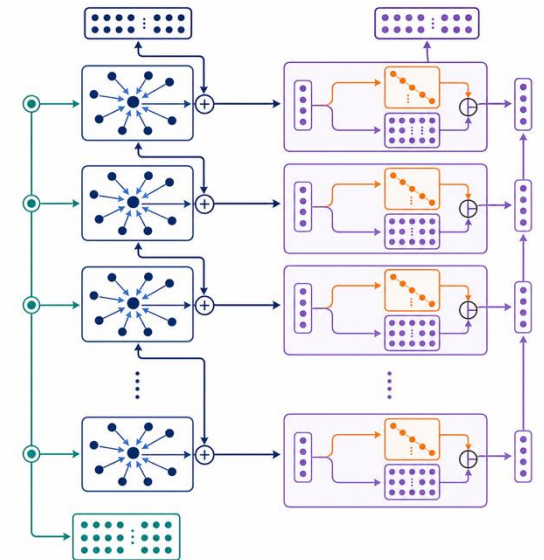
$$H^{(\ell+1)} = \sigma\left(\left[(1 - \alpha_\ell)\hat{S}H^{(\ell)} + \alpha_\ell H^{(0)}\right]\left[(1 - \beta_\ell)\mathbf{I} + \beta_\ell W^{(\ell)}\right]\right)$$

INITIAL RESIDUAL

Every layer receives the initial hidden representation, usually a learned input projection.

IDENTITY COMPONENT

The feature transform mixes an identity route with a learned matrix. Closeness to identity also depends on the learned weights.



Repeated input access is different from only skipping the previous layer.

APPNP and restart

Approximate Personalized Propagation of Neural Predictions

STEP 1: PREDICT

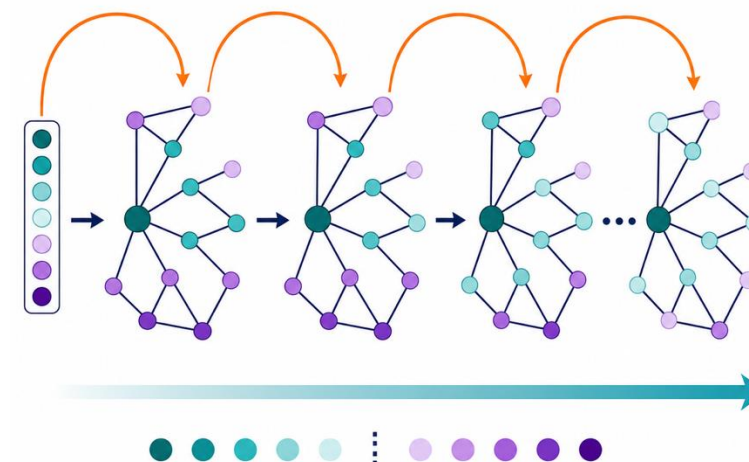
An MLP on node features

$$Z^{(0)} = f_{\theta}(X)$$

STEP 2: SPREAD

Mix with neighbors and keep the initial

$$Z^{(k+1)} = (1 - \alpha)\hat{S}Z^{(k)} + \alpha Z^{(0)}$$



Depth of network and range of propagation are decoupled.

Predict first, then spread; keeping a share of your own prediction stops the collapse.

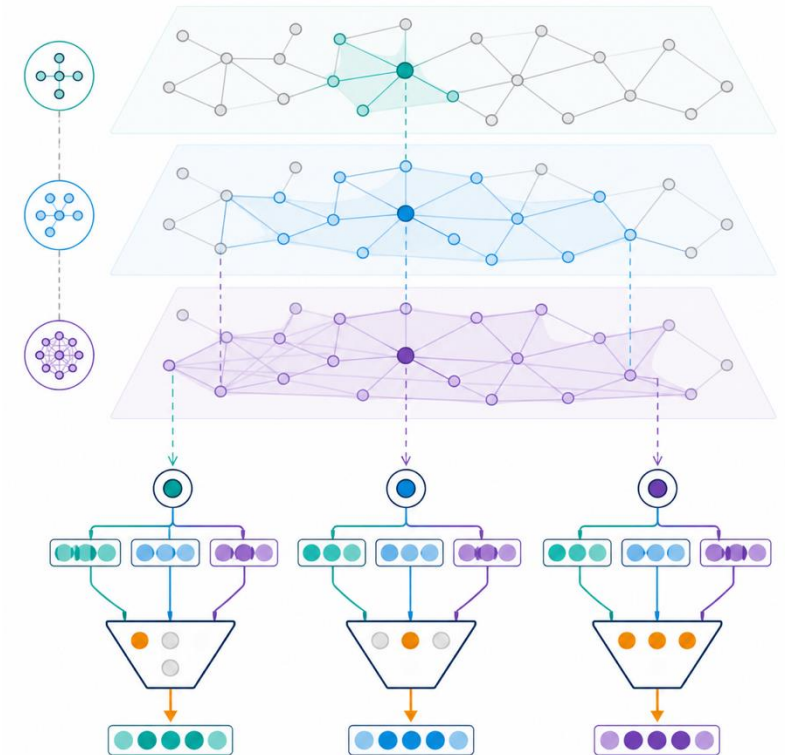
Jumping Knowledge and useful scales

$$h_v^{JK} = \text{COMBINE} \left(h_v^{(1)}, h_v^{(2)}, \dots, h_v^{(K)} \right)$$

Concatenation exposes several depths.

Max pooling or attention can select useful features or scales.

The input projection can be included as an extra skip.



A boundary cell may need finer context than a cell inside a homogeneous region.

PairNorm and embedding spread

STEP 1: CENTER

Subtract the average node so the mean is zero.

$$H_c = H - 1\mu^T, \quad \mu^T = \frac{1^T H}{n}$$

STEP 2: RESCALE

Stretch so the average node length equals s , a chosen constant (often 1).

$$H' = \frac{sH_c}{\sqrt{\frac{1}{n} \sum_i \|h_{c,i}\|_2^2}}$$

EXAMPLE: TWO NODES, ONE FEATURE EACH, $s = 1$

$$[10, 10.2] \rightarrow [-0.1, 0.1] \rightarrow [-1, 1]$$

PairNorm runs after every layer and pushes them back apart to the same fixed spread, so they can never collapse onto one point.

PairNorm keeps the nodes spread out, but spread out is not the same as separated by class.

DropEdge and stochastic propagation

Sample an undirected edge mask during training.

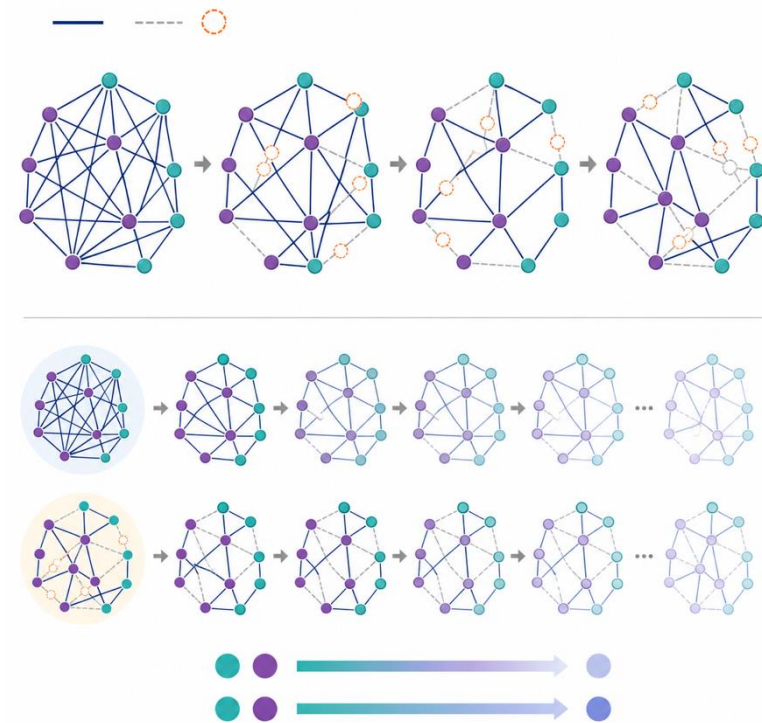
$$A^{(t)} = A \odot M^{(t)}$$

Drop both directions of each undirected edge.

Recompute degrees and normalization for each mask.

Retain the intended self-loops.

Check that useful long-range paths survive.



Edge removal can slow mixing, but its effect is not universally monotone.

Mitigation methods compared

Method	Intervention	How it helps	Main caveat
Tune depth	Limit propagation steps	Avoids unnecessary mixing	May miss long-range signals
Residual / skip	Identity or earlier states	Preserves accessible contrast	Deep branch may still smooth
Jumping Knowledge	Aggregate across layers	Selects a useful node scale	Extra memory / selector
APPNP	Restart to initial logits	Retains input contributions	Restart rate controls range
GCNII	Initial residual + identity	Supports deep stacks	More depth hyperparameters
PairNorm	Center + rescale	Fixes global squared spread	Spread need not preserve labels
DropEdge	Random edge masks	Can delay mixing and regularize	May remove useful paths

These are mechanisms to test, not guarantees that deeper models will perform better.

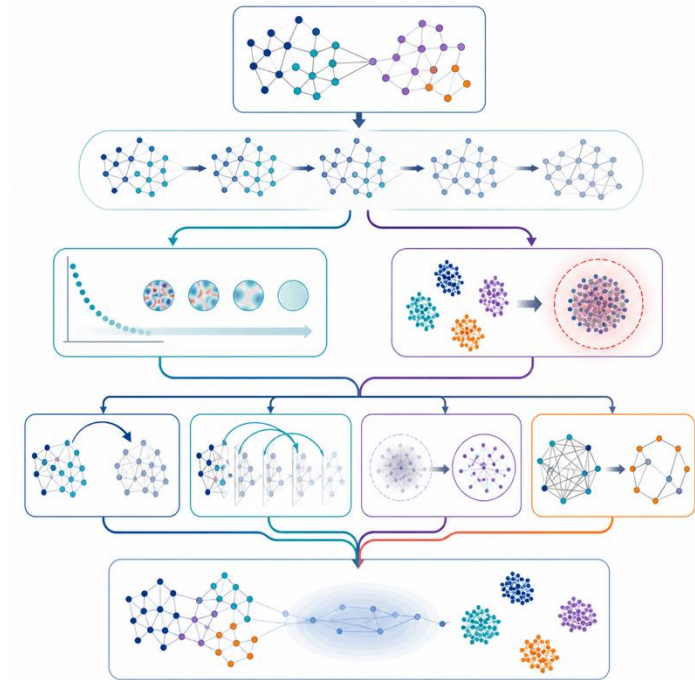
TAKEAWAYS

Useful context, distinct representations

Match the diagnostic to the propagation operator

Check task separation as well as feature spread

Test one intervention at a time



Choose graph context at a scale that helps the target.