Deep Learning with Knowledge Graphs

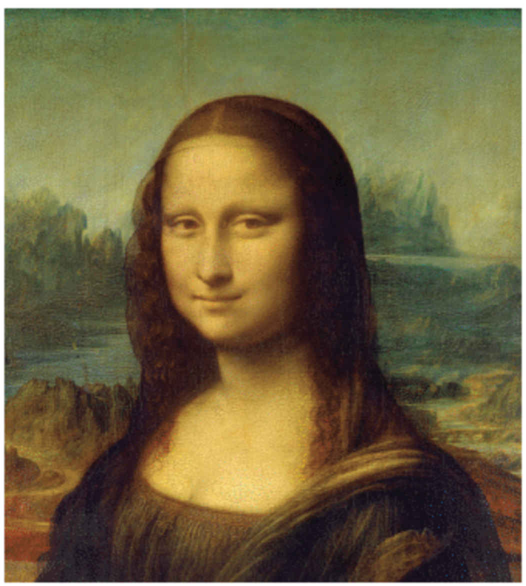
Jure Leskovec

Includes joint work with M. Fey, J. You, S. Jegelka, K. Xu, R. Ying, W. Hu, M. Zitnik, P. Battaglia, K. Subbian, N. Rao, P. Li, A. Wang

Doubt thou the stars are fire;
Doubt that the sun doth move;
Doubt truth to be a liar;
But never doubt I love...

Text

Audio signals

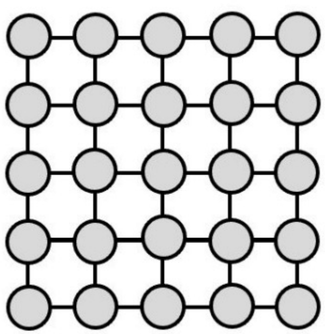


Images

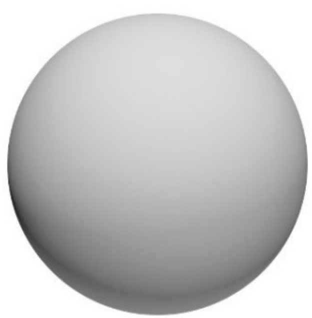
Modern
deep learning toolbox
is designed for
sequences & grids

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

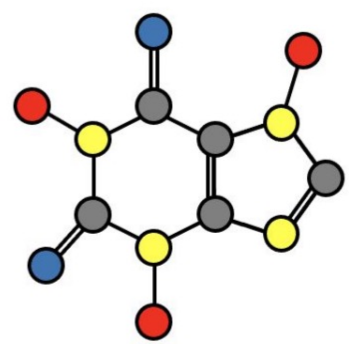
New frontiers beyond classic neural networks that learn on images and sequences



Grids



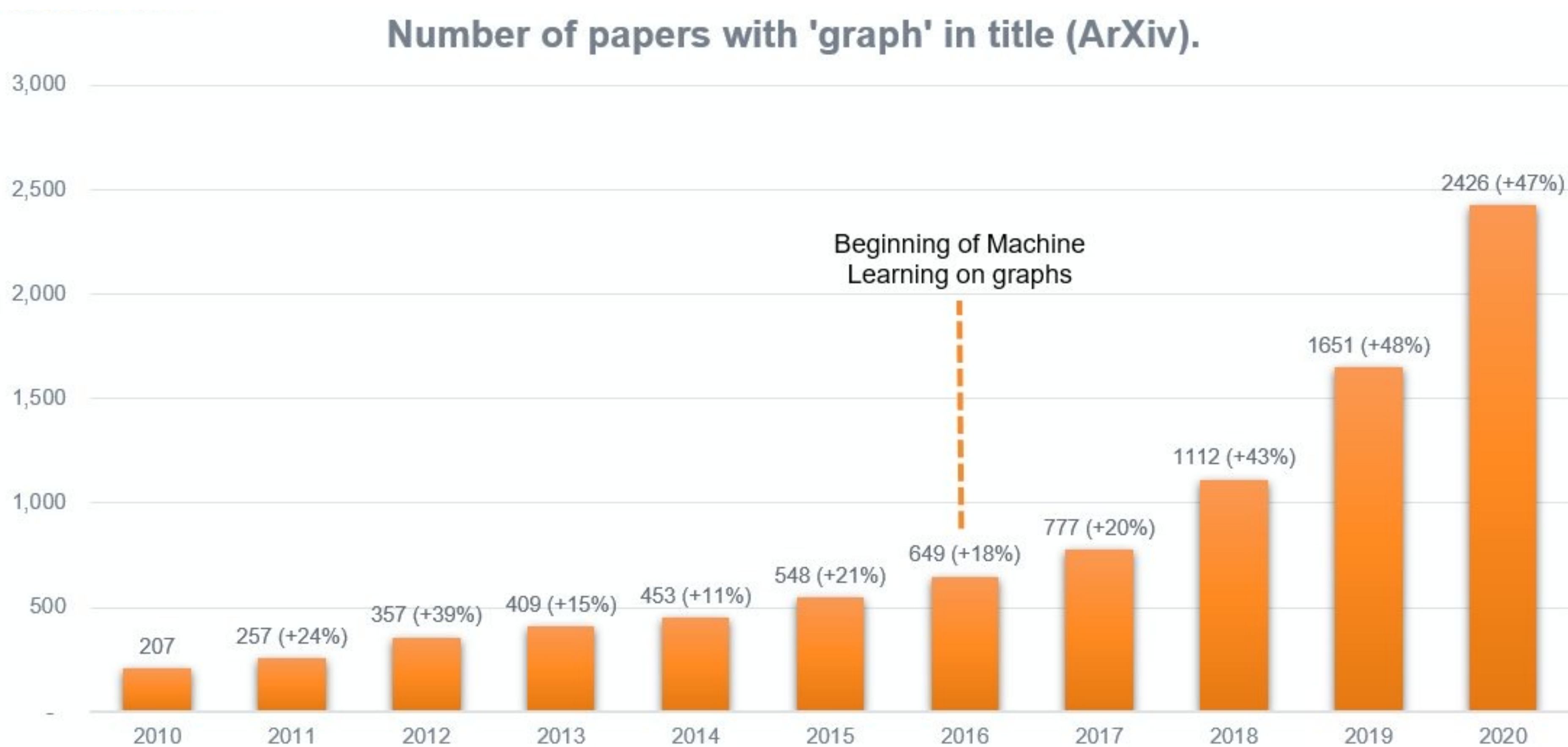
Groups



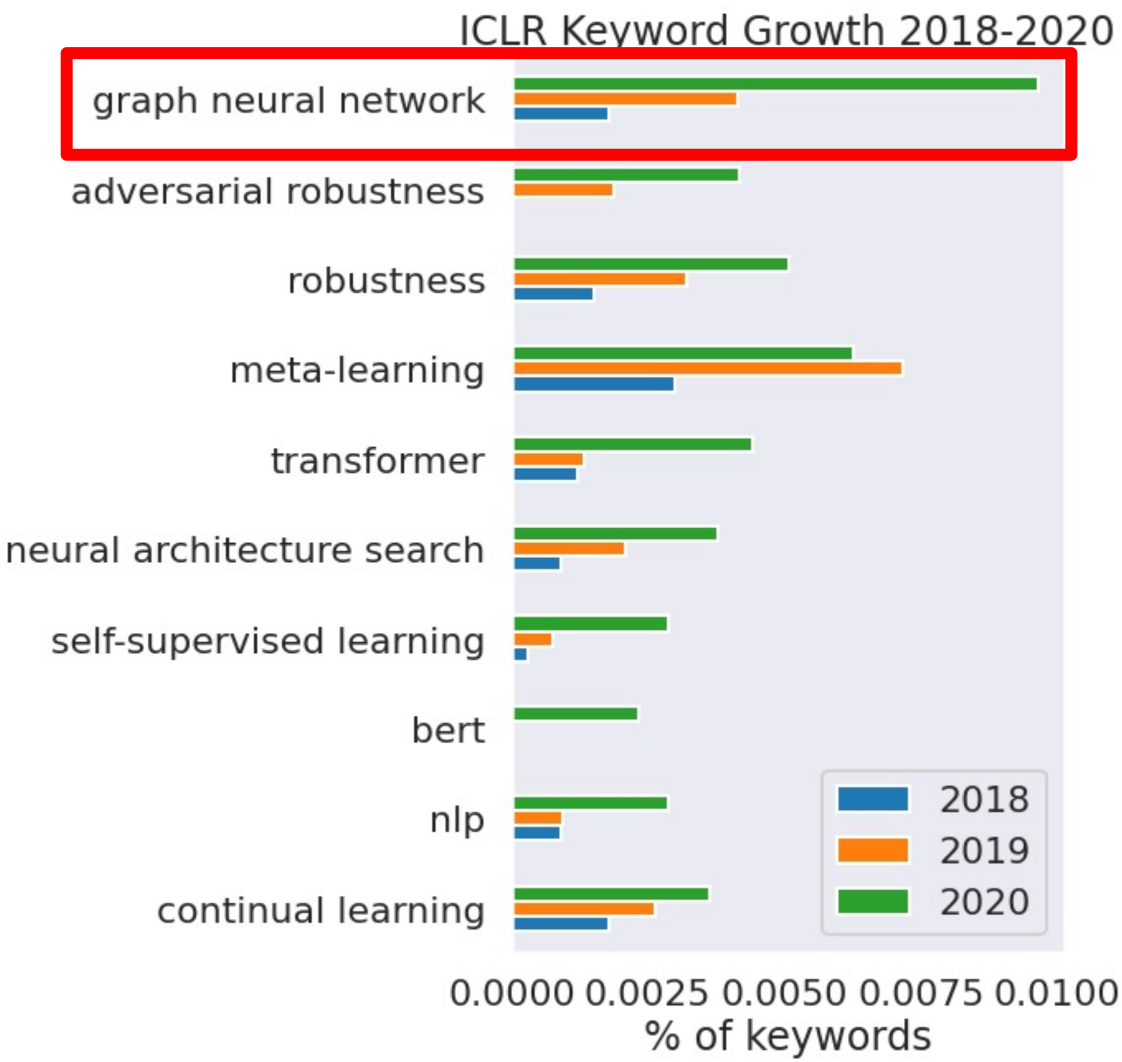
Graphs

Geodesics & Gauges

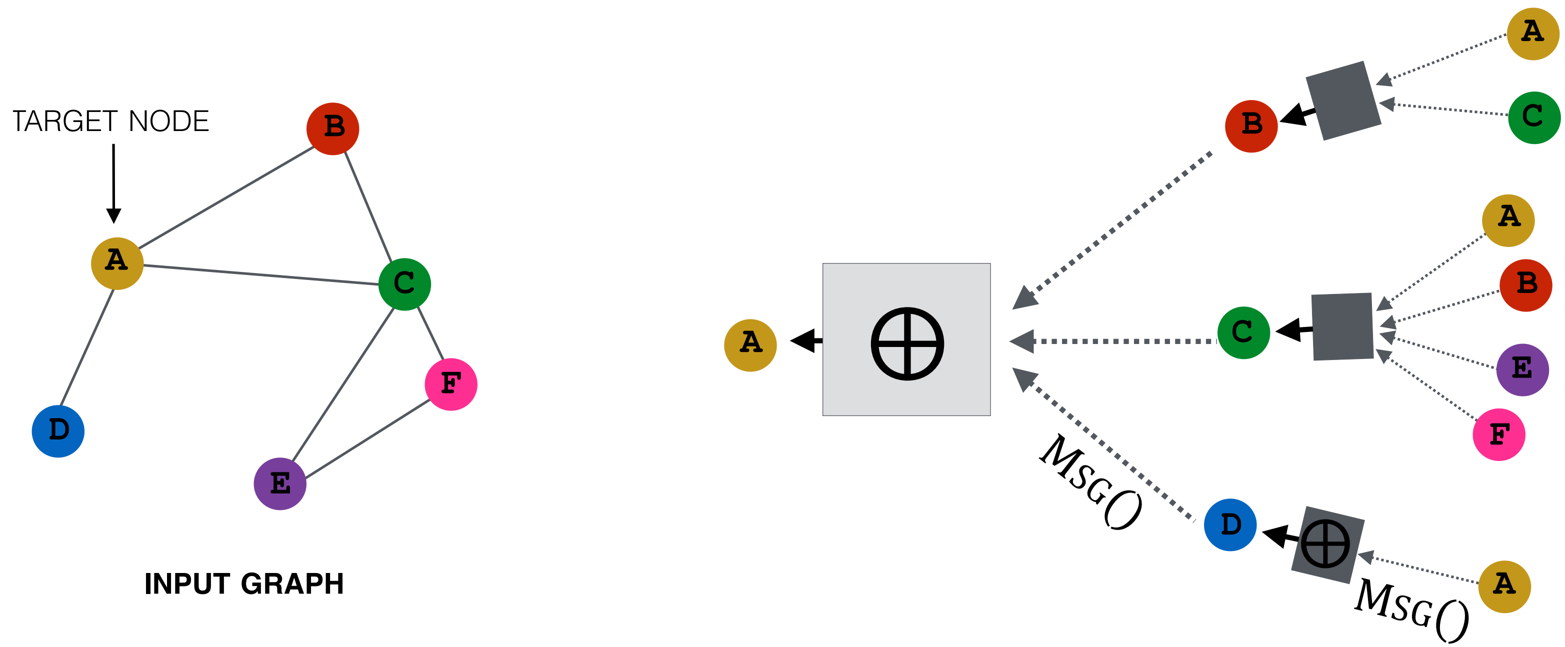
Graph ML is on Fire



Graph ML is on Fire



Graph Neural Network

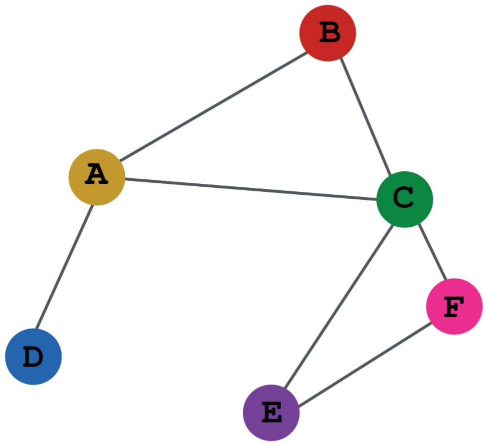


Each node defines a computation graph

- Each edge in this graph is a MSG function
- $\oplus$ is a message aggregation function

Scarselli et al. 2005. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*.

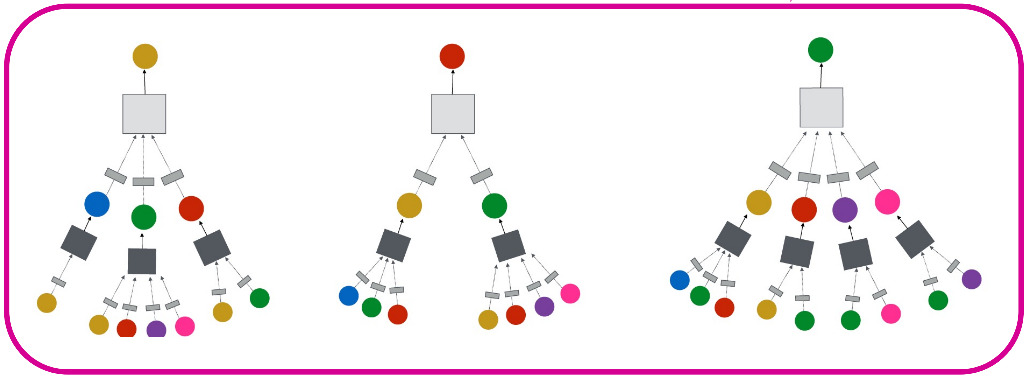
Inductive Capability

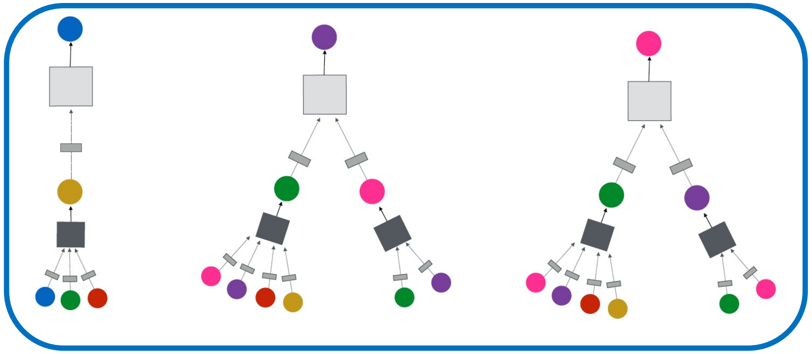


INPUT GRAPH

Train the model on a subset of nodes

Make predictions for nodes we never trained on!

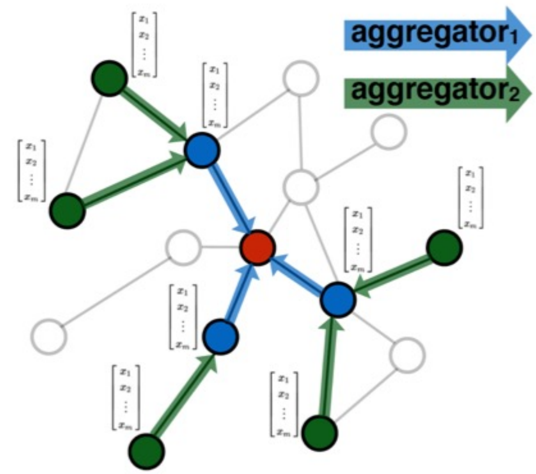




Nodes have different computation graphs

Two Views of GNNs

The GNN does two things:



- **1)** Powerful feature transformer/smoother
  - Node features get passed and transformed around node's L-hop neighborhood
- **2)** Each node can have a different computation graph and the network is also able to capture/learn its structure

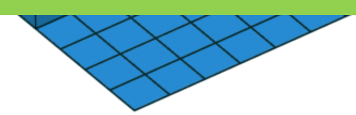
Key Benefits of GNNs

- GNNs adapt to the **shape** of data
  - Other Deep Learning architectures assume fixed input (matrix, sequence)
- GNNs can integrate multimodal data of various cardinalities and shapes

Key Benefits of GNNs

- GNNs subsume CNNs and Transformers as special cases

But scalability remains a challenge!



Image

Graph

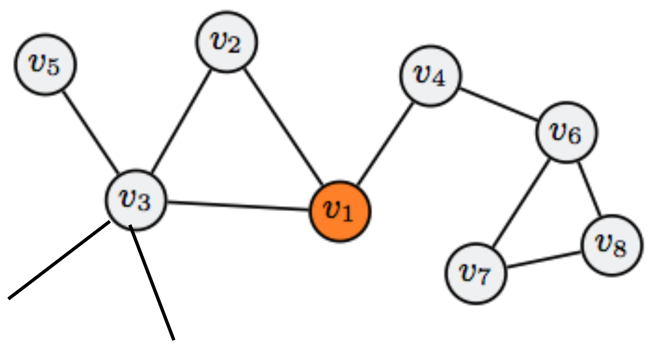
GNN formulation: $\mathrm{h}_v^{(l+1)} = \sigma(\mathbf{W}_l \sum_{u \in \mathrm{N}(v)} \frac{\mathrm{h}_u^{(l)}}{|\mathrm{N}(v)|} + \mathrm{B}_l \mathrm{h}_v^{(l)}), \forall l \in \{0, \dots, L-1\}$

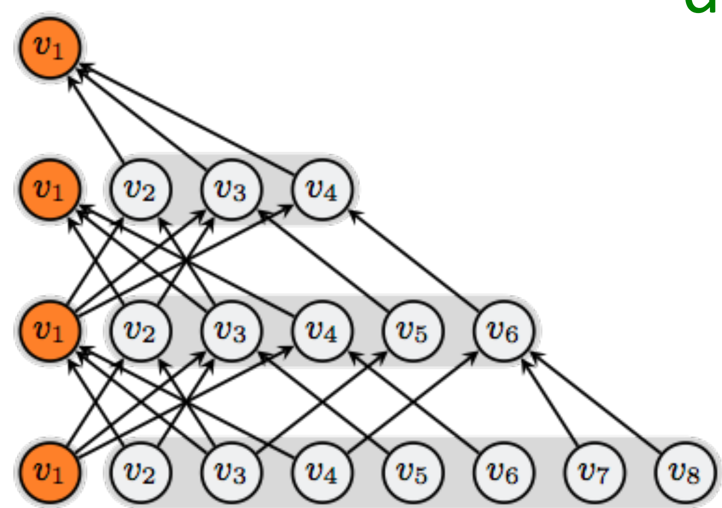
CNN formulation: $\mathrm{h}_v^{(l+1)} = \sigma(\sum_{u \in \mathrm{N}(v)} \mathbf{W}_l^u \mathrm{h}_u^{(l)} + \mathrm{B}_l \mathrm{h}_v^{(l)}), \forall l \in \{0, \dots, L-1\}$

Scalable & Deep GNNs

Deep GNNs on large-scale graphs result in neighbor explosions

- **Exponentially increasing dependency graph of nodes over layers**





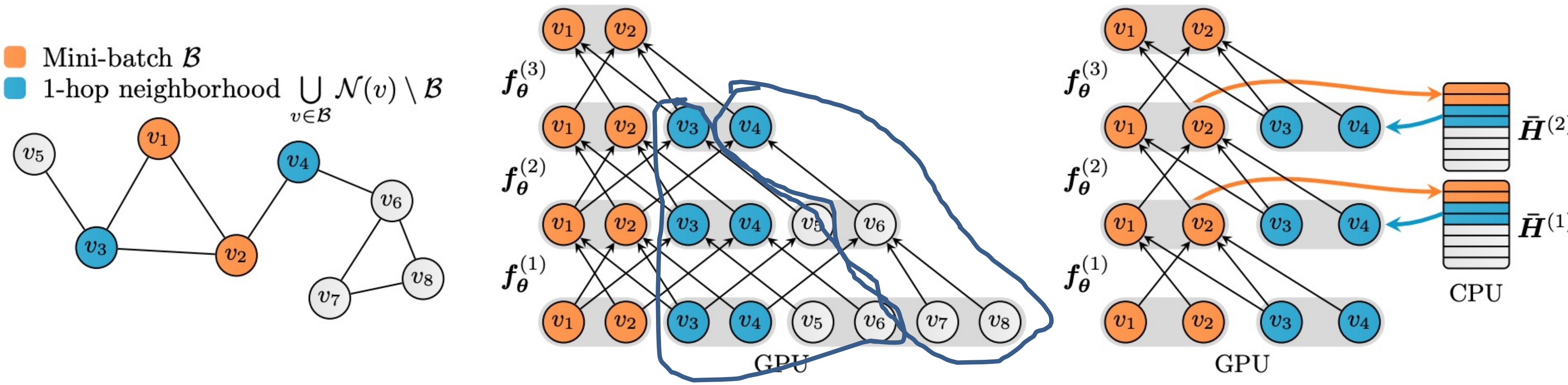
degree

d

d^2

d^3

Our Work: GNNAutoScale

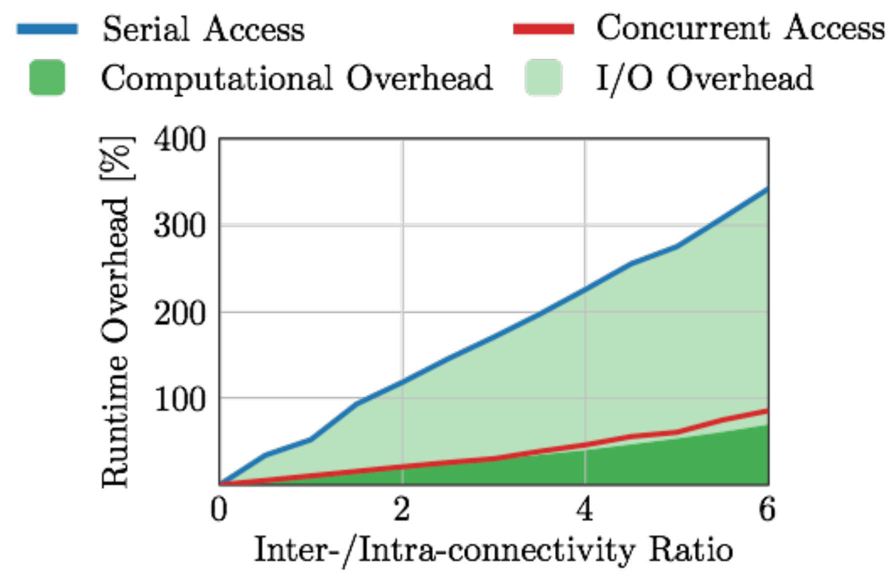


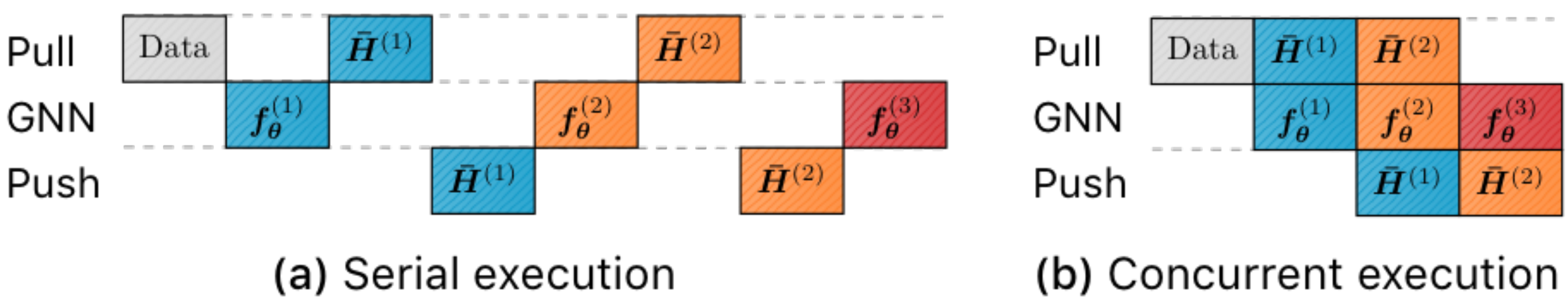
- Pull histories for out-of-minibatch nodes
- Push estimated embeddings to histories
- **Theory:**
  - (1) Bounded approximation error
  - (2) Provably as expressive as the WL-test

[GNNAutoScale: Scalable and Expressive Graph Neural Networks via Historical Embeddings](). ICML 2021.

GNNAutoScale

Fast history access via **asynchronous device transfers** to and from the GPU





Experimental Results

Dataset	Runtime (s) GTTF	Runtime (s) GAS	Memory (MB) GTTF	Memory (MB) GAS
CORA	0.077	**0.006**	18.01	**2.13**
PUBMED	0.071	**0.006**	28.79	**2.19**
PPI	0.976	**0.007**	134.86	**12.37**
FLICKR	1.178	**0.007**	325.97	**16.32**

	# nodes **# edges** **Method**	717K 7.9M YELP	169K 1.2M ogbn-arxiv	2.4M 61.9M ogbn-products
2-layer	Full-batch	6.64GB/100%	1.44GB/100%	21.96GB/100%
2-layer	GRAPHSAGE	0.76GB/ 9%	0.40GB/ 27%	0.92GB/ 2%
2-layer	CLUSTER-GCN	0.17GB/ 13%	0.15GB/ 40%	0.16GB/ 16%
2-layer	**GAS**	0.51GB/100%	0.22GB/100%	0.36GB/100%

- 10-100x speedup over GTTF [ICLR '21]
  - GTTF 10x better than naïve implementation
- 10-20x less memory over GTTF [ICLR '21]
  - GTTF 4x better than naïve implementation
- GNNAutoScale can be applied with any GNN architecture (GCN, SAGE, GAT)

PyG: GNN Library

PyG bundles the state-of-the-art in Graph Representation Learning

- 50+ GNN architectures
- 200+ benchmark datasets
- Extendable via a message passing interface
- Dedicated sparsity-aware CUDA kernels
- Support for various scalability techniques
- Heterogeneous Graph Support
- GNN Design Space Exploration

PyG: GNN Library

PyG is PyTorch-on-the-rocks

- Keeps design principles close to vanilla PyTorch
- Fits nicely into the PyTorch ecosystem

```
from torch.nn import Conv2d

class CNN(torch.nn.Module):
    def __init__(self):
        self.conv1 = Conv2d(3, 64)
        self.conv2 = Conv2d(64, 64)

    def forward(self, input):
        h = self.conv1(input)
        h = h.relu()
        h = self.conv2(h)
        return h
```

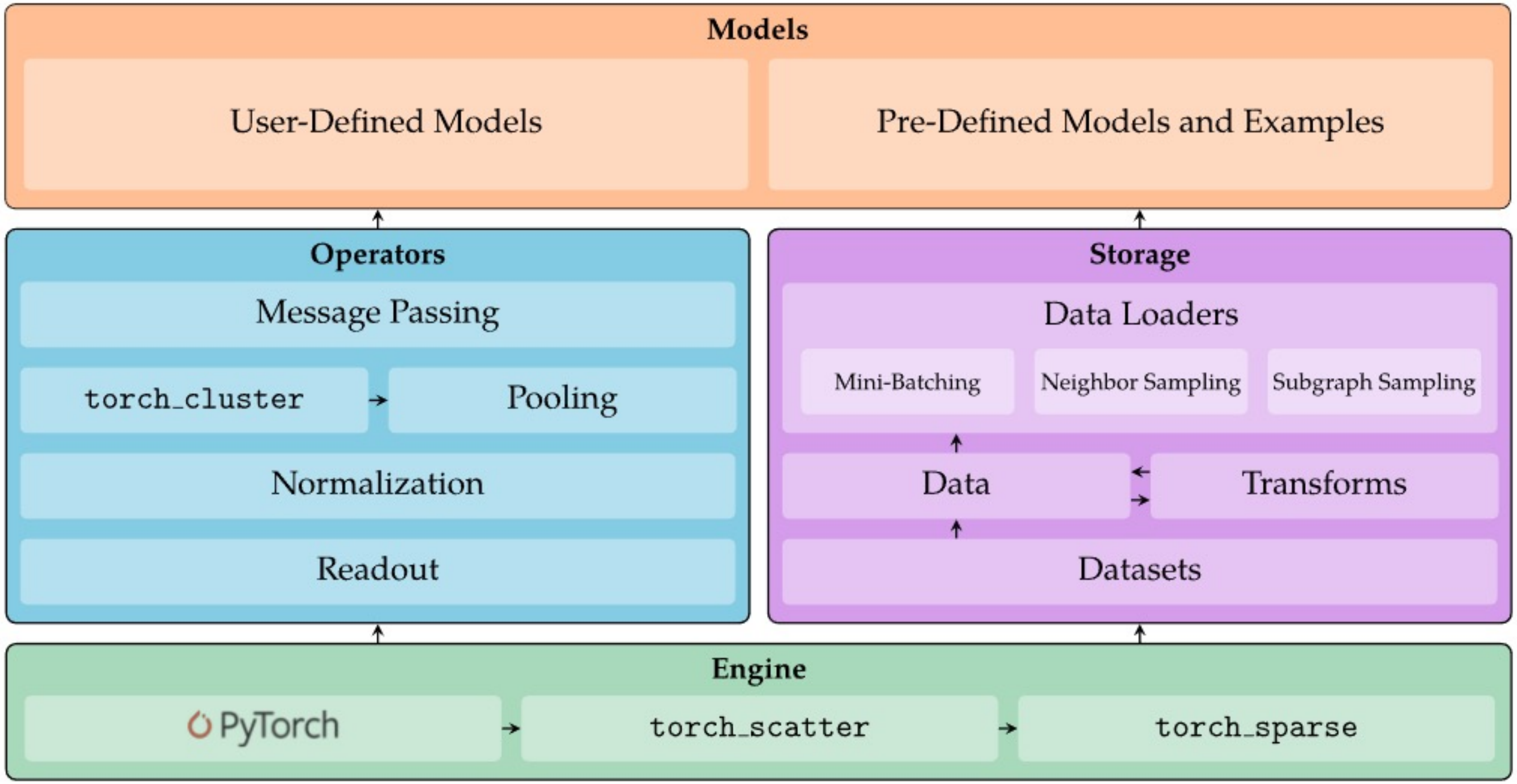
```
from torch_geometric.nn import GCNConv

class GNN(torch.nn.Module):
    def __init__(self):
        self.conv1 = GCNConv(3, 64)
        self.conv2 = GCNConv(64, 64)

    def forward(self, input, edge_index):
        h = self.conv1(input, edge_index)
        h = h.relu()
        h = self.conv2(h, edge_index)
        return h
```

PyG: GNN Library

- PyG provides the state-of-the-art in Graph Representation Learning



PyG: GNN Library

PYG.ORG:

- ~800 research papers written using PyG
- ~40K monthly downloads
- ~250 external contributors
- ~1800 members on Slack

MIT license

Cite this repository

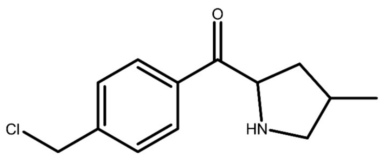
14.5k stars

241 watching

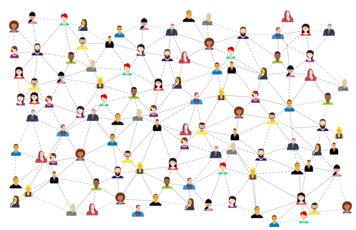
2.6k forks

What are some applications of PyG?

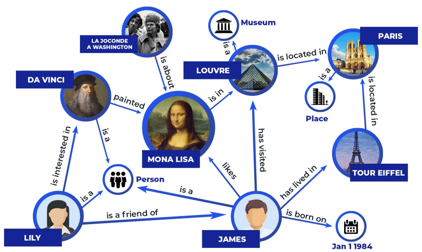
Graphs are Everywhere



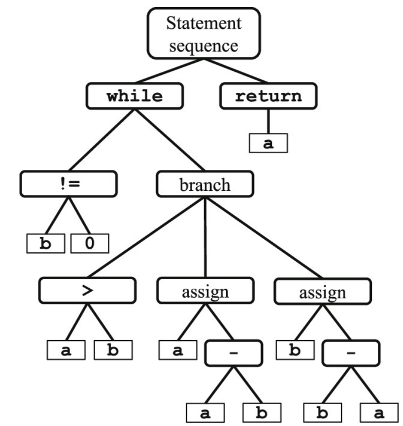
Molecular graphs



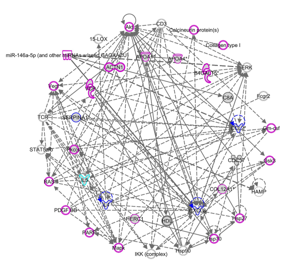
Social networks



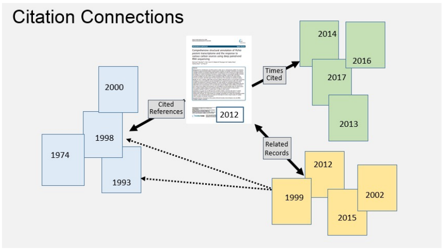
Knowledge graph



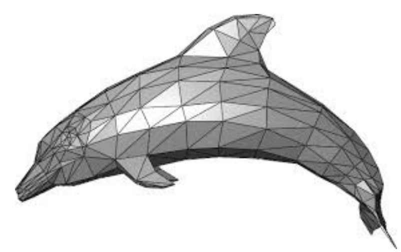
AST



Biological networks



Citation networks



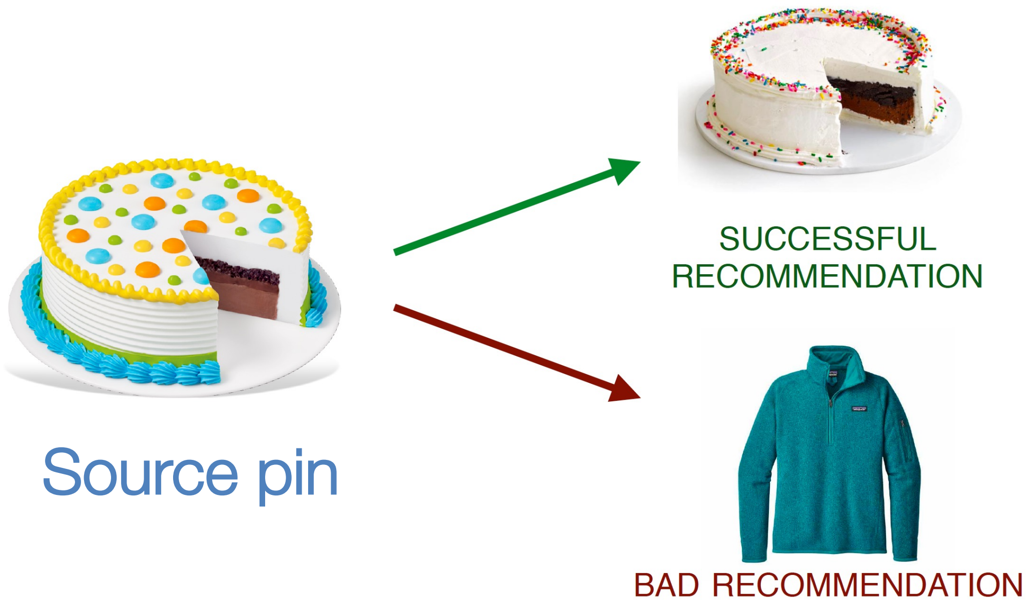
3D meshes

Real-world data is full of relations.

Graphs model how “stuff” is connected.

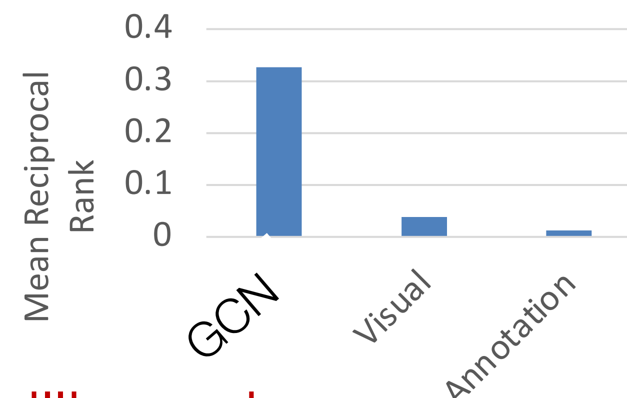
(1): Recommender Engines

Task: Recommend related pins



Task: Learn node embeddings z_i s.t.

$$d(z_{cake1}, z_{cake2}) < d(z_{cake1}, z_{sweater})$$



- **Challenges:**
  - Massive size: 3 billion nodes, 20 billion edges
  - Heterogeneous data: Rich image and text features

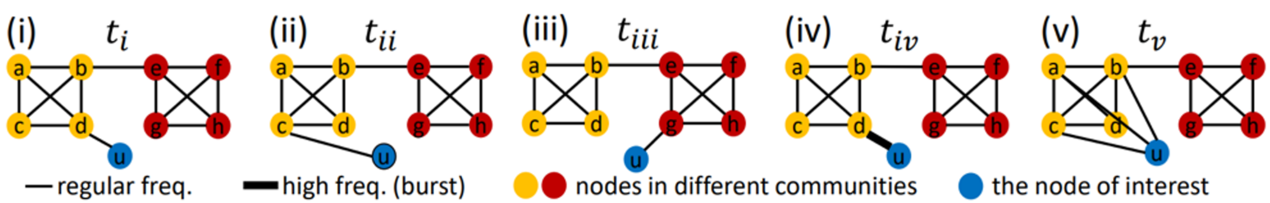
Graph Convolutional Neural Networks for Web-Scale Recommender Systems. R. Ying, et al. KDD, 2018.

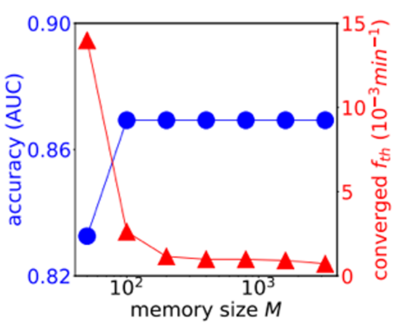
(2) Fraud & Intrusion Detection

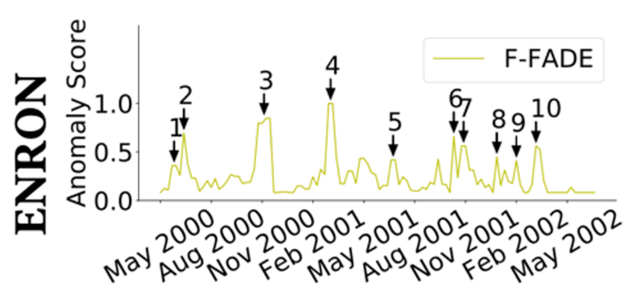
Fraud and intrusion detection in dynamic transaction graphs

Financial networks

Communication networks







(3) Text + Graph Reasoning

Question

If it is not used for **hair**, a **round brush** is an example of what?

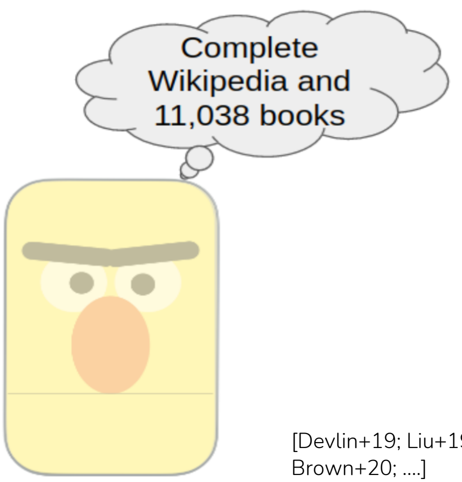
A. hair brush B. bathroom C. **art supplies*** D. shower

Knowledge sources

ConceptNet

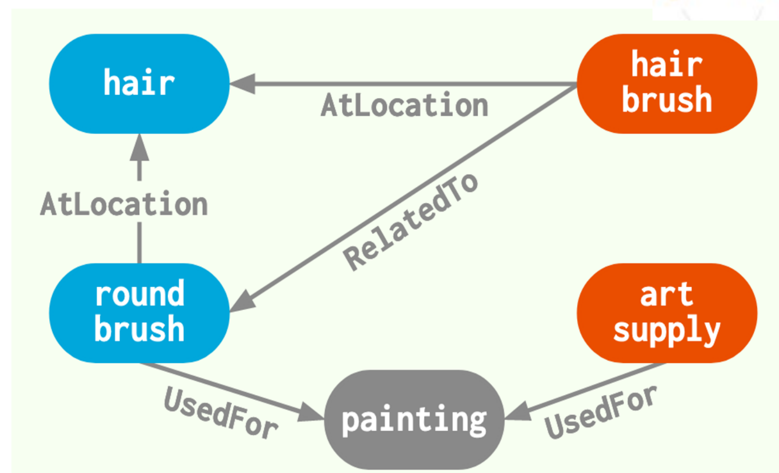
Freebase

Pre-trained language model (LM)



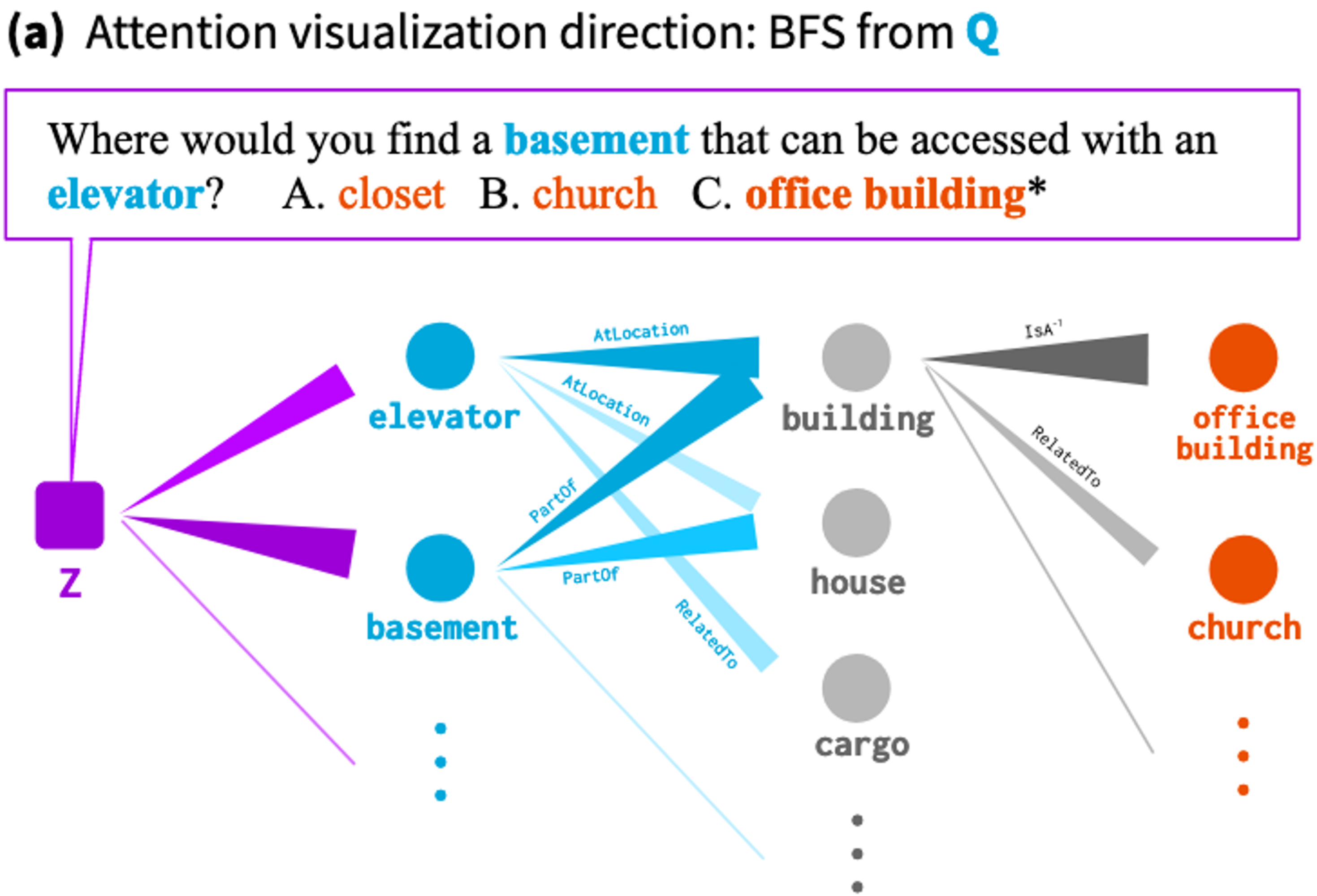
[Devlin+19; Liu+19; Brown+20; ....]

Knowledge Graph (KG)

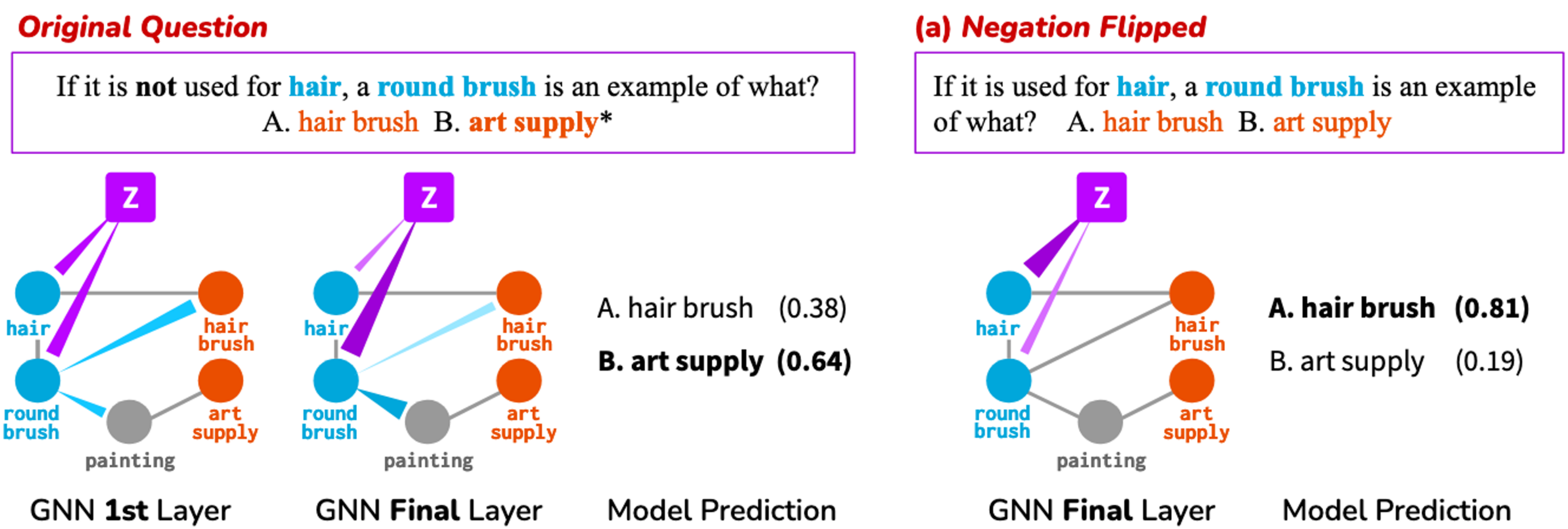


[Bollacker+08; Speer+16]

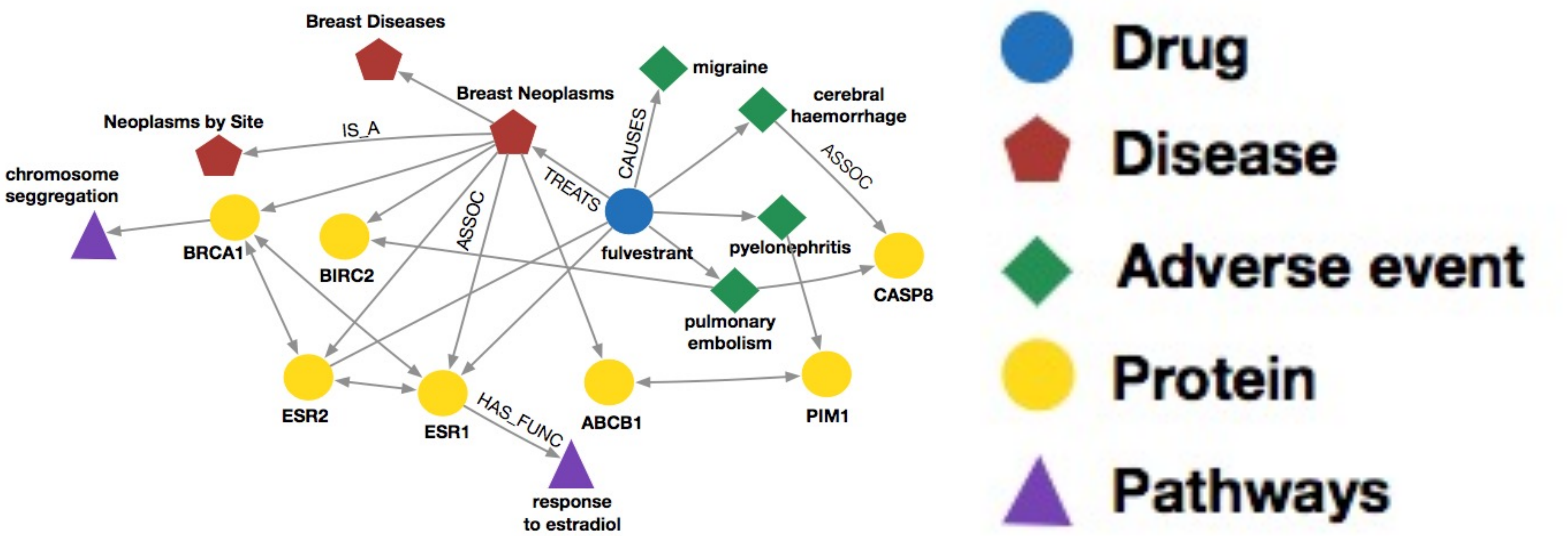
Interpretability



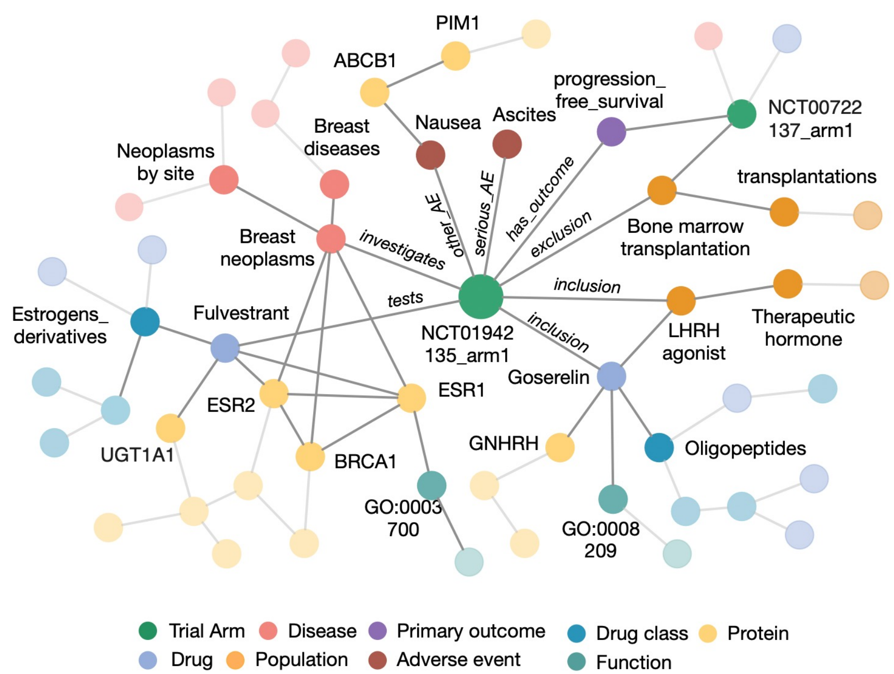
Structured Reasoning



(4) Data in Biomedicine



TrialNet Knowledge Graph

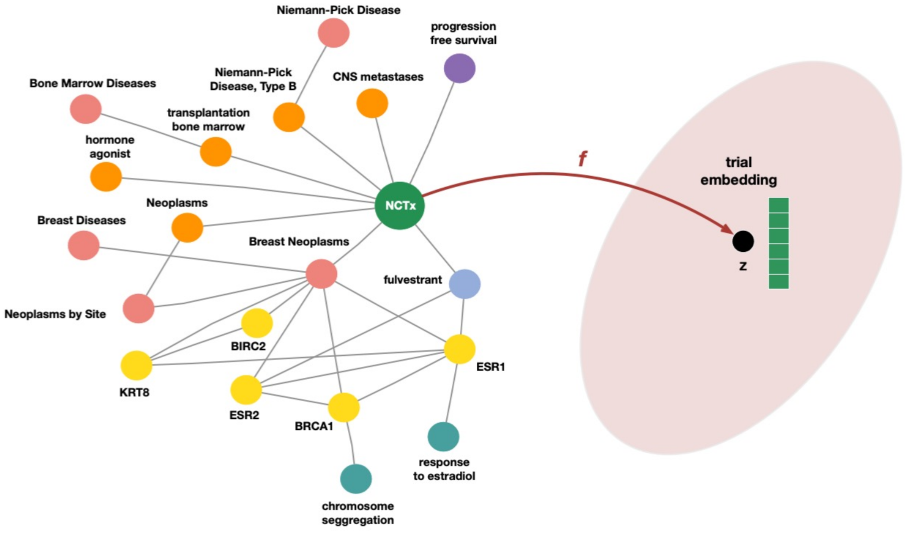


- Covers 70k interventional clinical trials
- 300k nodes (clinical trials protocol entities, biological and chemical entities)
- 10.8 millions edges across 18 relation types

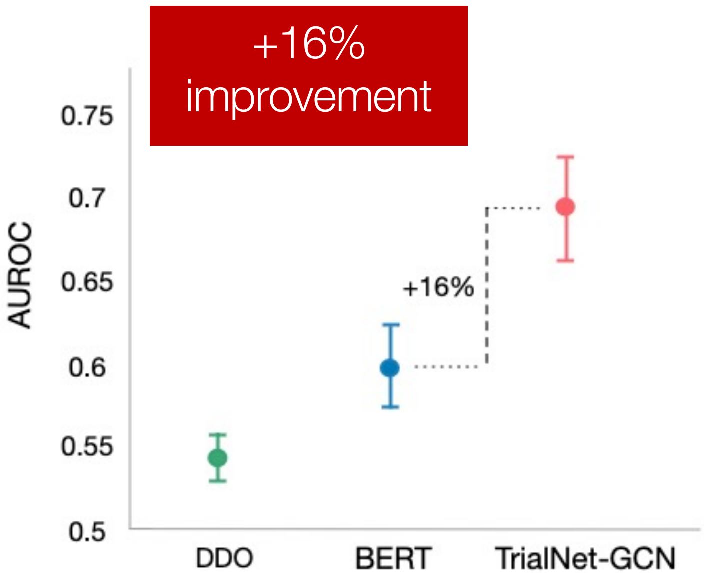
Learning Embeddings

How can we leverage TrialNet and build predictive models?

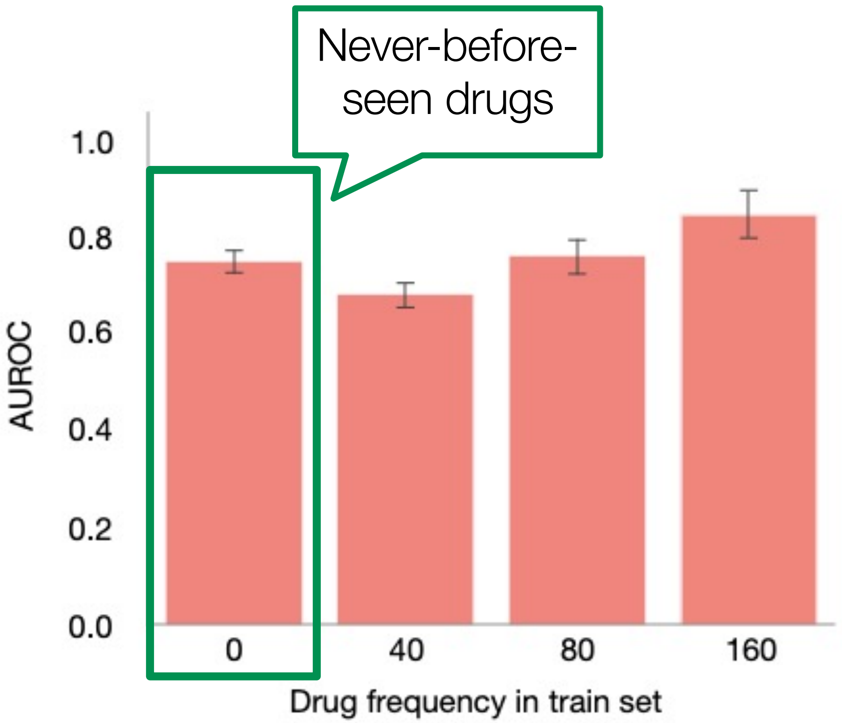
Key idea: Learn to embed trials, drugs, diseases, population, proteins…

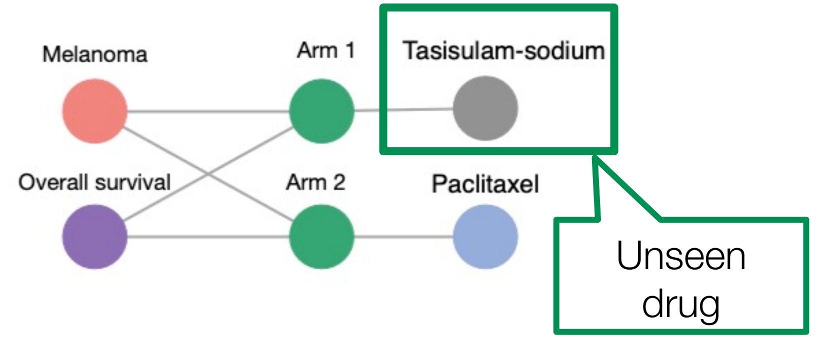


Predicting Drug Efficacy



Generalization to New Drugs





Which arm has better outcome??

Model prediction	Ground truth
Tasisulam: 0.25	Tasisulam: 6.8 months
Paclitaxel: 0.75	Paclitaxel: 9.4 months

Safety terminated study

(9) ML on Business data

Production ML

An on-call engineer's biggest nightmare

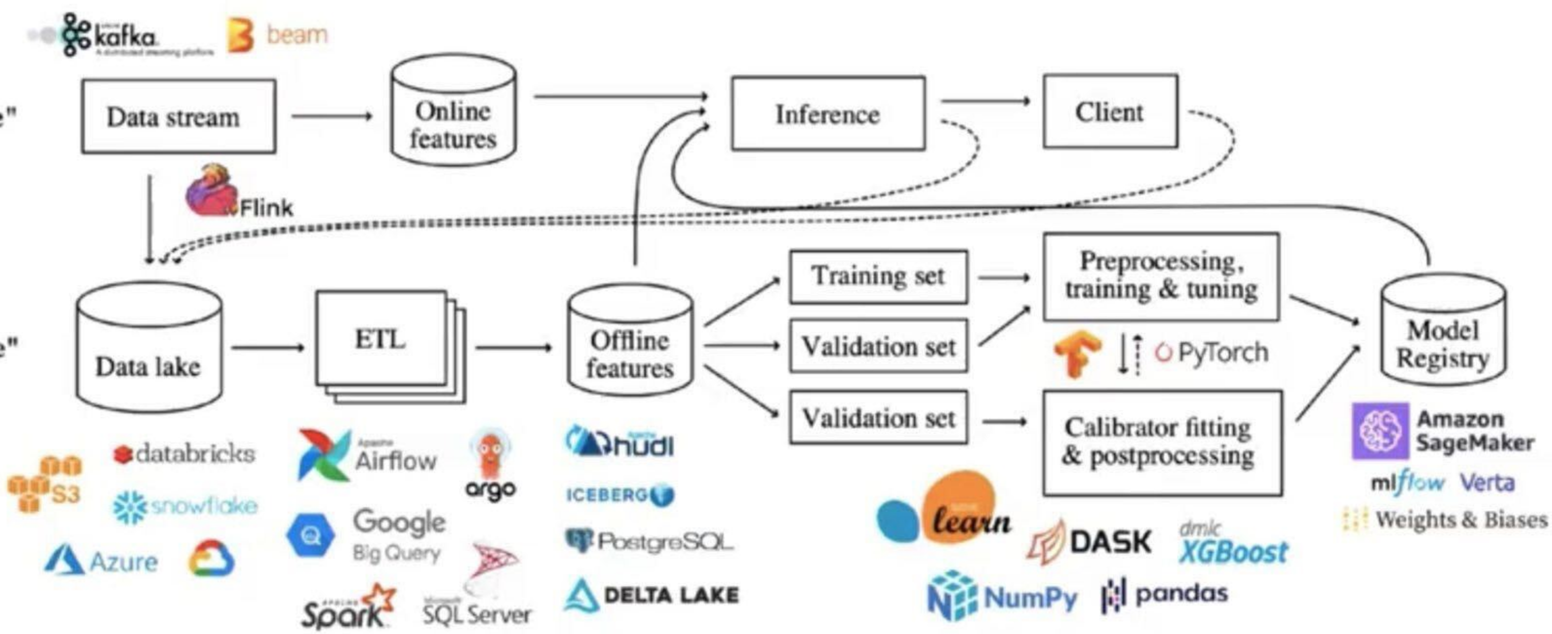
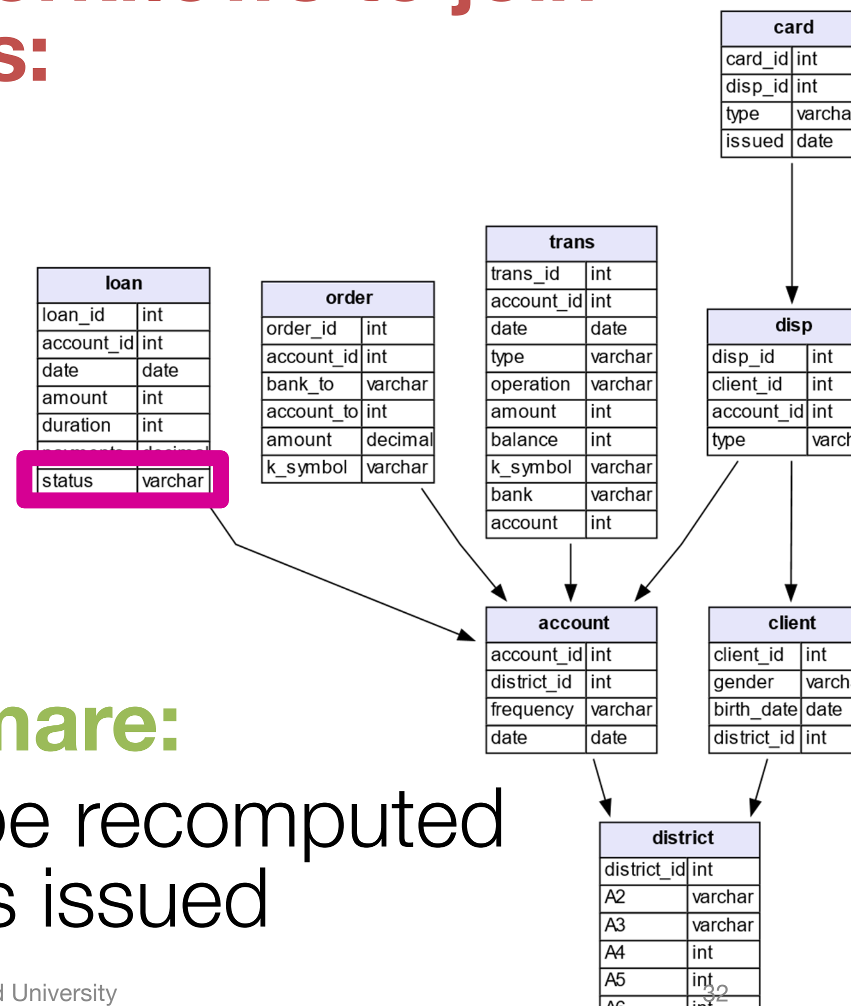


Figure 1: High-level architecture of a generic end-to-end machine learning pipeline. Logos represent a sample of tools used to construct components of the pipeline, illustrating heterogeneity in the tool stack. *Shankar et al. 2021*

Example: Predict Credit Risk

Data scientists writes workflows to join tables, generate features:

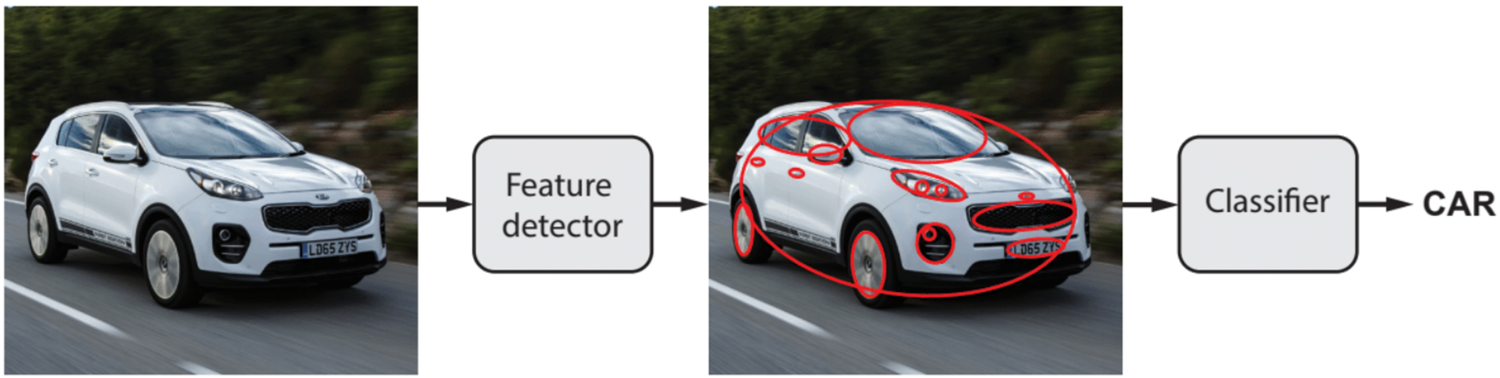
account age
avg. balance
total # transactions
cards
bad transactions
…



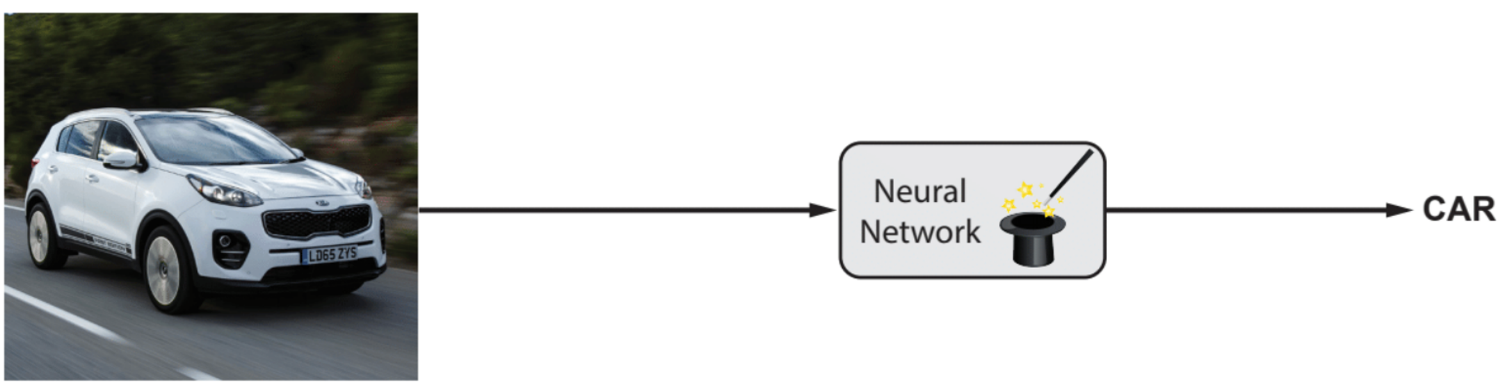
Time consistency nightmare:

Every feature needs to be recomputed for the time the loan was issued

Representation Learning

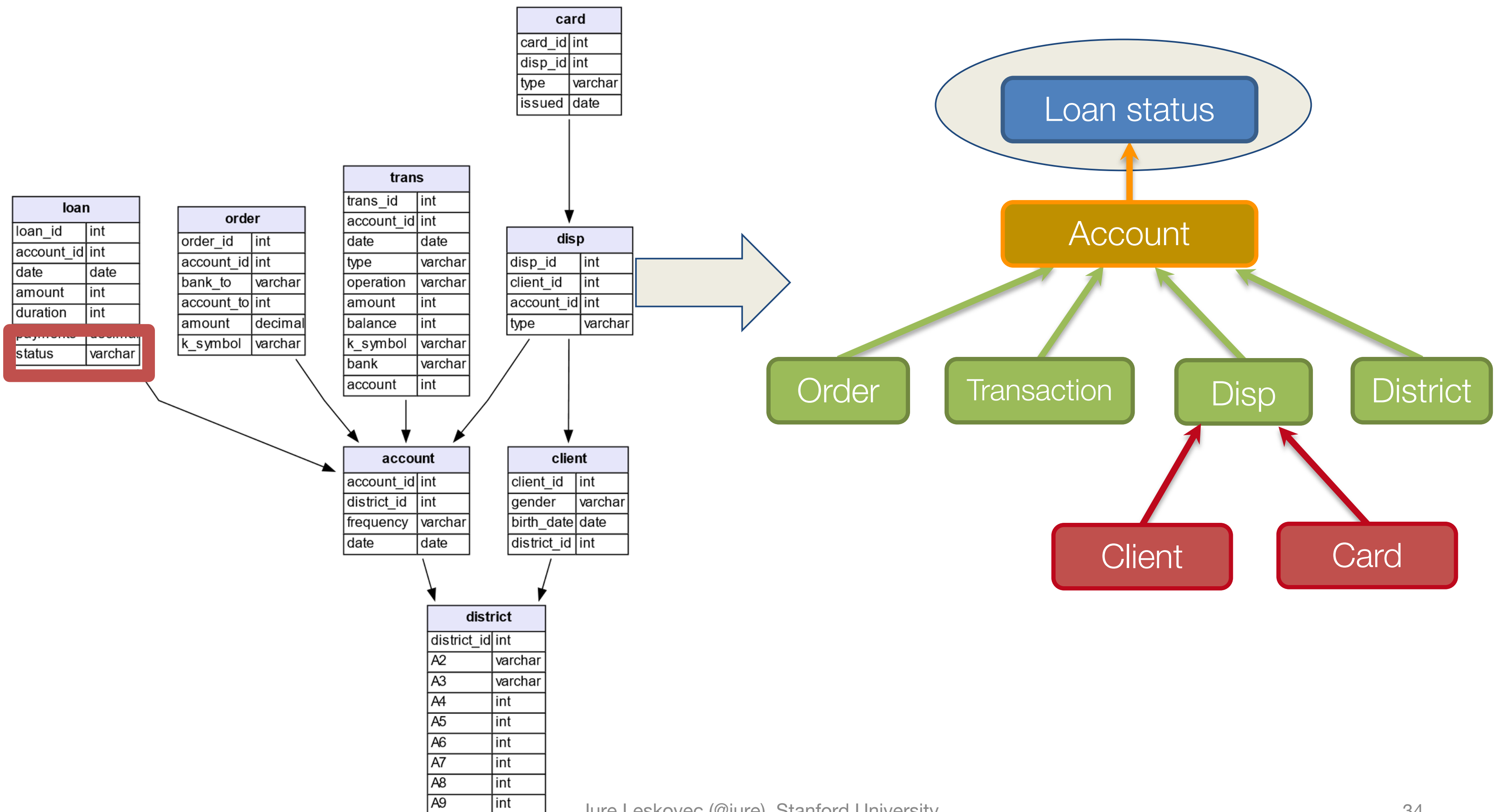


Classical computer vision: hand-crafted features (e.g. SIFT)
+ simple classifier (e.g. SVM)

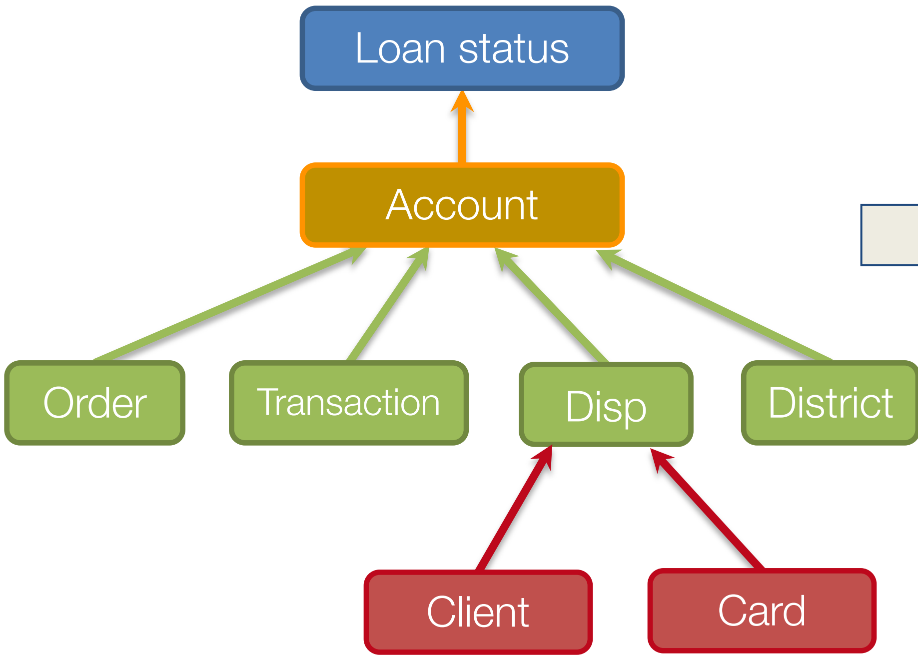


Modern computer vision: data-driven end-to-end systems

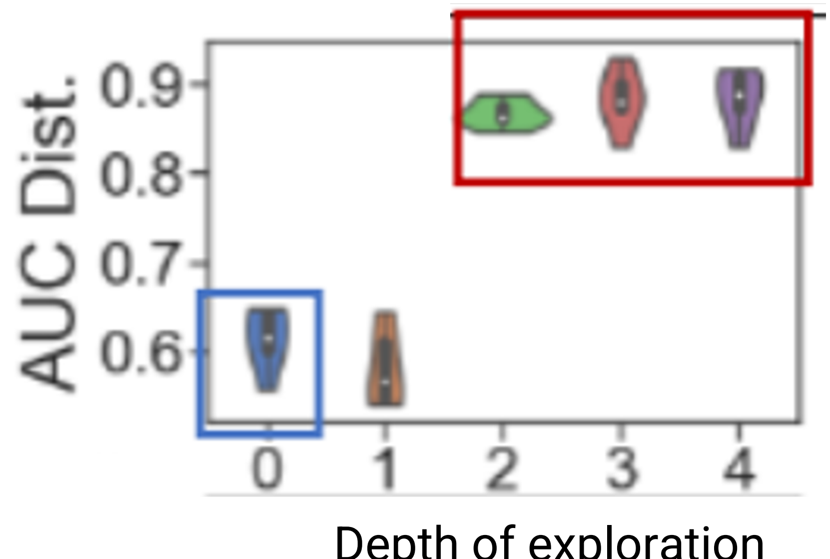
kumo: Example



kumo: State of the Art Results



Method	AUC
Naïve Neural Net baseline	0.65
Stanford PhD (XGBoost)	0.88
Best research paper	0.92
KUMO	**0.97**



kumo : Benefits

- Learned representation (features) is optimal for a given task
  - State of the art model performance
- Drastic simplification of the ML stack
  - No feature store, no backfills, no aggregations, no feature crosses
  - Easy productionisation
- Faster time to value
  - Amplifies data scientists to be more productive

Conclusion

GNNs allow for representation on complex relational data

- Fuse node features & relations
  - State-of-the-art accuracy graph machine learning tasks
- Model size independent of graph size; can scale to billions of nodes
  - Largest embedding to date (5B nodes, 20B edges)
- Leads to significant performance gains

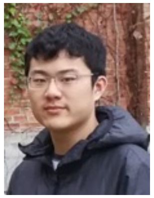
Conclusion

- GNNs are a very general type of NNs
- GNNs subsumes CNNs and Transformers as special cases
- GNNs have a huge range of applications

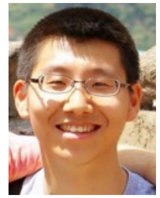
PhD Students


Alexandra Porter


Camilo Ruiz


Hongyu Ren

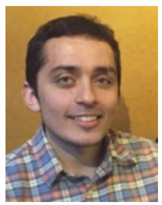

Hamed Nilforoshan

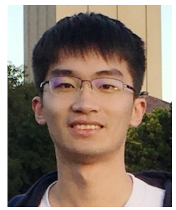

Michi Yasunaga


Jared Davis


Serina Chang


Weihua Hu


Yusuf Roohani

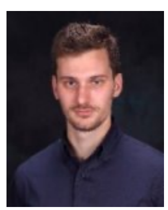

Kaidi Cao

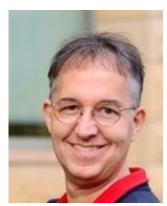
Post-Doctoral Fellows


Maria Brbic


Tailin Wu

Research Staff


Adrijan Bradaschia

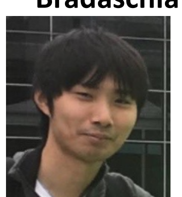

Rok Sosic

Industrial Visitors


Yanan Wang


Jing Huang


Yoshitaka Inoue


Shigeru Maya

Industry Partnerships

Pinterest, أرامكو السعودية saudi aramco, amazon, DELL, VISA, UnitedHealth Group, Wikipedia The Free Encyclopedia, TOSHIBA, BOEING, NVIDIA, gsk GlaxoSmithKline, intel, NTT docomo, Viaduct, JD.COM, HITACHI, KDDI, CHASE

Funding

ONR Office of Naval Research Science & Technology, NIH, DARPA, MURI ARO, NSF, CHAN ZUCKERBERG INITIATIVE, IARPA

Learn more:

http://pyg.org

http://cs.stanford.edu/~jure

Contact: jure@cs.stanford.edu