Deep Learning for Graphs

Artificial Intelligence for Digital Health (AID)
M.Sc. in Digital Health – University of Pisa
Davide Bacciu (davide.bacciu@unipi.it)

Lecture Outline

Deep learning for graphs

- Motivation
- Graph formalism
- Learning tasks: Graph prediction, induction, transduction and generation
- Fundamental components of a graph neural network
- Applications to healthcare and biology

Graph Fundamentals

Why Graphs?

Why Graphs?

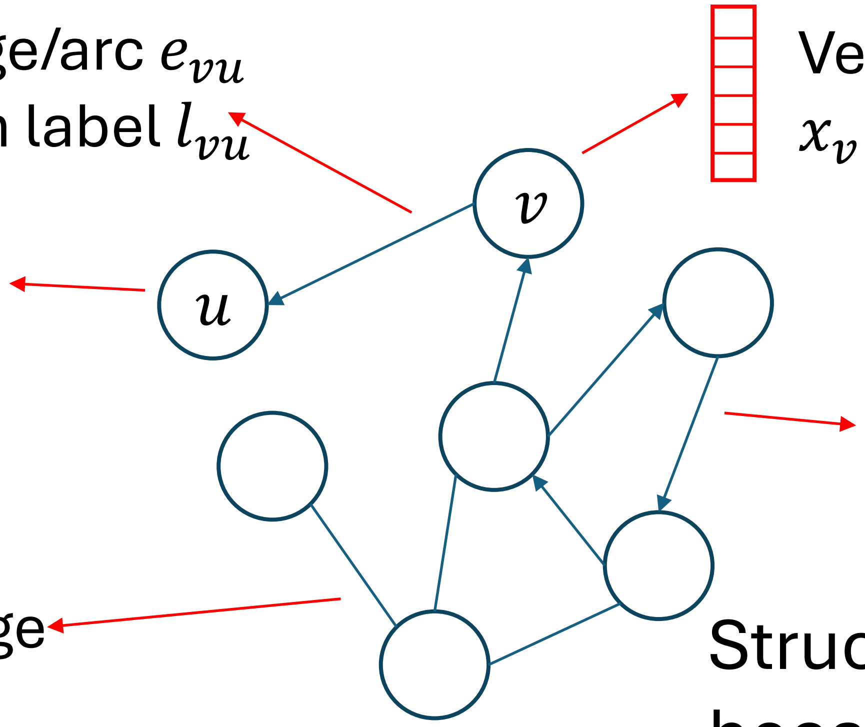
Context is fundamental for the correct interpretation of information

Graph Structured Data

Oriented edge/arc e_{vu}
possibly with label l_{vu}

Vectorial node label
x_v

Node/vertex u

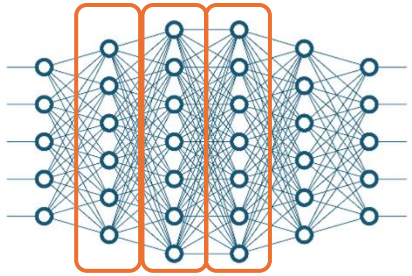


Cycle

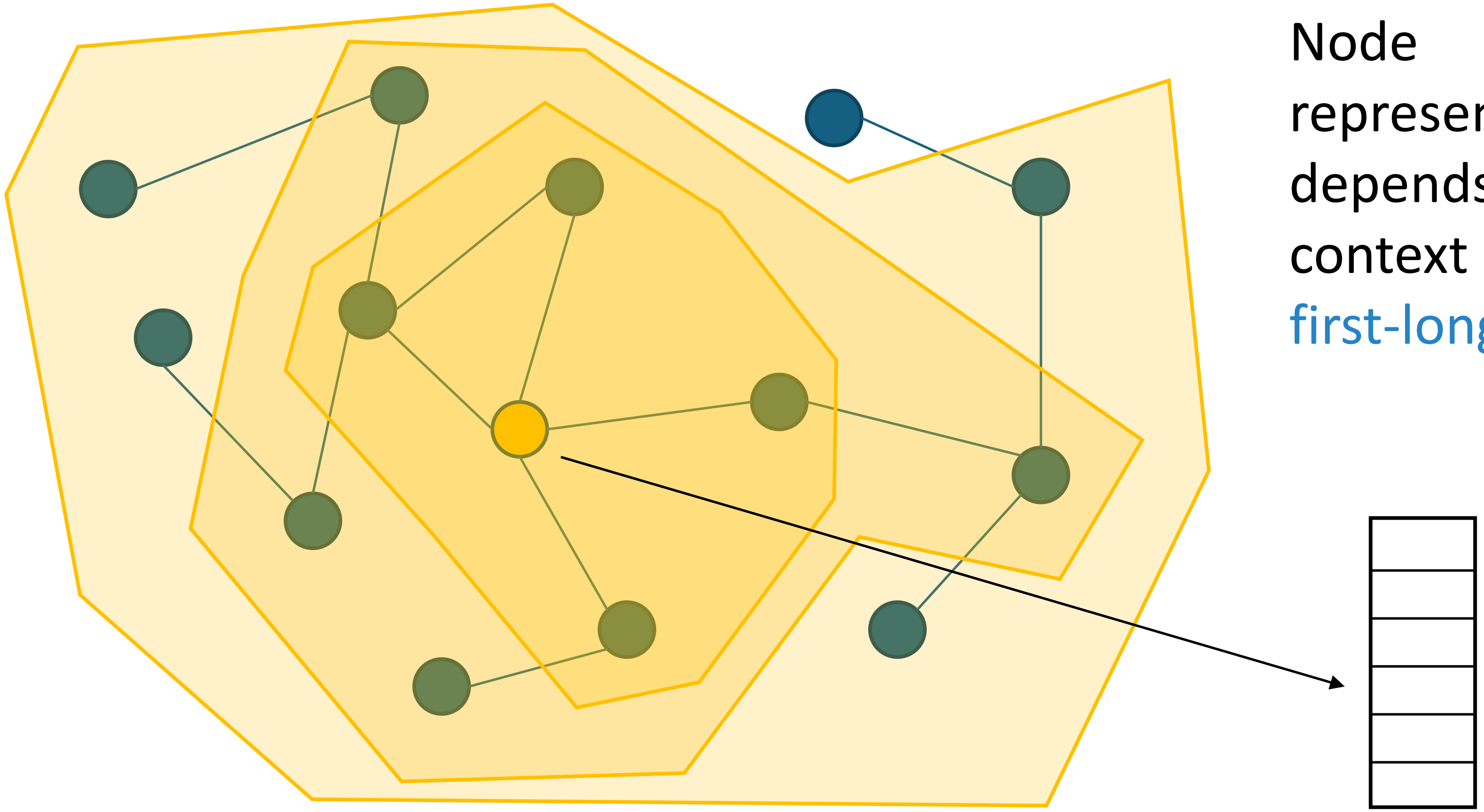
Undirected edge

Structures are useful because allow to represent relationships in the data

Deep Learning with graphs

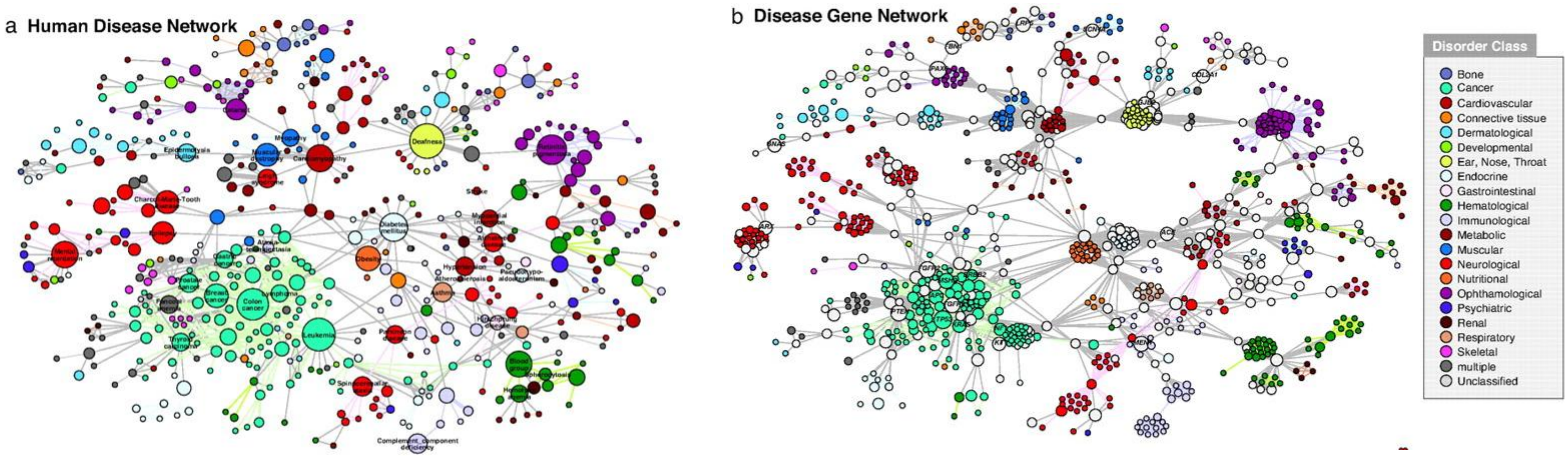


Hierarchical representation learning allows to efficiently diffuse information through graph structure



Node representation depends on its context (shorter first-longer later)

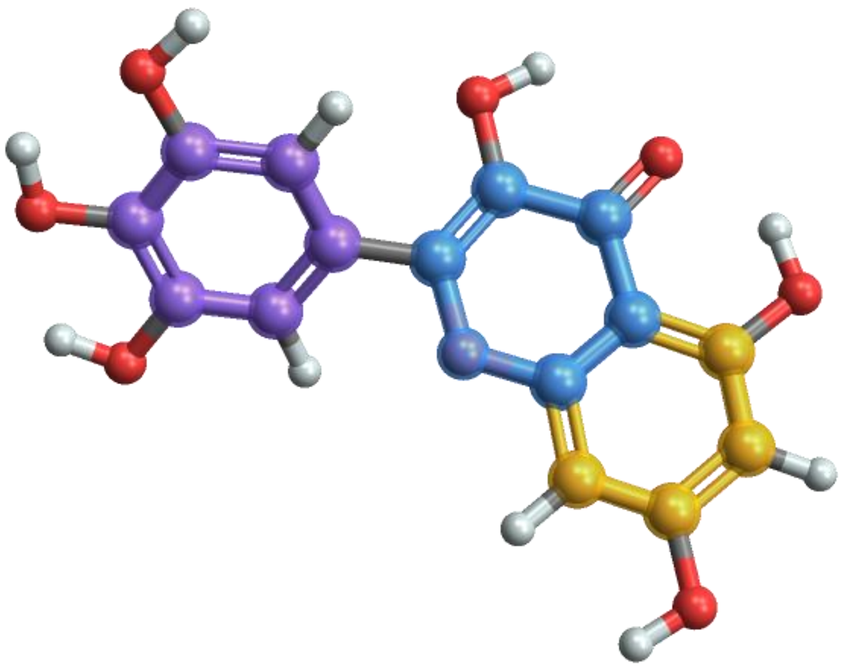
Why graphs in digital health?

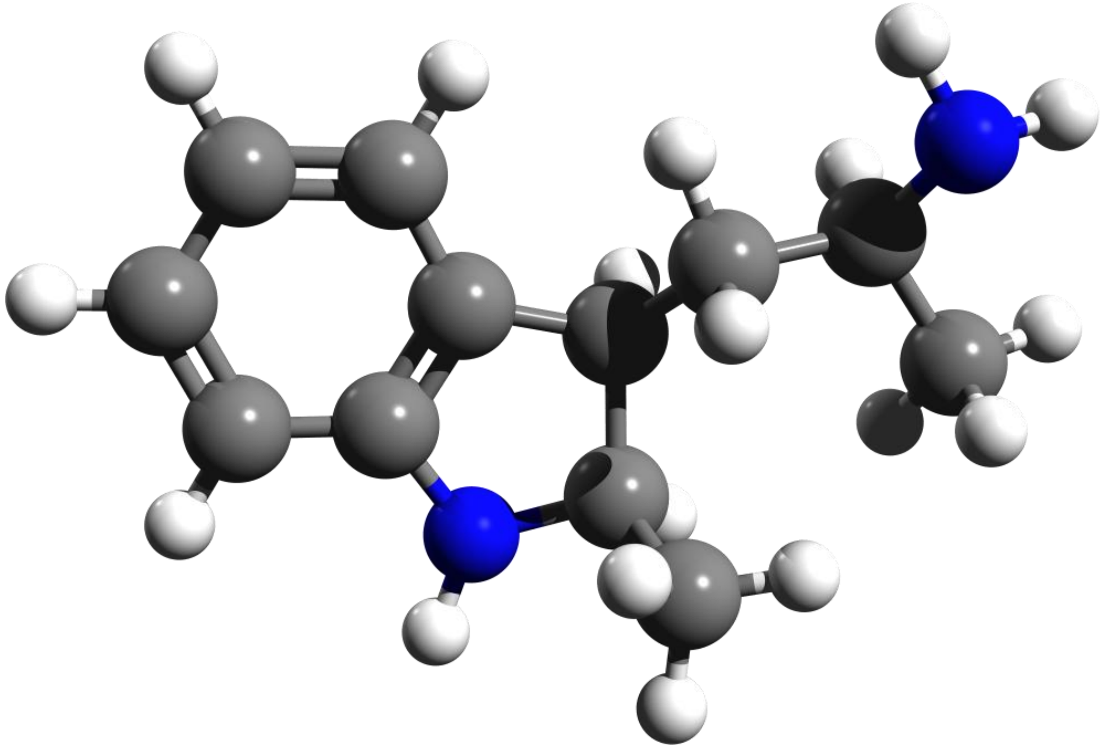


Source: Goh et al. PNAS 2007

Why graphs in digital health?

Molecular graph

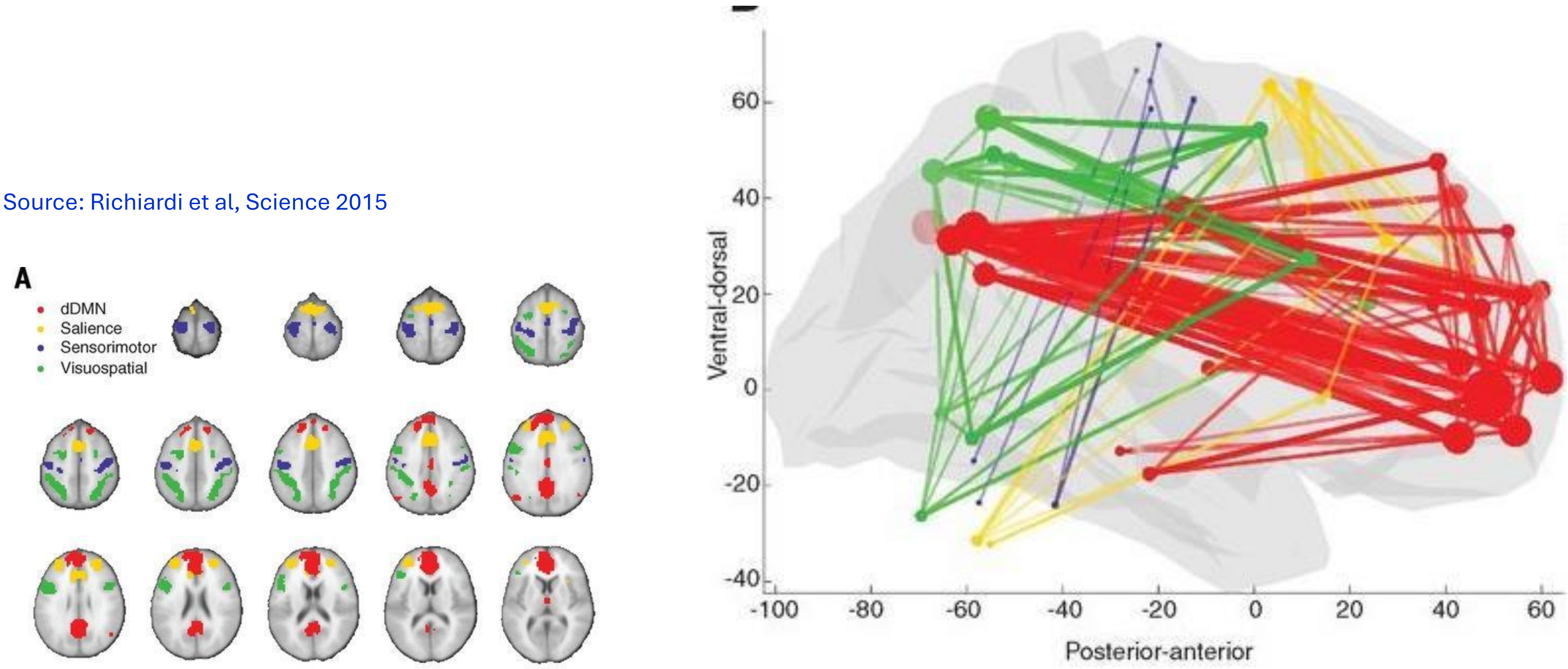




3D Structure

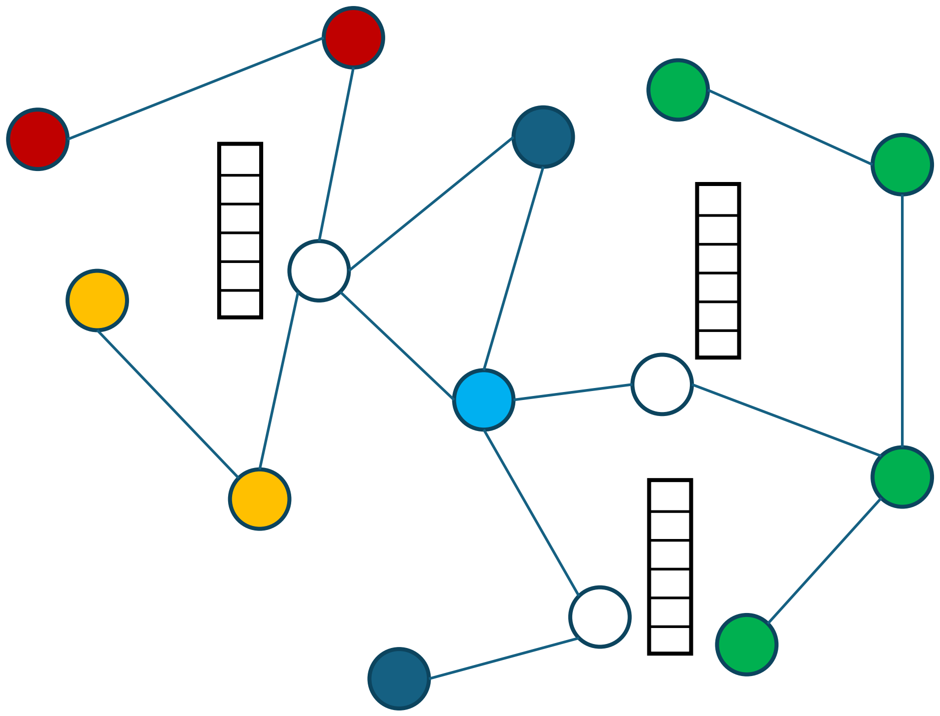
Why graphs in digital health?

Source: Richiardi et al, Science 2015



Predictive Tasks

Network data



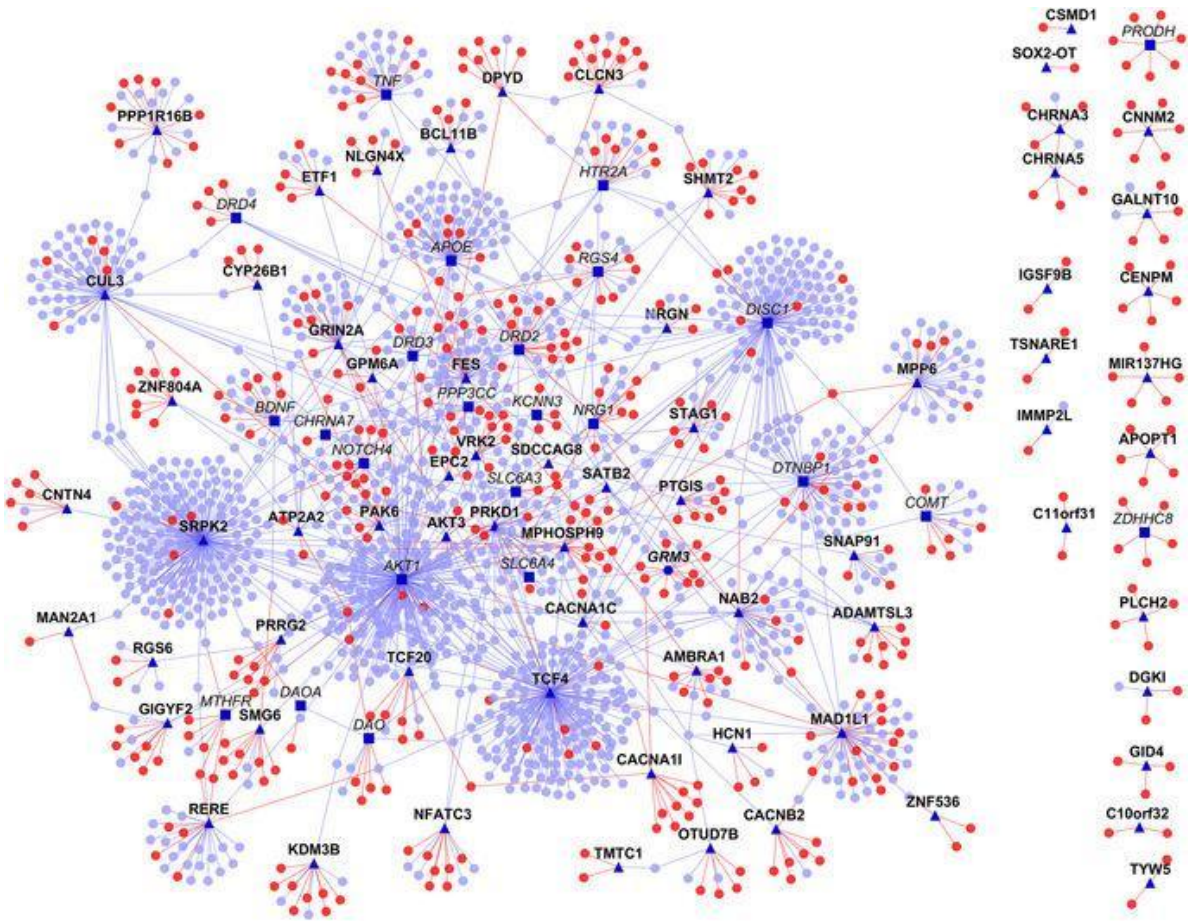
Node predictions
Predict a type or a continuous value for a given node

Link prediction
Predict whether two nodes are linked

Community/module detection
Identify clusters of linked nodes that are alike

Node classification example

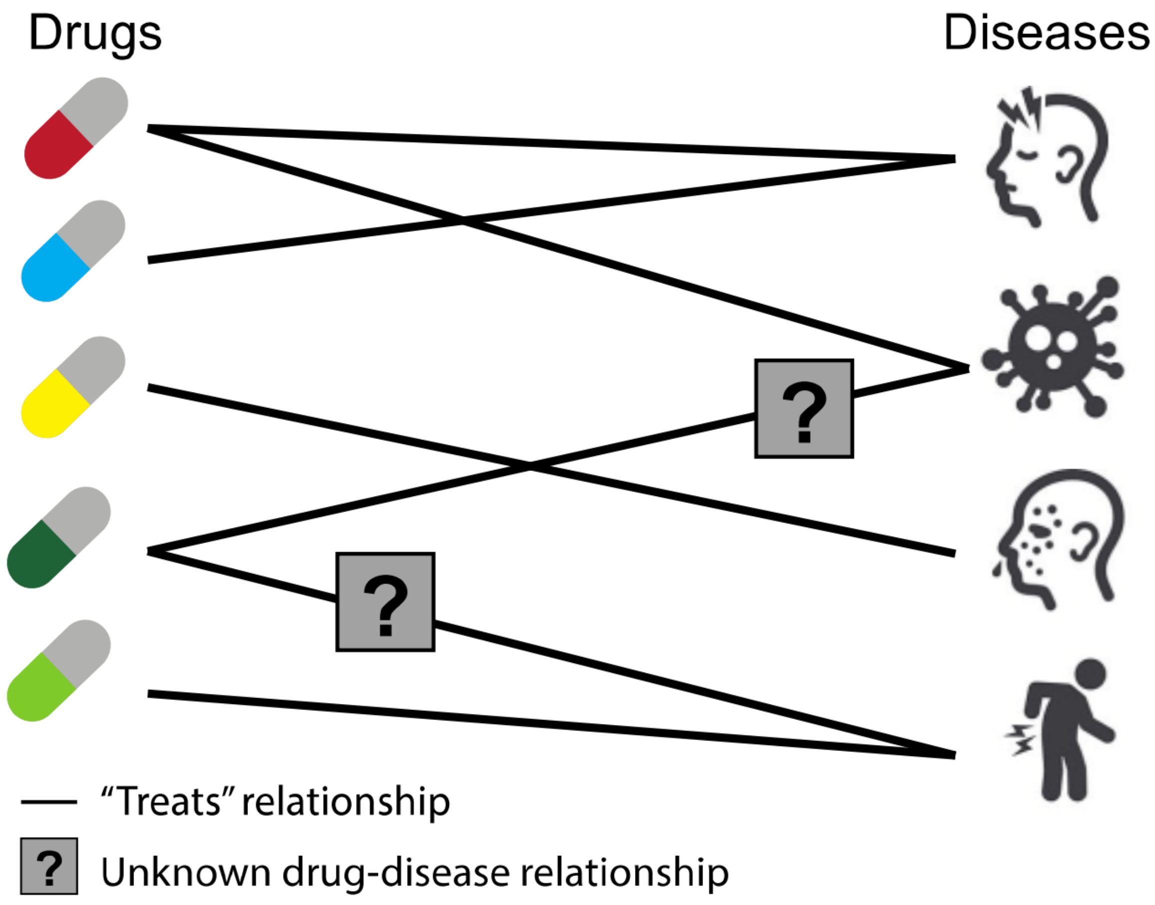
Assign a function to proteins in the interactome



Source: Ganapathiraju et al. Nature 2016

Link prediction example

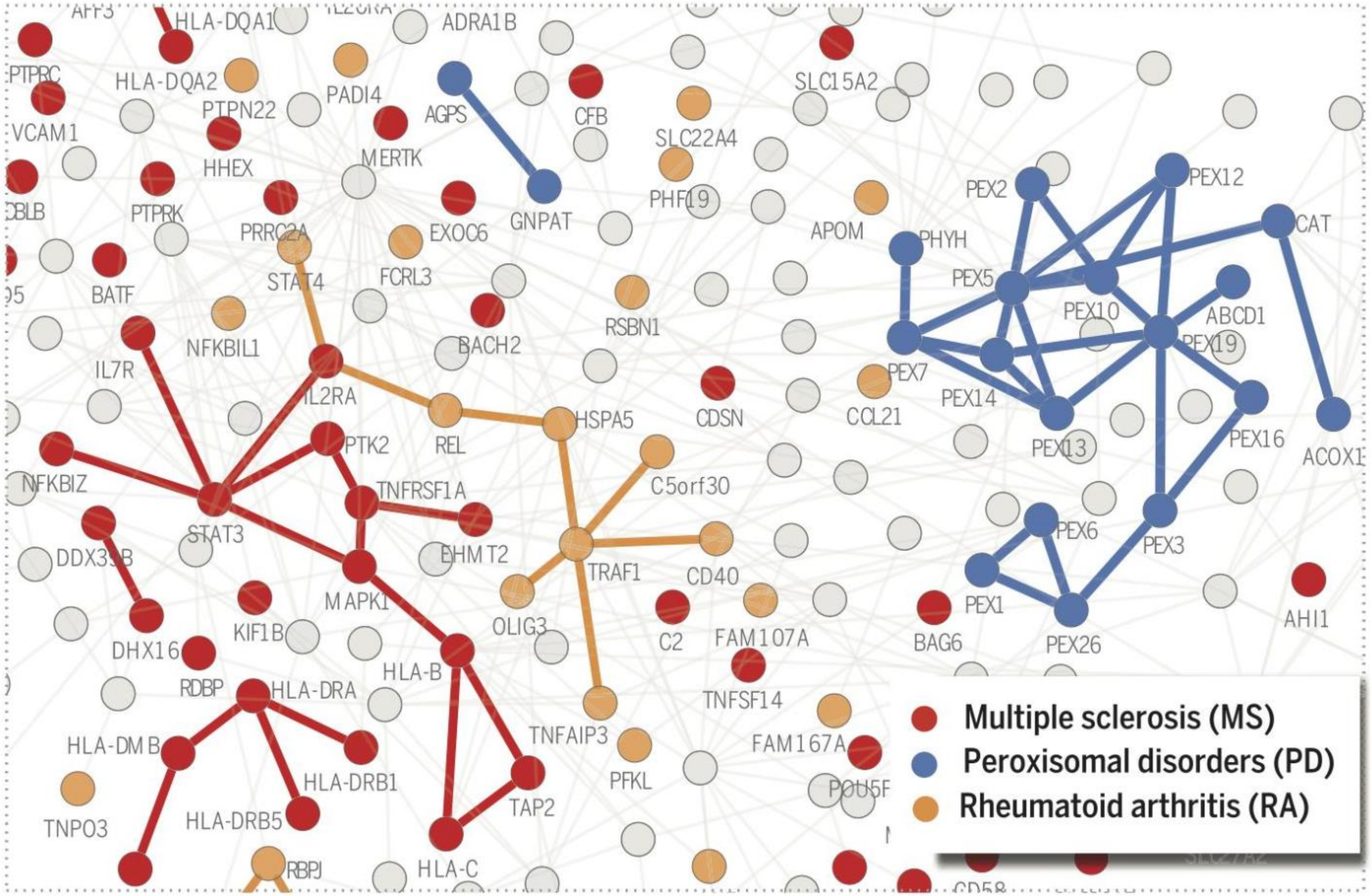
Predict which diseases can be treated by a new molecule



Source: Zitnik et al. 2020

Community prediction example

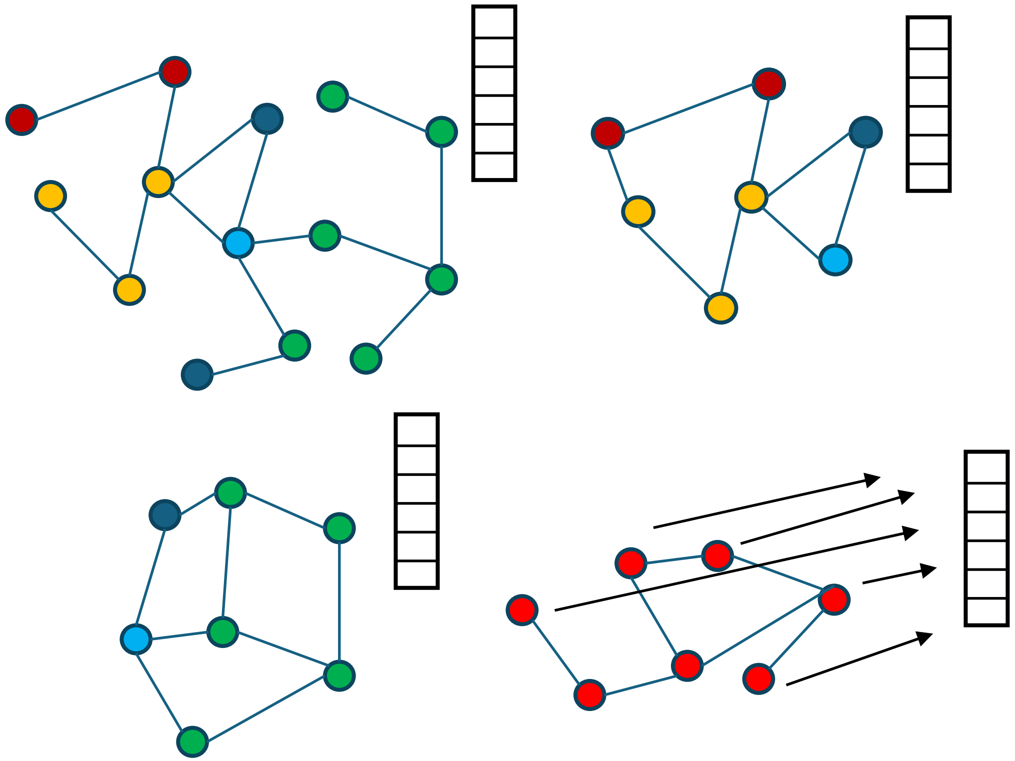
Identify disease proteins in the interactome



Source: Menche et al. Science, 2015

Predictive Tasks

Structure classification/regression



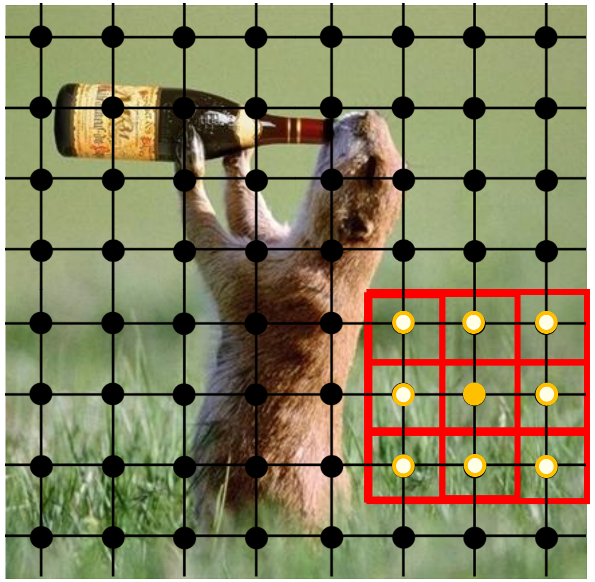
A dataset of i.i.d graphs

Graph classification
Assign whole structure to a specific class

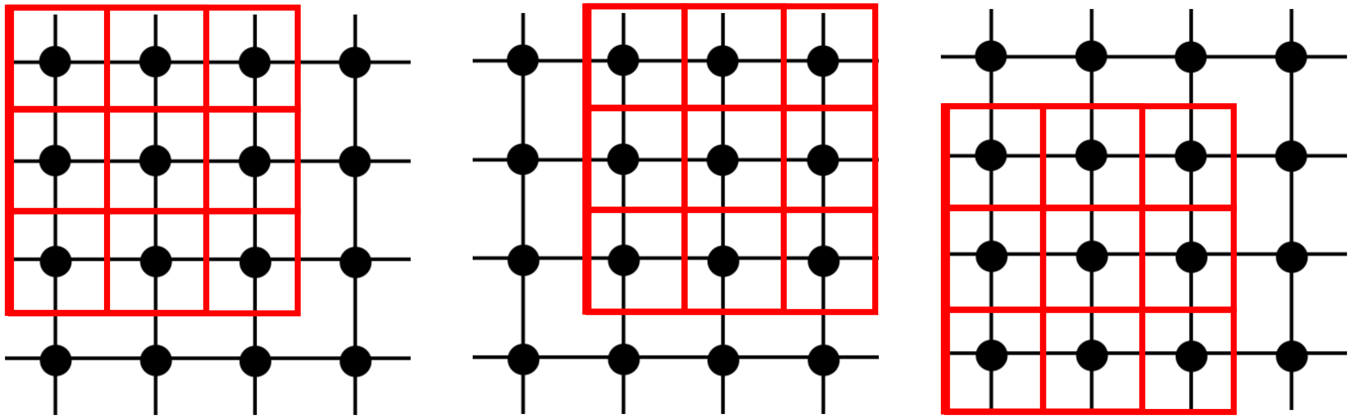
Graph regression
Regress a structure to a value (or a vector of values)

Deep graph networks

A Graph View on (Image) Convolutions



Visual convolutions are graph convolutions on a regular grid

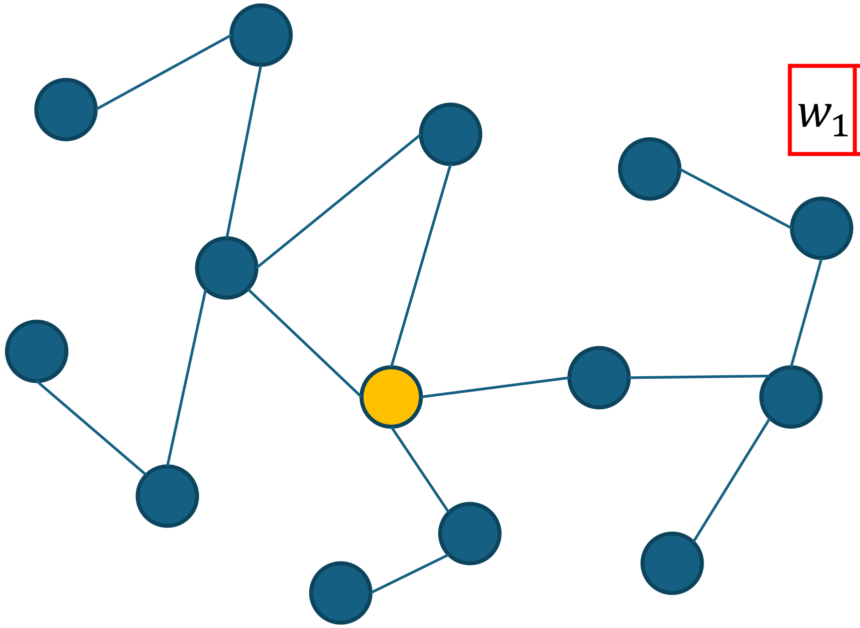


Plus some key assumptions which make it difficult to directly apply them to graphs

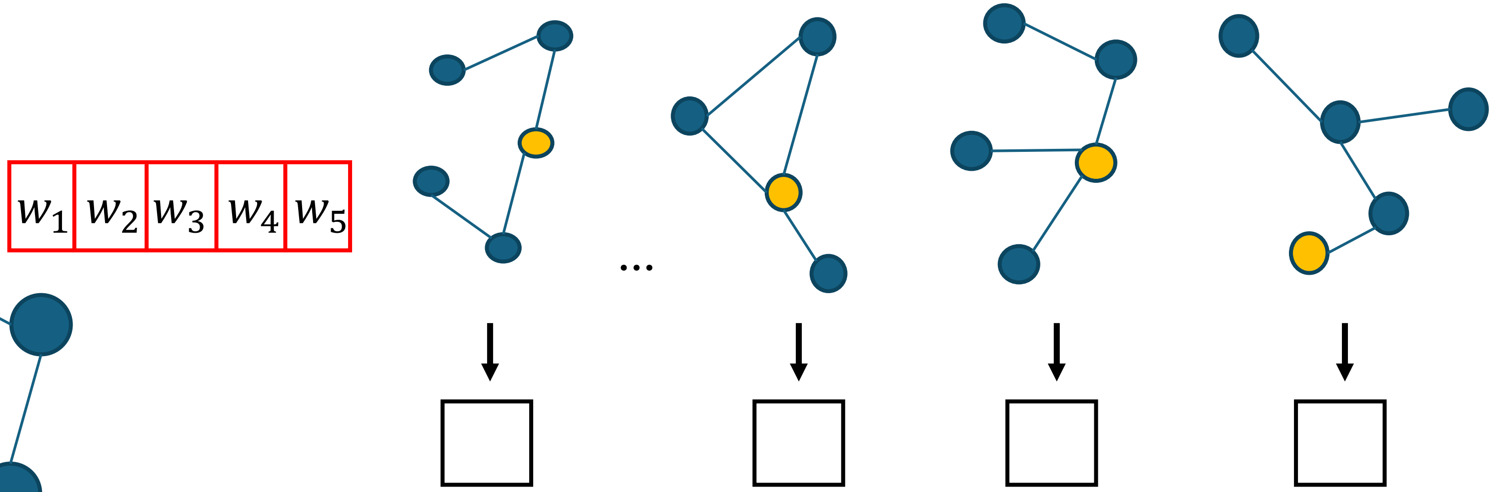
- Regular neighborhood
- Existence of a total node ordering

Node Neighborhoods

Example of 4-neighborhoods

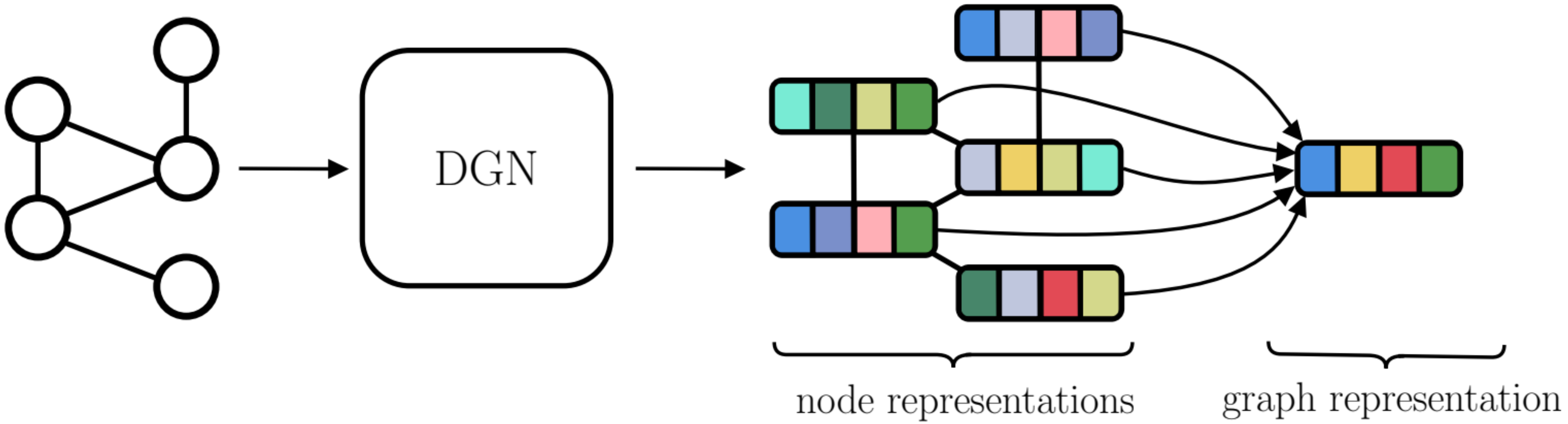


convolutions



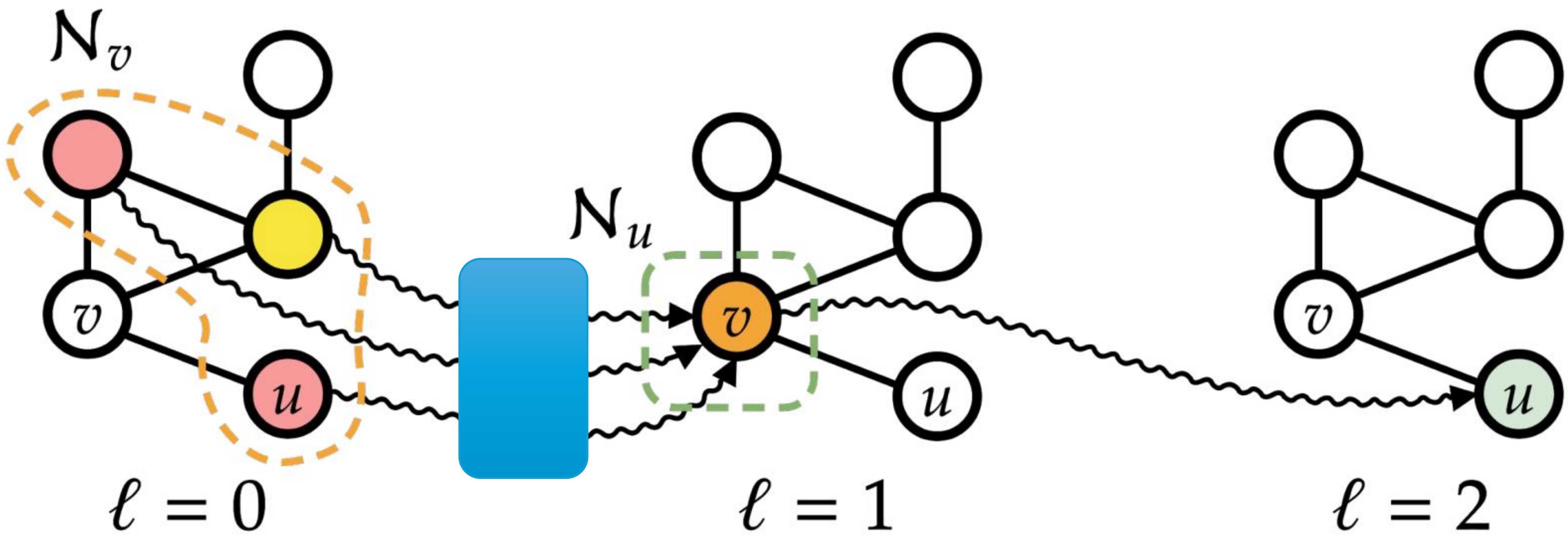
Neighborhoods depend on node ordering: how can I get coherent node ordering across multiple graphs?

Deep Graph Networks - The intuition



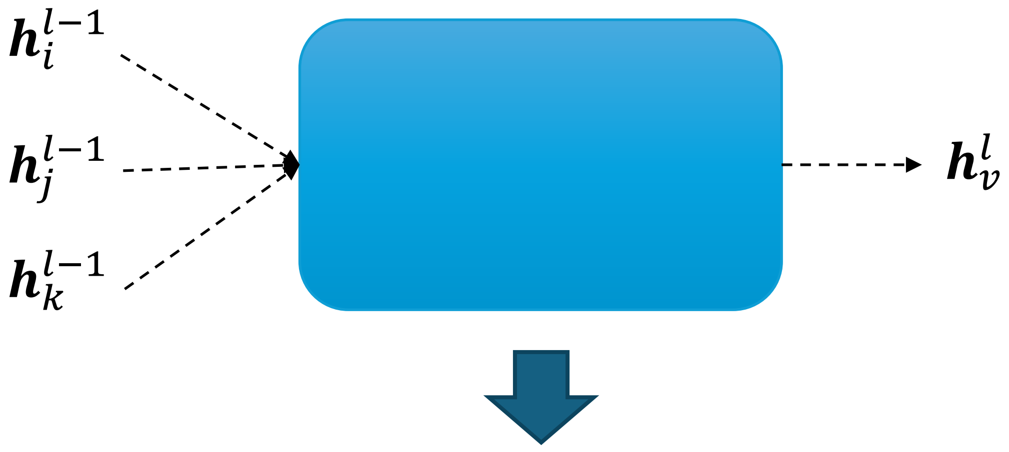
- Encode vertices and the graph itself into a vector space by means of an adaptive (learnable) mapping
- Use the learned encodings to solve predictive, explorative or generative tasks

Neighborhood Aggregation & Layering



What is inside of the Box?

A learning model of course (e.g. a neural network) including an aggregation function to handle size-varying neighborhoods



A simple model

$$\boldsymbol{h}_v^{\mathrm{l}} = \sigma\left(\boldsymbol{W}_l AGG\left(\left\{\boldsymbol{h}_i^{l-1} : i \in N(v)\right\}\right), \widehat{\boldsymbol{W}}_l \boldsymbol{h}_v^{l-1}\right)$$

General Graph Convolutional Layer

$$\underbrace{\mathbf{h}_v^{\ell+1}}_{\text{state}} = \overbrace{\phi^{\ell+1}}^{\text{MLP/Linear}}\Big(\mathbf{h}_v^{\ell},\ \underbrace{\Psi}_{\text{perm. invariant function}}(\{\overbrace{\psi^{\ell+1}}^{\text{MLP/Linear}}(\mathbf{h}_u^{\ell}) \mid u \in \mathcal{N}_v\})\Big)$$

Variants/extensions:

- **Edge-aware convolution**
- **Attention over neighbors**
- **Laplacian-normalized**

Model	Neighborhood Aggregation $\mathbf{h}_v^{\ell+1}$
NN4G [88]	$\sigma\Big({\mathbf{w}^{\ell+1}}^T\mathbf{x}_v + \sum_{i=0}^{\ell}\sum_{c_k\in\mathcal{C}}\sum_{u\in\mathcal{N}_v^{c_k}} w_{c_k}^i * \mathbf{h}_u^i\Big)$
GNN [104]	$\sum_{u\in\mathcal{N}_v} MLP^{\ell+1}\Big(\mathbf{x}_u, \mathbf{x}_v, \mathbf{a}_{uv}, \mathbf{h}_u^{\ell}\Big)$
GraphESN [44]	$\sigma\Big(\mathbf{W}^{\ell+1}\mathbf{x}_u + \hat{\mathbf{W}}^{\ell+1}[\mathbf{h}_{u_1}^{\ell}, \dots, \mathbf{h}_{u_{\mathcal{N}_v}}^{\ell}]\Big)$
GCN [72]	$\sigma\Big(\mathbf{W}^{\ell+1}\sum_{u\in\mathcal{N}(v)}\mathbf{L}_{vu}\mathbf{h}_u^{\ell}\Big)$
GAT [120]	$\sigma\Big(\sum_{u\in\mathcal{N}_v}\alpha_{uv}^{\ell+1} * \mathbf{W}^{\ell+1}\mathbf{h}_u\Big)$
ECC [111]	$\sigma\Big(\frac{1}{\lvert\mathcal{N}_v\rvert}\sum_{u\in\mathcal{N}_v} MLP^{\ell+1}(\mathbf{a}_{uv})^T\mathbf{h}_u^{\ell}\Big)$
R-GCN [105]	$\sigma\Big(\sum_{c_k\in\mathcal{C}}\sum_{u\in\mathcal{N}_v^{c_k}}\frac{1}{\lvert\mathcal{N}_v^{c_k}\rvert}\mathbf{W}_{c_k}^{\ell+1}\mathbf{h}_u^{\ell} + \mathbf{W}^{\ell+1}\mathbf{h}_v^{\ell}\Big)$
GraphSAGE [54]	$\sigma\Big(\mathbf{W}^{\ell+1}(\frac{1}{\lvert\mathcal{N}_v\rvert}[\mathbf{h}_v^{\ell}, \sum_{u\in\mathcal{N}_v}\mathbf{h}_u^{\ell}])\Big)$
CGMM [3]	$\sum_{i=0}^{\ell} w^i * \Big(\sum_{c_k\in\mathcal{C}} w_{c_k}^i * (\frac{1}{\lvert\mathcal{N}_v^{c_k}\rvert}\sum_{u\in\mathcal{N}_v^{c_k}}\mathbf{h}_u^i)\Big)$
GIN [131]	$MLP^{\ell+1}\Big((1+\epsilon^{\ell+1})\mathbf{h}_v^{\ell} + \sum_{u\in\mathcal{N}_v}\mathbf{h}_u^{\ell}\Big)$

A Message-Passing view on Deep Graph Networks

Algorithm 13.1: Simple message-passing neural network

Input: Undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
Initial node embeddings $\{\mathbf{h}_n^{(0)} = \mathbf{x}_n\}$
$\mathrm{Aggregate}(\cdot)$ function
$\mathrm{Update}(\cdot, \cdot)$ function
Output: Final node embeddings $\{\mathbf{h}_n^{(L)}\}$

```
// Iterative message-passing
```

for $l \in \{0, \ldots, L-1\}$ **do**
$\quad \mathbf{z}_n^{(l)} \leftarrow \mathrm{Aggregate}\left(\left\{\mathbf{h}_m^{(l)} : m \in \mathcal{N}(n)\right\}\right)$
$\quad \mathbf{h}_n^{(l+1)} \leftarrow \mathrm{Update}\left(\mathbf{h}_n^{(l)}, \mathbf{z}_n^{(l)}\right)$
end for
return $\{\mathbf{h}_n^{(L)}\}$

Graph Isomorphism Network

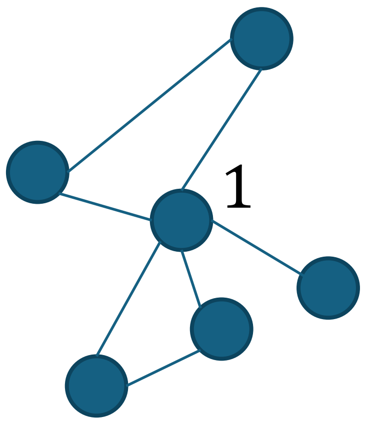
Xu et al, ICLR 2019

- ❖ A study of GNN expressivity
- ❖ Choice of aggregation functions influences what structures can be recognized
- ❖ Propose a simple aggregation and concatenation model

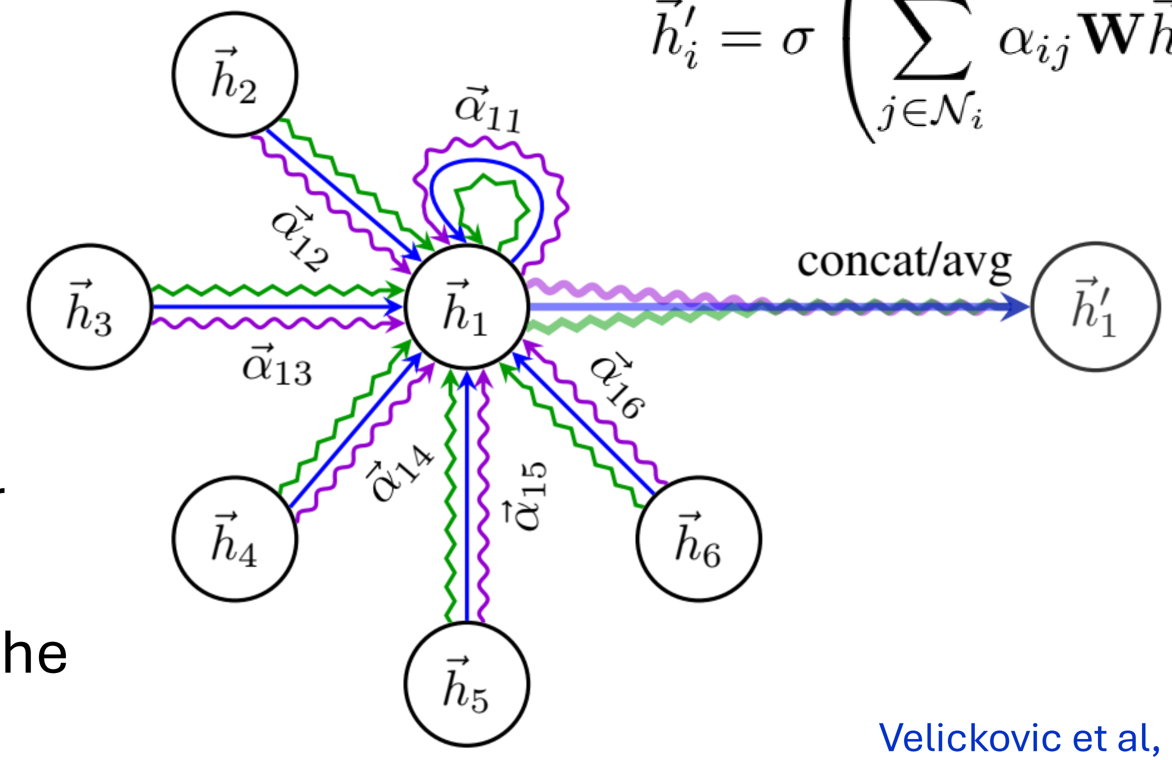
$$h_v^{(k)} = \mathrm{MLP}^{(k)} \left((1 + \epsilon^{(k)}) \cdot h_v^{(k-1)} + \sum_{u \in \mathcal{N}(v)} h_u^{(k-1)} \right)$$

$$h_G = \mathrm{CONCAT}(\mathrm{READOUT}\left(\{h_v^{(k)} | v \in G\} \right) | k = 0, 1, \cdots, K)$$

Graph Attention



$$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right)$$

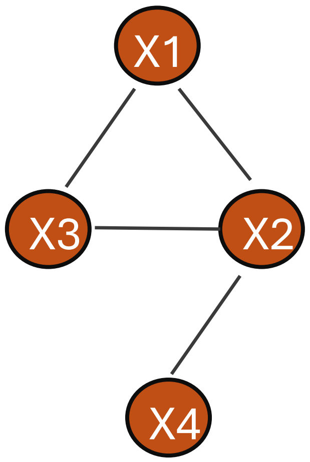


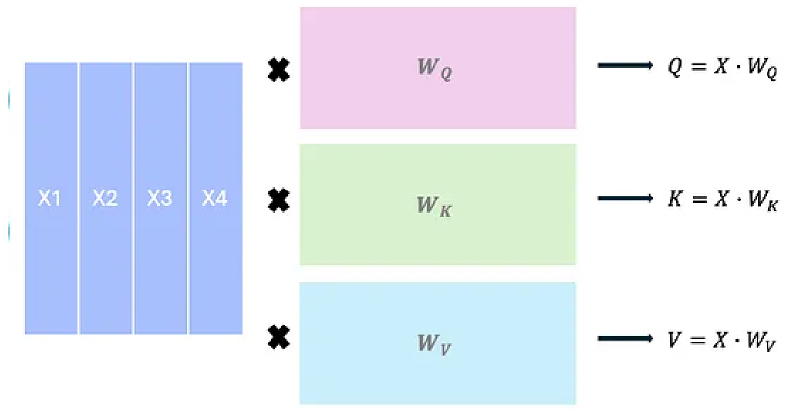
Learning to weight contribution of other nodes when aggregating to form the node embedding

Velickovic et al, ICLR 2018

Global Graph Attention

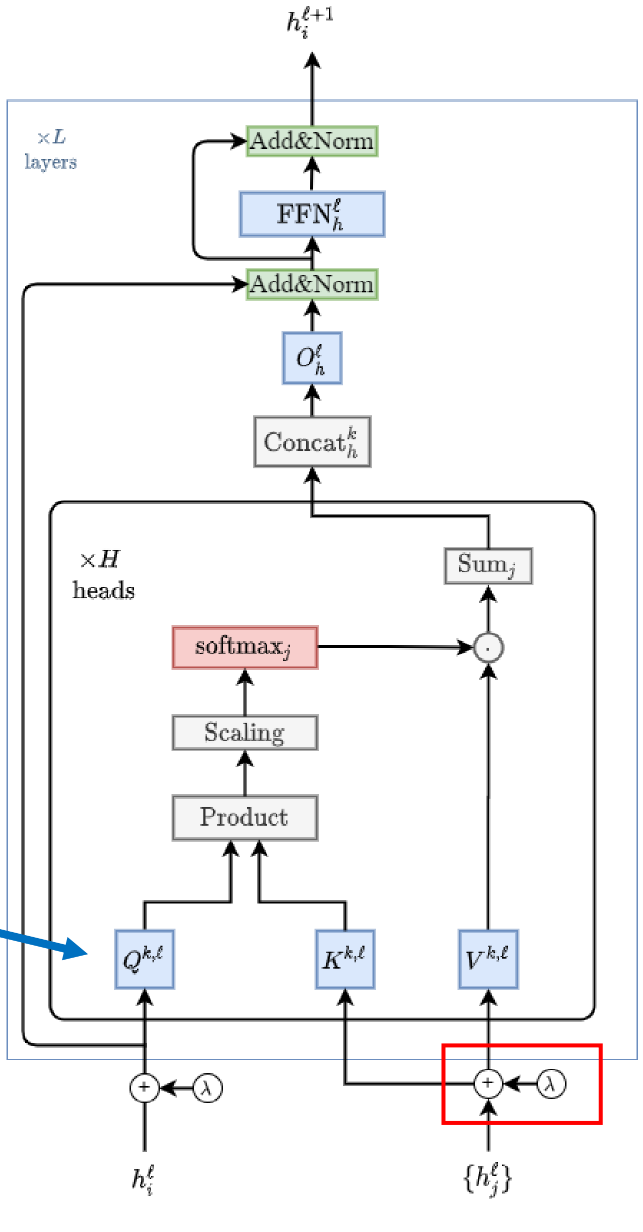
A direct generalization of transformer attention from sequence tokens to nodes





$$Attention(Q, K, V) = \text{softmax}\left(\frac{Q\,K^T}{\sqrt{d_k}}\right)V$$

Img adapted from Medium

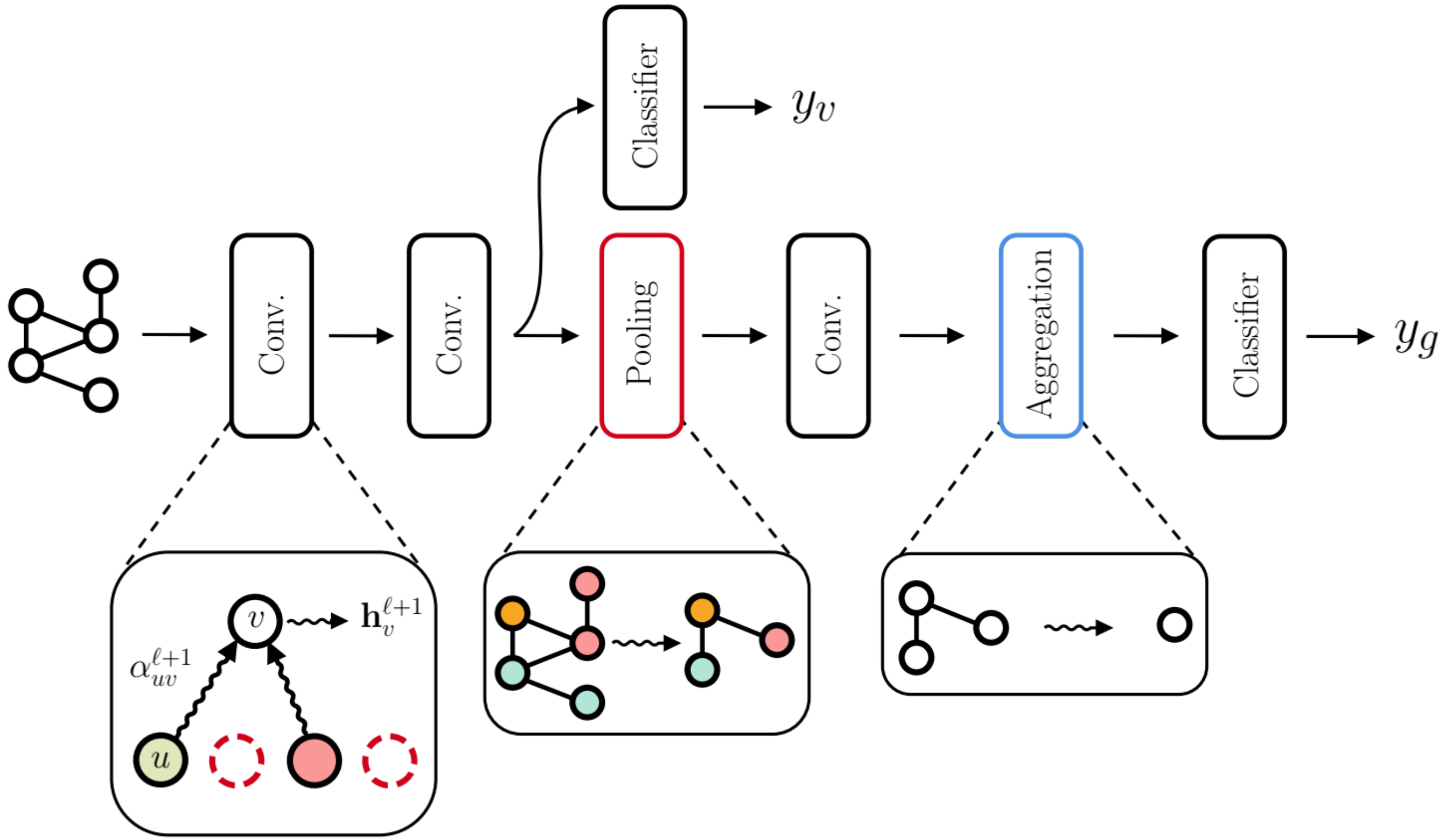


Graph Transformer Layer

Dwivedi & Bresson, AAAI-WS 2021

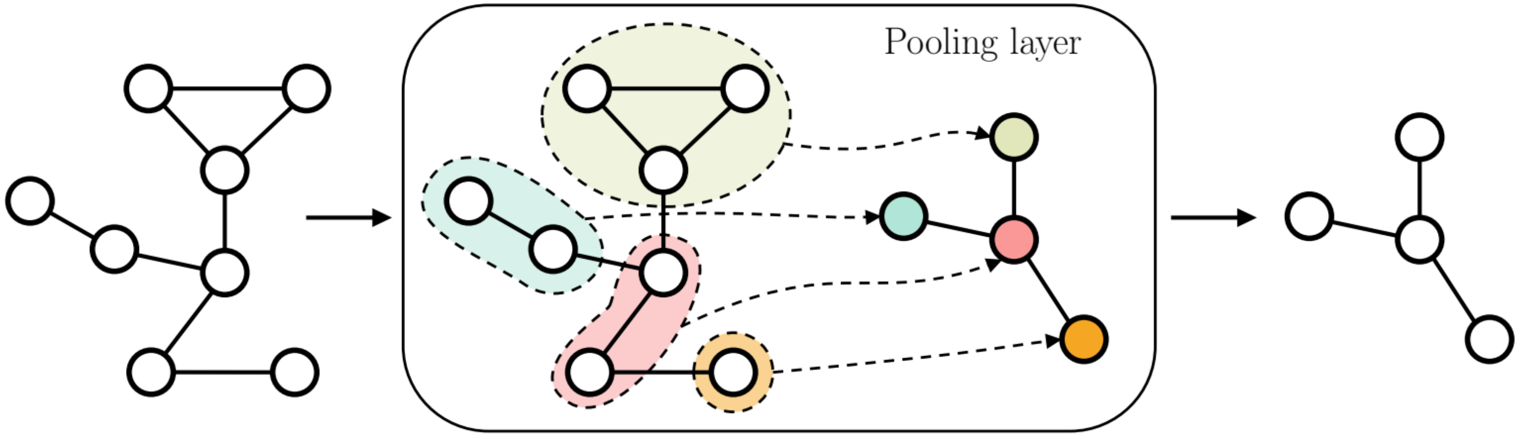
positional encodings need to be used to reintroduce structural bias

Deep Graph Networks - The Complete Picture

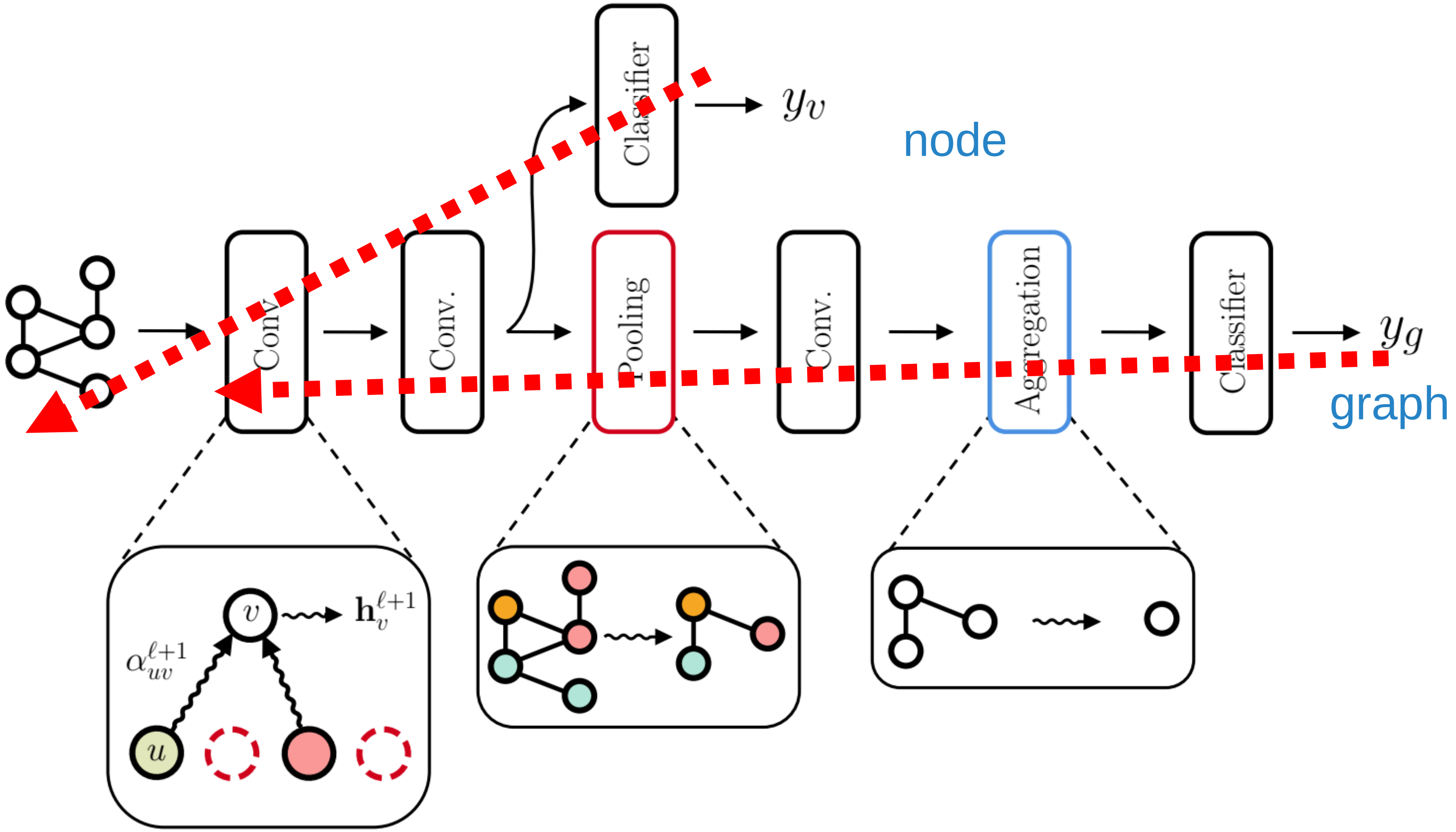


What About Pooling?

- Standard aggregation operates of predefined node subsets
- Ignore community/hierarchical structure in the graph
- Need graph coarsening (pooling) operators
  - Differentiable   Rex Ying et al, NIPS 2018
  - Graph theoretical   Bacciu et al, AAAI 2023
  - Graph signature



Training

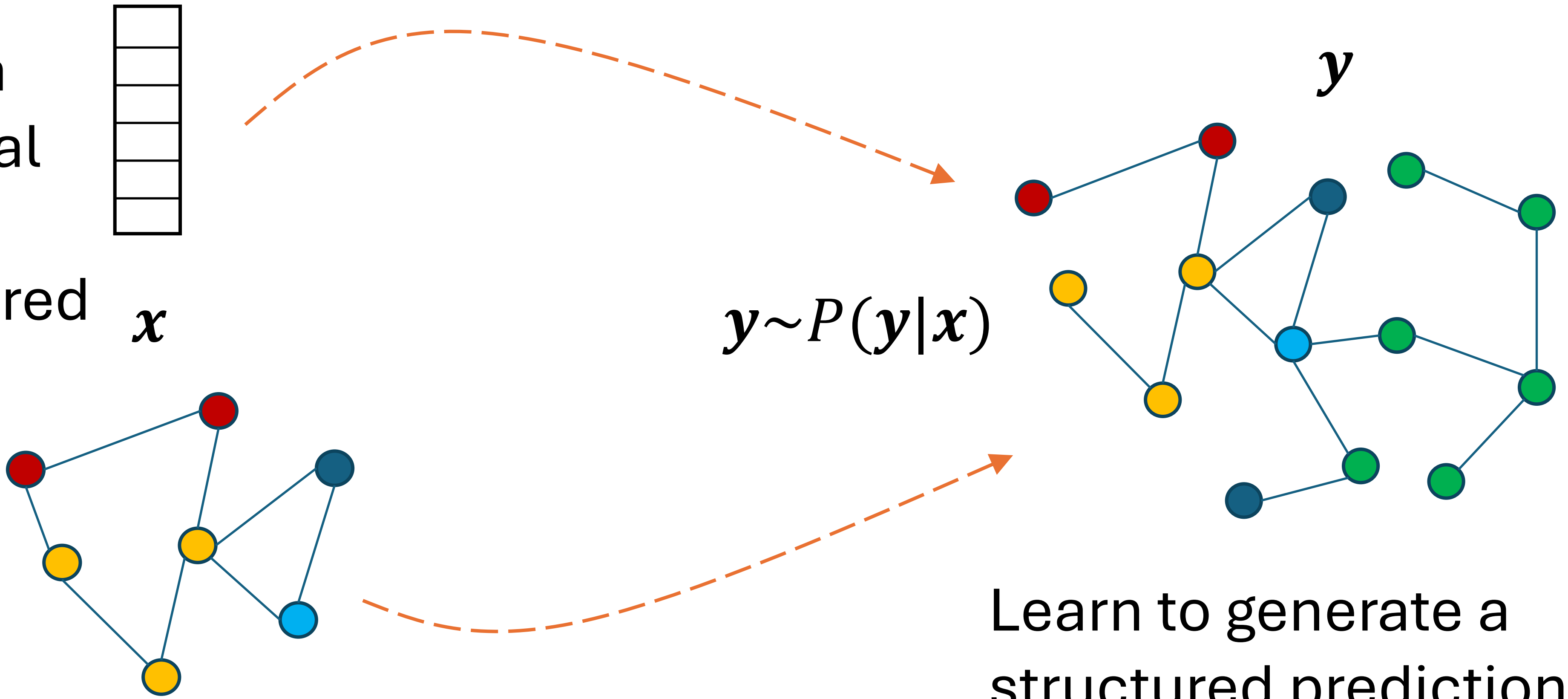


Backpropagate from the (graph or node level) error computed from the top layer embeddings to the early layers

Beyond Graph Prediction

Transductive tasks

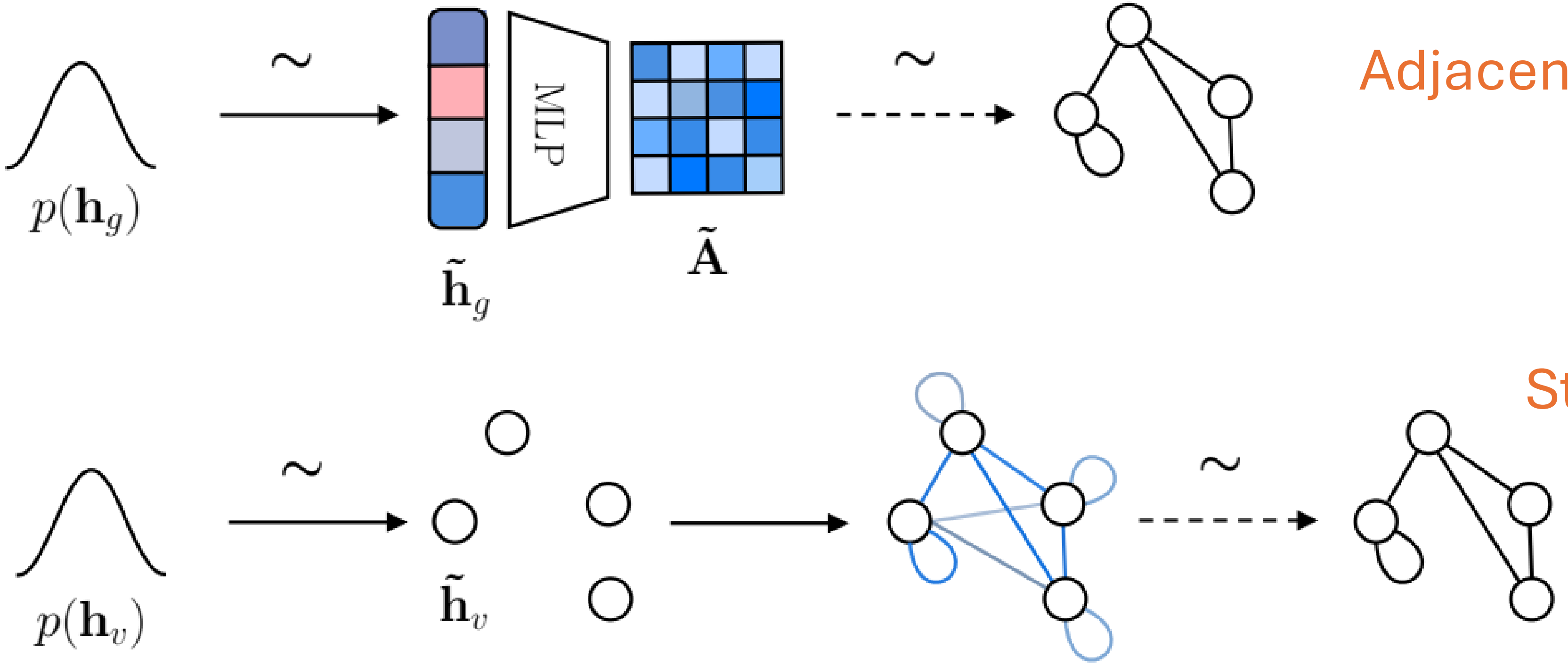
Given a vectorial and/or structured input



Learn to generate a structured prediction

Graph Generation

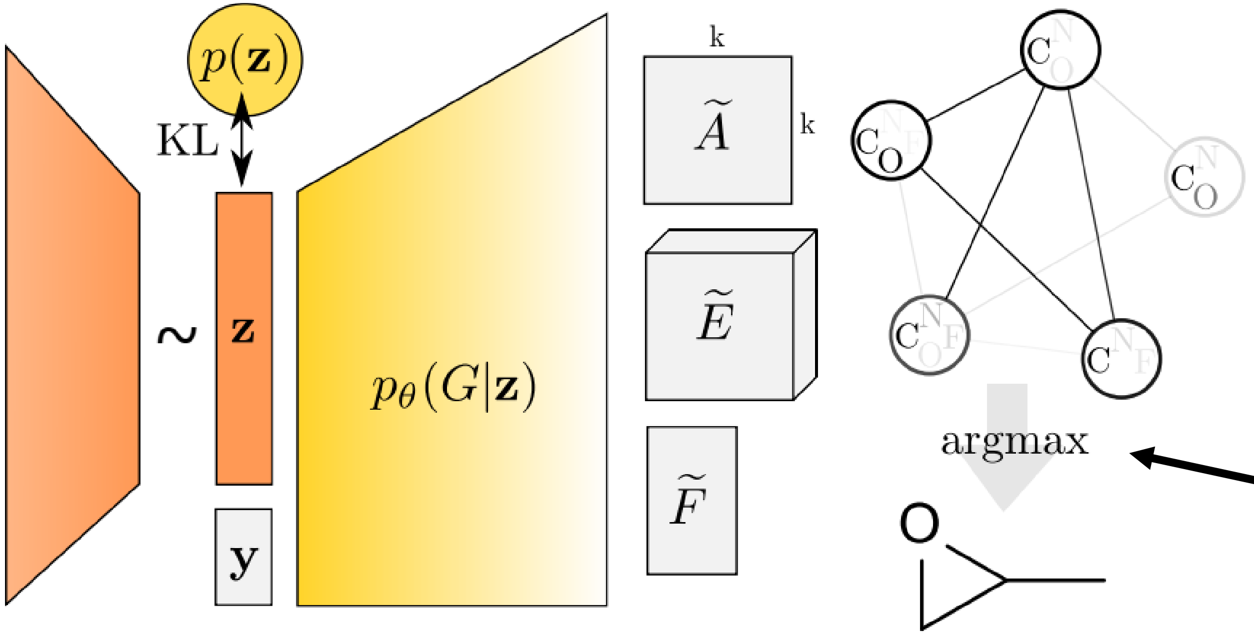
Generate a prediction that is itself a graph



Adjacency-based

Structure-based

Graph Autoencoder

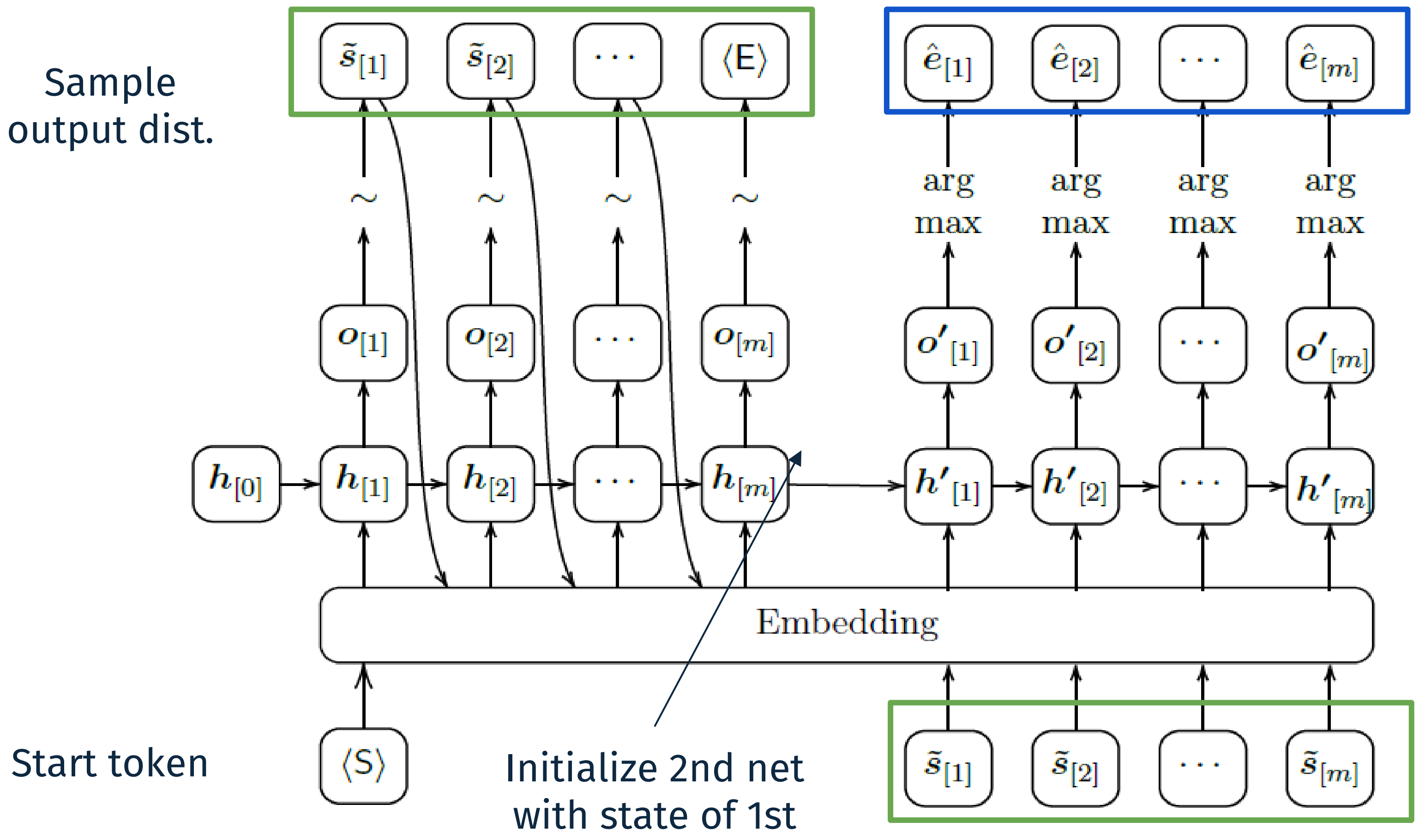


GraphVAE generates adjacency matrix up to k vertices along with the relevant edge/node features (for molecular data)

Argmax a.k.a. sampling $\Longrightarrow$ non-differentiable

Simonovsky, Komodakis, ICLR-WS 2018

Language-Based Graph Generation

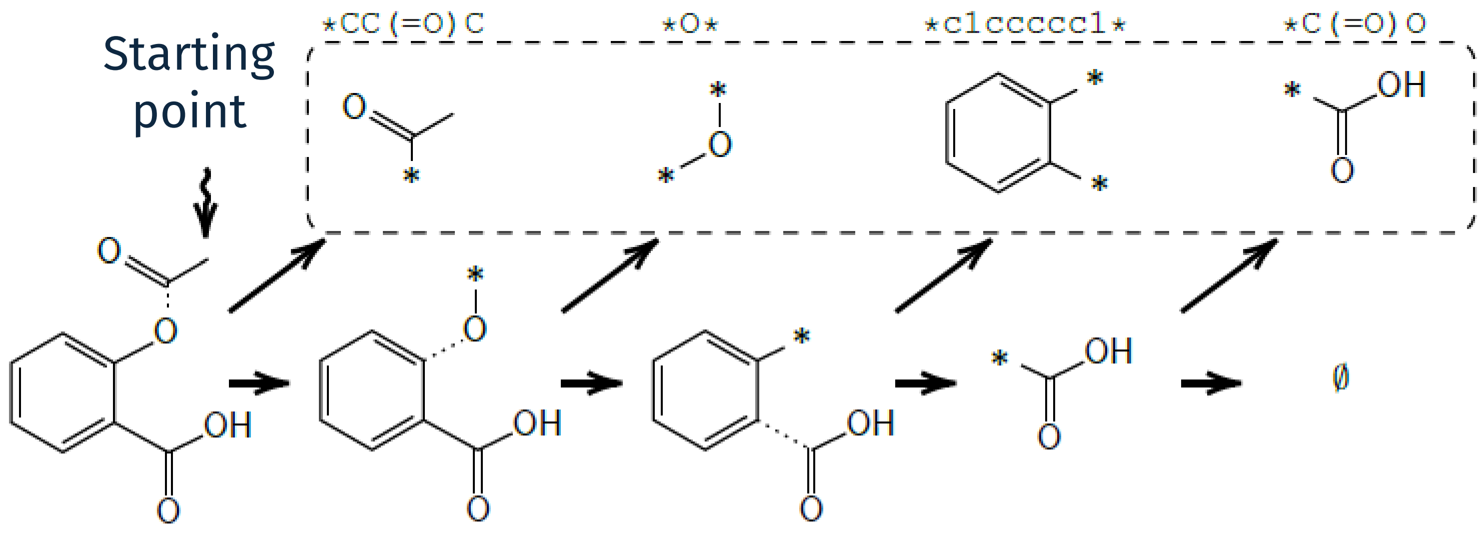


Generate a graph node-by-node and edge-by-edge through a language model

Bacciu et al, Neurocomputing 2020

Generate Molecules by Fragmentation

- ✓Molecule is scanned in SMILES order
- ✓Find first breakable bond
- ✓Break the molecule at that bond, set aside leftmost fragment
- ✓Proceed recursively on rightmost fragment



✓Order is **deterministic** and the molecule can be reconstructed

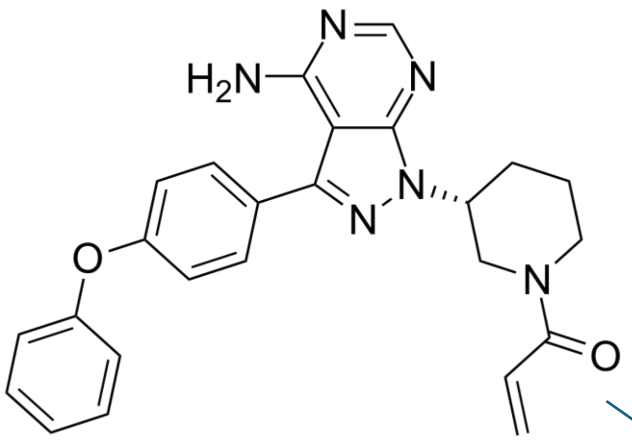
✓Keep a **vocabulary of all possible fragments** found in a dataset

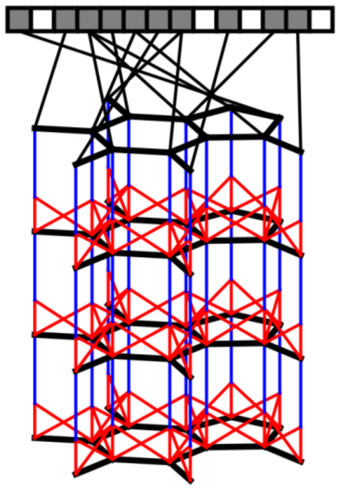
✓Graphs are transformed into **fragment sequences**

Podda et al, AISTATS 2020

Application cases

Predicting Properties of Chemical Compounds





$\sim 10^{-2}$ seconds

Simulation methods

$\sim 10^{3}$ seconds

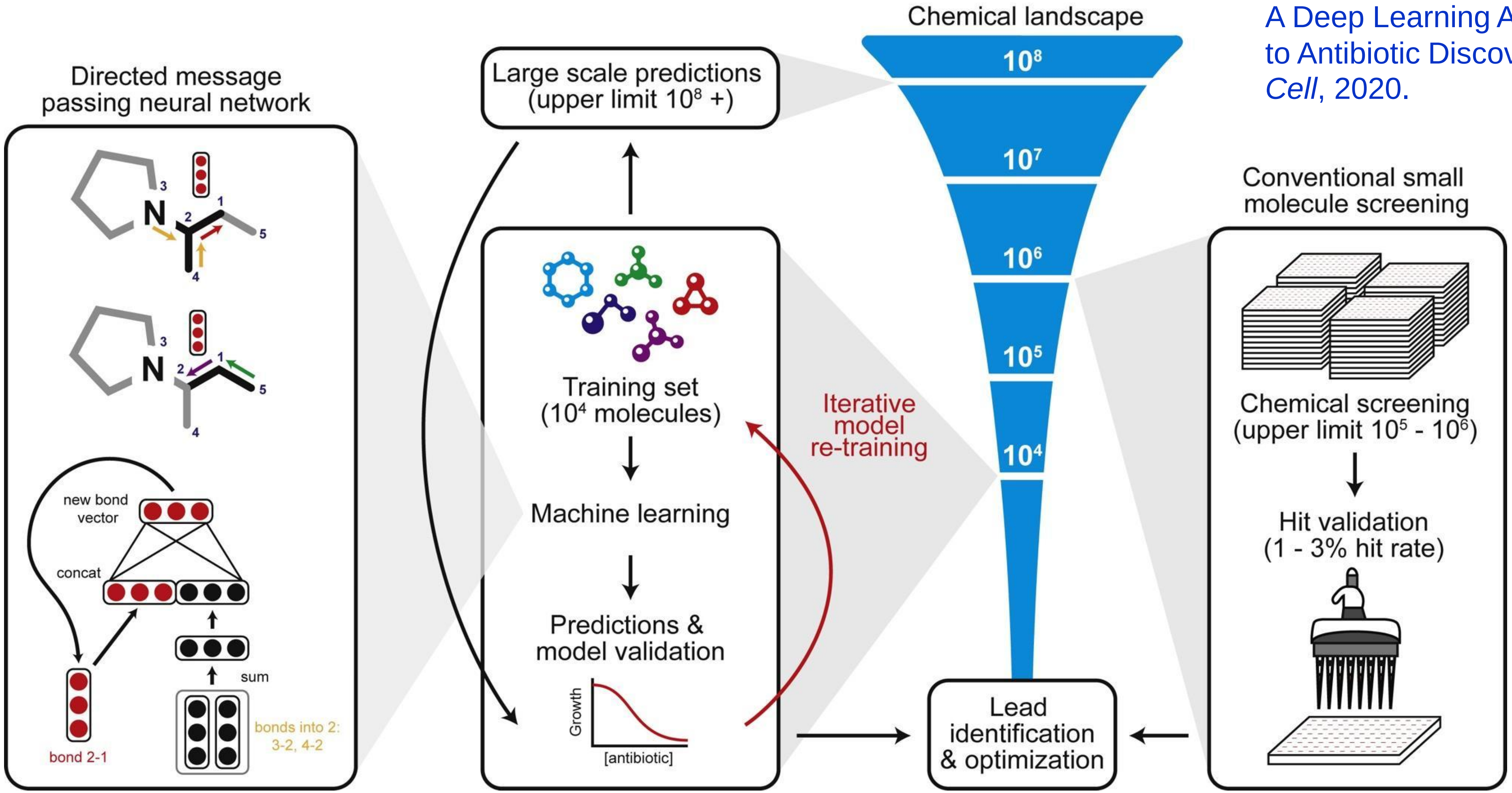
Toxicity

Solubility

Quantum mechanical properties

Gilmer et al, ICML 2017

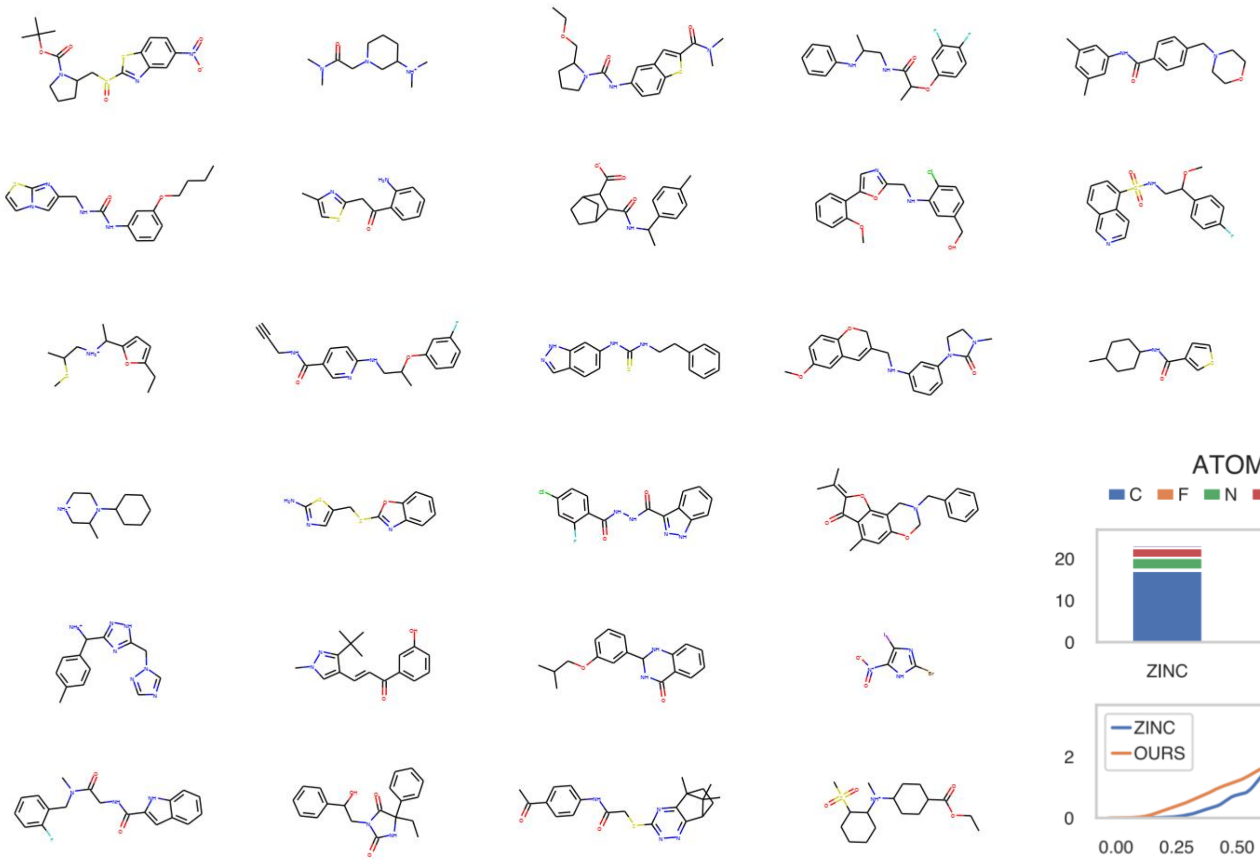
A molecular discovery pipeline



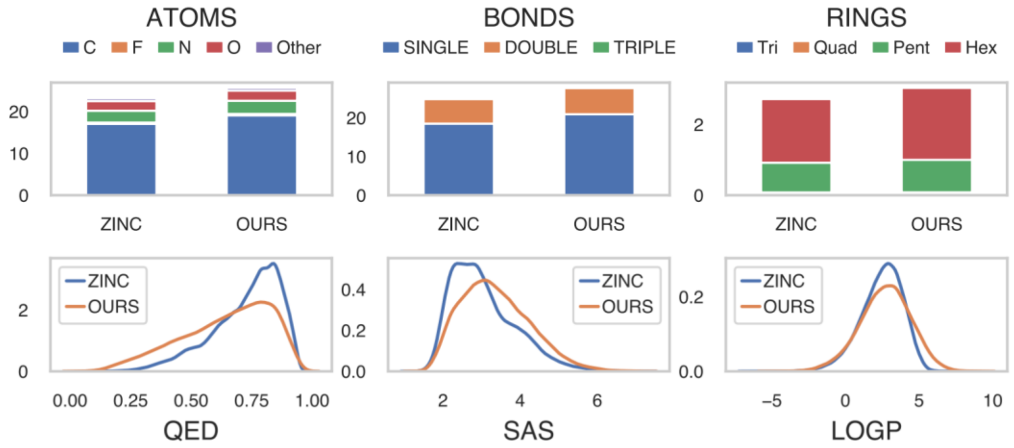
A Deep Learning Approach to Antibiotic Discovery, *Cell*, 2020.

Generating Molecules

Podda, Bacciu, Micheli, AISTATS 2020

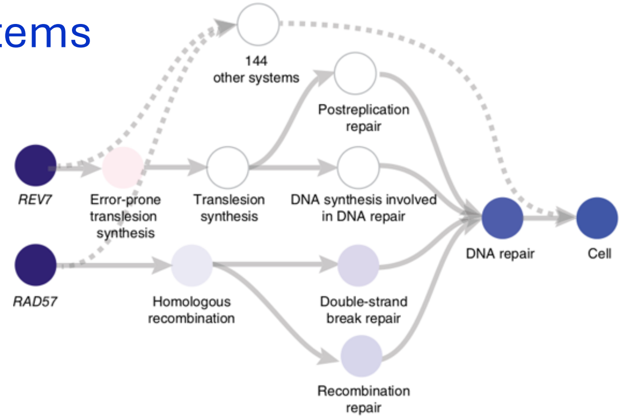


Fragment-based deep molecule generation

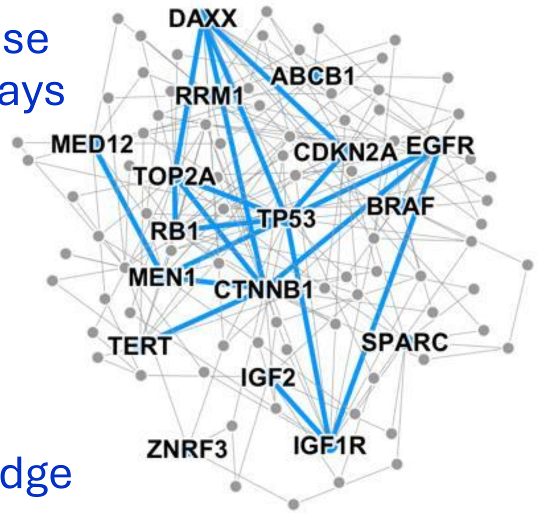


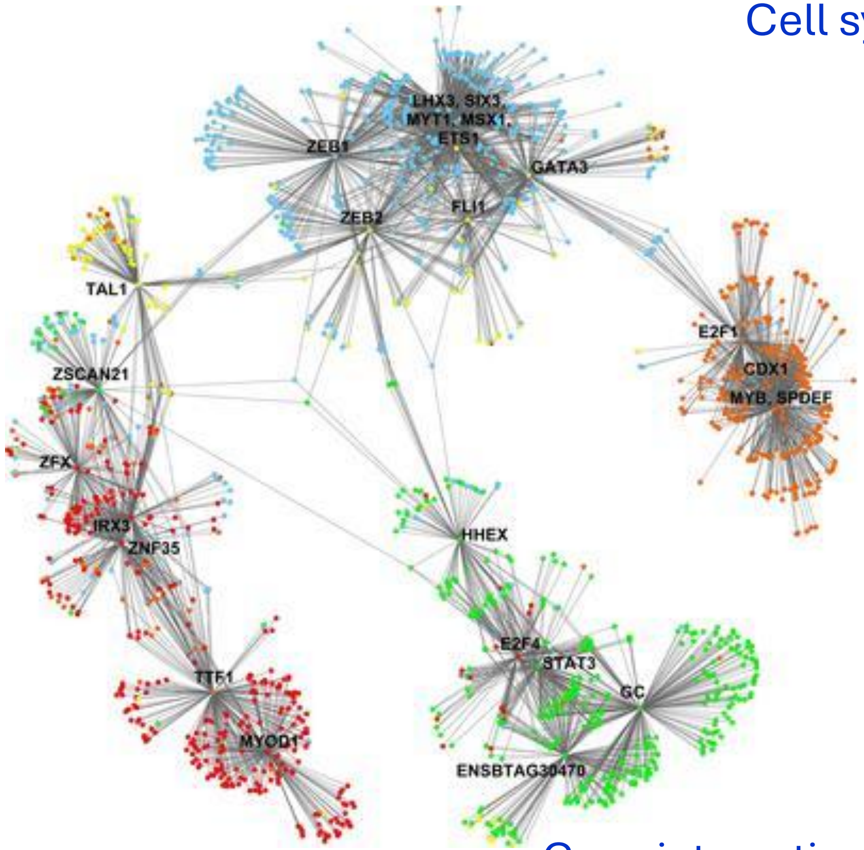
Graphs/Networks are everywhere

Cell systems



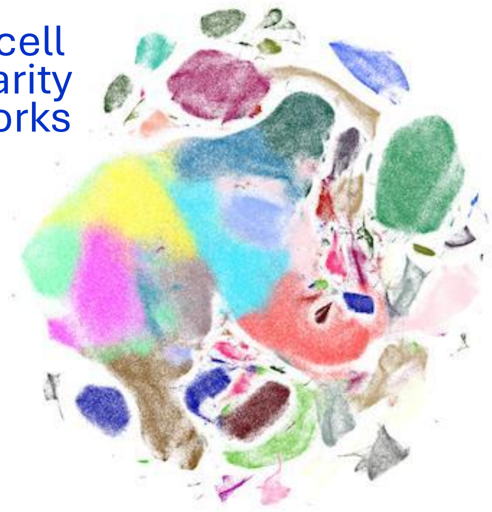
Disease pathways



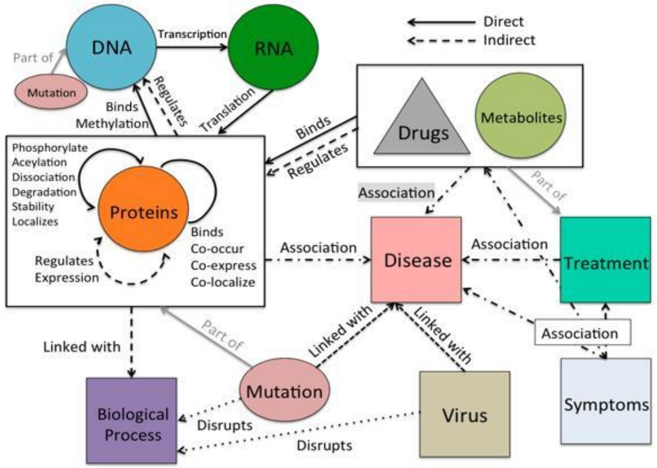


Gene interaction networks

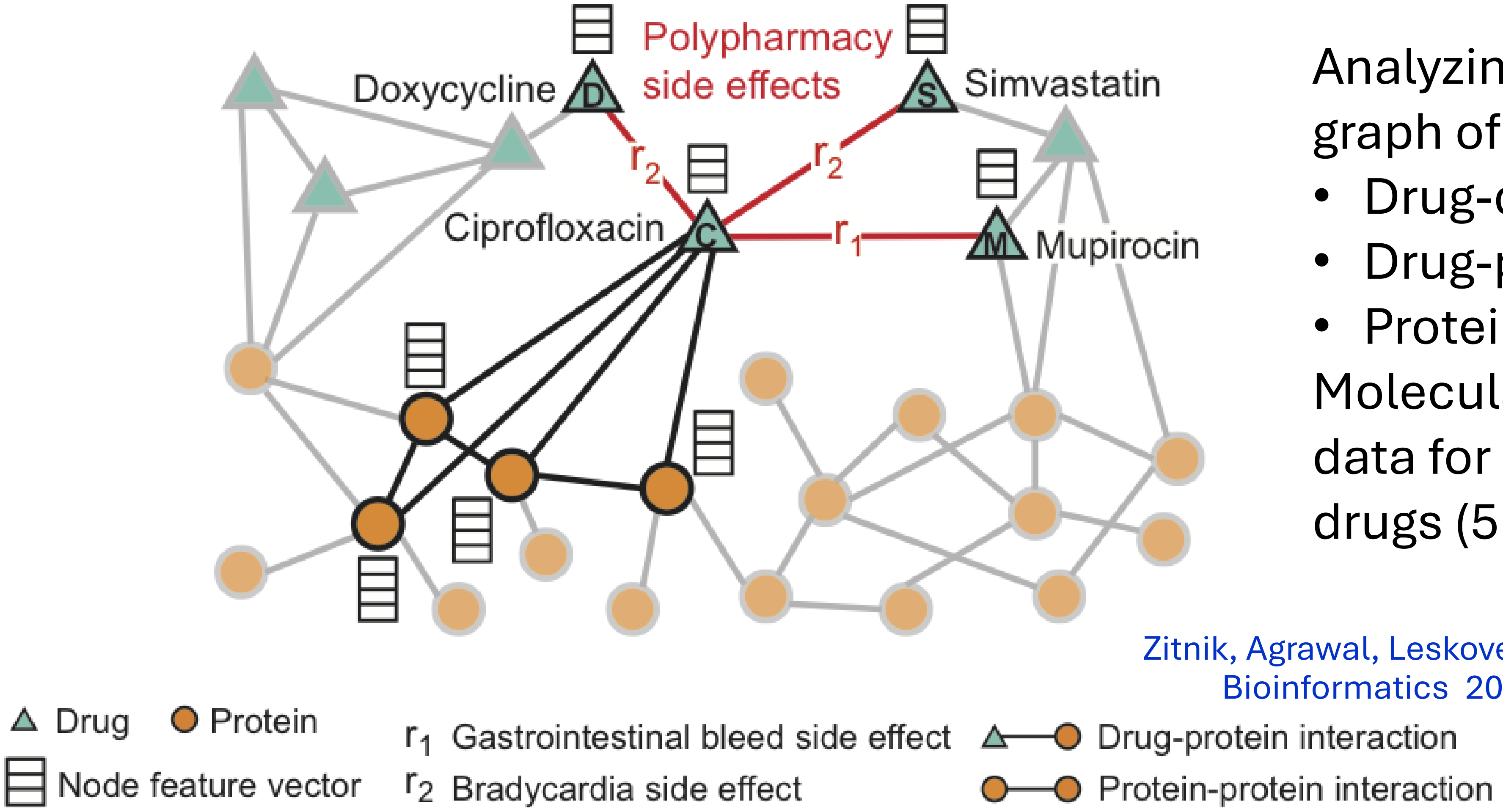
Cell-cell similarity networks



Knowledge graphs



Side Effects of Drug Combinations



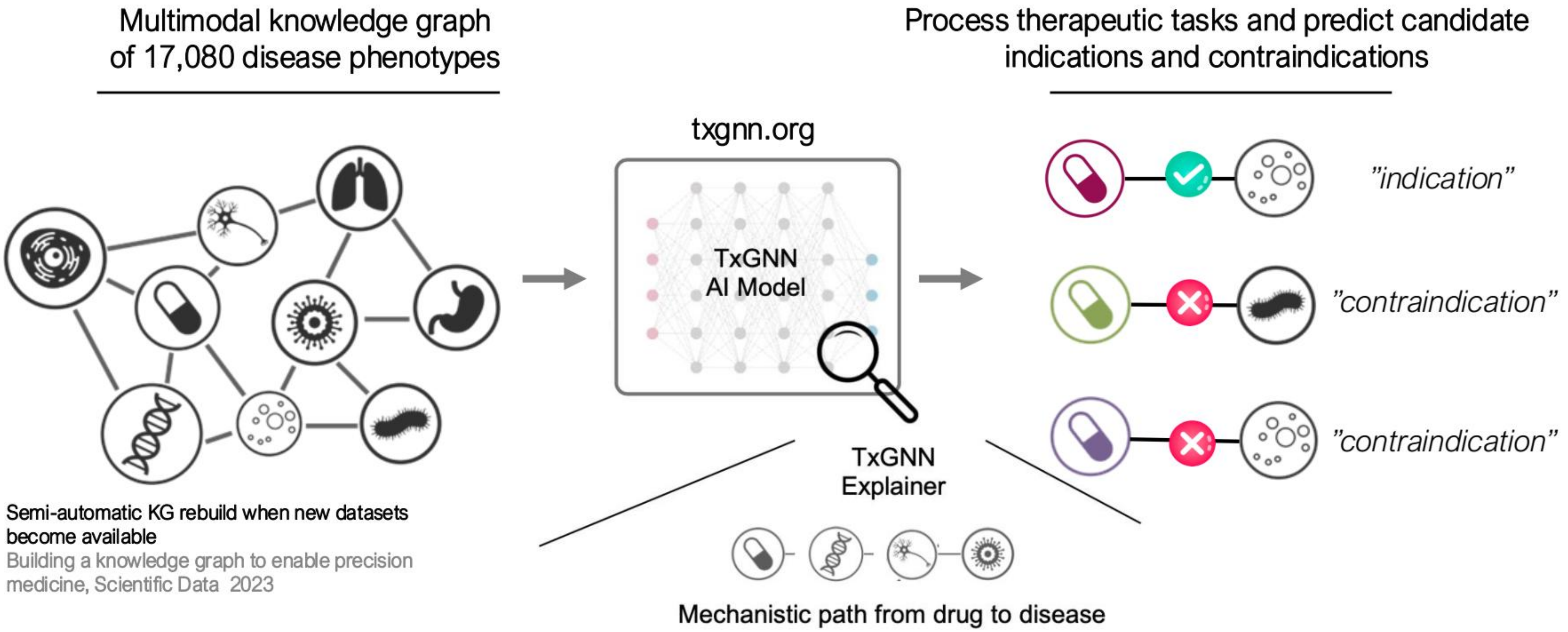
Analyzing a multimodal graph of interactions

- Drug-drug
- Drug-protein
- Protein-protein

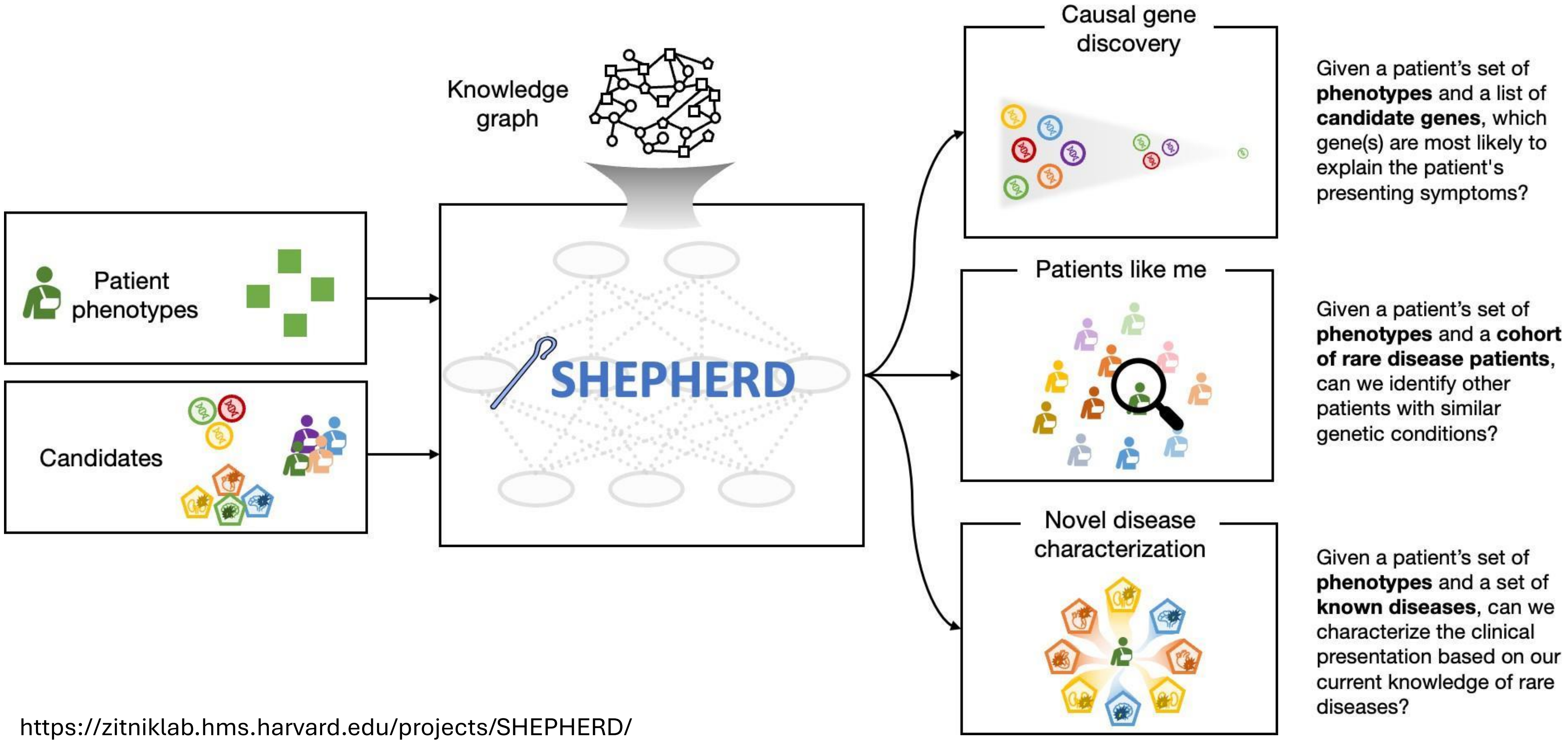
Molecular, drug, and patient data for all US-approved drugs (5M edges)

Zitnik, Agrawal, Leskovec, Bioinformatics 2018

Drug Repurposing

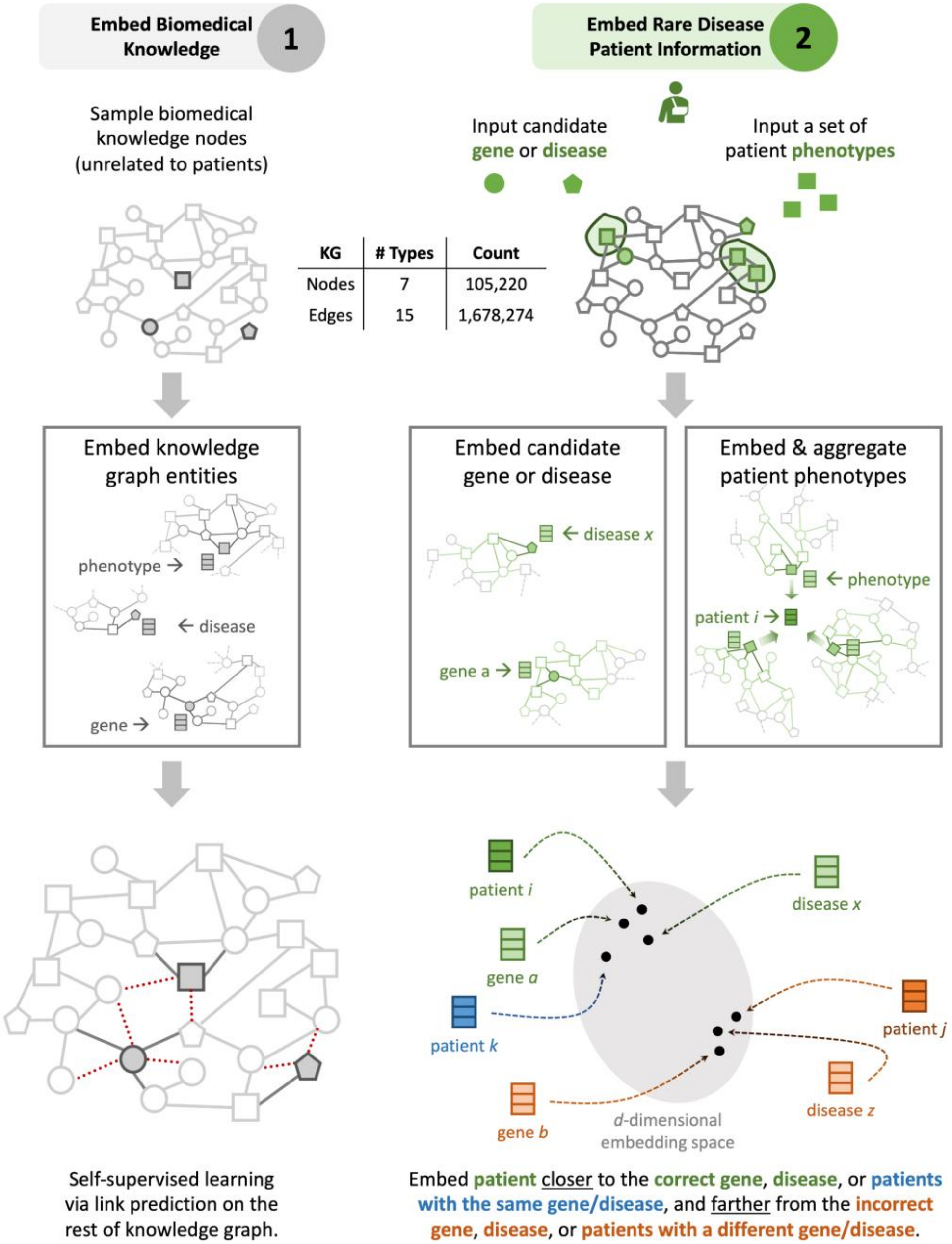


Knowledge-Based rare disease diagnosis



https://zitniklab.hms.harvard.edu/projects/SHEPHERD/

SHEPERD – Graph processing pipeline



https://huggingface.co/spaces/emilyalsentzer/SHEPHERD

Software

You can find most of the foundational models in this lecture implemented here

PyTorch geometric

DeepGraphLibrary

Structured Biobanks

ChEMBL

includes 1.6M assays
covering 2.4M compounds

Bgee

includes 31,467 bulk and
single-cell RNA-seq libraries

STRING

includes 20B interactions
between 59.3M proteins

GENEONTOLOGY Unifying Biology

includes 6M gene annotations
derived from 150K publications

reactome

includes 2,711 pathways
manually curated by PhDs

DRUGBANK

includes 17K FDA-approved
and experimental drugs

NIH National Library of Medicine National Center for Biotechnology Information

includes annotations for 192K
human genetic elements

SIDER

includes 139K adverse
reactions for marketed drugs

human phenotype ontology

includes 13K phenotypes and
156K disease annotations

Wrap-up

Take Home Lessons

- Deep learning for graphs is a now a consolidated research area
  - DGNs as natural extensions of convolutional and recurrent architectures to graphs
  - A candidate AI model for the integration of symbolic knowledge, numerical data and reasoning
- First wave of works (now almost over?) focusing mainly on
  - Different ways of implementing message passing and aggregation on static graphs
  - Graph reductions and pooling
  - Expressivity properties associated with different aggregation functions
  - Efficiency and efficacy of context creation and propagation
- New wave of works focusing on
  - Dynamic graphs
  - DGNs as dynamical systems and their physical interpretation
  - Learning and aligning with (graph) algorithms
  - Oversmoothing, oversquashing and problems of the sort
- ...in other words, plenty of opportunities for thesis work!

Advertisement time

Learning on Graph course

- Coming up on Semester 1, Year 2027
- 6 CFU Elective of the AI curriculum (M.Sc. Computer Science)
- From foundations of learning on graphs to edge-of-research models

Next Lecture

- Graph Learning Laboratory
- Final lecture (exams and all)