

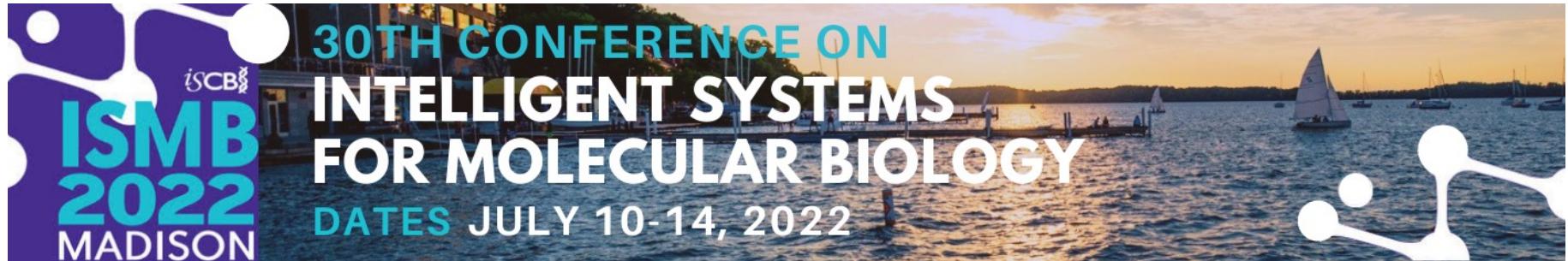
Towards Precision Medicine with Graph Representation Learning

Michelle M. Li & Marinka Zitnik

Department of Biomedical Informatics
Broad Institute of Harvard and MIT
Harvard Data Science

zitniklab.hms.harvard.edu/biomedgraphml





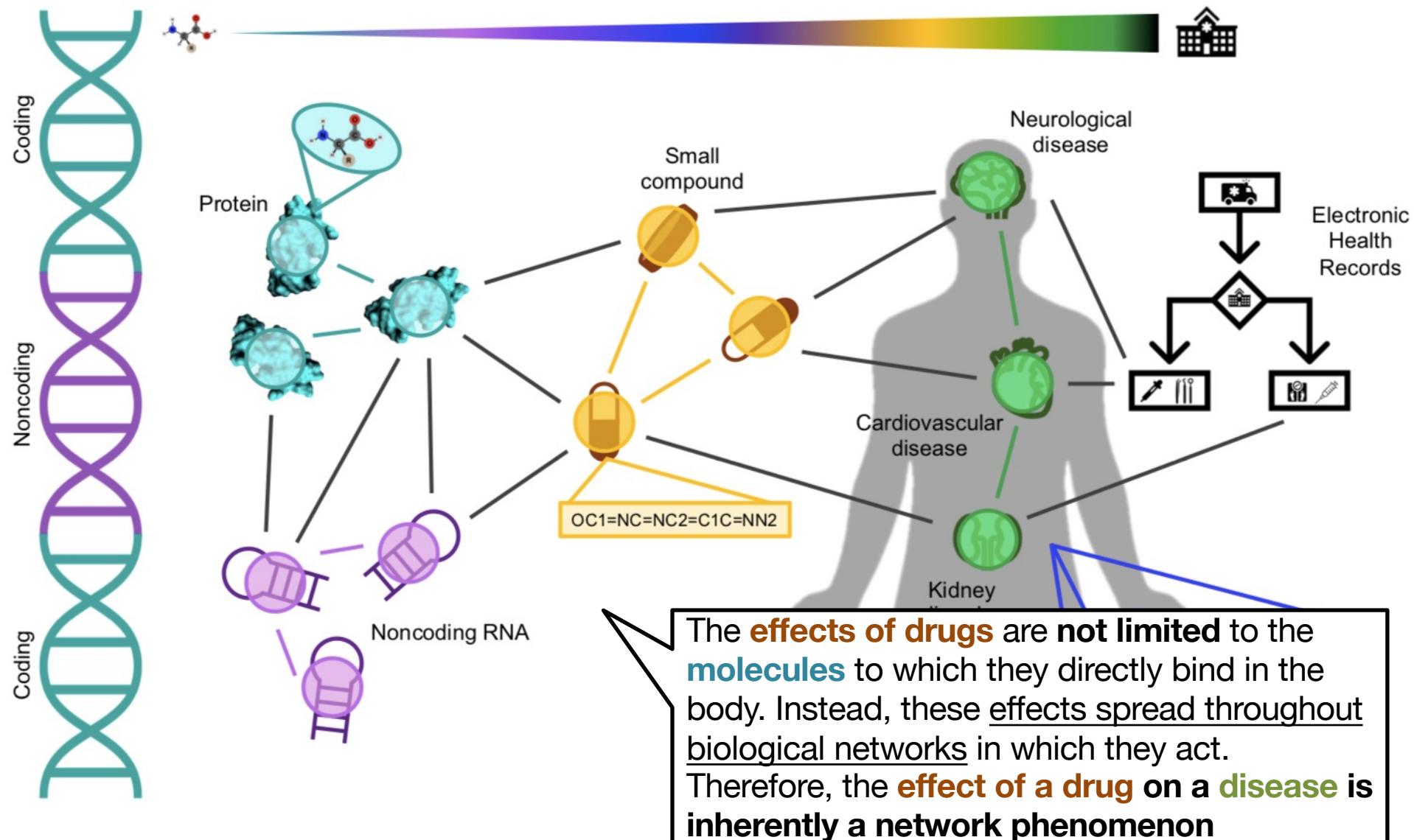
Tutorial VT4

July 7, 2022 at 9am – 1pm CDT

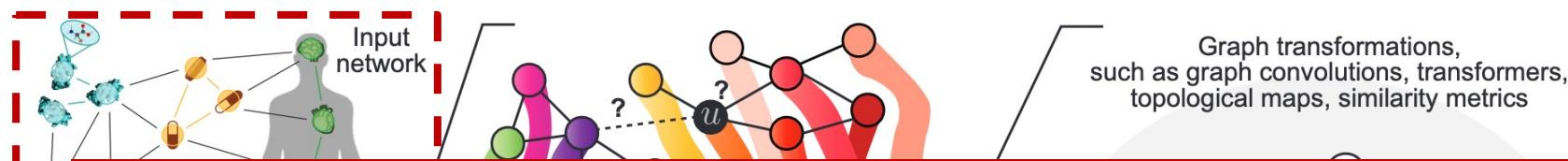


All tutorial materials are available at
zitniklab.hms.harvard.edu/biomedgraphml

Biology is interconnected



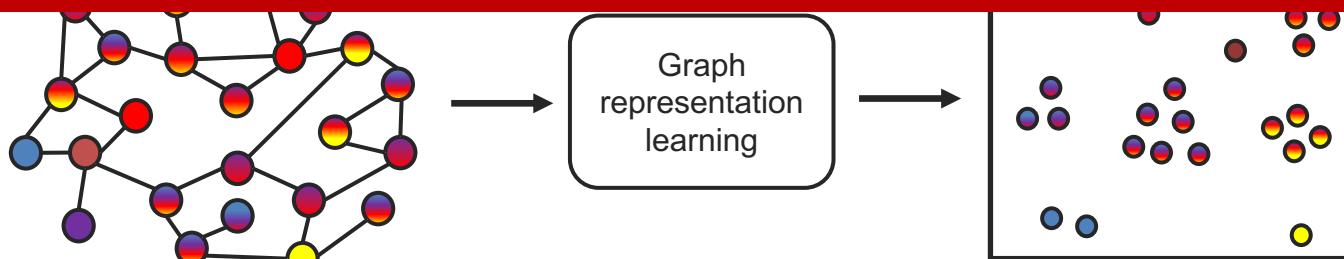
Graph representation learning realizes key network principles for data-rich biomedicine



Cellular components associated with a specific disease (phenotype) show a tendency to cluster in the same network neighborhood



Deep graph representation learning methods are well-suited for the analysis of biological networks

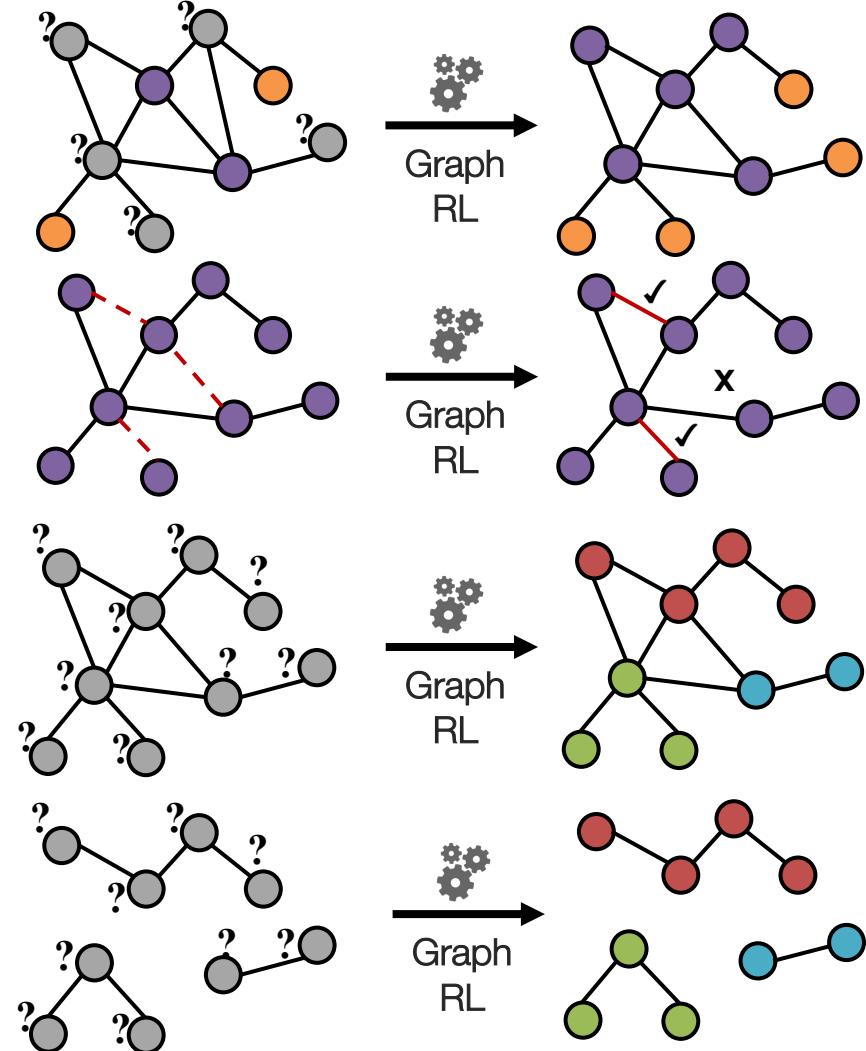


This Tutorial

- 
1. Methods: Network diffusion, shallow network embeddings, graph neural networks, equivariant neural networks
 2. Applications: Fundamental biological discoveries and precision medicine
 3. Outlook: Future directions and Q&A session
 4. Hands-on exercises: Demos, implementation details, tools, and tips

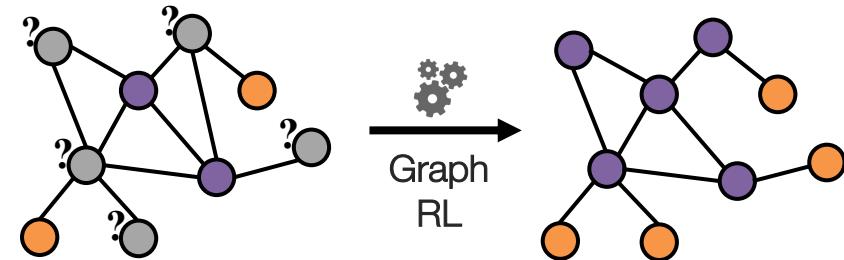
Graph representation learning tasks

- Node-level
 - Characteristics of a given node
- Edge-level
 - Whether or how two nodes are connected
- Subgraph-level
 - How clusters of nodes interact with each other and the rest of the graph
- Graph-level
 - Similarity of graph to other graphs



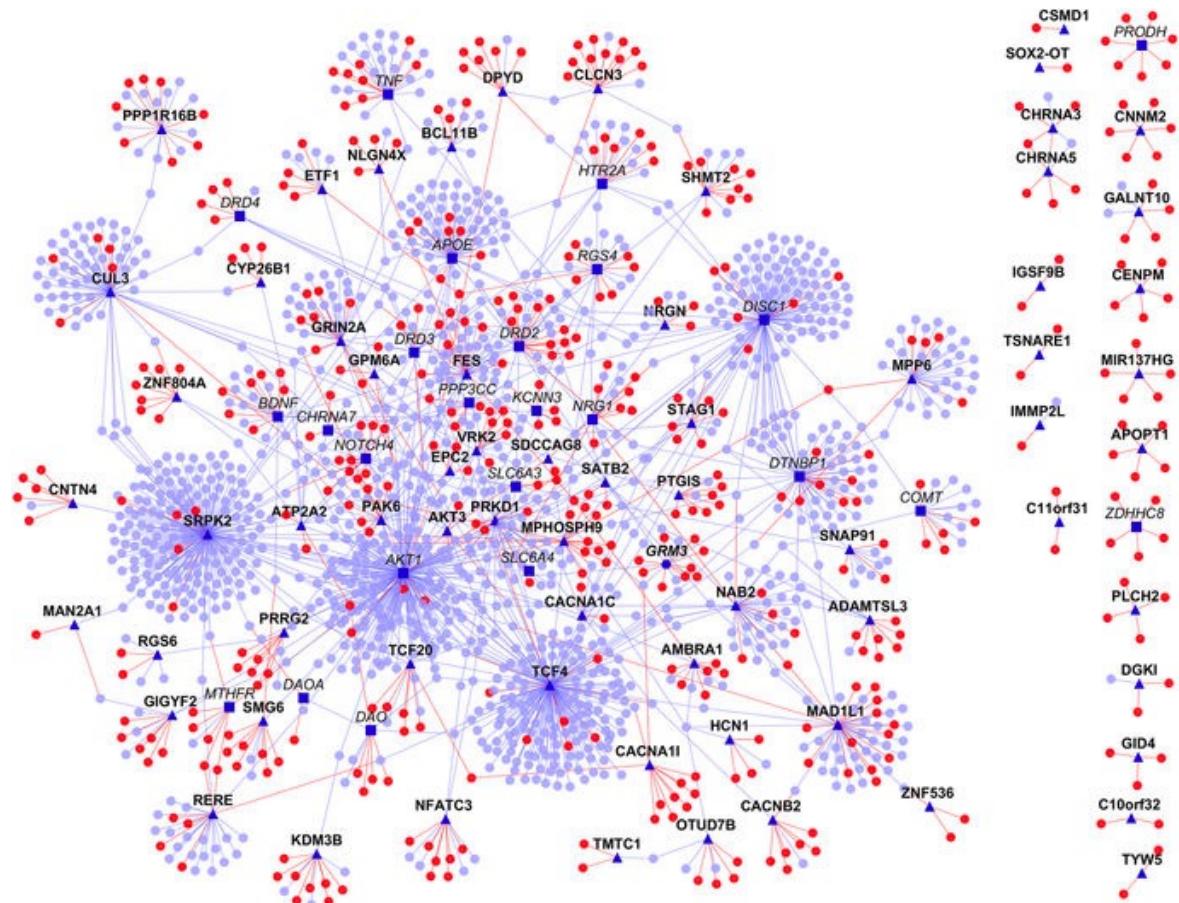
Graph representation learning tasks

- Node-level
 - Characteristics of a given node



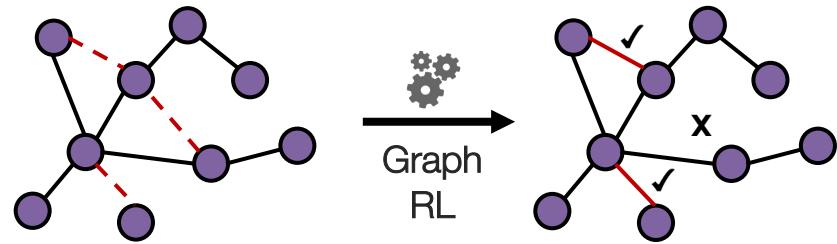
Node classification: Example

Classifying the function of proteins in the interactome!



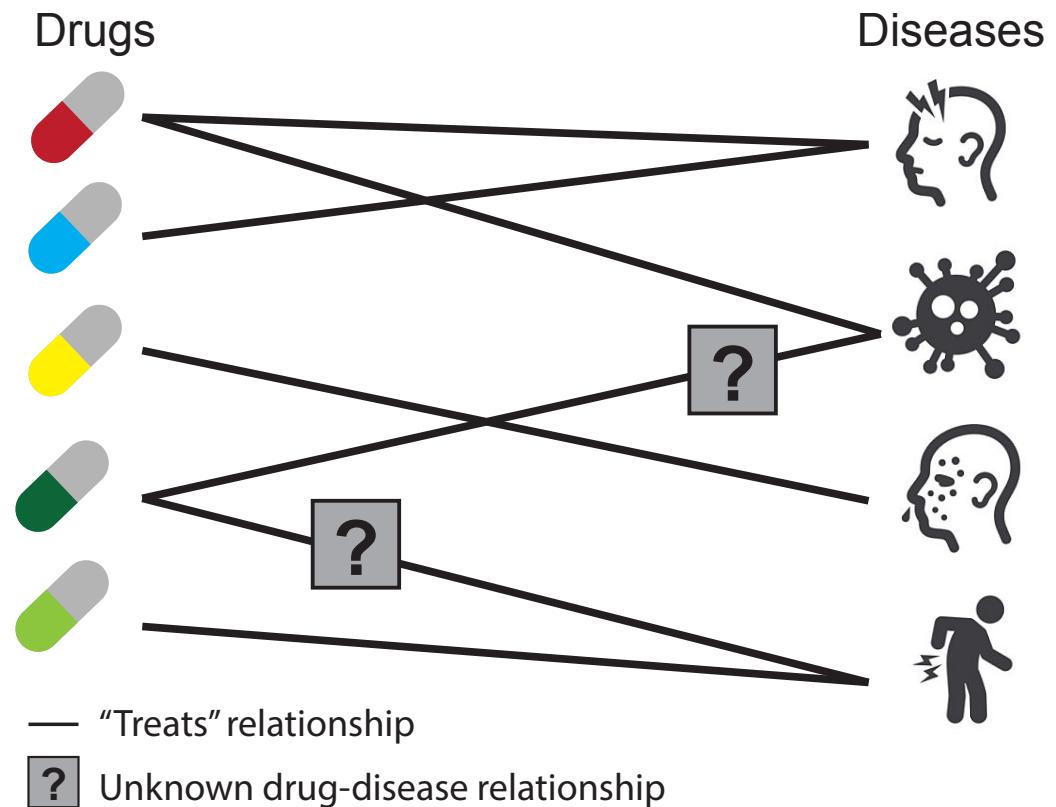
Graph representation learning tasks

- Edge-level
 - Whether or how two nodes are connected



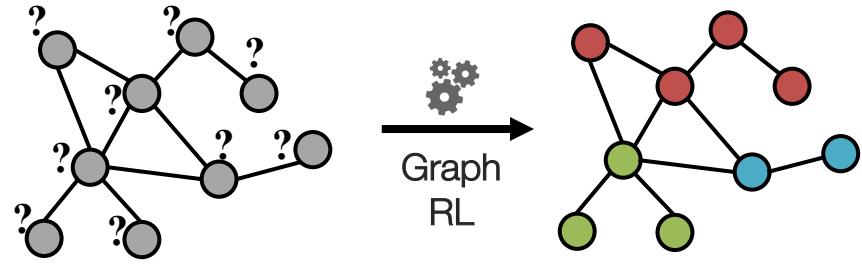
Link prediction: Example

Predicting
which
diseases a
new molecule
might treat!



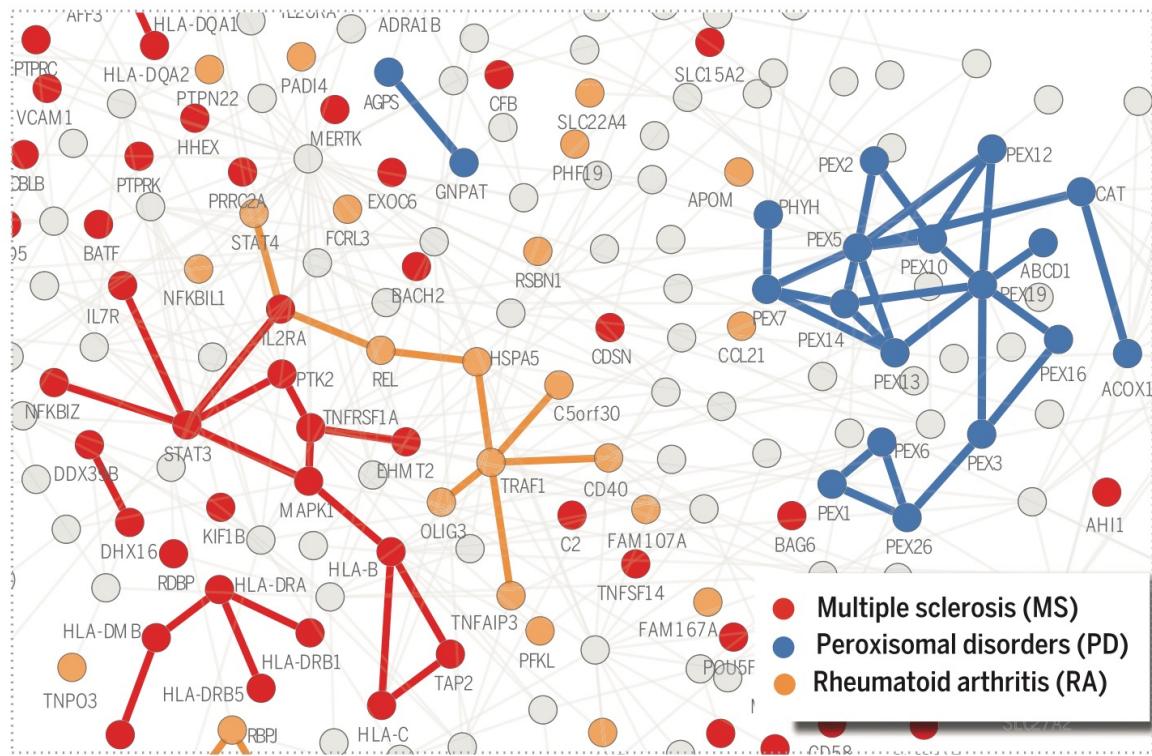
Graph representation learning tasks

- Subgraph-level
 - How clusters of nodes interact with each other and the rest of the graph



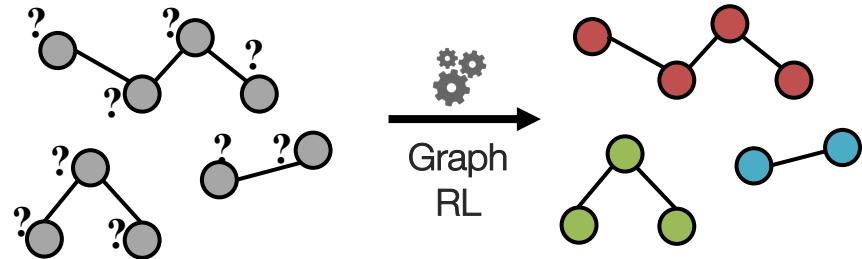
Subgraph classification: Example

Identifying
disease
proteins in the
interactome!



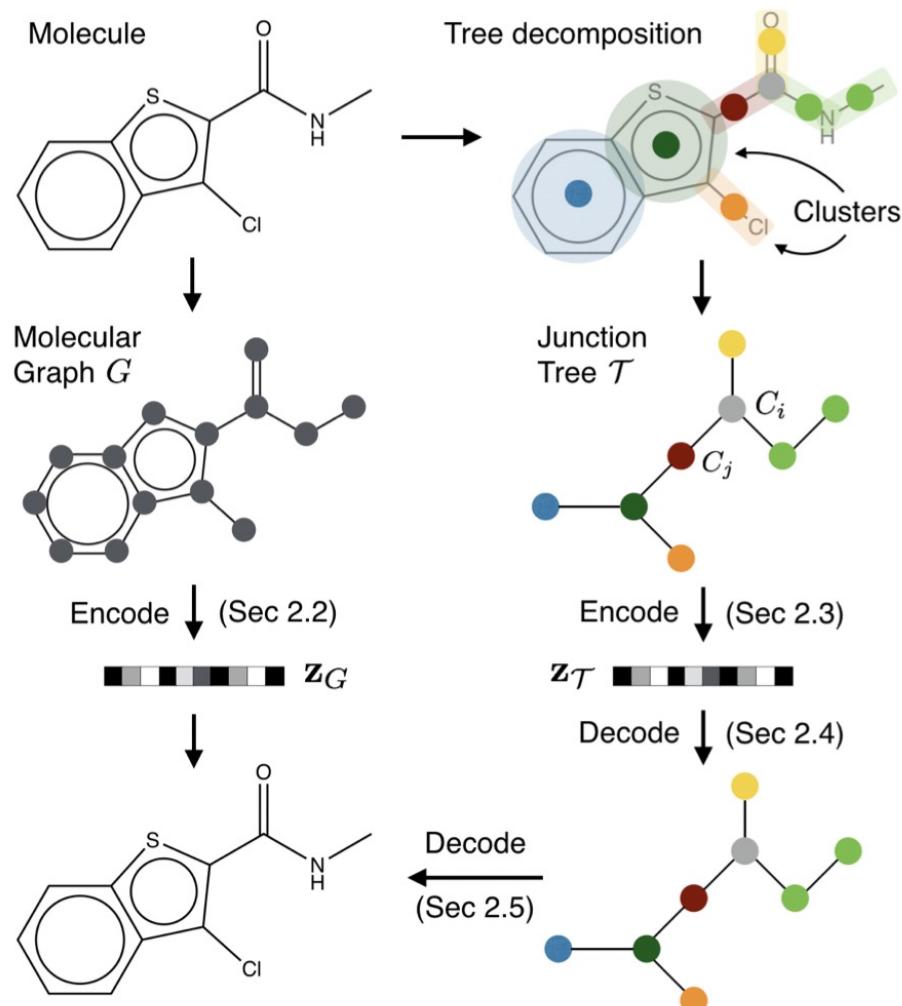
Graph representation learning tasks

- Graph-level
 - Similarity of graph to other graphs



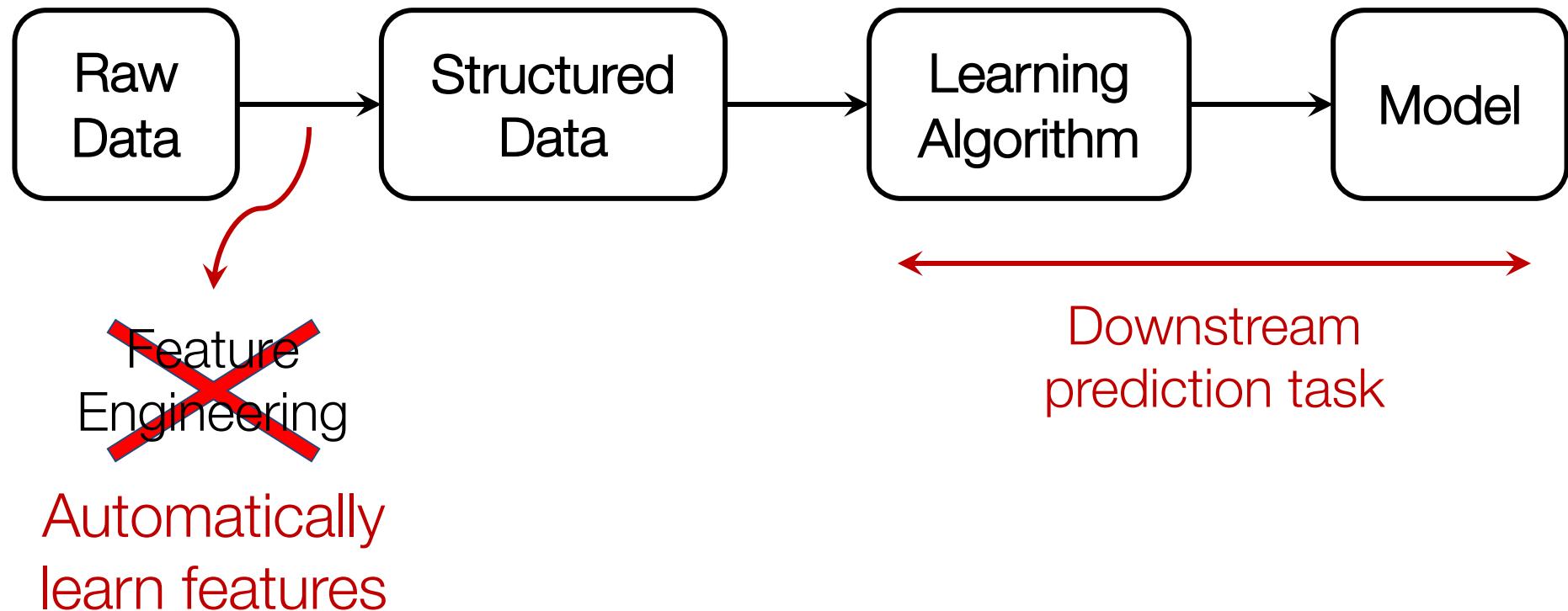
Graph classification: Example

Designing new
small molecule
compounds to
treat a disease!



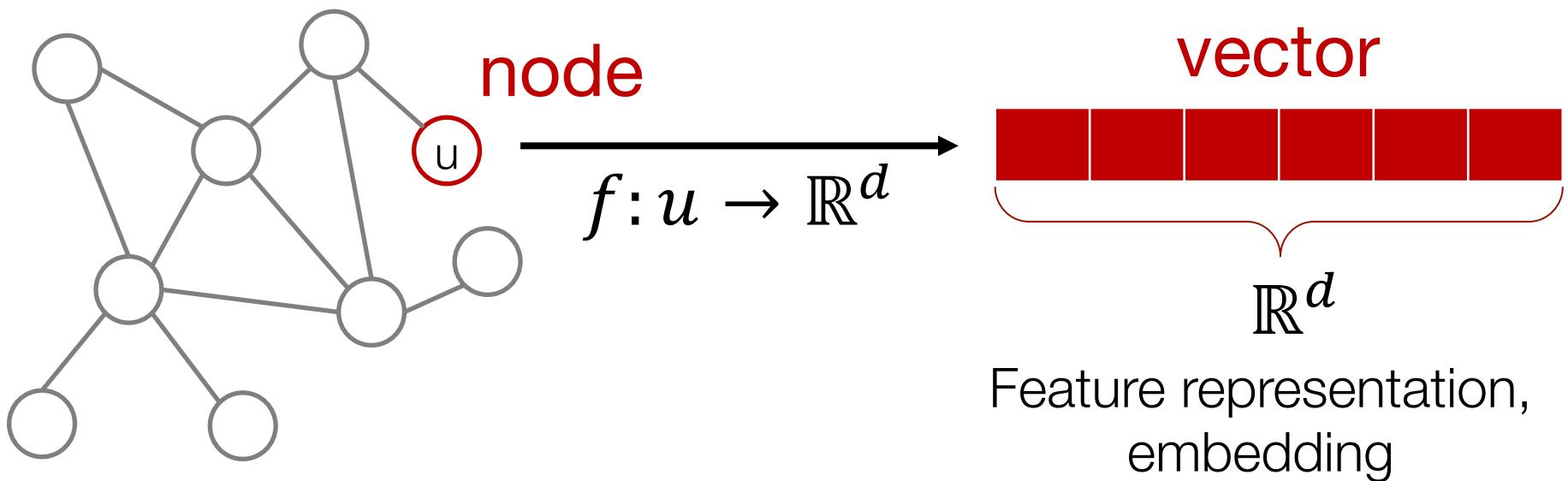
Predictive modeling lifecycle

(Supervised) Machine learning lifecycle: This feature, that feature. **Every single time!**



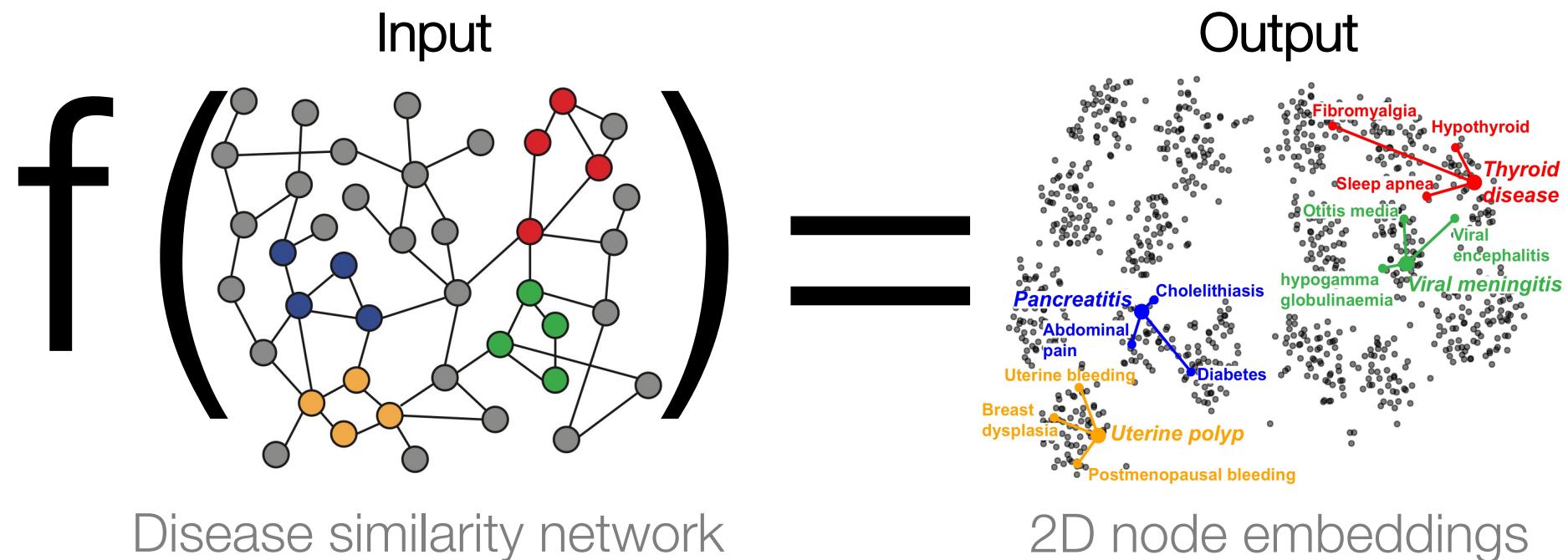
Feature learning in graphs

Goal: Efficient task-independent feature learning for machine learning in networks!



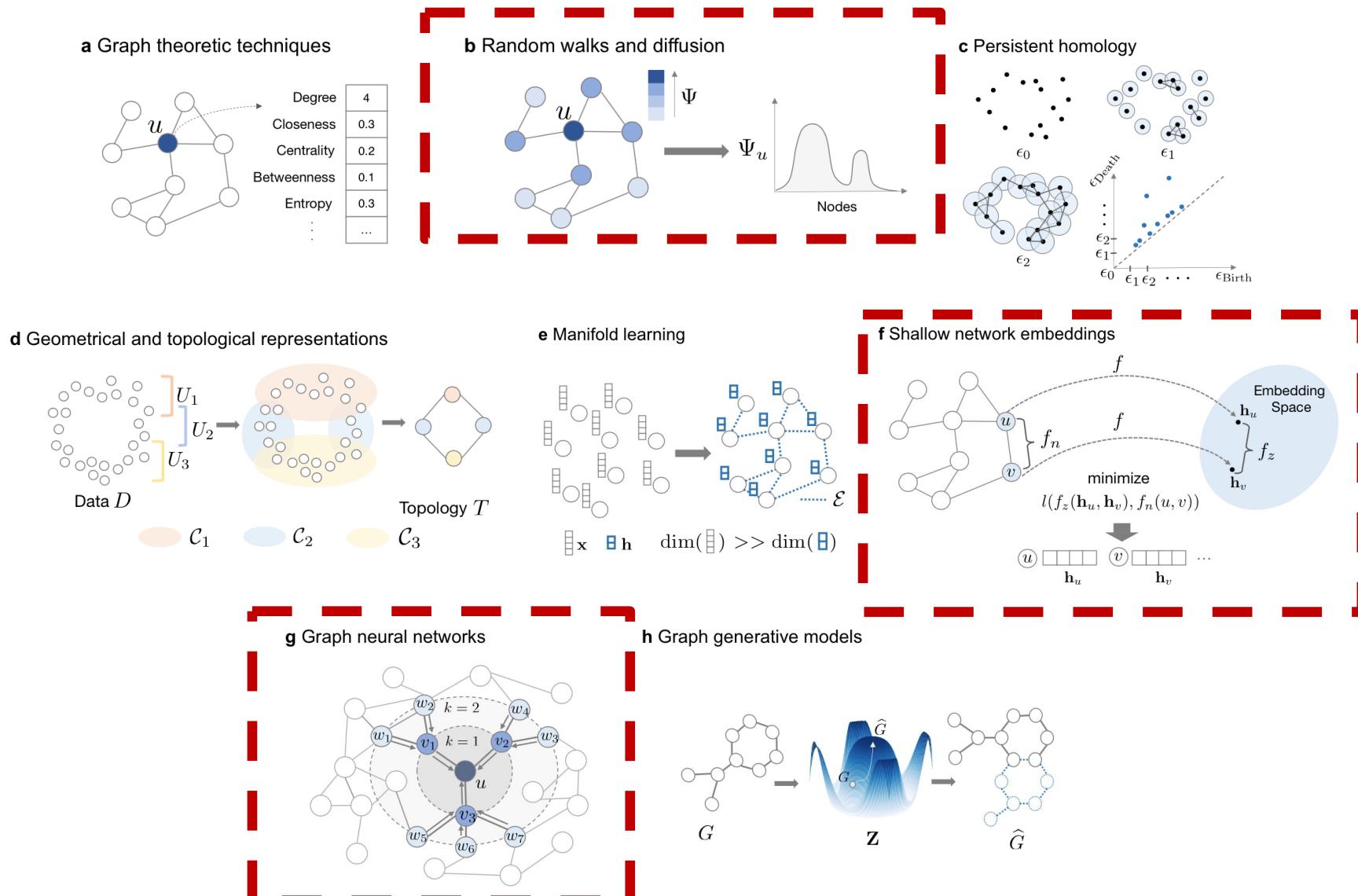
Embedding nodes

Intuition: Map nodes to embeddings such that similar nodes in graph are embedded closeby



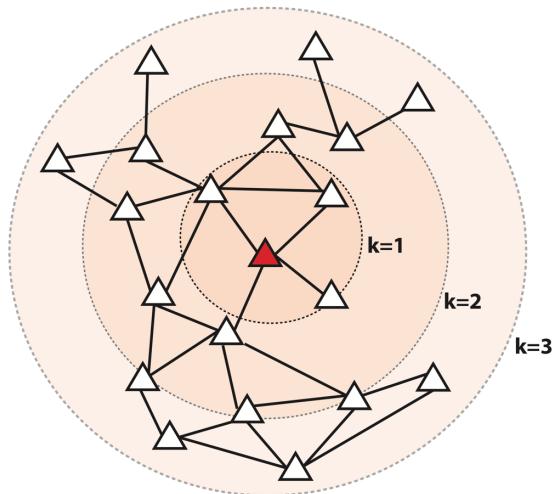
How to learn mapping function f ?

Predominant graph learning paradigms



Random walks and diffusion

- Nodes in a graph **influence** each other along paths
- **Diffusion** measures these spreads of influences

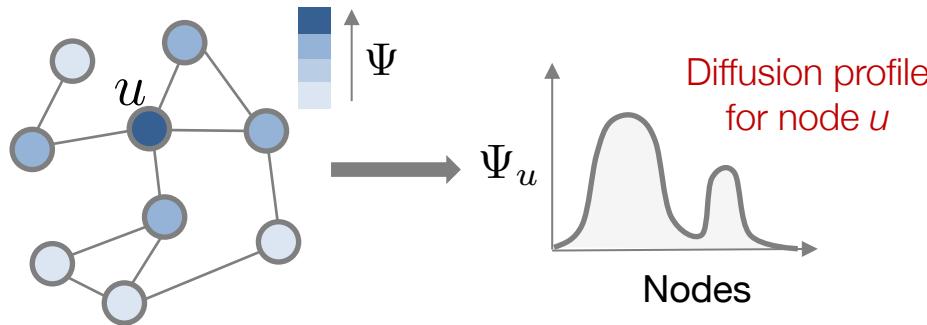


Red: Target node
 $k = 1$: 1-hop neighbors
 $k = 2$: 2-hop neighbors
 $k = 3$: 3-hop neighbors

- Intuition
 - Capture the local connectivity patterns for each node
 - Define node similarity function based on higher-order neighborhoods

Random walks and diffusion

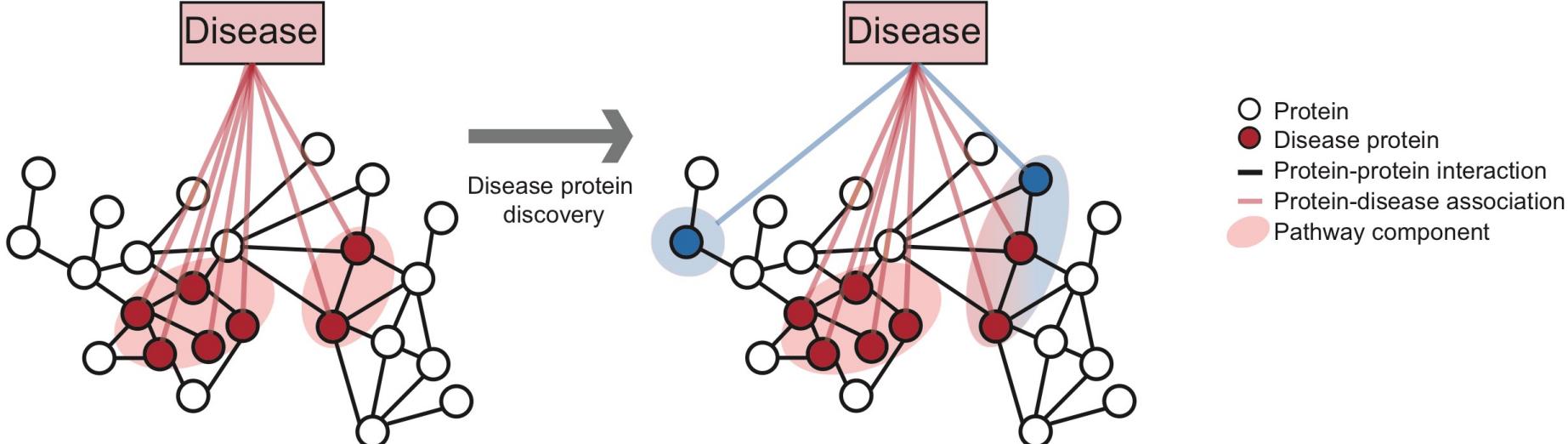
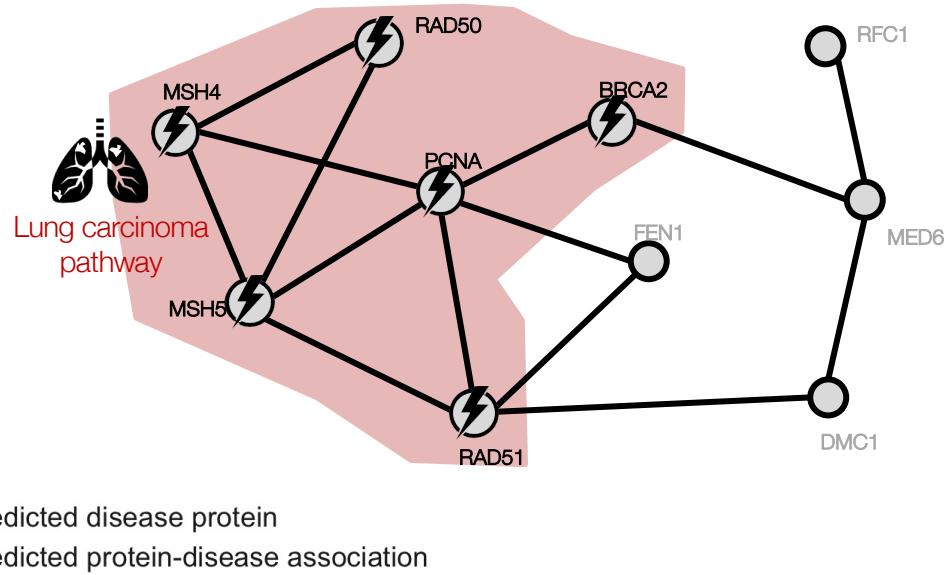
- Method: Diffusion state distance (DSD)
 - Simulate random walks from source node u
 - Count the number of random walks of length k that start at u and visit a destination node v
- Node representation: Each node u has vector Ψ_u that represents its influence on its k -hop neighborhood



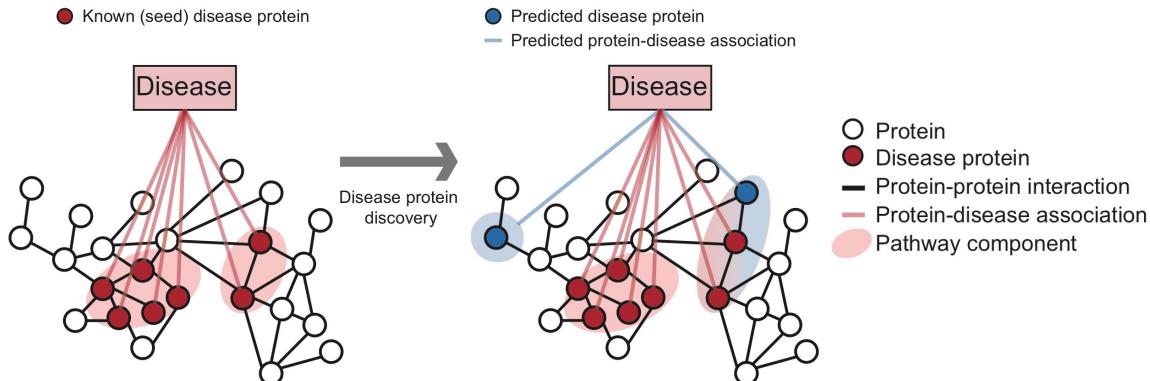
- Comparison: Calculate whether nodes u and v have similar local connectivity by $DSD(u, v) = \|\Psi_u - \Psi_v\|_1$

Application: Identify disease pathways

- **Pathway:** Subnetwork of interacting proteins associated with a disease

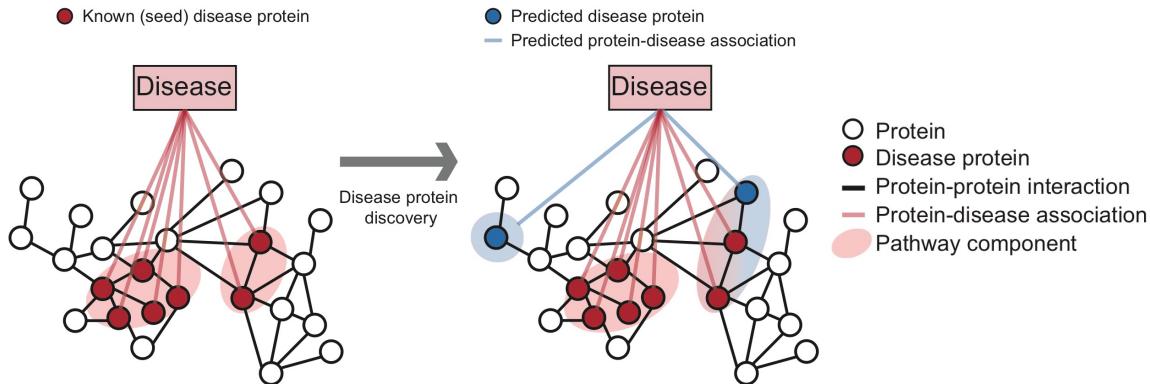


Disease pathway dataset



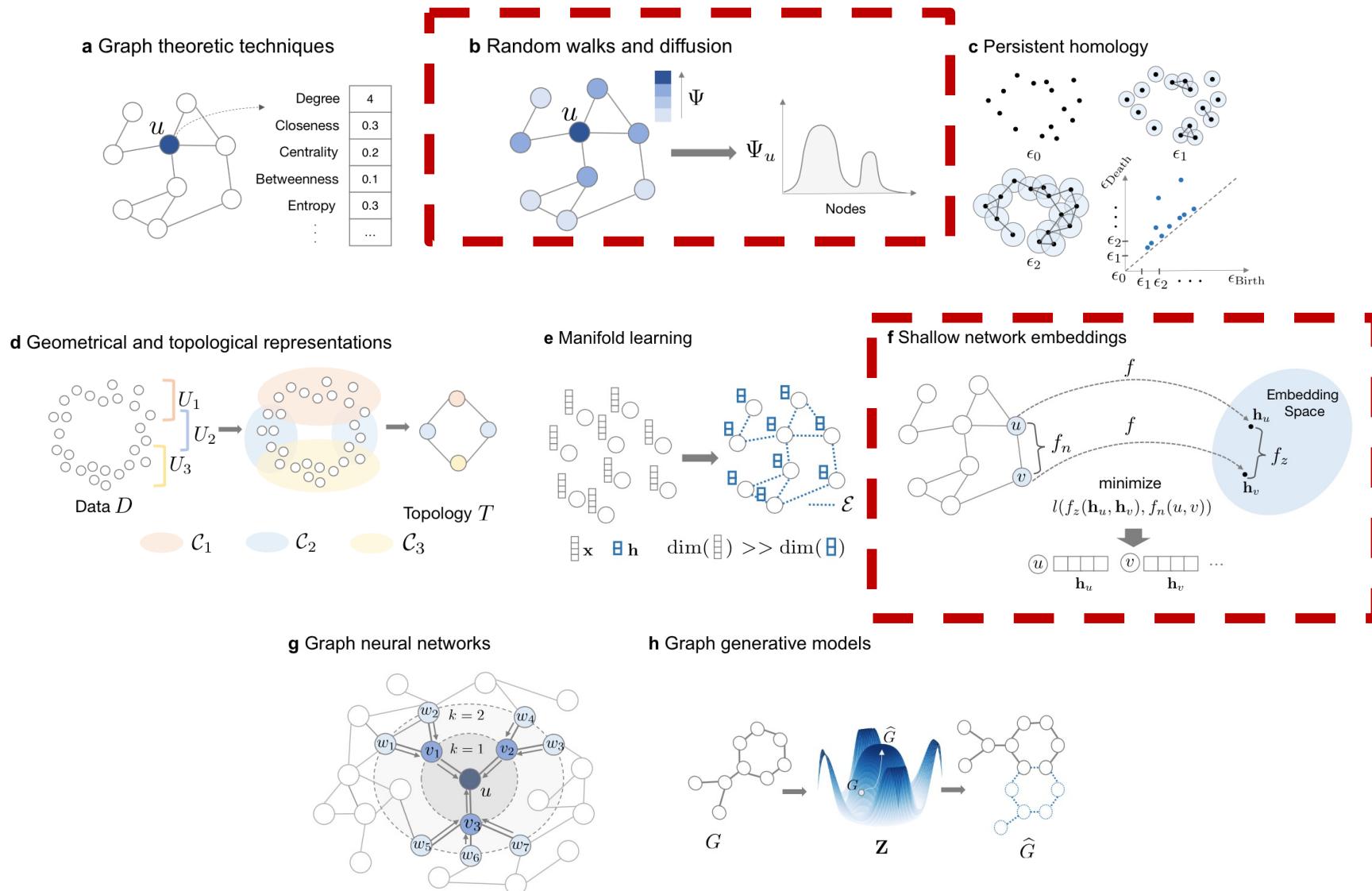
- Protein-protein interaction (PPI) network culled from 15 knowledge databases:
 - 350k physical interactions
 - Examples: metabolic enzyme-coupled interactions, signaling interactions, protein complexes
 - All protein-coding human genes (21k)
- Protein-disease associations
 - 21k associations split among 519 diseases
- Multi-label node classification
 - Every node (i.e. protein) can have 0, 1, or more labels (i.e. disease associations)

Experimental setup



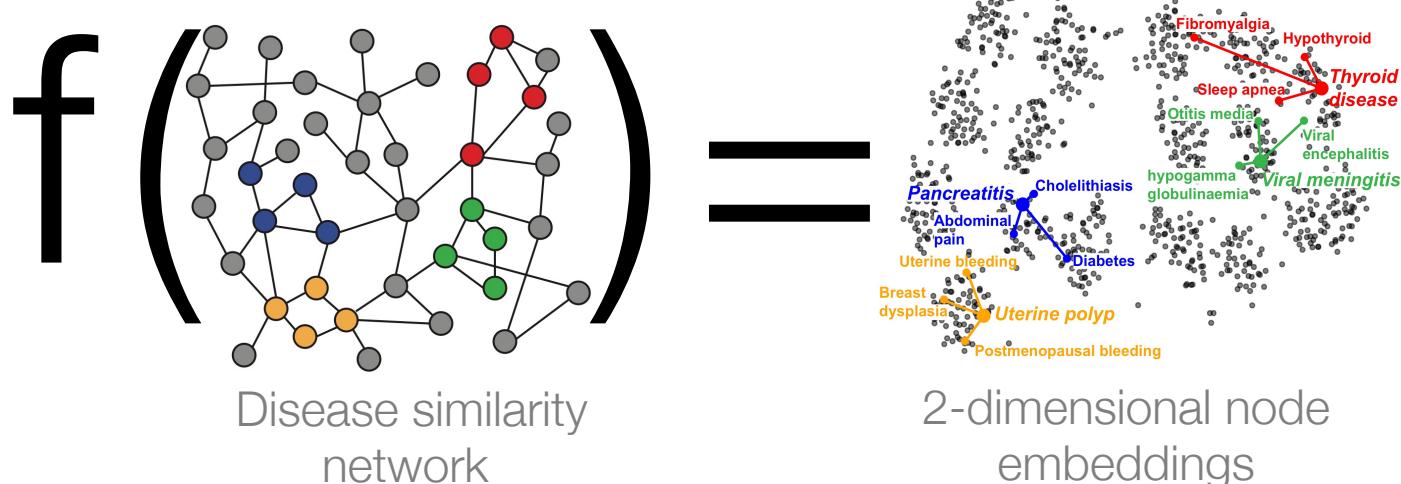
- Two main stages:
 1. Take the PPI network and use DSD to compute a **vector representation for every node**
 2. For each disease, fit a logistic regression **classifier** that predicts disease proteins based on the vector representations:
 - Train the classifier using training proteins
 - Predict disease proteins in the test set (i.e. the **probability** that a particular protein is associated with the disease)

Predominant graph learning paradigms



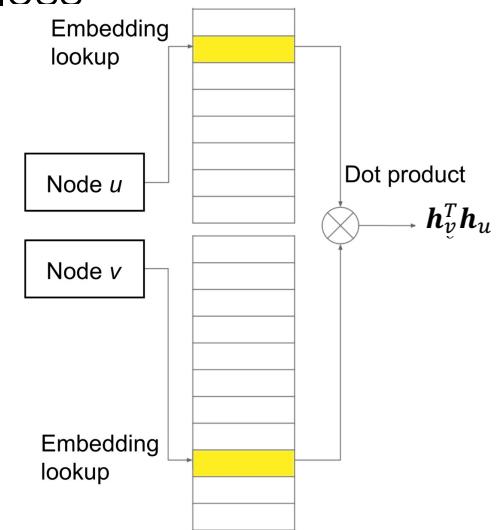
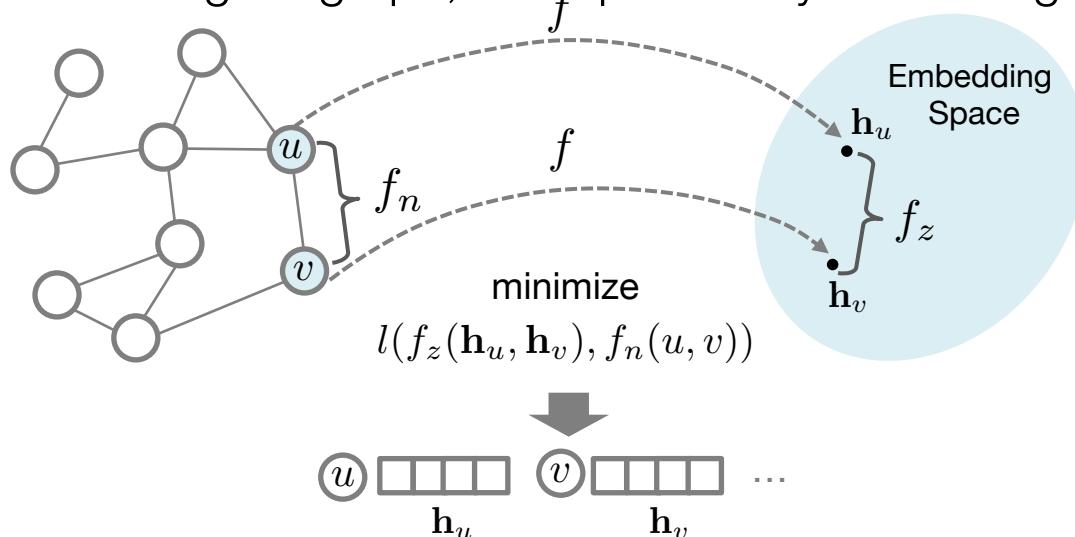
Shallow network embeddings

- **Intuition:** Map nodes to d -dimensional embeddings such that similar nodes in the graph are embedded close together
- Assume we have a graph G :
 - V is the vertex set
 - A is the adjacency matrix (assume binary)
 - No node features or extra information is used!



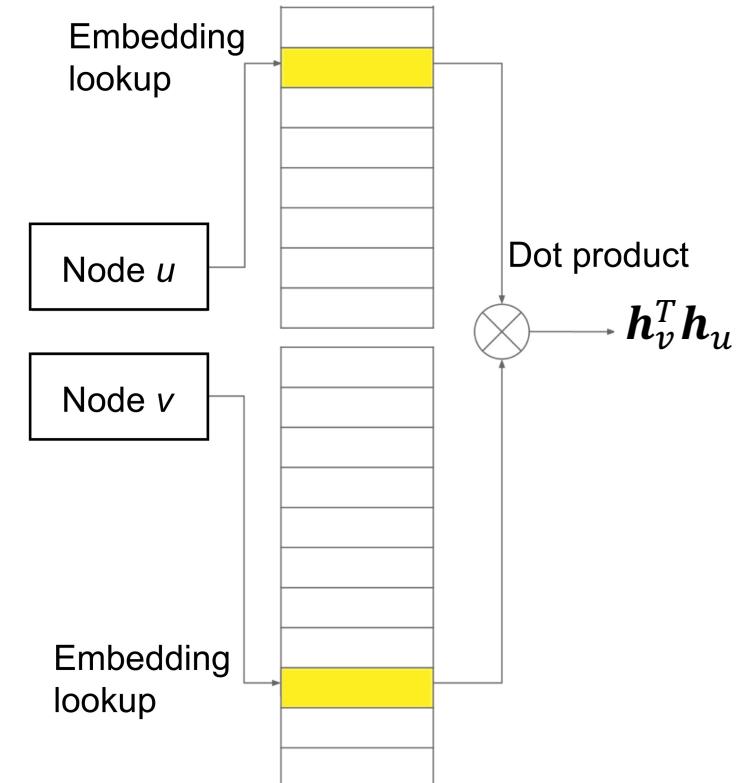
Shallow network embeddings

- **Goal:** Similarity in the embedding space approximates similarity in the network
- **Three main stages:**
 1. Given a pair of nodes u, v in network G , obtain function f to map these nodes to an embedding space to generate \mathbf{h}_u and \mathbf{h}_v
 2. Define network similarity $f_n(u, v)$ and embedding similarity $f_z(\mathbf{h}_u, \mathbf{h}_v)$
 3. Define loss l to measure whether embedding preserves distance in original graph, and optimize by minimizing the loss



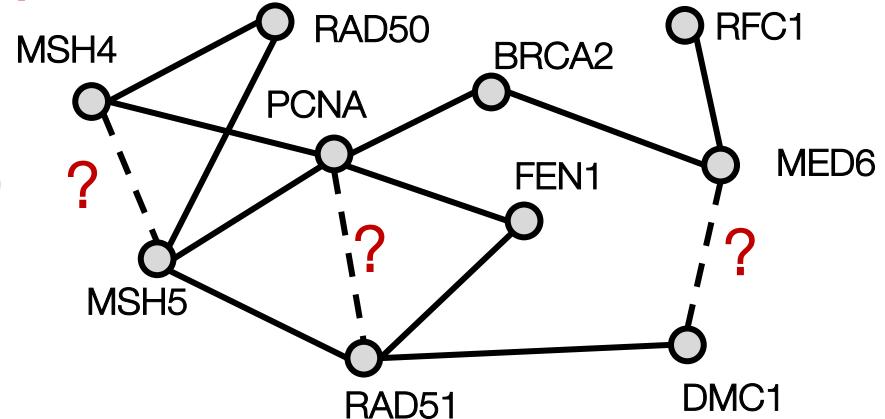
Shallow network embeddings

- Summary:
 - One-layer of data transformation
 - A single hidden layer maps node u to embedding \mathbf{h}_u via function f
- Limitations:
 - $O(|V|)$ parameters are needed:
 - No sharing of parameters between nodes
 - Every node has its own unique embedding
 - Inherently “transductive”
 - Cannot generate embeddings for nodes **not** seen during training
 - Do not incorporate node features
 - Many graphs have features that we can and should leverage



Application: Predict protein interactions

- Human PPI network:
 - Experimentally validated physical protein-protein interactions (BioGRID)
- Link prediction: Given two proteins, predict probability that they interact



How to address tasks involving pairs of nodes (e.g., link prediction)?

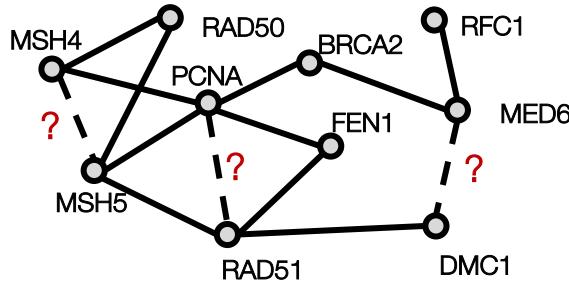
- Given u and v , define an operator g that generates an embedding for pair (u, v) :

$$\mathbf{h}_{(u,v)} = g(u, v)$$

- Examples of choices for g

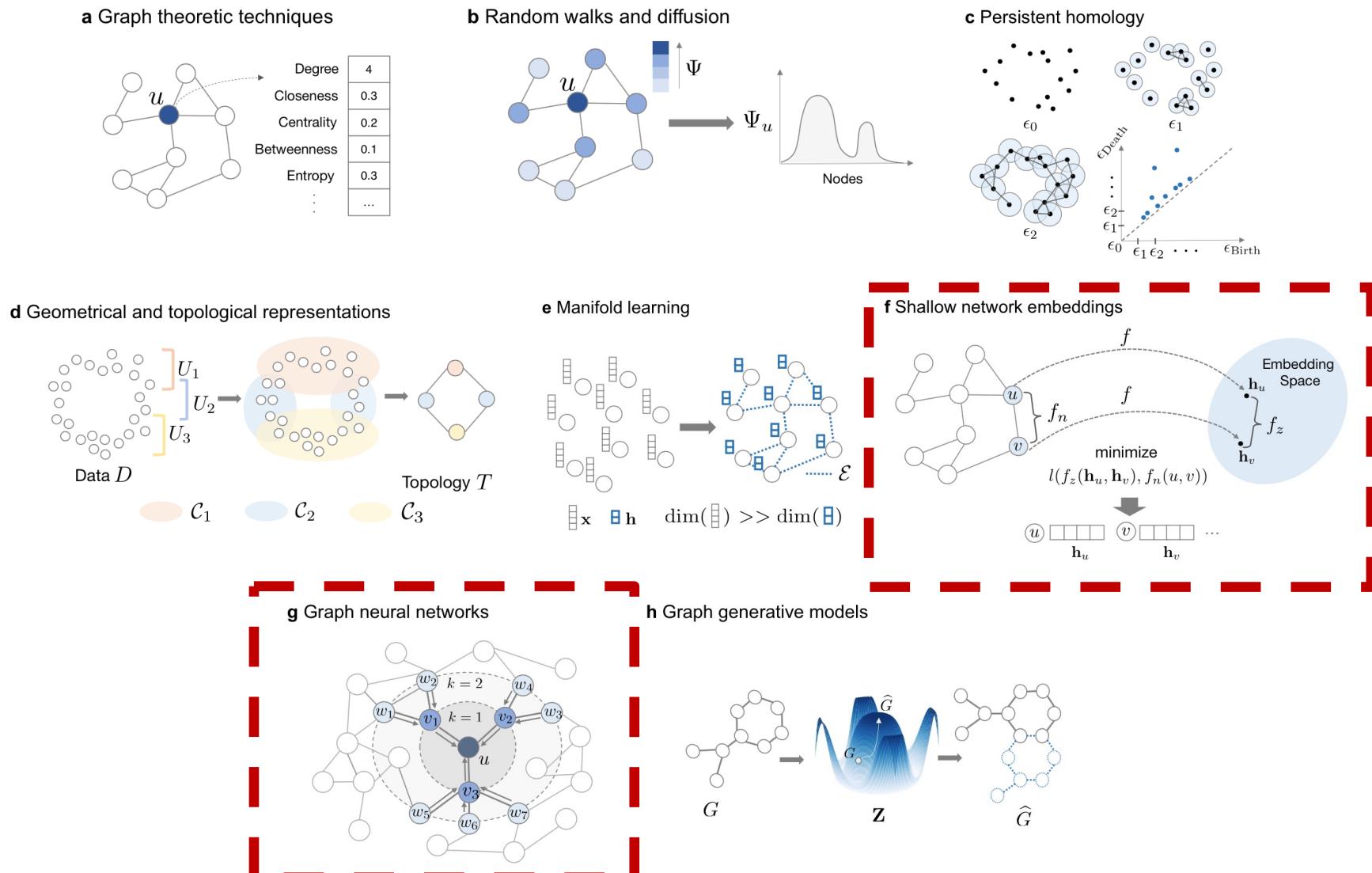
Scoring node pairs	Definition
(a) Average	$[\mathbf{z}_u \boxplus \mathbf{z}_v]_i = \frac{\mathbf{z}_u(i) + \mathbf{z}_v(i)}{2}$
(b) Hadamard	$[\mathbf{z}_u \boxdot \mathbf{z}_v]_i = \mathbf{z}_u(i) * \mathbf{z}_v(i)$
(c) Weighted-L1	$\ \mathbf{z}_u \cdot \mathbf{z}_v\ _{\bar{1}i} = \mathbf{z}_u(i) - \mathbf{z}_v(i) $
(d) Weighted-L2	$\ \mathbf{z}_u \cdot \mathbf{z}_v\ _{\bar{2}i} = \mathbf{z}_u(i) - \mathbf{z}_v(i) ^2$

Experimental Setup



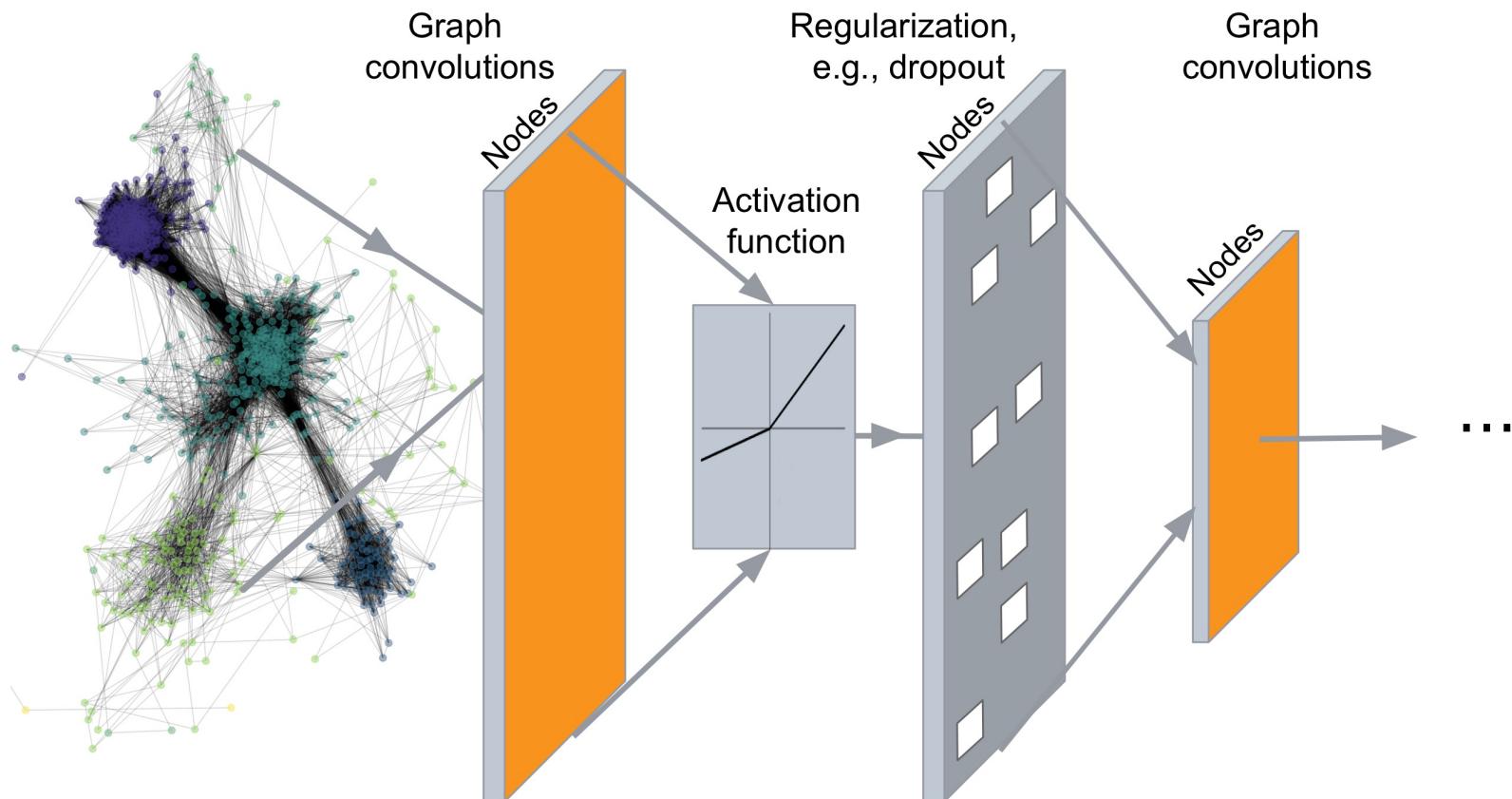
- We are given a PPI network with a certain fraction of edges removed:
 - Remove about 50% of edges
 - Randomly sample an equal number of node pairs that have no edge connecting them
 - Explicitly removed edges and non-existent (or false) edges form a balanced test data set
- Two main stages:
 1. Learn an embedding for every node in the filtered PPI network
 2. Predict a score for every protein pair in the test set based on the embeddings

Predominant graph learning paradigms



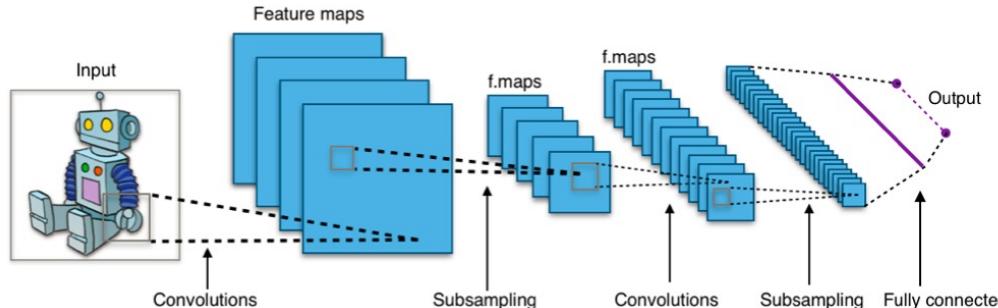
Graph neural networks

- Encoder: Multiple layers of nonlinear transformation of graph structure

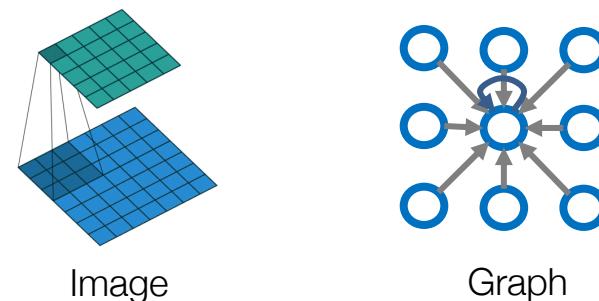


Convolutional networks

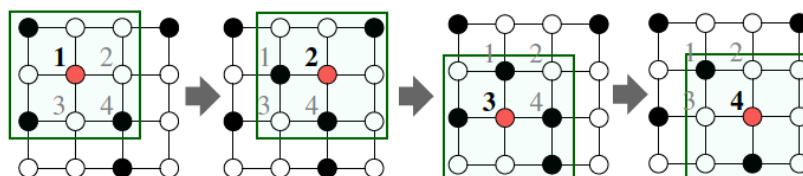
- Let's start with convolutional networks on an image:



- Single convolutional network with a 3×3 filter:

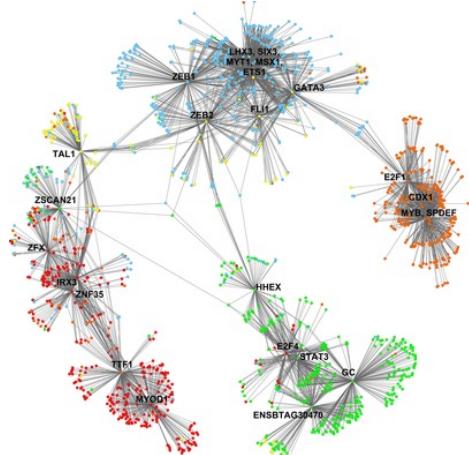


- Transform information (or messages) from the neighbors and combine them: $\sum_i W_i h_i$

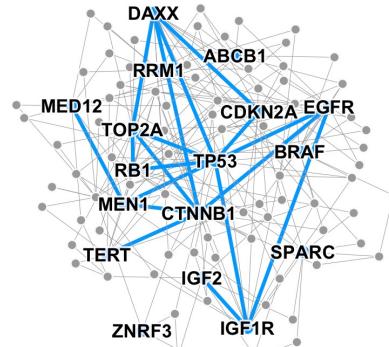


Real world graphs

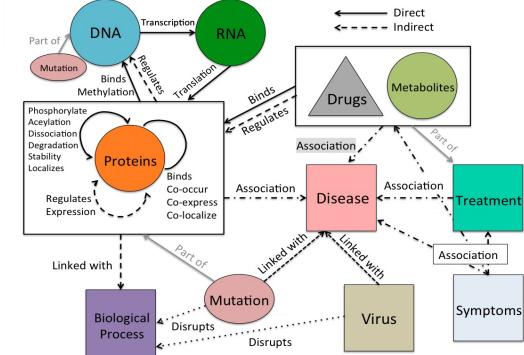
- But what if your graphs look like this?



Gene interaction network



Disease pathways



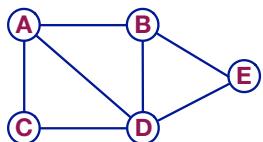
Biomedical knowledge graphs

- Examples:

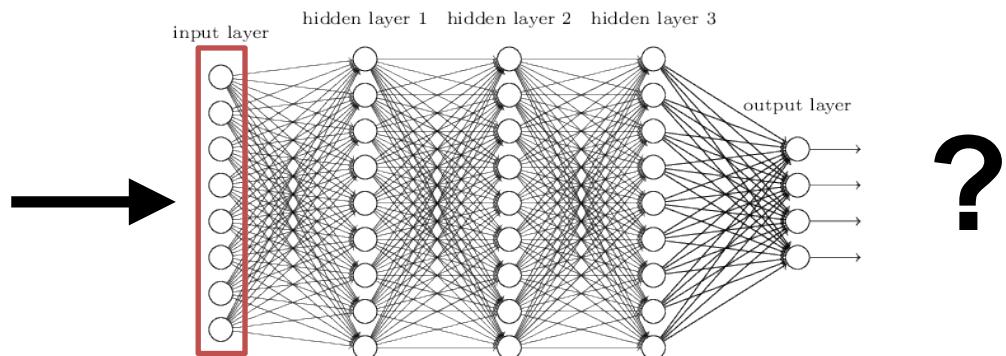
- Biological or medical networks
- Social networks
- Information networks
- Knowledge graphs
- Communication networks
- Web graphs
- ...

Naïve approach

- Join adjacency matrix and features
- Feed them into a deep neural network:



	A	B	C	D	E	Feat	
A	0	1	1	1	0	1	0
B	1	0	0	1	1	0	0
C	1	0	0	1	0	0	1
D	1	1	1	0	1	1	1
E	0	1	0	1	0	1	0

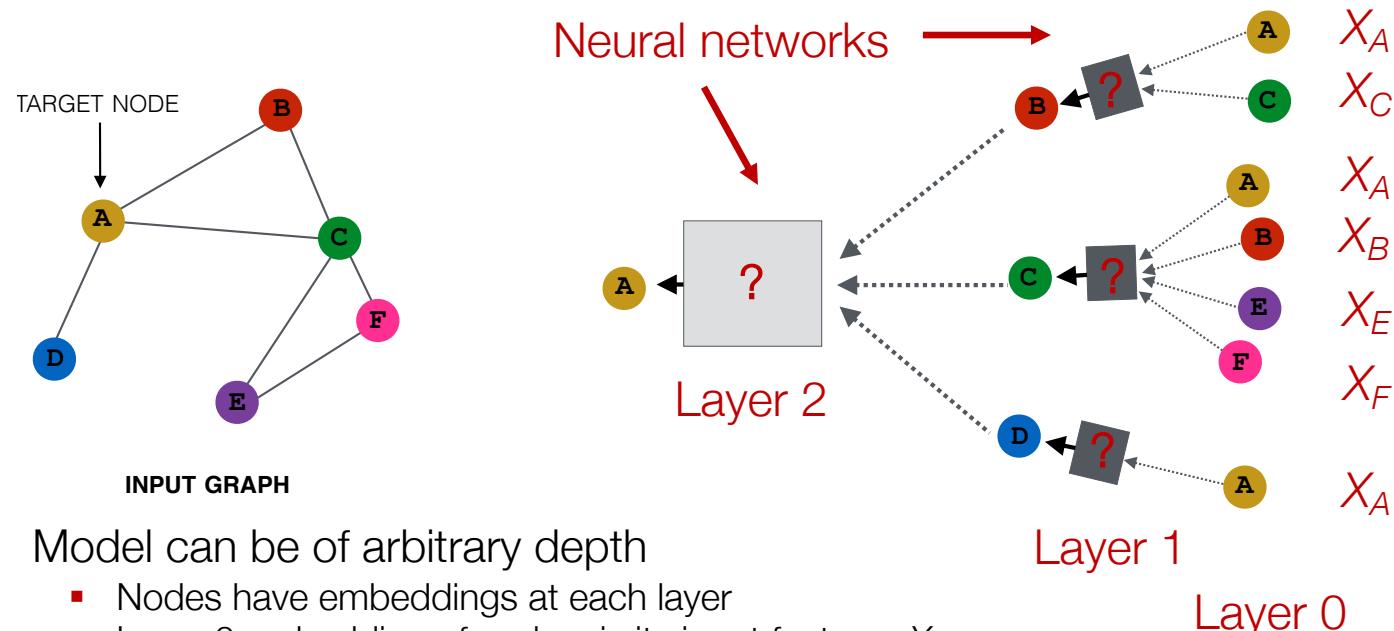
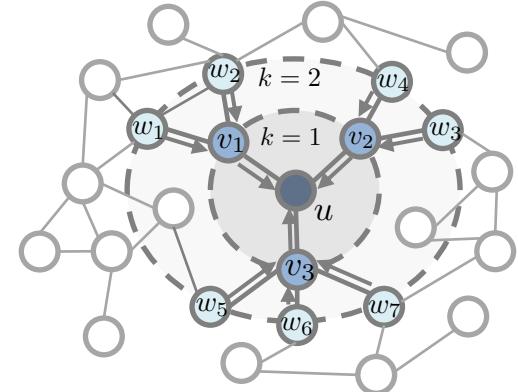


- Issues with this idea:
 - $O(N)$ parameters
 - Not applicable to graphs of different sizes
 - Not invariant to node ordering

?

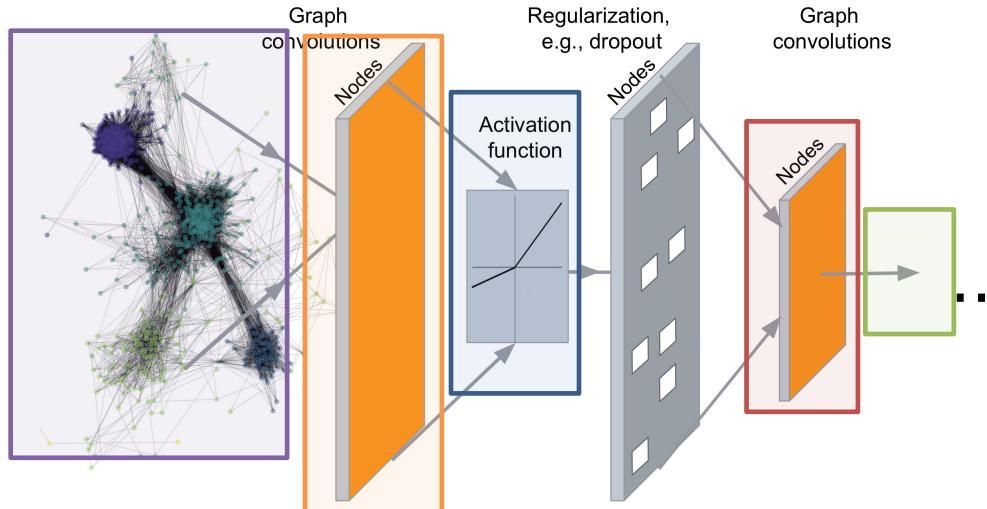
Graph neural networks

- Intuition:
 - Each node's neighborhood defines a computational graph
 - Generate node embeddings based on local network neighborhoods
- Neighborhood aggregation:



- Model can be of arbitrary depth
 - Nodes have embeddings at each layer
 - Layer 0 embedding of node u is its input features X_u
- **Basic neighborhood aggregation approach (i.e. □):** Average information from neighbors and apply a neural network

Basic approach

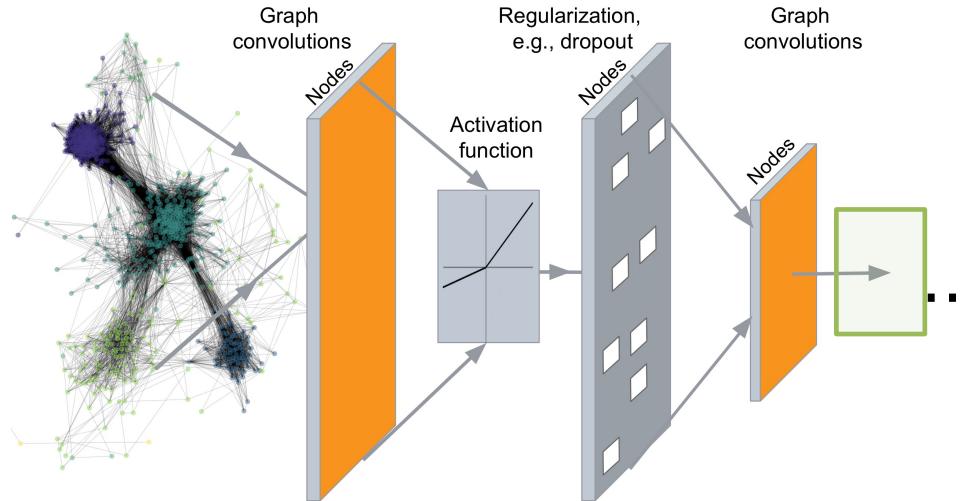


$$\begin{aligned}
 h_v^0 &= x_v && \text{Initial 0-th layer embeddings are equal to node features} \\
 h_v^k &= \sigma \left(\mathbf{W}_k \left(\sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1} \right) \right), \quad \forall k \in \{1, \dots, K\} && \text{Previous layer embedding of } v \\
 \mathbf{z}_v &= \mathbf{h}_v^K &&
 \end{aligned}$$

Annotations for the equations:

- $\mathbf{h}_v^0 = x_v$: Initial 0-th layer embeddings are equal to node features
- $\mathbf{h}_v^k = \sigma \left(\mathbf{W}_k \left(\sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1} \right) \right)$, $\forall k \in \{1, \dots, K\}$: Previous layer embedding of v
- $\mathbf{z}_v = \mathbf{h}_v^K$: Embedding after K layers of neighborhood aggregation
- $\mathbf{h}_v^k = \sigma \left(\mathbf{W}_k \left(\sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1} \right) \right)$: Average of neighbor's previous layer embeddings
- $\mathbf{h}_v^k = \sigma \left(\mathbf{W}_k \left(\sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1} \right) \right)$: Non-linearity (e.g., ReLU)

Basic approach



trainable weight matrices

$$\mathbf{h}_v^0 = \mathbf{x}_v$$

(i.e., what we learn)

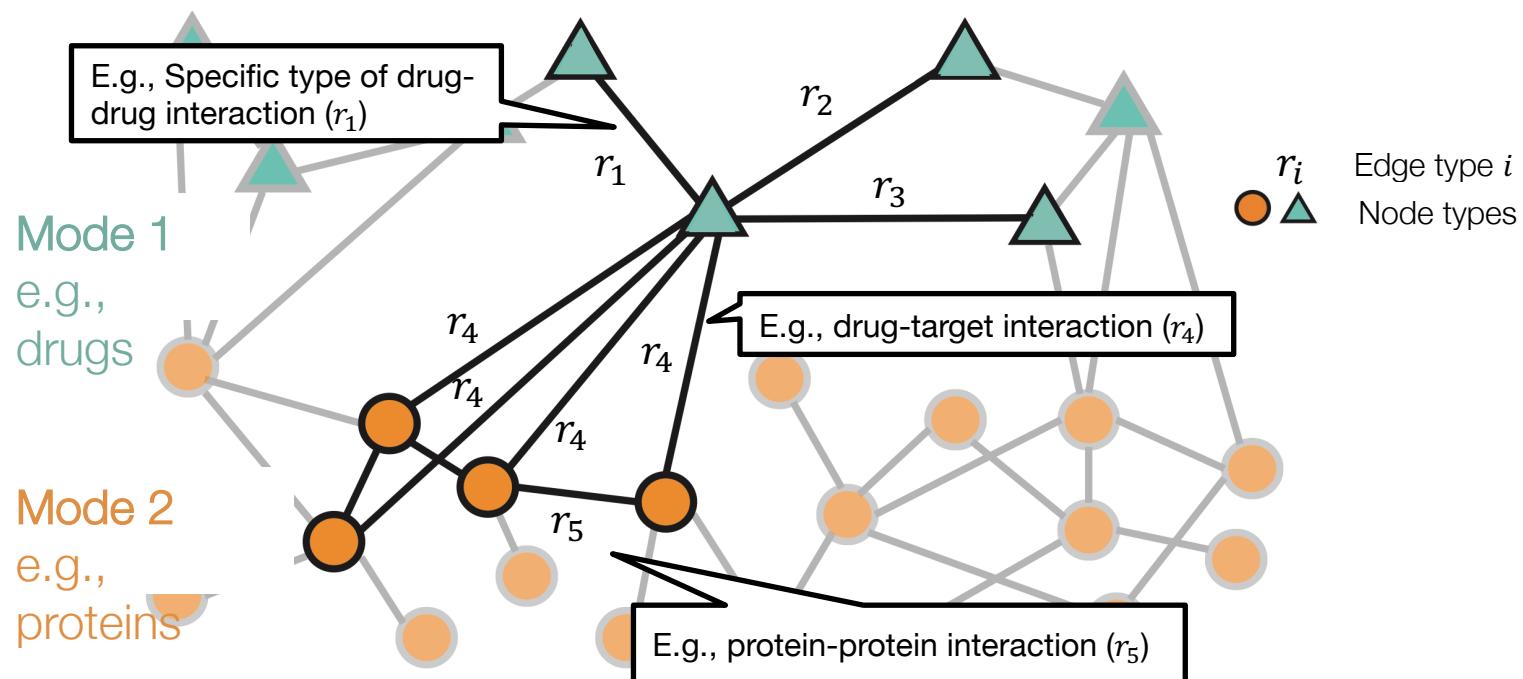
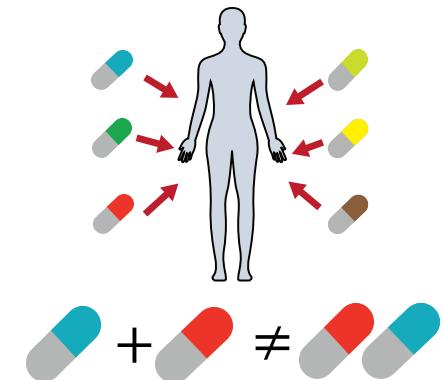
$$\mathbf{h}_v^k = \sigma \left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1} \right), \quad \forall k \in \{1, \dots, K\}$$

$$\boxed{\mathbf{z}_v} = \mathbf{h}_v^K$$

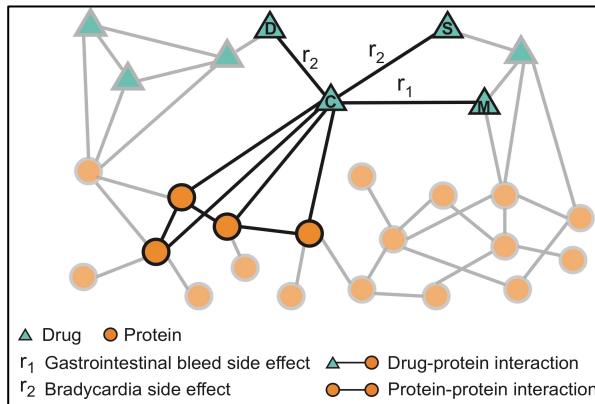
We can feed these into any loss function and run stochastic gradient descent to train the weight parameters

Application: Prioritize drug combos

- Combinatorial explosion
 - >13 million possible combinations of 2 drugs
 - >20 billion possible combinations of 3 drugs
- Non-linear & non-additive interactions
 - Different effect than the additive effect of individual drugs
- Small subsets of patients
 - Side effects are interdependent
 - No info on drug combinations not yet used in patients

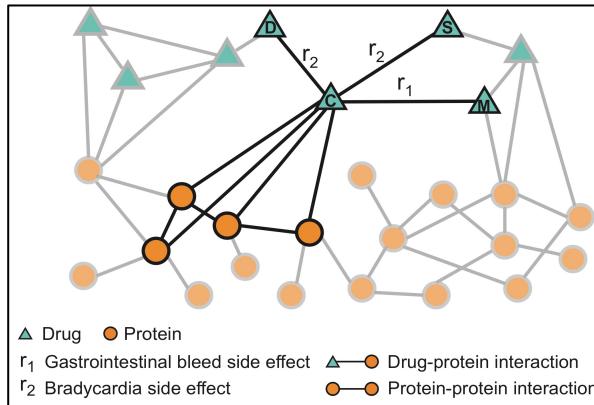


Polypharmacy dataset

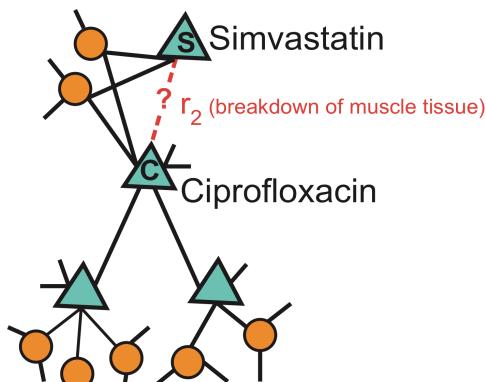


- Molecular, drug, and patient data for all drugs prescribed in US
 - 4,651,131 drug-drug edges: Patient data from adverse event system, tested for confounders [FDA]
 - 18,596 drug-protein edges
 - 719,402 protein-protein edges: Physical, metabolic enzyme-coupled, and signaling interactions
 - Drug and protein features: drugs' chemical structure, proteins' membership in pathways
- Gives multimodal network with over 5 million edges separated into 1,000 different edge types

Experimental setup



- Two main stages:
 1. Learn an **embedding** for every node in polypharmacy network
 2. Predict a score for **every drug-drug, drug-protein, protein-protein pair** in the test set based on the embeddings



Example: How likely will Simvastatin and Ciprofloxacin, when taken together, break down muscle tissue?

This Tutorial

- ✓ 1. Methods: Network diffusion, shallow network embeddings, graph neural networks, equivariant neural networks
- 2. Applications: Fundamental biological discoveries and precision medicine
- 3. Outlook: Future directions and Q&A session
- 4. Hands-on exercises: Demos, implementation details, tools, and tips

Resources

- Books & survey papers
 - William Hamilton, *Graph Representation Learning*
(morganclaypool.com/doi/abs/10.2200/S01045ED1V01Y202009AIM046)
 - Li et al., Graph Representation Learning for Biomedicine
(arxiv.org/abs/2104.04883)
- Keynotes
 - Michael Bronstein, “Geometric Deep Learning: The Erlangen Programme of ML” (ICLR 2021 keynote)
(youtube.com/watch?v=w6Pw4MOzMuo)
- Software & packages
 - PyTorch Geometric
 - NetworkX
 - Stanford Network Analysis Platform (SNAP)

Resources

- **Conferences & summer schools**
 - London Geometry and Machine Learning Summer School (logml.ai)
 - Learning on Graphs Conference (logconference.github.io)
- **Tutorials & code bases**
 - Pytorch Geometric Colab Notebooks (pytorch-geometric.readthedocs.io/en/latest/notes/colabs.html)
 - Zitnik Lab Graph ML Tutorials (github.com/mims-harvard/graphml-tutorials)
 - Stanford University's CS224 (web.stanford.edu/class/cs224w)
- **Datasets**
 - Precision Medicine Oriented Knowledge Graph (PrimeKG) (zitniklab.hms.harvard.edu/projects/PrimeKG)
 - Therapeutic Data Commons (TDC) (tdcommons.ai)
 - BioSNAP (snap.stanford.edu/biodata/)
 - Open Graph Benchmark (OGB) (ogb.stanford.edu)