

Part 2: Next-Gen Machine Learning for Network Biology



Marinka Zitnik
Stanford University



Two Lectures

Part 1: May 15, 2019, 2:30 pm - 4:00 pm

- Methodology: Shallow network embeddings:
 - Map nodes to low-dimensional features
- Resources: Data, tools, codebases
- Applications: PPIs, Disease pathways, Tissues

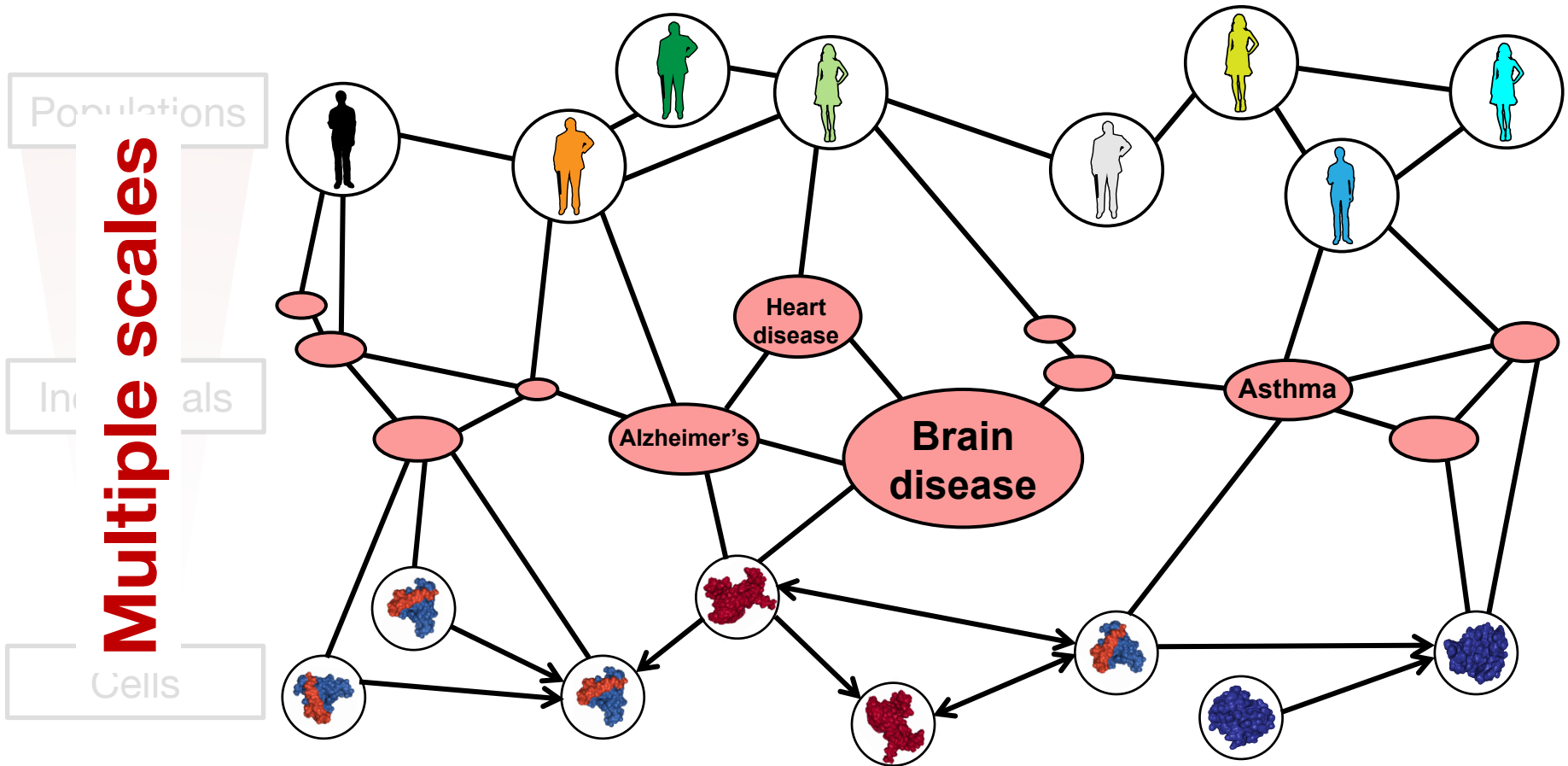


Part 2: May 16, 2019, 9:00 am – 10:30 am

- Methodology: Deep network embeddings:
 - Graph neural networks for rich biomedical graphs
- Resources: Data, practical advice and demos
- Applications: Polypharmacy, Drug repurposing

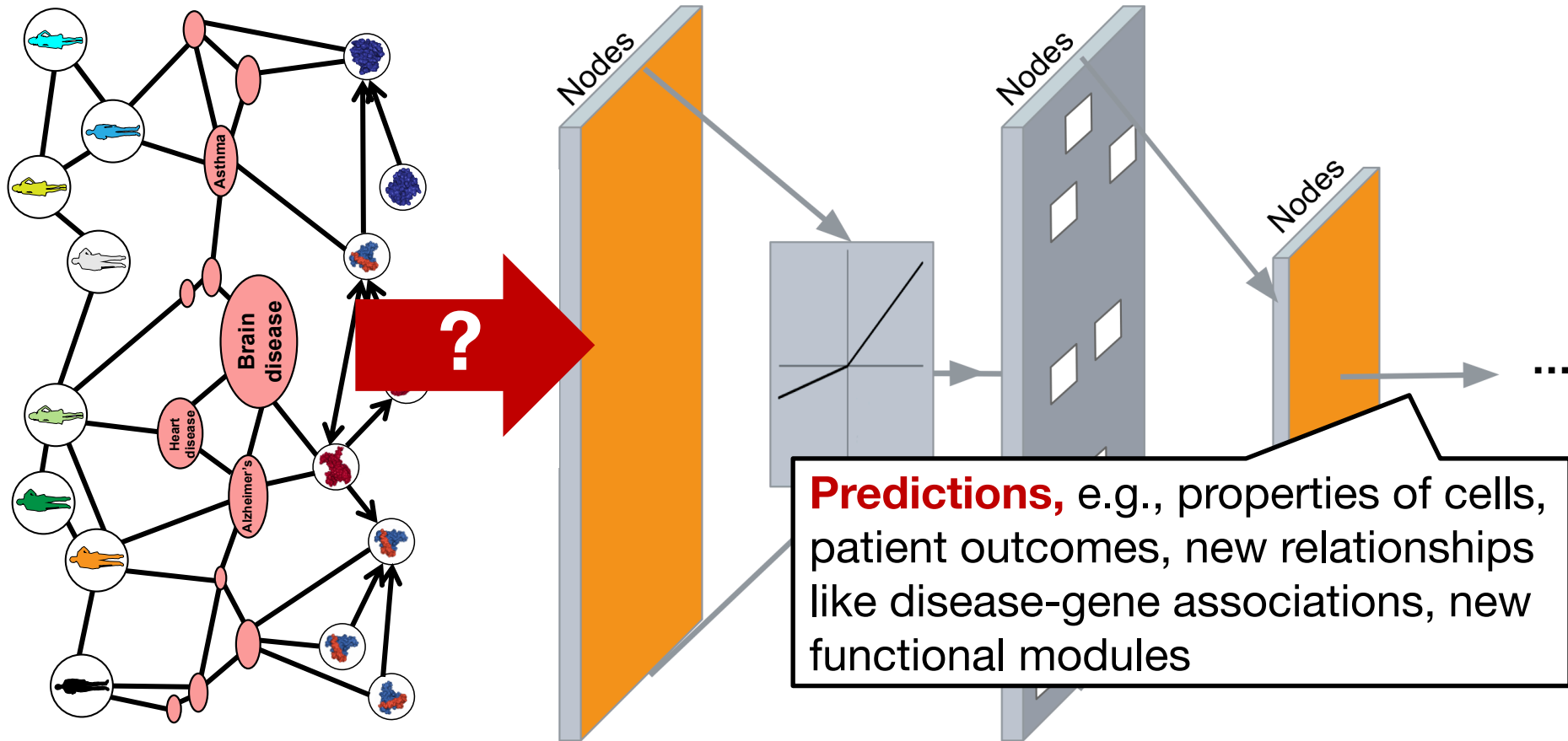


Networks allow for integration of biomedical data




Heterogeneity

How to learn deep models on biomedical networks?



Outline of this Lecture

- 1) Deep Graph Neural Networks 
- 2) Polypharmacy & Drug Interactions
- 3) Drug Repurposing
- 4) New Directions and Opportunities
- 5) Practical Advice and Demos

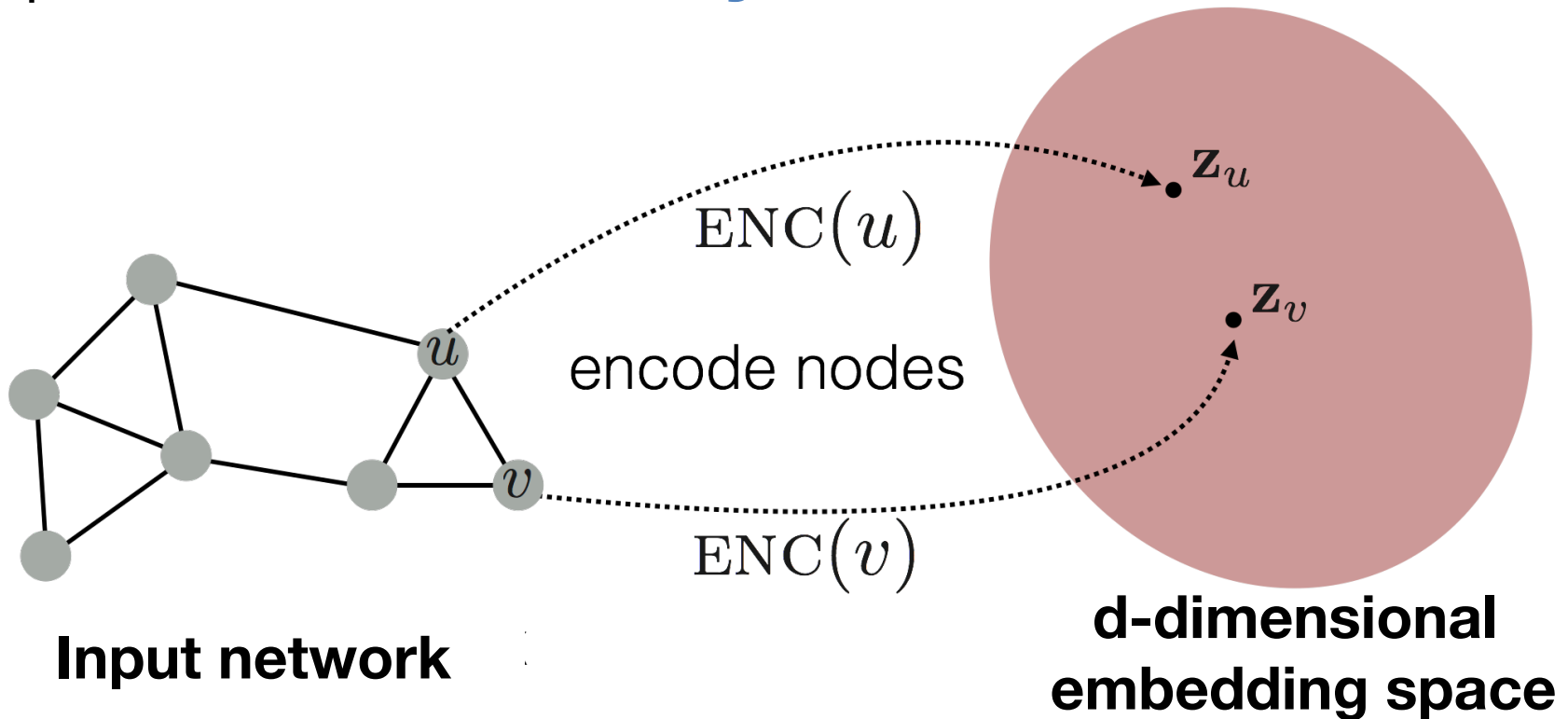
Deep Learning for Multimodal Networks

Based on material from:

- Zitnik et al. 2018. Deep Learning for Network Biology. *ISMB*.
- Zitnik et al. 2018. Modeling Polypharmacy Side Effects with Graph Convolutional Networks. *ISMB & Bioinformatics*.

Embedding Nodes

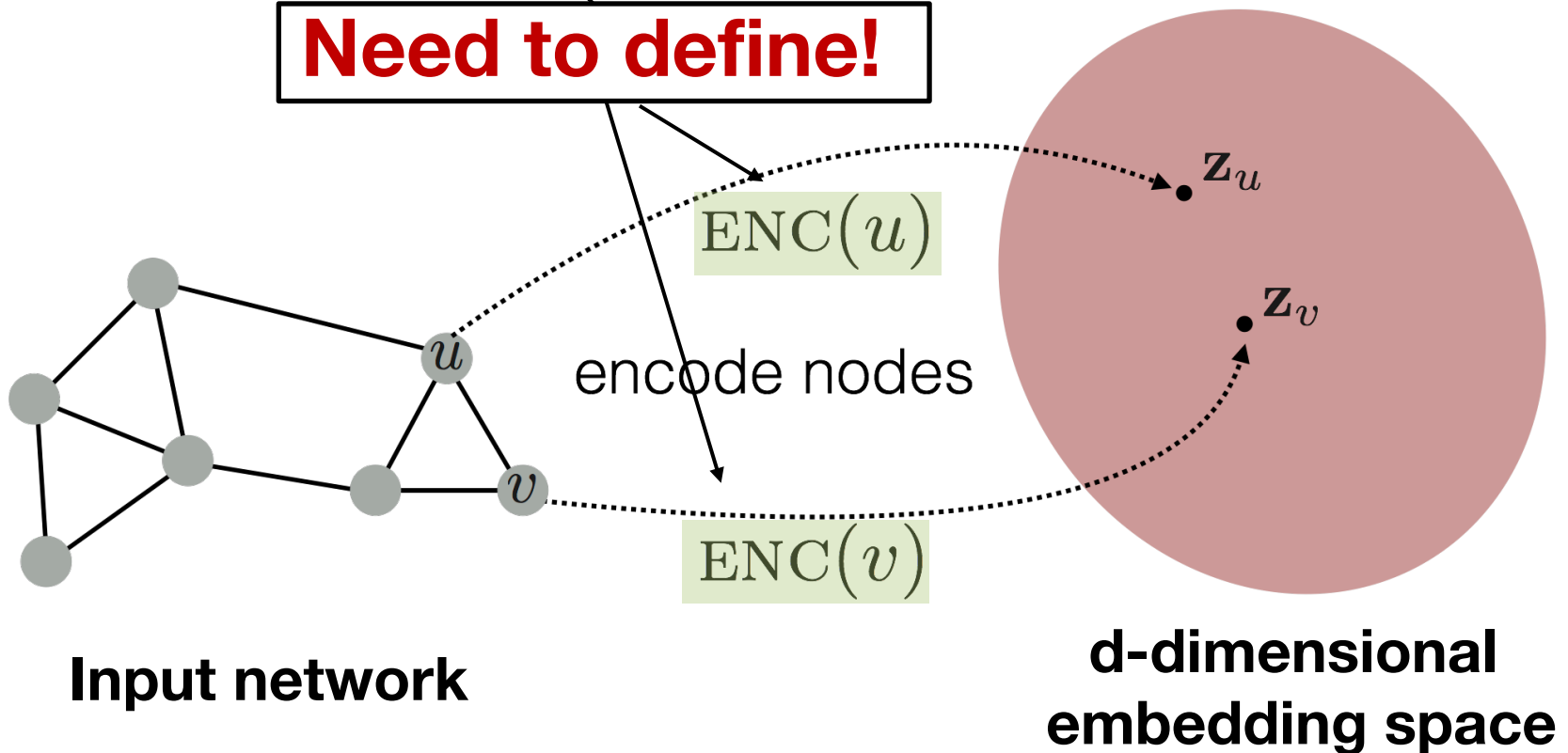
Goal: Map nodes so that **similarity in the embedding space** (e.g., dot product) approximates **similarity in the network**



Embedding Nodes

Goal: $\text{similarity}(u, v) \approx \mathbf{z}_v^\top \mathbf{z}_u$

Need to define!



Two Key Components

- **Encoder:** Map a node to a low-dimensional vector:

$$\text{ENC}(v) = \mathbf{z}_v$$

node in the input graph

d-dimensional embedding

- **Similarity function** defines how relationships in the input network map to relationships in the embedding space:

$$\text{similarity}(u, v) \approx \mathbf{z}_v^T \mathbf{z}_u$$

Similarity of u and v in the network

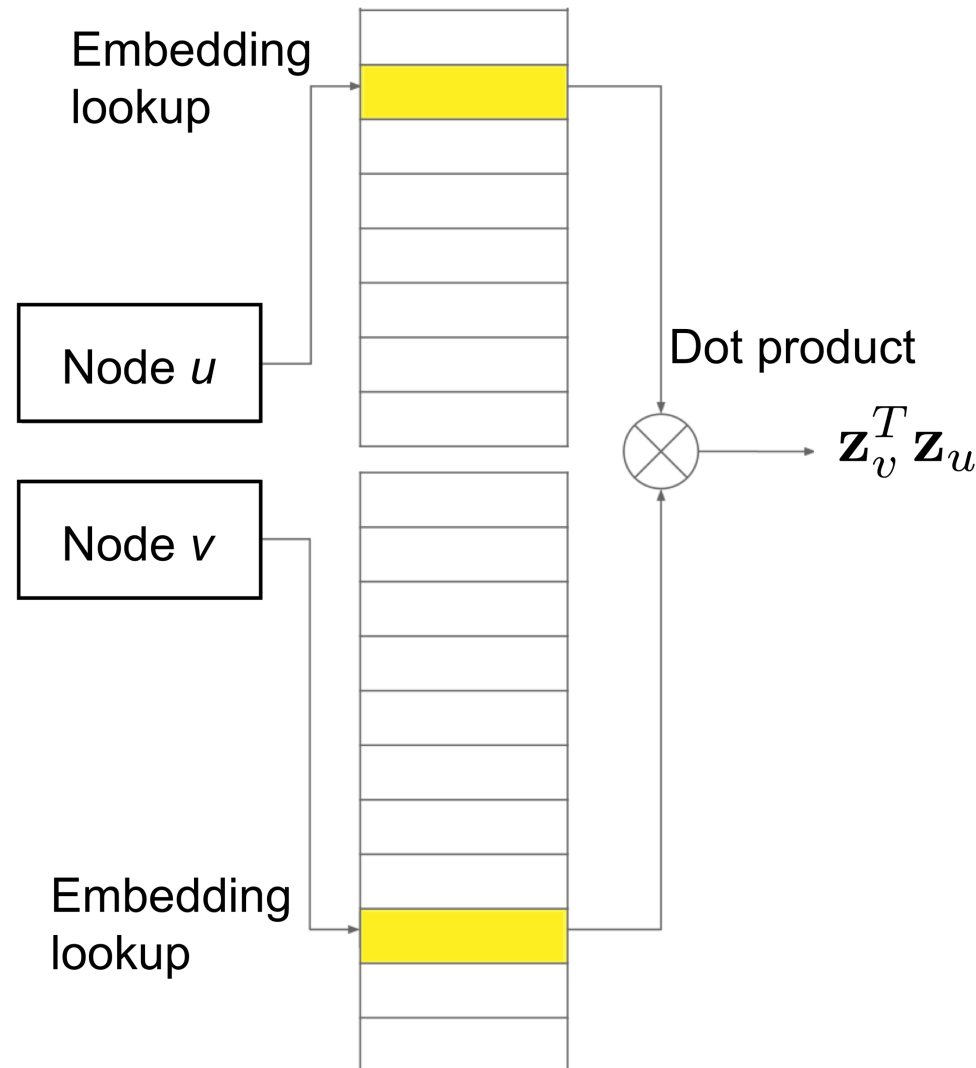
dot product between node embeddings

So Far: Shallow Encoders

Shallow encoders:

- One-layer of data transformation
- A single hidden layer maps node u to embedding \mathbf{z}_u via function f :

$$\mathbf{z}_u = f(\mathbf{z}_v, v \in N_R(u))$$



Shallow Encoders

Limitations of shallow encoding:

- **$O(|V|)$ 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, extra information:**
 - Many graphs are rich, have features for nodes/edges that we can and should leverage

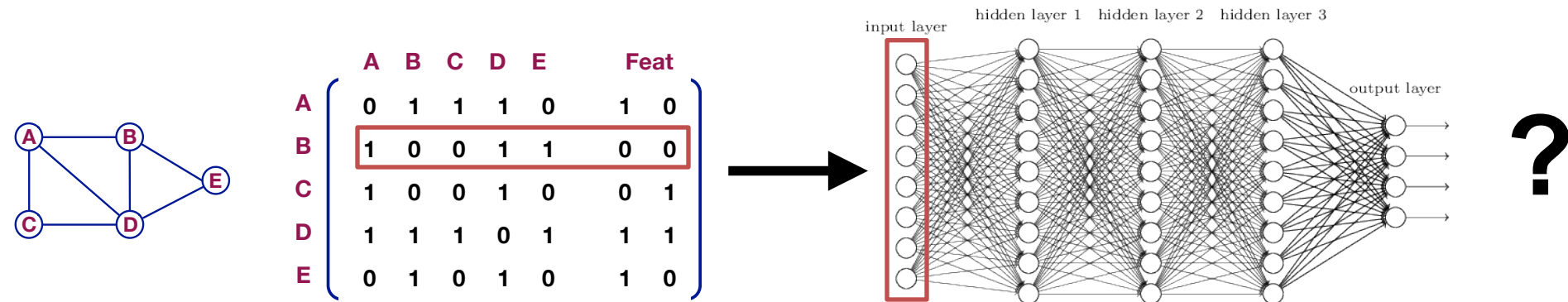
Deep Graph Encoders

Next: We discuss deep methods based on **graph neural networks:**

$\text{ENC}(v) =$ multiple layers of non-linear transformation of graph structure

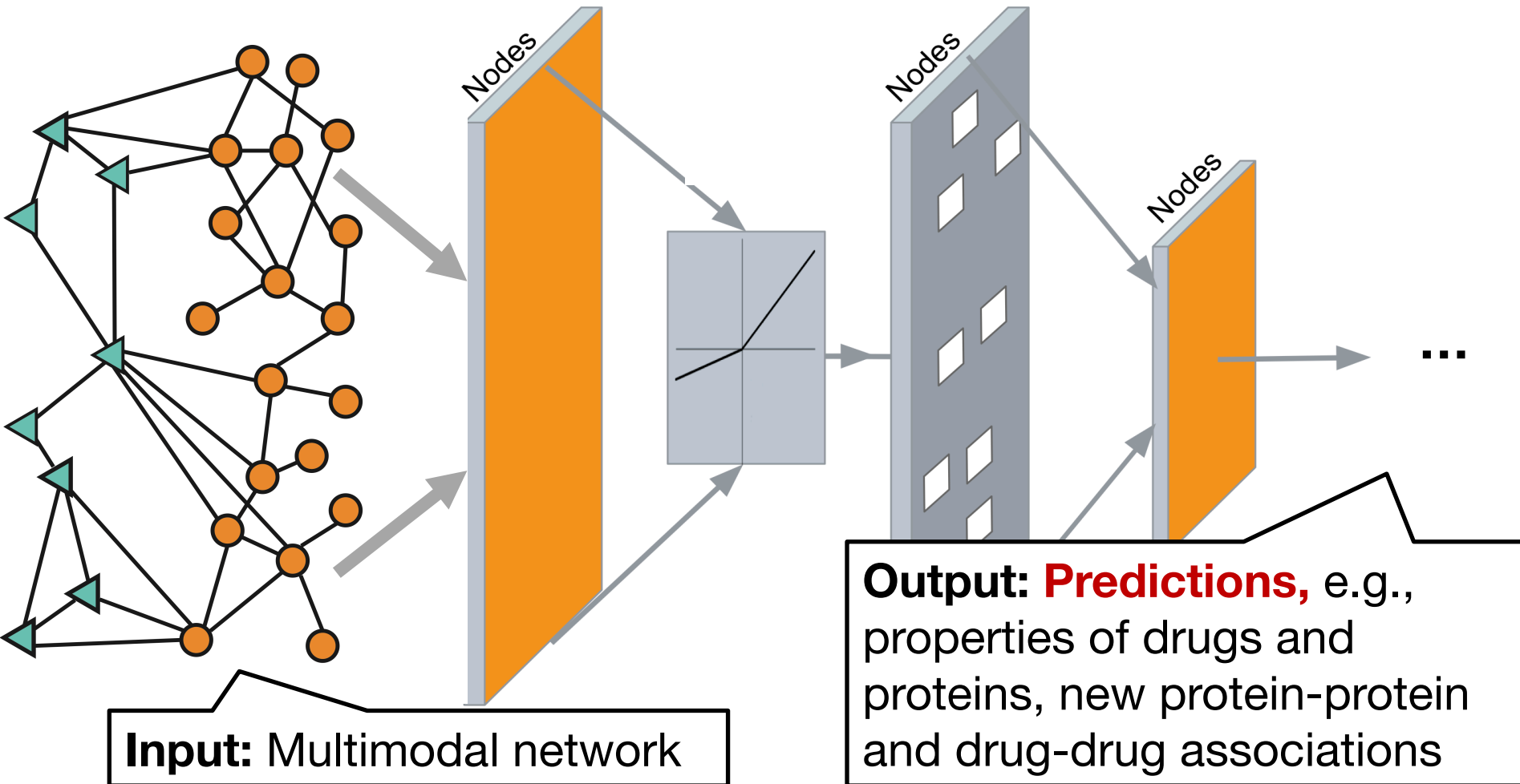
A Naïve Approach

- Join adjacency matrix and features
- Feed them into any classic neural net:



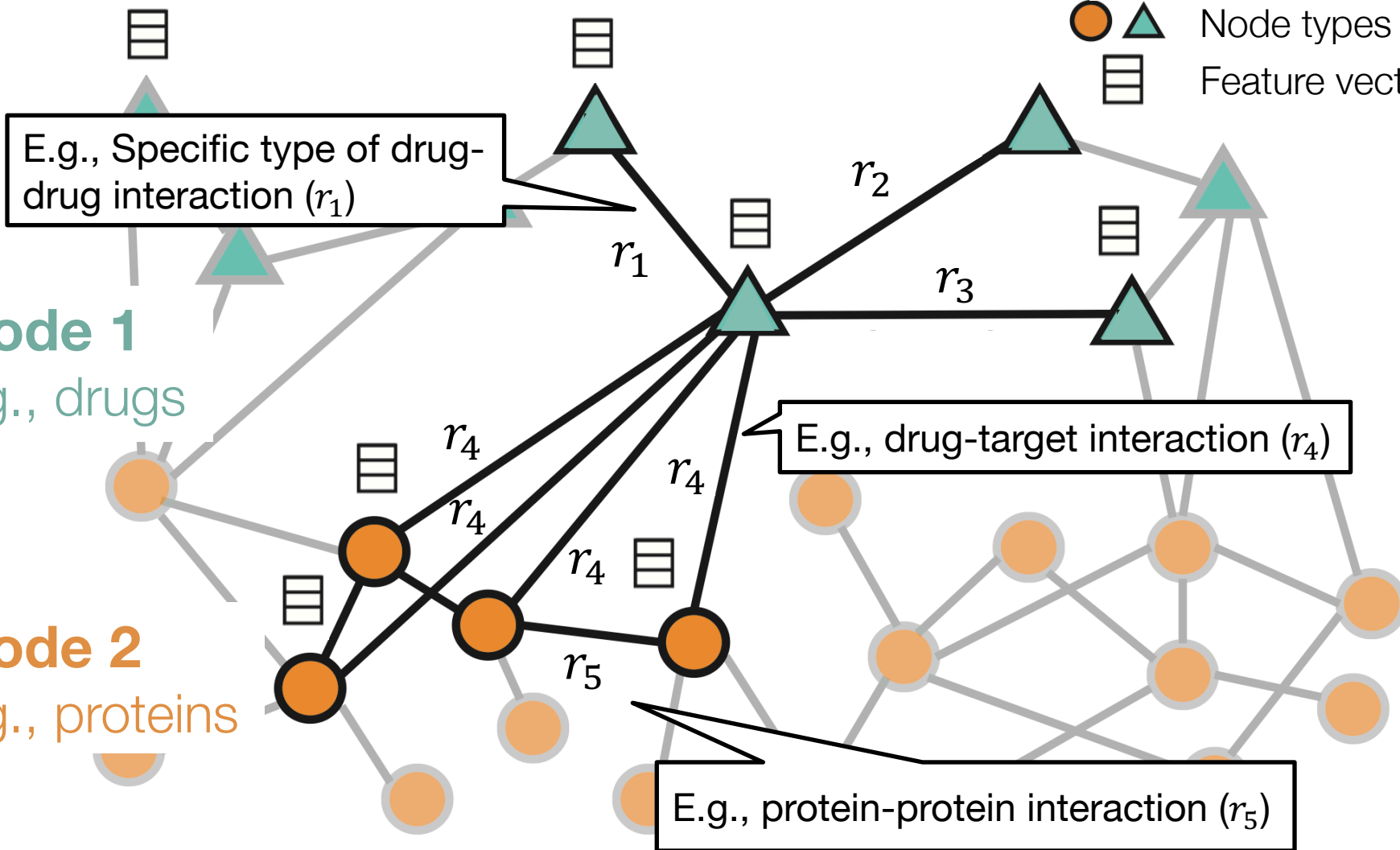
We need to generalize convolutions beyond simple lattices to multimodal networks and leverage node features/attributes

Approach: Deep Learning for Multimodal Networks



Multimodal Networks

r_i Edge type i
● ▲ Node types
≡ Feature vector



Preview of Results

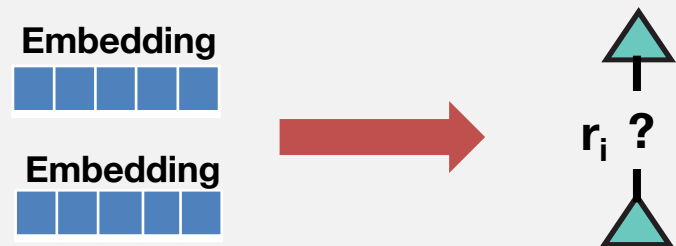
1. Used new approach to **predict safety and side effects of drug combinations in real patients:**
 - First-ever systematic and predictive study of **drug combinations**
 - **Follow-up research** on prostate cancer and validations in the clinic
2. Used new approach to **repurpose old drugs for new diseases:**
 - Outperforms baselines by up to 172%
 - **Correctly predicted drugs repurposed** at Stanford

Overview of our deep learning approach for multimodal networks

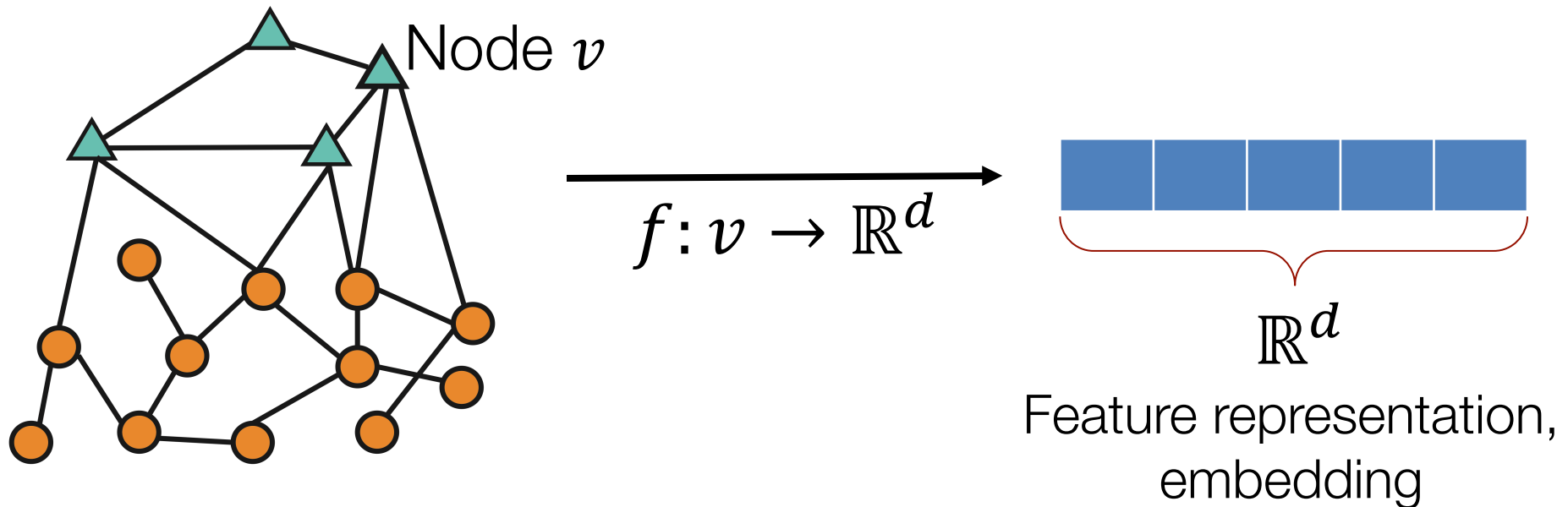
1. Encoder: Take a multimodal network and learn an *embedding* for every node



2. Decoder: Use the learned embeddings to predict labeled edges between nodes



Embedding Nodes



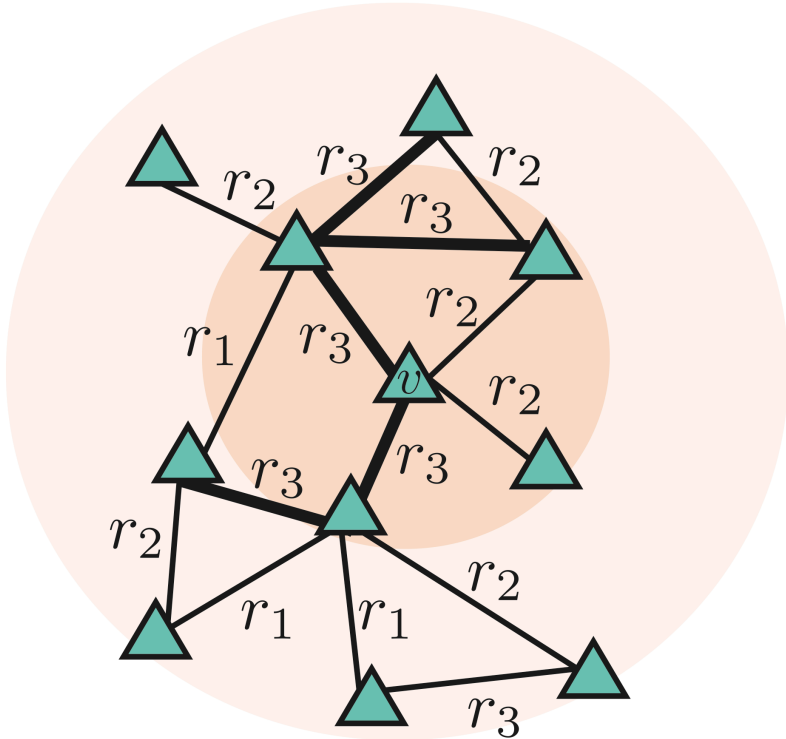
Objective: Map nodes to d -dimensional embeddings such that nodes with similar network neighborhoods are embedded close together

Next: How to learn mapping function f ?

Key Idea: Aggregate Neighbors

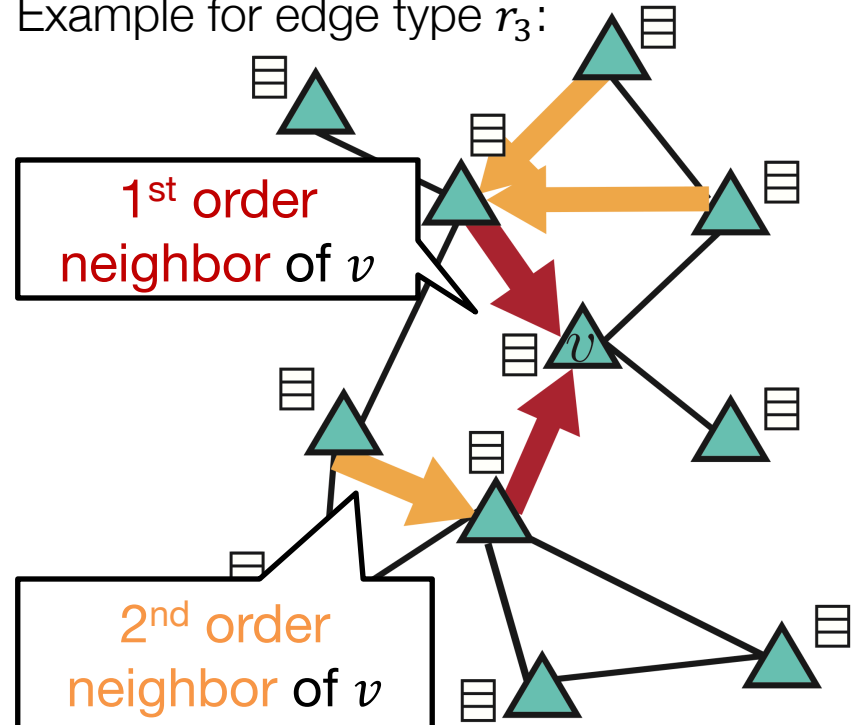
Generate embeddings based on **local network neighborhoods separated by edge type**

1) Determine a node's computation graph for each edge type



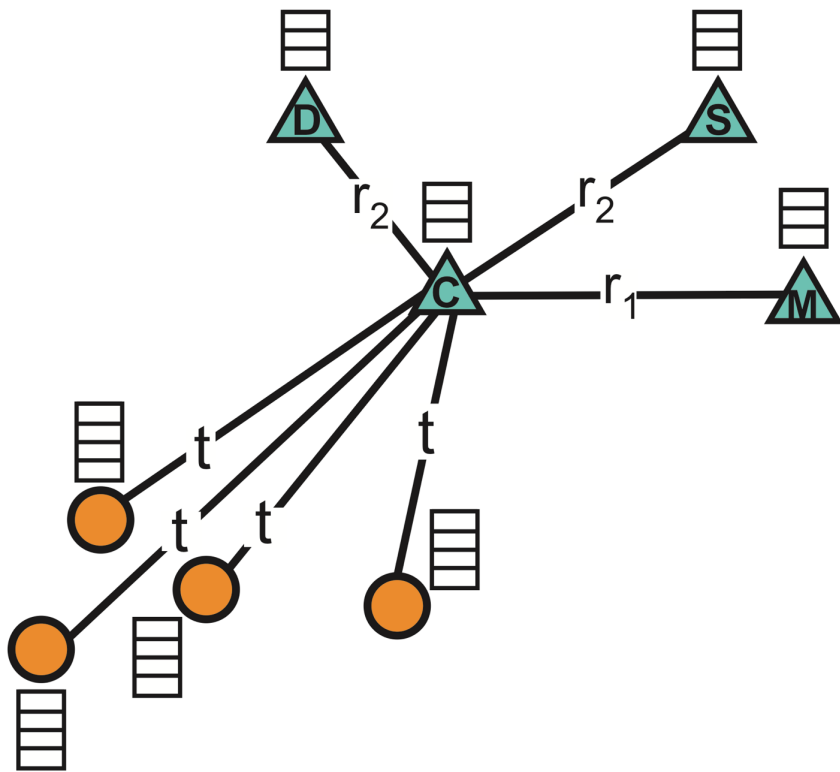
2) Learn how to transform and propagate information across computation graph

Example for edge type r_3 :

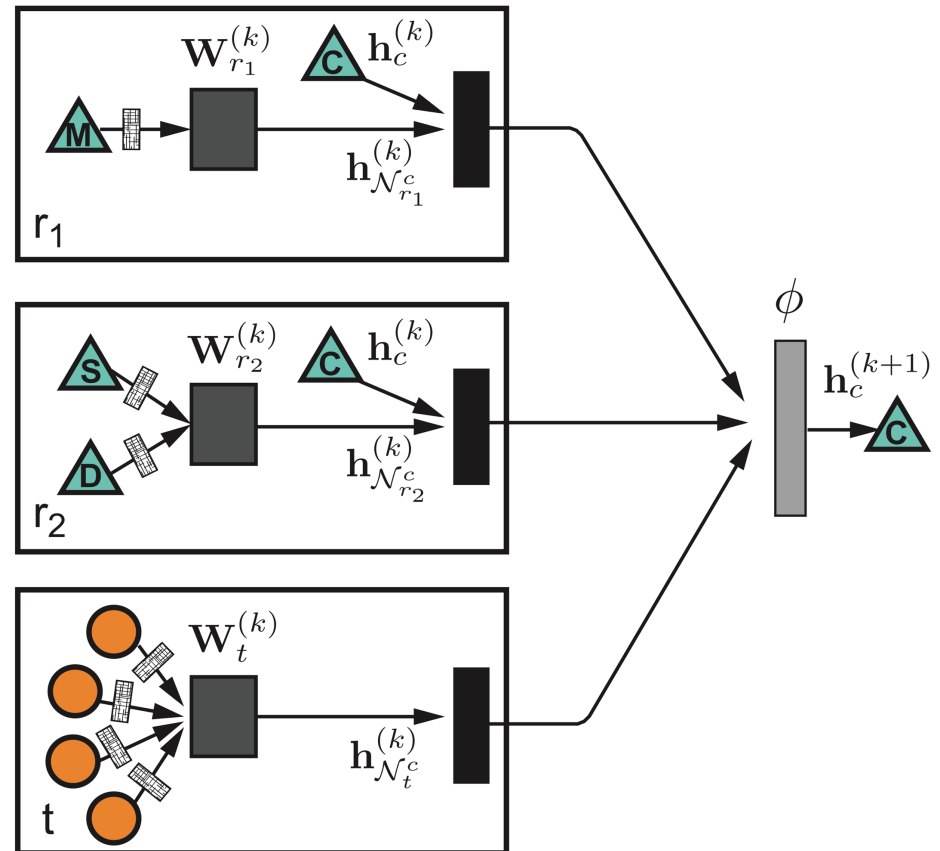


Example: Aggregate Neighbors

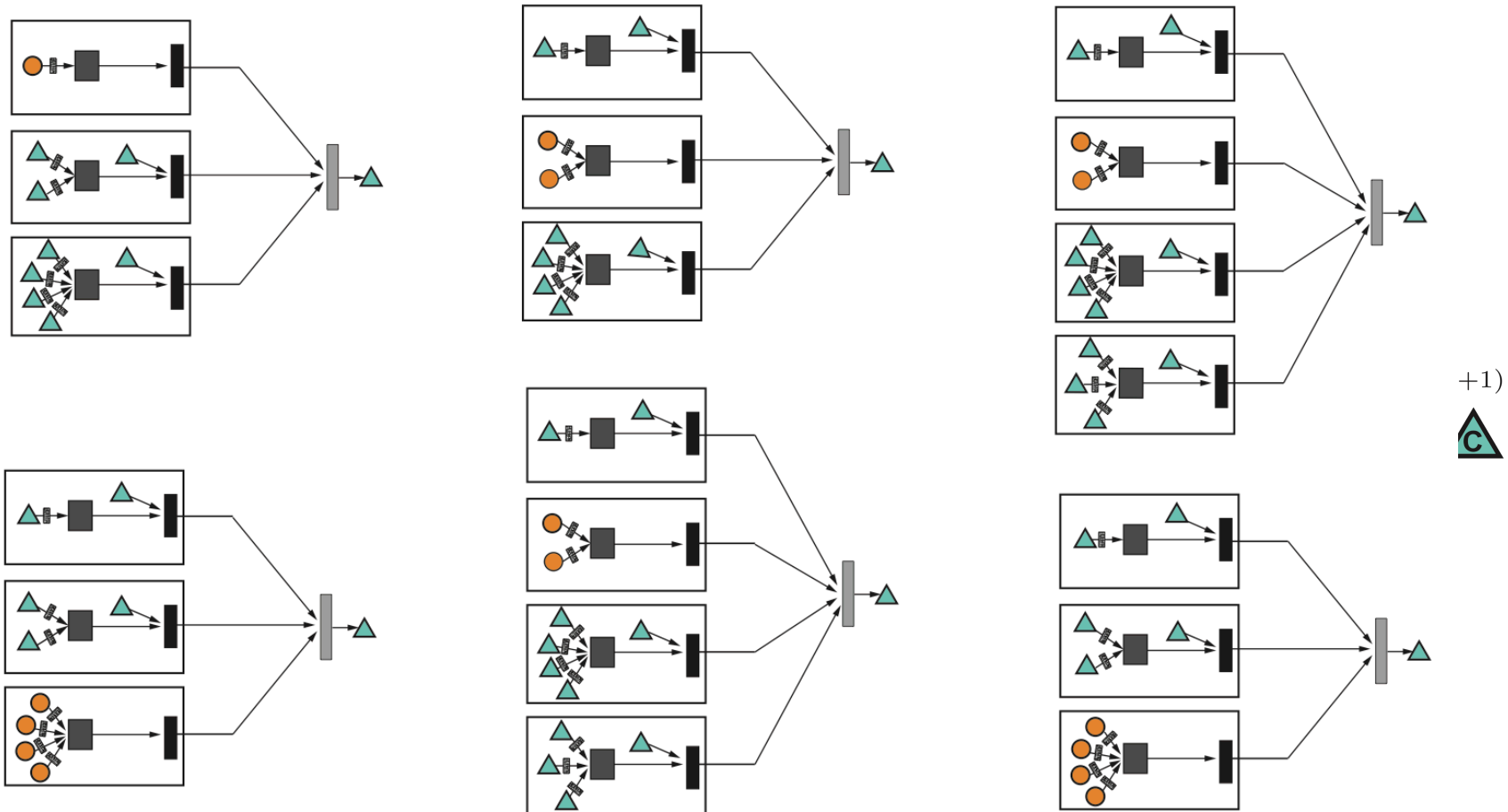
1st order network neighborhood of node C



1st order computation graph of node C

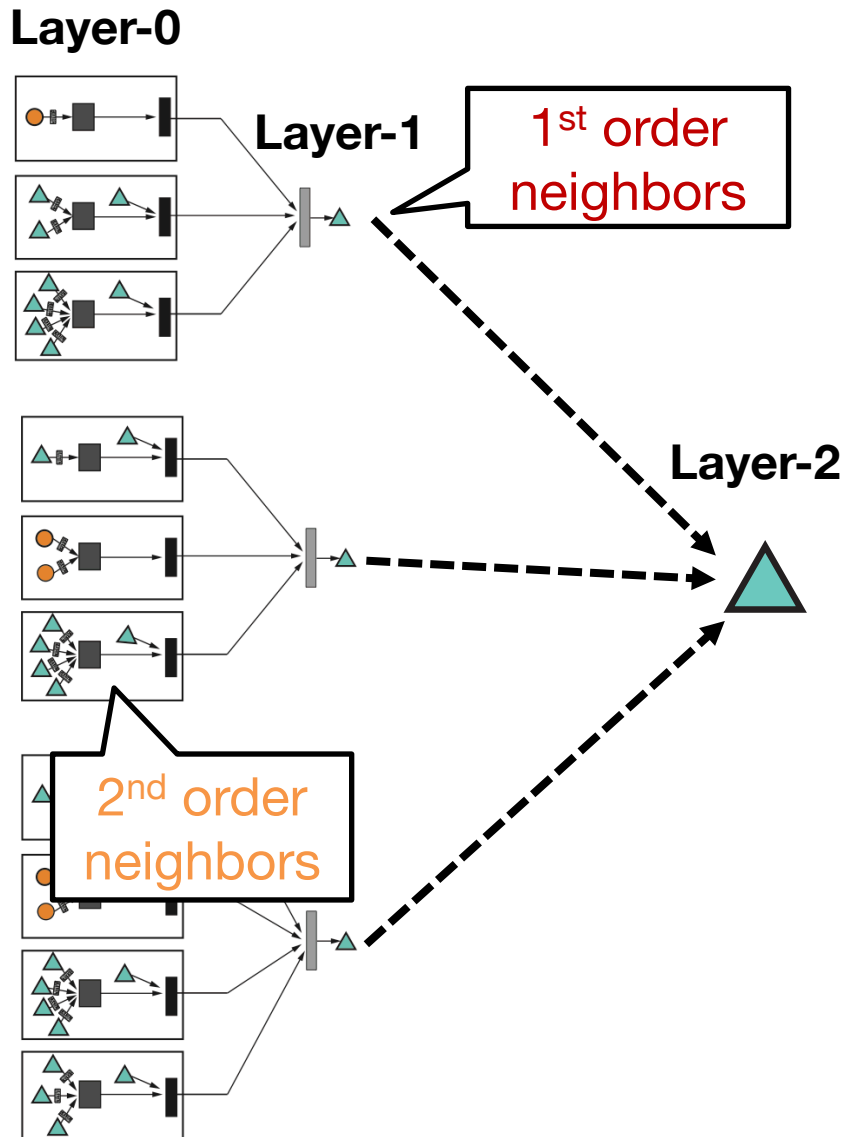


Every node learns how to aggregate its own neighbors



Every node defines a unique computation graph

Deep Model: Many Layers



Model can be of arbitrary depth:

- Nodes have embeddings at each layer
- Layer-0 embeddings are nodes' input features

Deep model with K layers:

- Convolves information across K^{th} order neighborhood
- Embedding of a node depends on nodes at most K hops away

Recap: Nodes with **similar network neighborhoods** are embedded **close together**

The Math: Deep Graph Encoder

Key element: Each node's computation graph defines a neural network with a different architecture

- Initial 0-th layer embeddings are equal to node features:

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

Aggregate neighbor's previous-layer embeddings, separated by edge type

Ability to integrate side information about nodes

- Per-layer update of node embeddings:

$$\mathbf{h}_v^{(k)} = \phi \left(\sum_r \sum_{u \in N_v^r} c_r^{uv} \mathbf{W}_r^{(k-1)} \mathbf{h}_u^{(k-1)} + c_r^v \mathbf{h}_v^{(k-1)} \right) \quad k = 1, \dots, K$$

Previous-layer embedding of v

- Embeddings after K layers of neighborhood aggregation:

$$\mathbf{z}_v = \mathbf{h}_v^{(K)}$$

Normalization constant, fixed e.g., $1/|N_v^r|$, or learned

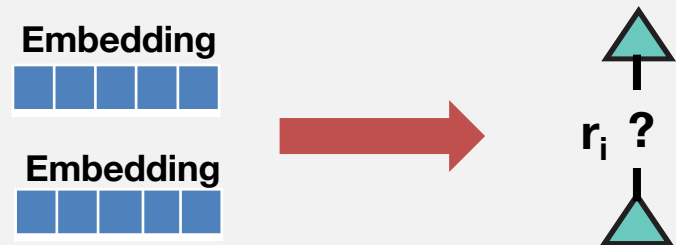
$\mathbf{W}_r^{(k)}$ Part

Overview of our deep learning approach for multimodal networks

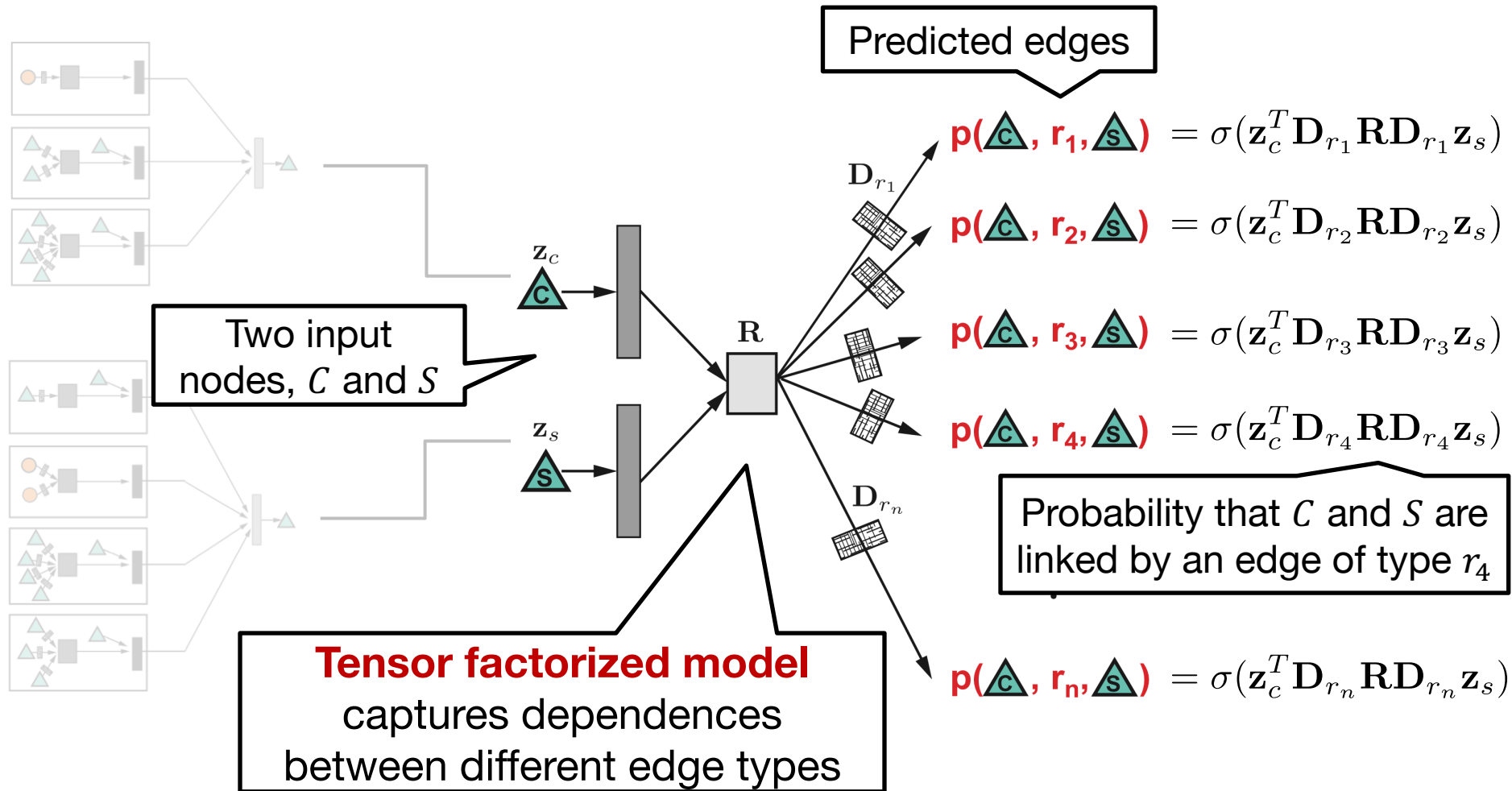
1. Encoder: Take a multimodal network and learn an *embedding* for every node



2. Decoder: Use the learned embeddings to predict labeled edges between nodes



Heterogeneous Edge Decoder



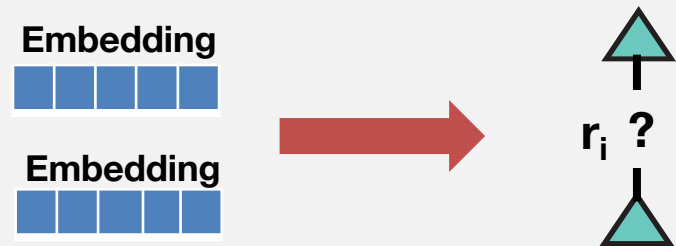
R, D_{r_i} Parameter weight matrices

Overview of our deep learning approach for multimodal networks

✓ **1. Encoder:** Take a multimodal network and learn an *embedding* for every node



✓ **2. Decoder:** Use the learned embeddings to predict labeled edges between nodes



Training the model: Feed embeddings into any loss function and run stochastic gradient descent to train weight parameters:

- Use a loss based on e.g., random walks, node proximity in the graph
- Directly train the model for a supervised task (e.g., node classification)

Recap: Deep Learning for Multimodal Networks

Key new insights:

- Advances Graph Neural and Graph Convolutional models [e.g., Kipf et al., ICLR'17; Hamilton et al., NIPS'17] by treating **multimodal networks**
- Generates **powerful deep embeddings**, unlike shallow embedding methods [e.g., Matrix factorization, Node2vec, DeepWalk]
- Can define and tackle **new prediction problems** [e.g., graph-level classification and subgraph-level link prediction]

We can now apply deep learning much more broadly, to any multimodal network

New frontiers for applications in **biology** and **medicine**

Outline of this Lecture

- 1) Deep Graph Neural Networks ✓
- 2) Polypharmacy & Drug Interactions
- 3) Drug Repurposing
- 4) New Directions and Opportunities
- 5) Practical Advice and Demos



Polypharmacy and Drug-Drug Interactions

Based on material from:

- Zitnik et al. 2018. Deep Learning for Network Biology. *ISMB*.
- Zitnik et al. 2018. Modeling Polypharmacy Side Effects with Graph Convolutional Networks. *ISMB & Bioinformatics*.

Polypharmacy

Patients **take multiple drugs** to treat **complex or co-existing diseases**

46% of people over 65 years take more than 5 drugs






Many take more than **20** drugs to treat heart diseases, depression or cancer

15% of the U.S. population affected by unwanted side effects

Annual costs in treating side effects exceed **\$177** billion in the US alone








Reports on Unwanted Side Effects

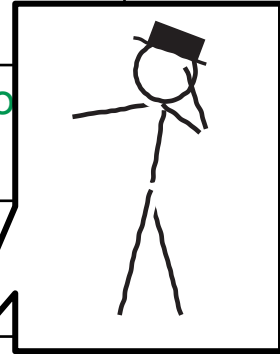
The FDA Adverse Event Reporting System (FAERS)

Drugs taken	Unwanted side effects
	Peliosis hepatis (5%), Heart rate increased (10%), Aortic aneurysm (3%)
	Joint stiffness (30%), Joint swelling (10%), Bone marrow fibrosis (3%)
	Anaemia (15%), Bone marrow fibrosis (5%), Intestinal ulcer (5%)
	Anaemia (15%), Bone marrow fibrosis (5%), Intestinal ulcer (5%), Joint stiffness (30%), Joint swelling (10%), Colon cancer (4%), Fatigue (40%)
	Peliosis hepatis (5%), Heart rate increased (10%), Aortic aneurysm (3%), Joint stiffness (30%), Joint swelling (10%), Bone marrow fibrosis (3%)
...	...

Reports on Unwanted Side Effects

The FDA Adverse Event Reporting System (FAERS)

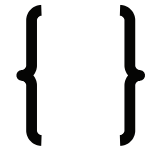
Drugs taken	Unwanted side effects
	Peliosis hepatis (5%), Heart rate increased (10%), Aortic aneurysm (3%)
	Joint stiffness (30%), Joint swelling (10%), Bone marrow fibrosis (3%)
	Anaemia (15%), Bone marrow fibrosis (5%), Intestinal (5%)
 	Anaemia (15%), Bone marrow fibrosis (5%), Intestinal ulcer (5%), Joint stiffness (30%), Joint swelling (10%), Colon cancer (4%), Fatigue (40%)
 	Peliosis hepatis (5%), Heart rate increased (10%), Aortic aneurysm (3%), Joint stiffness (30%), Joint swelling (10%), Bone marrow fibrosis (3%)
...	...



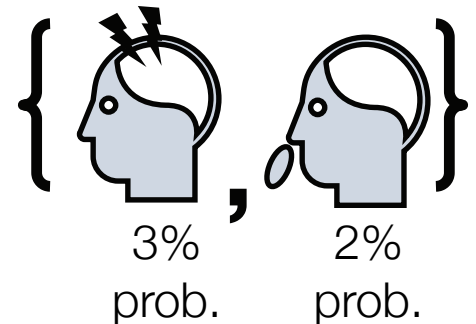
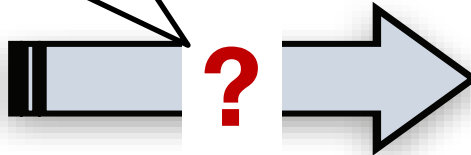
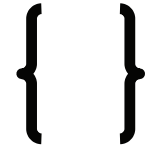
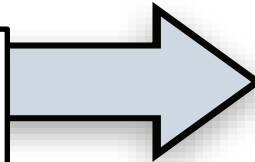
Unexpected Drug Interactions

Co-prescribed drugs

Side Effects



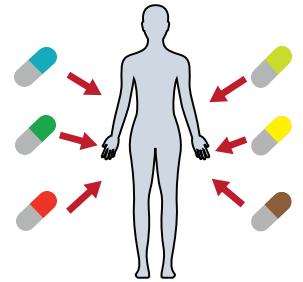
Task: How likely will a particular combination of drugs lead to a particular side effect?



Why is modeling polypharmacy hard?

Combinatorial explosion

- >13 million possible combinations of 2 drugs
- >20 billion possible combinations of 3 drugs



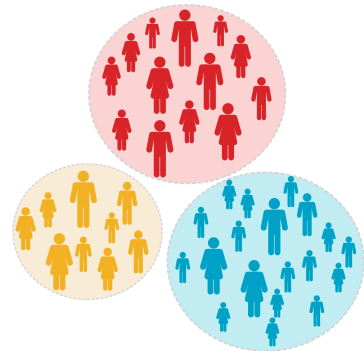
Non-linear & non-additive interactions

- 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

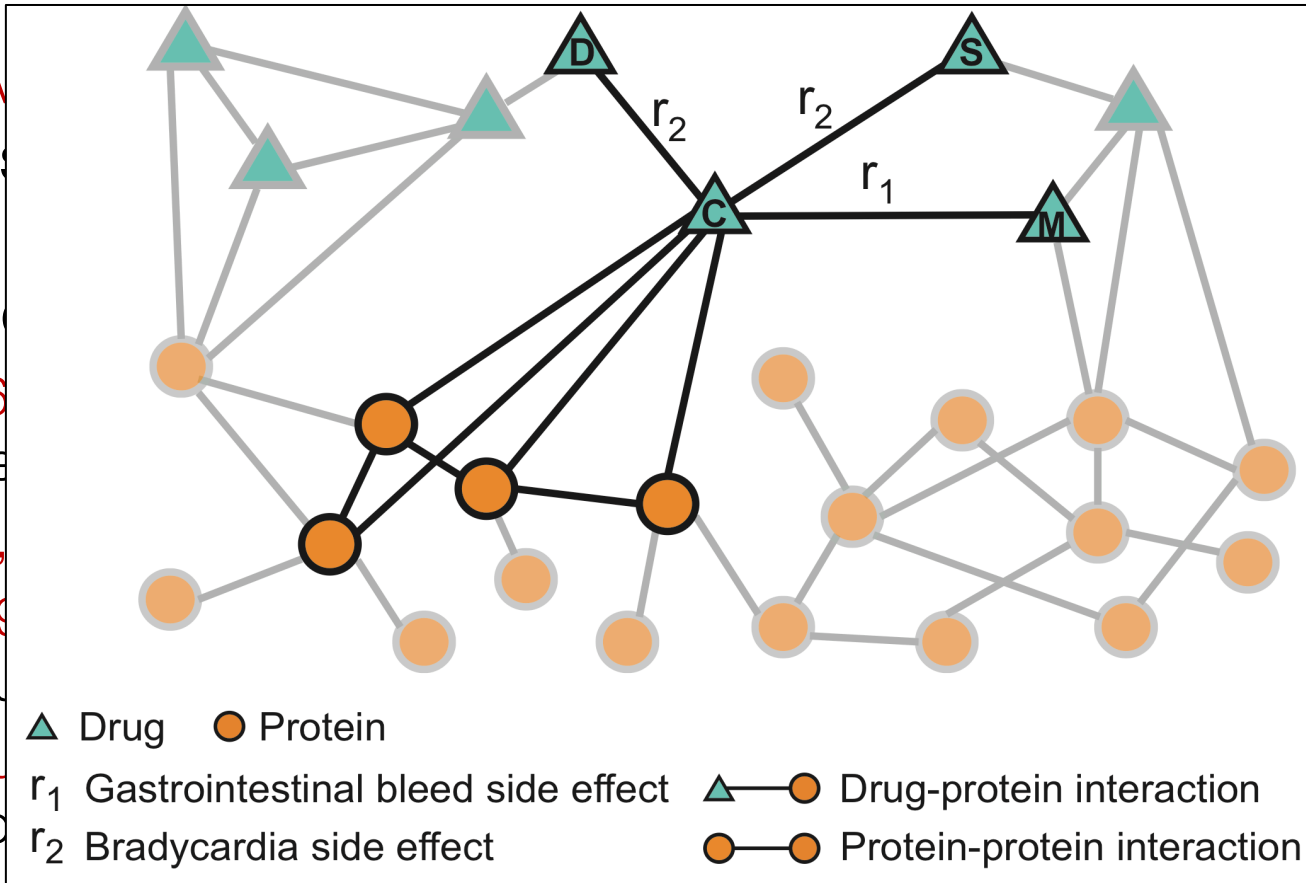


We need Polypharmacy Dataset

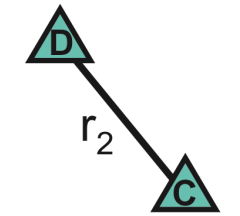
Objective
all drugs

We build

- 4,6
- even
- 18,
- 719
- cou
- Dr
- pro



Drug-drug



Drug-protein



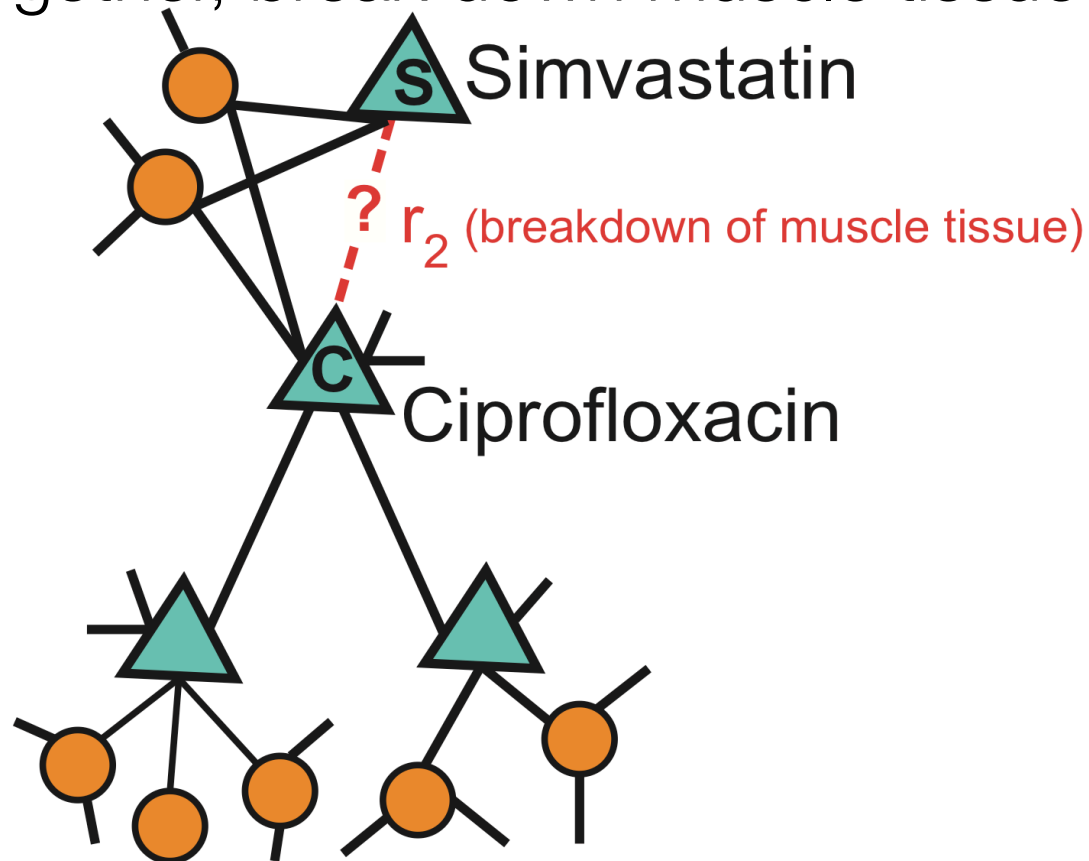
Protein-protein



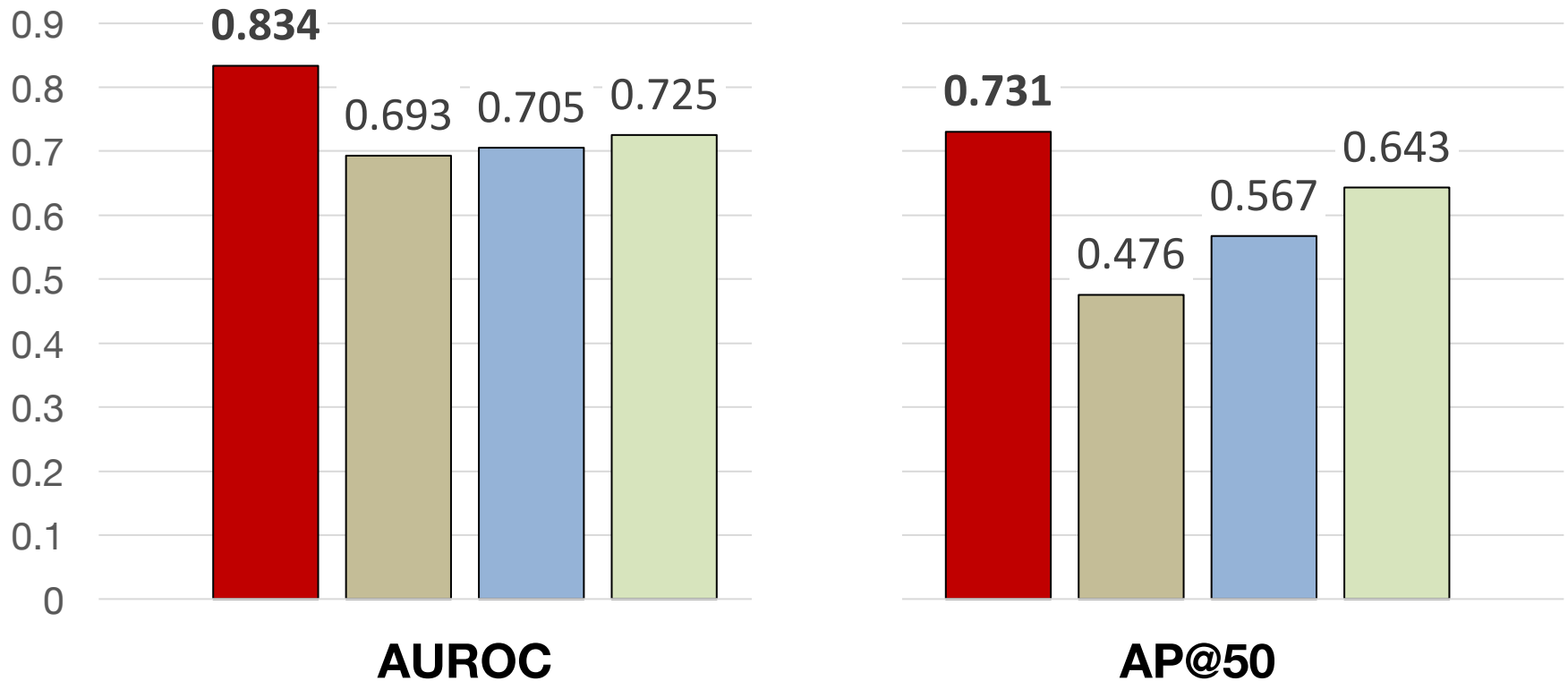
Gives multimodal network with over 5 million edges separated into 1,000 different edge types

We apply our deep approach to the polypharmacy network

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



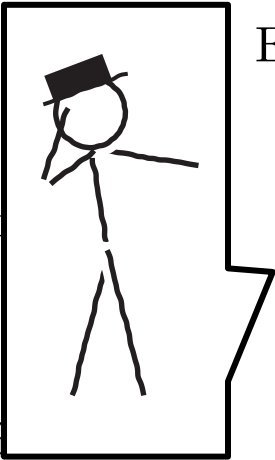
Results: Side Effect Prediction



- Our method (Decagon)
- RESCAL Tensor Factorization [Nickel et al., ICML'11]
- Multi-relational Factorization [Perros, Papalexakis et al., KDD'17]
- Shallow Network Embedding [Zong et al., Bioinformatics'17]

Novel Predictions

- Train deep model on data generated **prior to 2012**
- How many **predictions** have been **confirmed after 2012?**

Rank	Drug	Drug	Side effect	Evidence found
1	Pyrimethamine	Aliskiren	Sarcoma	
2	Tigecycline	Bimatoprost	Autonomic r	
3	Telangiectases	Omeprazole	Dacarbazine	
4	Tolcapone	Pyrimethamine	Blood brain	
5	Mifepistole	Desferrioxamine	Glucose head	
				ular acidosis
				Cerebral thrombosis
8	Atorvastatin	Amlodipine	Muscle inflammation	
9	Aliskiren	Tioconazole	Breast inflammation	
10	Estradiol	Nadolol	Endometriosis	

Case Report

Severe Rhabdomyolysis due to Presumed Drug Interactions between Atorvastatin with Amlodipine and Ticagrelor

Utility of Predictions in the Clinic

Clinical validation: Drug-drug interaction markers, lab values, and surrogates



Robert Martin
22 Feb 1953 Male

Medication List Simple List Timeline Back to the Book Feedback Task List

show brand prn current (16) all (23)

Medication	Brand	Dose	Frequency	Quantity	Refills	Condition	Provider	Prescribed	2011	2012	2013	2014	Renew by
beclomethasone HFA	QVAR HFA	2 puffs	bid	12	12	Asthma	Barnes	19 Feb 2011	█				19 Sep 2013
chlorthalidone		25 mg	1 daily	90	3	Hypertension	Barnes	19 Sep 2006	█				19 Sep 2013
insulin glargine	Lantus	28 u	daily	90	11	Diabetes	Ballard	19 Nov 2012			█		19 Sep 2013
metformin		1000 mg	1 bid	180	3	Diabetes	Barnes	4 Mar 2008	█				19 Sep 2013
naproxen	Aleve	500 mg	1 bid	90	0	Rheumatoid arthritis	Barnes	4 Mar 2008	█				19 Sep 2013

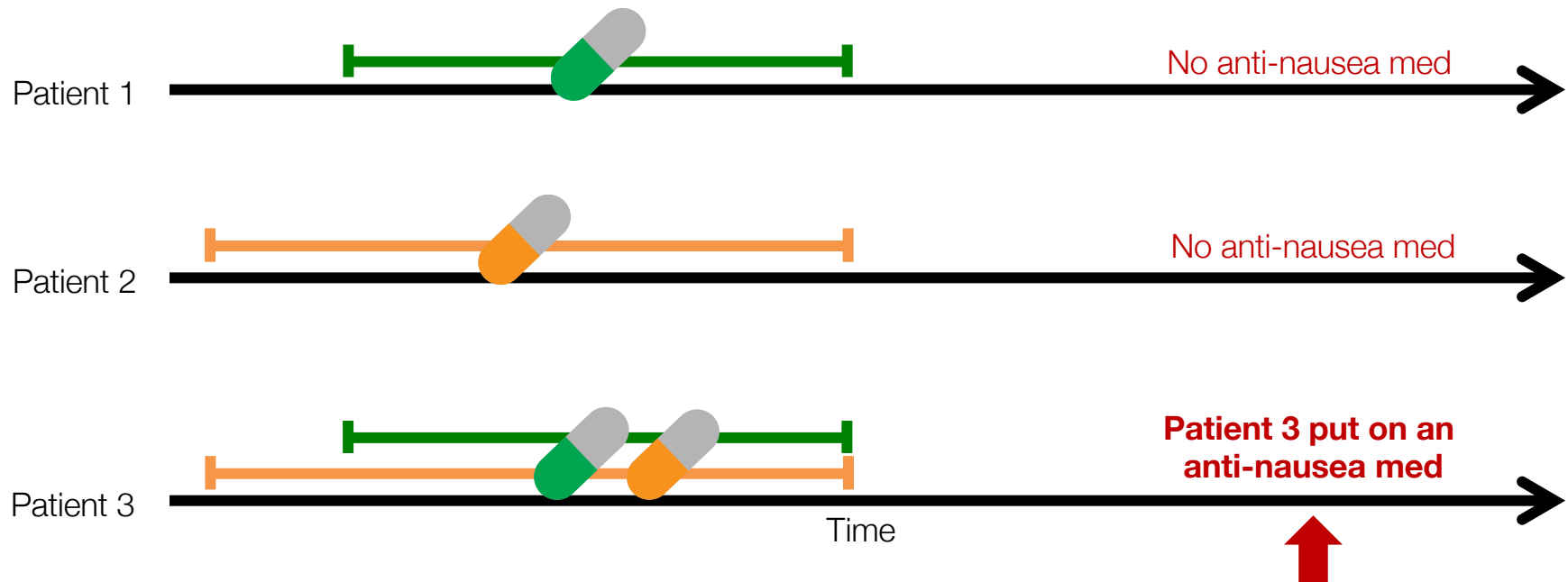
The approach is used for **personalized treatments** and **design of new combinatorial drug therapies**

terbinafine 250 mg 1 daily 84 0 Onychomycosis Foote 30 Jul 2013 19 Oct 2013


Validation in the Clinic: Key Idea

Question: Is it a good idea to prescribe a particular combination of drugs to a particular patient?

- E.g., Prediction: { ,  } cause nausea as a side effect

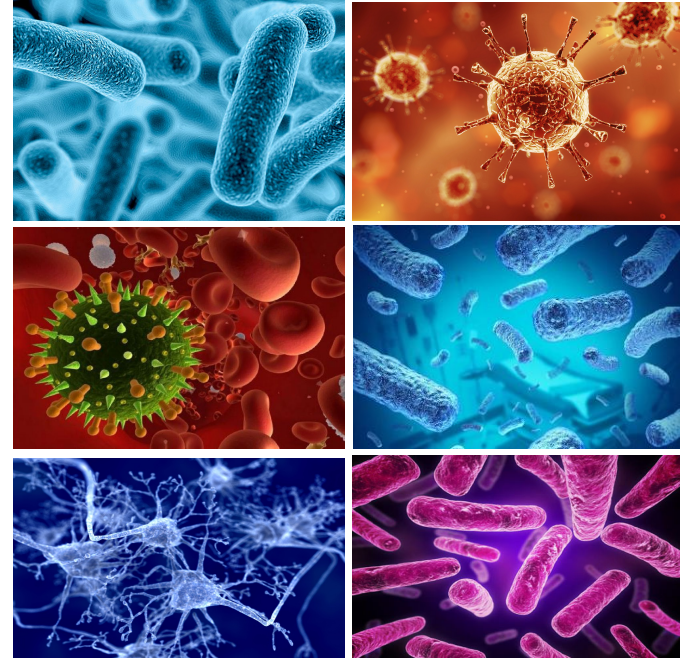


Outline of this Lecture

- 1) Deep Graph Neural Networks ✓
- 2) Polypharmacy & Drug Interactions ✓
- 3) Drug Repurposing 
- 4) New Directions and Opportunities
- 5) Practical Advice and Demos

Computational Drug Repurposing

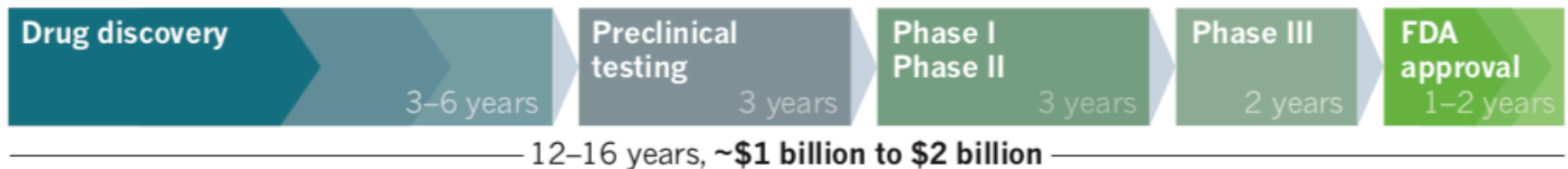
Computational Drug Discovery



Goal: Find which diseases a new drug (molecule) could treat

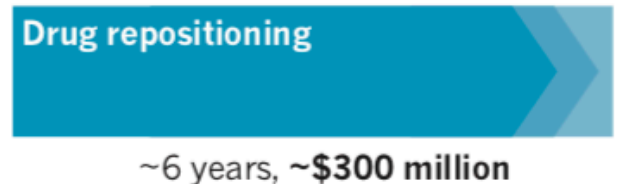
New tricks for old drugs

Faced with skyrocketing costs for developing new drugs, researchers are looking at ways to repurpose older ones — and even some that failed in initial trials.



A SHORTER TIMESCALE

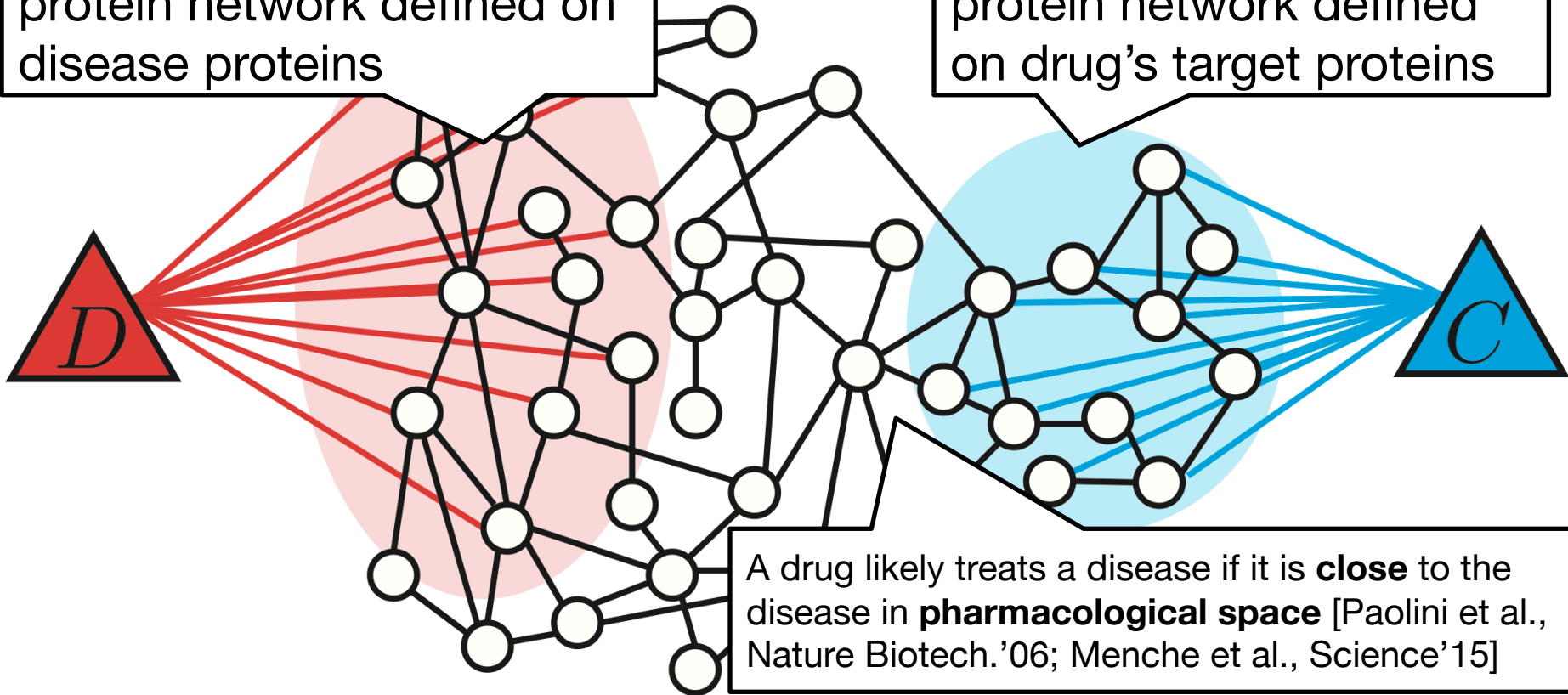
Because most repositioned drugs have already passed the early phases of development and clinical testing, they can potentially win approval in less than half the time and at one-quarter of the cost.



Key Insight: Subgraphs

Disease: Subgraph of rich protein network defined on disease proteins

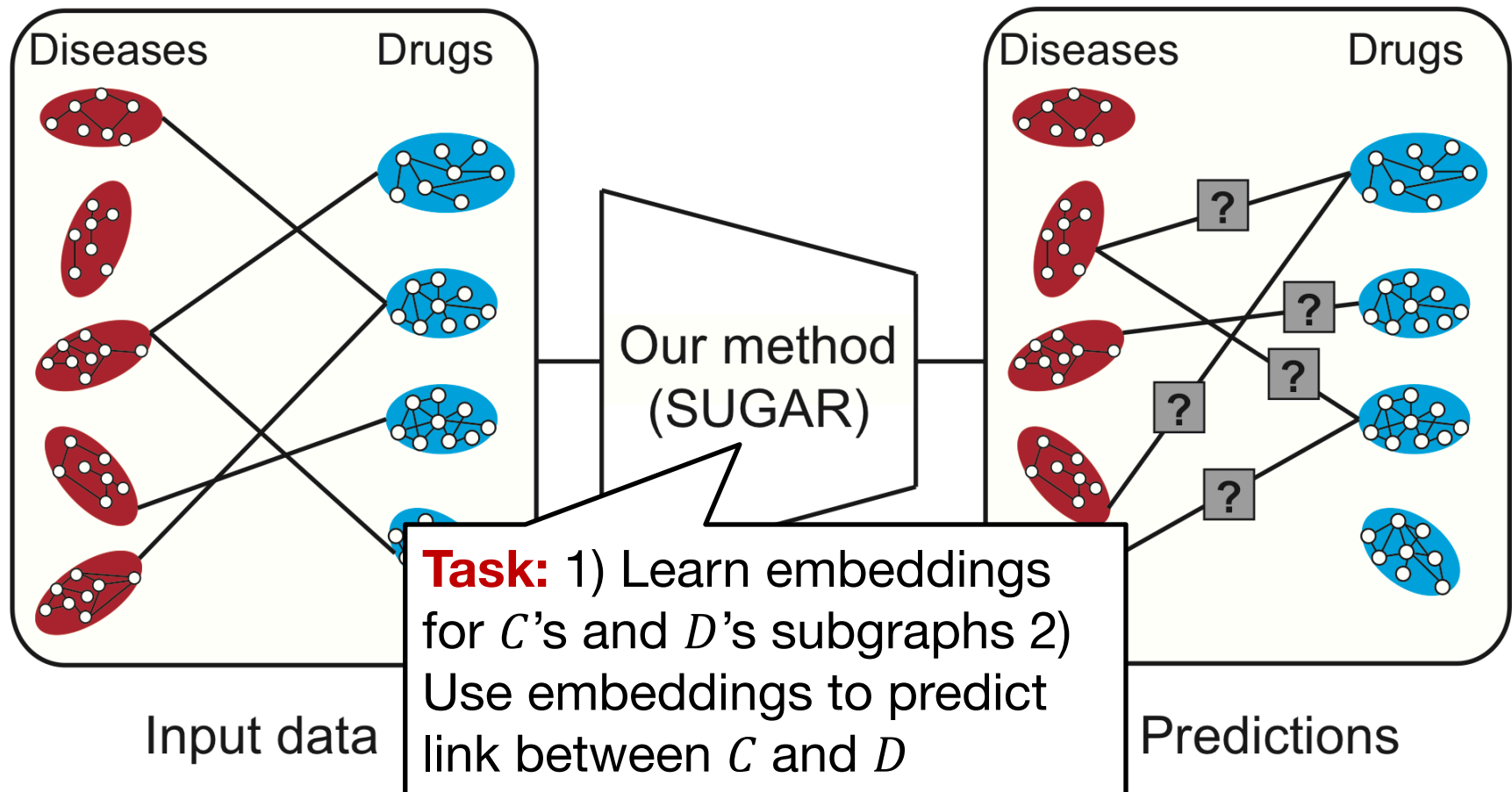
Drug: Subgraph of rich protein network defined on drug's target proteins



Idea: Use the paradigm of embeddings to operationalize the concept of closeness in pharmacological space

Link Prediction Between Subgraphs

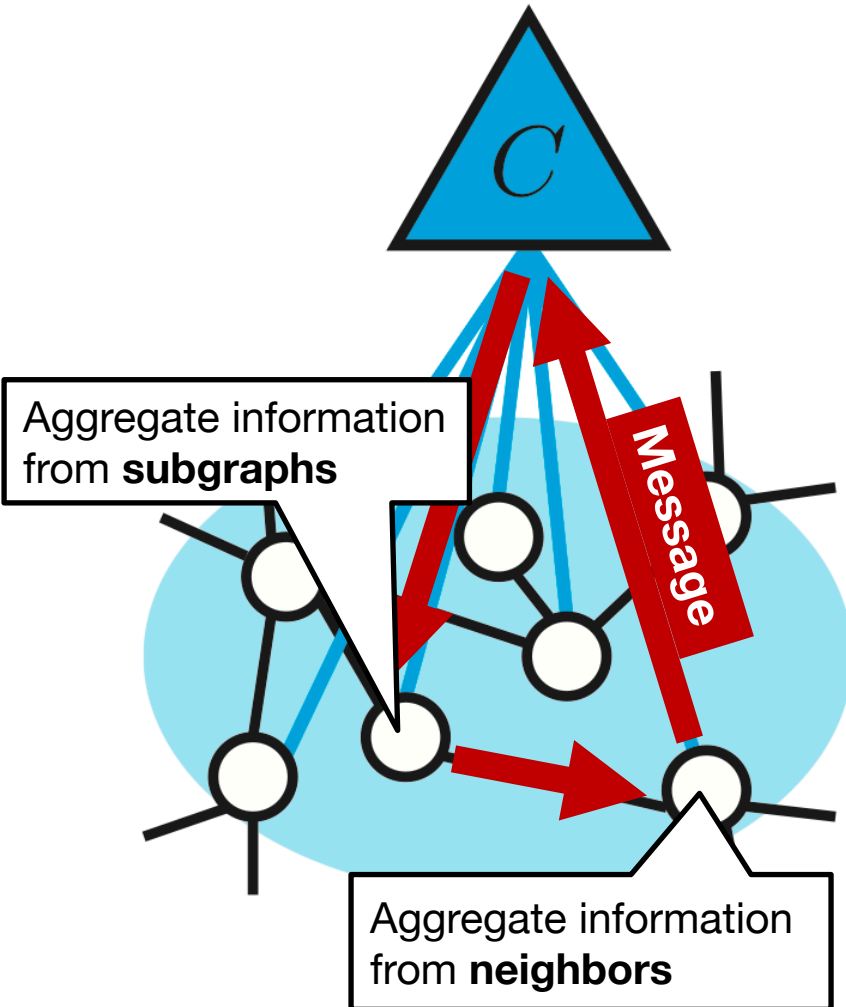
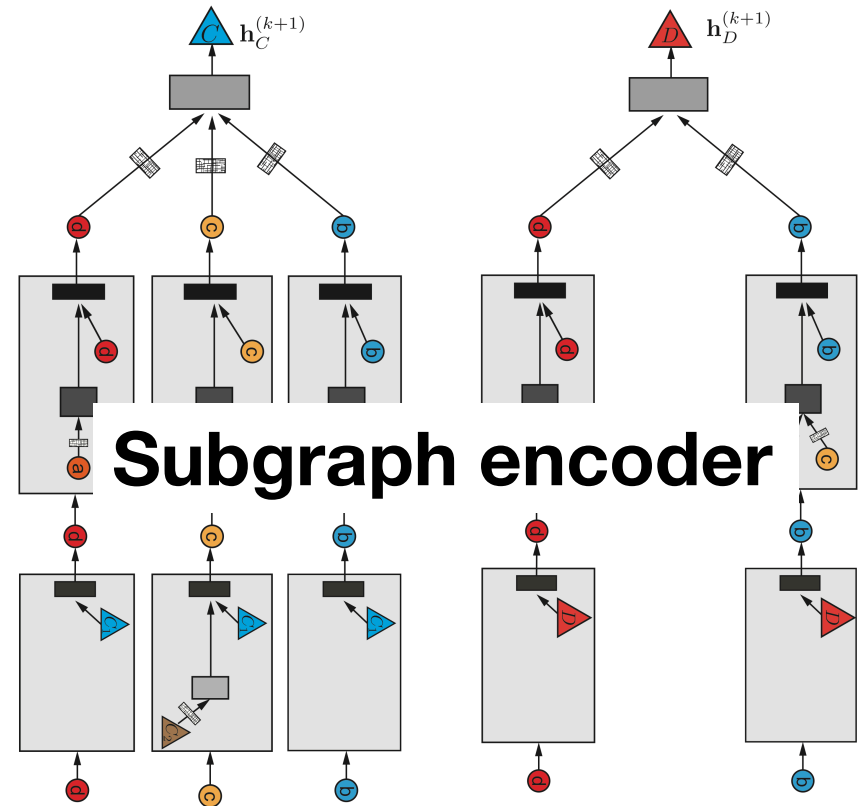
Task: Given drug C and disease D , predict if C treats D



SUGAR: Neural Message Passing

$$p(\triangle_C, \triangle_D)$$

Edge decoder



We need Drug Repurposing Dataset

- Protein-protein interaction network culled from 15 knowledge databases [Menche et al. *Science* 15]
 - 19K nodes, 350K edges
- Drug-protein and disease-protein links:
 - DrugBank, OMIM, DisGeNET, STITCH DB and others
 - 5K drugs, 20K diseases
 - 20K drug-protein links, 560K disease-protein links
- Drug medical indications:
 - DrugBank, MEDI-HPS, DailyMed, RepoDB and others
 - 6K drug-disease indications
- Side information on drugs, diseases, proteins, etc.:
Molecular pathways, disease symptoms, side effects

Predictive Performance



Task: Given a disease and a drug, predict if the drug could treat the disease

Approach

AUPRC AUROC

Our method (SUGAR)

0.851 0.888

Graphlets [Bioinformatics'13]

PREdicting Drug IndiCaTions [Mol. Sys. Biol.'11]

Bi-directional random walks [Bioinformatics'16]

Heterogeneous graph inference [Bioinformatics'14]

Drug-disease closeness [Nat. Commun.'17]

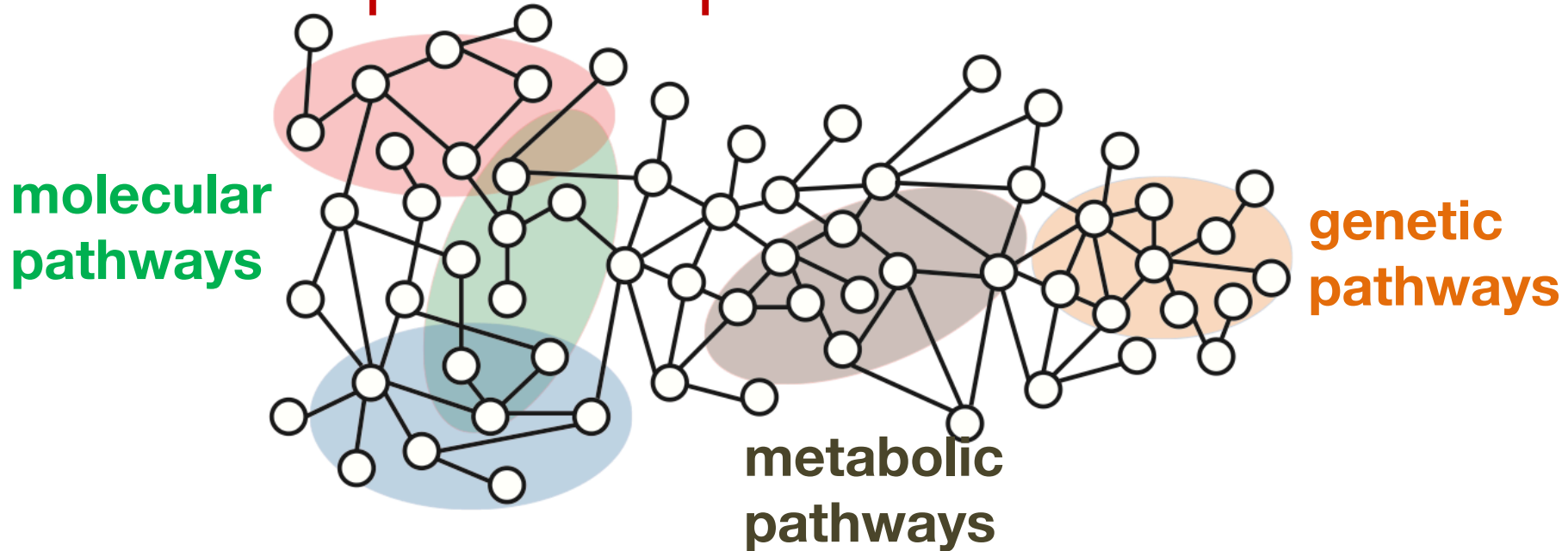
Drug-disease dispersion [Nat. Commun.'17]

Gene-based network overlap [Nat. Commun.'17]

**Up to 49%
improvement**

**Up to 172%
improvement**

Side Information further improves performance



Metabolic pathways	Molecular functions	Biological processes	Cellular components	AUPRC	AUROC
				0.851	0.888
✓				0.869	0.893
✓	✓			0.874	0.912
✓	✓	✓		0.893	0.912
✓	✓	✓	✓	0.901	0.928

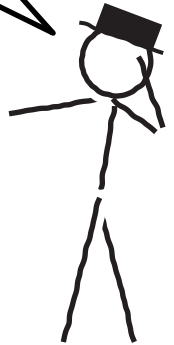
Drug Repurposing at Stanford



Stanford
MEDICINE

SPARK Translational Research Program
From Bench to Bedside

Task: Predict if an existing drug can be repurposed for a new disease



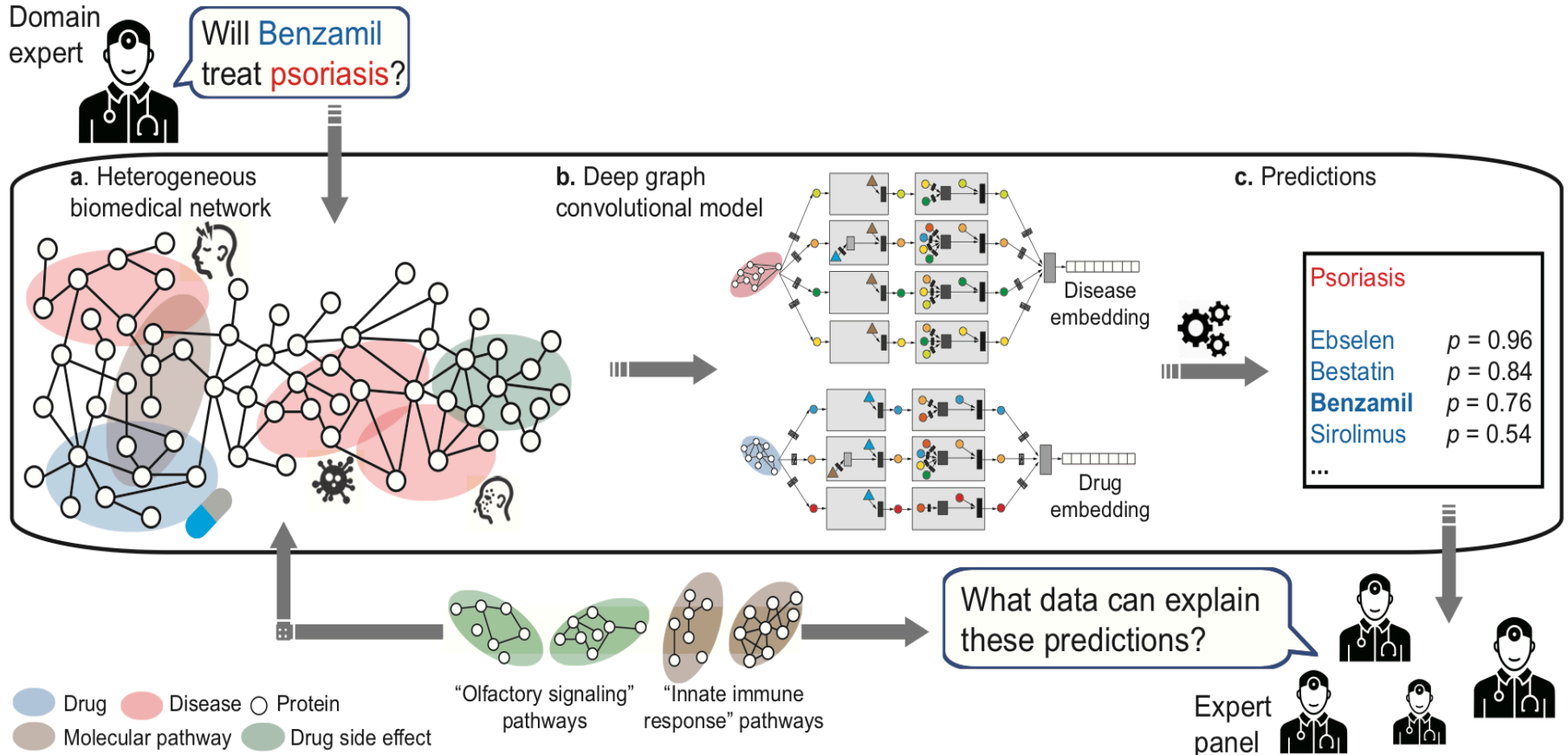
Drug

Disease

N-acetyl-cysteine	cystic fibrosis	Rank: 36/5000
Xamoterol	neurodegenerat	Rank: 10/5000
Plerixafor	cancer	Rank: 26/5000
Sodium selenite	cancer	Rank: 11/5000
Ebselen	C difficile	Rank: 16/5000
Itraconazole	cancer	Rank: 28/5000
Bestatin	lymphedema	Rank: 26/5000
Bestatin	pulmonary arterial hypertension	Rank: 46/5000
Ketaprofen	lymphedema	Rank: 114/5000
Sildenafil	lymphatic malformation	Rank: 9/5000
Tacrolimus	pulmonary arterial hypertension	
Benzamil	psoriasis	
Carvedilol	Chagas' disease	

Led to follow-up research on prostate cancer and schizophrenia at Stanford Medical School

Introducing Feedbacks for AI Loop



Stanford
MEDICINE

SPARK Translational Research Program
From Bench to Bedside

Outline of this Lecture

1) Deep Graph Neural Networks ✓

2) Polypharmacy & Drug Interactions ✓

3) Drug Repurposing ✓

4) New Directions and Opportunities

5) Practical Advice and Demos



New Directions and Opportunities

Material based on:

- Zitnik et al. 2019. Machine Learning for Integrating Data in Biology and Medicine: Principles, Practice, and Opportunities. *Information Fusion*.
- Camacho et al. 2018 Next-Generation Machine Learning for Biological Networks. *Cell*.

New Directions

1. Construct **contextual explanatory models** and introduce feedbacks for the AI loop
2. Design models to **train more with less data**
3. Create deep learning models for **rich interaction data** and **computations over graphs**

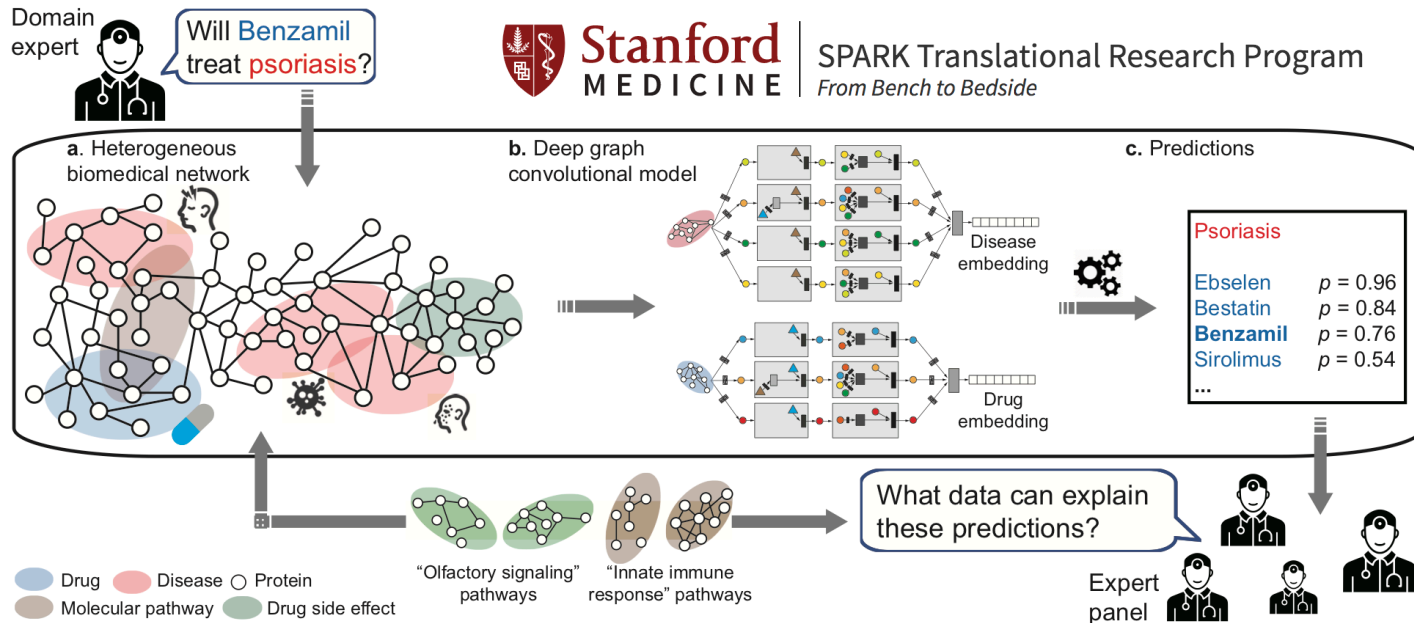
1st New Direction: Explanations

- Exciting phrase is not only

Networks

Networks

Initial results: Introduce feedback for the AI loop



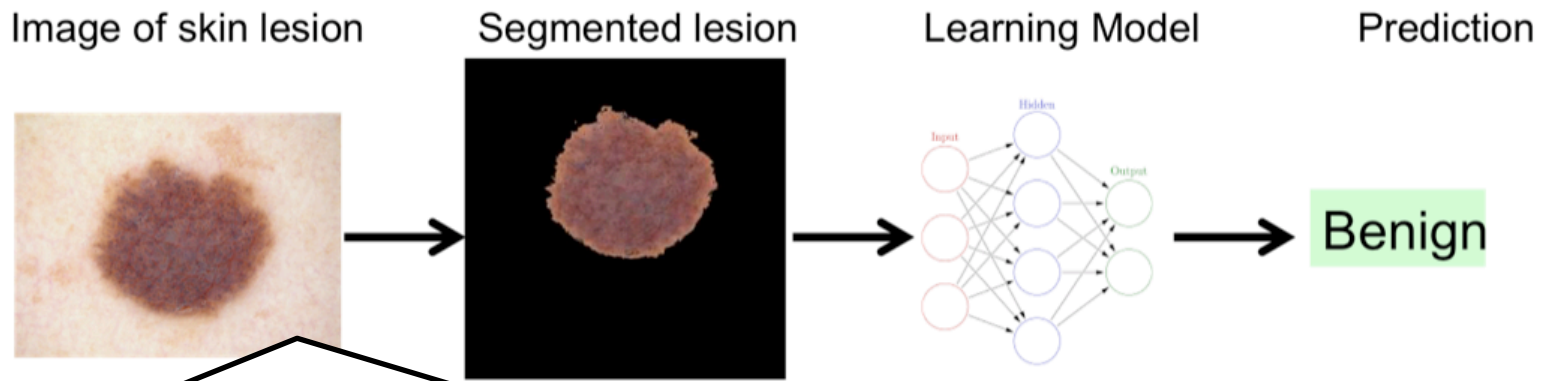
meaningful interpretations

Prediction

Prediction

2nd New Direction: Train with Less Data

- Algorithms to **train more with less network data**
- Learn about **never-before-seen systems**, generalize across **contexts**, e.g., patients, diseases, environments
- Natural case studies:
 - Single-cell genomics, Polygenic analyses, Health informatics



Current machine learning models:

- Only image at input
- Little or no knowledge about the disease or patient

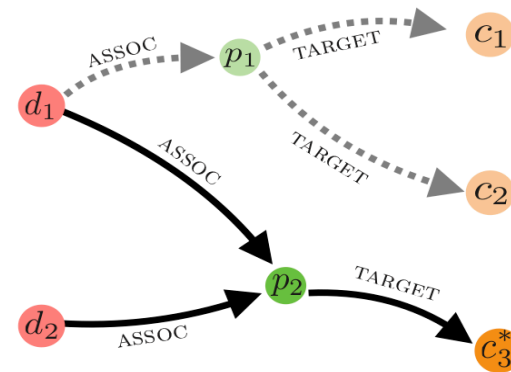
3rd New Direction: Rich Interactions

Initial results: Deep framework for logical [NeurIPS'18]

- queries on knowledge graphs

Query:

Predict drugs \mathcal{C} that might TARGET proteins, which are in turn ASSOCIATED with diseases d_1 and d_2



- Prediction beyond node classification and link prediction

Initial results: Biomedical network data and tools


🔗 Networks and re snap.stanford.edu/biodata

Name	Edges	Entities	Description
CC-Neuron	49,471,006	cell, cell	Similarity network between cells in embryonic mouse brain
ChCh-Miner	96,137	drug, drug	Interactions between FDA-approved drugs
ChChSe-Decagon	4,649,441	drug, drug, side-effect	Side effects of drug combinations
ChG-InterDecagon	131,034	drug, gene	Chemical-gene interaction network
ChG-Miner	15,424	drug, gene	Drug-target interaction network
ChG-TargetDecagon	18,690	drug, gene	Drug-target interaction network

snap.stanford.edu/mambo

Scales to terabytes of data, e.g., networks with 2.3 billion edges and over 2,000 nodes

Outline of this Lecture

- 1) Deep Graph Neural Networks ✓
- 2) Polypharmacy & Drug Interactions ✓
- 3) Drug Repurposing ✓
- 4) New Directions and Opportunities ✓
- 5) Practical Advice and Demos 

Practical Advice and Demos

Deep Learning for Network Biology

How to Start?

Lecture Resources

- **MAMBO:** Multimodal biomedical networks
 - Scales to networks with 2.3 billion edges and over 2,000 modes
 - snap.stanford.edu/mambo
- **Network data:**
 - snap.stanford.edu/projects.html:
 - [CRank](#), [Decagon](#), [MAMBO](#), [NE](#), [OhmNet](#), [Pathways](#), [Tree of Life](#), and many others
 - snap.stanford.edu/biodata
 - Algorithm benchmarking, method development
 - Easy to link entities across datasets

🔗 Networks and relationships

Name	Edges	Entities	Description
CC-Neuron	49,471,006	cell, cell	Similarity network between cells in embryonic mouse brain
ChCh-Miner	96,137	drug, drug	Interactions between FDA-approved drugs
ChChSe-Decagon	4,649,441	drug, drug, side-effect	Side effects of drug combinations
ChG-InterDecagon	131,034	drug, gene	Chemical-gene interaction network

Easy Deep Learning on Graphs

- Node2vec:
 - <https://github.com/aditya-grover/node2vec> (Python)
 - <https://github.com/snap-stanford/snap/tree/master/examples/node2vec> (C++)
- Graph Convolutional Networks (GCNs):
 - <https://github.com/kipf/gcn> (Tensorflow)
 - <https://github.com/kipf/pygcn> (PyTorch)
 - <https://github.com/kipf/keras-gcn> (Keras)
- GraphSAGE:
 - <https://github.com/williamleif/GraphSAGE> (Tensorflow)
 - <https://github.com/williamleif/graphsage-simple> (Pytorch)
- Metapath2vec and metapath2vec++ (Python):
 - <https://ericdongyx.github.io/metapath2vec/m2v.html>
- OhmNet (Python):
 - <https://github.com/marinkaz/ohmnet>
- Decagon (Tensorflow):
 - <https://github.com/marinkaz/decagon>
- GraphNets (Tensorflow):
 - https://github.com/deepmind/graph_nets
- Deep Graph Library (PyTorch):
 - https://github.com/deepmind/graph_nets

Recap

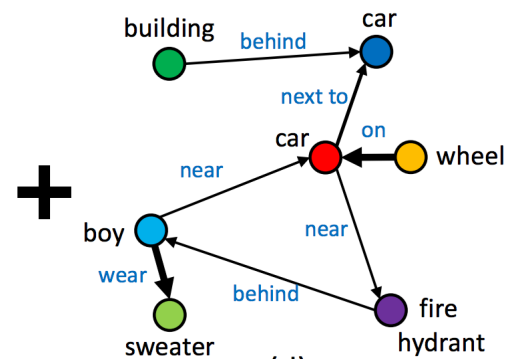
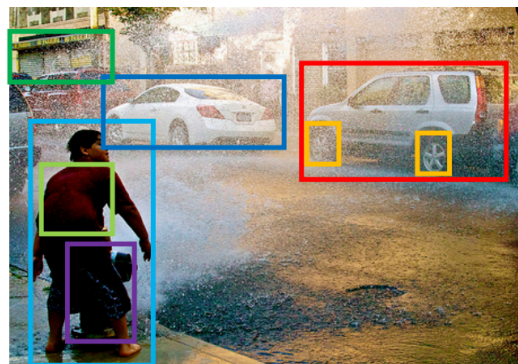
1. You have been using graphs in your projects:

- Keep on using them!
- No tedious feature engineering necessary anymore
- Combine node/edge attributes with extra information
- End-to-end training can achieve SotA performance

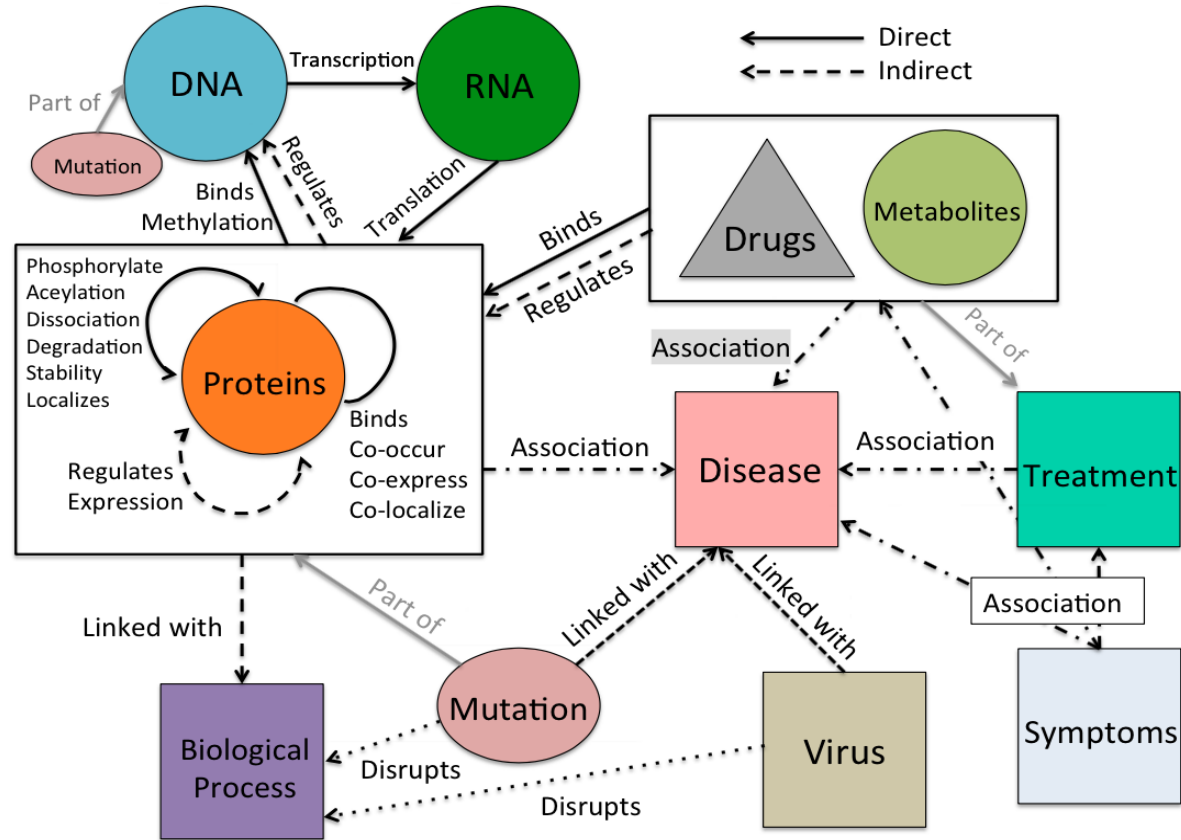
1. You haven't used graphs yet:



VS.



Network Biology and Medicine



- Large networks of interactions
- Opportunity to **integrate knowledge with diverse experimental readouts and perform discovery**

How can this technology be used for biomedical problems?

- **Node prediction:** E.g., Predicting tissue-specific protein functions
- **Pairs of nodes:** E.g., Predicting side-effects of drug combinations
- **Subgraph prediction:** E.g., Predicting which drug treats what disease
- **Graph prediction:** E.g., Predicting properties of molecular graphs

Demo: Diseases

Human Disease Network x Marinka

← → ↻ 🏠 ⓘ snap.stanford.edu/deepnetbio-ismb/ipynb/Human+Disease+Network.html 🔍 ☆ 📄 🔍 🗑️ 🔄 🌐 ⋮

Embedding the Human Disease Network

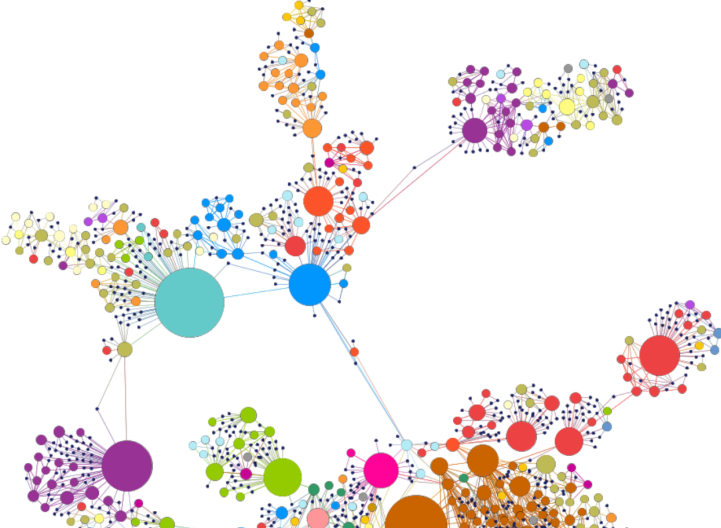
(This demo is a part of [Deep Learning for Network Biology](#) tutorial.)

Human disease network is a network, in which nodes represent diseases and two diseases are connected to each other if they share at least one gene in which mutations are associated with both diseases.

The network is described in Goh et al., [The Human Disease Network](#), PNAS 2007.

The figure below show the human disease network.

Although the layout of the network was generated independently of any knowledge of disease classes, the resulting network is naturally and visibly clustered according to major disease classes (e.g., bone, cancer, cardiovascular, skeletal, or metabolic diseases; each disease class is represented by a different color). The size of a node is proportional to the number of genes participating in the corresponding disease.



Demo: Protein Interactions

Graph Convolutional Prediction of Protein Interactions in Yeast

(This demo is a part of [Deep Learning for Network Biology](#) tutorial.)

In this example, we demonstrate the utility of deep learning methods for an important prediction problem on biological graphs. In particular, we consider the problem of predicting [protein-protein interactions](#) (PPIs).

Protein-protein interactions (PPIs) are essential to almost every process in a cell. Understanding PPIs is crucial for understanding cell physiology in normal and disease states. Furthermore, knowledge of PPIs can be used:

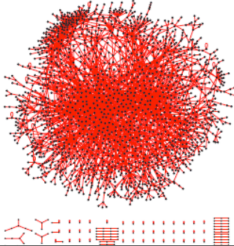
- for drug development, since drugs can affect PPIs,
- to assign roles (i.e., protein functions) to uncharacterized proteins,
- to characterize the relationships between proteins that form multi-molecular complexes, such as the proteasome.

We represent the totality of PPIs that happen in a cell, an organism or a specific biological context with a [protein-protein interaction network](#). These networks are mathematical representations of all physical contacts between proteins in the cell.

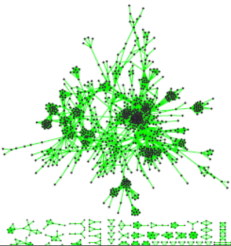
The development of large-scale PPI screening techniques, especially [high-throughput affinity purification combined with mass-spectrometry](#) and the [yeast two-hybrid assay](#), has caused an explosion in the amount of PPI data and the construction of ever more complex and complete interaction networks. For example, the figure below is a graphical representation of three different types of protein-protein interaction networks in yeast *S. cerevisiae*. The structure of the binary interaction network is obviously different from the structure of the co-complex interaction network. The network structure of the literature-curated dataset resembles that of the co-complex dataset, even though the literature-curated datasets are reported to contain mostly binary interactions.

However, current knowledge of protein-protein interaction networks is both [incomplete and noisy](#), as PPI screening techniques are limited in how many true interactions they can detect. Furthermore, PPI screening techniques often have high false positive and negative rates. These limitations present a great opportunity for computational methods to predict protein-protein interactions.

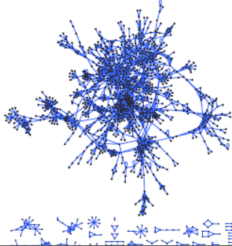
Binary
(Y2H-union)



Co-complex
(Combined-AP/MS)



Literature
(LC-multiple)



General Tips

- 1) Network preprocessing is important:
 - Renormalization tricks, variance-scaled initialization, data whitening
- 2) Use the ADAM optimizer, decay the learning rate
- 3) ReLU often works really well
- 4) No activation function at output layer:
 - Easy mistake if layers are built with a shared function
- 5) Include bias term in every layer
- 6) Graph convolution layer of size 64 or 128 is plenty
- 7) Large graphs that cannot fit on one GPU card:
 - Sampling and batching across samples for maximum parallelism

General Tips

<https://d2l.ai>

The screenshot shows a web browser displaying the 'Dive into Deep Learning' website. The page title is 'Mathematics + Figures + Code'. The main text reads: 'We offer an interactive learning experience with mathematics, figures, code, text, and discussions, where concepts and techniques are illustrated and implemented with experiments on real data sets.'

On the left, a navigation menu lists the following sections:

- Preface
- Installation
- 1. Introduction
- 2. The Preliminaries: A Crashcourse
- 3. Linear Neural Networks
- 4. Multilayer Perceptrons
- 5. Deep Learning Computation
- 6. Convolutional Neural Networks
- 7. Modern Convolutional Networks
- 8. Recurrent Neural Networks
- 9. Attention Mechanism
- 10. Optimization Algorithms
- 11. Computational Performance

The main content area features three panels:

- Text Panel:** 'dden units and, for a given time step t , the mini-er of inputs: d) and the hidden state of the last $\mathbb{R}^{n \times h}$, forget gate $F_t \in \mathbb{R}^{n \times h}$, and output gate
- Equation Panel:**
$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i),$$
$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f),$$
$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o),$$
and $W_{hi}, W_{hf}, W_{ho} \in \mathbb{R}^{h \times h}$ are weight param
- Diagram Panel:** A diagram of an LSTM cell. It shows the flow of information from the previous hidden state H_{t-1} and input X_t through various gates (Forget gate F_t , Input gate I_t , Candidate memory \tilde{C}_t , Output gate O_t) to produce the current hidden state H_t and cell state C_t . The diagram includes a legend for 'FC layer with activation function' (sigma symbol) and 'Element-wise Operator' (circle with x symbol).
- Code Panel:** A Python code snippet for an LSTM function:

```
def lstm(inputs, state, params):
    [W_xi, W_hi, b_i, W_xf, W_hf, b_f, W_xo, W_ho, b_o,
     W_hq, b_q] = params
    (H, C) = state
    outputs = []
    for X in inputs:
        I = nd.sigmoid(nd.dot(X, W_xi) + nd.dot(H, W_hi) + b_i)
        F = nd.sigmoid(nd.dot(X, W_xf) + nd.dot(H, W_hf) + b_f)
        O = nd.sigmoid(nd.dot(X, W_xo) + nd.dot(H, W_ho) + b_o)
        C_tilda = nd.tanh(nd.dot(X, W_xc) + nd.dot(H, W_hc) + b_c)
        C = F * C + I * C_tilda
        H = O * C.tanh()
        Y = nd.dot(H, W_hq) + b_q
        outputs.append(Y)
    return outputs, (H, C)
```

Debugging Deep Networks

- Debug?!:
 - Loss/accuracy not converging during training
- Important for model development:
 - **Overfit on training data:**
 - Accuracy should be essentially 100% or error close to 0
 - If neural network cannot overfit a single data point, something is wrong
 - **Scrutinize your loss function!**
 - **Scrutinize your visualizations!**

Outline of this Lecture

- 1) Deep Graph Neural Networks ✓
- 2) Polypharmacy & Drug Interactions ✓
- 3) Drug Repurposing ✓
- 4) New Directions and Opportunities ✓
- 5) Practical Advice and Demos ✓

Two Lectures

Part 1: May 15, 2019, 2:30 pm - 4:00 pm

- Methodology: Shallow network embeddings:
 - Map nodes to low-dimensional features
- Resources: Data, tools, codebases
- Applications: PPIs, Disease pathways, Tissues



Part 2: May 16, 2019, 9:00 am – 10:30 am

- Methodology: Deep network embeddings:
 - Graph neural networks for rich biomedical graphs
- Resources: Data, practical advice and demos
- Applications: Polypharmacy, Drug repurposing

