Challenges Of Applying Graph Neural Networks

Bryan Perozzi

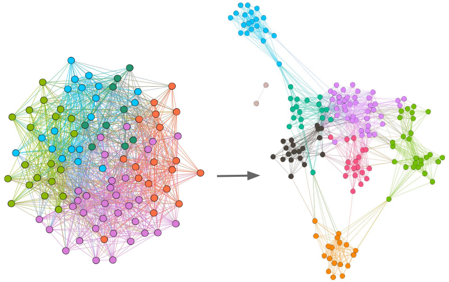
Graph Mining @ Google --
Algorithms, Learning, & Systems for Impact

Because graph representations are so flexible, we often want to use them on Google-scale data.

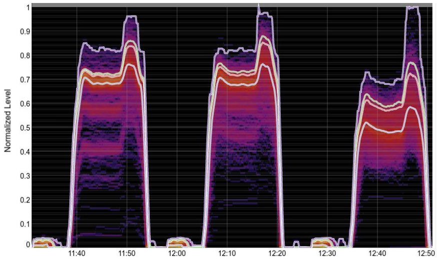
We are often dealing with billions of nodes and many more edges. To work with data at this scale, we have to combine algorithmic ideas with the right systems and ML models.

This can be very hard, and the devil is in details.

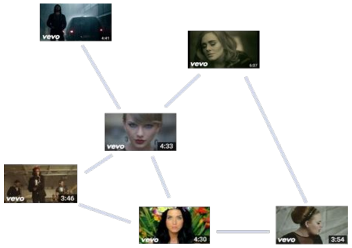
These tools power hundreds of projects at Google in Search, Ads, Youtube, Play, Cloud, Maps, Payments, and more.



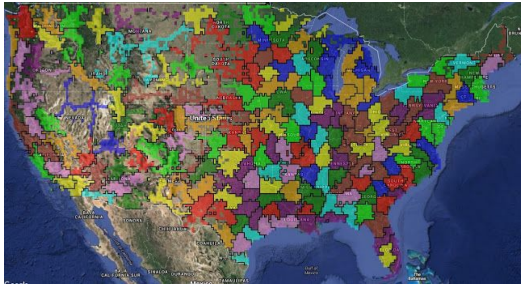
Same-meaning queries for Keyword matching systems



Better Caching for saving 32% Flash I/O for Search Infra(VLDB'19).

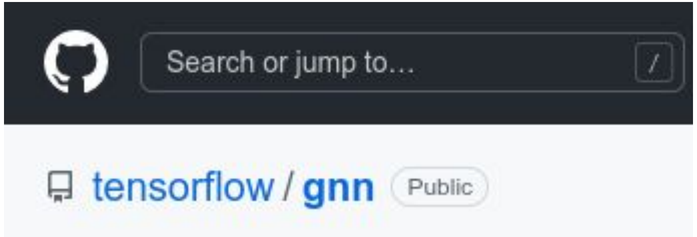


Collaborative Filtering for YouTube Recommendations



Finding micro-markets in designing A/B experiments [KDD'19, NeurIPS'19]

Tensorflow GNNs (tf-gnn)


Search or jump to...

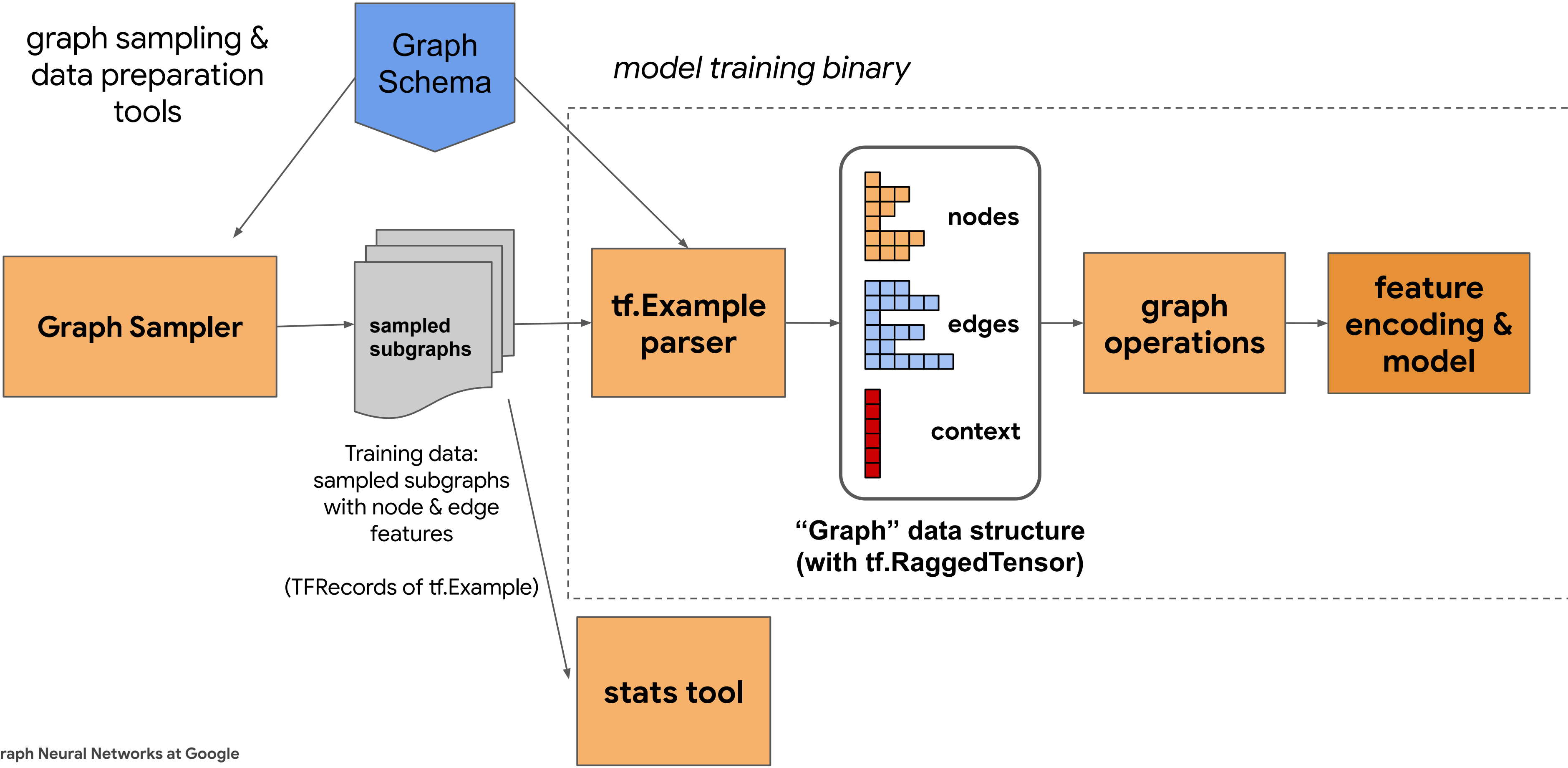
tensorflow / **gnn** Public

“Build the ultimate toolkit for building and training GNN models on very large graphs on top of TensorFlow.”

- Heterogeneous-first framework
- Supports many **model types**, **graph types**, arbitrary **feature shapes**
- **Scalable** and **distributed** by default
- Handles irregular representation, sampling and I/O out of the box
- Integrates well with **TensorFlow** **& Keras**

***Version** 0.2 on the way out soon!*

Library Overview



graph sampling & data preparation tools

Graph Schema

model training binary

Graph Sampler

sampled subgraphs

Training data: sampled subgraphs with node & edge features

(TFRecords of tf.Example)

tf.Example parser

nodes

edges

context

“Graph” data structure (with tf.RaggedTensor)

graph operations

feature encoding & model

stats tool

What challenges for GNNs are there?

Challenges I'll Talk About

1. What graph should I use?
2. Benchmarks that aren't Representative
3. Generalizing to unseen data

Challenge 1: What Graph Do We Use?

Grale: Designing Networks for Graph Learning

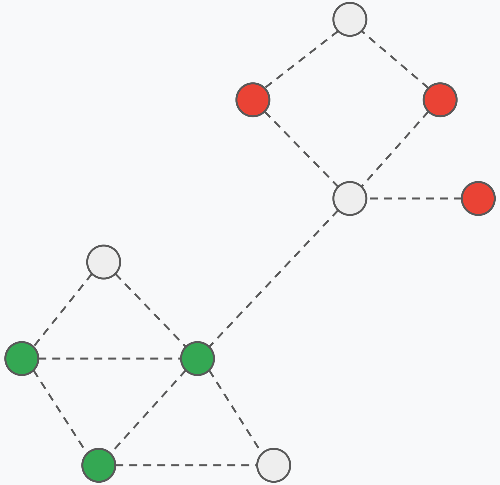
*Jonathan Halcrow, Alexandru Mosoi, Sam Ruth, Bryan Perozzi (**KDD'20**)*

Pathfinder Discovery Networks for Neural Message Passing

Benedek Rozemberczki, Peter Englert, Amol Kapoor, Martin Blais, Bryan Perozzi (**WWW'21**)

A Cartoon Example

In a toy example, we may be given a partially labeled set of nodes and a graph indicating some similarity relation on the nodes.

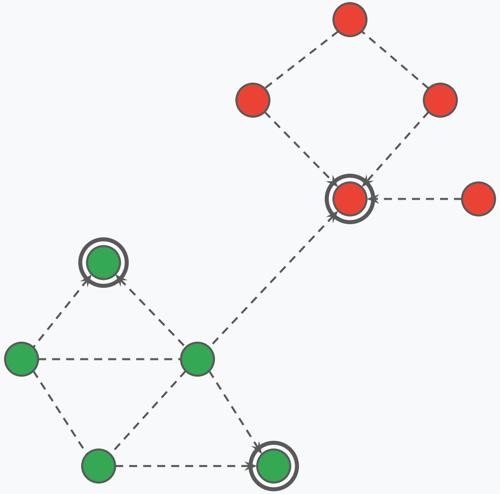


Google Research

A Cartoon Example

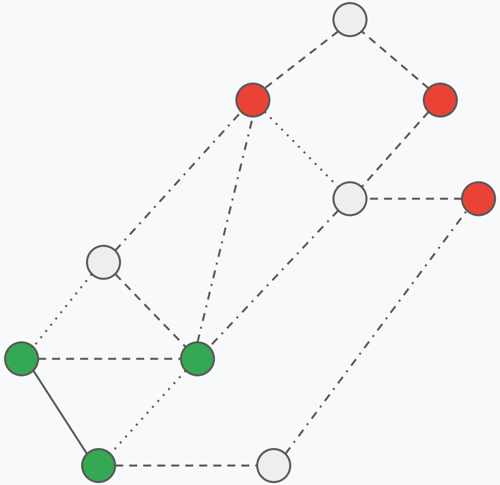
In a toy example, we may be given a partially labeled set of nodes and a graph indicating some similarity relation on the nodes.

We use the graph to infer labels for the unlabeled set, by spreading from the labeled nodes.



A "Real-World" Example

In real world examples, the picture is rarely this clear. Instead of a single set of relationships closely aligned with our target labels, we usually have many types of relationships to pick from, of varying quality.

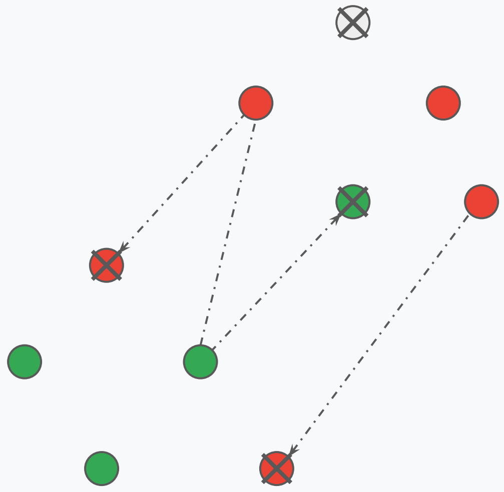


Google Research

A "Real-World" Example

In real world examples, the picture is rarely this clear. Instead of a single set of relationships closely aligned with our target labels, we usually have many types of relationships to pick from, of varying quality.

A bad choice of graph will yield a poorly performing graph learning algorithm.



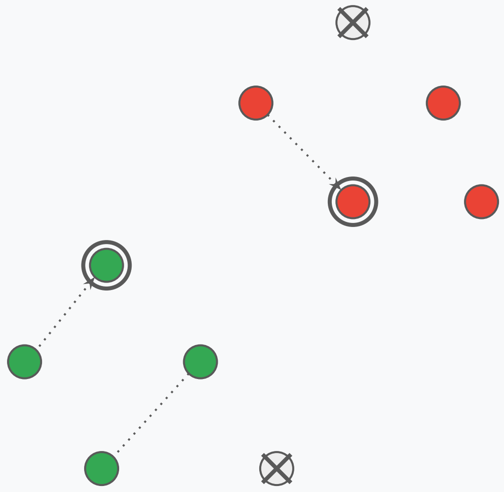
A Bad Choice

Google Research

A "Real-World" Example

In real world examples, the picture is rarely this clear. Instead of a single set of relationships closely aligned with our target labels, we usually have many types of relationships to pick from, of varying quality.

The choice of graph is critical for the performance of graph learning algorithms.



A Better Choice

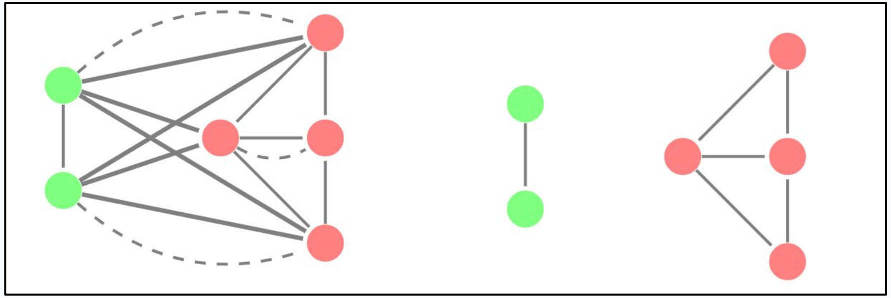
Google Research

The Graph Design Problem

Given:

- A multi-modal feature space $\mathbb{X}$, each mode with a natural distance measure, κ_i
- A partial labeling on this feature space
- A learning algorithm which is a function of some graph G having vertex set equal to the elements of $\mathbb{X}$

Find: An edge weighting function which allows us to construct a graph which optimizes the performance of the learning algorithm



Observed relationships vs. an ideal similarity measure

Google Research

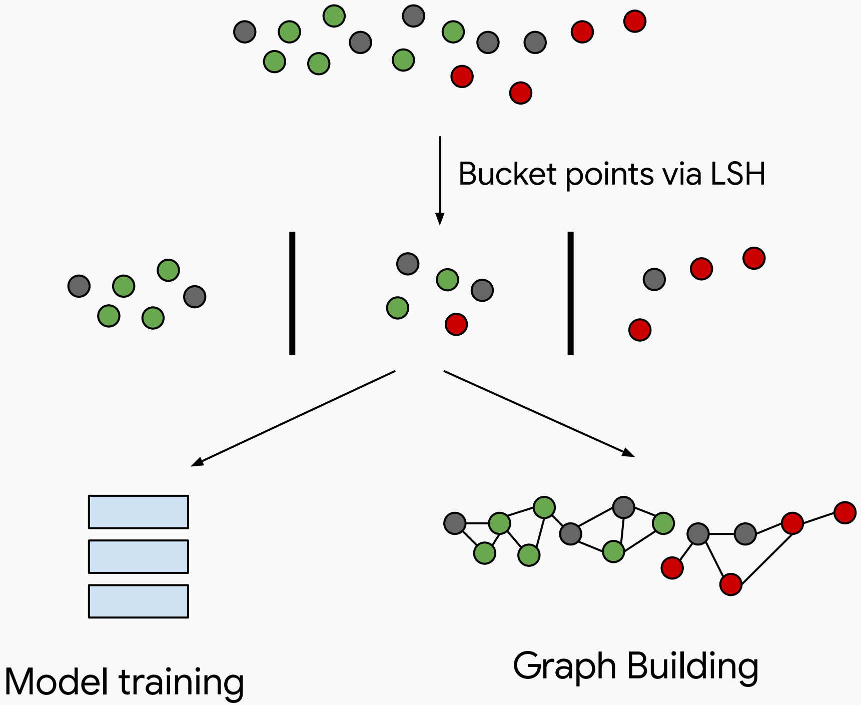
Grale: A Scalable Solution

Step 1:

Generate candidate pairs via locality sensitive hashing

Step 2:

Train a pairwise model to predict same class membership, or apply the model to infer similarity on pairs



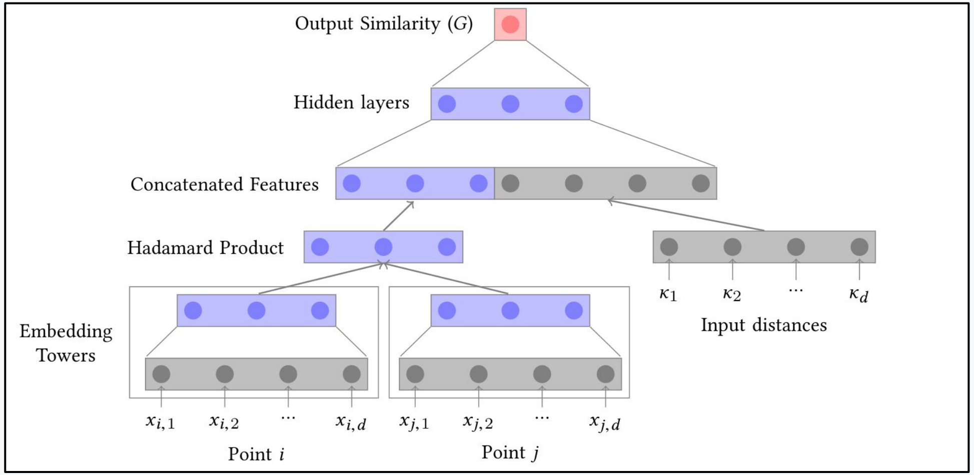
Google Research

Grale: Designing Networks for Graph Learning

Jonathan Halcrow, Alexandru Mosoi, Sam Ruth, Bryan Perozzi (***KDD'20***)

Model Structure

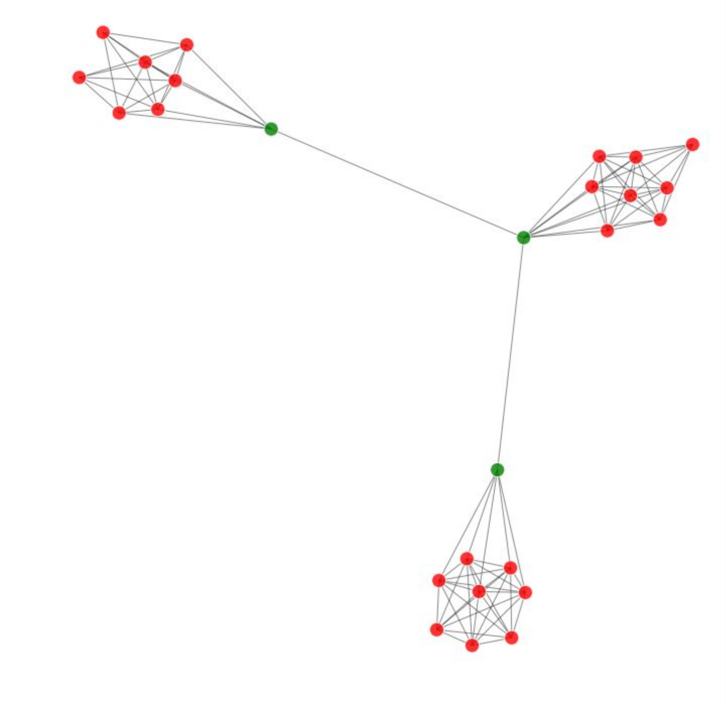
The specific choice of model structure may vary depending on the application, but most commonly we use a neural net which combines a two-tower structure to learn embeddings over the nodes and combines it with the 'natural' distances in the data.



Deployment for YouTube

Grale has been deployed in many different settings within Google. In particular it is used by YouTube to detect malicious actors.

We train the Grale model in this case to differentiate pairs of abusive items from pairs where at least one item is non-abusive.



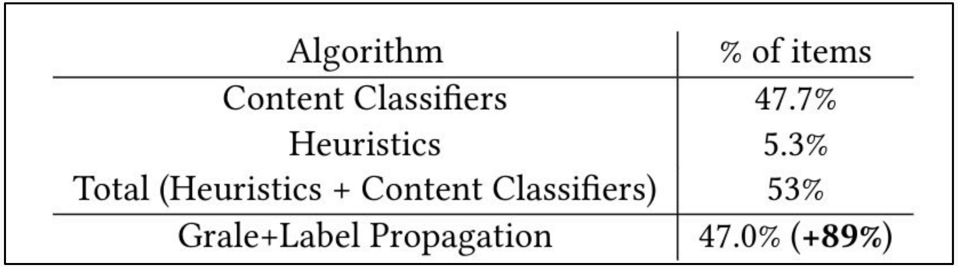
A subgraph of related items on YouTube found by Grale

Google Research

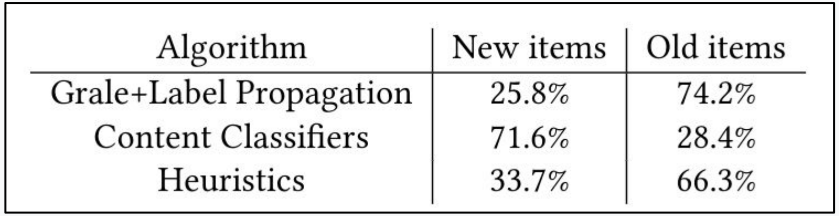
Comparison to other approaches

The Grale+Label Propagation system is deployed alongside various heuristics and content based classifiers.

For the type of items that we target here, we increase recall by 89% vs these other approaches alone. In particular we find many items that are missed by a first pass by purely content based classifiers.



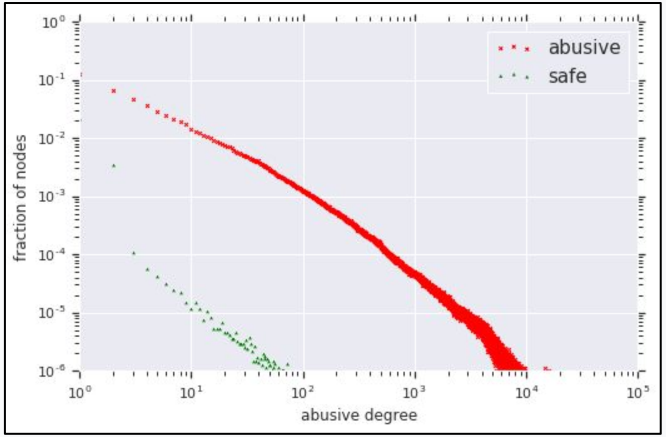
Algorithm	% of items
Content Classifiers	47.7%
Heuristics	5.3%
Total (Heuristics + Content Classifiers)	53%
Grale+Label Propagation	47.0% (**+89%**)

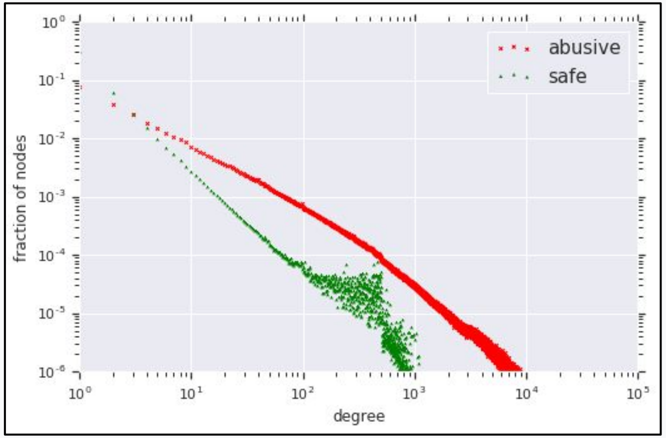


Algorithm	New items	Old items
Grale+Label Propagation	25.8%	74.2%
Content Classifiers	71.6%	28.4%
Heuristics	33.7%	66.3%

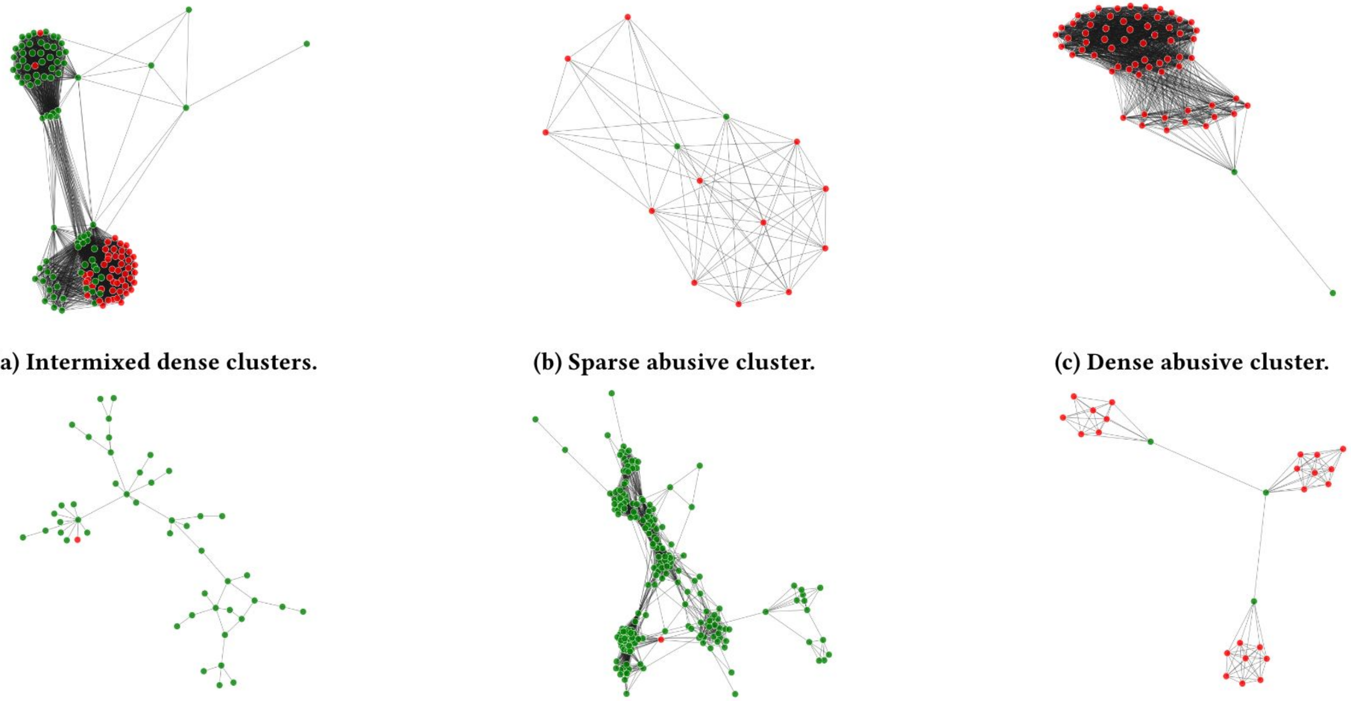
YouTube Graph Structure

Sorting the degree distribution in the graph by abuse status. We see that abusive nodes have much higher degree on average and are particularly strongly connected to other abusive nodes. This is precisely what we are hoping to achieve.





Example Clusters Found on YouTube



(a) Intermixed dense clusters.

(b) Sparse abusive cluster.

(c) Dense abusive cluster.

(d) Sparse non-abusive subgraph.

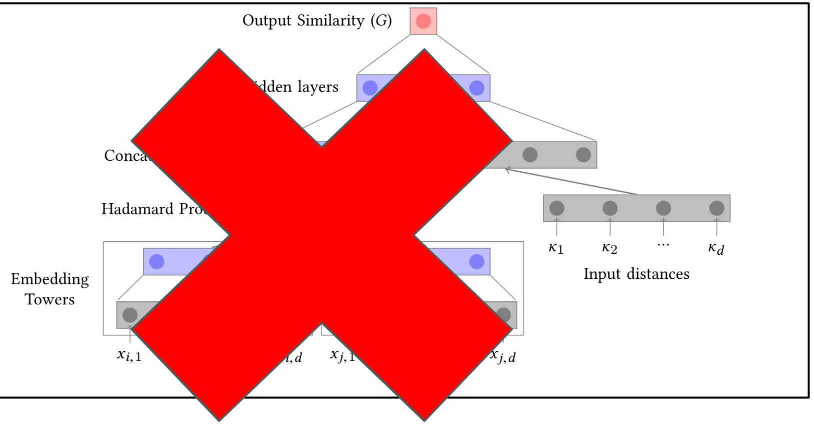
(e) Dense non-abusive clusters.

(f) Small abusive clusters.

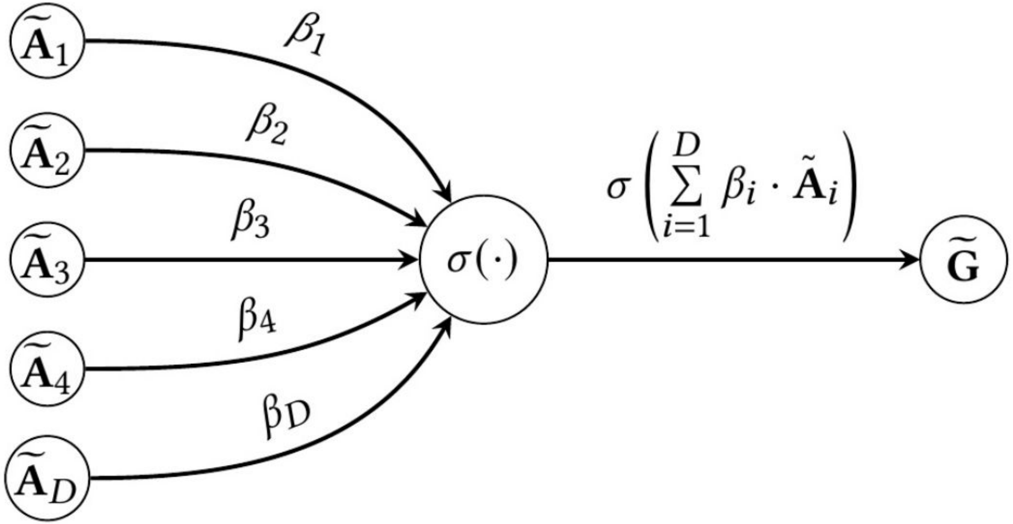
Grale: Designing Networks for Graph Learning

*Jonathan Halcrow, Alexandru Mosoi, Sam Ruth, Bryan Perozzi (**KDD'20**)*

A simple Grale Model

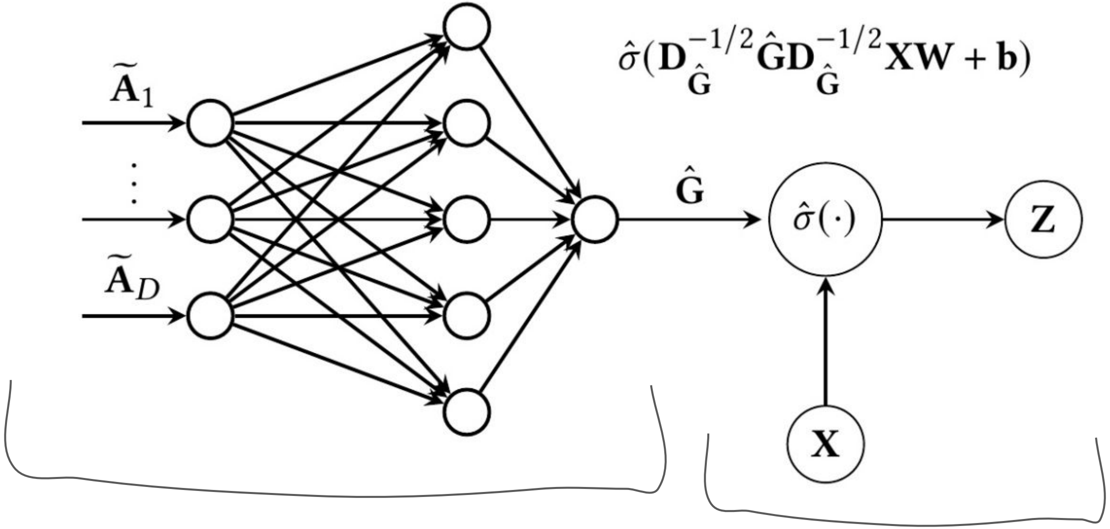


Remove the complicated parts



Keep a logit over similarity types

Pathfinder Discovery Networks


$\widetilde{\mathbf{A}}_1$
$\vdots$
$\widetilde{\mathbf{A}}_D$
$\hat{\sigma}(\mathbf{D}_{\hat{\mathbf{G}}}^{-1/2}\hat{\mathbf{G}}\mathbf{D}_{\hat{\mathbf{G}}}^{-1/2}\mathbf{X}\mathbf{W}+\mathbf{b})$
$\hat{\mathbf{G}}$
$\hat{\sigma}(\cdot)$
$\mathbf{Z}$
$\mathbf{X}$

Latent Graph
Representation Learning

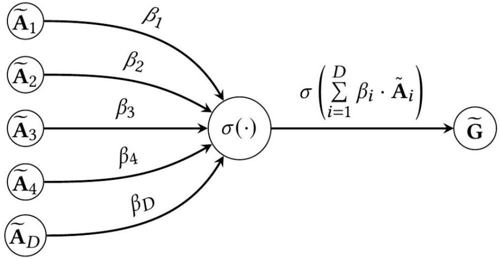
Normal GNN Model

Pathfinder Discovery Networks for Neural Message Passing

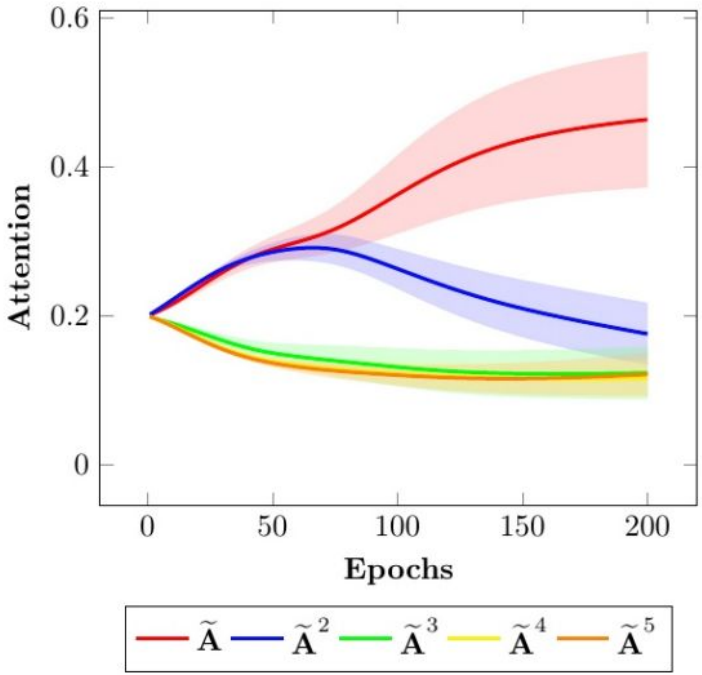
Benedek Rozemberczki, Peter Englert, Amol Kapoor, Martin Blais, Bryan Perozzi (**WWW'21**)

PDNs have Interpretability

The attention weights of shallow PDN models provide insight into what the model is doing.



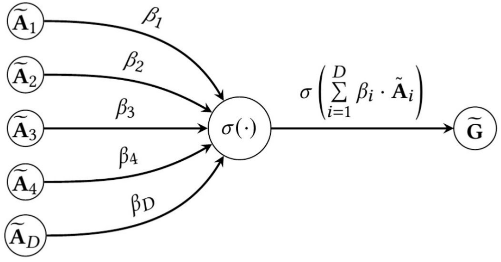
$\widetilde{\mathbf{A}}_1$
$\widetilde{\mathbf{A}}_2$
$\widetilde{\mathbf{A}}_3$
$\widetilde{\mathbf{A}}_4$
$\widetilde{\mathbf{A}}_D$
β_1
β_2
β_3
β_4
β_D
$\sigma(\cdot)$
$\sigma\left(\sum_{i=1}^{D} \beta_i \cdot \tilde{\mathbf{A}}_i\right)$
$\widetilde{\mathbf{G}}$

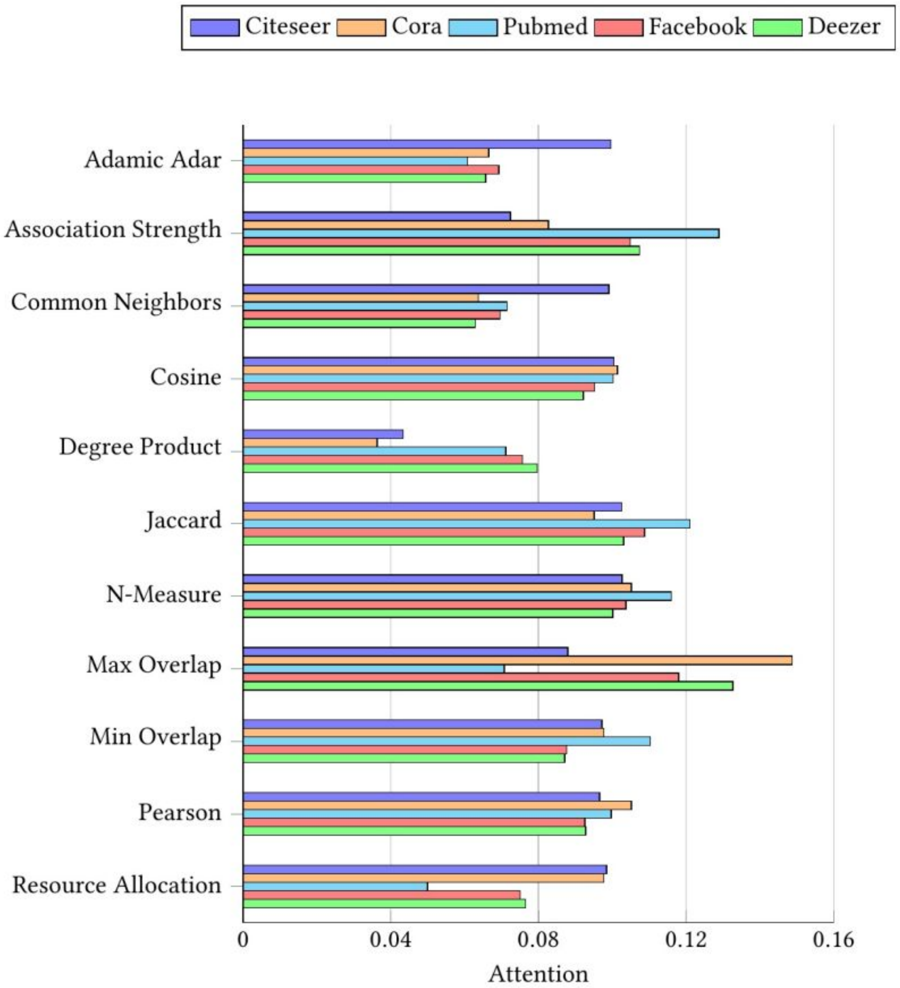


(a) Cora

PDNs have Interpretability

The attention weights of shallow PDN models provide insight into what the mo is doing.


$\widetilde{\mathbf{A}}_1$
$\widetilde{\mathbf{A}}_2$
$\widetilde{\mathbf{A}}_3$
$\widetilde{\mathbf{A}}_4$
$\widetilde{\mathbf{A}}_D$
β_1
β_2
β_3
β_4
β_D
$\sigma(\cdot)$
$\sigma\left(\sum_{i=1}^{D} \beta_i \cdot \tilde{\mathbf{A}}_i\right)$
$\widetilde{\mathbf{G}}$



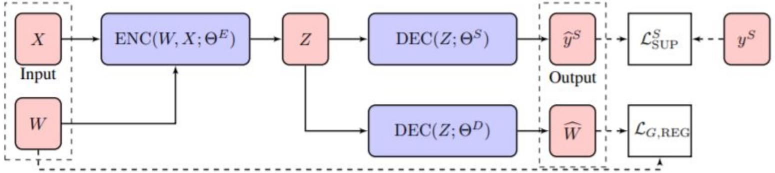
Google Research

What models do we use?

Taxonomy of GNNs

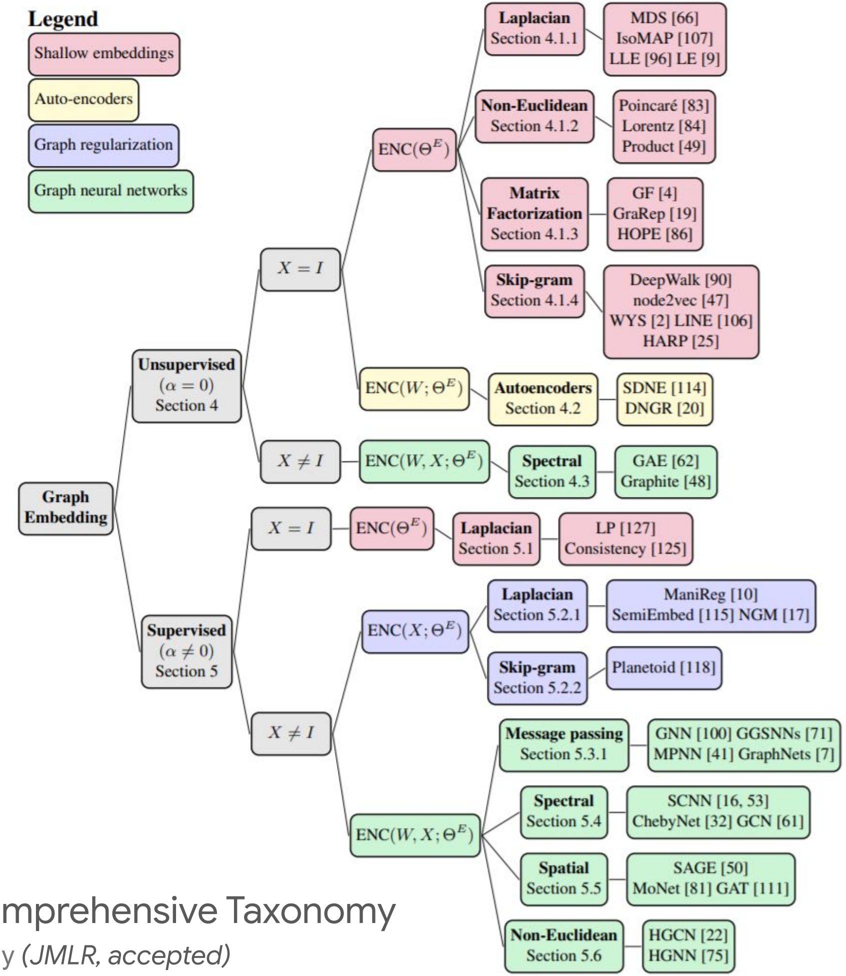
Increased interest in the area has led to an explosion of models for all kinds of graph data.

Thankfully many models share common elements, such as an encoder/decoder paradigm:



X
Input
W
$\mathrm{ENC}(W, X; \Theta^E)$
Z
$\mathrm{DEC}(Z; \Theta^S)$
$\mathrm{DEC}(Z; \Theta^D)$
$\widehat{y}^S$
Output
$\widehat{W}$
$\mathcal{L}^S_{\mathrm{SUP}}$
$\mathcal{L}_{G,\mathrm{REG}}$
y^S

The **GraphEDM** model.



Legend
Shallow embeddings
Auto-encoders
Graph regularization
Graph neural networks
Graph Embedding
Unsupervised ($\alpha = 0$) Section 4
Supervised ($\alpha \neq 0$) Section 5
$X = I$
$X \neq I$
$\mathrm{ENC}(\Theta^E)$
$\mathrm{ENC}(W; \Theta^E)$
$\mathrm{ENC}(W, X; \Theta^E)$
$\mathrm{ENC}(X; \Theta^E)$
Laplacian Section 4.1.1
MDS [66] IsoMAP [107] LLE [96] LE [9]
Non-Euclidean Section 4.1.2
Poincaré [83] Lorentz [84] Product [49]
Matrix Factorization Section 4.1.3
GF [4] GraRep [19] HOPE [86]
Skip-gram Section 4.1.4
DeepWalk [90] node2vec [47] WYS [2] LINE [106] HARP [25]
Autoencoders Section 4.2
SDNE [114] DNGR [20]
Spectral Section 4.3
GAE [62] Graphite [48]
Laplacian Section 5.1
LP [127] Consistency [125]
Laplacian Section 5.2.1
ManiReg [10] SemiEmbed [115] NGM [17]
Skip-gram Section 5.2.2
Planetoid [118]
Message passing Section 5.3.1
GNN [100] GGSNNs [71] MPNN [41] GraphNets [7]
Spectral Section 5.4
SCNN [16, 53] ChebyNet [32] GCN [61]
Spatial Section 5.5
SAGE [50] MoNet [81] GAT [111]
Non-Euclidean Section 5.6
HGCN [22] HGNN [75]

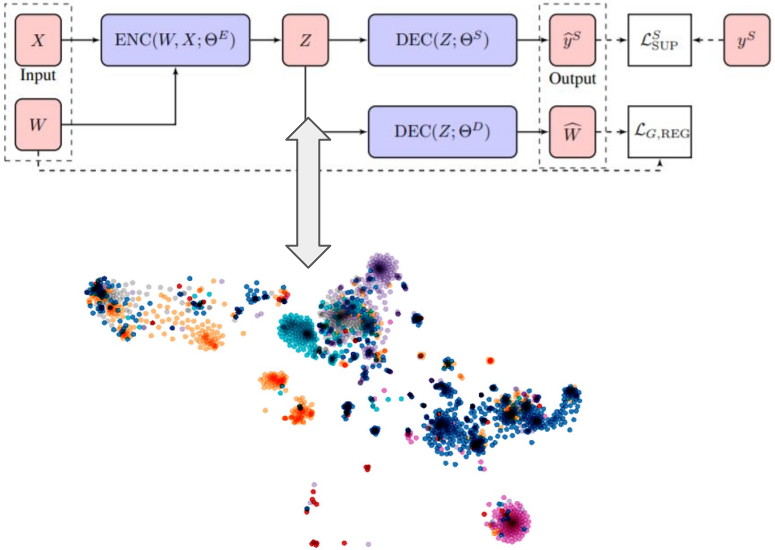
Machine Learning on Graphs: A Model and Comprehensive Taxonomy

I Chami, S Abu-El-Haija, B Perozzi, C Ré, K Murphy *(JMLR, accepted)*

Message passing allows flexibility

We typically use custom message passing operations (in the style of MPNNs) for our graph encoders (and decoders).

Peak task performance typically occurs from a bespoke architecture that best models the underlying phenomenon.


X
Input
W
$\text{ENC}(W, X; \Theta^E)$
Z
$\text{DEC}(Z; \Theta^S)$
$\text{DEC}(Z; \Theta^D)$
$\widehat{y}^S$
Output
$\widehat{W}$
$\mathcal{L}^S_{\text{SUP}}$
y^S
$\mathcal{L}_{G,\text{REG}}$

Neural Message Passing for Quantum Chemistry

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, George E. Dahl (ICML'17)

Challenge 2: Benchmarks aren’t Representative

GraphWorld: Fake Graphs Bring Real Insights for GNNs

John Palowitch, Anton Tsitsulin, Brandon Mayer, Bryan Perozzi (submitted)

https://arxiv.org/abs/2203.00112

Major GNN papers re-use the same benchmark datasets

You have run thousands of experiments on:

1. Cora
2. Citeseer
3. PubMed

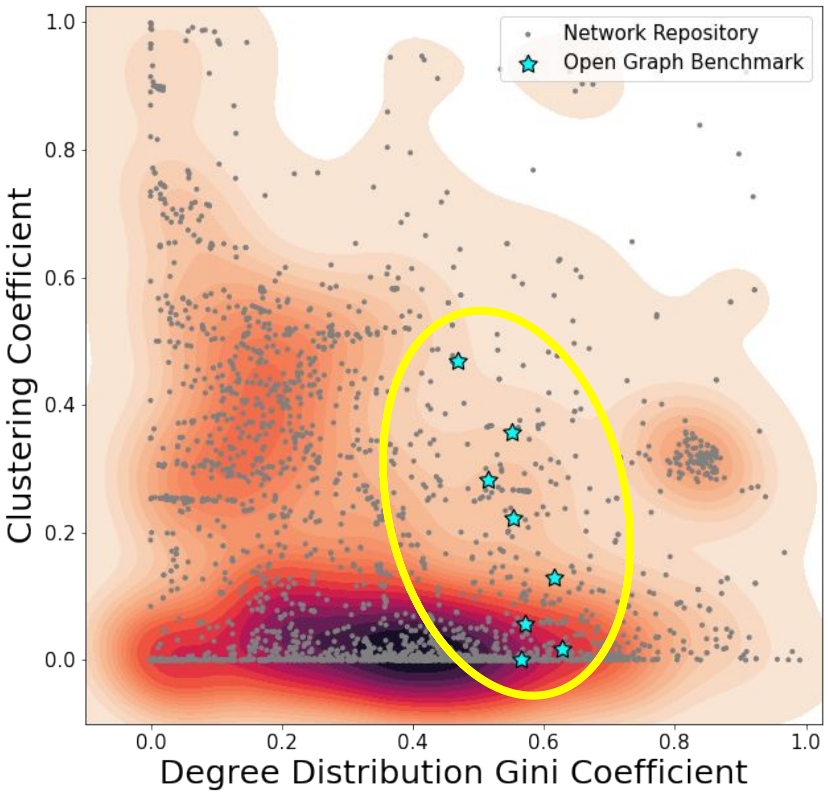
How many times have these results mattered?

Is Benchmark Reuse Good?

This will be more common with tools like Open Graph Benchmark[1] (OGB)

However, any single collection of datasets is limited in diversity

The OGB datasets fall in low-density regions of the total distribution of graphs in the Network Repository



[1] Open Graph Benchmark: Datasets for Machine Learning on Graphs (NeurIPS 2020)
[2] The Network Data Repository with Interactive Graph Analytics and Visualization (AAAI 2015)

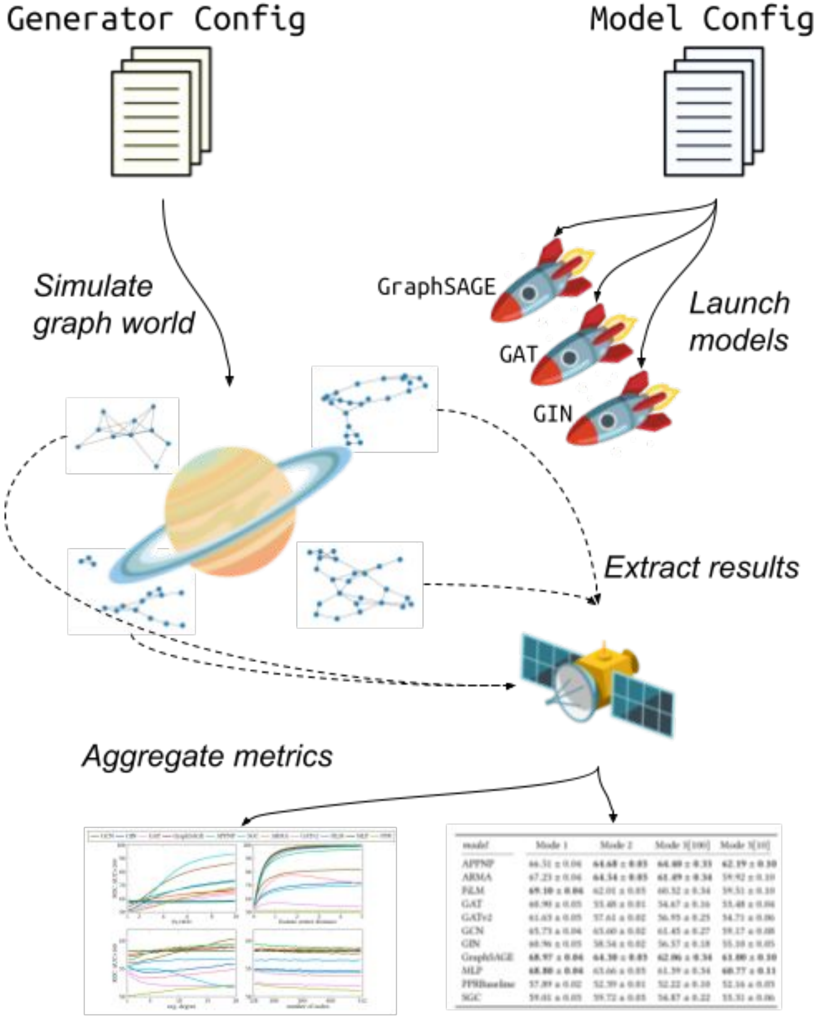
GraphWorld

Simulate millions of GNN task datasets using random graph models

Benchmark any number of GNN models on each data set

Explore and aggregate test metrics from GraphWorld tests

Idea: use GraphWorld as a complement to benchmarks on natural datasets



model	Mode 1	Mode 2	Mode 3[100]	Mode 3[10]
APPNP	66.31 ± 0.06	**64.68 ± 0.03**	**64.00 ± 0.33**	**62.19 ± 0.10**
ARMA	67.23 ± 0.06	**64.34 ± 0.03**	**61.49 ± 0.34**	59.92 ± 0.10
FiLM	**69.10 ± 0.04**	62.01 ± 0.03	60.52 ± 0.34	59.51 ± 0.10
GAT	60.90 ± 0.05	55.48 ± 0.01	54.67 ± 0.16	55.48 ± 0.04
GATv2	61.63 ± 0.05	57.61 ± 0.02	56.95 ± 0.25	54.71 ± 0.06
GCN	65.73 ± 0.04	63.60 ± 0.02	61.45 ± 0.27	59.17 ± 0.08
GIN	60.96 ± 0.05	58.54 ± 0.02	56.57 ± 0.18	55.10 ± 0.05
GraphSAGE	**68.97 ± 0.04**	**64.30 ± 0.03**	**62.06 ± 0.34**	**61.00 ± 0.10**
MLP	**68.80 ± 0.04**	63.66 ± 0.03	61.59 ± 0.34	**60.77 ± 0.11**
PPRBaseline	57.89 ± 0.02	52.39 ± 0.01	52.22 ± 0.10	52.16 ± 0.05
SGC	59.01 ± 0.05	59.72 ± 0.03	56.87 ± 0.22	55.31 ± 0.06

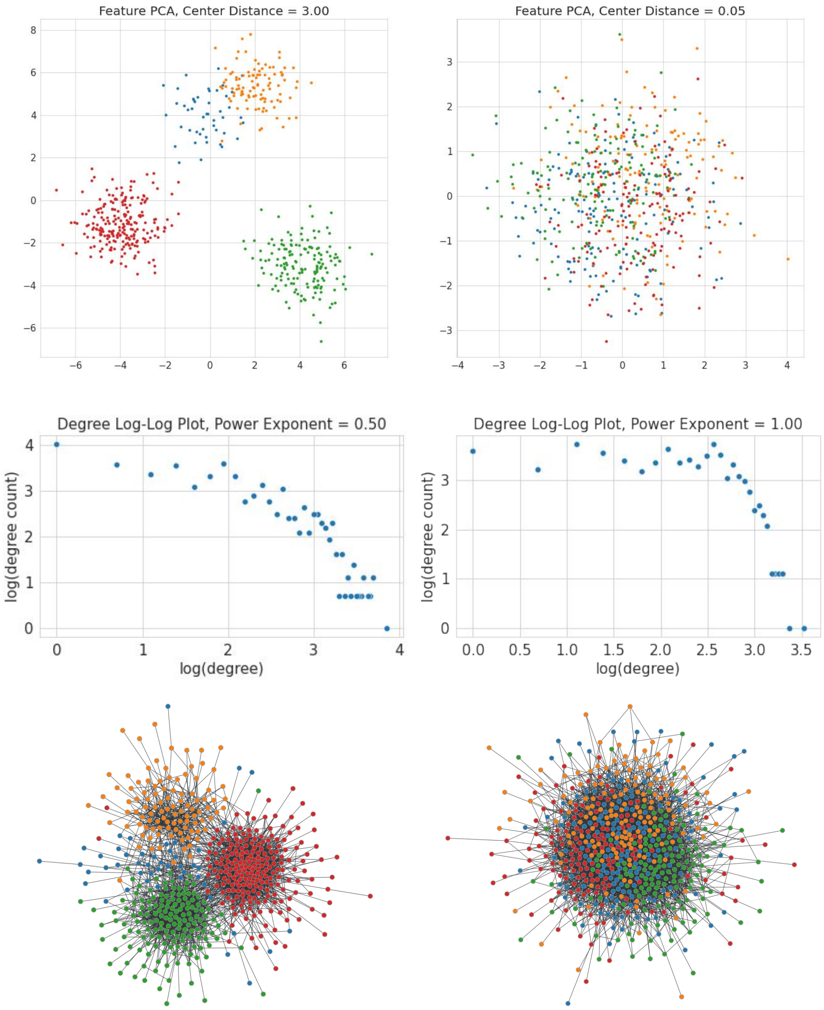
Example: Node Classification (NC)

Task: Node classification with $\mathscr{L} \in \{0, 1, 2, 3, ..., k\}$

Generator: Stochastic Block Model w/features.

Parameter	Support	Description
p-to-q ratio	[1.0, max_val]	Ratio between in-cluster & out-cluster edge probability
power exponent	(0.0, 1.0]	Degree distribution power-law
center distance	[0.0, max_val]	Cluster feature center variance

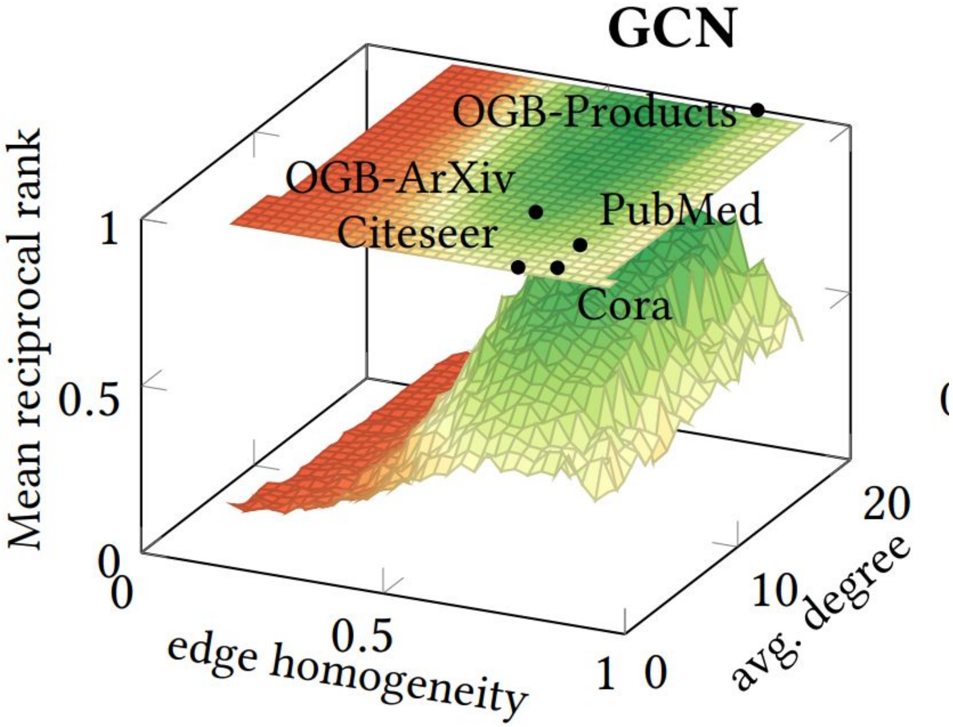
Metric: F1 one-vs-rest



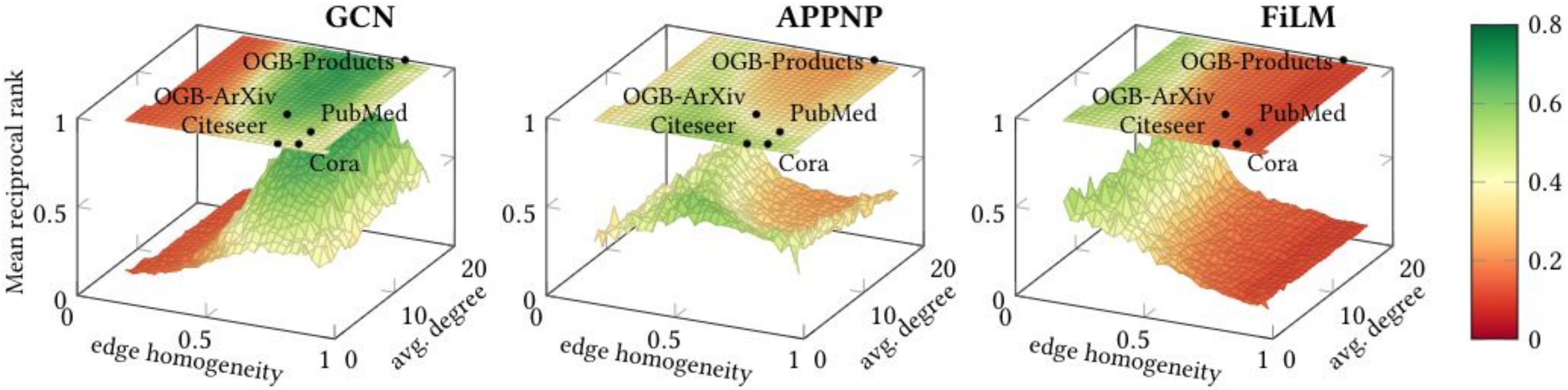
Model Performance as a Function of Graphs

We can run models on millions of different graphs that span a wide range of graph structure.

As we vary models the space, we see patterns start to emerge.



GraphWorld dataset exposes unseen model rankings



GraphWorld: Fake Graphs Bring Real Insights for GNNs

John Palowitch, Anton Tsitsulin, Brandon Mayer, Bryan Perozzi (submitted)

https://arxiv.org/abs/2203.00112

Challenge 3: Generalizing Better with GNNs

Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data

Qi Zhu, Natalia Ponomareva, Jiawei Han, Bryan Perozzi (NeurIPS'21)

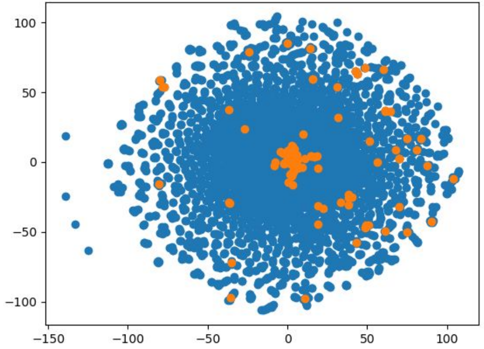
Google Research

IID vs. localized training data

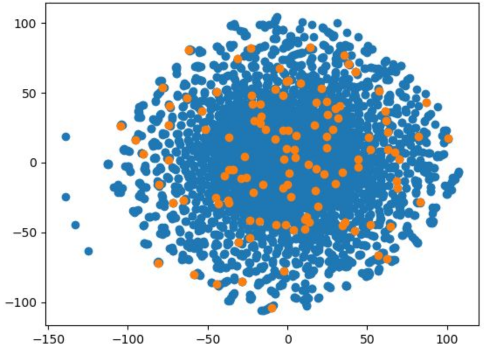
Its typical for training labels in anti-abuse to not be uniformly sampled from the dataset.

Instead, abusers are typically found and labelled by heuristics (simple rules, or human analyst).

Non-IID training data presents a challenge for GNNs, which can memorize sophisticated graph fingerprints.

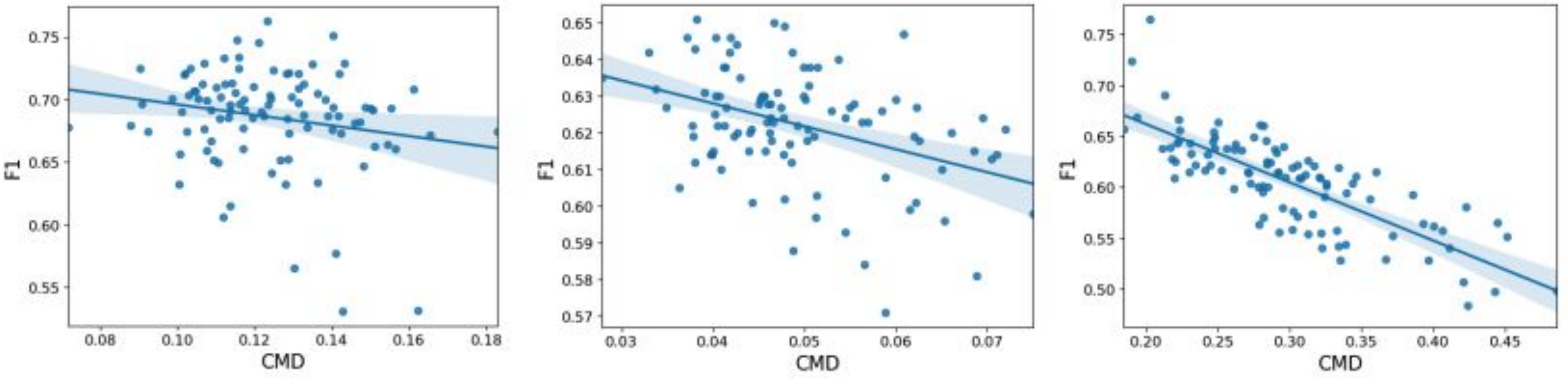


Biased samples



Uniform samples

Negative effect of distribution shifts



(a) Cora

(b) Citeseer

(c) Pubmed

Distribution shift (CMD) between training and testing data could be a good indicator of performance (F1) !

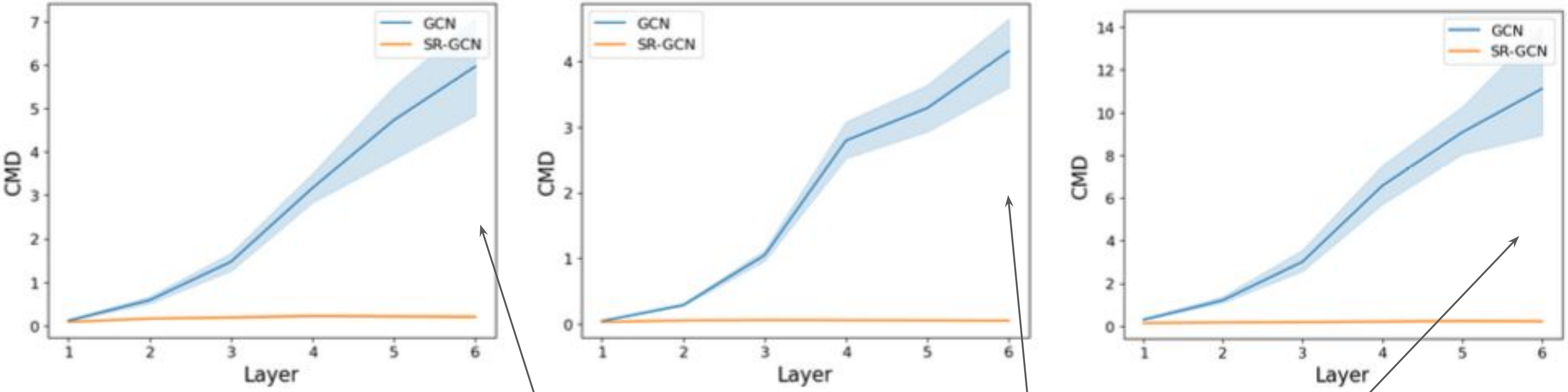
Localized annotations in real-world

- Spam and abuse detection problems typically have a very imbalanced label distribution (e.g., < 1% positive).

- Labeling nodes IID may not be feasible in practice!
  - We want to have a reasonable amount of data points from the rare positive class.

- Practical problem!

More Motivation:
Domain Shift is a Problem in Deeper GNNs



(a) Cora

(b) Citeseer

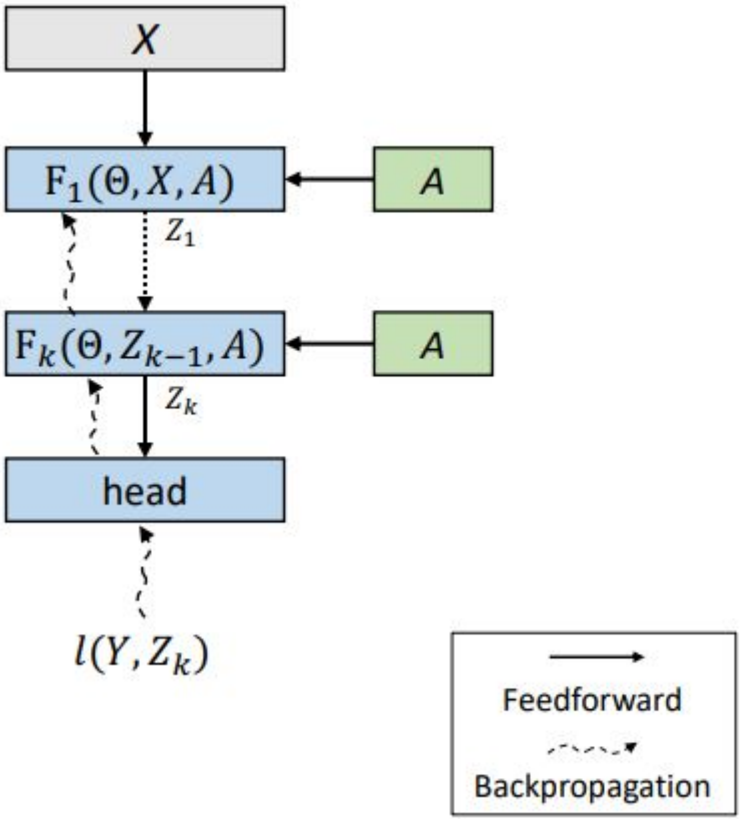
(c) Pubmed

Shift gets bigger as models get deeper!

Shift-Robust GNNs

Solution: We propose regularizations to make GNNs robust against domain shift.

***Normal* GNN** - Fully differentiable deep models allow applying domain shift regularization at any layer.



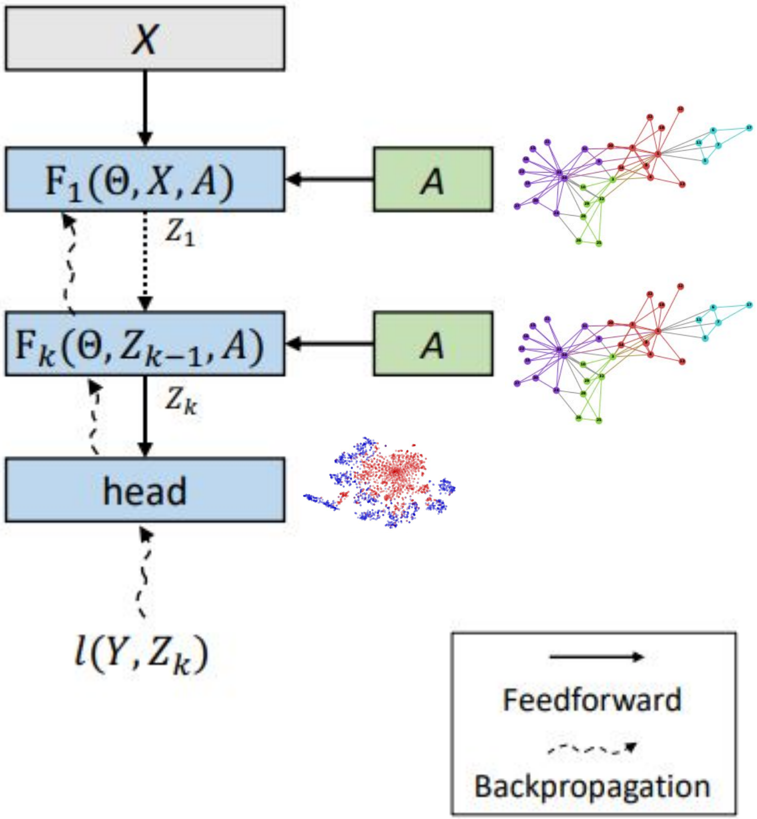
Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data

Qi Zhu, Natalia Ponomareva, Jiawei Han, Bryan Perozzi (NeurIPS'21)

Shift-Robust GNNs

Solution: We propose regularizations to make GNNs robust against domain shift.

***Normal* GNN** - Fully differentiable deep models allow applying domain shift regularization at any layer.



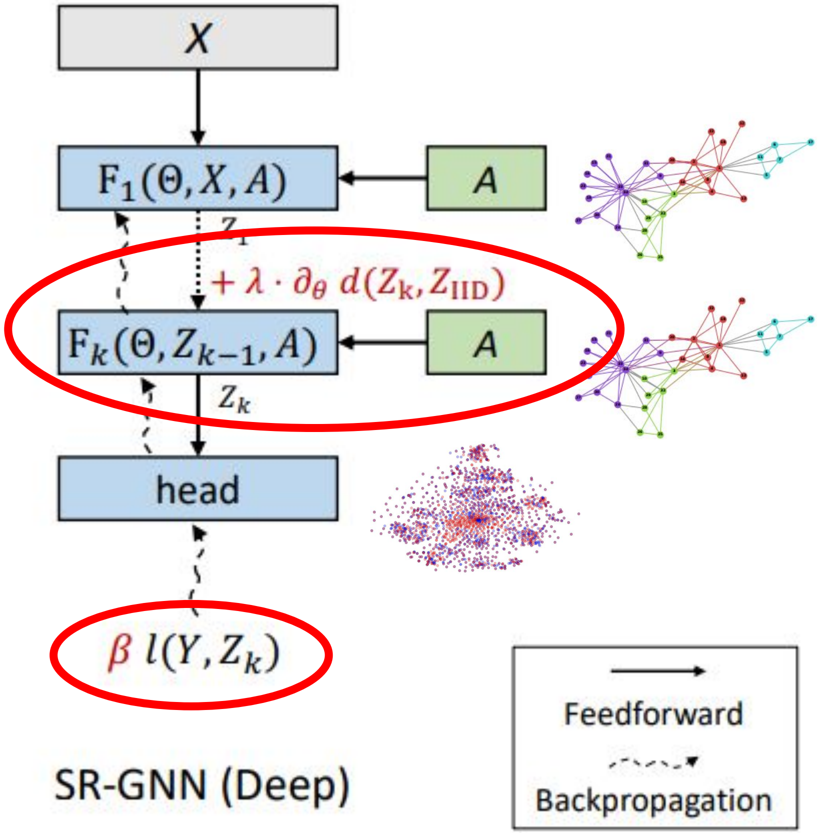
Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data

Qi Zhu, Natalia Ponomareva, Jiawei Han, Bryan Perozzi (NeurIPS'21)

Shift-Robust GNNs

Solution: We propose regularizations to make GNNs robust against domain shift.

***Normal* GNN** - Fully differentiable deep models allow applying domain shift regularization at any layer.



Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data

Qi Zhu, Natalia Ponomareva, Jiawei Han, Bryan Perozzi (NeurIPS'21)

Shift-Robust GNNs

Solution: We propose regularizations to make GNNs robust against domain shift.

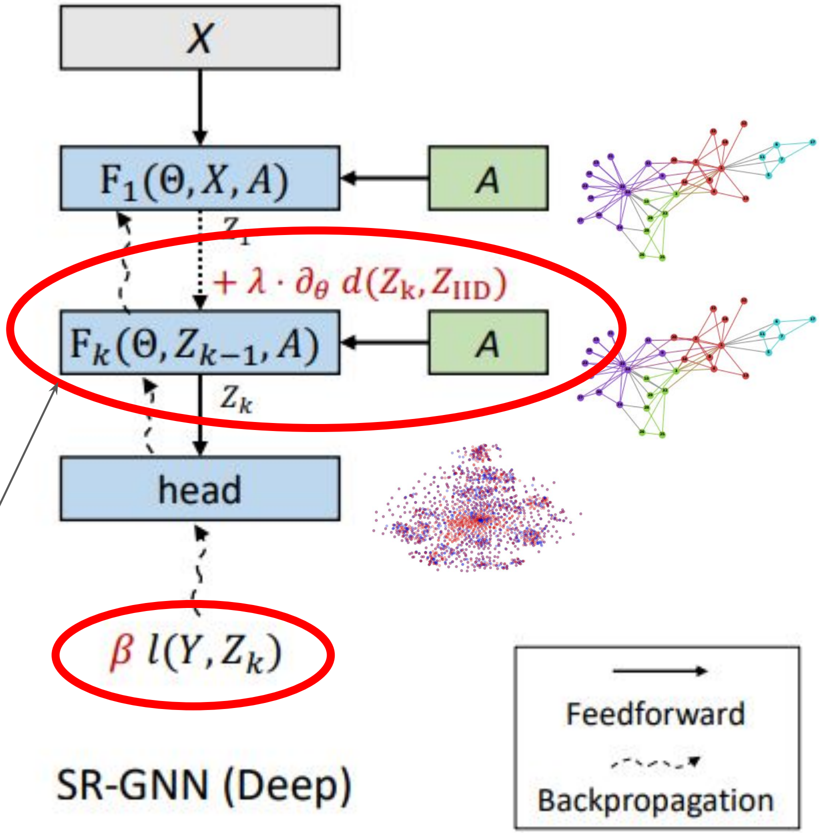
***Normal* GNN** - Fully differentiable deep models allow applying domain shift regularization at any layer.

Case 1: Normal GNNs

At each layer, the graph inductive bias is multiplicative.

We can regularize a layer in this network to force the features to be representative for both a biased and unbiased samples:

$$\mathcal{L} = \frac{1}{M}\sum_i l(y_i, z_i) + \lambda \cdot d(Z_{\text{train}}, Z_{\text{IID}})$$


X
$F_1(\Theta, X, A)$
A
Z_1
$+\lambda \cdot \partial_\theta\, d(Z_k, Z_{\text{IID}})$
$F_k(\Theta, Z_{k-1}, A)$
A
Z_k
head
$\beta\, l(Y, Z_k)$
Feedforward
Backpropagation

SR-GNN (Deep)

Shift-Robust GNNs: Overcoming the Limitations of Localized Graph Training data

Qi Zhu, Natalia Ponomareva, Jiawei Han, Bryan Perozzi (NeurIPS'21)

Thanks -- Please reach out!

bperozzi@google.com