Deep Learning (1470)

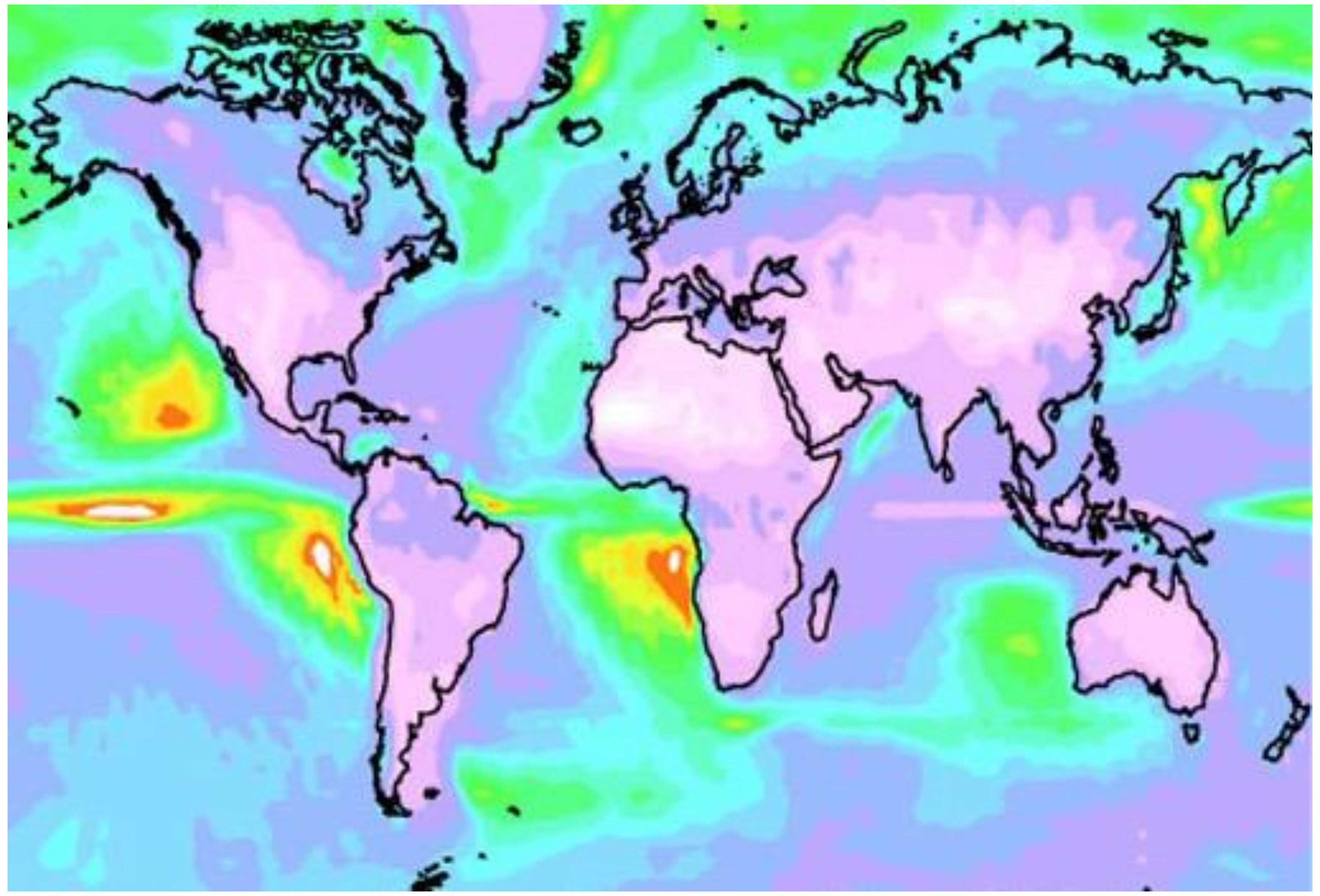
Randall Balestriero

Class 18: Geometric Deep Learning

Recap!

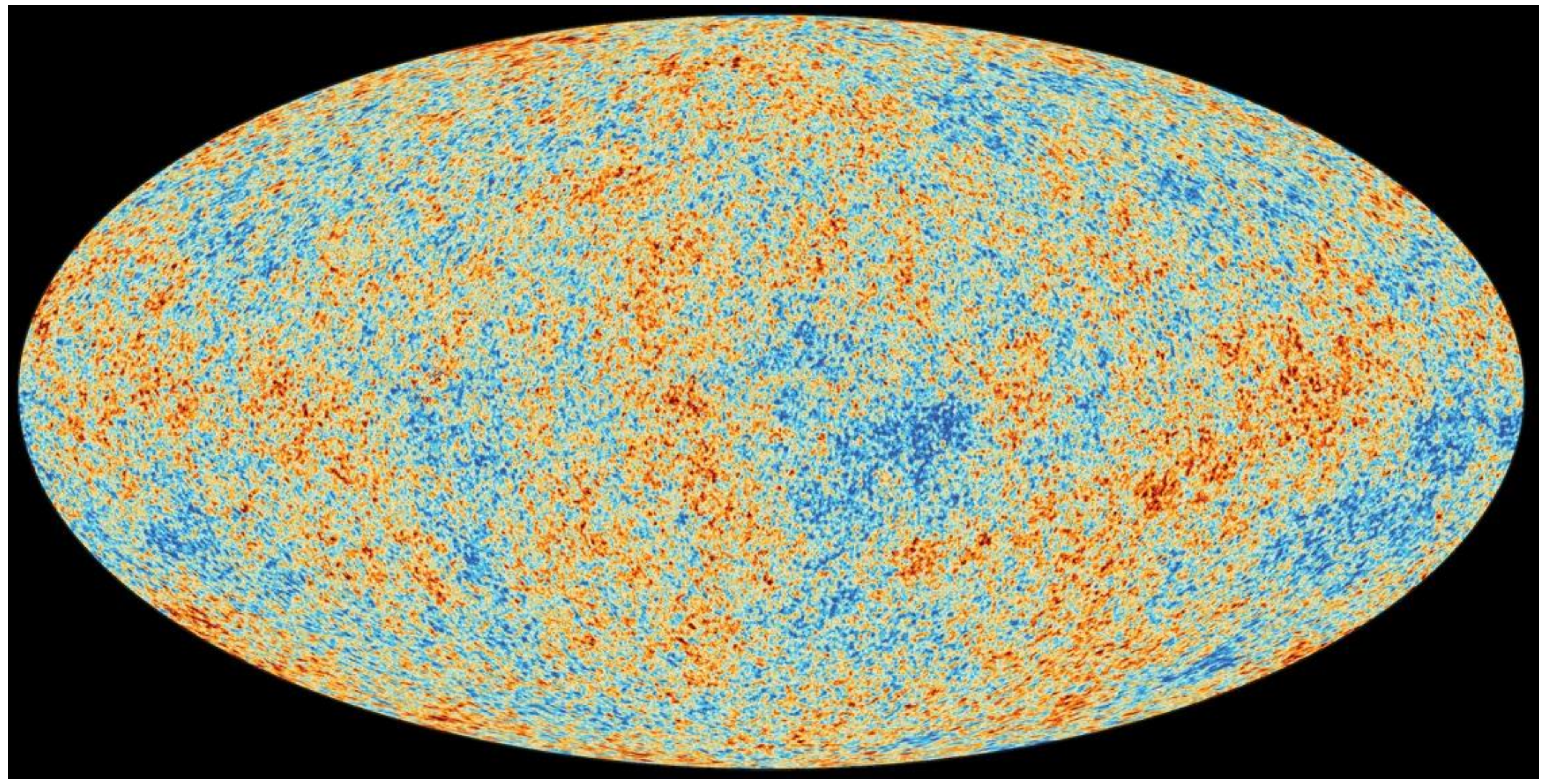
Goal: Take as input data from satellites, predict carbon dioxide

Idea 1: Take all of the data as input, make it look like an image, use CNNs to learn to output carbon monoxide amounts

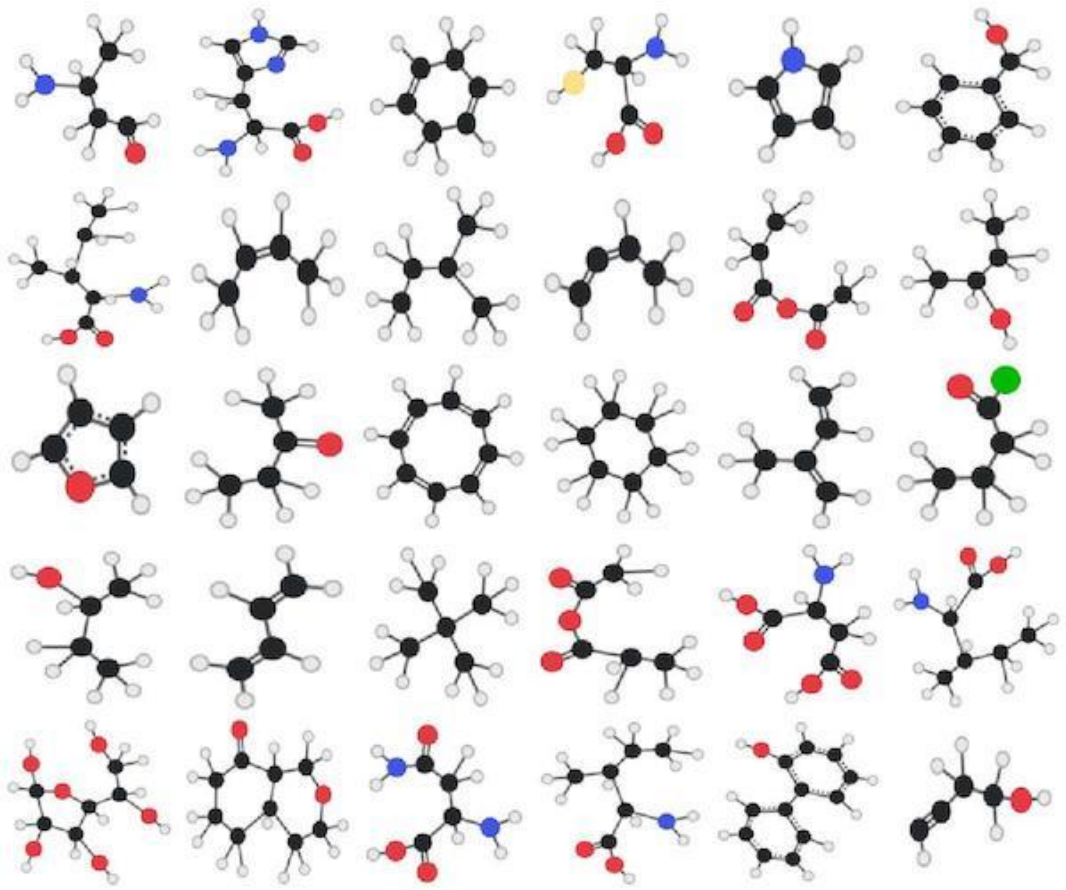


https://newton-climate.github.io/#research

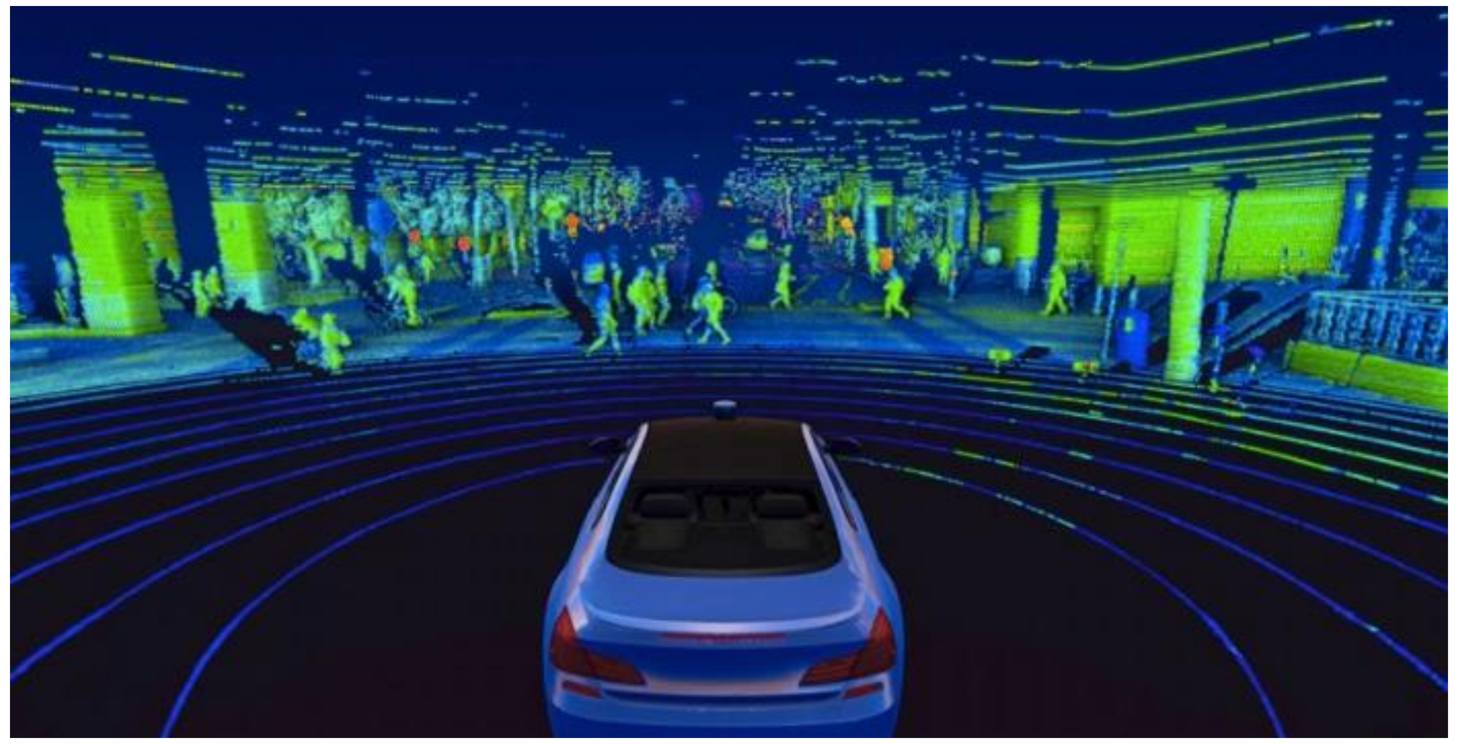
Other Types of Structured Data Types



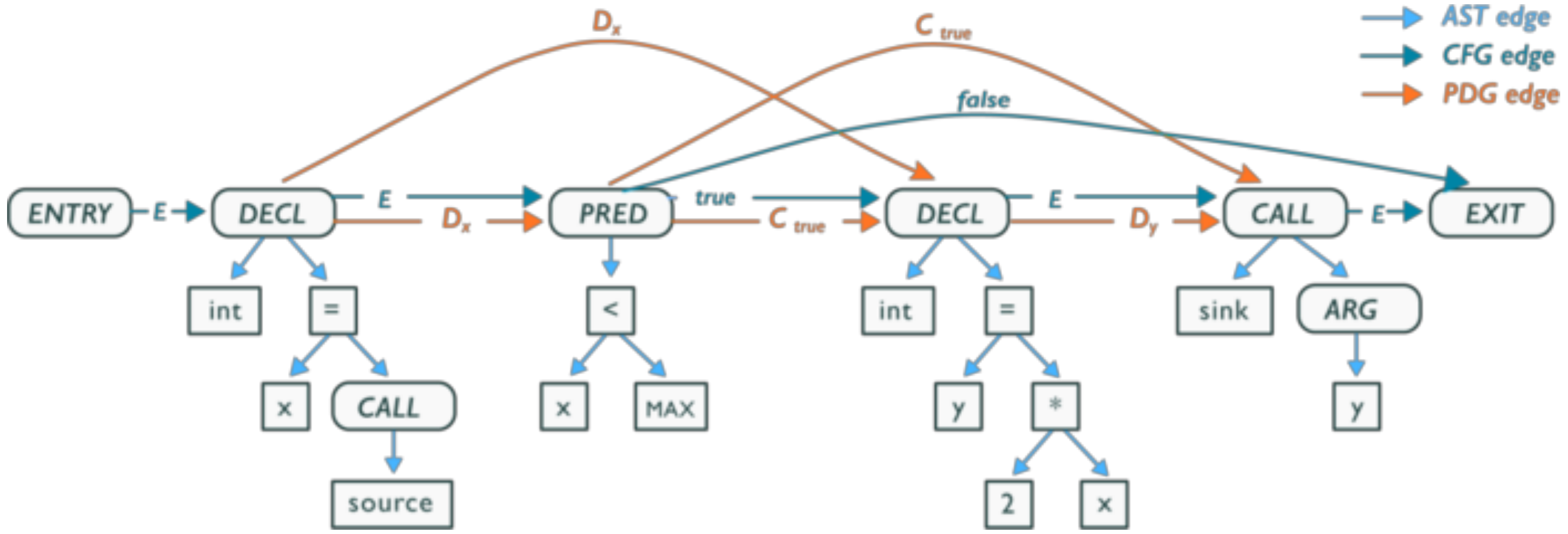
Microwave Background Radiation



Molecule Data



Point Clouds (from LIDAR)

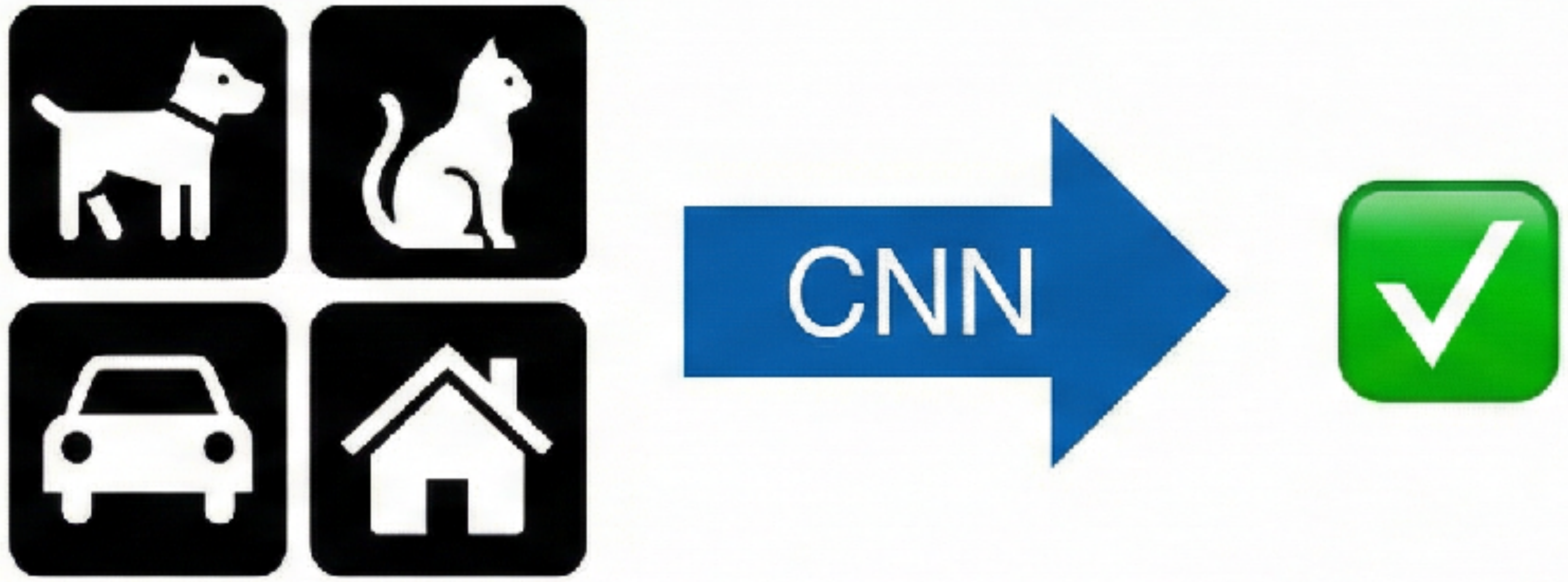


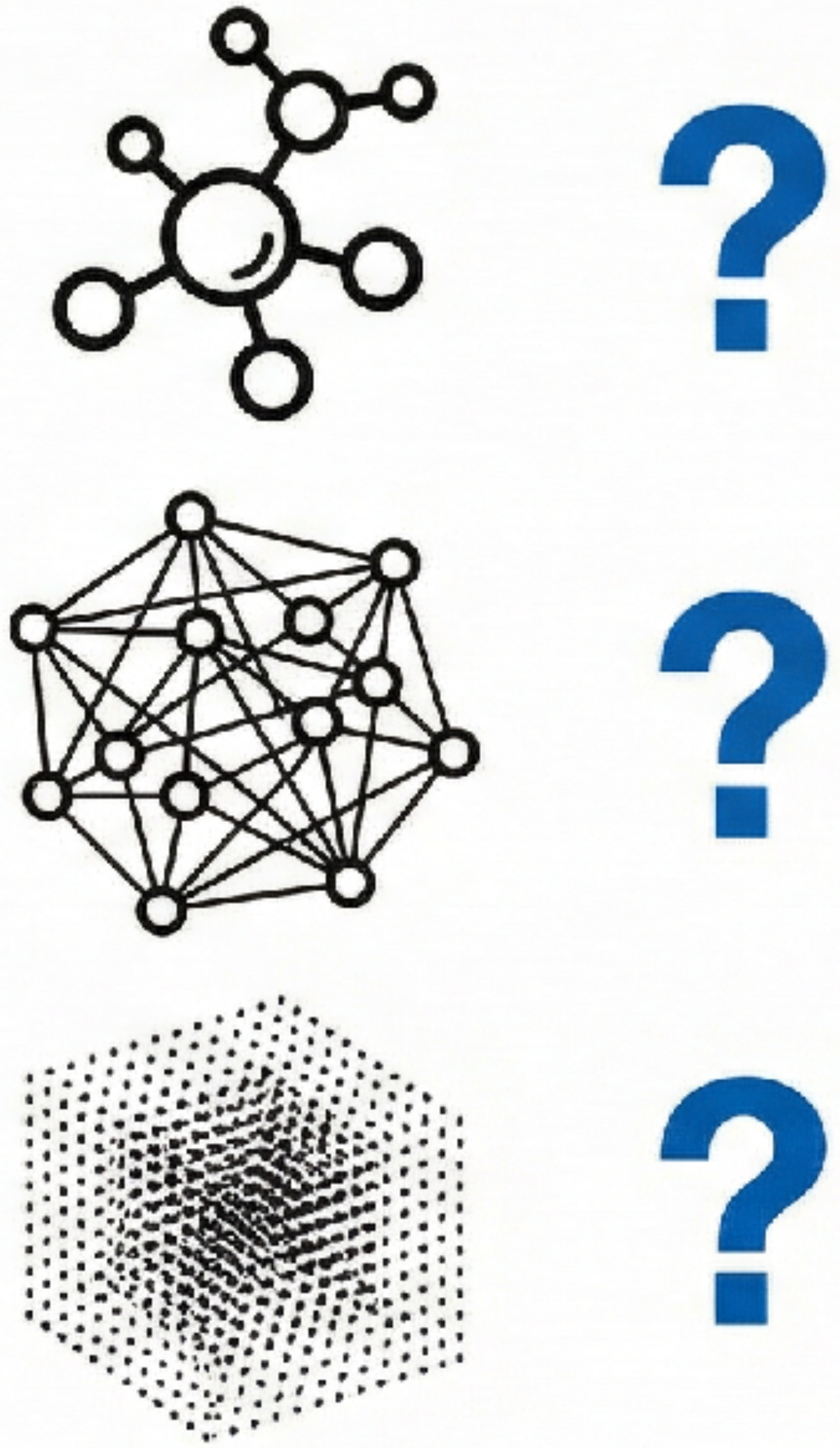
Code Graphs

THE BIG PICTURE

Works

Challenge





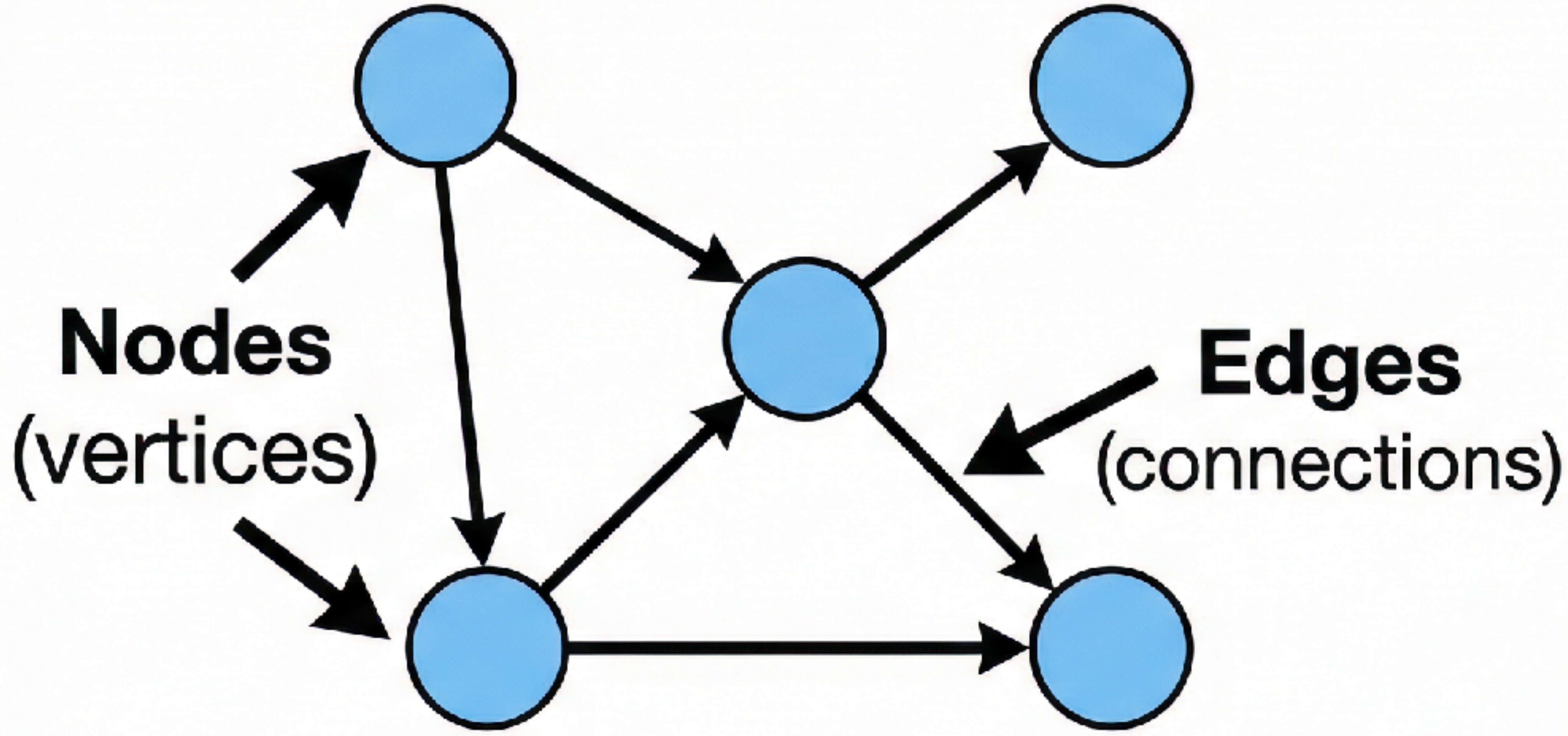
Data with structure needs special networks

Geometric Deep Learning

- We turned to convolutions for image data to give our networks "spatial reasoning"
- By Explicitly modelling the relationship of our data, we can achieve better results
- Geometric Deep Learning is the subfield of DL dedicated to learning representations of *structured* data.

GRAPHS 101: A QUICK INTRO

What is a Graph?



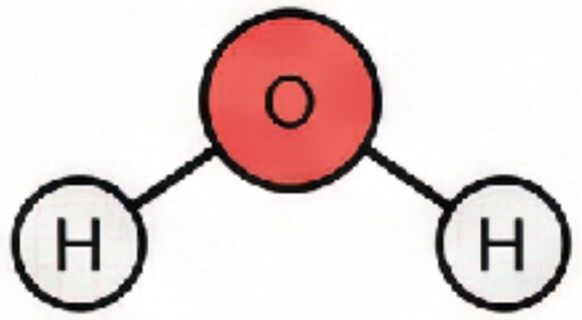
Like a map of relationships!

Examples

Small social network:

Alice - Bob - Carol

Small molecule:



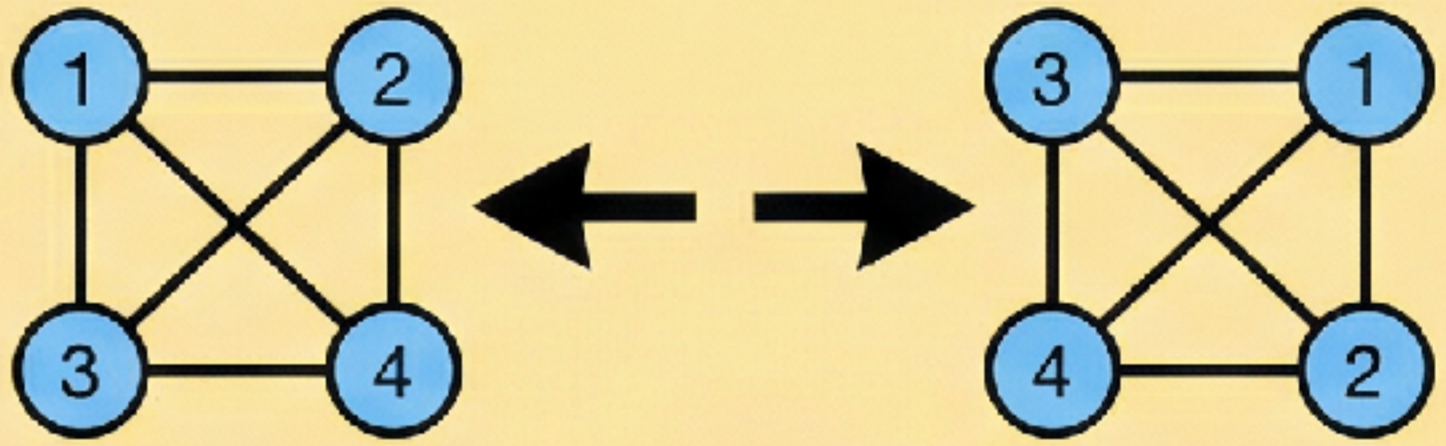
H - O - H

Small citation:

Paper A cites Paper B

Key Insight

Graphs have NO fixed ordering of nodes!



Both represent the SAME graph

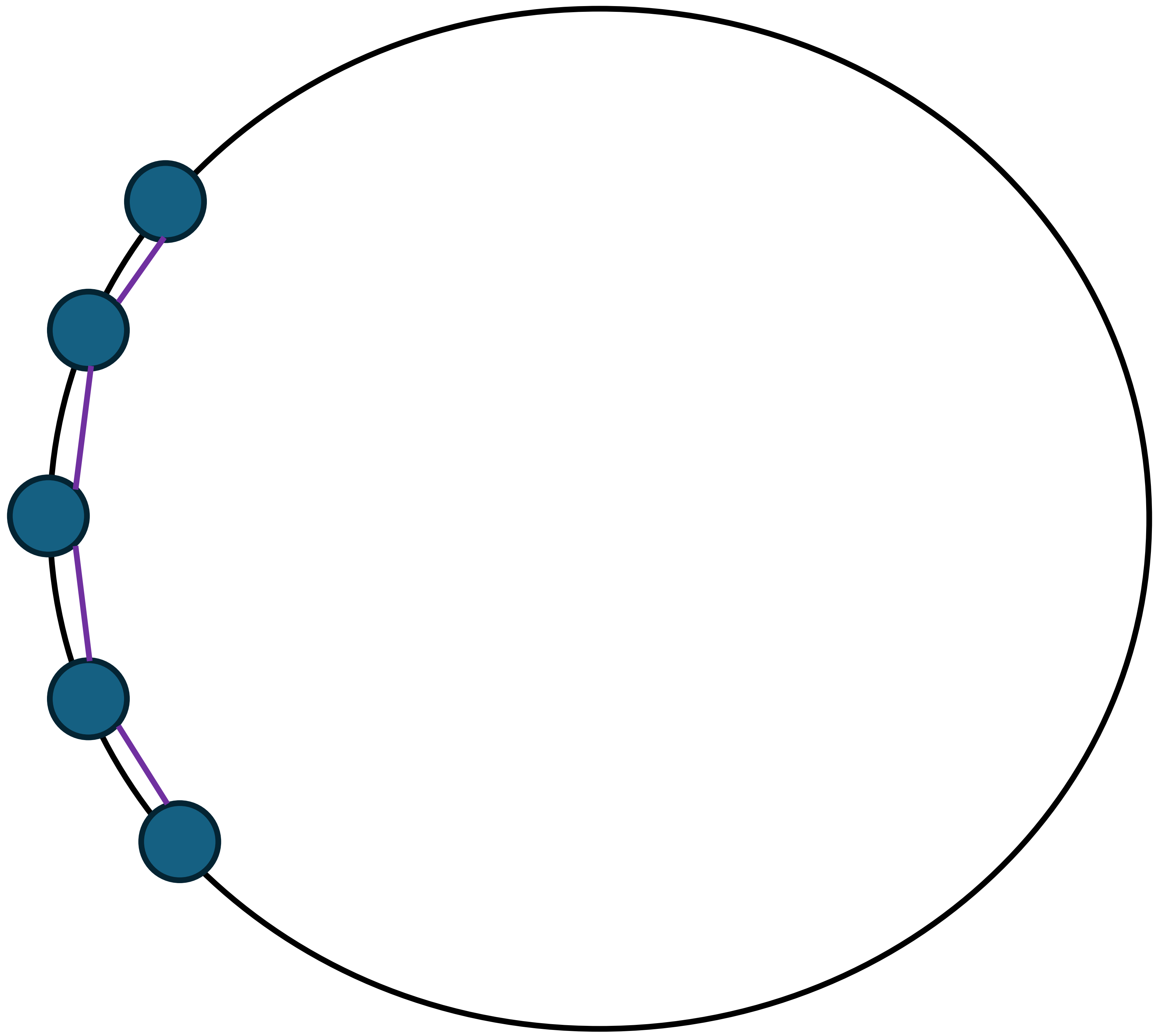
This is **PERMUTATION INVARIANCE**

Extending Convolutions to Manifolds

A circle is a manifold in 2D
(like a sphere or taurus in 3D)

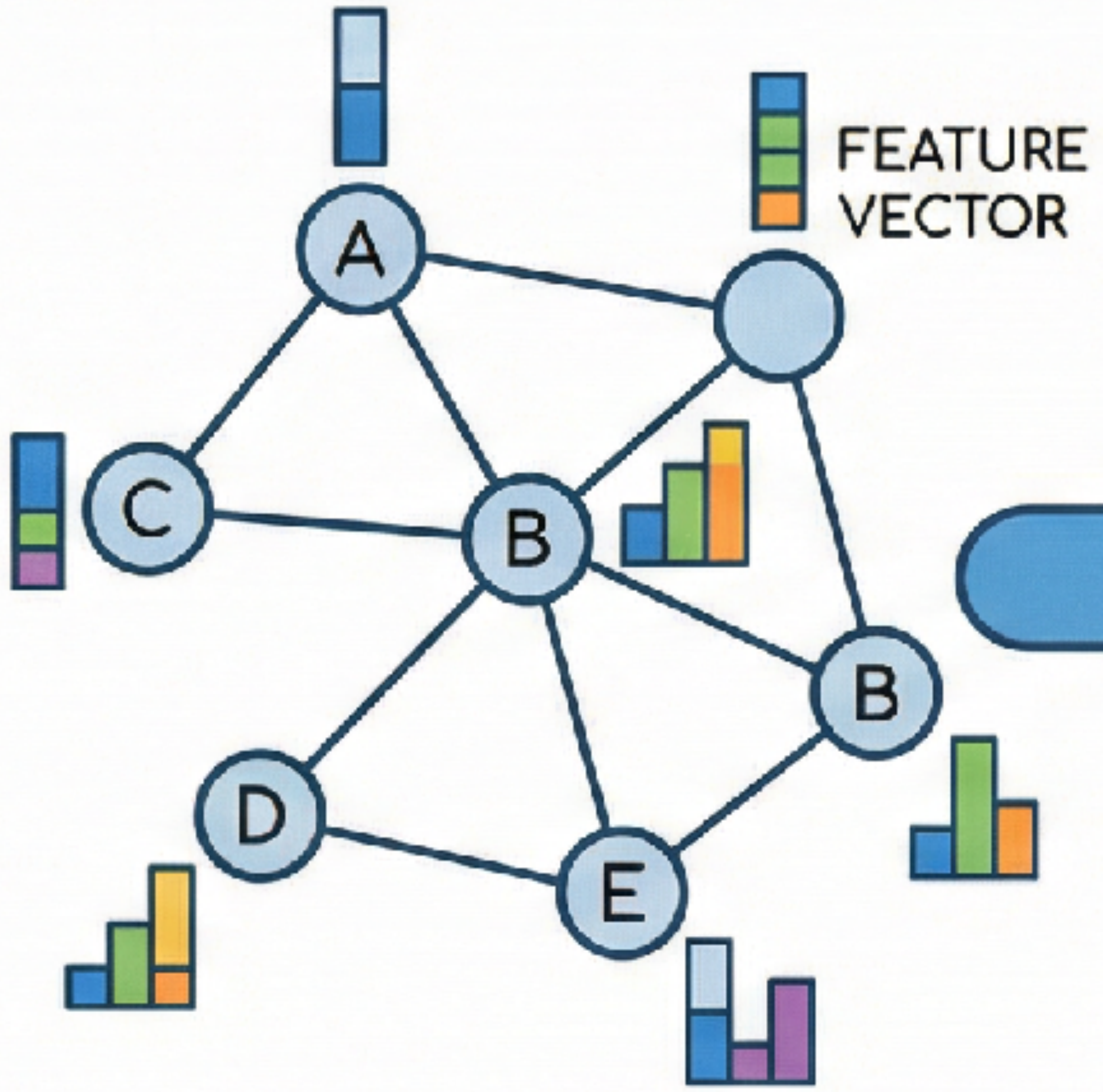
For every point on the manifold, connect it to “close” points

What data structure does this lead to?



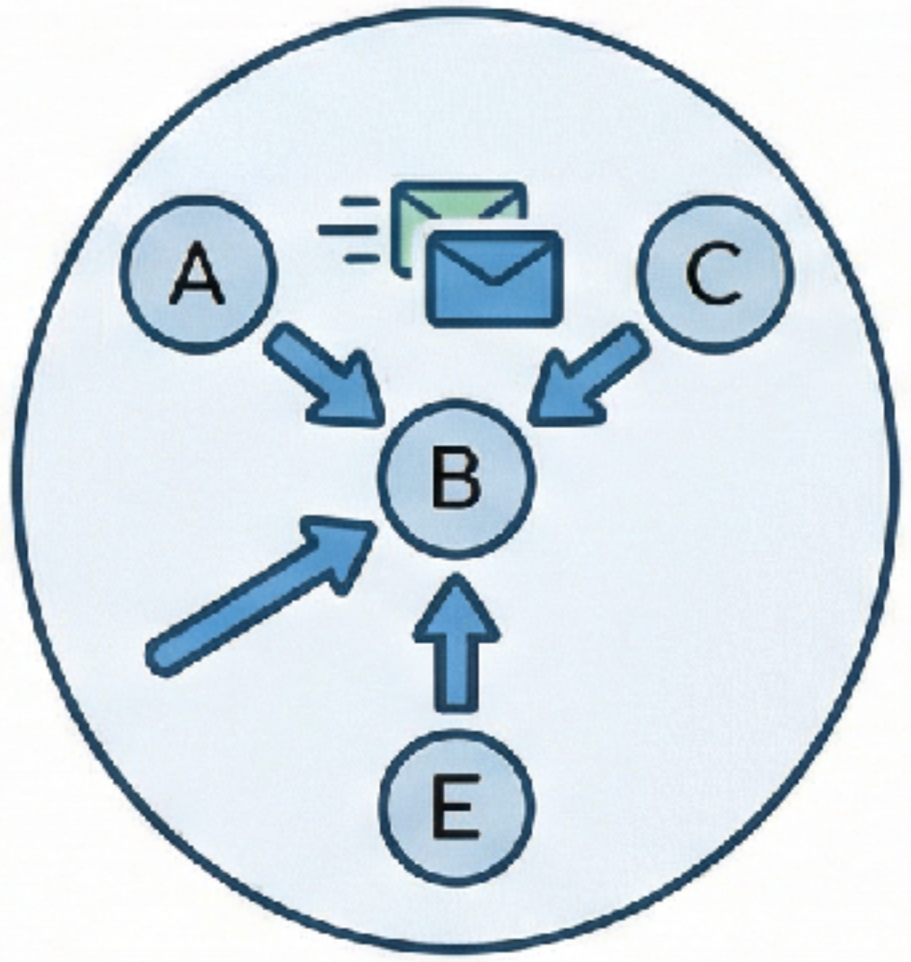
HOW DO GRAPH NEURAL NETWORKS WORK?

STEP 1 - START



Each node has features (like a profile)

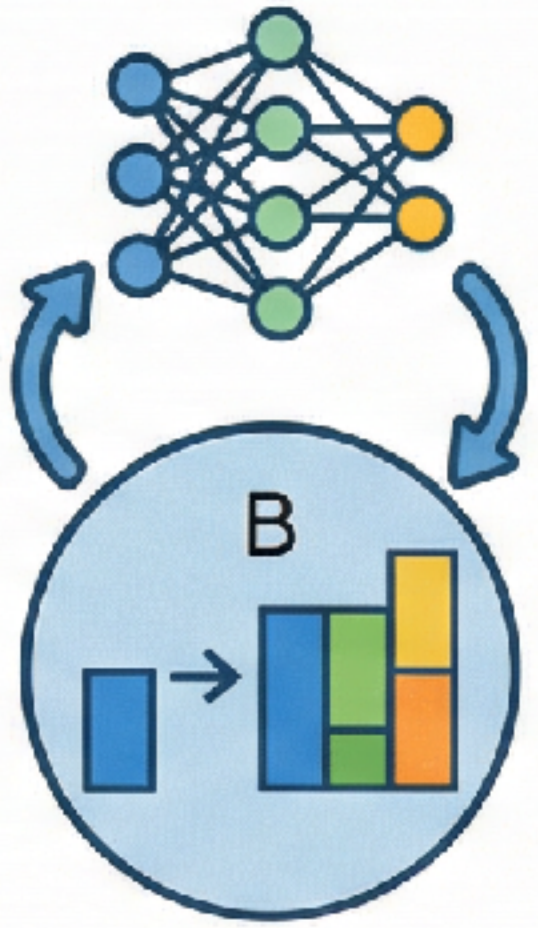
STEP 2 - TALK TO NEIGHBORS



Each node collects info from neighbors

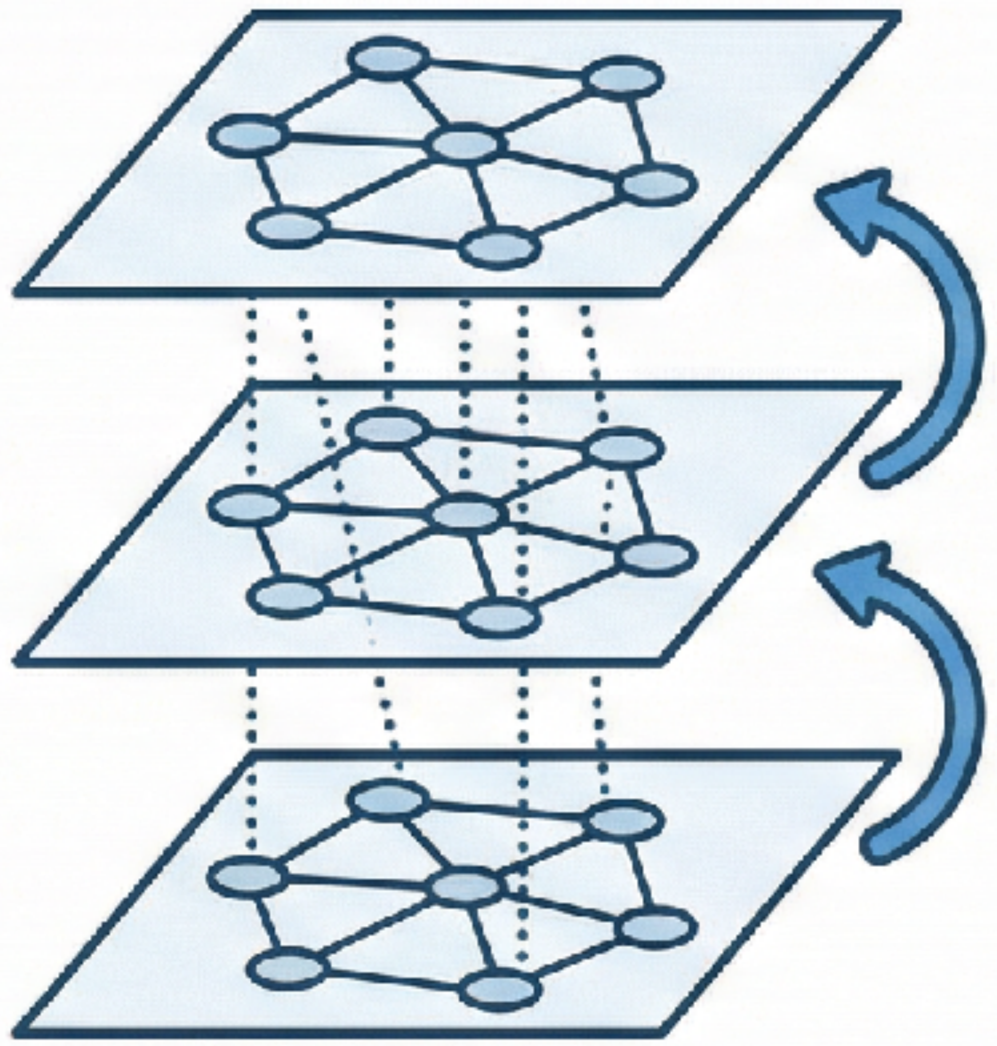
Like asking friends for advice!

STEP 3 - UPDATE



Combine and update your own features

STEP 4 - REPEAT



Stack layers = see farther in graph

After K layers, each node knows about nodes K hops away!

Node Prediction

Each node v has a learned representation z_v, we can learn a fully-connected layer to go from features z_v to output

$$f(z_v) = \sigma(W z_v)$$

Link Prediction

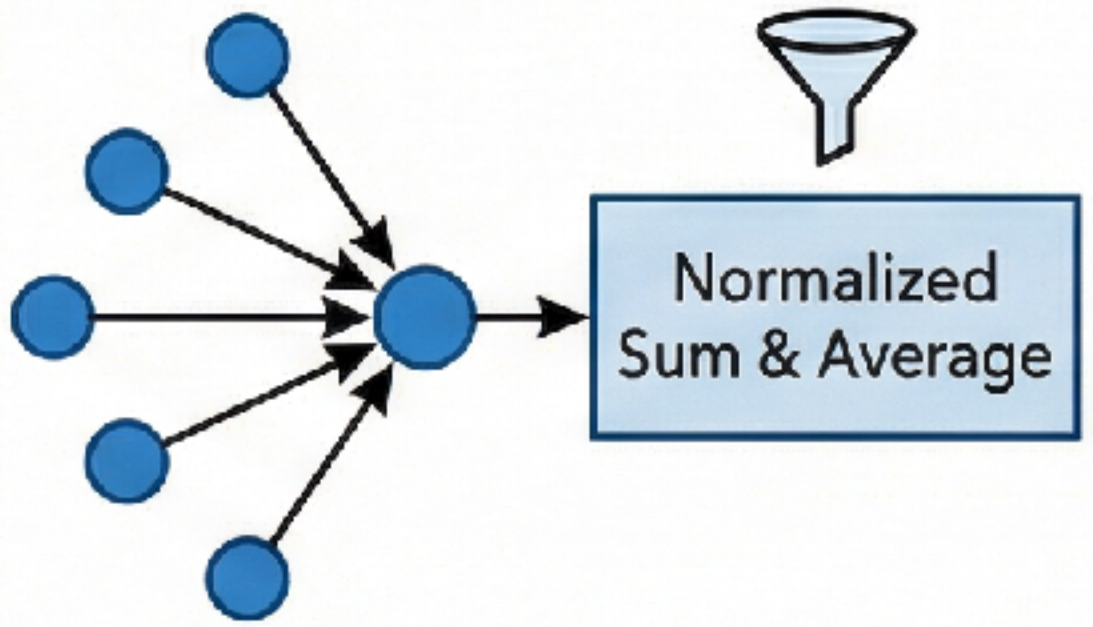
Learn a function that takes two nodes as input and predicts the presence (or absence) of an edge

$$f(z_v, z_u) = \sigma(W(z_v \odot z_u))$$

Element-wise product

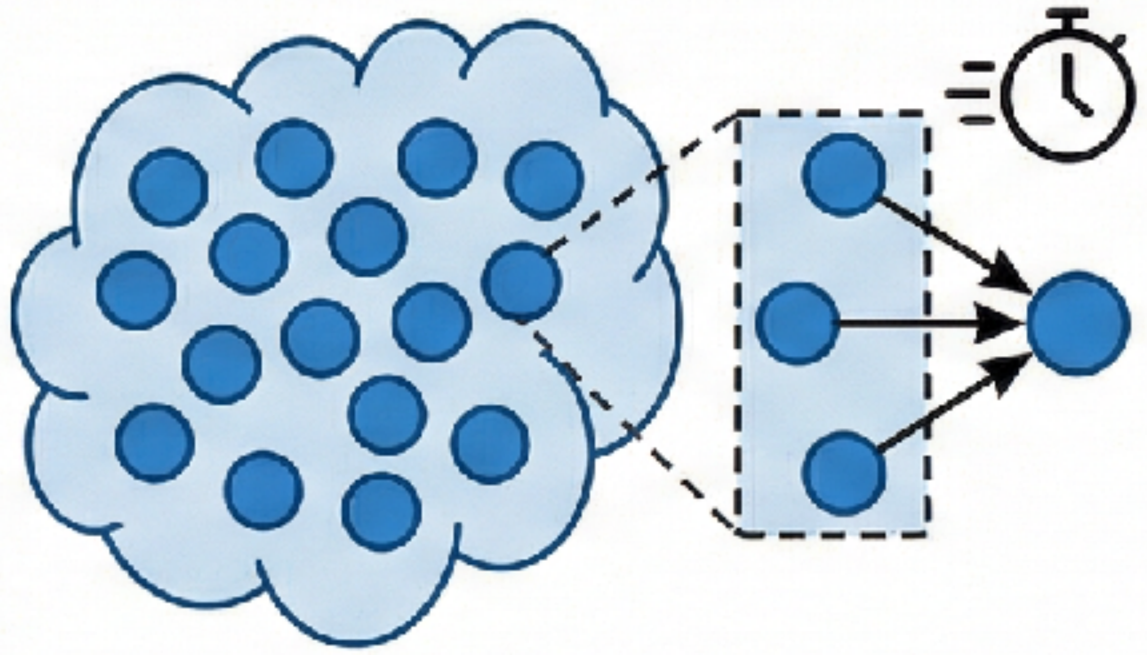
POPULAR GNN ARCHITECTURES

GCN (2017)



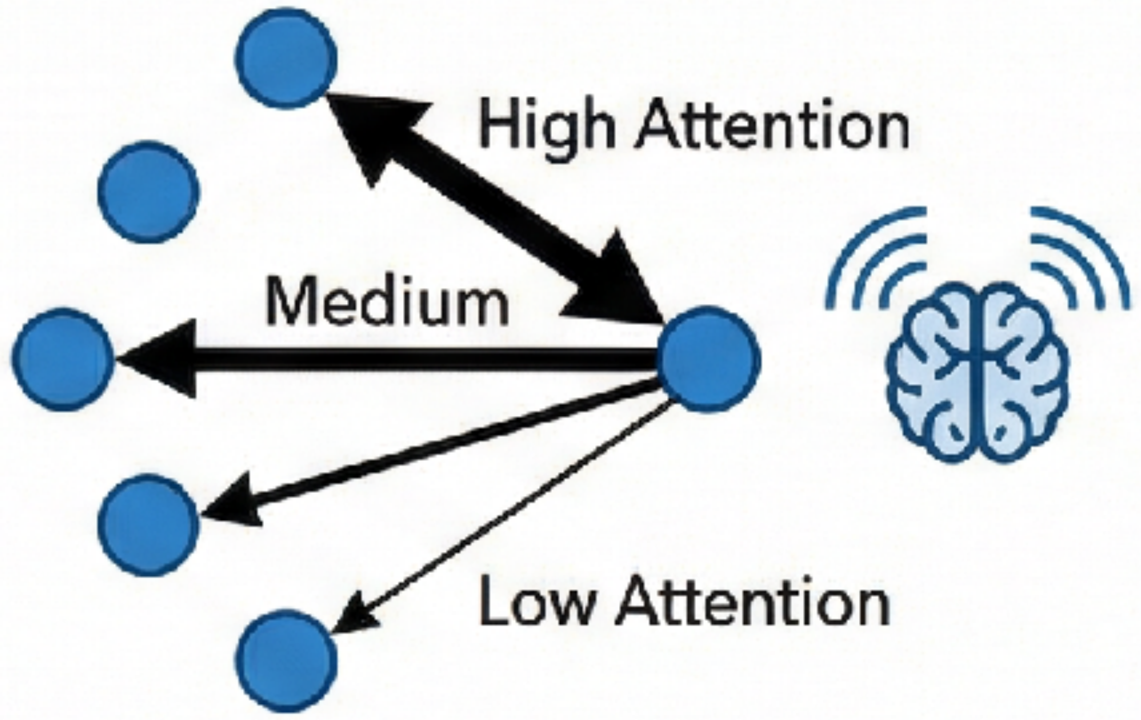
- Average neighbor features
- Simple and effective
- **Used for:** Node classification

GraphSAGE



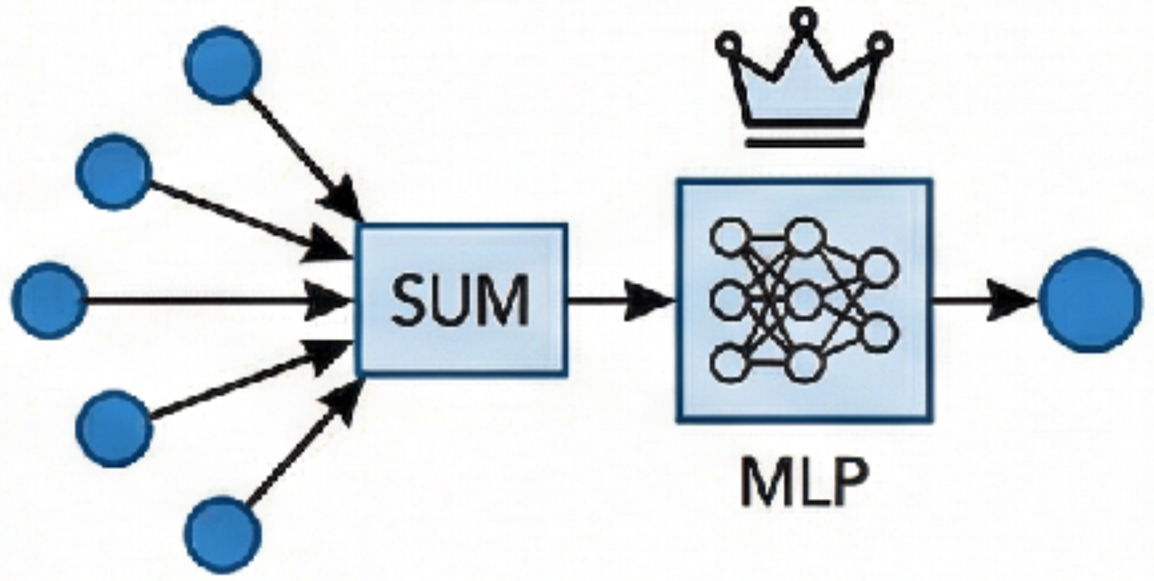
- Sample neighbors (scalable!)
- Works on huge graphs
- **Used for:** Pinterest, Uber

GAT



- Learn which neighbors matter more
- Attention mechanism
- **Used for:** When some edges more important

GIN



- Most expressive architecture
- Can distinguish more graphs
- **Used for:** Graph classification

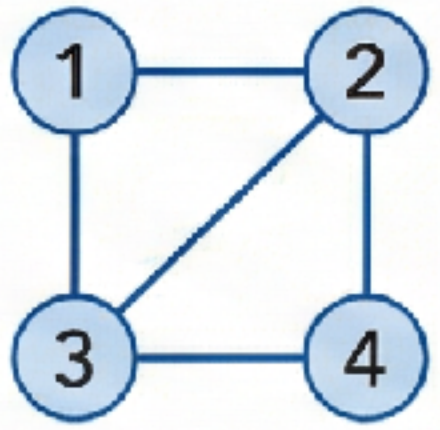
All follow the same message passing recipe!

GCN: THE CORE EQUATION

$$\mathbf{H}' = \sigma\left(\mathbf{D}^{-1/2}\,\mathbf{A}\,\mathbf{D}^{-1/2}\,\mathbf{H}\,\mathbf{W}\right)$$

A = Adjacency Matrix

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$



Who connects to whom

D = Degree Matrix

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

How many neighbors each node has

H, W = Features and Weights

H (Features) **W** (Weights)

$$\underset{4 \times F}{\begin{bmatrix} 4 & 1 & 3 \\ 1 & 2 & 2 \\ \vdots & \vdots & \vdots \\ 4 & \vdots & 3 \end{bmatrix}}_{N \times F} \rightarrow \underset{F \times F'}{\begin{bmatrix} 3 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 4 & 3 \end{bmatrix}}_{F \times F'}$$

Learnable transformation

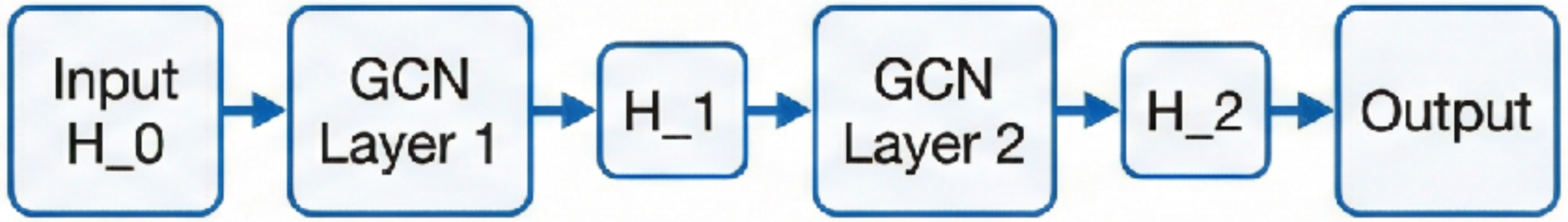
Aggregate neighbors –> Normalize by degree –> Transform –> Activate

GCN: IMPLEMENTATION DETAILS

PyTorch Code Structure

```
class GCNLayer:
  - W: weight matrix (F_in, F_out)
  - forward(H, A):
      - A_hat = A + I (add self-loops)
      - D_hat = degree(A_hat)
      - norm = D^(-1/2) A_hat D^(-1/2)
      - return ReLU(norm @ H @ W)
```

Layer Stacking



2-3 layers typical (avoid over-smoothing)

Key Implementation Choices

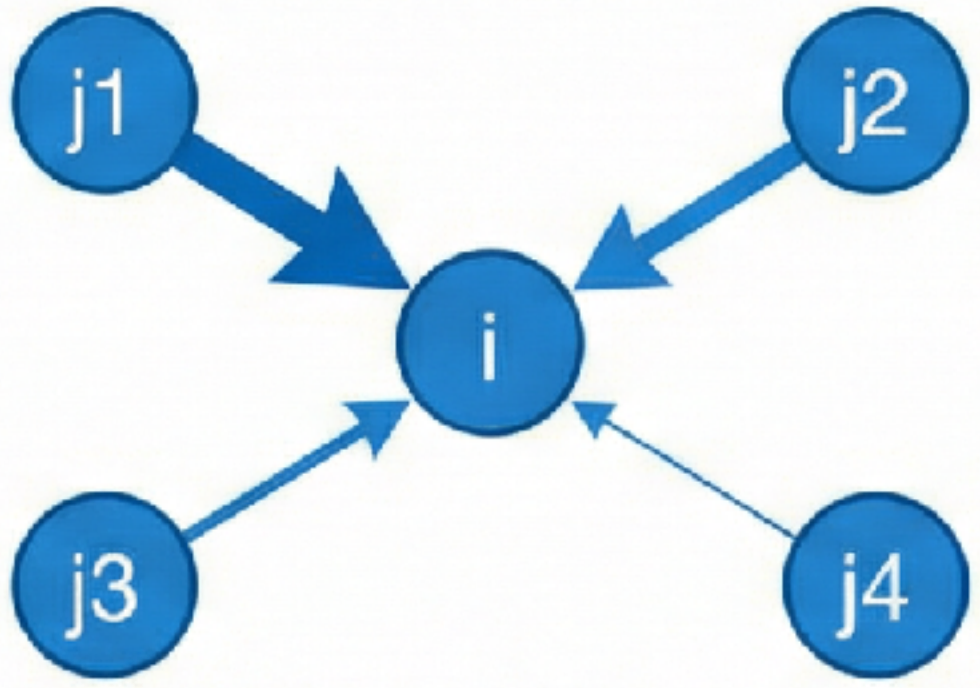
Activation
ReLU, LeakyReLU, none on final

Dropout
Between layers for regularization

Residual
H’ = GCN(H) + H helps with depth

GAT: ATTENTION MECHANISM

The Key Idea
Not all neighbors are equally important!



Learn attention weights for each edge

The Attention Equation

Step 1: Project features

$$\mathbf{z}_i = \mathbf{W}\mathbf{h}_i$$

(linear transformation)

Step 2: Compute attention scores

$$e_{ij} = \mathrm{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{z}_i \,\|\, \mathbf{z}_j\right]\right)$$

Concatenate and score with vector **a**

Step 3: Normalize with softmax

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

Weights sum to 1 over neighbors

Final aggregation

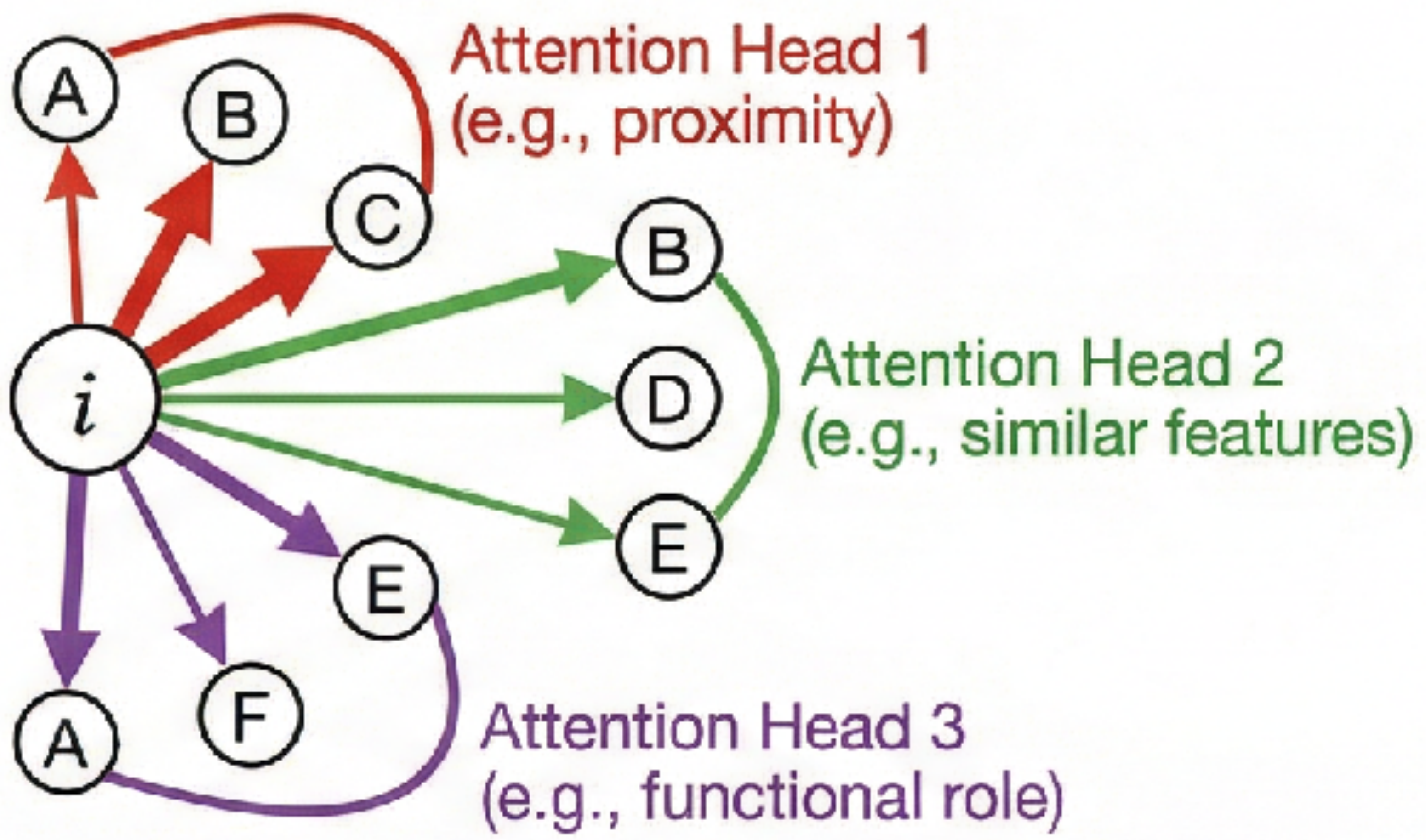
$$\mathbf{h}_i' = \sigma\left(\sum_j \alpha_{ij}\mathbf{W}\mathbf{h}_j\right)$$

Weighted sum of neighbor features

GAT: MULTI-HEAD ATTENTION

Why Multiple Heads?

Different heads learn different relationship types!



Like having multiple perspectives

Multi-Head Formulation

$$\mathbf{h}_i' = \text{CONCAT}(\text{ head}_1, \text{head}_2, ..., \text{head}_K) \text{ or MEAN(heads)}$$

$$\text{Each head}_k = \sigma\left(\sum_j \alpha_{ij}^k \mathbf{W}^k \mathbf{h}_j\right)$$

K independent attention mechanisms

Concatenate (middle layers) or Average (final layer)

Implementation Summary

Learnable Parameters

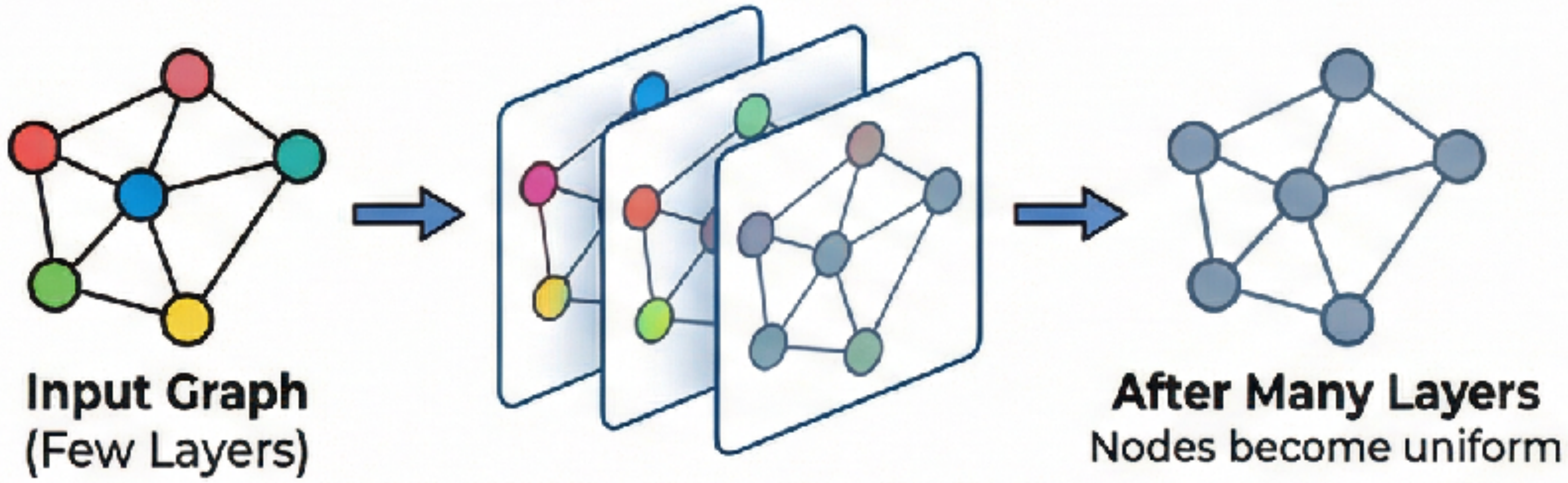
- W: feature projection (F x F')
- a: attention vector ($2F'$ x 1)
- Per head: W^k and a^k

Architecture Choices

- Heads K: typically 8
- Hidden dim F': 8 per head
- Dropout on attention weights
- Add self-loops for self-attention

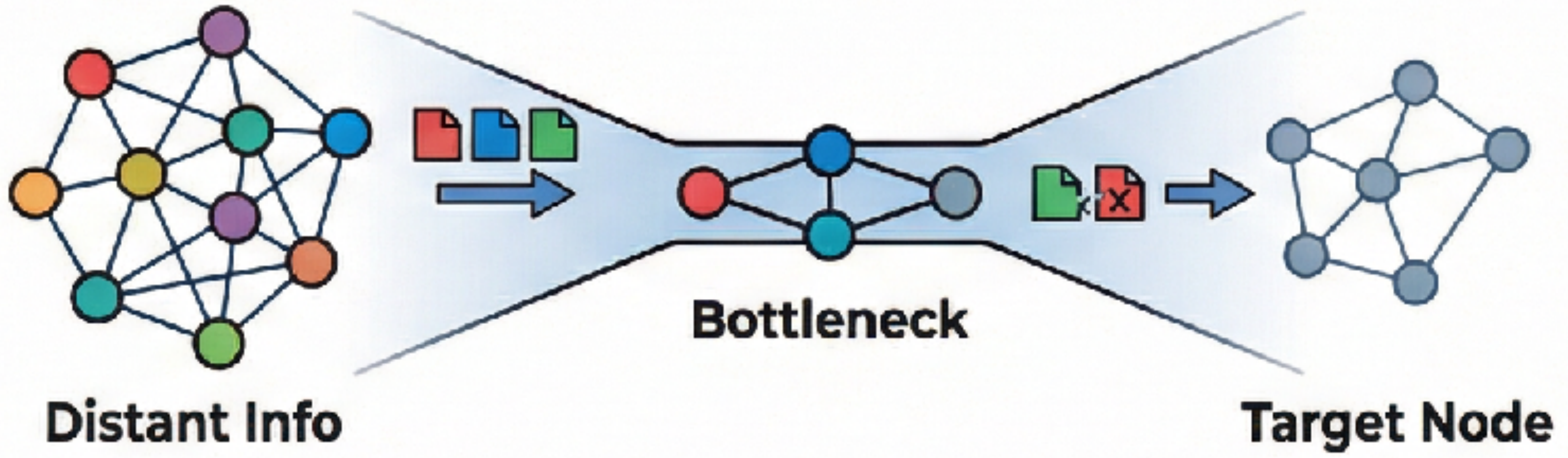
CHALLENGES IN GEOMETRIC DL

OVER-SMOOTHING



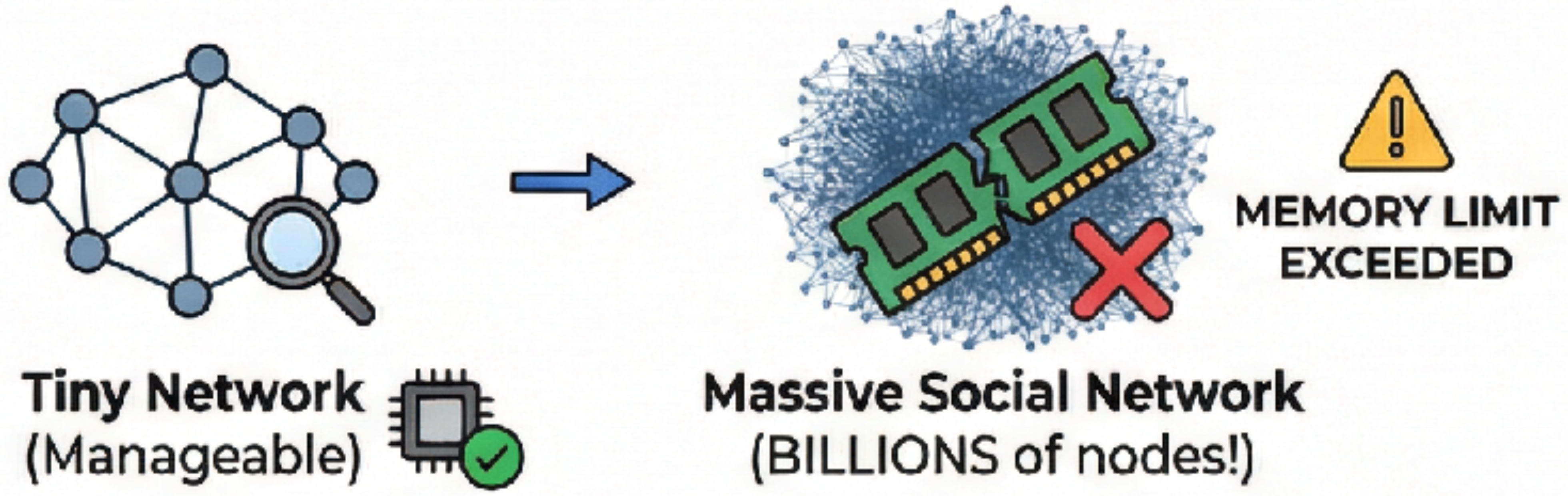
- Too many layers = all nodes look the same!
- Like a game of telephone - info gets blurry

OVER-SQUASHING



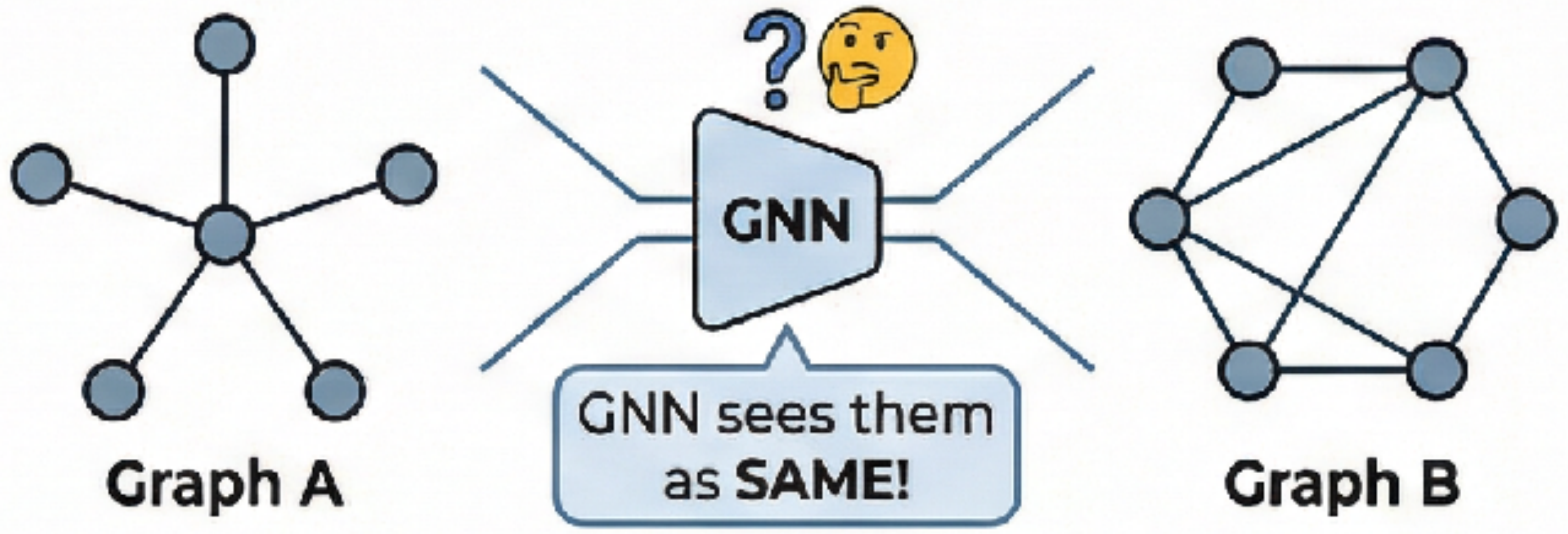
- Hard to send info through narrow graph paths
- Distant nodes can't communicate well

SCALABILITY



- Social networks have BILLIONS of nodes
- Can't fit in memory!

EXPRESSIVITY



- Some graphs GNNs cannot tell apart!
- Fundamental theoretical limits

Active research areas - lots of room for new ideas!

See you on Friday!