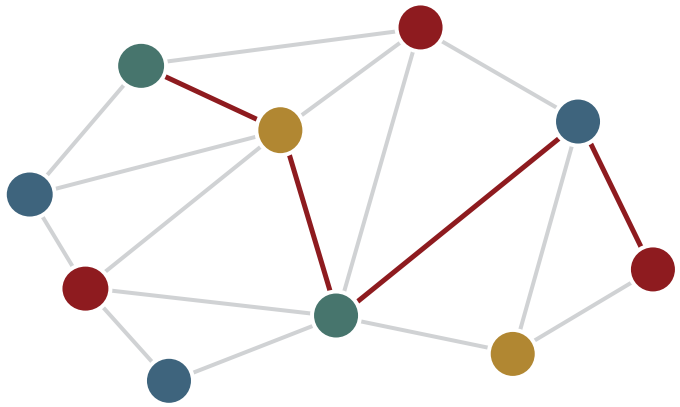
Foundations of Graph Signal Processing and Graph Machine Learning



Zhiyang Wang · Electrical & Systems Engineering · WashU

An introduction to the course and the field

About Me - Zhiyang

- Signal processing, machine learning -- develop scalable, stable learning methods over structured data
- Applications in medical data analysis, wireless communication networks, multi-agent systems, reasoning over graphs, etc…
- Hope the class provides you new tools and perspectives when thinking about your own research interests
- zhiyangwang.com
- wzhiyang@wustl.edu -- best way to find me **McKelvey Hall 2052**
- Announcements, slides and other materials on **Canvas**
- Office hour: Poll link on Canvas

What we will do today

01	**Why graphs?**	A shared language for irregular, relational, and networked data.
02	**What is graph signal processing?**	Classical signal ideas—operators, frequency, filtering—without a regular line or grid.
03	**How will this course work?**	A low-overhead path from foundations to reading and creating research. Workload depends on you.

How the semester is organized

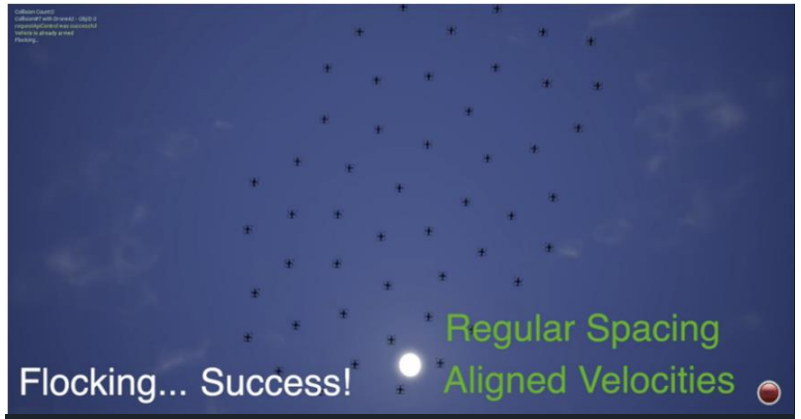
14 weeks · 5 focused Colabs · one paper talk · one semester project

WEEK	TOPIC
1	Graph signals, Laplacian, and GFT
2	Graph convolutions: locality, invariance, stability
3	Filter design, types of graph learning problems
4	Random graphs, clustering, structure inference
5	Message-passing GNNs: ChebNet, GraphSAGE
6	Oversmoothing and oversquashing
7	Graphons and size transfer

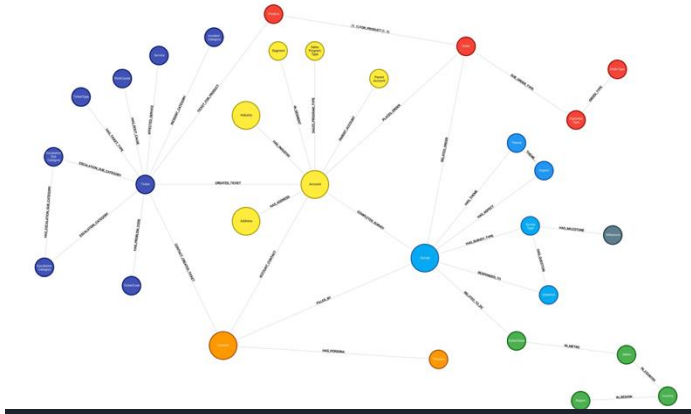
WEEK	TOPIC
8	Manifolds and continuous graph limits
9	Graph attention and transformers
10	Graph generation: VAEs, diffusion, flows
11	Learning graph structure from signals
12	Dynamic and temporal graph learning
13	Applications: molecules, materials, brains
14	Project presentations and research directions

Graphs are useful when relationships matter

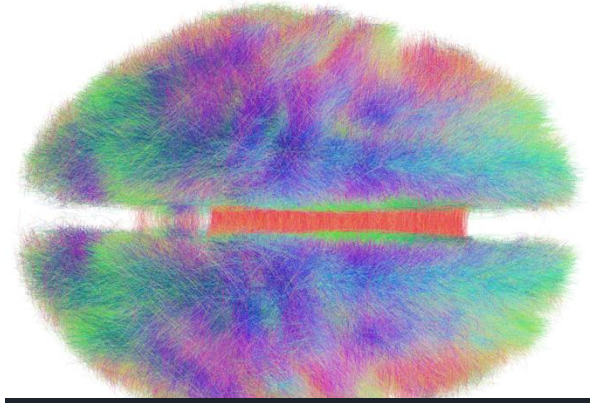
INFRASTRUCTURE · roads shape mobility



COMMUNICATION · links carry information

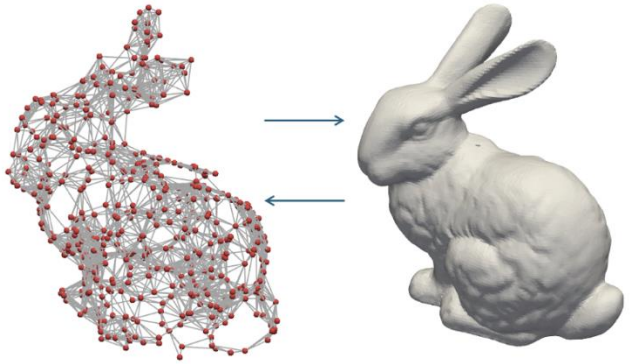


KNOWLEDGE GRAPH · inference



BRAIN · function prediction

MOLECULES · structure generation



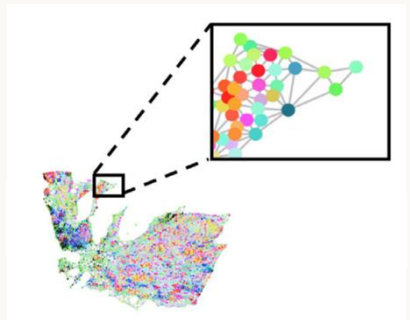
3D SHAPE · recognition and segmentation

The graph records both the objects and the relationships we care about.

Image credits: Jean Rebiffé / Wikimedia Commons · NASA Scientific Visualization Studio · Andreashorn / Wikimedia Commons · Neo4j

A graph model needs four ingredients

system → nodes and edges → graph signal → practical task

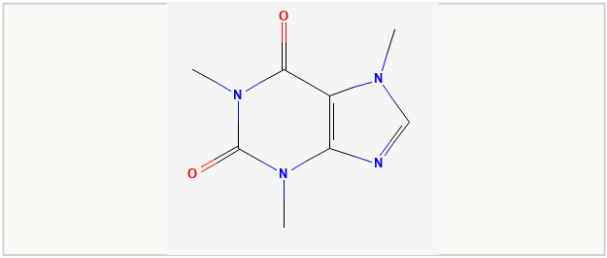


PATHOLOGY SLIDES

cells ↔ proximity

SIGNAL cell type and morphology

TASK predict HER2 status from H&E

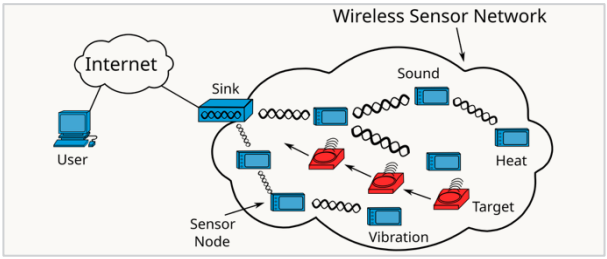


MOLECULES

atoms ↔ chemical bonds

SIGNAL atom and bond attributes

TASK predict activity, toxicity, or properties



SENSOR NETWORKS

devices ↔ communication links

SIGNAL measurements and local state

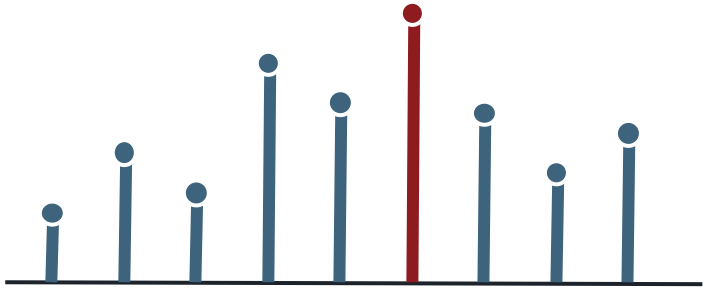
TASK recover data or coordinate decisions

Example: SlideGraph+ — HER2 status from H&E slides, Medical Image Analysis, 2022 · Image credits: PubChem CID 2519 · Michel Bakni / Wikimedia Commons

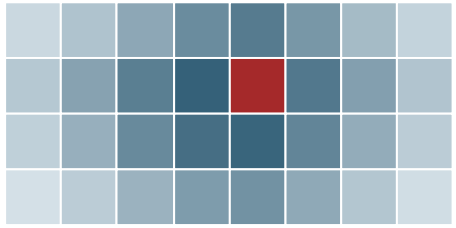
A graph signal is a value on every node

Regular domain

time / space



1-D grid · audio signals

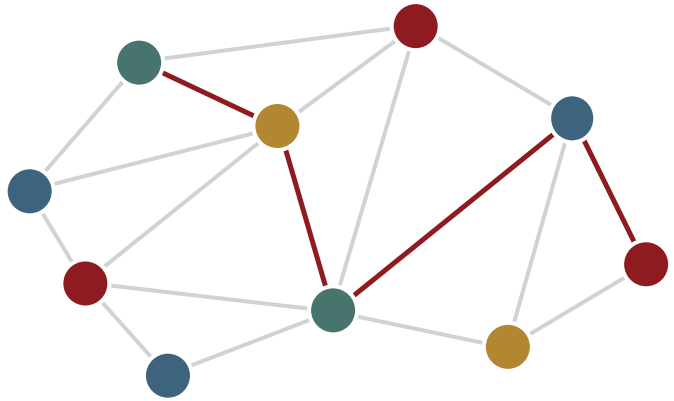


2-D grid · image pixels

indexed by ordered samples

Irregular domain

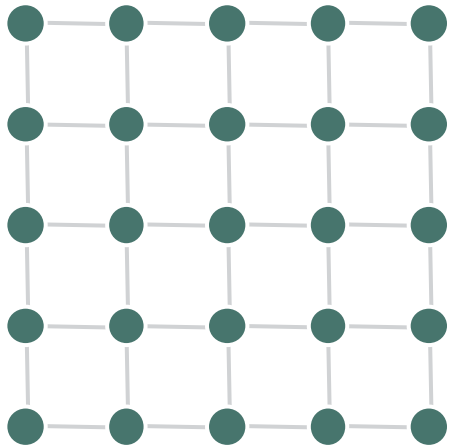
graph



indexed by nodes and relationships

The signal is familiar. The indexing set has changed.

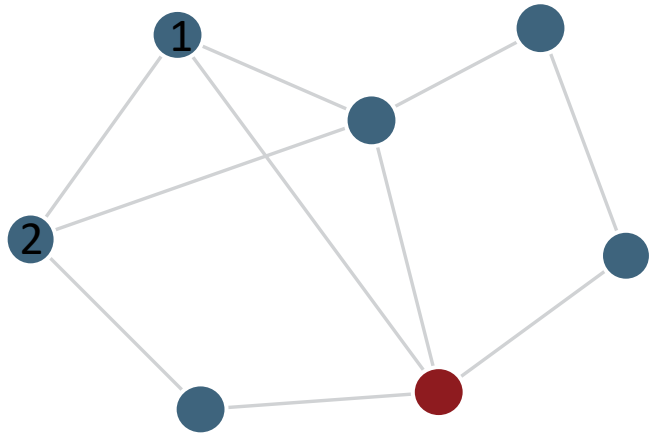
Irregular domains break three grid rules



GRID

fixed neighborhoods

≠



NETWORK

arbitrary topology

RULE 1 · NO CANONICAL ORDER

Relabeling nodes should not change the answer.

RULE 2 · IRREGULAR NEIGHBORHOODS

Nodes have different numbers of neighbors.

RULE 3 · DYNAMIC + MULTIMODAL

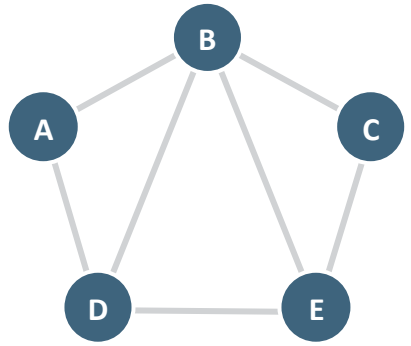
Edges, features, and targets can all evolve.

The graph you build decides what the model can see

UNDIRECTED

COAUTHORSHIP

mutual collaboration



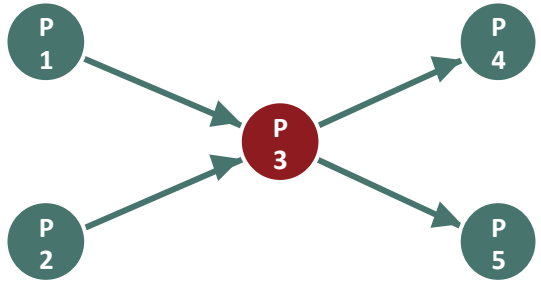
QUESTION

Who collaborates with whom?

DIRECTED

CITATION FLOW

direction matters



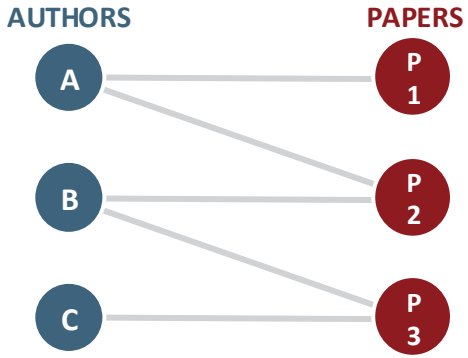
QUESTION

Which paper influences later work?

BIPARTITE

AUTHOR–PAPER

two entity types



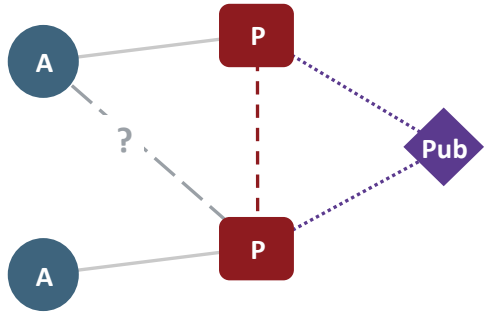
QUESTION

Which author wrote which paper?

HETEROGENEOUS

CITATION NETWORK

three node types



QUESTION

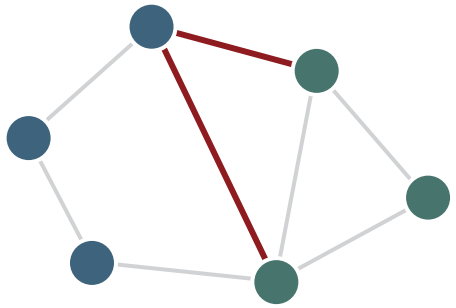
Which connections are missing?

Different data and tasks lead to different models.

Three questions we can ask about one graph

Imagine temperature measurements on a six-sensor network.

01 UNDERSTAND

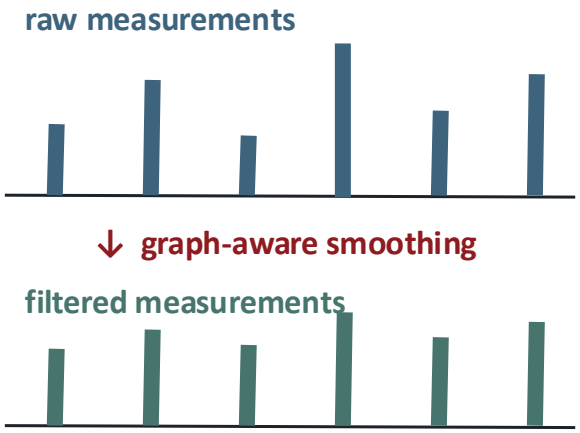


two regions joined by
a few cross-region edges

Which structure is present?

communities · bottlenecks · spectral structure

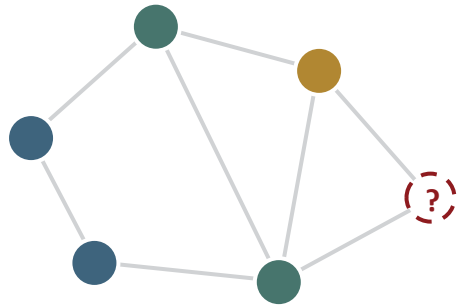
02 PROCESS



How should information move?

diffusion · denoising · filtering

03 LEARN



predict from its
connected neighbors

What can the graph predict?

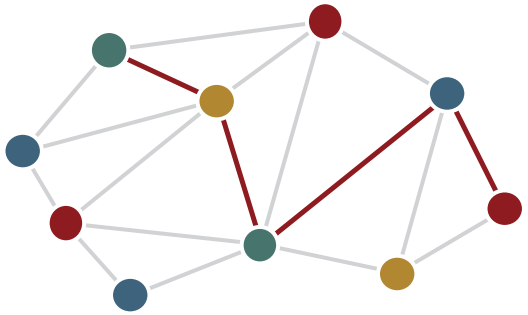
nodes · links · graphs · dynamics

A research project often combines two of these questions.

From a graph picture to symbols

A shared vocabulary before the operators

$$G = (V, E, A)$$



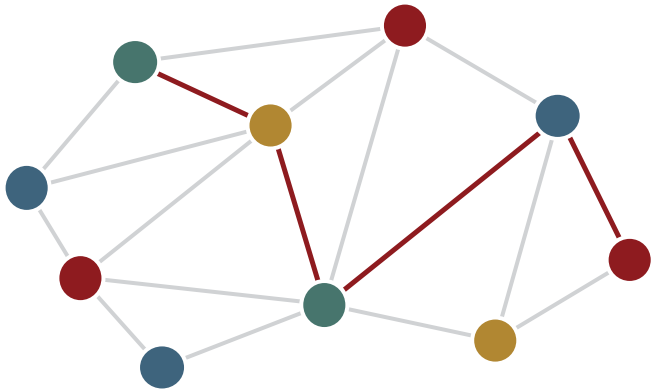
$n = |V|$ nodes

edge weight A

$V = \{1, \dots, n\}$	entities or samples
$A = [a_{ij}]$ $(n \times n)$	relationships and their weights
$X = [x_1; \dots ; x_n]$ $(n \times d)$	signals or features on the nodes
$\mathcal{N}(i) = \{j : a_{ij} \neq 0\}$	one-hop neighbors of i
$f_\theta(G, X)$	a filter, predictor, controller, or generator

We will use this notation throughout. Next, we turn A into an operator.

The Laplacian makes connectivity an operator



graph + signal

$$(G, x)$$

$$L = D - A$$

$x^{\mathrm{T}}Lx$ measures variation across connected nodes

$$L = U\Lambda U^{\mathrm{T}}$$

eigenvectors → graph modes
eigenvalues → graph frequencies

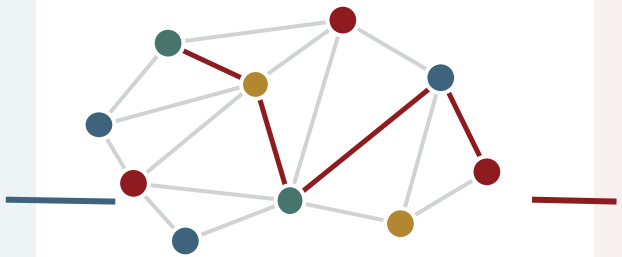
Structure becomes algebra, algebra becomes a frequency language.

EE and CS ask different questions about the same graph

EE lens

Signals + systems

- operators and spectra
- filtering and diffusion
- stable implementation



graph operator

CS lens

Data + learning

- representation learning
- prediction and optimization
- expressivity + generalization

Graph filters have a local and a spectral view

$$h(\boldsymbol{L})\boldsymbol{x} = \sum_k h_k \boldsymbol{L}^k \boldsymbol{x}$$

LOCAL VIEW

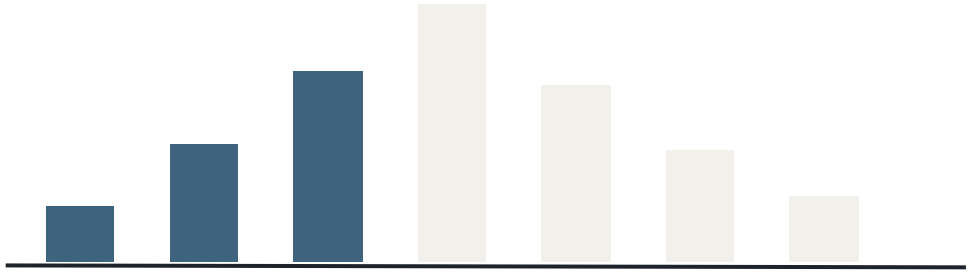
Each power of L reaches one more hop.

Distributed computation emerges naturally from neighborhood exchange.

$$h(\boldsymbol{\Lambda})\hat{\boldsymbol{x}} = \sum_k h_k \boldsymbol{\Lambda}^k \hat{\boldsymbol{x}}, \quad \hat{\boldsymbol{x}} = \boldsymbol{U}^T \boldsymbol{x}$$

SPECTRAL VIEW

$h(\Lambda)$ shapes graph frequencies.



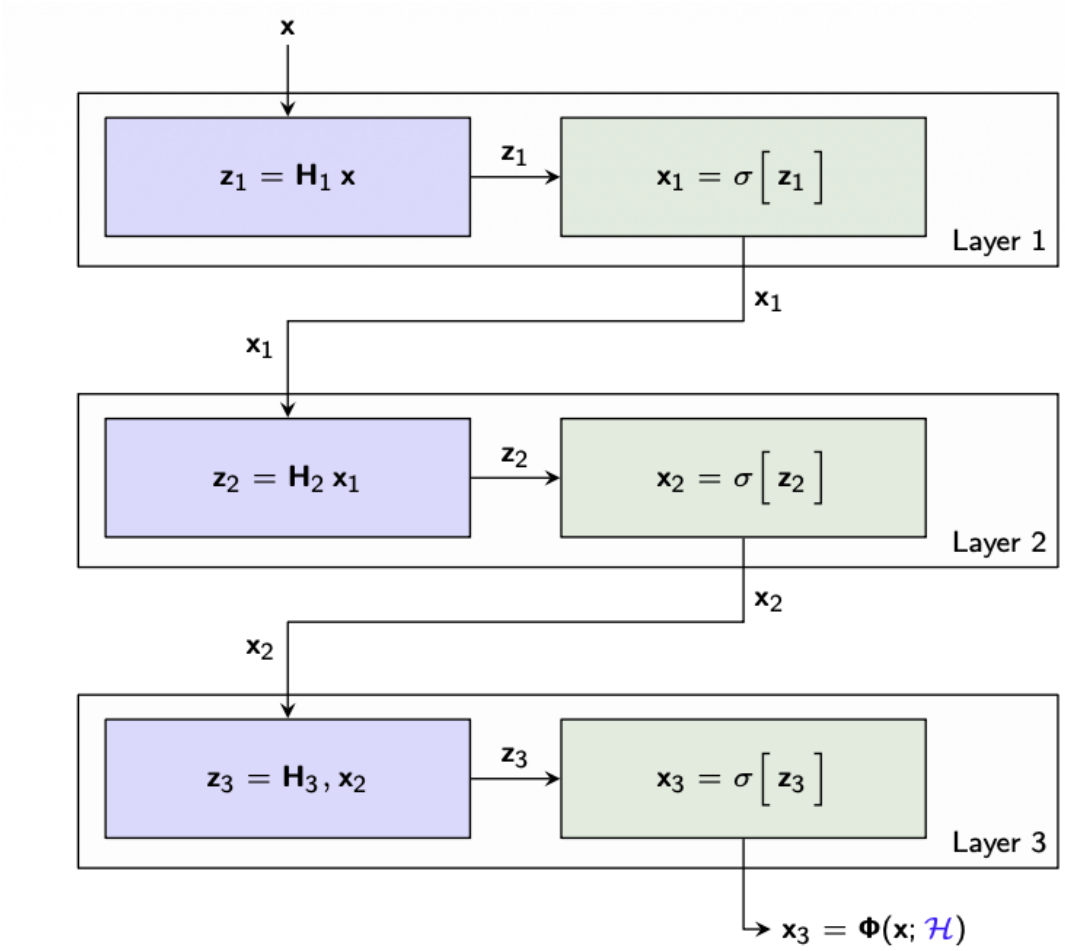
The same operator can be read in two ways.

From NNs to CNNs to GNNs

NEURAL NETWORK

$$\mathbf{z} = \mathbf{H}\mathbf{x}, \mathbf{x} = \sigma(\mathbf{z})$$

Linear map.
Parameters grow with the size of the signal.

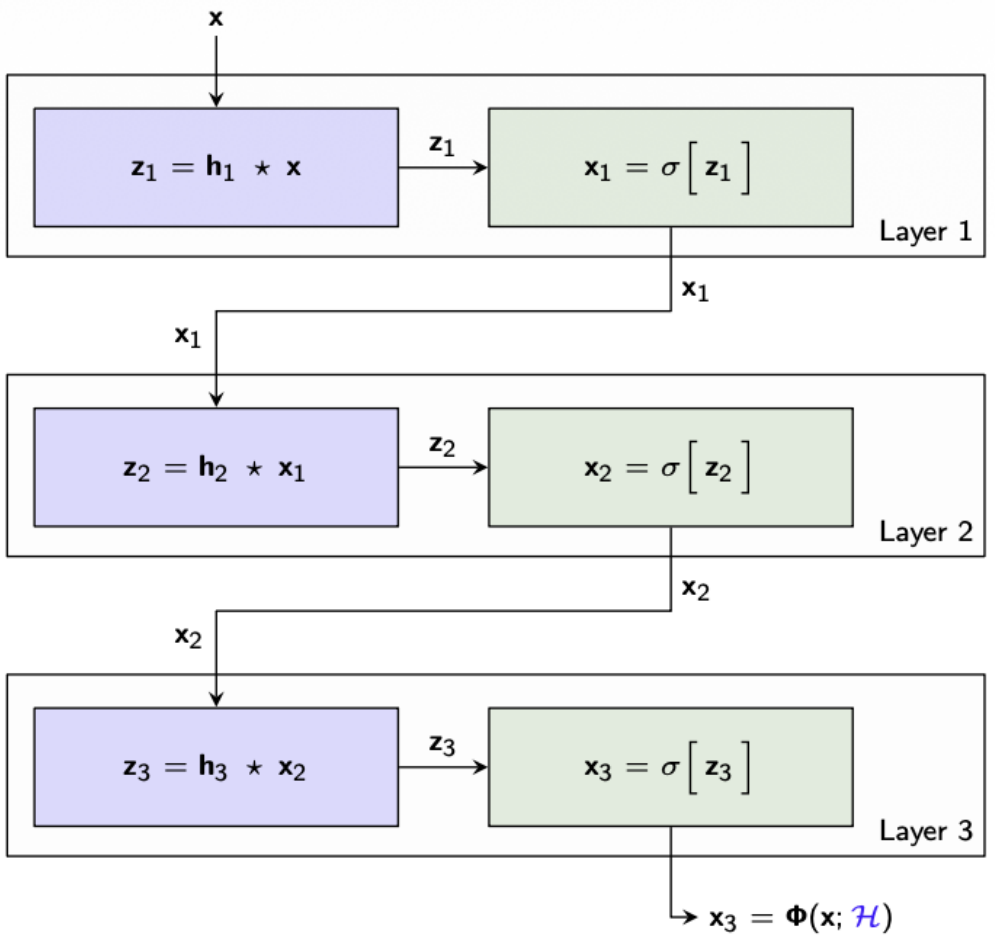


From NNs to CNNs to GNNs

CONVOLUTIONAL NN

$$\mathbf{z} = \mathbf{h} \star \mathbf{x}, \mathbf{x} = \sigma(\mathbf{z})$$

Use a convolution instead: local, shared weights.
It scales as convolution scales.
Fixed neighborhood pattern.

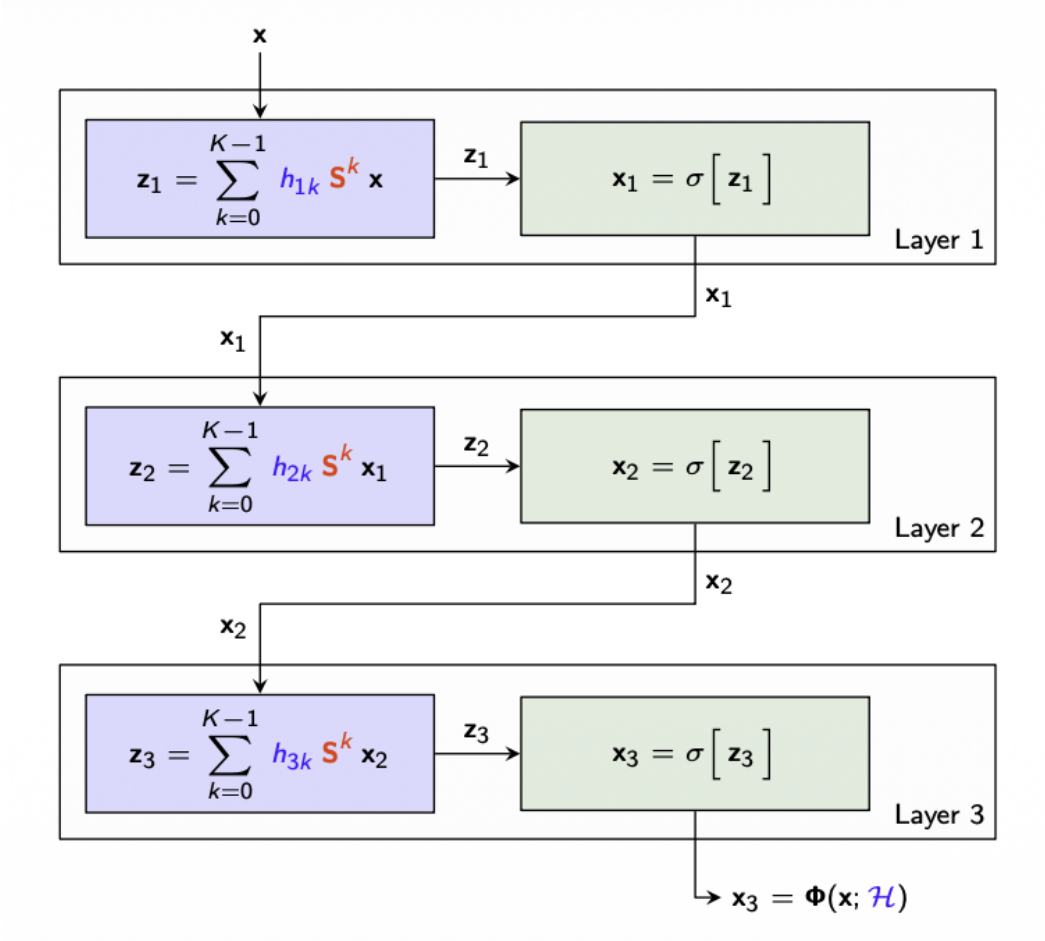


From NNs to CNNs to GNNs

GRAPH NEURAL NETWORK

$$\mathbf{z} = \Sigma_k h_k \mathbf{S}^k \mathbf{x}, \mathbf{x} = \sigma(\mathbf{z})$$

Generalize CNNs with graphs.
Scales as graph maintains structure.
Replace ordered domain with graphs



From NNs to CNNs to GNNs

NEURAL NETWORK

$$\mathbf{z} = \mathbf{H}\mathbf{x}, \mathbf{x} = \sigma(\mathbf{z})$$

Linear map.
Parameters grow with the size of the signal.

CONVOLUTIONAL NN

$$\mathbf{z} = \mathbf{h} \star \mathbf{x}, \mathbf{x} = \sigma(\mathbf{z})$$

Use a convolution instead: local, shared weights.
It scales as convolution scales.

GRAPH NEURAL NETWORK

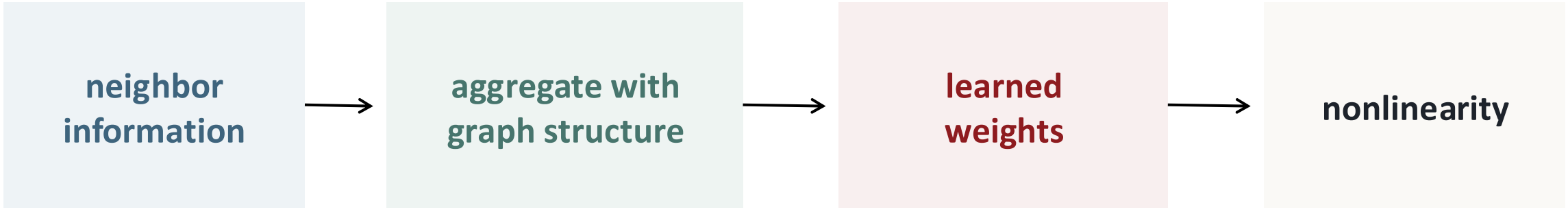
$$\mathbf{z} = \Sigma_{\mathrm{k}} h_{\mathrm{k}} \mathbf{S}^{\mathrm{k}} \mathbf{x}, \mathbf{x} = \sigma(\mathrm{z})$$

Generalize CNNs with graphs.
Scales as graph maintains structure.

Same recipe each time: a linear map, then a pointwise nonlinearity, stacked in layers.

Only the linear map changes: matrix → convolution → graph convolution.

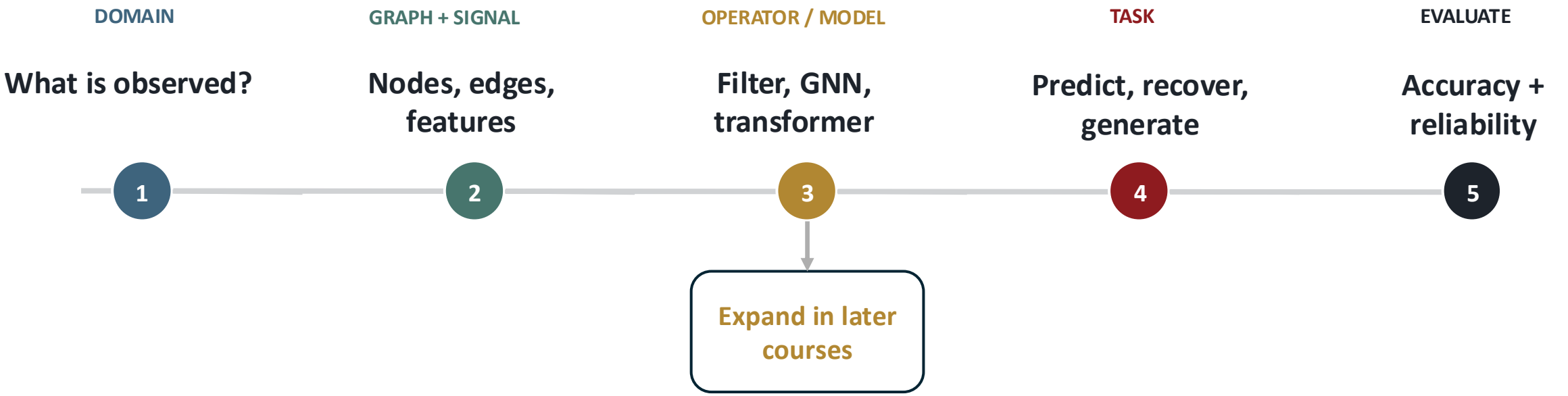
A GNN layer learns how to aggregate



$$X^{(l+1)} = \sigma(\text{ graph operator} \times X^{(l)} \times \text{learnable weights })$$

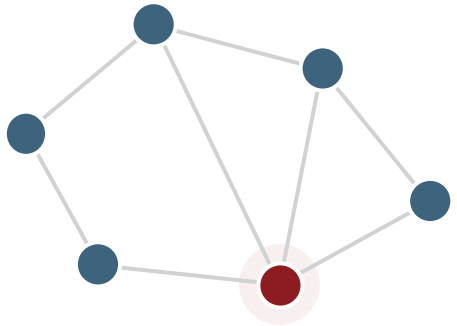
A GNN turns this local operator into a trainable update.

A project begins with five modeling choices



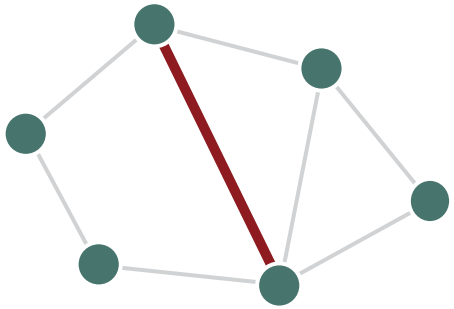
The early modeling choices usually matter more than swapping one architecture for another.

Graph learning operates at several scales



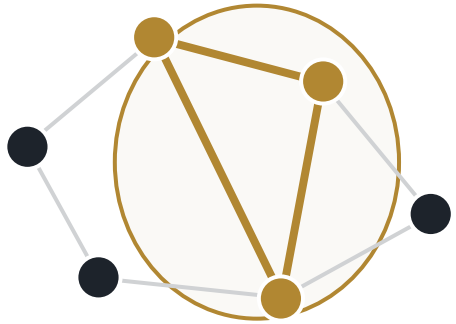
NODE

Which sensor is faulty?



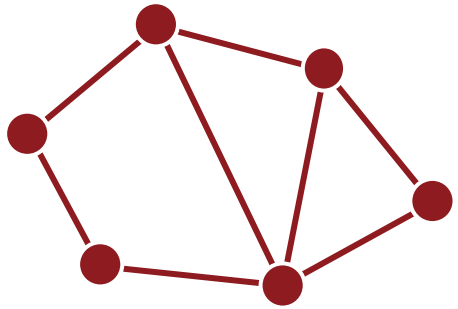
EDGE

Will two entities interact?



SUBGRAPH

Is this community unusual?



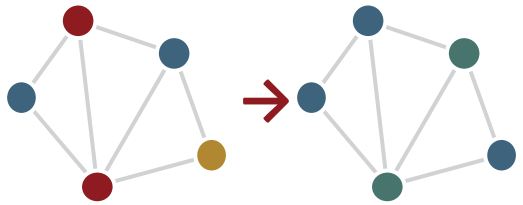
GRAPH

What property does this molecule have?

The output scale changes. The representation challenge remains relational.

Four basic graph-signal operations

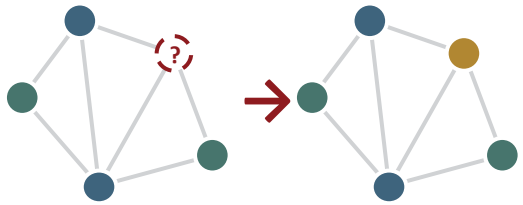
DENOISE



irregular → smooth

Remove high-frequency variation while preserving structure.

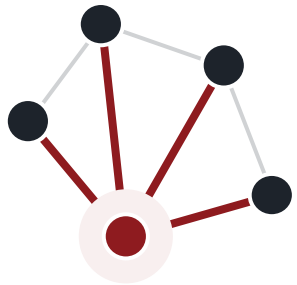
INTERPOLATE



missing → recovered

Recover values at unobserved or failed sensors.

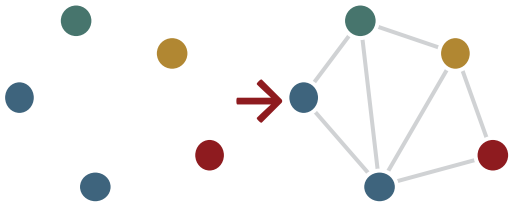
DETECT



local inconsistency

Find nodes or edges that violate local patterns.

INFER



signals → structure

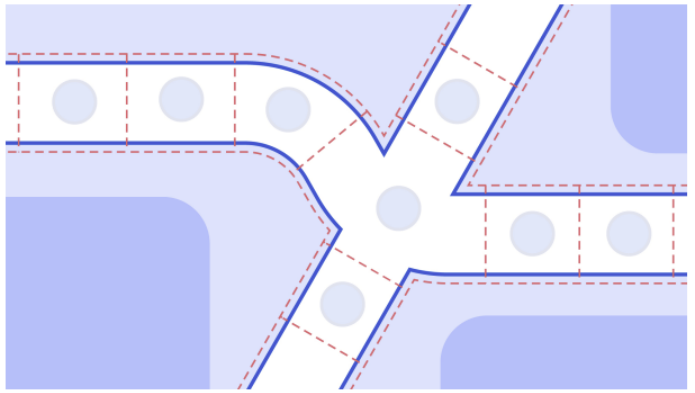
Learn the graph itself from observed signals.

Modern graph-learning pipelines often learn or compose these same operations.

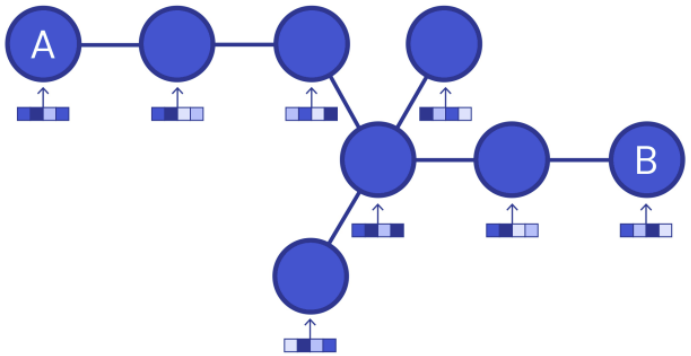
Traffic forecasting becomes a graph problem

Example 1 · formulation

NODES	road segments
EDGES	consecutive or intersecting segments
INPUT $x_v(t)$	recent speed, flow, and time context
TARGET	ETA for a candidate route A → B
Why a graph?	Travel time on one segment depends on connected upstream and downstream segments.



A road network with shared traffic volume

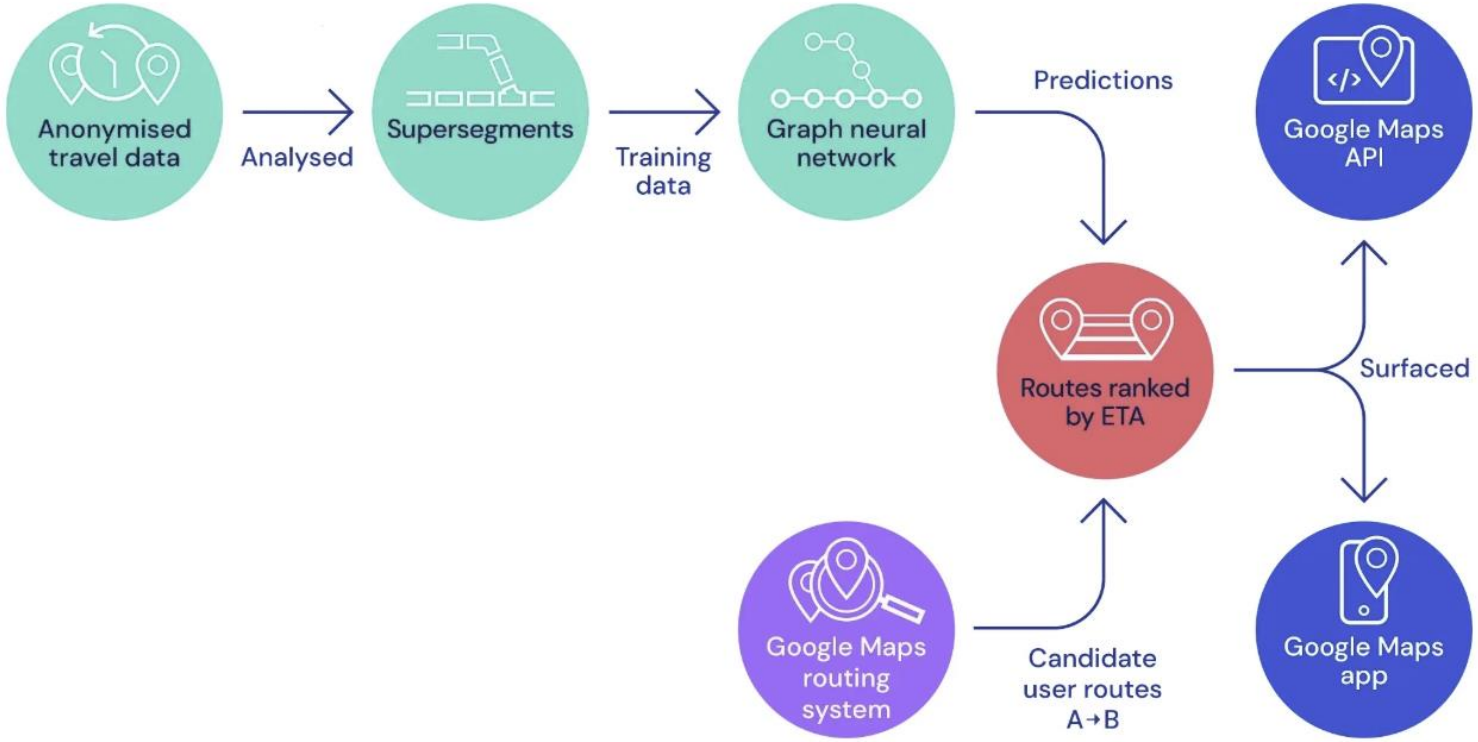


Connect adjacent segments with edges

Image credits: DeepMind blog

Traffic GNNs share one local update

Example 1 · solution pipeline



The model architecture for determining optimal routes and their travel time.

The update is local, and the same parameters are used at every road segment.

Molecular design can be conditional graph generation

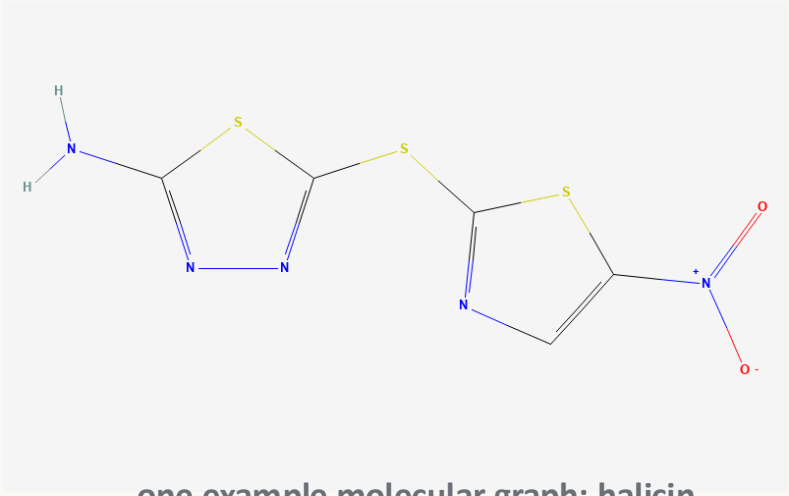
Example 2 · formulation

OBJECT G	a molecule with atom and bond types
CONDITION c	desired activity, scaffold, or geometry
CONSTRAINTS	valid valence, connectivity, synthesizability
GOAL	sample $G \sim p(G \mid c)$, then score and rank
Key change	the output is a new graph—not only a label

Prediction and generation answer different questions

predict: $G \rightarrow \hat{y}$

generate: $c \rightarrow$ new G

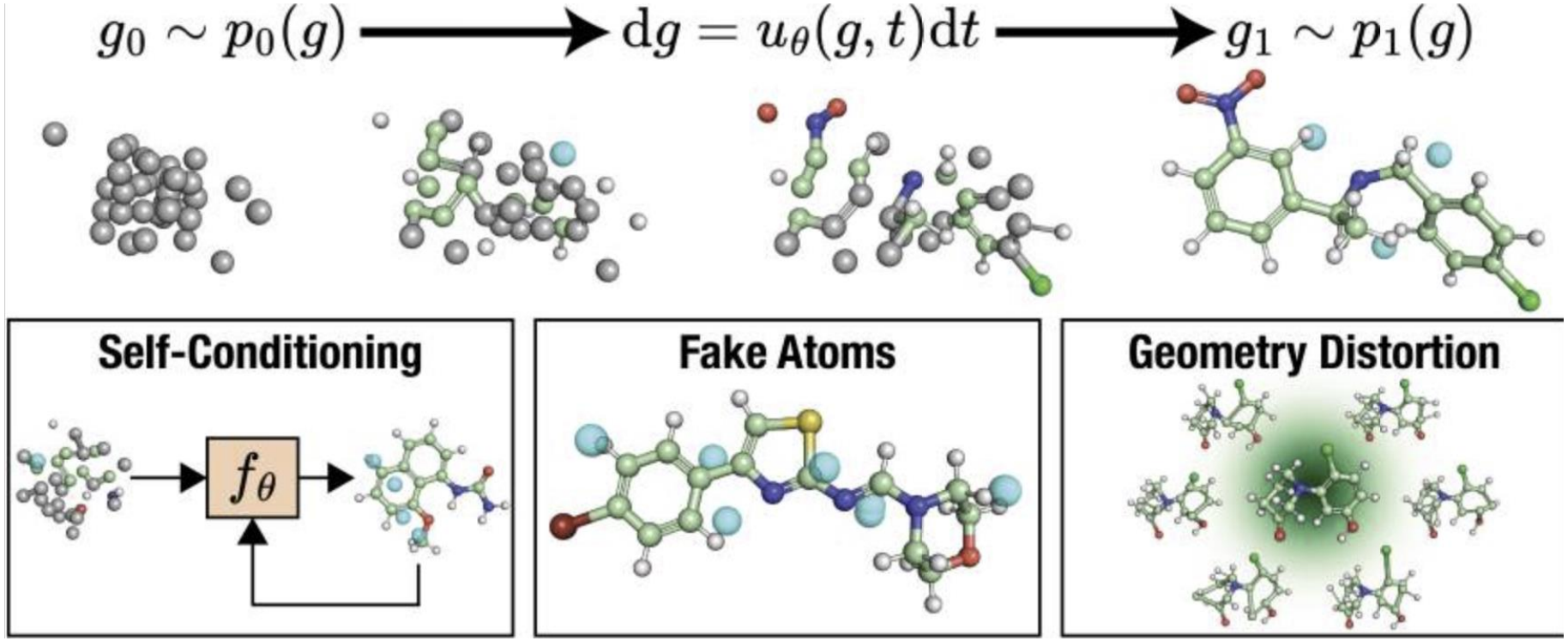


one example molecular graph: halicin

Structure image: PubChem CID 11837140

Graph Generative Models generates candidates

Example 2 · generative pipeline



- Multi-modal flow matching model for unconditional de novo molecule generation
- Atomic coordinates are generated by continuous flow matching
- Atom types, charges, and bond orders are generated by discrete flow matching

..

Generated molecules are proposals. Chemistry and experiments still decide.

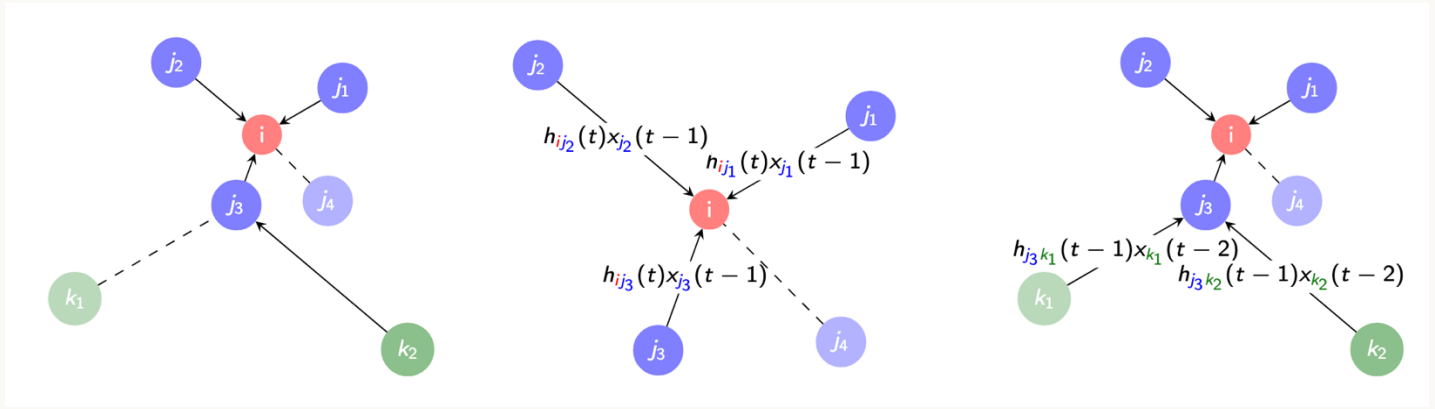
Image credits: Ian Dunn, David R. Koes; FlowMol3: flow matching for 3D de novo small-molecule generation. Digital Discovery 2026

Wireless control is a graph decision problem

Example 3 · formulation

NODES	transmitter–receiver links
EDGES	interference coupling between links
INPUT $x_i(t)$	local channel state, queue, power
OUTPUT $a_i(t)$	local power or scheduling action
Constraint	each agent uses only local state and K-hop messages

Interference graph at time t



neighbors send messages to link i

$$\mathcal{H}_i(t) := \bigcup_{k=0}^{K-1} \left\{ [\mathbf{H}(t-k+1)]_{jj'}, [\mathbf{x}(t-k)]_{j'} \mid j \in \mathcal{N}_i^{k-1}(t), j' \in \mathcal{N}_i^{k}(t) \right\}$$

$a_i(t) = \pi_\theta(\mathcal{H}_i(t))$ ---- policy π_θ

One policy is trained globally and executed locally

Example 3 · solution pipeline

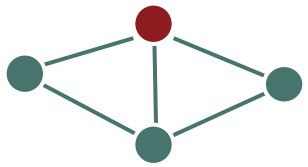
TRAINING

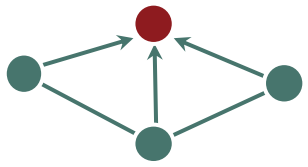
network states + utility function → **shared graph policy π_θ** → local actions → network utility + reward → update θ

repeat over sampled network states

DEPLOYMENT

1 local state $x_i(t)$ → 2 K neighbor exchanges → 3 shared policy π_θ → 4 local action $a_i(t)$





same θ at every node

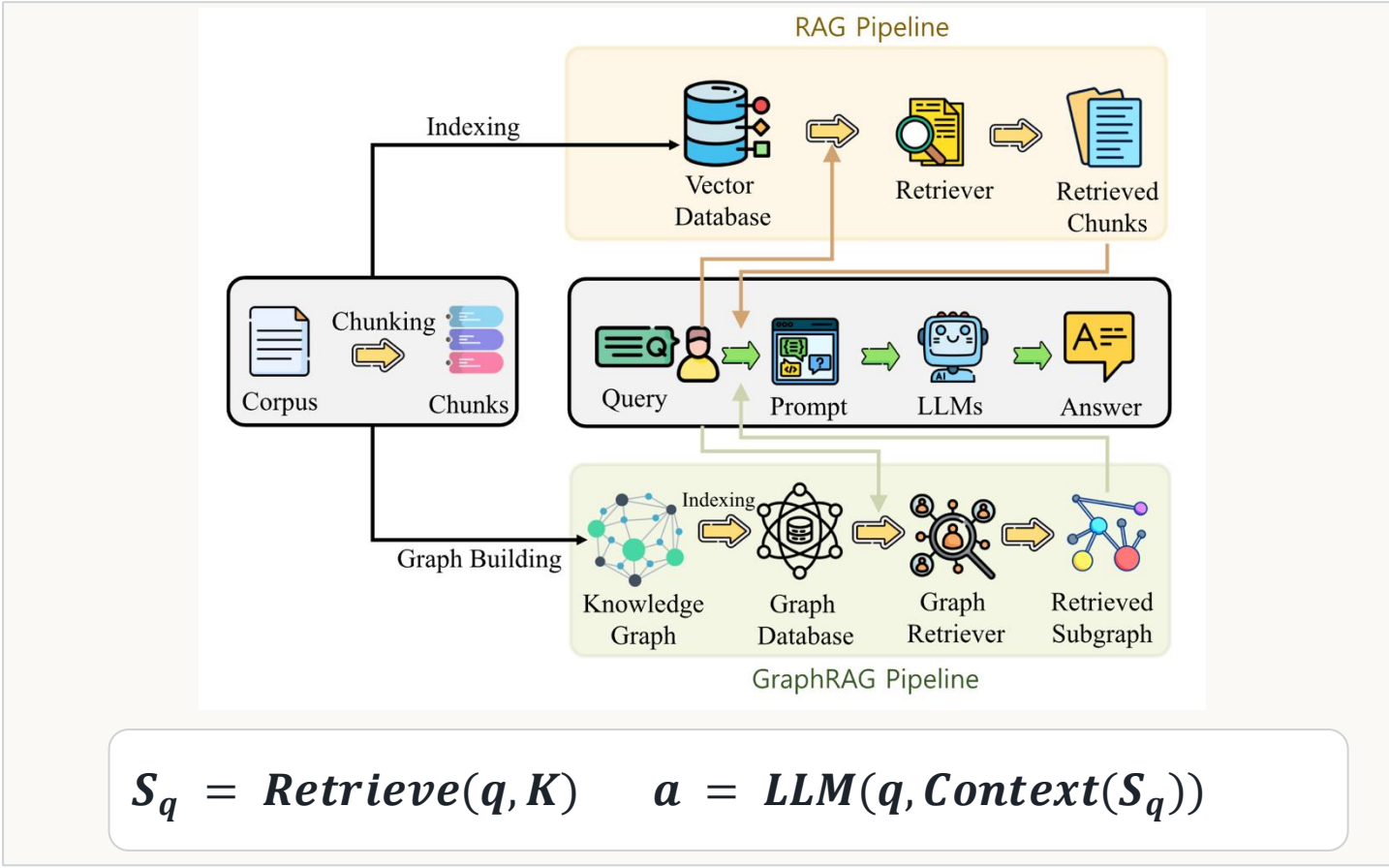
power or schedule

Training can use global feedback, while deployment uses only local messages.

Graph RAG turns retrieved evidence into a subgraph

Example 4 · knowledge-grounded language models

MEMORY K	**entities, document chunks, and claims**
EDGES	**relations, citations, and provenance**
QUERY q	**a natural-language information need**
TARGET	**answer** a **+ supporting subgraph** S_q
Why a graph?	**Retrieved context remains structured and traceable.**



$$S_q = Retrieve(q, K) \qquad a = LLM(q, Context(S_q))$$

Open question: which nodes, paths, or communities should be retrieved?

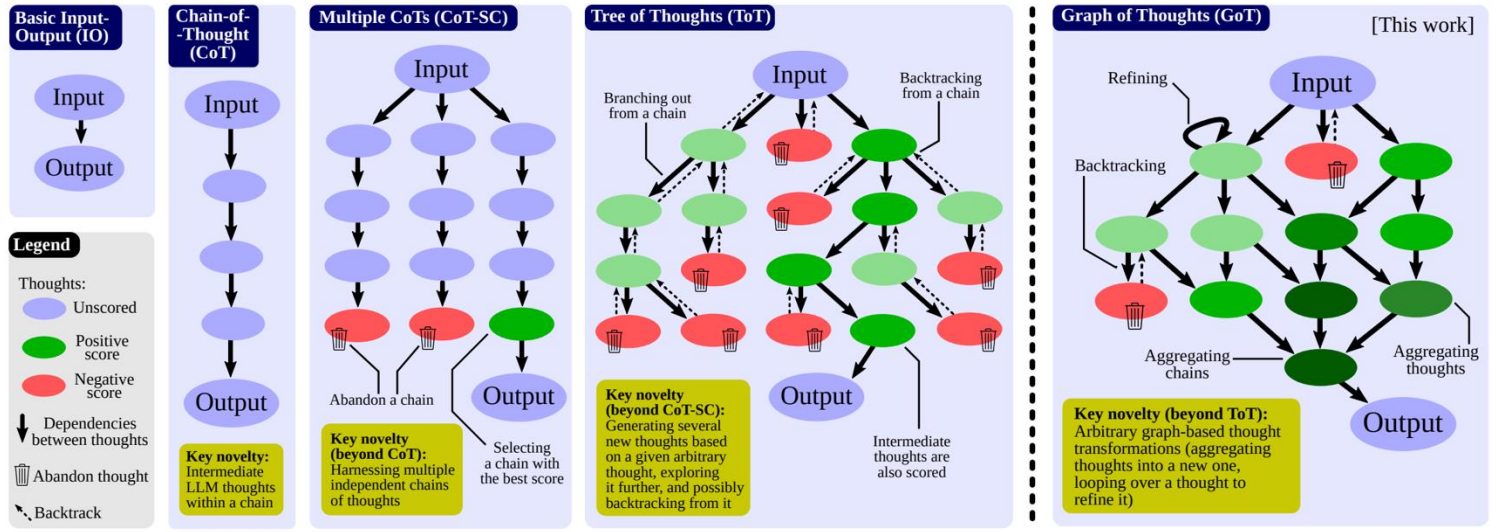
Image credits: Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2025. Graph Retrieval-Augmented Generation: A Survey. ACM Trans. Inf. Syst.

Reasoning graphs make intermediate states explicit

Example 5 · structured LLM reasoning

NODES	hypotheses, tool results, intermediate states
EDGES	inferences, dependencies, revisions, feedback
SEARCH	expand, score, merge, and prune candidates
OUTPUT	an answer plus a visible reasoning trace
Why a graph?	Branches can merge evidence; a chain cannot.

Expand → score → merge → answer



$$v^* = \arg\max \boldsymbol{score}_{\varphi}(v \mid q, \boldsymbol{evidence})$$

Retrieved, generated, and revised graphs are not fixed ground truth.

Next question: what remains reliable when the graph changes? →

Image credits: Besta, Maciej, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda et al. "Graph of thoughts: Solving elaborate problems with large language models." In *Proceedings of the AAAI conference on artificial intelligence*, 2024.

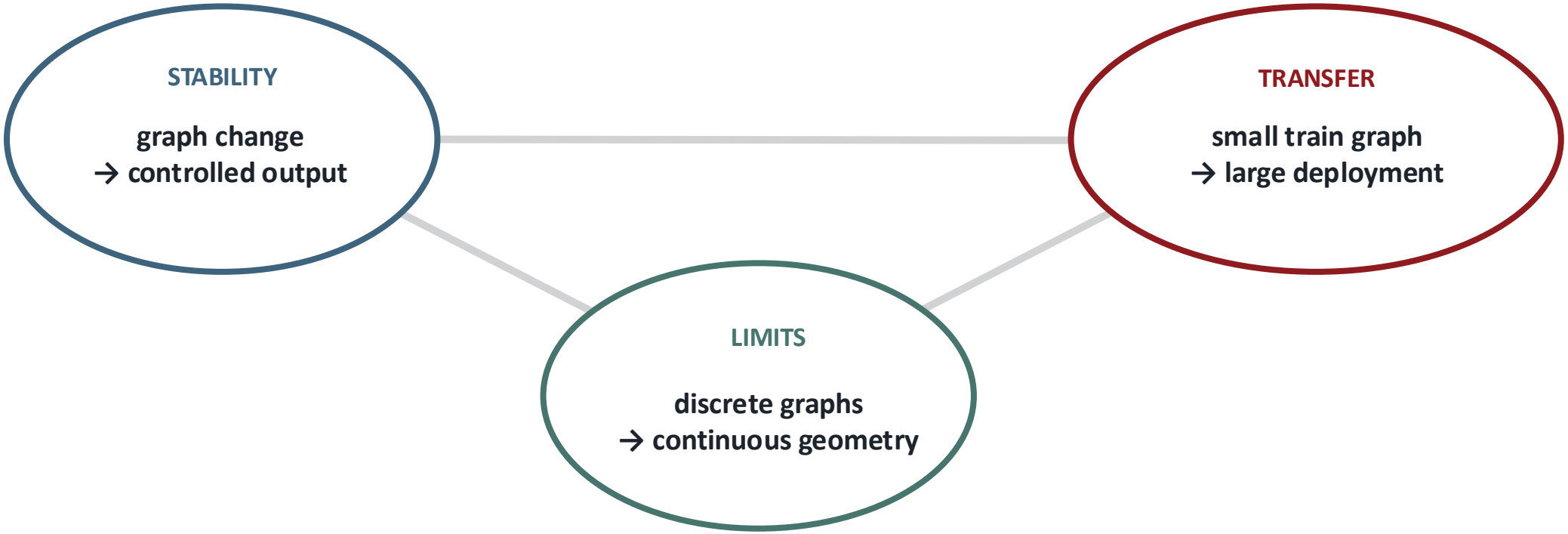
Three questions guide the theory part of this course

Observed, generated, and retrieved graphs can all be noisy, larger, or out of distribution.

PERTURBATION	**What if observed or inferred edges are wrong?**	stability
SCALE	**What if deployment graphs are much larger?**	transferability
DISTRIBUTION	**What if train and test graph distributions differ?**	generalization

These questions—stability, transferability, and generalization—will recur throughout the course.

When does graph learning survive change?

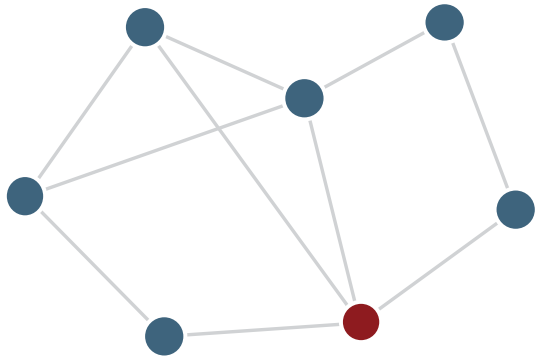


A useful tension: stable and transferable representations may discard distinctions that matter.

Case study: can a wireless policy transfer?

Train

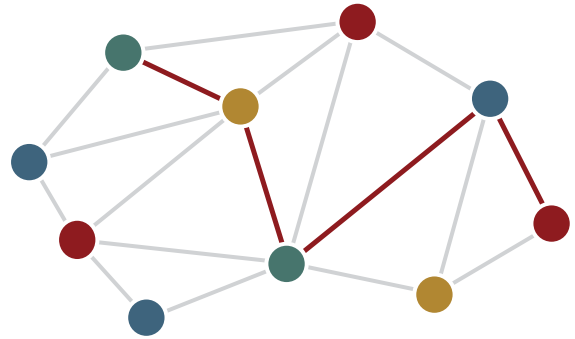
small random network



same learned policy

Deploy

larger, changing network

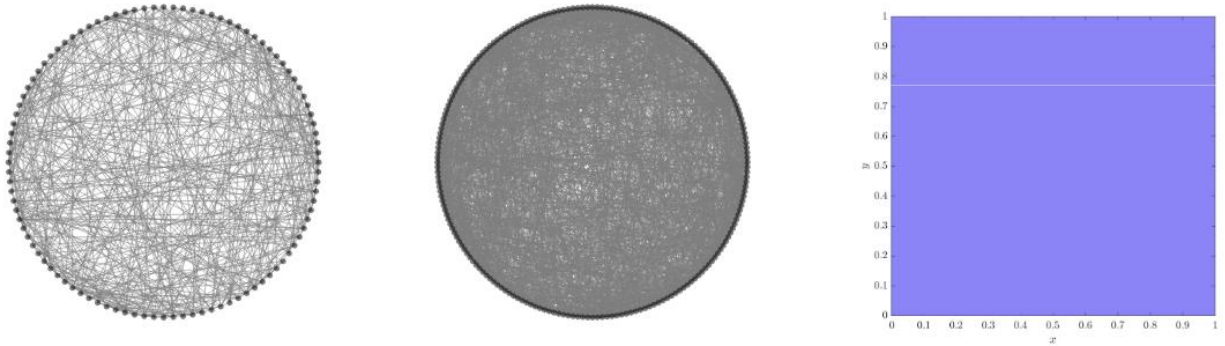


The graph encodes interference. Local aggregation enables decentralized decisions.

When does performance transfer across network configurations?

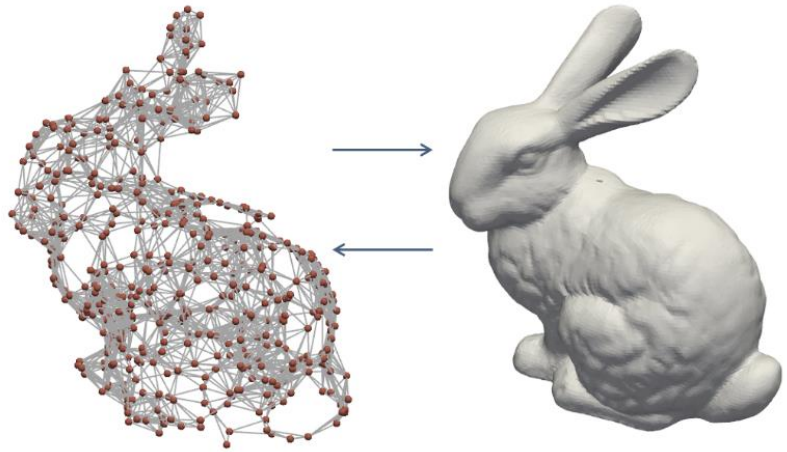
A continuous limit can explain a family of graphs

Graphon



$$Y(v) = (T_H X)(v) = \sum_{k=0}^{K-1} h_k \left(T_W^{(k)}\right)(v)$$

Manifold



$$g(x) = \sum_{k=0}^{K-1} h_k \, e^{-k\mathcal{L}} f(x)$$

If graphs approximate the same geometry, their filters and GNNs may admit common limits.

The semester moves from foundations to a project

WEEKS 1–4

FOUNDATIONS

Laplacians · GFT · filters

WEEKS 5–6

CORE GNNS

message passing · failure modes

WEEKS 7–10

LIMITS + MODERN MODELS

limits · attention · generation

WEEKS 11–14

INFERENCE + PRACTICE

inference · dynamics · applications

Each phase adds a few tools that give you insights in the final project.

Four kinds of work, used together

01	DERIVE	reason with graph operators and spectral filters
02	IMPLEMENT	build and diagnose graph filters and GNNs
03	EVALUATE	question stability, transfer, and generalization
04	CREATE	turn a paper or application into a research direction

The aim is to move comfortably between derivation, code, evidence, and design.

The paper talk and project ask for different skills

Paper presentation

Read beyond the abstract.

- technical summary
- methodological critique
- reproducibility analysis
- future research directions

30 minutes · individual or pair

Course project

Build one idea far enough to learn from it.

- define a question
- choose theory, experiments, or both
- report successes and failures honestly

proposal → midpoint → final

A project can be theoretical, applied, or both

A THEORY

stability bounds
spectral analysis
size transfer

best for a focused proof or analysis

B SYSTEMS

sensor networks
multi-agent systems
cross-domain transfer

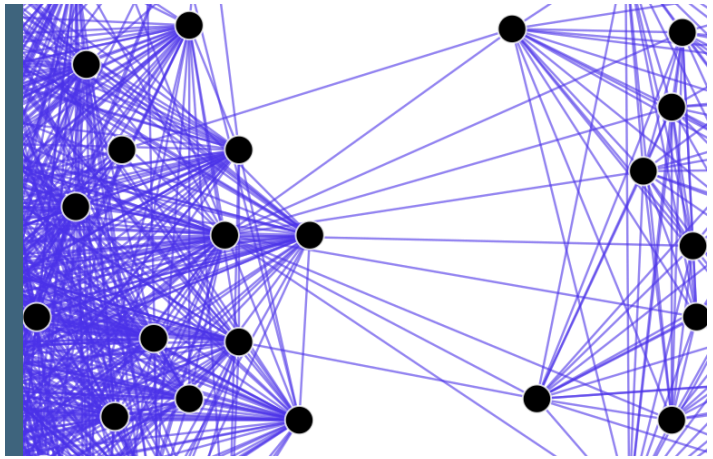
best for a domain-driven prototype

C HYBRID

mathematical claim
+ empirical validation

encouraged when scope stays realistic

First steps for different backgrounds



INFORMATION + CS

Recommendation and discovery

Rank papers or products; predict missing citations or interactions.

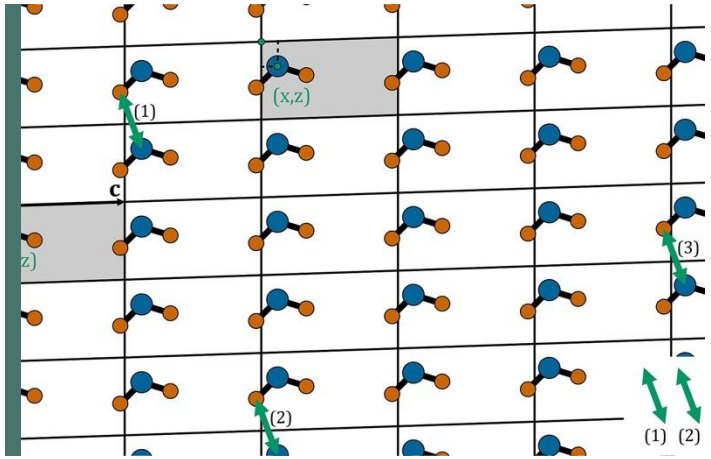
FIRST MOVE · link prediction

SYSTEMS + EE

Networked autonomy

Coordinate robots or radios using only local measurements and messages.

FIRST MOVE · decentralized control



SCIENCE + ENGINEERING

Scientific design

Predict properties of new molecules, materials, or biological systems.

FIRST MOVE · graph regression

Image credits: DarwinPeacock & Maklaan / Wikimedia Commons · asuscreative / Wikimedia Commons · BT-Kristall / Wikimedia Commons

Enjoy the semester and journey to graphs!

Reach out to me with your questions