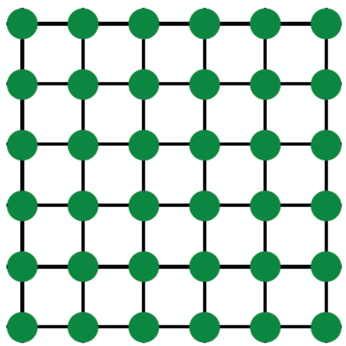
Advancements in Graph Neural Networks

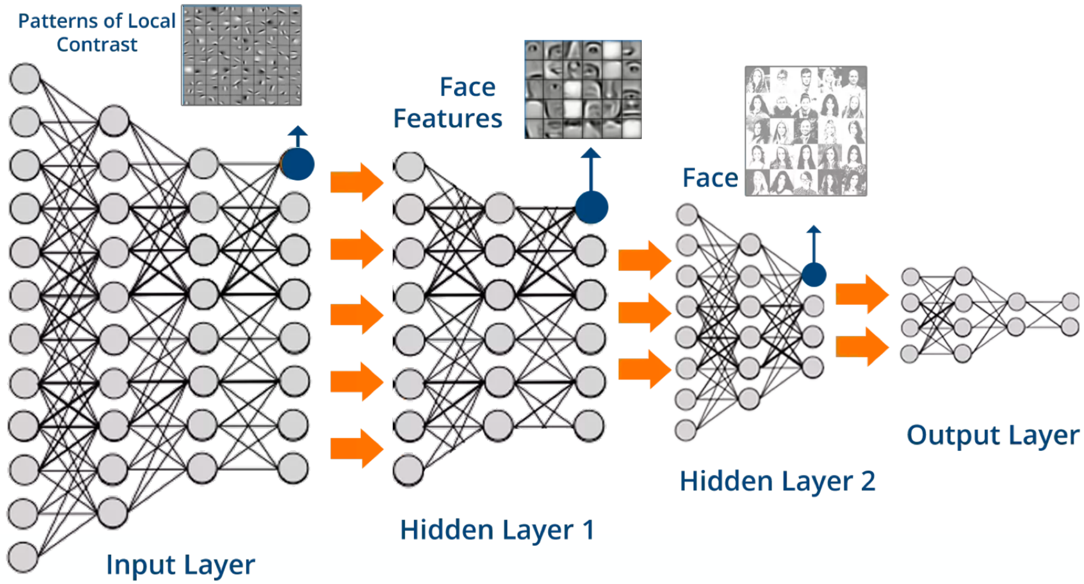
Jure Leskovec

Includes joint work with H. Ren, W. Hamilton, R. Ying, J. You, M. Zitnik, W. Hu, K. Xu, S. Jegelka

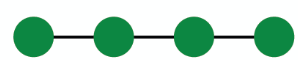
Modern ML Toolbox

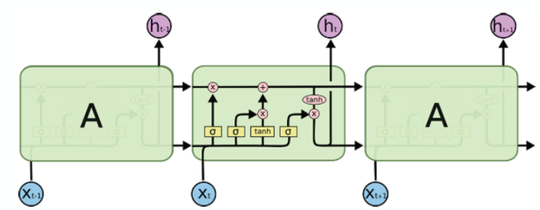


Images



Text/Speech





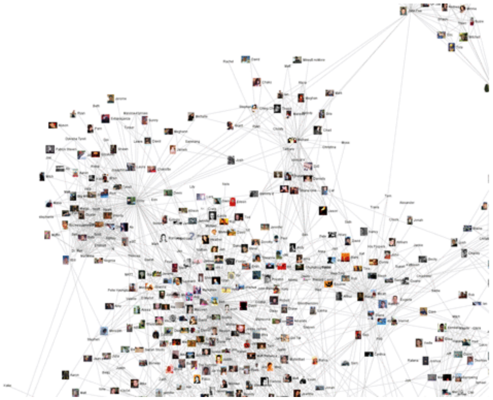
Modern deep learning toolbox is designed for simple sequences & grids

But not everything can be represented as a sequence or a grid

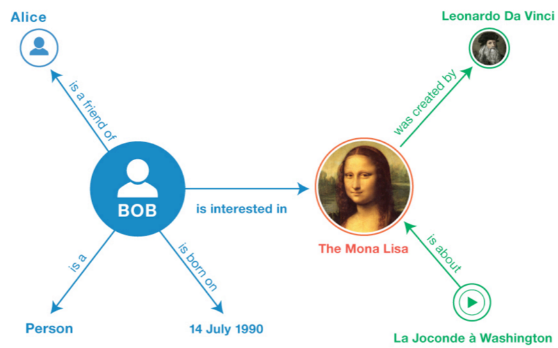
How can we develop neural networks that are much more broadly applicable?

New frontiers beyond classic neural networks that learn on images and sequences

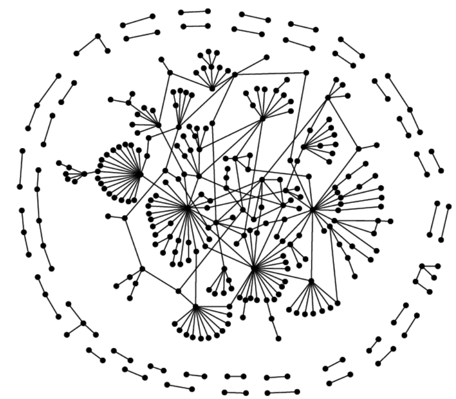
Networks of Interactions



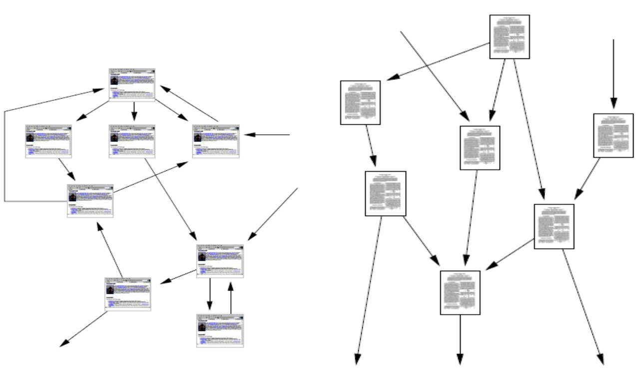
Social networks



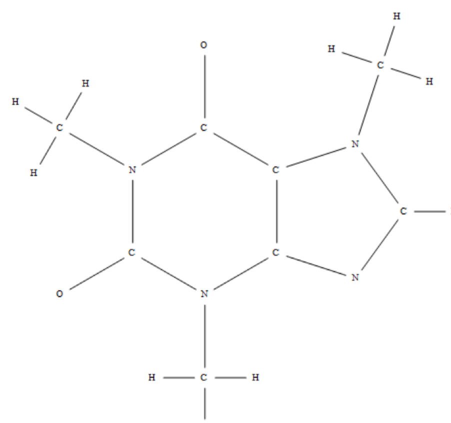
Knowledge graphs



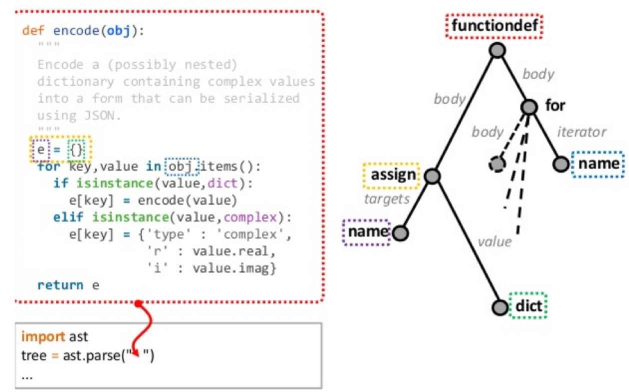
Biological networks



Complex Systems



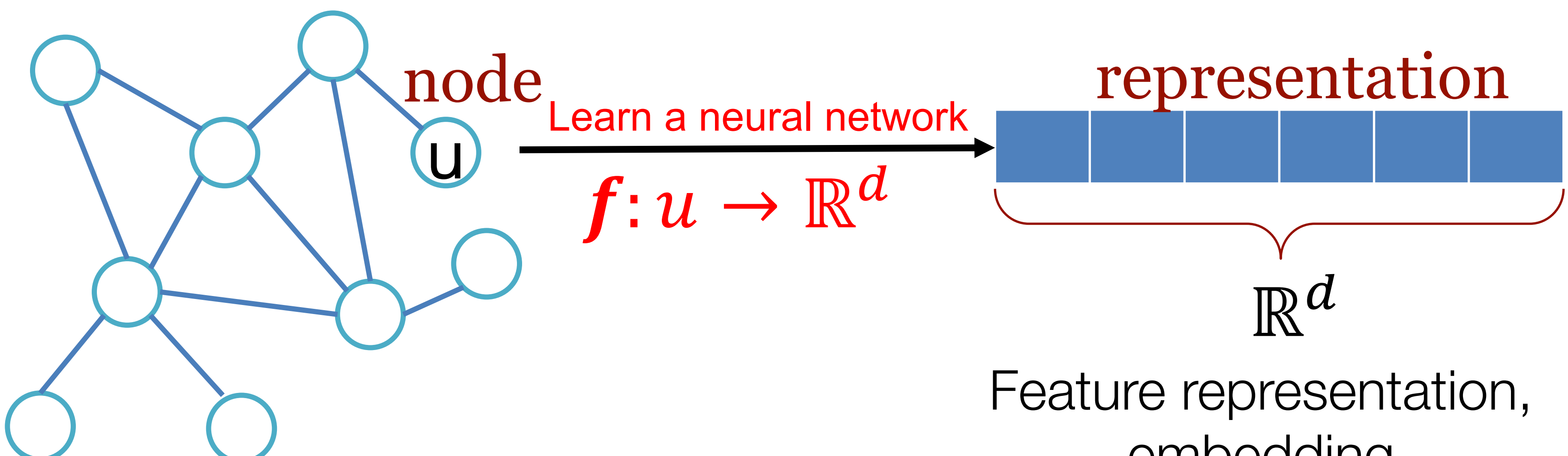
Molecules



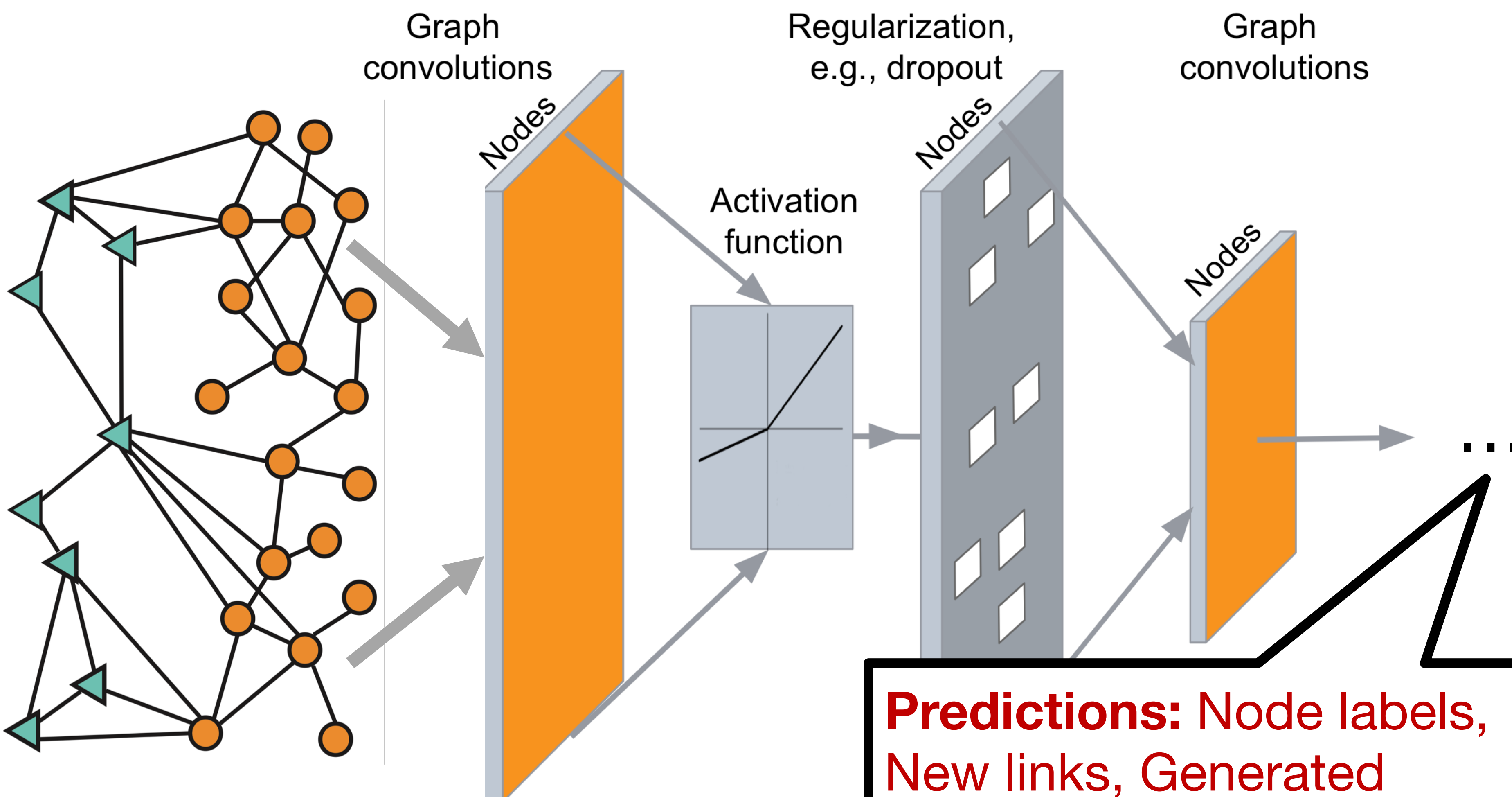
Code

Goal: Representation Learning

Map nodes to d-dimensional **embeddings** such that similar nodes in the network are embedded close together



Deep Learning in Graphs



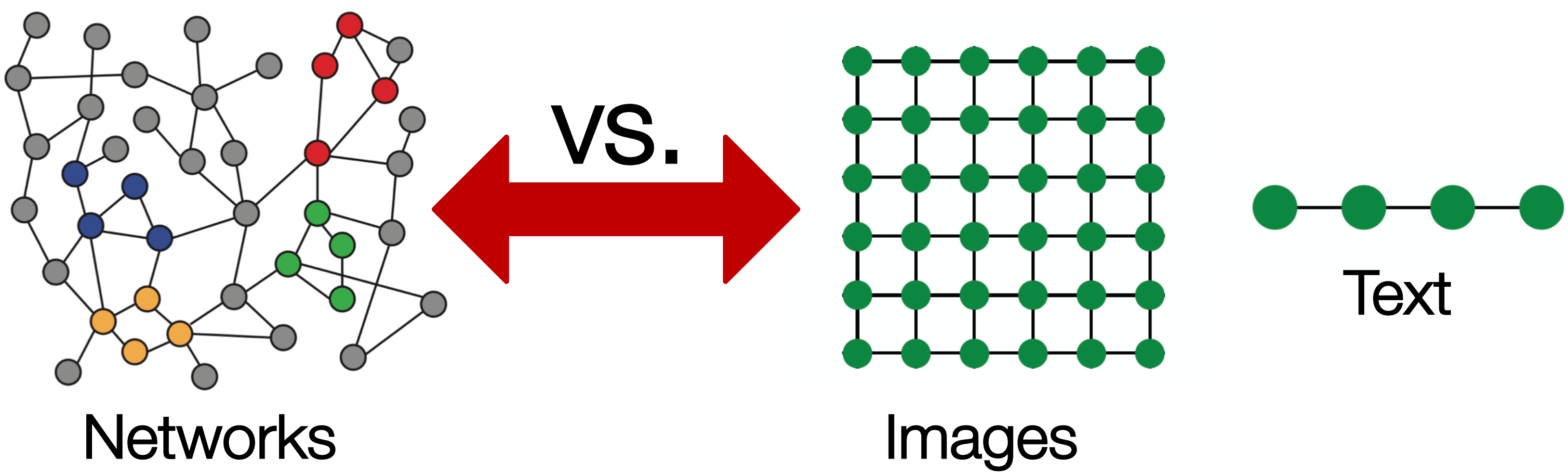
Predictions: Node labels, New links, Generated graphs and subgraphs

Input: Network

Why is it Hard?

Networks are complex!

- Arbitrary size and complex topological structure (i.e., no spatial locality like grids)



- No fixed node ordering or reference point
- Often dynamic and have multimodal features

GraphSAGE: Graph Neural Networks

[Inductive Representation Learning on Large Graphs](#).
W. Hamilton, R. Ying, J. Leskovec. Neural Information Processing Systems (NIPS), 2017.
[Representation Learning on Graphs: Methods and Applications](#).
W. Hamilton, R. Ying, J. Leskovec. IEEE Data Engineering Bulletin, 2017.

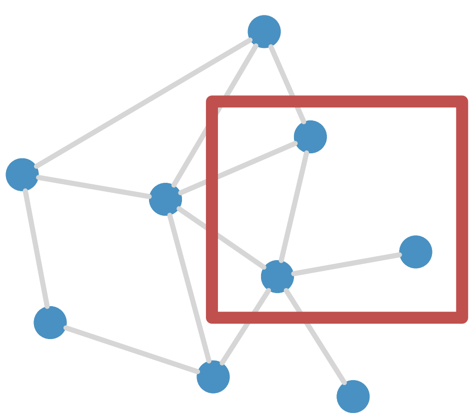
http://snap.stanford.edu/graphsage

Idea: Convolutional Networks

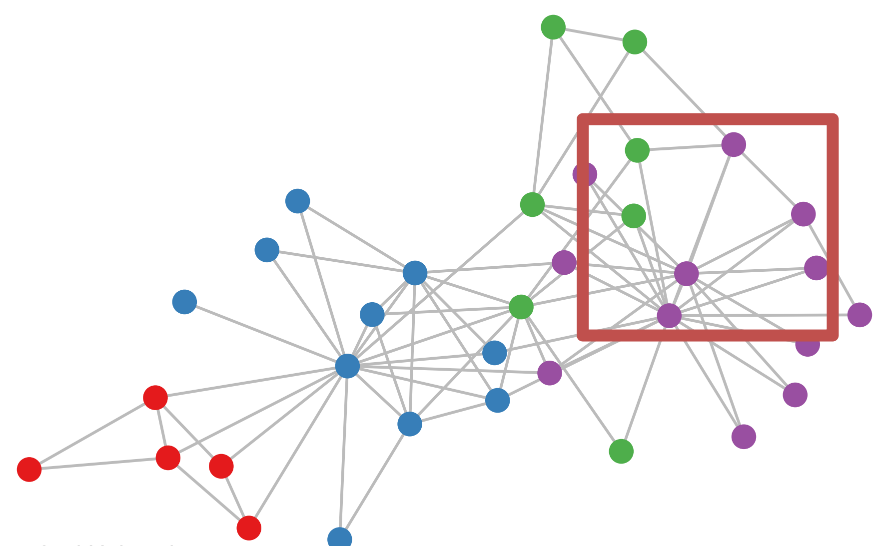
Goal is to generalize convolutions beyond simple lattices

Leverage node features (text, images)

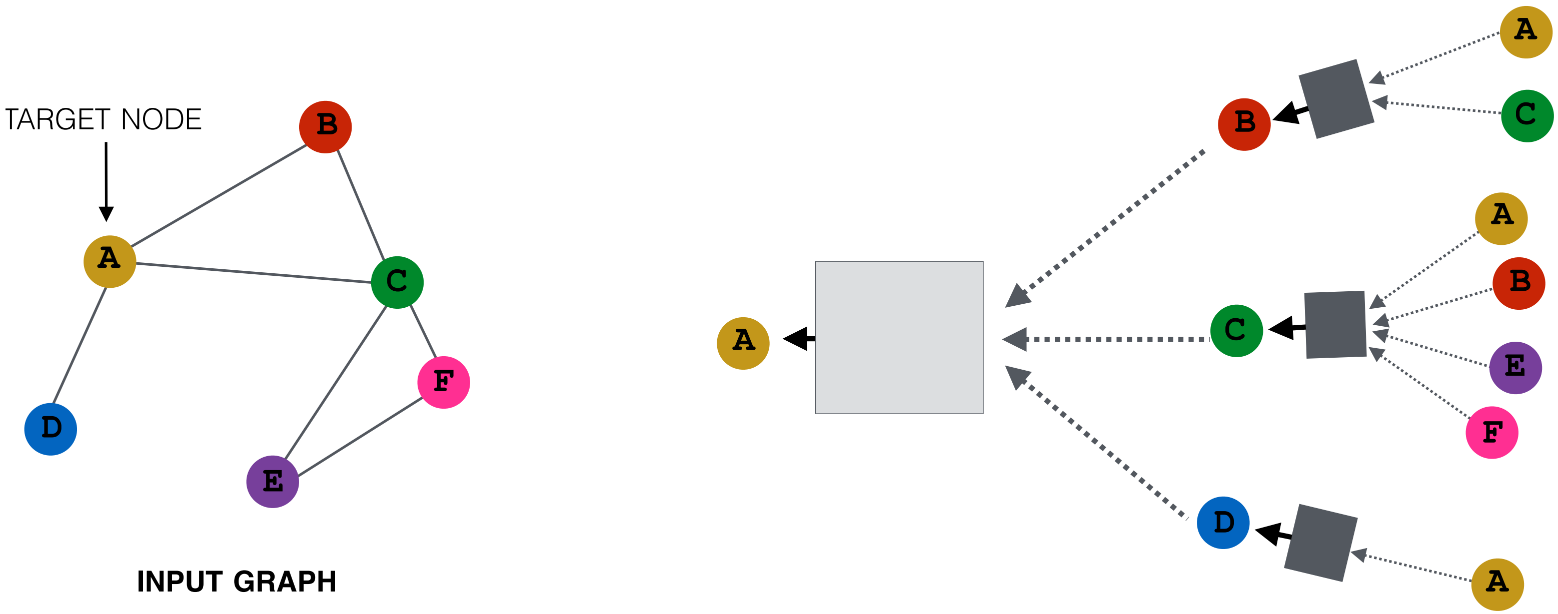
But real-world graphs look like this:



or this:



Graph Neural Networks

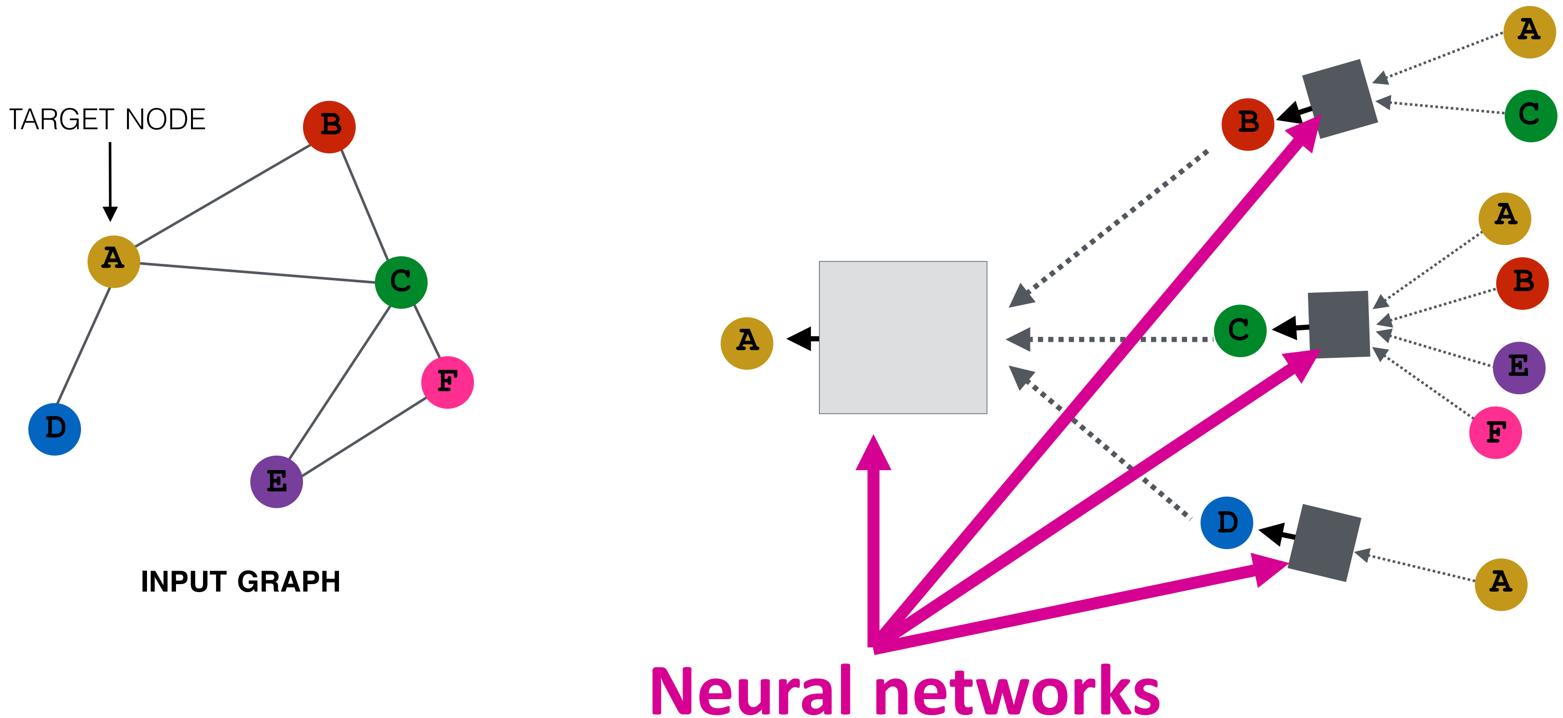


Each node defines a computation graph

- Each edge in this graph is a transformation/aggregation function

Scarselli et al. 2005. [The Graph Neural Network Model](). *IEEE Transactions on Neural Networks*.

Graph Neural Networks

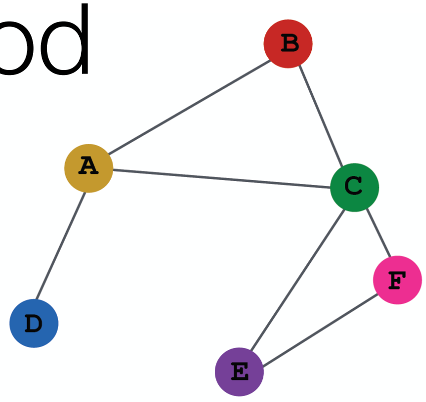


Intuition: Nodes aggregate information from their neighbors using neural networks

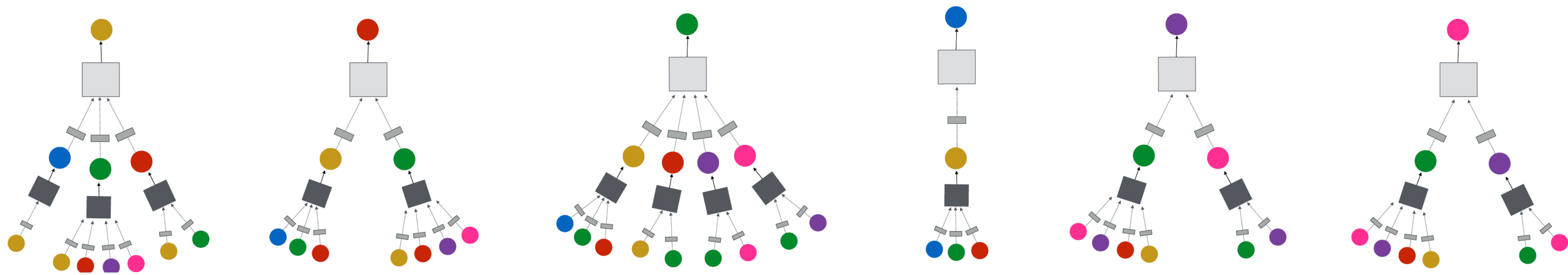
[Inductive Representation Learning on Large Graphs](). W. Hamilton, R. Ying, J. Leskovec. NIPS, 2017.

Idea: Aggregate Neighbors

Intuition: Network neighborhood defines a computation graph



Every node defines a computation graph based on its neighborhood!



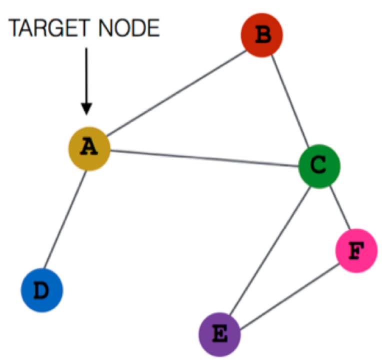
Can be viewed as learning a generic linear combination of graph low-pass and high-pass operators

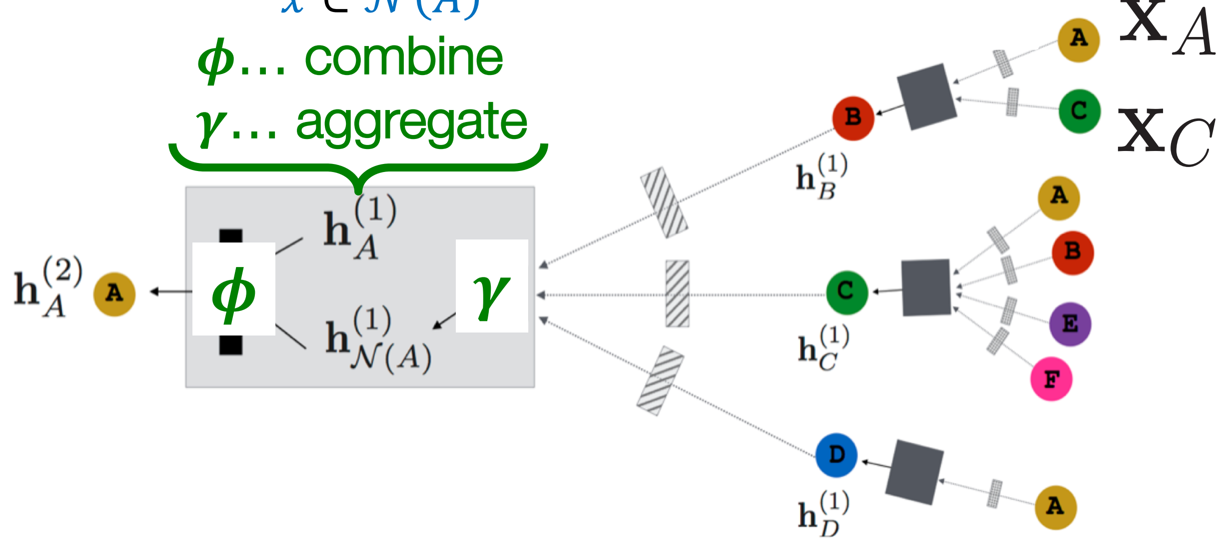
[Bronstein et al., 2017]

[NIPS '17]

Our Approach: GraphSAGE

$$\phi\Big(\gamma_{x \in \mathcal{N}(A)}\big(h_x\big)\Big)$$





Update for node $\boldsymbol{A}$:

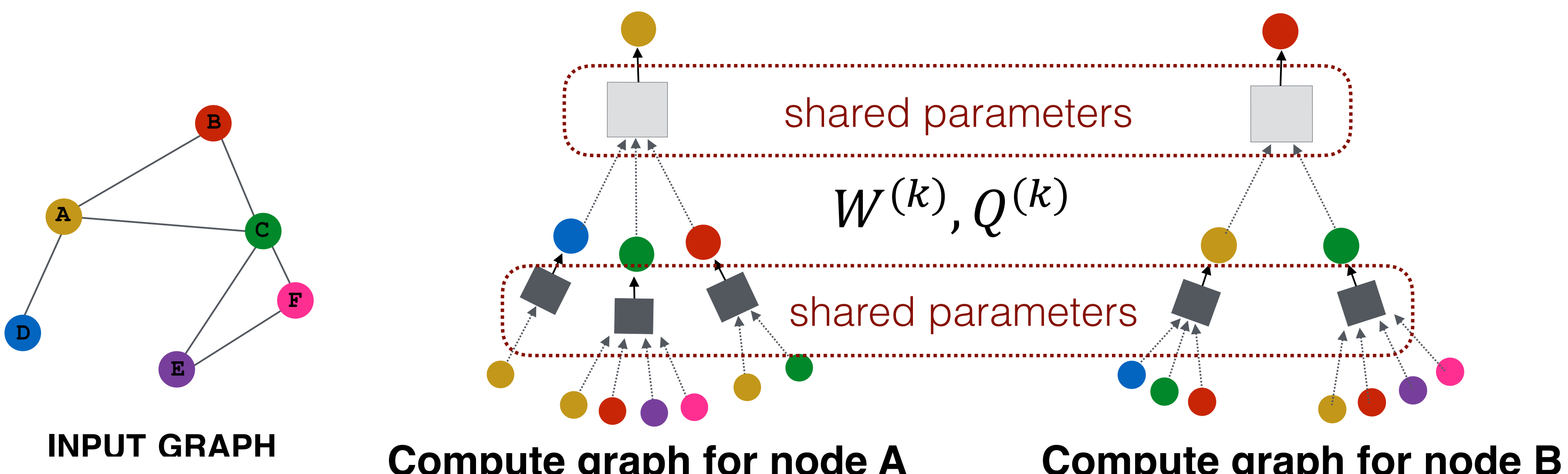
$$\underbrace{h_A^{(k+1)}}_{k+1^{st}\text{ level embedding of node }A} = \sigma\Bigg(\underbrace{W^{(k)} h_A^{(k)}}_{\text{Transform }A\text{'s own embedding from level }k},\ \gamma_{x \in \mathcal{N}(A)}\Big(\underbrace{\sigma\big(Q^{(k)} h_x^{(k)}\big)}_{\text{Transform and aggregate embeddings of neighbors }n}\Big)\Bigg)$$

- $h_A^{(0)}$ = attributes X_A of node A, $\sigma(\cdot)$ is a sigmoid activation function

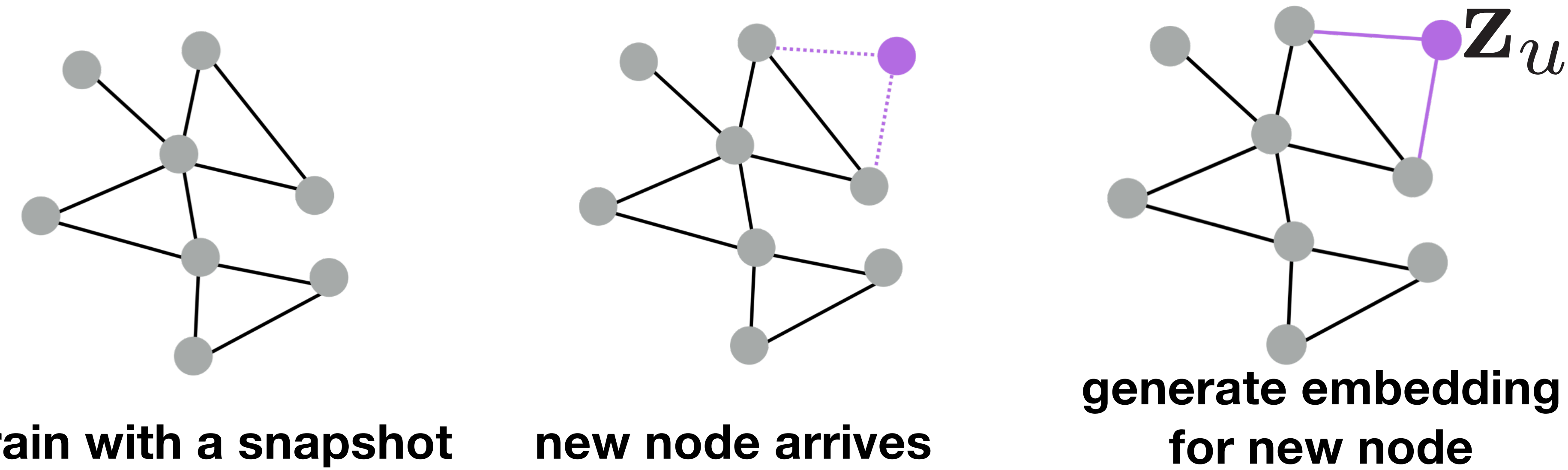
[NIPS '17]

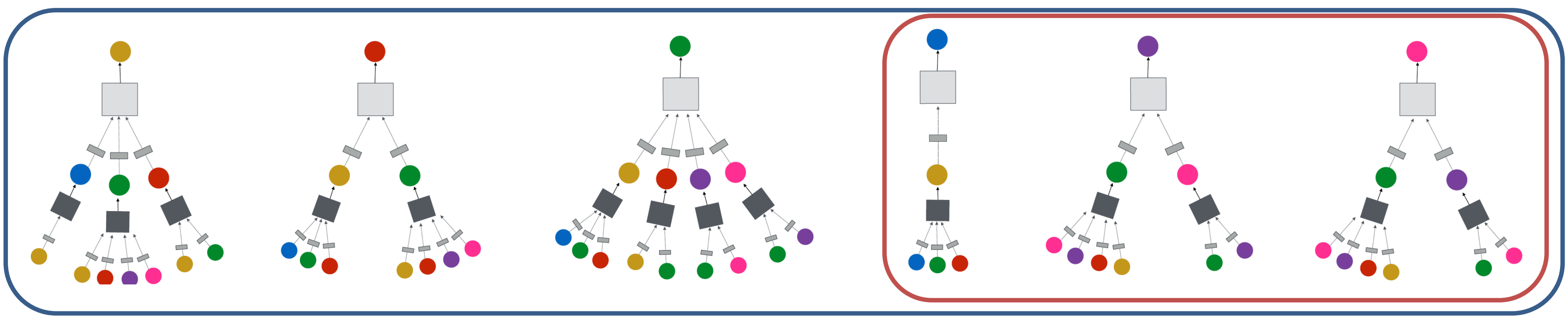
GraphSAGE: Training

- Aggregation parameters are shared for all nodes
- Number of model parameters is independent of |V|
- Can use different loss functions:
  - Classification/Regression: $\mathcal{L}(h_A) = ||y_A - f(h_A)||^2$
  - Pairwise Loss: $\mathcal{L}(h_A, h_B) = \max(0, 1 - dist(h_A, h_B))$



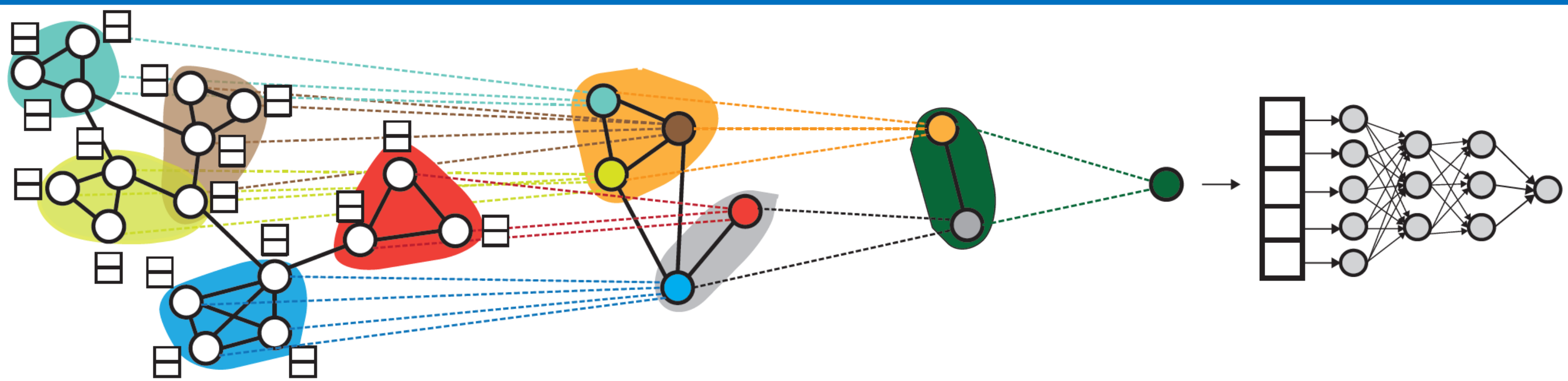
Inductive Capability





Even for nodes we never trained on!

DIFFPOOL: Pooling for GNNs



Don't just embed individual nodes. Embed the entire graph.

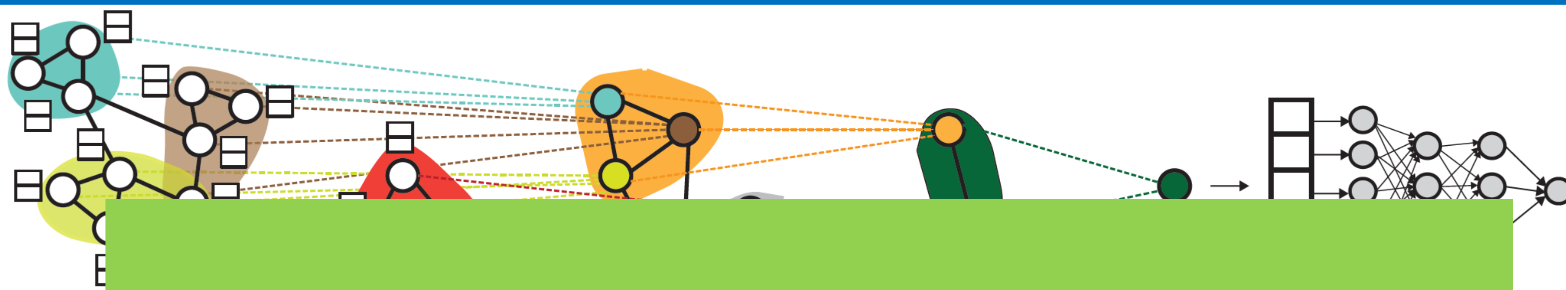
Problem: Learn how to hierarchical pool the nodes to embed the entire graph

Our solution: DIFFPOOL

- Learns hierarchical pooling strategy
- Sets of nodes are pooled hierarchically
- Soft assignment of nodes to next-level nodes

Hierarchical Graph Representation Learning with Differentiable Pooling. R. Ying, et al. NeurIPS 2018.

DiffPool: Pooling for GNNs



Don the
enti

How expressive are
Graph Neural Networks?

Prob e
nod

Our

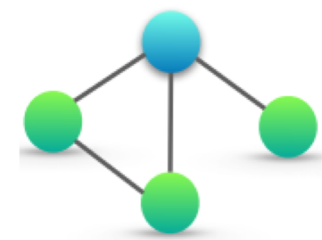
- Learns hierarchical pooling strategy
- Sets of nodes are pooled hierarchically
- Soft assignment of nodes to next-level nodes

Hierarchical Graph Representation Learning with Differentiable Pooling. R. Ying, et al. NeurIPS 2018.

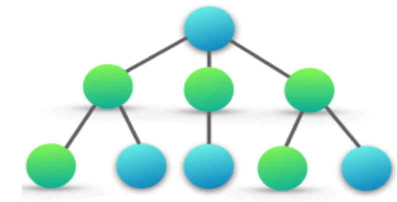
How expressive are GNNs?

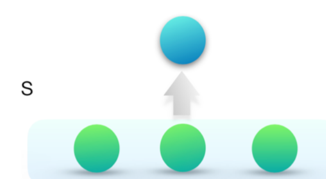
Theoretical framework: Characterize GNN's discriminative power:

- Characterize upper bound of the discriminative power of GNNs
- Propose a maximally powerful GNN
- Characterize discriminative power of popular GNNs



GNN tree:

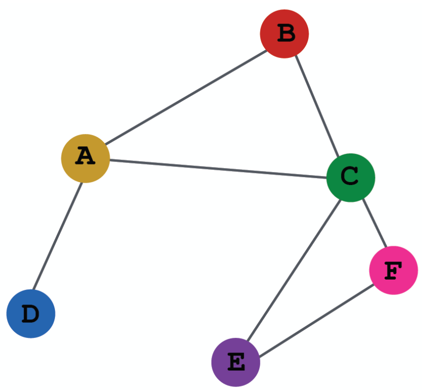




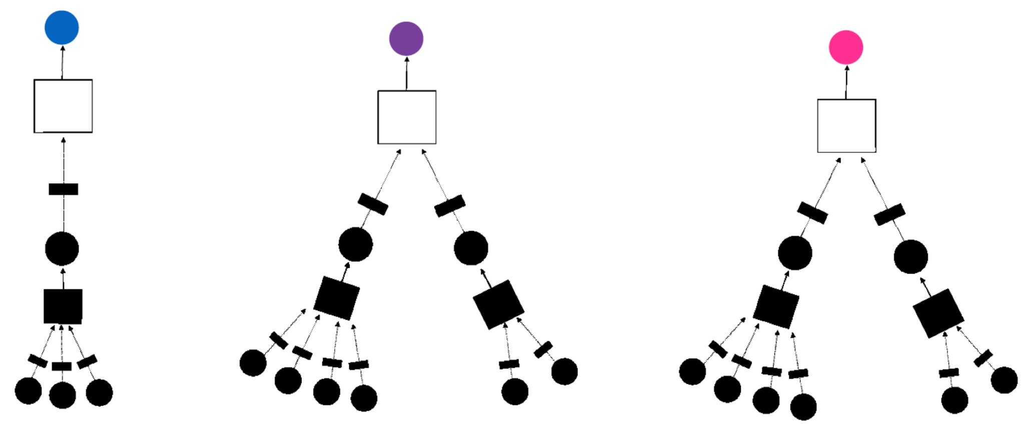
[How Powerful are Graph Neural Networks?](#) K. Xu, et al. ICLR 2019.

Key Insight: Rooted Subtrees

Graph:

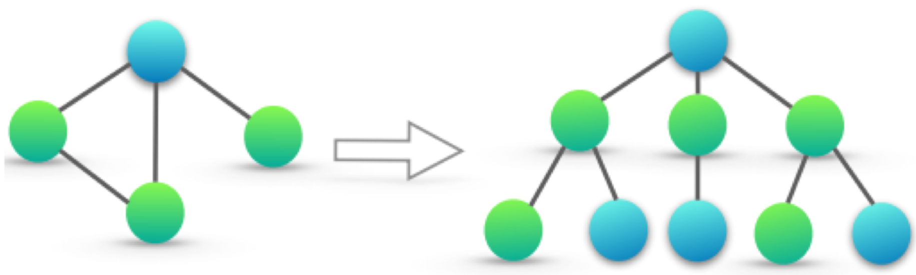


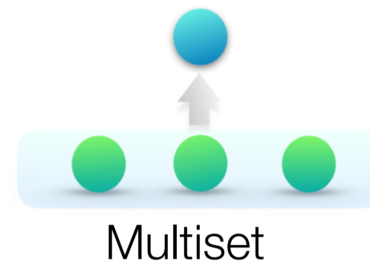
GNN distinguishes:



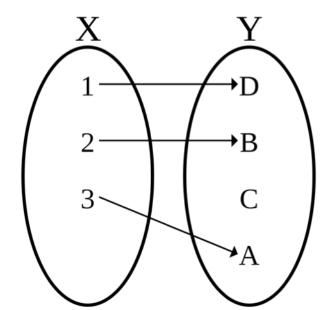
The most powerful GNN is able to distinguish rooted subtrees of different structure

Discriminative Power of GNNs





Idea: If GNN functions are <ins>injective</ins>, GNN can capture/distinguish the rooted subtree structures



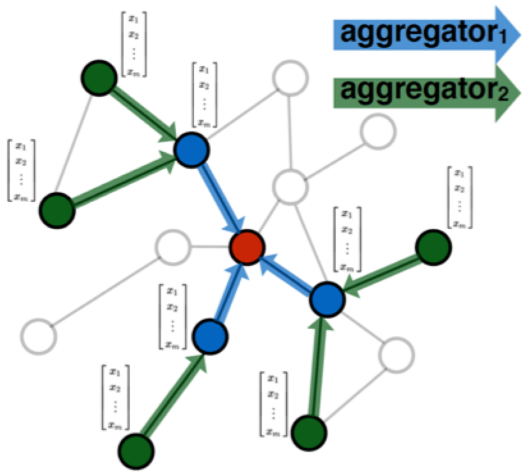
Theorem: The most discriminative GNN uses <ins>injective multiset function</ins> for neighbor aggregation

If the aggregation function is <ins>injective</ins>, GNN can fully capture/distinguish the rooted subtree structures

Three Consequences of GNNs

1) The GNN does two things:

- Learns how to "borrow" feature information from nearby nodes to enrich the target node

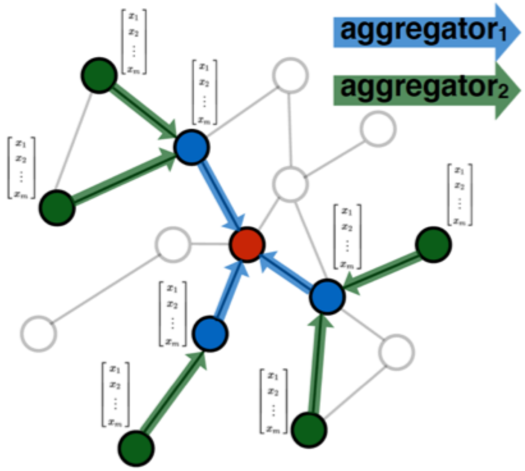


- Each node can have a different computation graph and the network is also able to capture/learn its structure

Three Consequences of GNNs

2) Computation graphs can be chosen:

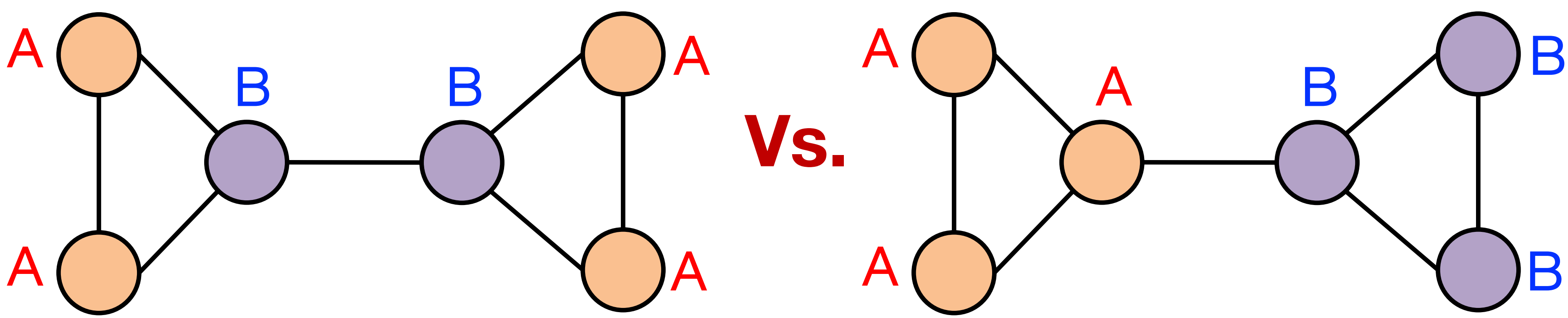
- Aggregation does not need to happen across all neighbors
- Neighbors can be strategically chosen/sampled
- Leads to big gains in practice



Three Consequences of GNNs

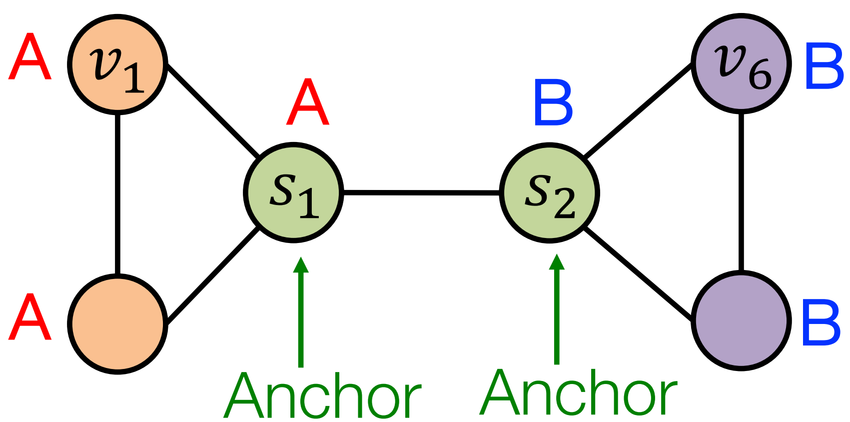
3) We understand GNN failure cases:

- GNNs fail to distinguish isomorphic nodes
- Structure-aware **Vs.** Position-aware



PGNN: Position Aware GNNs

- Key idea: Anchors
  - Characterize node's position relative to a set of randomly selected anchor nodes and sets of nodes



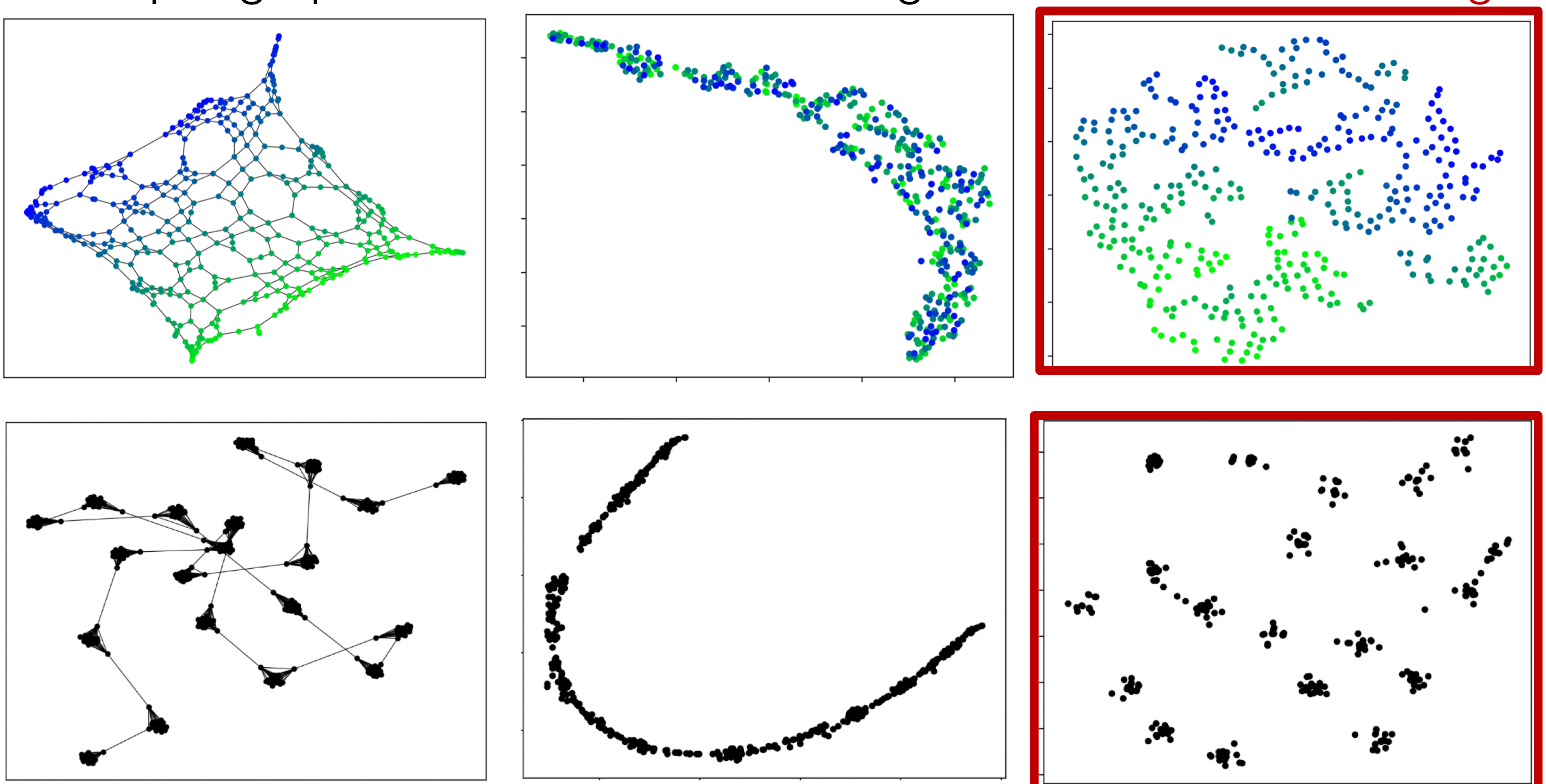
Distance to Anchor:

	s_1	s_2
v_1	1	2
v_6	2	1

Position-aware Graph Neural Networks. J. You, R. Ying, J. Leskovec. ICML, 2019.

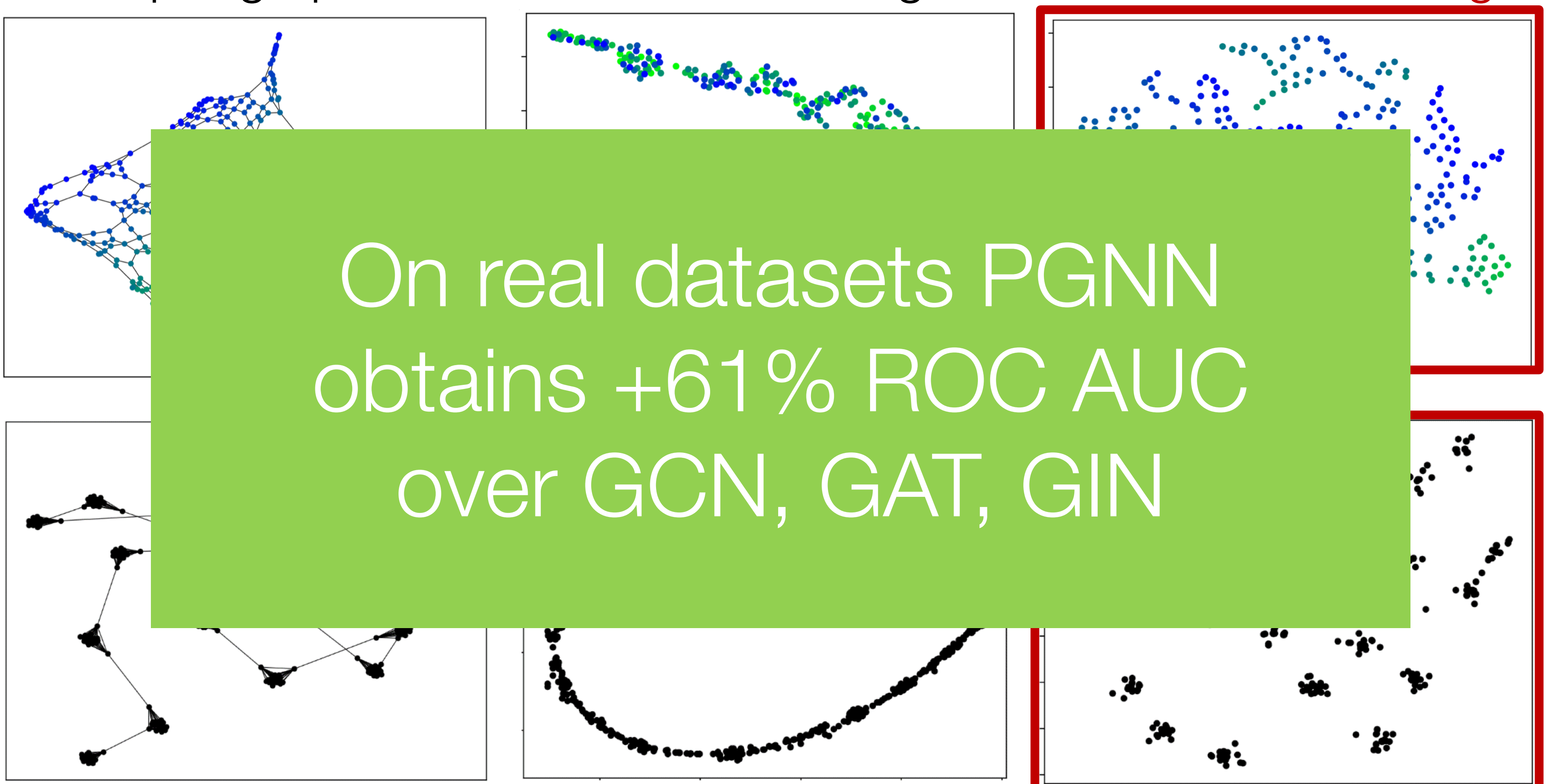
PGNN: Visualizing Embeddings

Input graph | GNN embedding | P-GNN embedding



PGNN: Visualizing Embeddings

Input graph | GNN embedding | P-GNN embedding

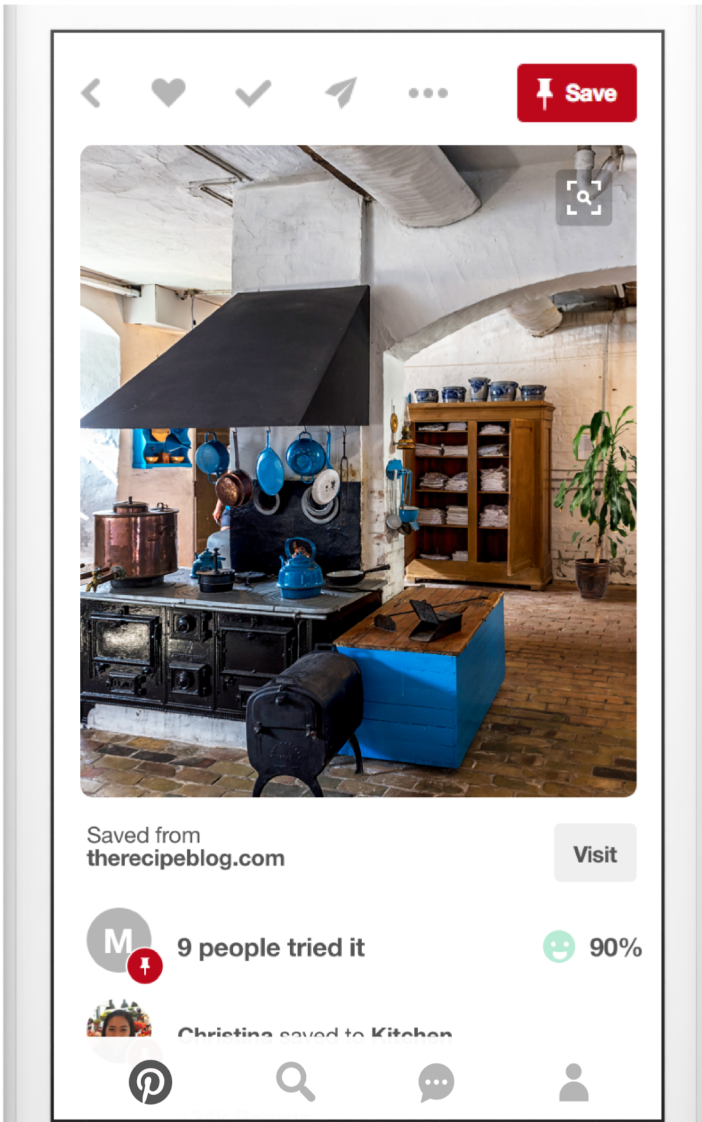


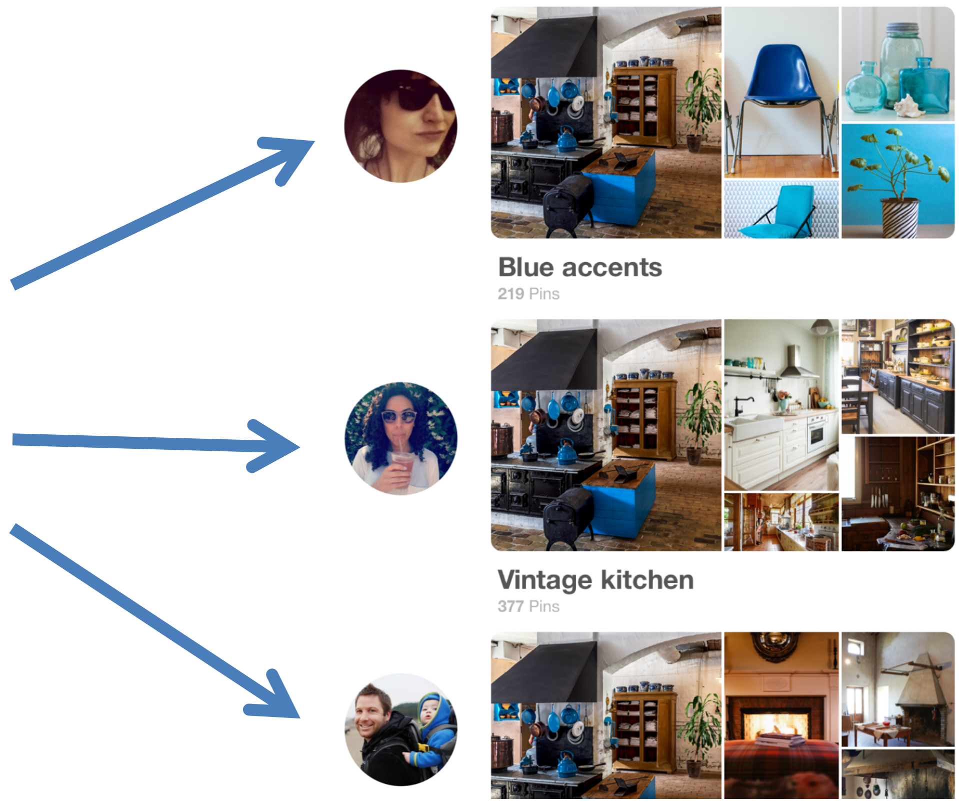
On real datasets PGNN obtains +61% ROC AUC over GCN, GAT, GIN

PinSAGE for Recommender Systems

Graph Convolutional Neural Networks for Web-Scale Recommender Systems. R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec. *KDD*, 2018.

Pinterest





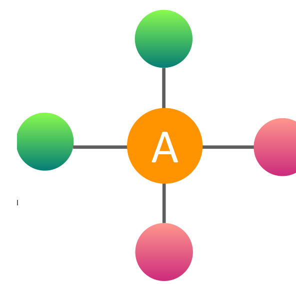
- 300M users
- 4+B pins, 2+B boards

Application: Pinterest

PinSage graph convolutional network:

- Goal: Generate embeddings for nodes in a large-scale Pinterest graph containing billions of objects
- Key Idea: Borrow information from nearby nodes
  - E.g., bed rail Pin might look like a garden fence, but gates and beds are rarely adjacent in the graph

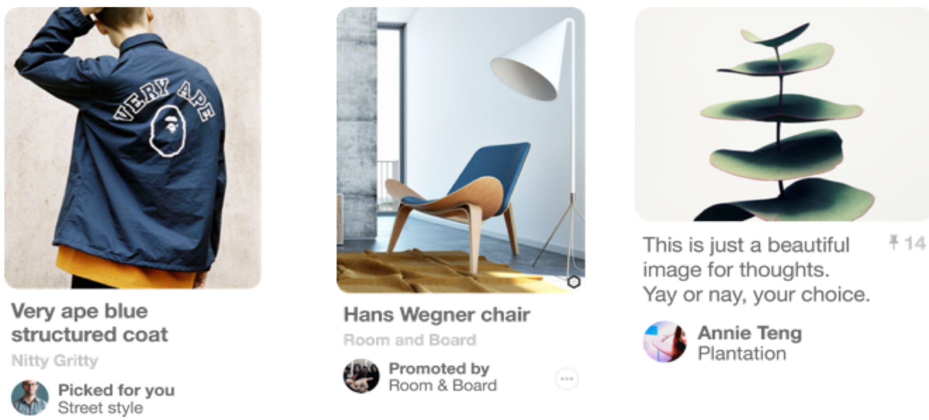




  - Pin embeddings are essential to various tasks like recommendation of Pins, classification, ranking
    - Services like “Related Pins”, “Search”, “Shopping”, “Ads”

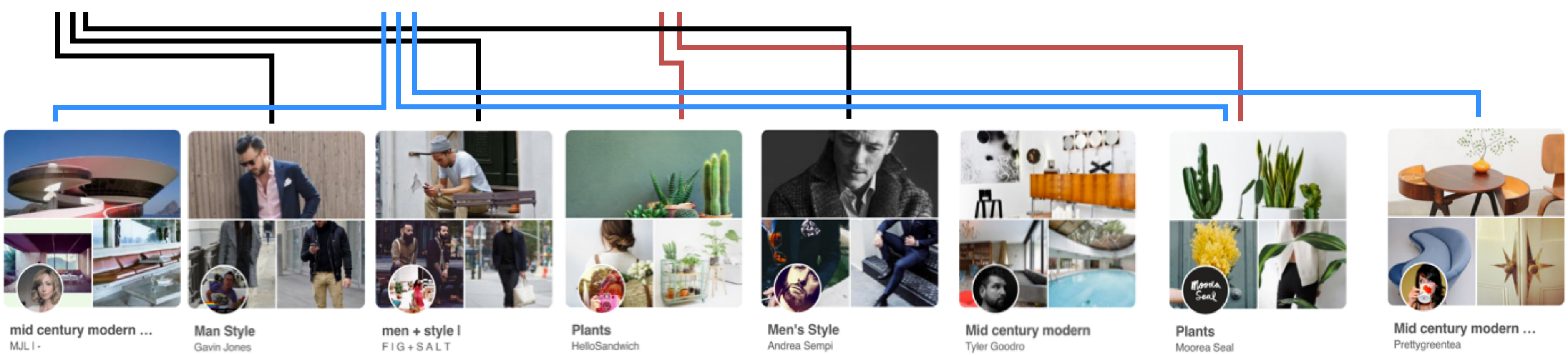
Pinterest Graph

Human curated collection of pins



Pins: Visual bookmarks someone has saved from the internet to a board they've created.

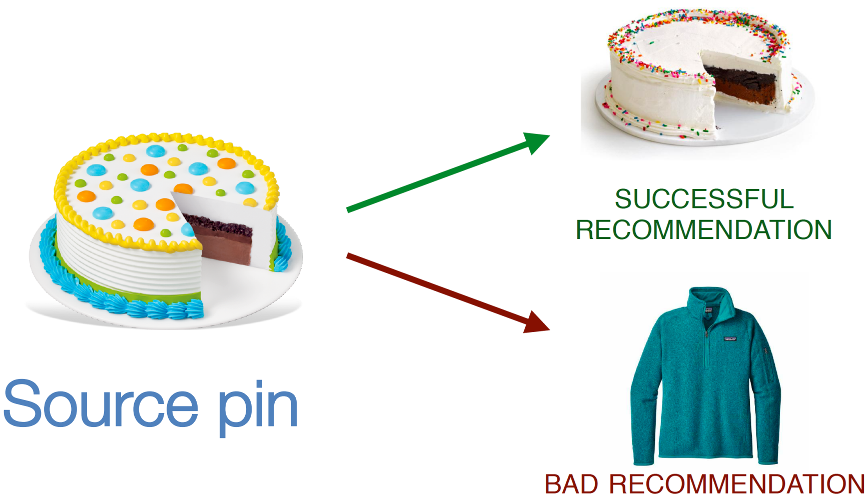
Pin features: Image, text, links



Boards

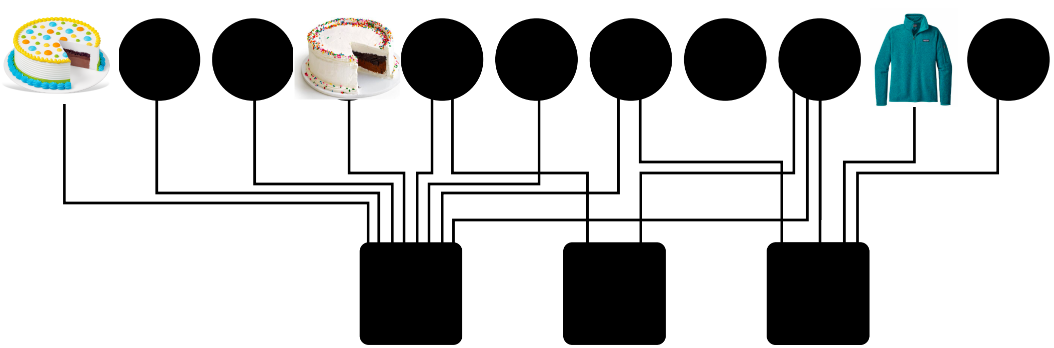
Pin Recommendation

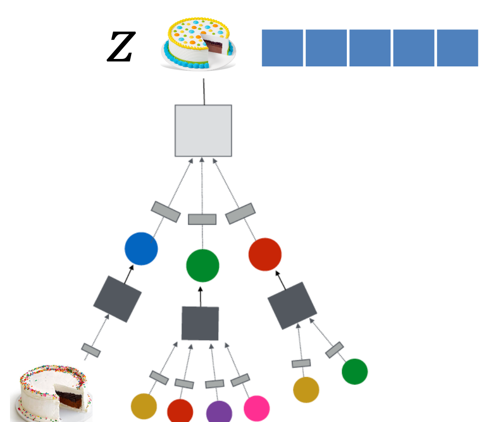
Task: Recommend related pins to users



Task: Learn node embeddings z_i such that
$d(z_{cake1}, z_{cake2}) < d(z_{cake1}, z_{sweater})$

Predict whether two nodes in a graph are related





PinSAGE Training

Goal: Identify target pin among 3B pins

- Issue: Need to learn with resolution of 100 vs. 3B
- Massive size: 3 billion nodes, 20 billion edges
- Idea: Use harder and harder negative samples

Source pin

Positive

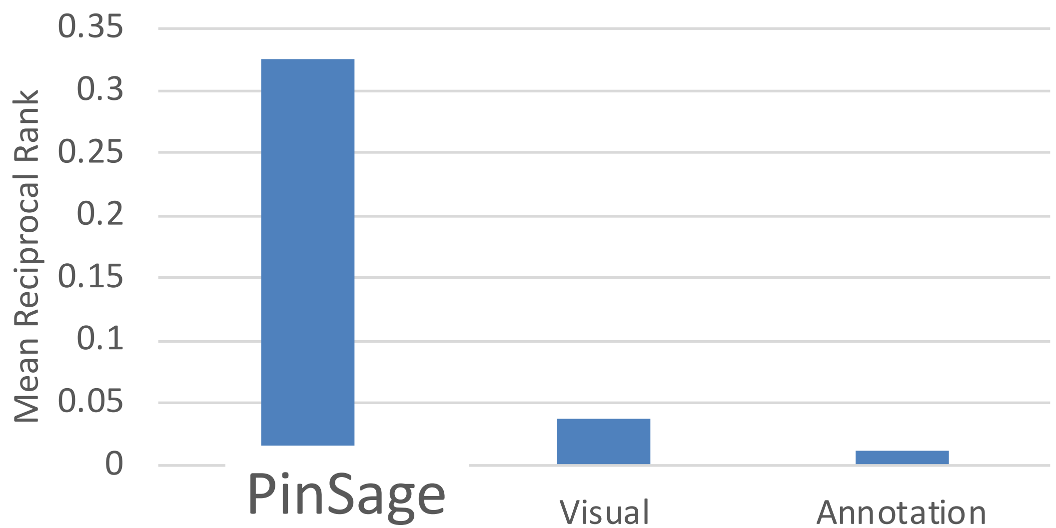
Easy negative

Hard negative

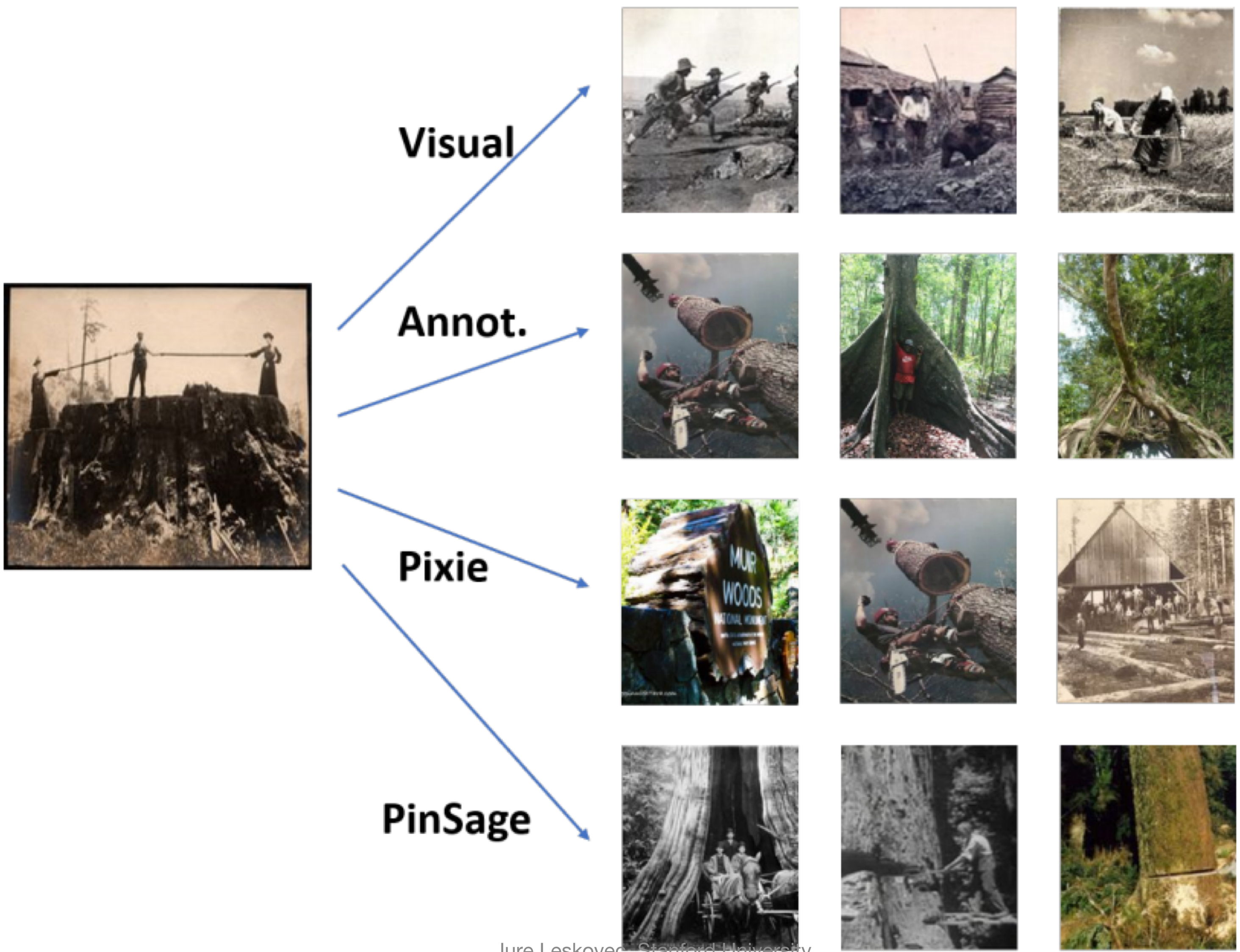
PinSAGE Performance

Related Pin recommendations

- Given a user is looking at pin **Q**, predict what pin **X** are they going to save next
- **Setup:** Embed 3B pins, perform nearest neighbor to generate recommendations



PinSAGE Example



Computational Drug Discovery: Drug Side Effect Prediction

Modeling Polypharmacy Side Effects with Graph Convolutional Networks. M. Zitnik, M. Agrawal, J. Leskovec. Bioinformatics, 2018.

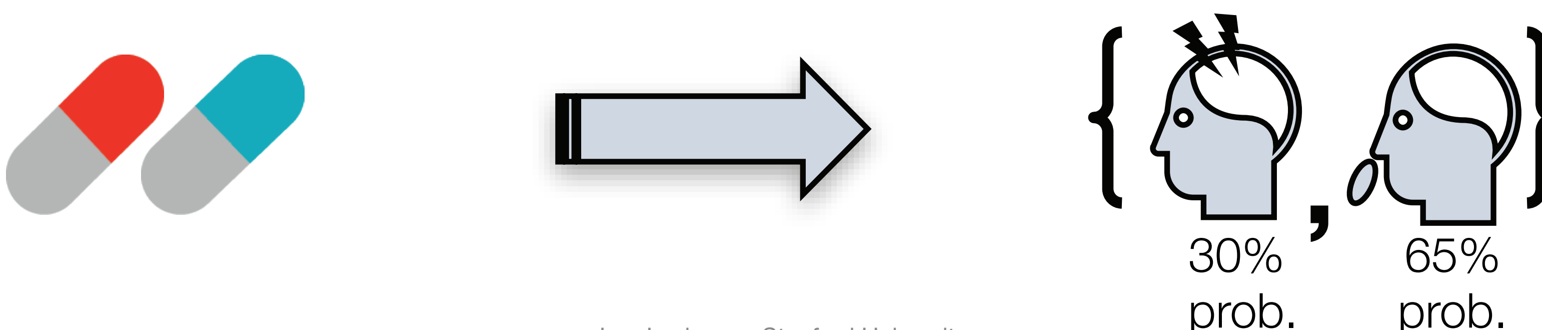
http://snap.stanford.edu/decagon/

Polypharmacy side effects

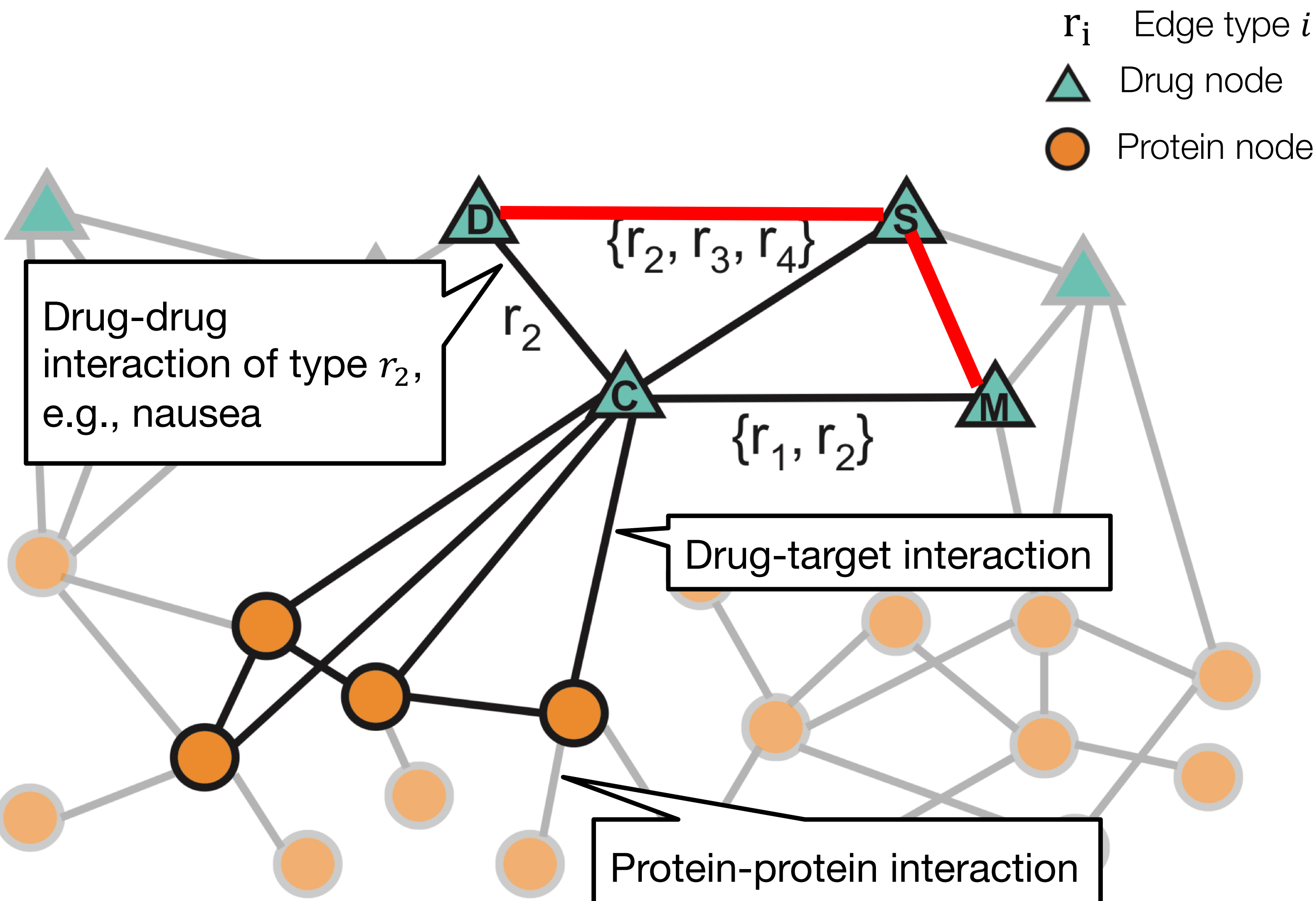
Many patients take multiple drugs to treat complex or co-existing diseases:

- 46% of people ages 70-79 take more than 5 drugs
- Many patients take more than 20 drugs to treat heart disease, depression, insomnia, etc.

Task: Given a pair of drugs predict adverse side effects



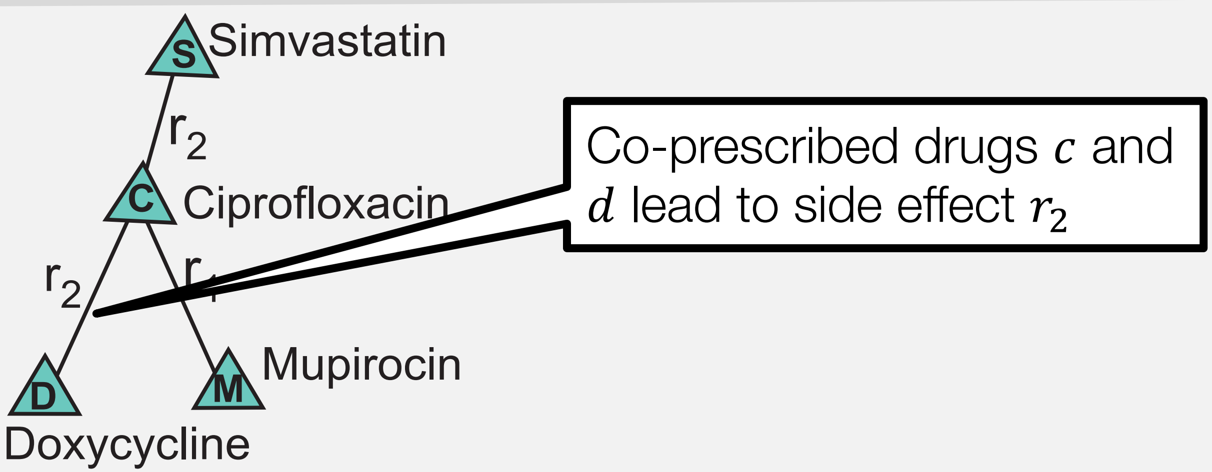
Approach: Build a Graph



Task: Link Prediction

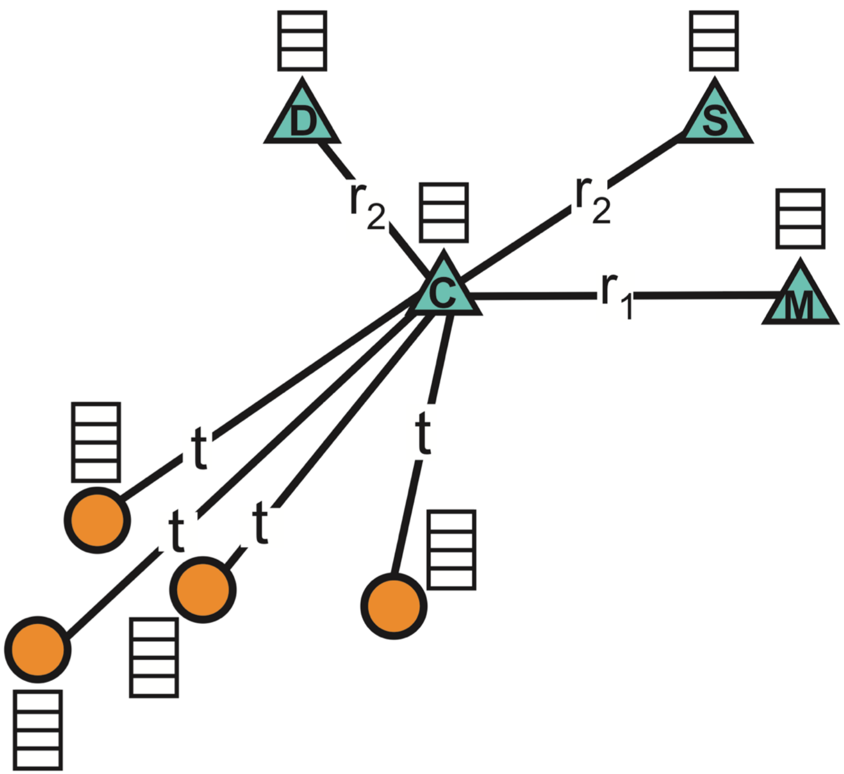
Task: Given a partially observed graph, predict labeled edges between drug nodes

Example query: Given drugs c, d, how likely is an edge (c, r_2, d)?

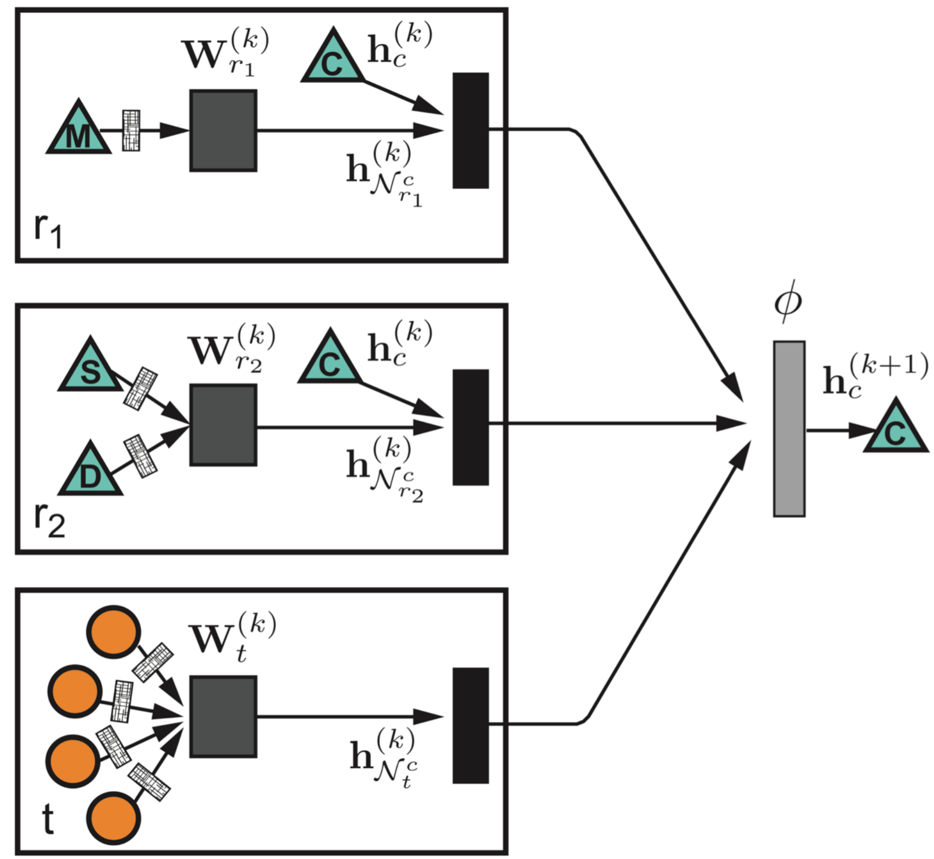


Decagon: Graph Neural Net

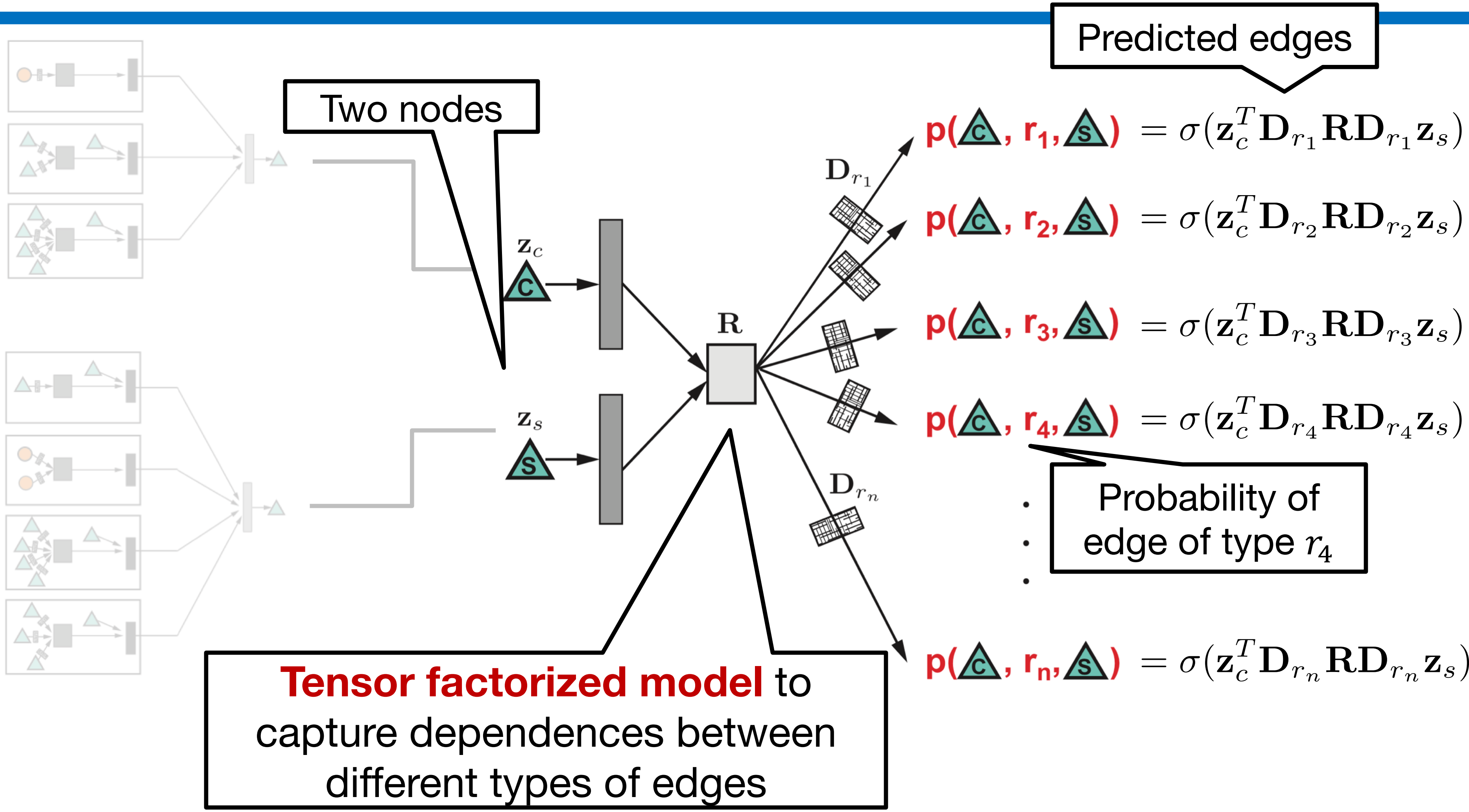
Network neighborhood of node C



Node C's computation graph

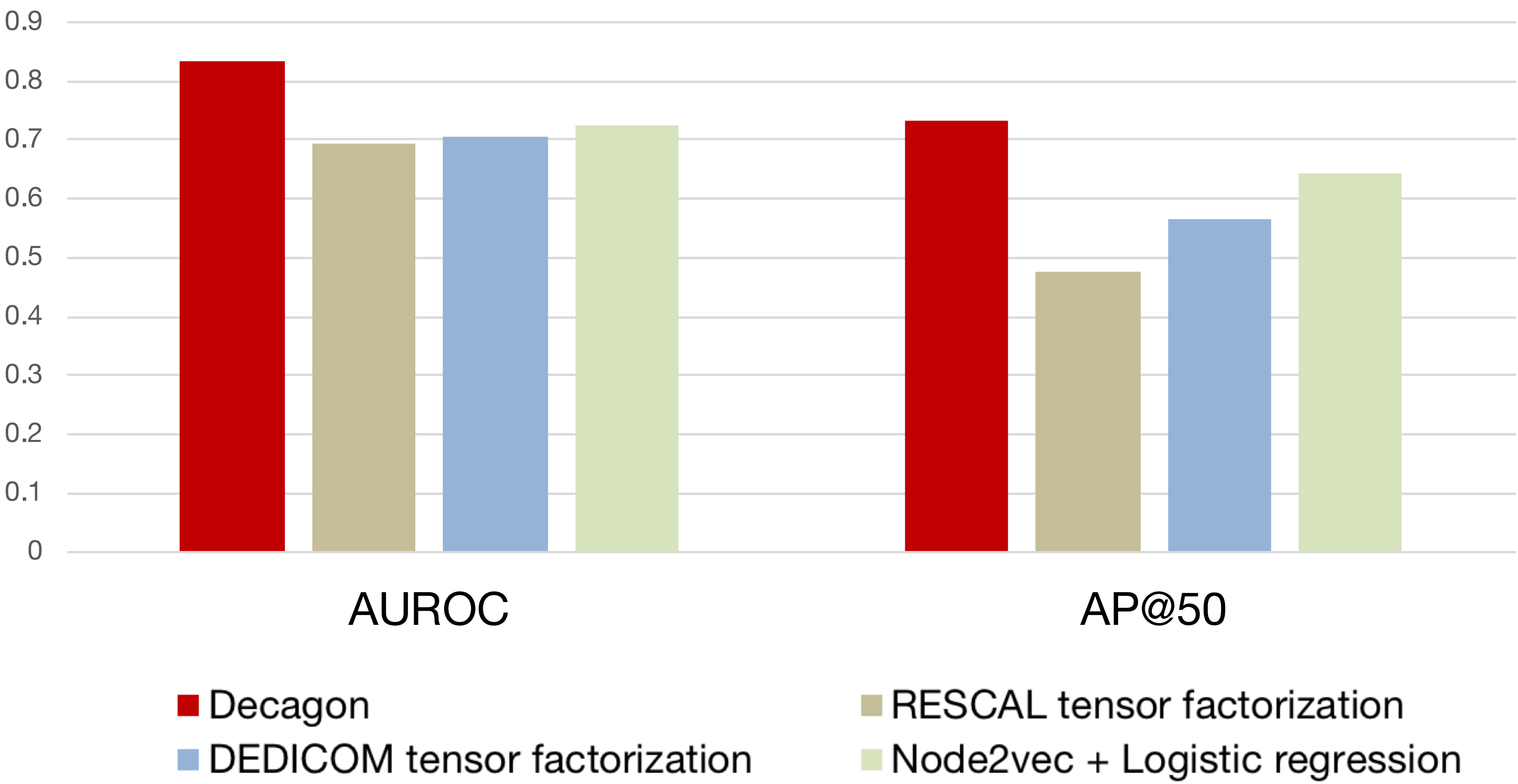


Decoder: Link Prediction



$\mathbf{R}, \mathbf{D}_{r_i}$ Parameter weight matrices

Results: Side Effect Prediction



36% average in AP@50 improvement over baselines

De novo Predictions

Rank	Drug c	Drug d	Side effect r
1	Pyrimethamine	Aliskiren	Sarcoma
2	Tigecycline	Bimatoprost	Autonomic neuropathy
3	Omeprazole	Dacarbazine	Telangiectases
4	Tolcapone	Pyrimethamine	Breast disorder
5	Minoxidil	Paricalcitol	Cluster headache
6	Omeprazole	Amoxicillin	Renal tubular acidosis
7	Anagrelide	Azelaic acid	Cerebral thrombosis
8	Atorvastatin	Amlodipine	Muscle inflammation
9	Aliskiren	Tioconazole	Breast inflammation
10	Estradiol	Nadolol	Endometriosis

De novo Predictions

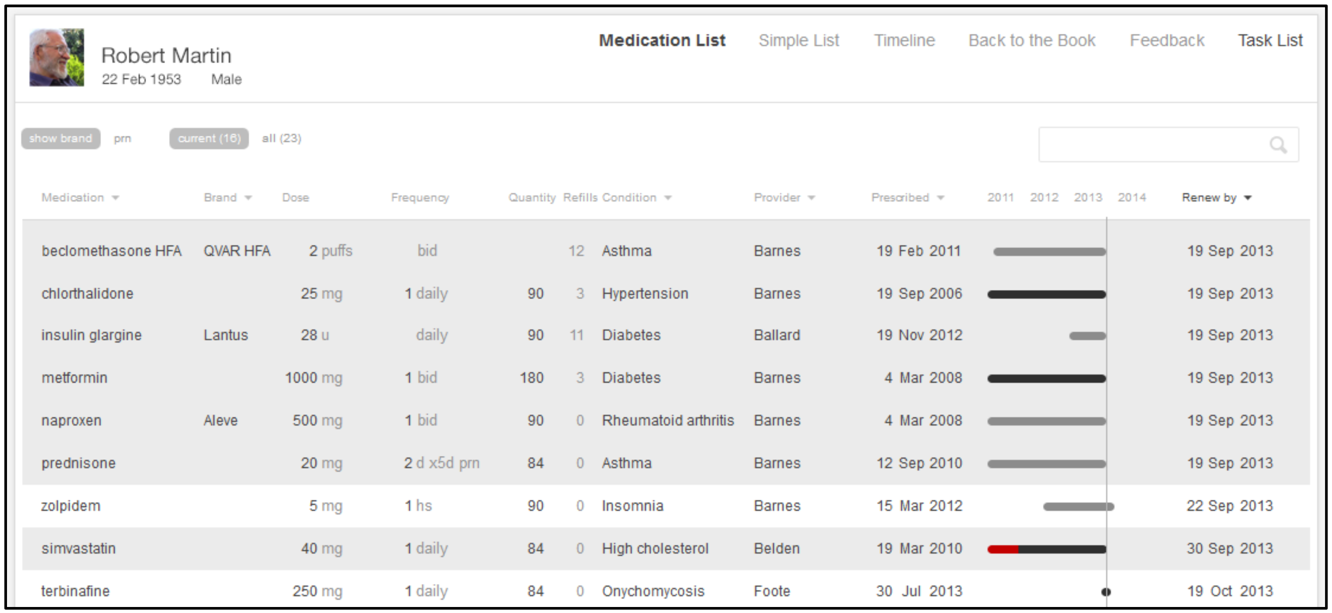
Rank	Drug c	Drug d	Side effect r	Evidence found
1	Pyrimethamine	Aliskiren	Sarcoma	Stage *et al.* 2015
2	Tigecycline	Bimatoprost	Autonomic neuropathy	
3	Omeprazole	Dacarbazine	Telangiectases	
4	Tolcapone	Pyrimethamine	Breast disorder	Bicker *et al.* 2017
5	Minoxidil	Paricalcitol	Cluster headache	
6	Omeprazole	Amoxicillin	Renal tubular acidosis	Russo *et al.* 2016
7	Anagrelide	Azelaic acid	Cerebral thrombosis	
8	Atorvastatin	Amlodipine	Muscle inflammation	Banakh *et al.* 2017
9	Aliskiren	Tioconazole	Breast inflammation	Parving *et al.* 2012
10	Estradiol	Nadolol	Endometriosis	

Case Report

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

Predictions in the Clinic

Clinical validation via drug-drug interaction markers, lab values, and



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	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
prednisone		20 mg	2 d x5d prn	84	0	Asthma	Barnes	12 Sep 2010		19 Sep 2013
zolpidem		5 mg	1 hs	90	0	Insomnia	Barnes	15 Mar 2012		22 Sep 2013
simvastatin		40 mg	1 daily	84	0	High cholesterol	Belden	19 Mar 2010		30 Sep 2013
terbinafine		250 mg	1 daily	84	0	Onychomycosis	Foote	30 Jul 2013		19 Oct 2013

NEWTON-WELLESLEY HOSPITAL

MASSACHUSETTS GENERAL HOSPITAL

Stanford MEDICINE

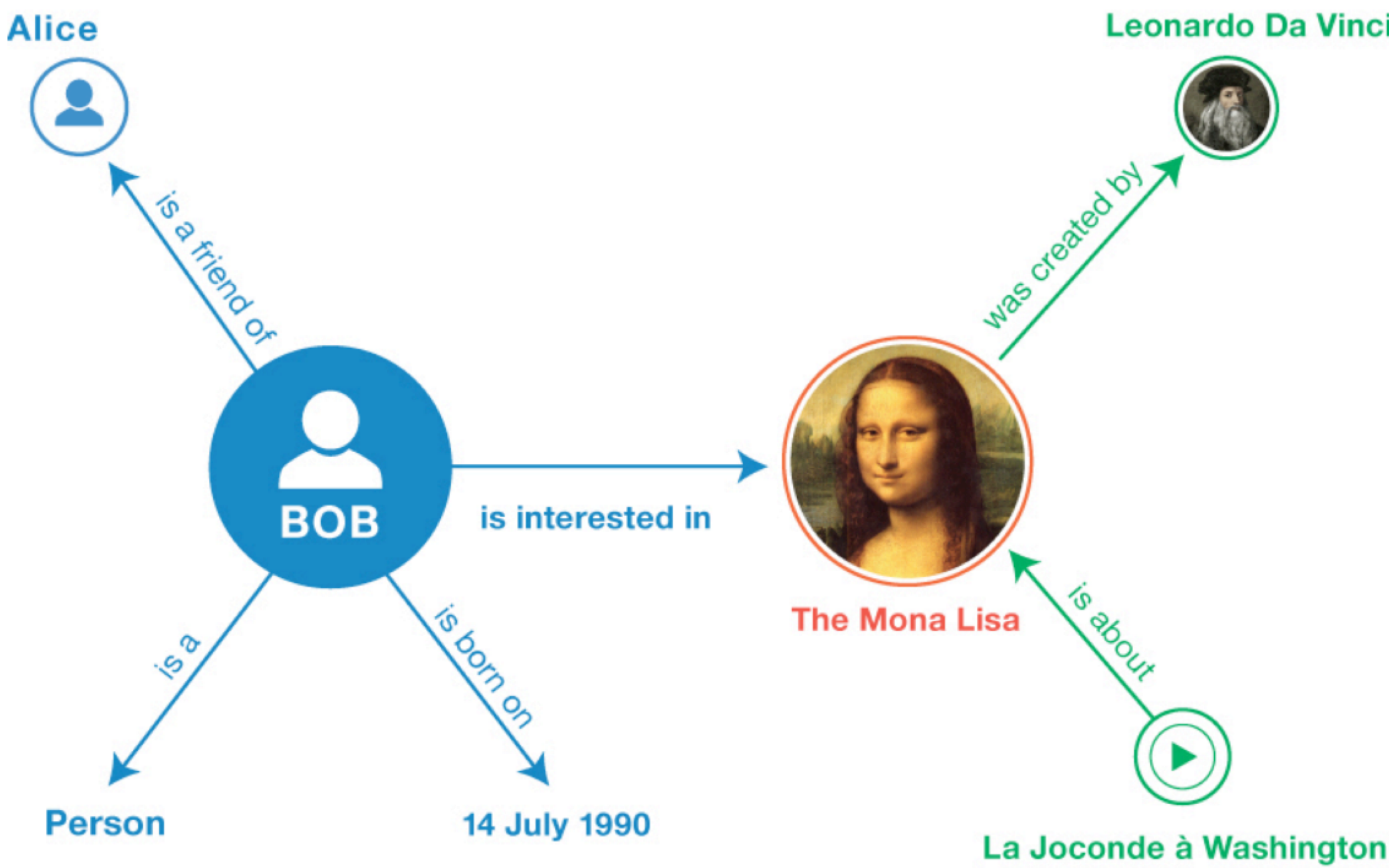
HARVARD MEDICAL SCHOOL

First method to **predict side effects of drug pairs,** even for drug combinations not yet used in patients

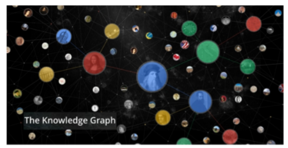
Reasoning in Knowledge Graphs

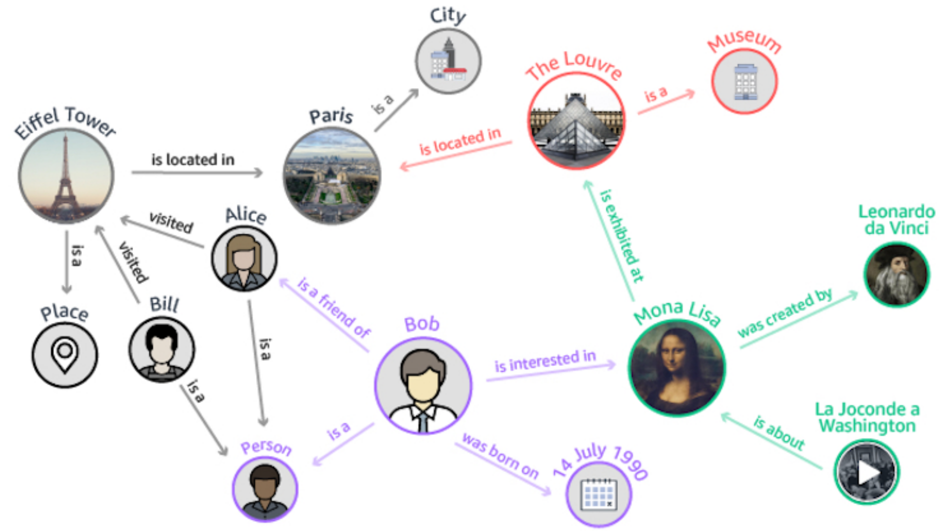
Embedding Logical Queries on Knowledge Graphs. W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J. Leskovec. *Neural Information Processing Systems (NeurIPS)*, 2018.

Knowledge as a Graph



Knowledge Graphs (KGs)





- Knowledge Graphs are heterogenous graphs
  - Multiple types of entities and relations exist
- Facts are represented as triples (h, r, t)
  - (‘Alice’, ‘friend_with’, ‘Bob’)
  - (‘Paris’, ‘is_a’, ‘City’)

Traditional Tasks

Knowledge Graph Competion/Link Prediction

- Predict the missing head or tail for a given triple (h, r, t)
- Example:

Barack Obama BornIn United States

Barack Obama Nationality American

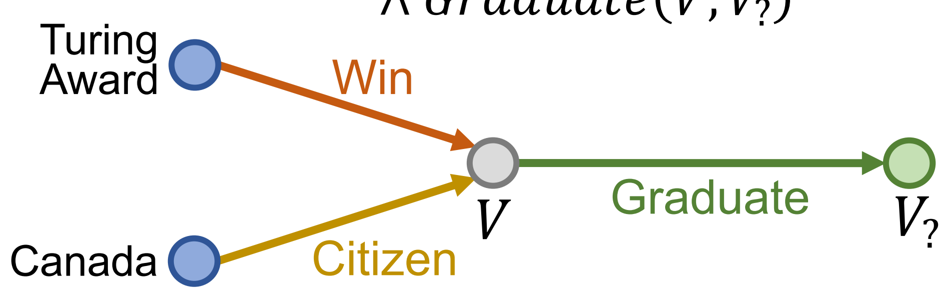
Our work: Beyond Link Prediction

Our goal: Reason over the knowledge graph using complex multi-hop queries

- Conjunctive queries: Subset of first-order logic with existential quantifier (∃) and conjunction (∧)

"Where did all Canadian citizens with Turing Award graduate?"

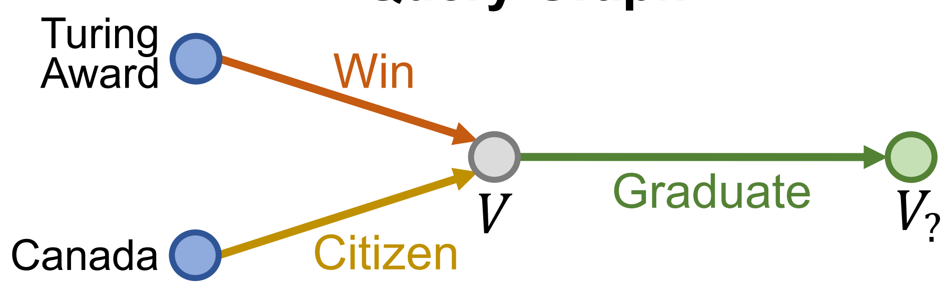
$$q = V_? \,.\, \exists\, V : Win(TuringAward, V) \wedge Citizen(Canada, V) \wedge Graduate(V, V_?)$$



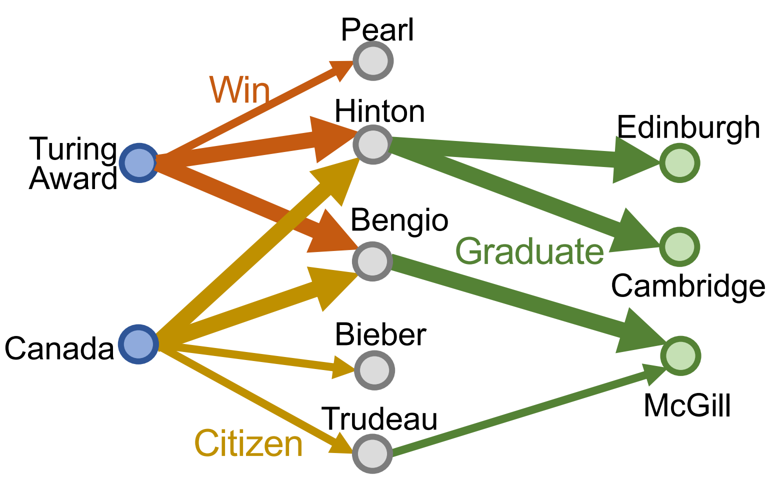
Answering Queries in KGs

"Where did Canadian citizens with Turing Award graduate?"

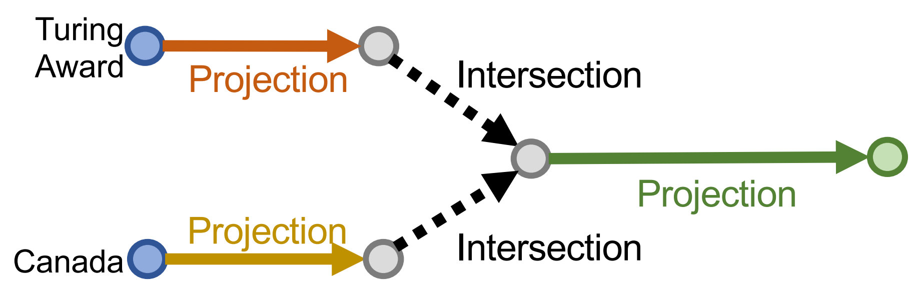
Query Graph



Knowledge Graph



Computation Graph

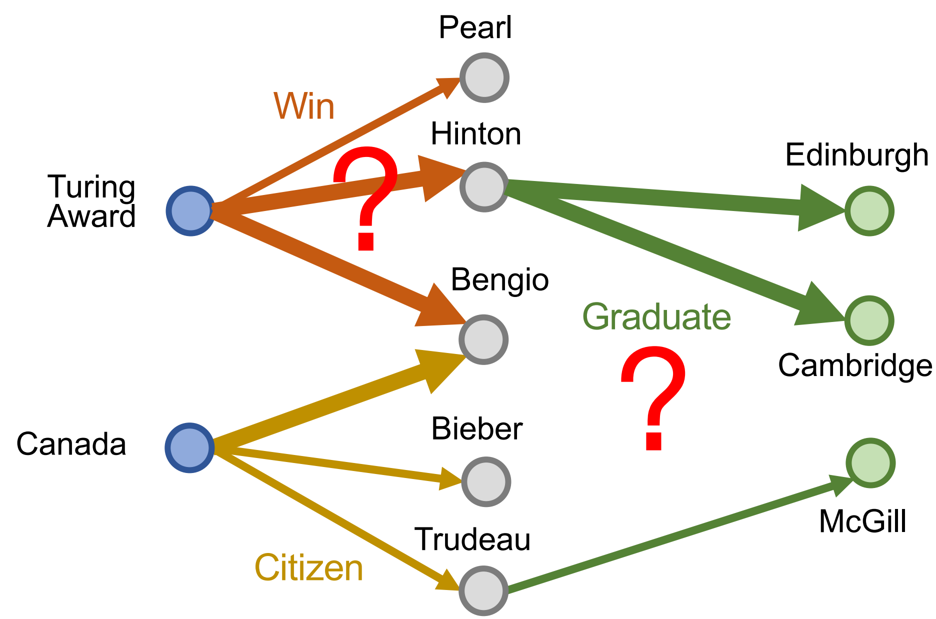


Each point corresponds to a set of entities

Why is it Hard?

Key challenge: Big graphs and queries can involve noisy and unobserved data!

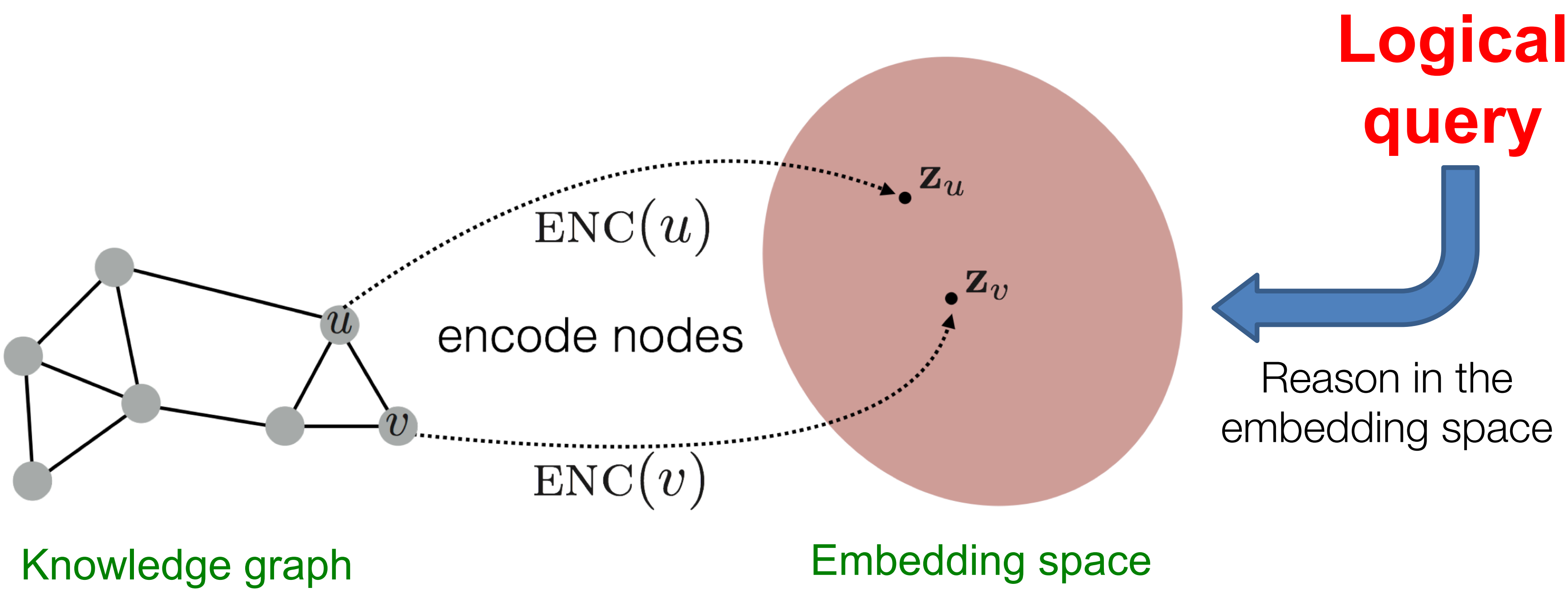
Some links might be noisy or missing



Problem: Naïve link prediction and graph template matching are too expensive

Our Idea: Query Embedding

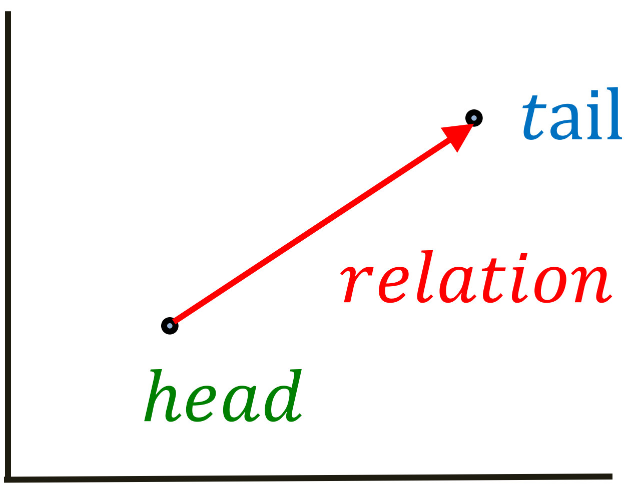
Use representation learning to map a graph into a Euclidean space and **learn to reason in that space**

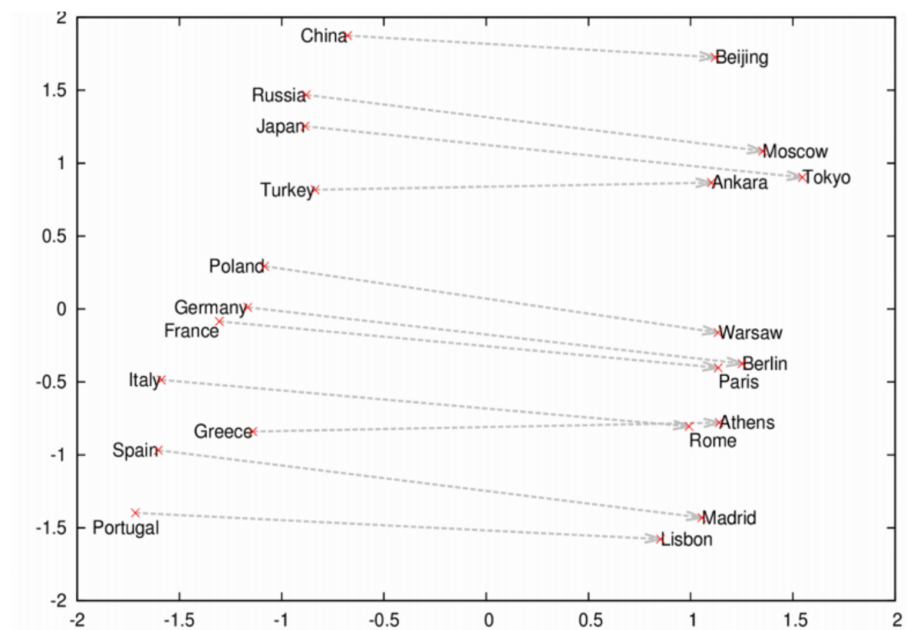


Semantic Embeddings

Remember Word2vec:

- TransE [Bordes et al., 2013]:
  For a triple (h, r, t): $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$





Our Idea: **Query2Box**

Idea:

- **1)** Embed nodes of the graph
- **2)** For every logical operator learn a spatial operator

So that:

- **1)** Take an arbitrary logical query. Decompose it into a set of logical operators (∃,∧,∨)
- **2)** Apply a sequence of spatial operators to embed the query
- **3)** Answers to the query are entities close to the embedding of the query

Our Idea: **Query2Box**

Idea:

- **1)** Embed nodes of the graph
- **2** ... al
  o

So t

- **1** ... pose
  it
- **2** ... to
  e

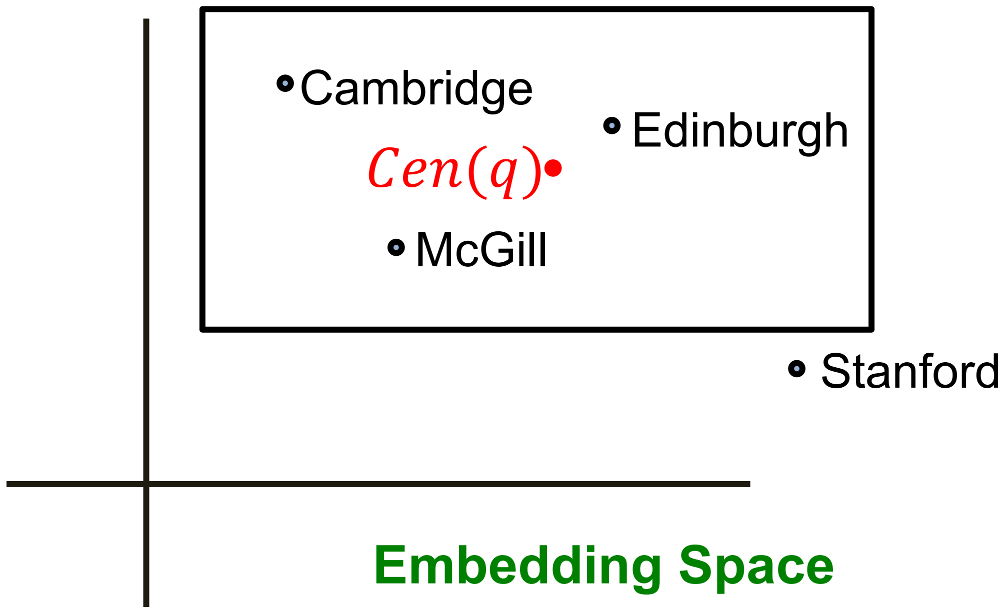
Key insight:
Represent query as a box.
Operations (union, intersection)
are well defined over boxes.

- **3)** Answers to the query are entities close to the embedding of the query

Embedding Queries

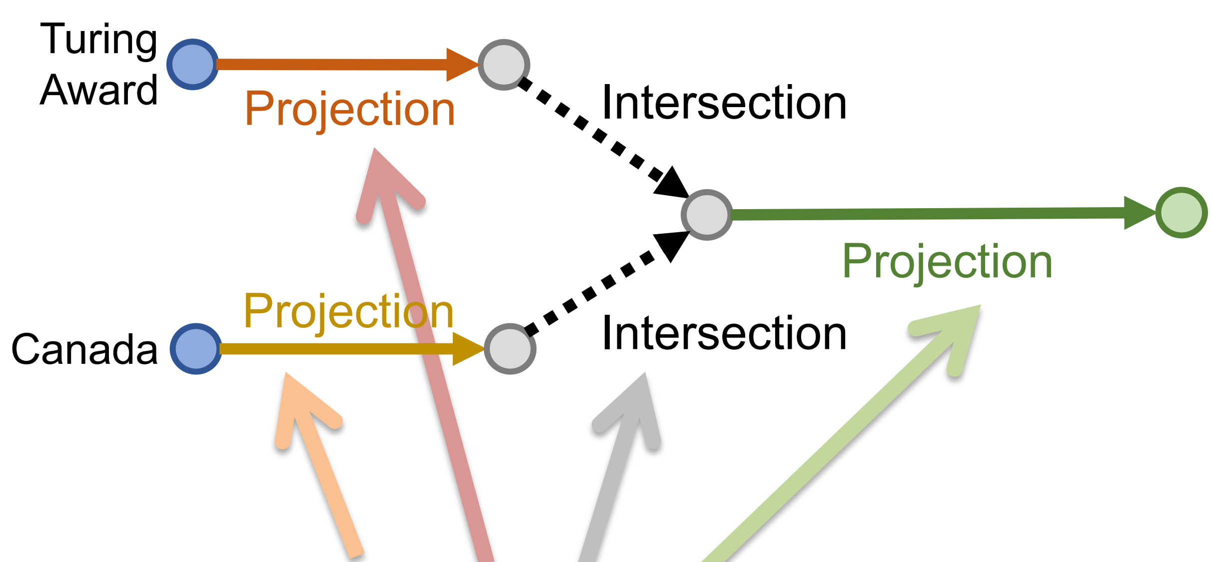
Query2Box embedding:

Embed queries with hyper-rectangles (boxes): $\mathbf{q} = (Cen(q), Off(q))$.



Embedding Queries

Computation Graph

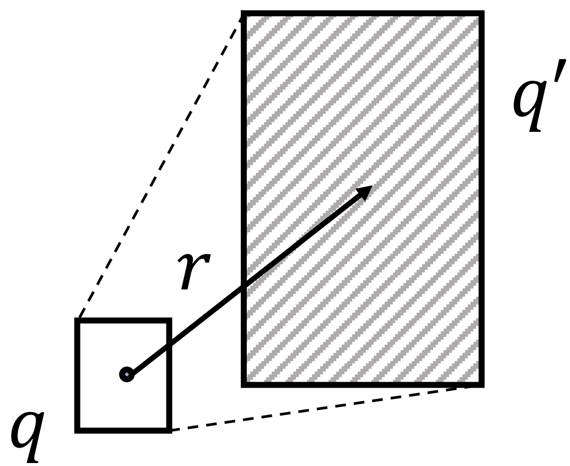


- Geometric Projection Operator
- Geometric Intersection Operator

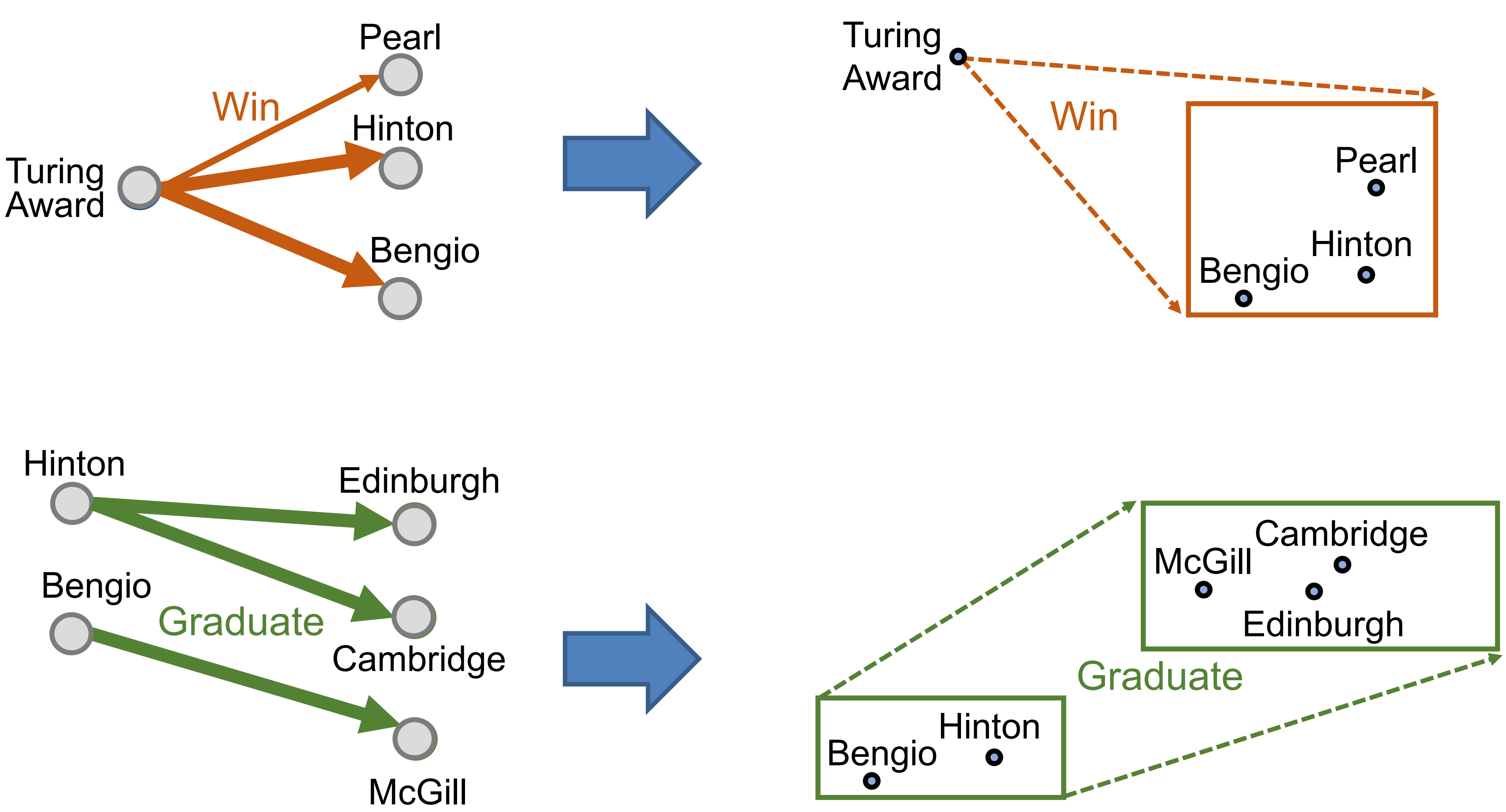
Projection Operator

Geometric Projection Operator $\boldsymbol{\mathcal{P}}$

- $\mathcal{P}$: Box × Relation → Box



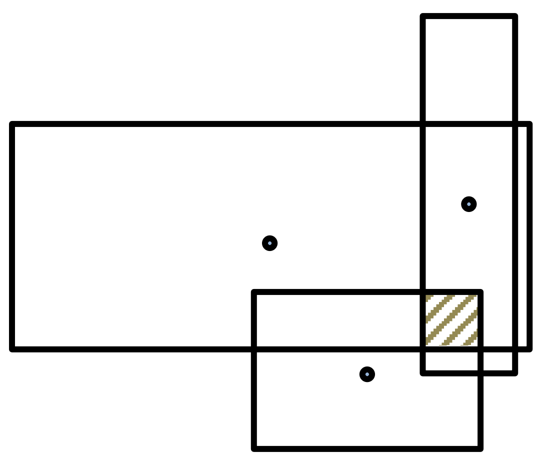
Projection Operator: Example



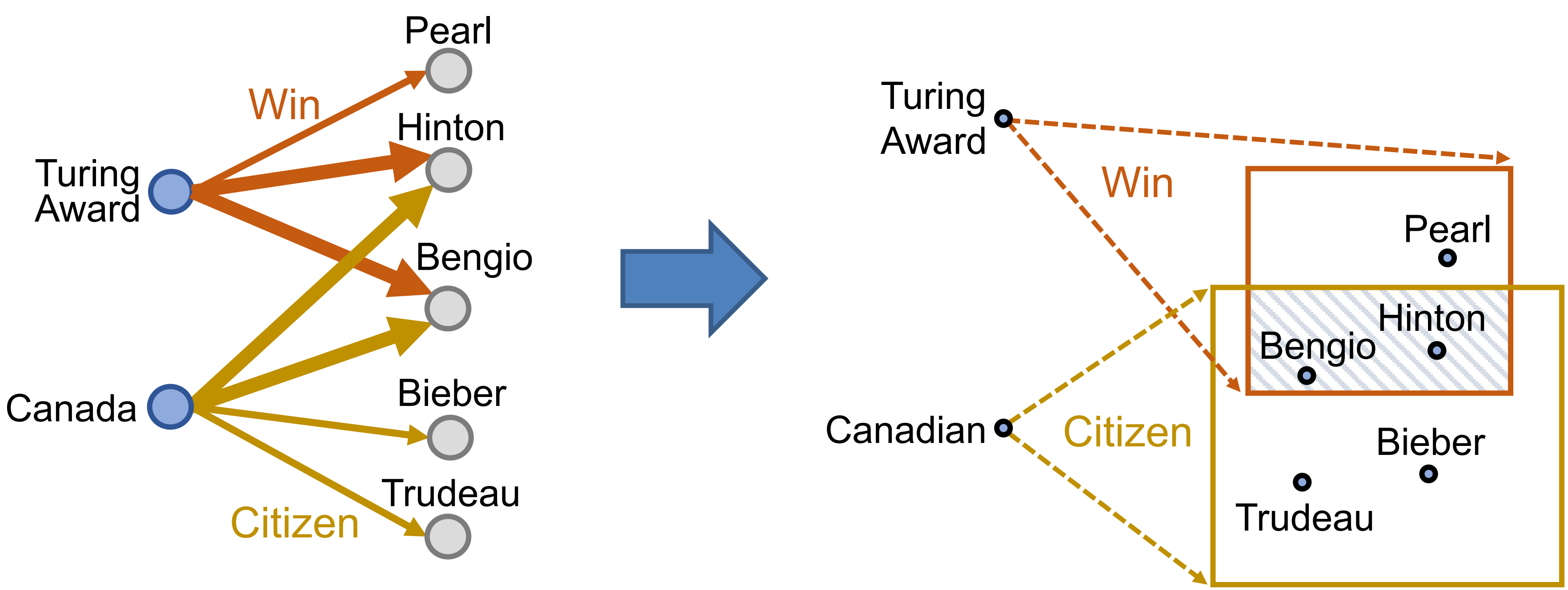
Intersection Operator

Geometric Intersection Operator $\mathcal{J}$

- $\mathcal{J}$: Box $\times \cdots \times$ Box $\rightarrow$ Box
  - The new center is a weighted average
  - The new offset shrinks



Intersection Operator: Example



Benefits of **Query2Box**

Scalability and efficiency:

- Any query can be reduced to a couple of matrix operations and a single k-nearest neighbor search

Generality:

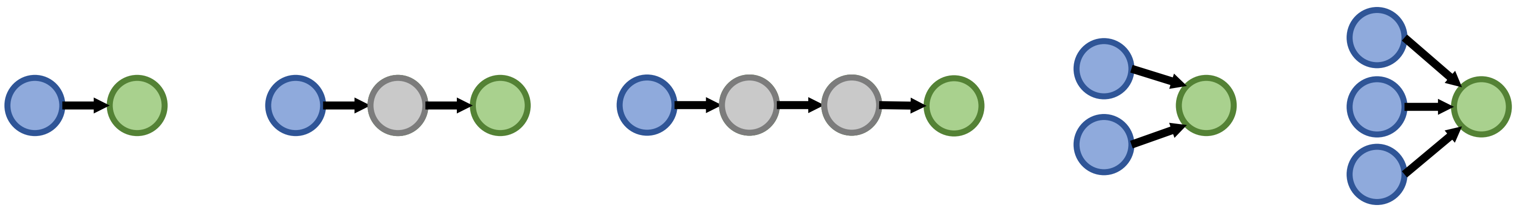
- We can answer any query (even those we have never seen before)

Robustness to noise:

- Graph can contain missing and noisy relationships

Query2Box : Model Training

Training examples: Queries on the graph



- **Positives:** Path with a known answer
- **Negatives:** Random nodes of the correct answer type
- **Goal:** Find embeddings and operators so that that queries give correct answers

Experimental Setup

We essentially learn to “memorize” the answers to queries

- We embed entities so that our geometric operators give correct answers

Questions:

- Does our method generalize to new unseen queries?
- Does our method generalize to new query structures?
- Can method handle missing relations?

Experimental Setup

- **Training:**
  - Remove 10% of KG edges
  - Sample training queries and (non)answers
  - Train the model
- **Test set:**
  - Test queries/answers from the full graph
  - Ensure that the test queries are **not** directly answerable in the training graph
    - Every test query has at least one deleted edge
    - Note: Query template matching would have accuracy of random guessing

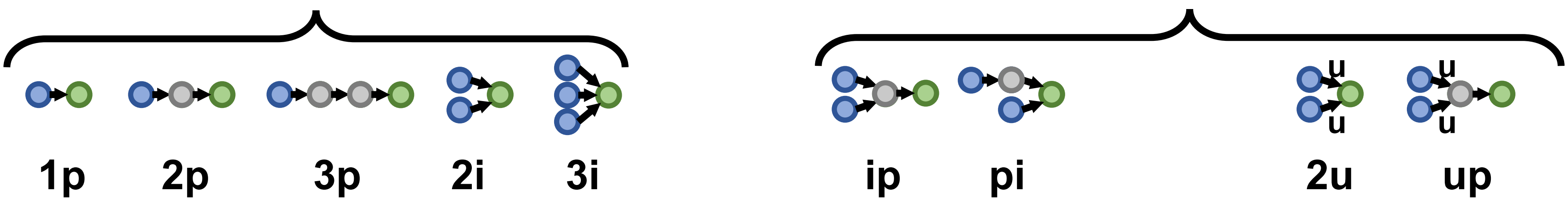
KG and Query Statistics

- **Freebase:** FB15K, FB15K-237

Dataset	Entities	Relations	Training Edges	Validation Edges	Test Edges	Total Edges
FB15k	14,951	1,345	483,142	50,000	59,071	592,213
FB15k-237	14,505	237	272,115	17,526	20,438	310,079

- **Queries:**

Training Conjunctive Queries **Unseen Conjunctive Queries** **Union Queries**



Queries	Training		Validation		Test	
Dataset	1p	others	1p	others	1p	others
FB15k	273,710	273,710	59,097	8,000	67,016	8,000
FB15k-237	149,689	149,689	20,101	5,000	22,812	5,000

Experimental Results

Method	Avg	1p	2p	3p	2i	3i	ip	pi	2u	up
Q2B	**0.268**	**0.467**	**0.24**	**0.186**	**0.324**	**0.453**	**0.108**	**0.205**	**0.239**	**0.193**
GQE	0.228	0.402	0.213	0.155	0.292	0.406	0.083	0.170	0.169	0.163
GQE-DOUBLE	0.230	0.405	0.213	0.153	0.298	0.411	0.085	0.182	0.167	0.160

Table 3: H@3 on test set for QUERY2BOX vs. GQE on FB15k-237.

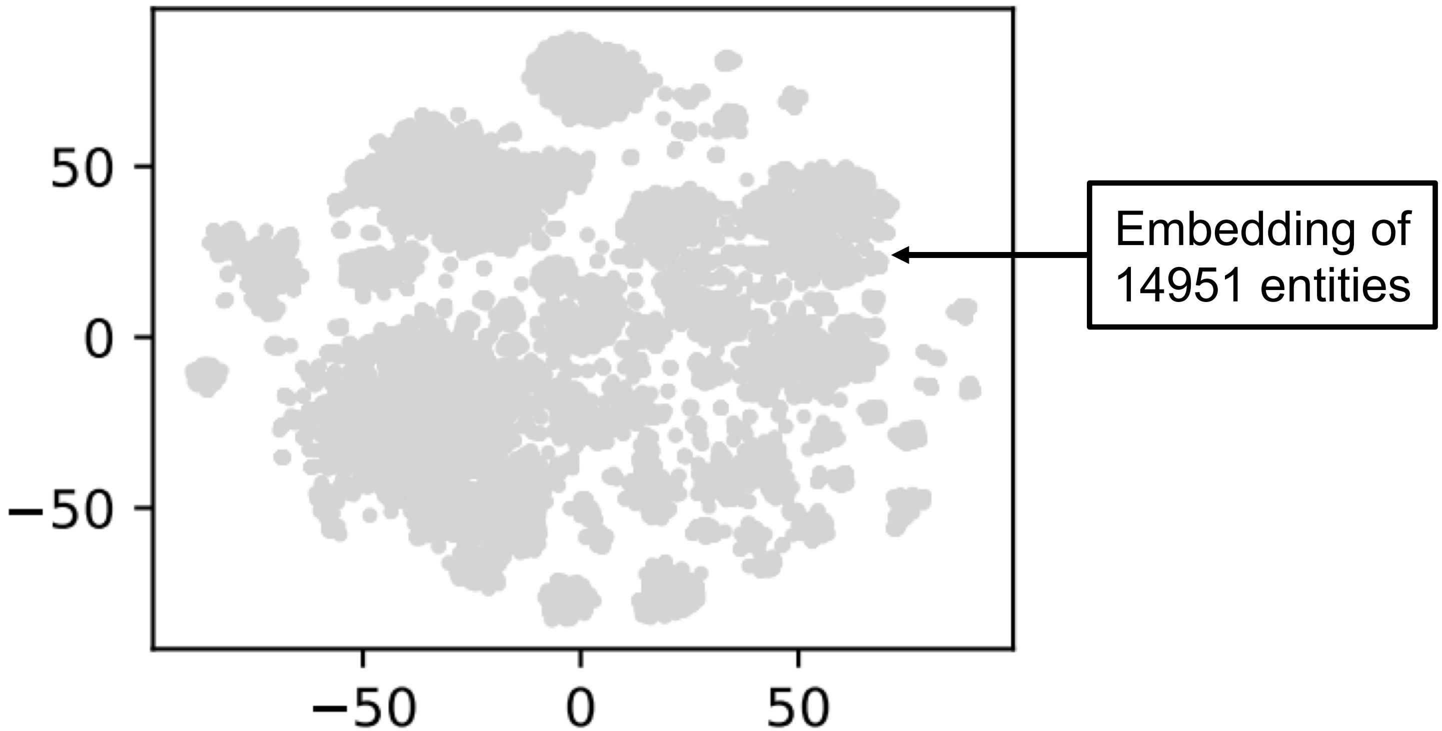
Method	Avg	1p	2p	3p	2i	3i	ip	pi	2u	up
Q2B	**0.484**	**0.786**	**0.413**	**0.303**	**0.593**	**0.712**	**0.211**	**0.397**	**0.608**	**0.330**
GQE	0.386	0.636	0.345	0.248	0.515	0.624	0.151	0.31	0.376	0.273
GQE-DOUBLE	0.384	0.63	0.346	0.250	0.515	0.611	0.153	0.32	0.362	0.271

Table 4: H@3 on test set for QUERY2BOX vs. GQE on FB15k.

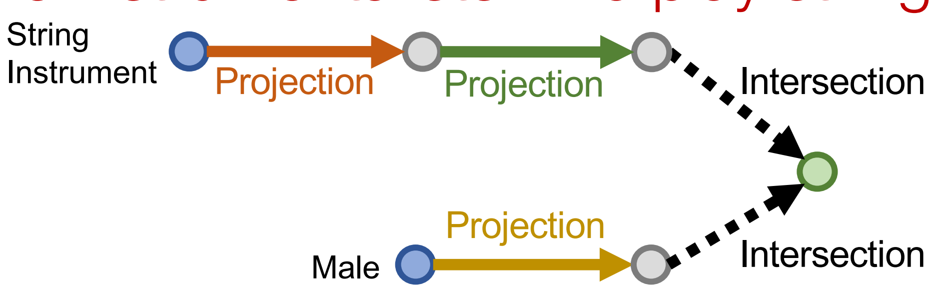
Observations:

- On “training” queries: +20% H@3
- On new conjunctive query structures: +15%
- On disjunctive queries: +36%

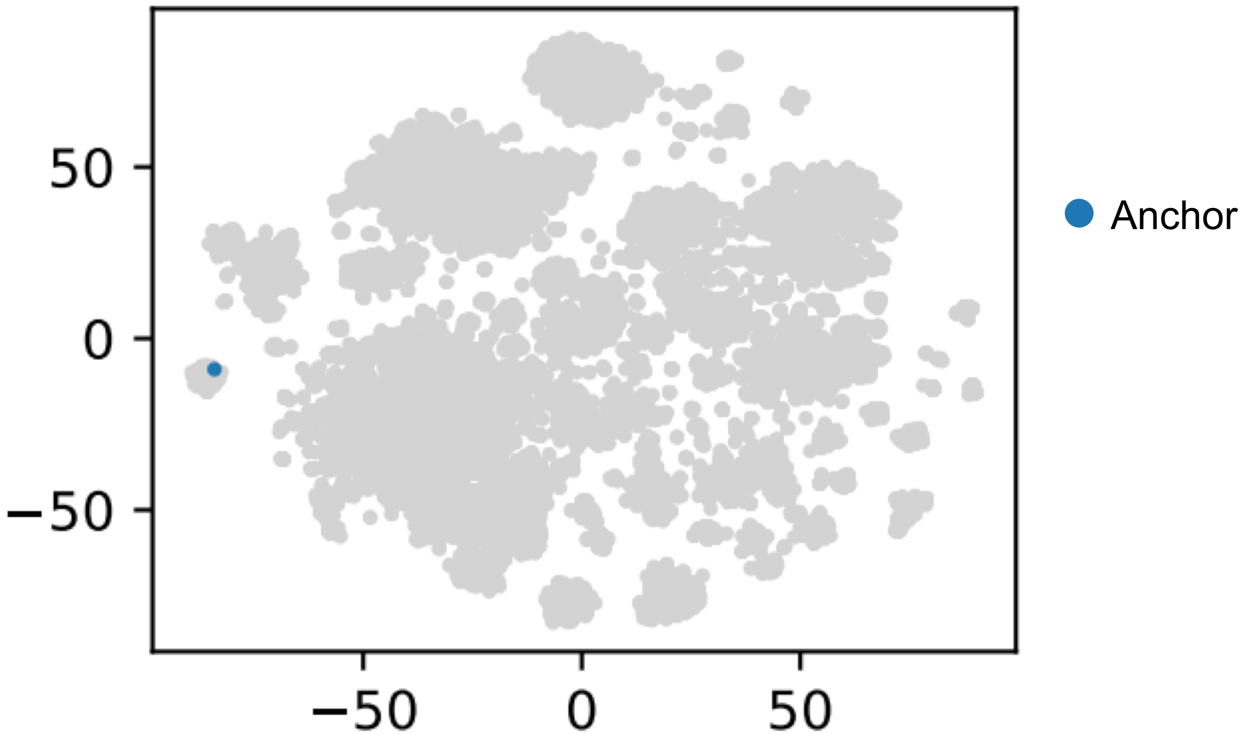
Embedding Space



“List male instrumentalists who play string instruments”

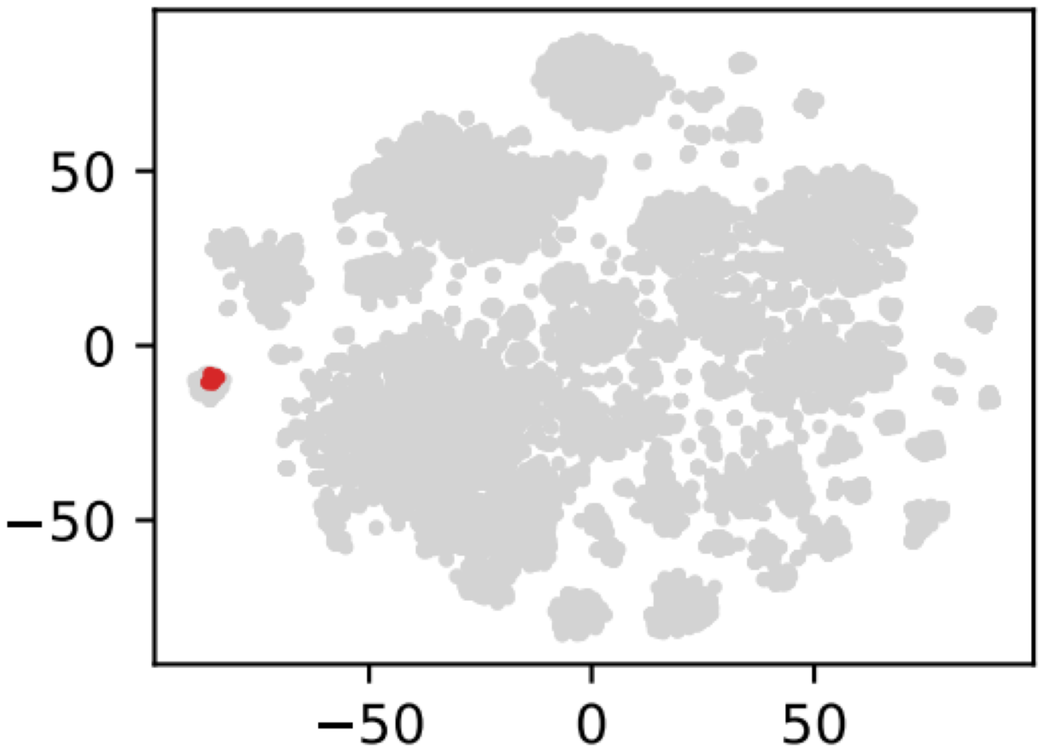


Embedding Space



“List male instrumentalists who play string instruments”

Embedding Space



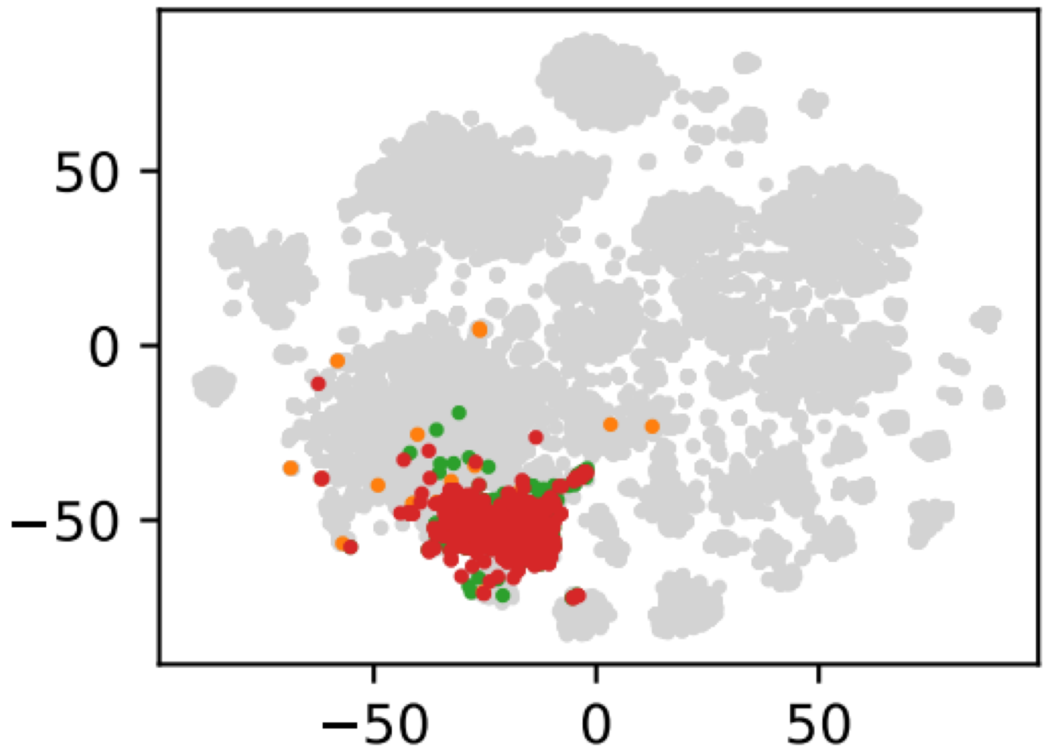
# of string instruments: 10

TP
FN
FP
TN

TPR: 100%
FPR: 0%

“List male instrumentalists who play string instruments”

Embedding Space



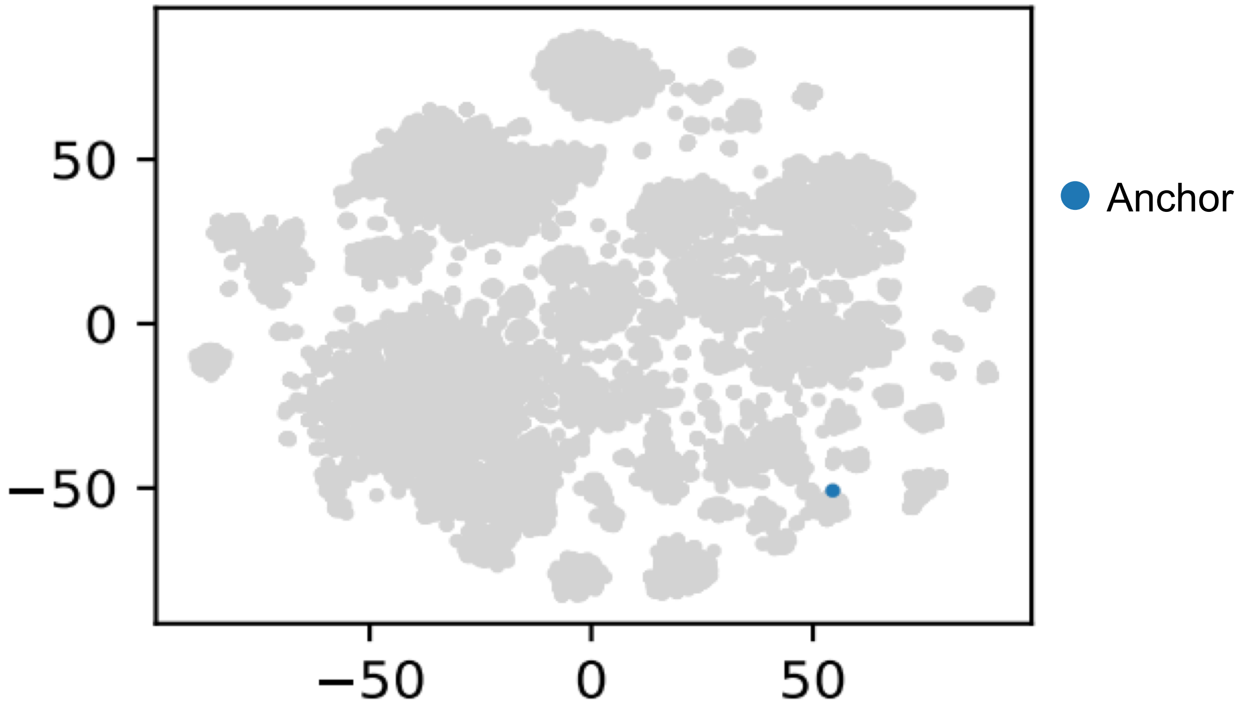
# of instrumentalists: 472

- TP
- FN
- FP
- TN

TPR: 98.4%
FPR: 0.01%

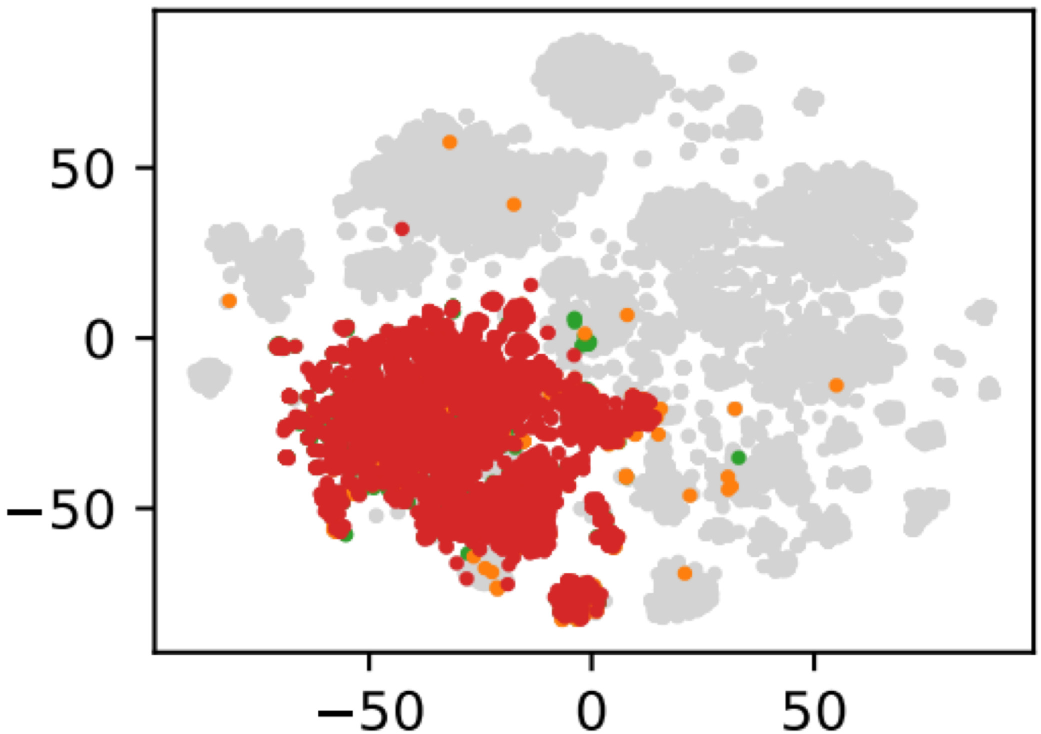
"List male instrumentalists who play string instruments"

Embedding Space



"List male instrumentalists who play string instruments"

Embedding Space



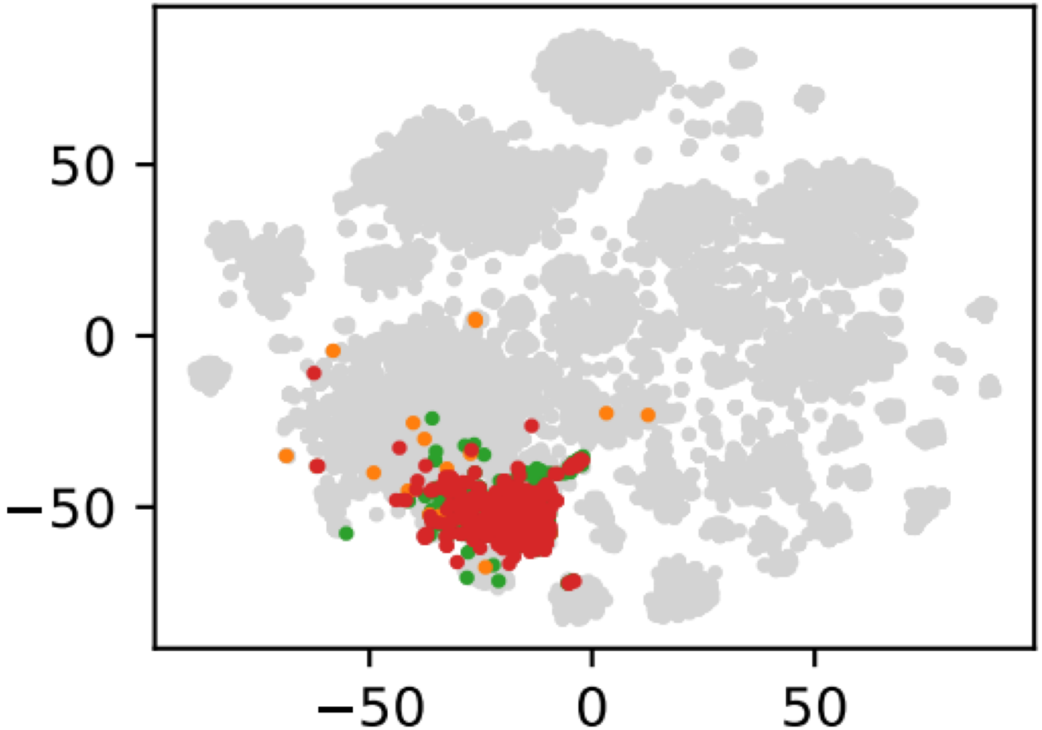
# of men: 3555

- TP
- FN
- FP
- TN

TPR: 97.9%
FPR: 0.01%

“List male instrumentalists who play string instruments”

Embedding Space

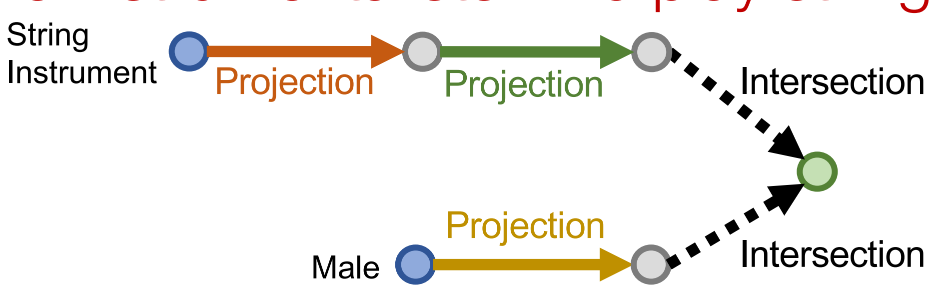


# of answers: 396

- TP
- FN
- FP
- TN

TPR: 99.4%
FPR: 0.01%

“List male instrumentalists who play string instruments”



Query2Box: Summary

- **Query2Box:**
  - Embed the query as a box
  - Logical operations become spatial operations
- Composability of queries:
  - Generalize well to unseen, extrapolated queries
  - Explicitly training for composability is important
- Instance vs. multi-hop generalization

How can this technology be used for other problems?

We can now apply neural networks much more broadly

New frontiers beyond classic neural networks that learn on images and sequences

Many other applications:

- **Nodes:** Predict tissue-specific protein functions
- **Subgraphs:** Predict which drug treats what disease
- **Graph generation:** Generate molecules/drugs

Summary

- Graph Convolutional Neural Networks
  - Generalize beyond simple convolutions
- Fuses node features & graph info
  - State-of-the-art accuracy for node classification and link prediction
- Model size independent of graph size; can scale to billions of nodes
  - Largest embedding to date (3B nodes, 20B edges)
- Leads to significant performance gains

Conclusion

Results from the past 2-3 years have shown:

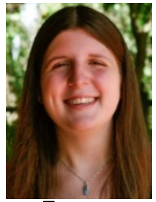
- Representation learning paradigm can be extended to graphs
- No feature engineering necessary
- Can effectively combine node attribute data with the network information
- State-of-the-art results in a number of domains/tasks
- Use end-to-end training instead of multi-stage approaches for better performance

PhD Students

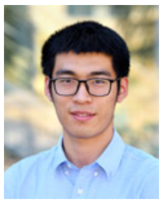

Alexandra Porter


Camilo Ruiz


Claire Donnat

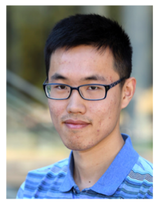

Emma Pierson


Weihua Hu


Jiaxuan You


Bowen Liu


Mohit Tiwari

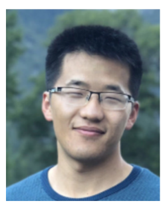

Rex Ying

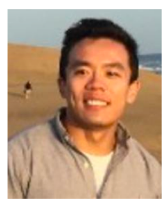
Post-Doctoral Fellows


Baharan Mirzasoleiman


Marinka Zitnik


Michele Catasta

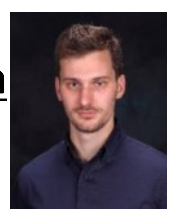

Pan Li


Shantao Li


Srijan Kumar


Hingwei Wang

Research Staff


Adrijan Bradaschia


Rok Sosic

Industry Partnerships

Funding

Collaborators

Dan Jurafsky, Linguistics, Stanford University
David Grusky, Sociology, Stanford University
Stephen Boyd, Electrical Engineering, Stanford University
David Gleich, Computer Science, Purdue University
VS Subrahmanian, Computer Science, University of Maryland
Sarah Kunz, Medicine, Harvard University
Russ Altman, Medicine, Stanford University
Jochen Profit, Medicine, Stanford University
Eric Horvitz, Microsoft Research
Jon Kleinberg, Computer Science, Cornell University
Sendhill Mullainathan, Economics, Harvard University
Scott Delp, Bioengineering, Stanford University
James Zou, Medicine, Stanford University

Postdoc positions in 3 topics:

(1) Core ML on Graphs

(2) Biomedical, Common Sense Reasoning

(3) Societal Applications of ML

References

- Tutorial on Representation Learning on Networks at WWW 2018 http://snap.stanford.edu/proj/embeddings-www/
- Inductive Representation Learning on Large Graphs. W. Hamilton, R. Ying, J. Leskovec. NIPS 2017.
- Representation Learning on Graphs: Methods and Applications. W. Hamilton, R. Ying, J. Leskovec. IEEE Data Engineering Bulletin, 2017.
- Graph Convolutional Neural Networks for Web-Scale Recommender Systems. R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, J. Leskovec. KDD, 2018.
- Modeling Polypharmacy Side Effects with Graph Convolutional Networks. M. Zitnik, M. Agrawal, J. Leskovec. Bioinformatics, 2018.
- Graph Convolutional Policy Network for Goal-Directed Molecular Graph Generation. J. You, B. Liu, R. Ying, V. Pande, J. Leskovec, NeurIPS 2018.
- Embedding Logical Queries on Knowledge Graphs. W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J. Leskovec. NeuIPS, 2018.
- How Powerful are Graph Neural Networks? K. Xu, W. Hu, J. Leskovec, S. Jegelka. ICLR 2019.
- Position-aware Graph Neural Networks. J. You, R. Ying, J. Leskovec. *ICML*, 2019.

- Code:
  - http://snap.stanford.edu/graphsage
  - http://snap.stanford.edu/decagon/
  - https://github.com/bowenliu16/rl_graph_generation
  - https://github.com/williamleif/graphqembed
  - https://github.com/snap-stanford/GraphRNN