Applying Deep Graph Representation Learning to the Malware Graph

C. Bayan Bruss
Conference on Applied Machine Learning For Information Security
October 25th, 2019

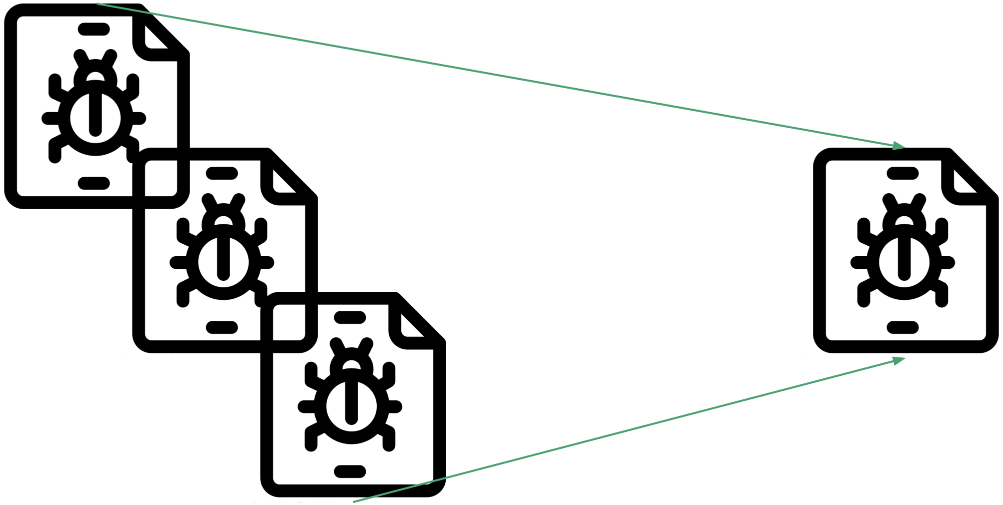
6

‘New’ malware specimens detected per second

325

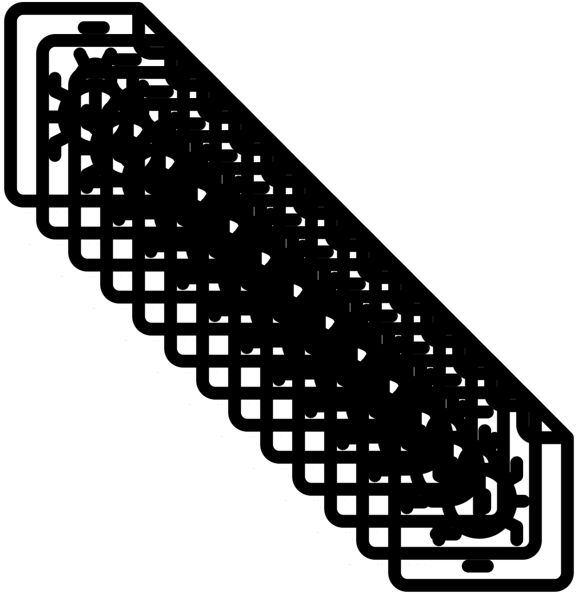
Publicly reported malware related breaches in 2017

Technical Problem



Often specimens are variants on existing malware or combinations of older functionality

This creates a need to identify if a piece of malware is similar to existing malware and what functionality is known about the new sample



At the same time, the scale and latency requirements leave little room for full reverse engineering of each sample

Traditional ML Solutions

1. Traditional ML relies on features that can be quantified at an individual sample level

2. Leaves rich information about overlapping sets of components unused

3. Struggles to induce similarity at scale on new samples

Solution:
Apply graph
machine learning to
malware

Agenda

1. Brief Introduction to Graph Representation Learning
2. Malware as a Graph
3. Applying Graph Representation Learning to Malware Graphs
4. Thoughts and Future Directions

Introduction

Graphs

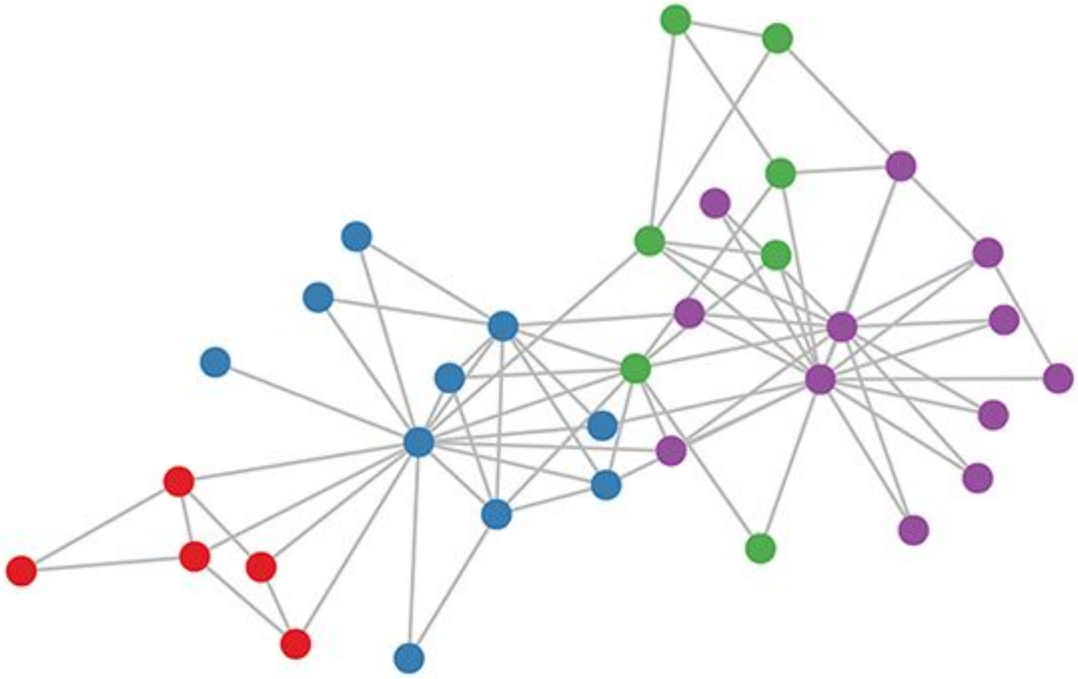
Vertices, Edges

Types of graphs

- Directed vs Undirected
- Weighted vs Unweighted
- Attributed

Graph Properties

- Degree
- Density
- Communities

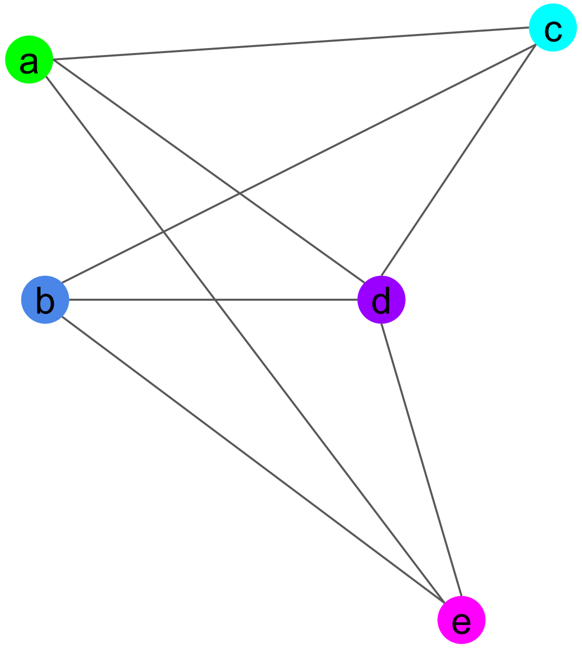


https://tkipf.github.io/graph-convolutional-networks/

Graphs

Ways of representing

- Edge list
- Adjacency matrix
- Laplacian



Graphs

Ways of representing

- **Edge list**
- Adjacency matrix
- Laplacian

a	c
a	d
a	e
b	c
b	d
b	e
d	e
d	c

Graphs

Ways of representing

- Edge list
- **Adjacency matrix**
- Laplacian

	a	**b**	**c**	**d**	**e**
a	0	0	1	1	1
b	0	0	1	1	1
c	1	1	0	1	0
d	1	1	1	0	1
e	1	1	0	1	0

Graphs

Ways of representing

- Edge list
- Adjacency matrix
- **Laplacian**

	a	**b**	**c**	**d**	**e**
a	3	0	-1	-1	-1
b	0	3	-1	-1	-1
c	-1	-1	3	-1	0
d	-1	-1	-1	4	-1
e	-1	-1	0	-1	3

Graph Representation Learning

Graph Embeddings

A set of unsupervised (or semi-supervised) machine learning techniques for projecting graphs into low dimensional continuous vector space while preserving key graph properties.

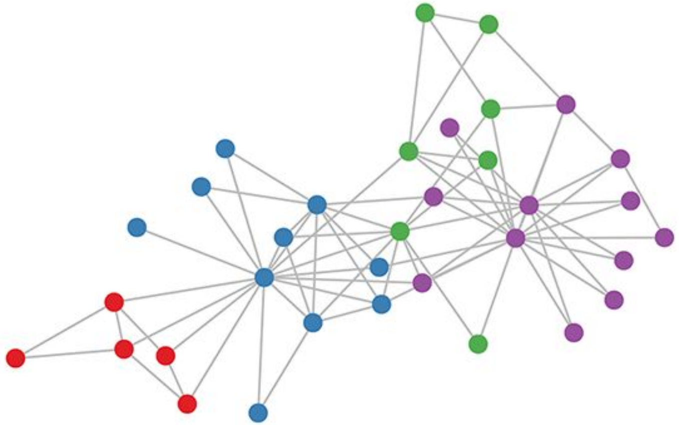
What does embedding a graph mean?

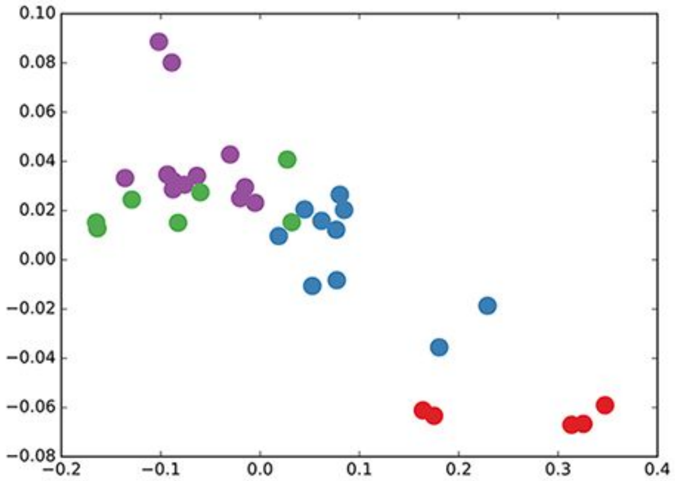
Learn a representation for each node in such a way that you can recover certain properties of the original graph.

- Neighborhood
- Semantic structure

But Why?

- Node classification
- Node clustering & search
- Edge Anomaly Detection





https://tkipf.github.io/graph-convolutional-networks/

Shallow Random Walk Embeddings

If we take the word2vec approach, which takes sentences containing words and generalize that to any sequence of tokens can be an input to the skipgram model.

[‘the’, ‘quick’, ‘red’, ‘fox’, ‘jumped’, ‘over’, ‘the’, ‘red’, ‘rose’]

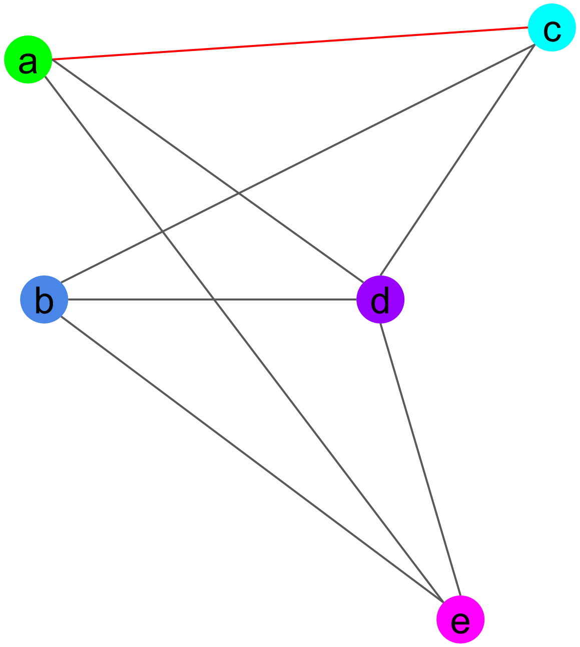
[0, 1, 2, 3, 4, 5, 0, 2, 6]

Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
   a. What is a random walk
   b. How can you bias the random walk

[a, c]

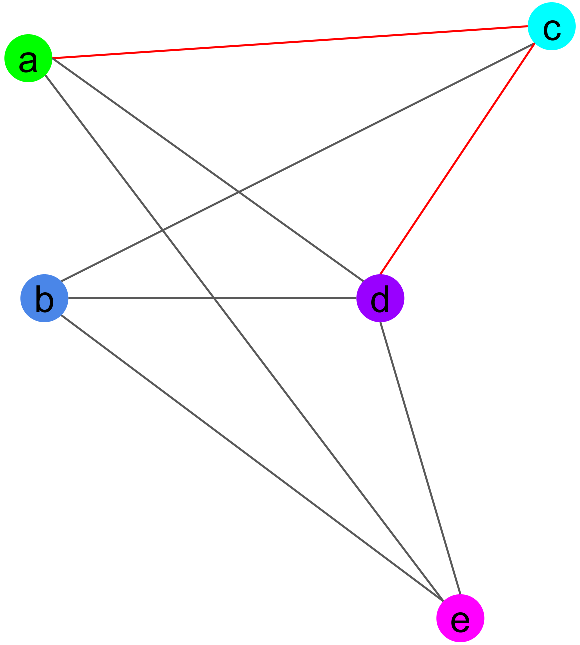


Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
   a. What is a random walk
   b. How can you bias the random walk

[a, c, d]

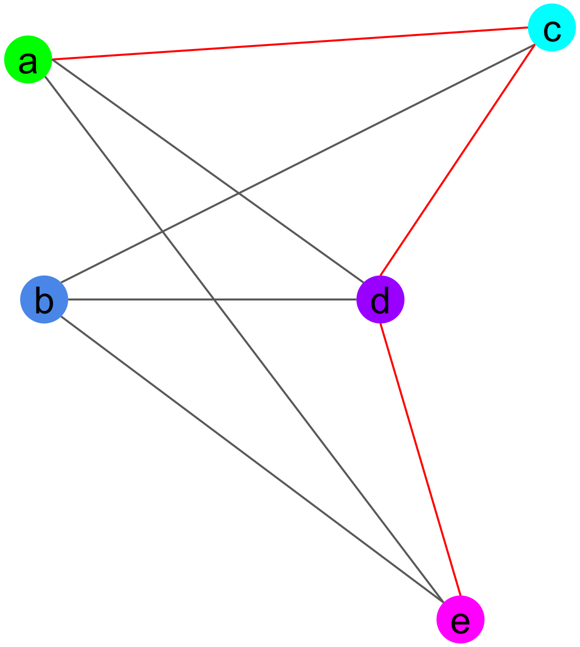


Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
   a. What is a random walk
   b. How can you bias the random walk

[a, c, d, e]

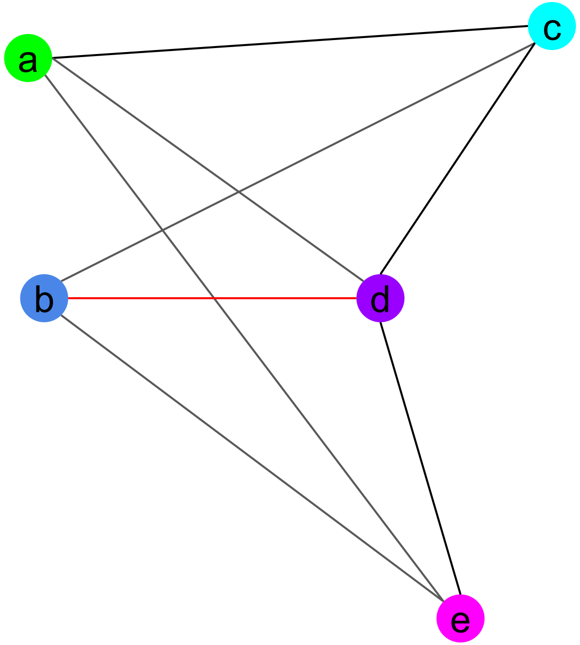


Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
   a. What is a random walk
   b. How can you bias the random walk

[b, d]

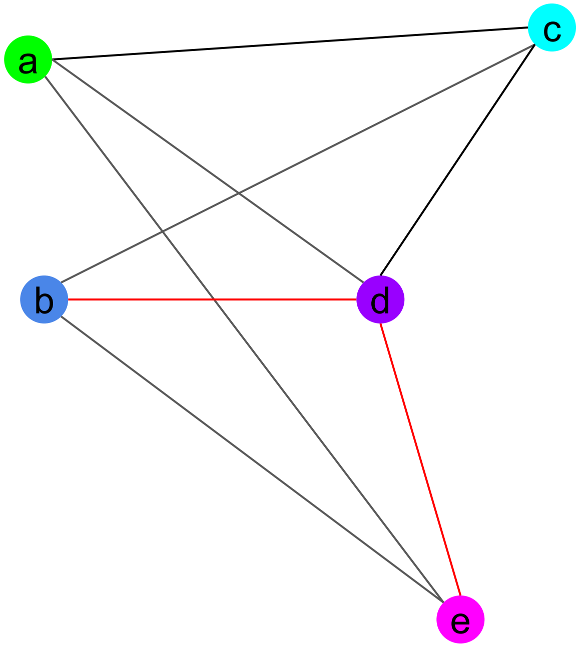


Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
   a. What is a random walk
   b. How can you bias the random walk

[b, d, e]

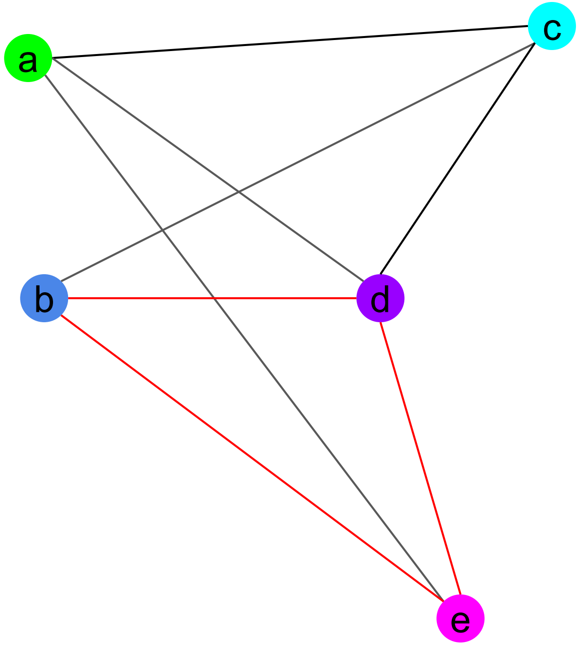


Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
    a. What is a random walk
    b. How can you bias the random walk

[b, d, e, b]

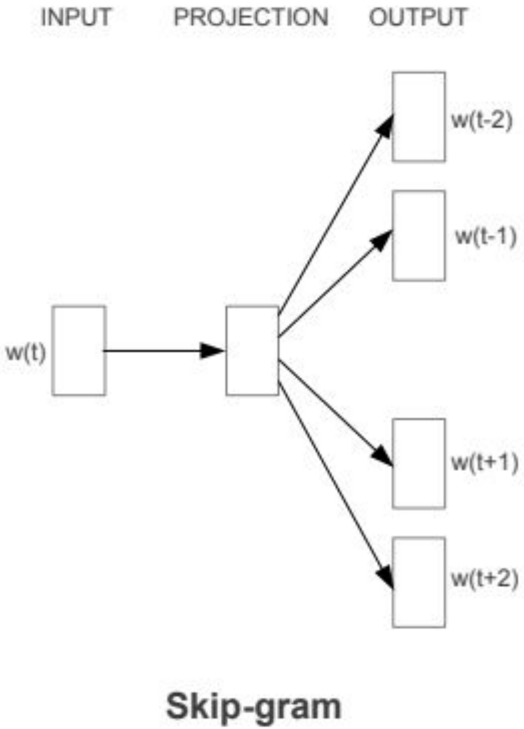


Shallow Random Walk Embeddings

How do we create a training dataset of sequences for the skipgram model?

1. First do long random walks on the graph
   a. What is a random walk
   b. How can you bias the random walk
2. Feed random walk sequences into Skip-Gram Model

[a, c, d, e]
[b, d, e, b]

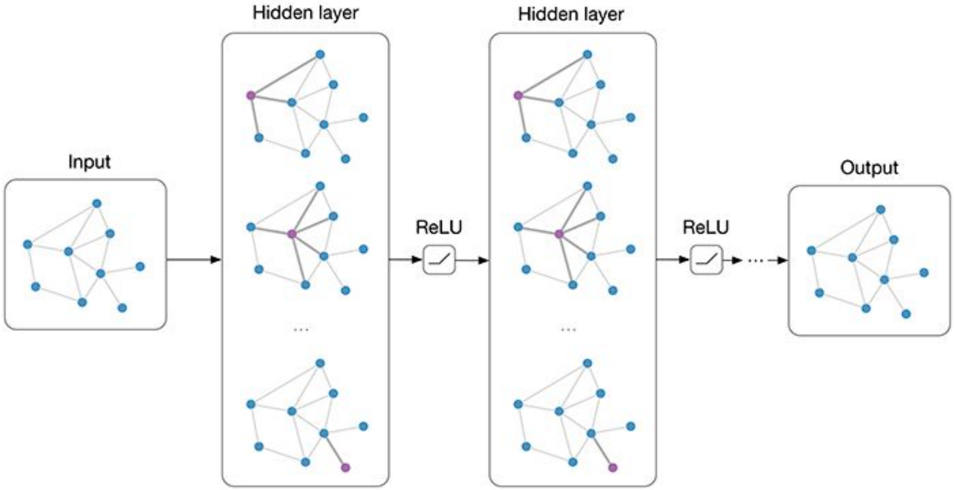


Deep Graph Embeddings

- Instead of single layer skip gram model, deep models learn to recover more information from neighborhood (2nd order connections)

- Modify popular filter mechanisms from CNN domain to apply to graph domain

- Can also be applied to semi and fully supervised graph learning



https://tkipf.github.io/graph-convolutional-networks/

Malware as a Graph

Defining the Edges of a Graph

What makes a good graph

- Many to many relationships
- Edges define some kind of semantic or meaningful relationship
- Can be easily generated on ‘new’ nodes
- Isn’t easily represented as a node-attribute

Ways to Define Edges for Malware

Static

- Dll Imports
- Yara matches

Dynamic

- IP Calls
- Contacted Domains
- Contacted URLs
- Files downloaded

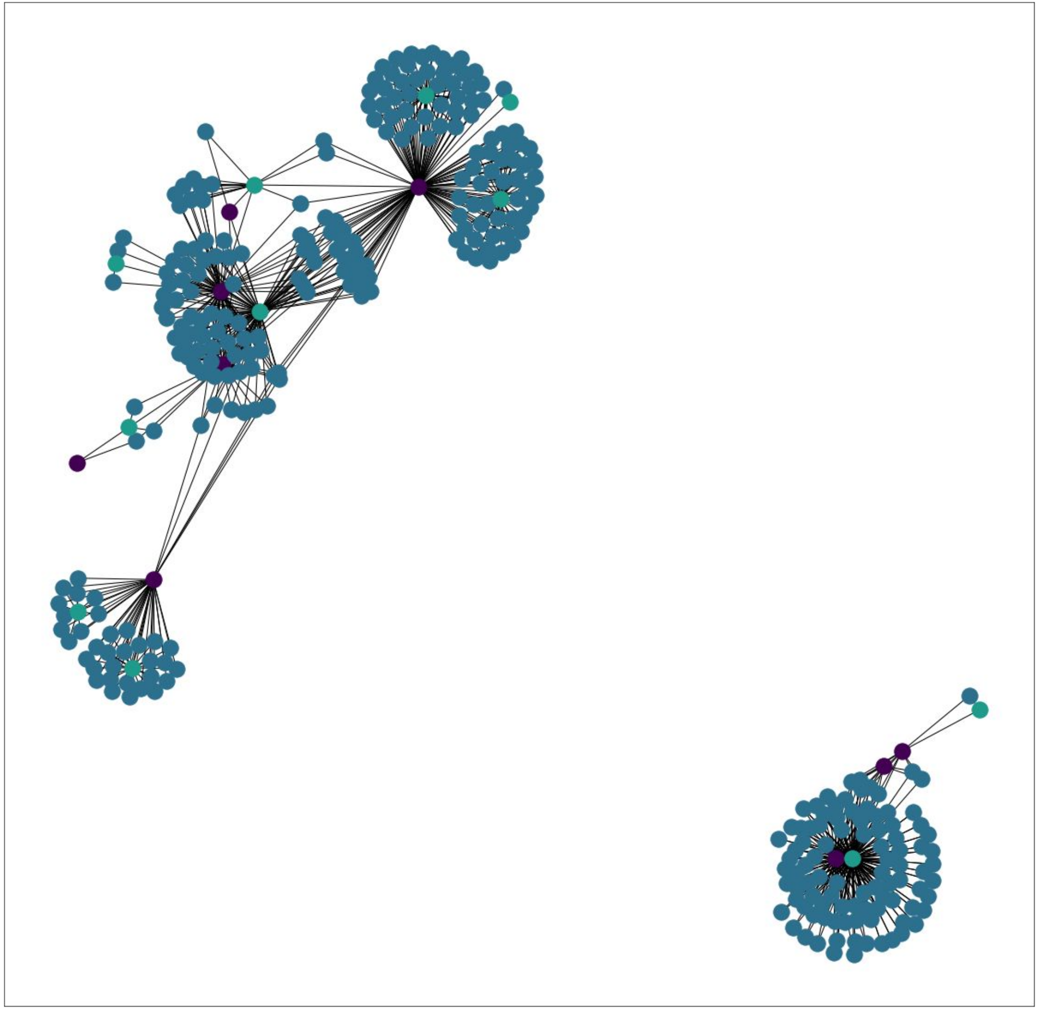
Possible Node Attributes

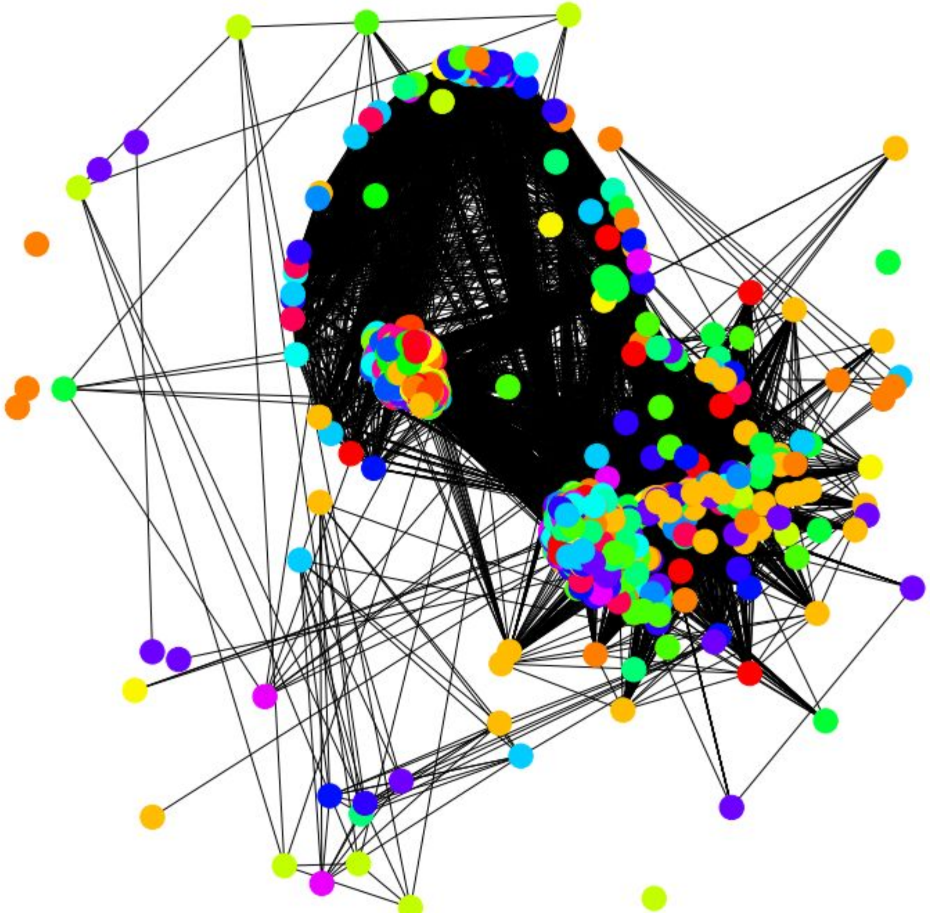
Static

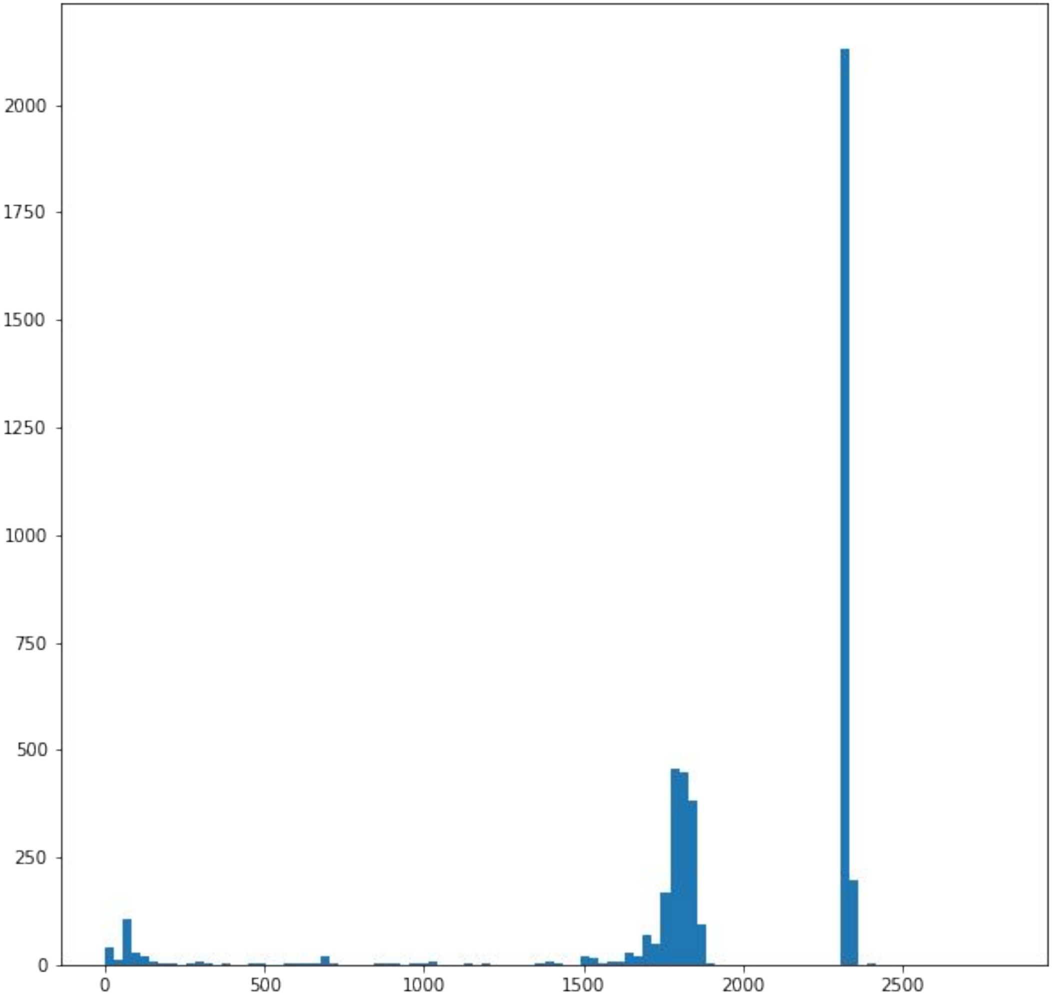
- File size
- Section Info
- File type

Dynamic

- Specific behaviors






2000
1750
1500
1250
1000
750
500
250
0
0
500
1000
1500
2000
2500

Results

Three Tasks

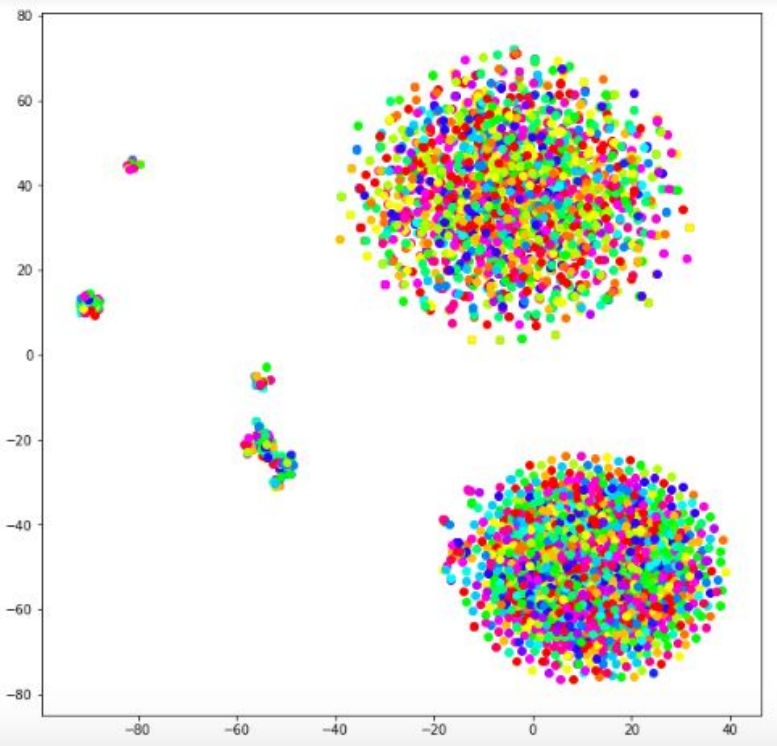
1) Node Clustering: How well does the model recover the original topology of the graph

2) Node Induction: Given a new node can you quickly infer its location in the latent space

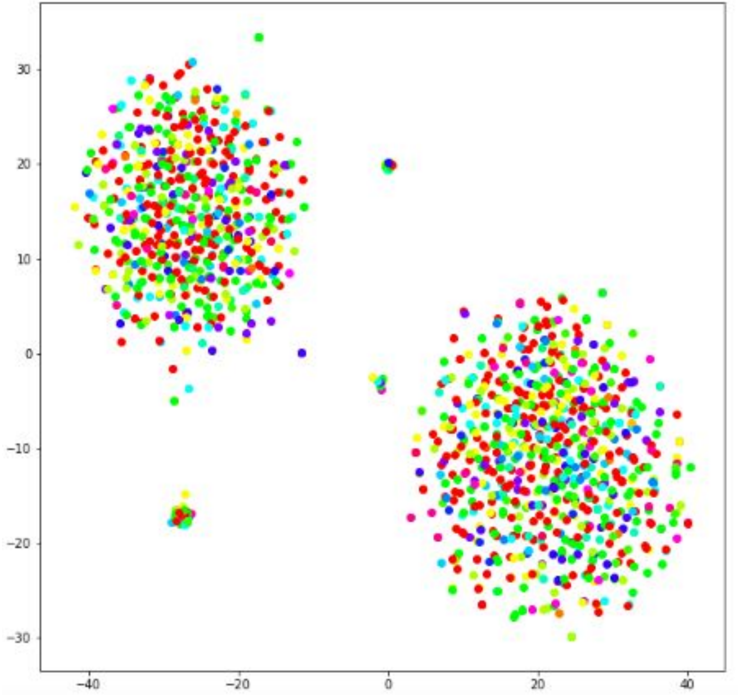
3) Node Classification: Can you assign a label to the node given its position on the graph

Node Clustering

Shallow Random Walk Embedding

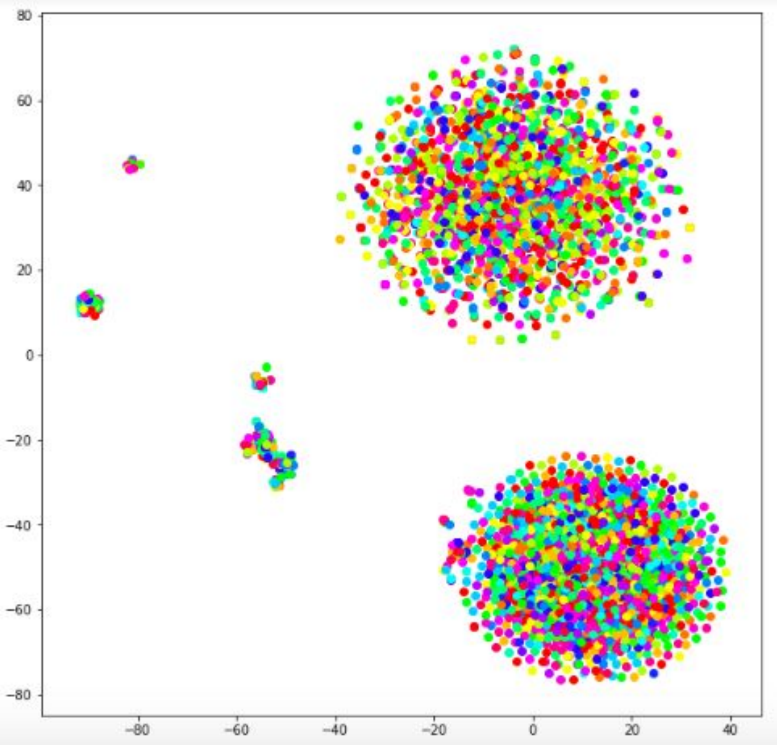


Deep Graph Embedding

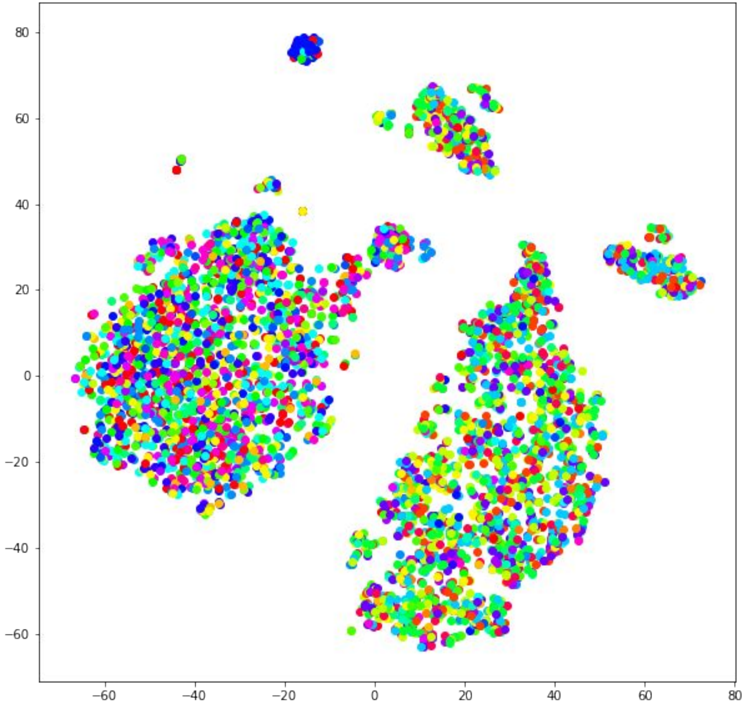


Node Clustering

Shallow Random Walk Embedding

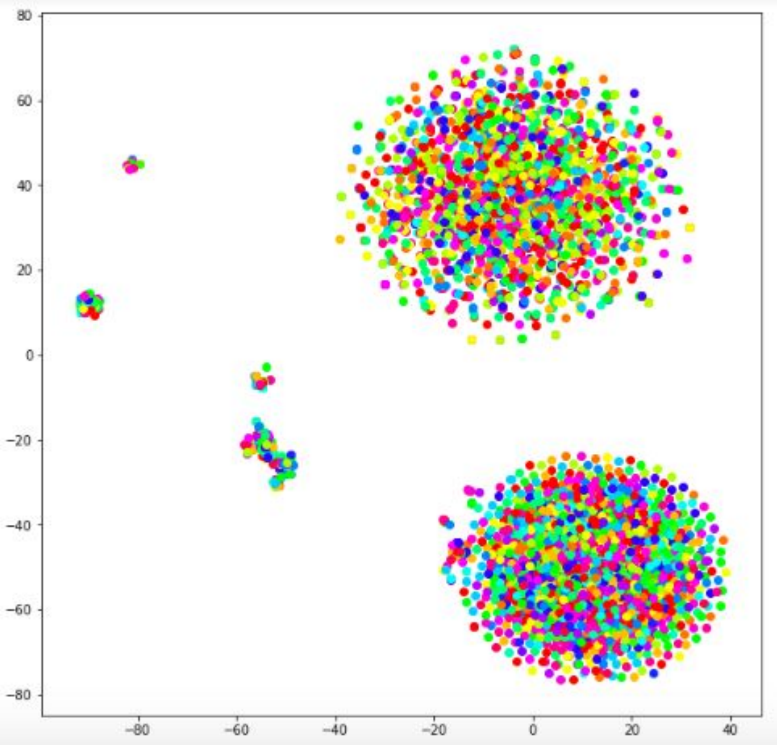


Deep Graph Embedding w/Node Attributes

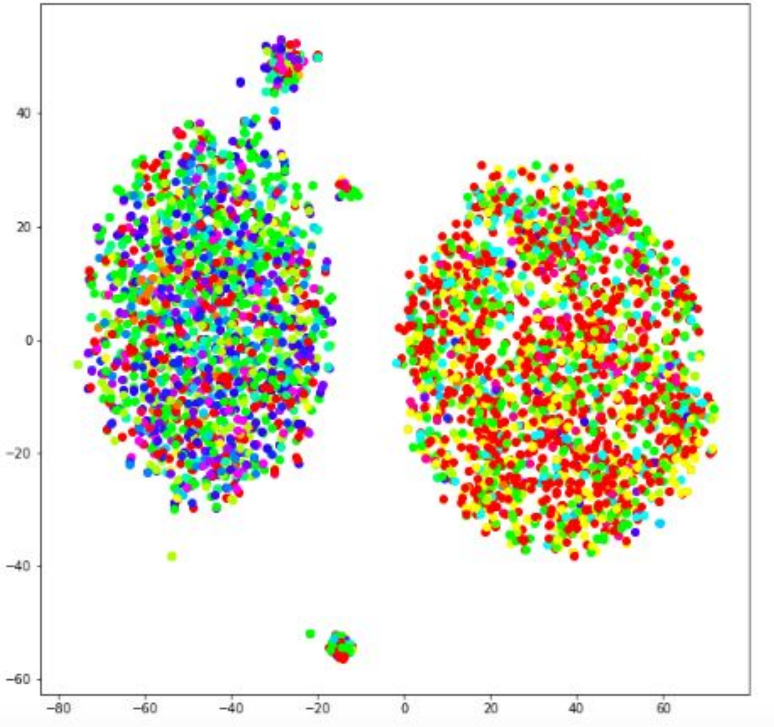


Node Clustering

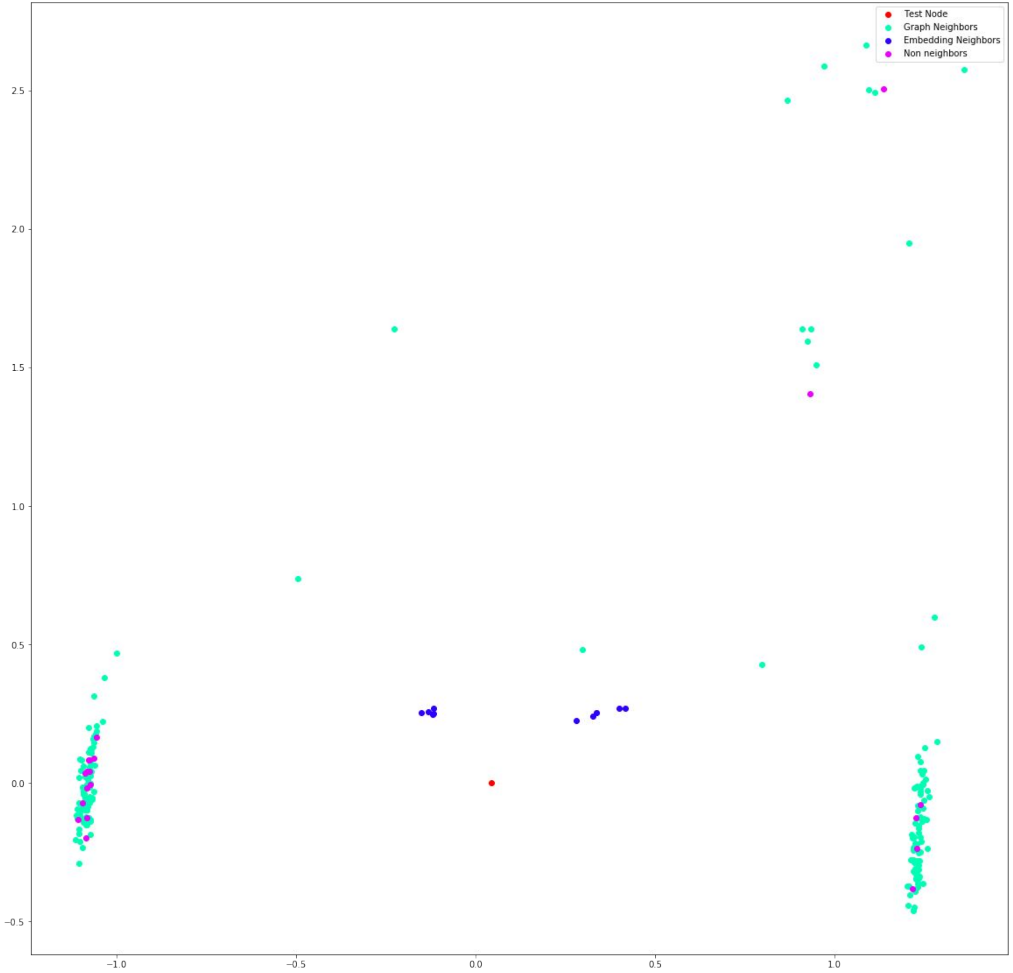
Shallow Random Walk Embedding



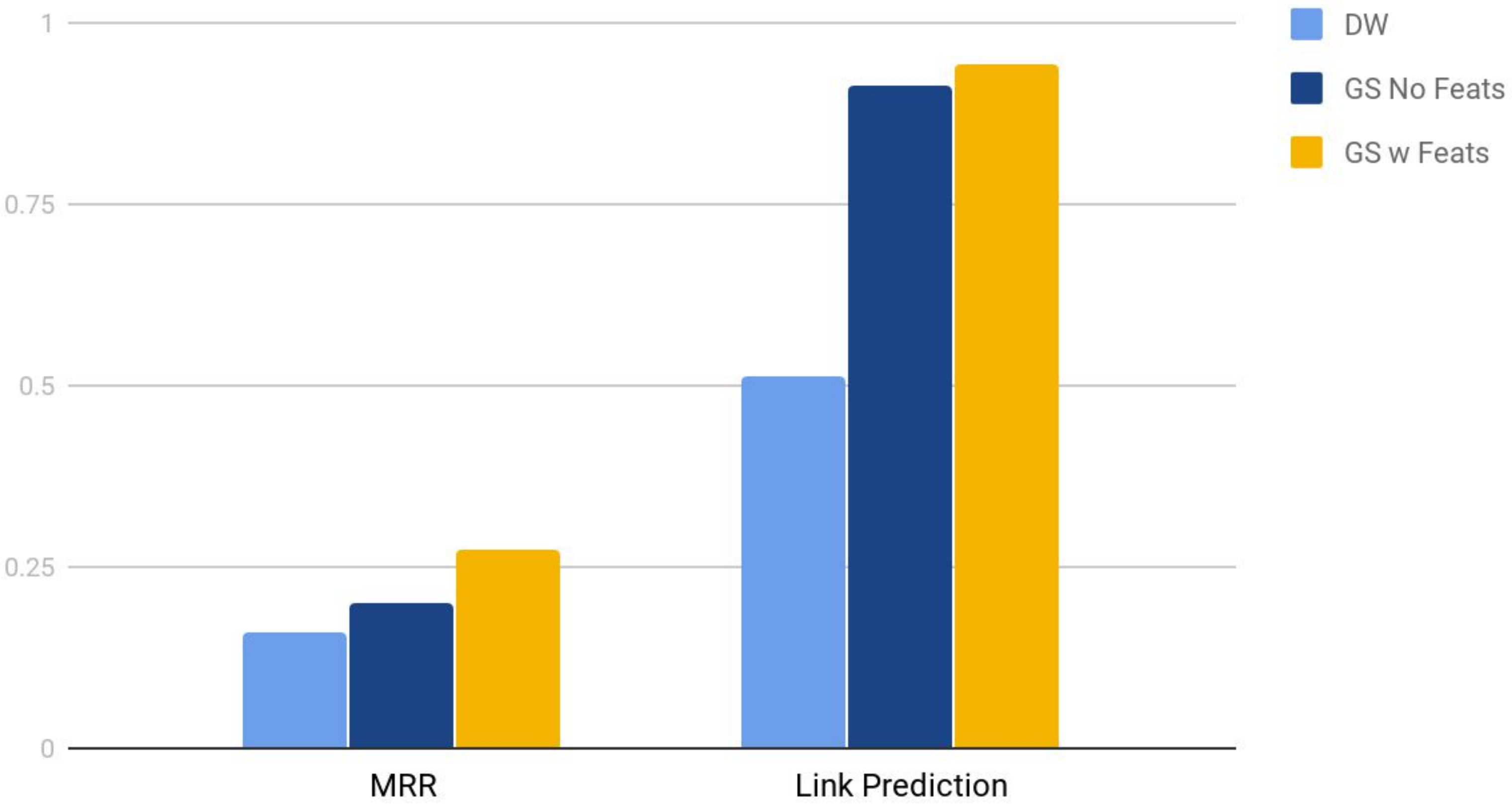
Deep Graph Embedding w/VT label as attribute



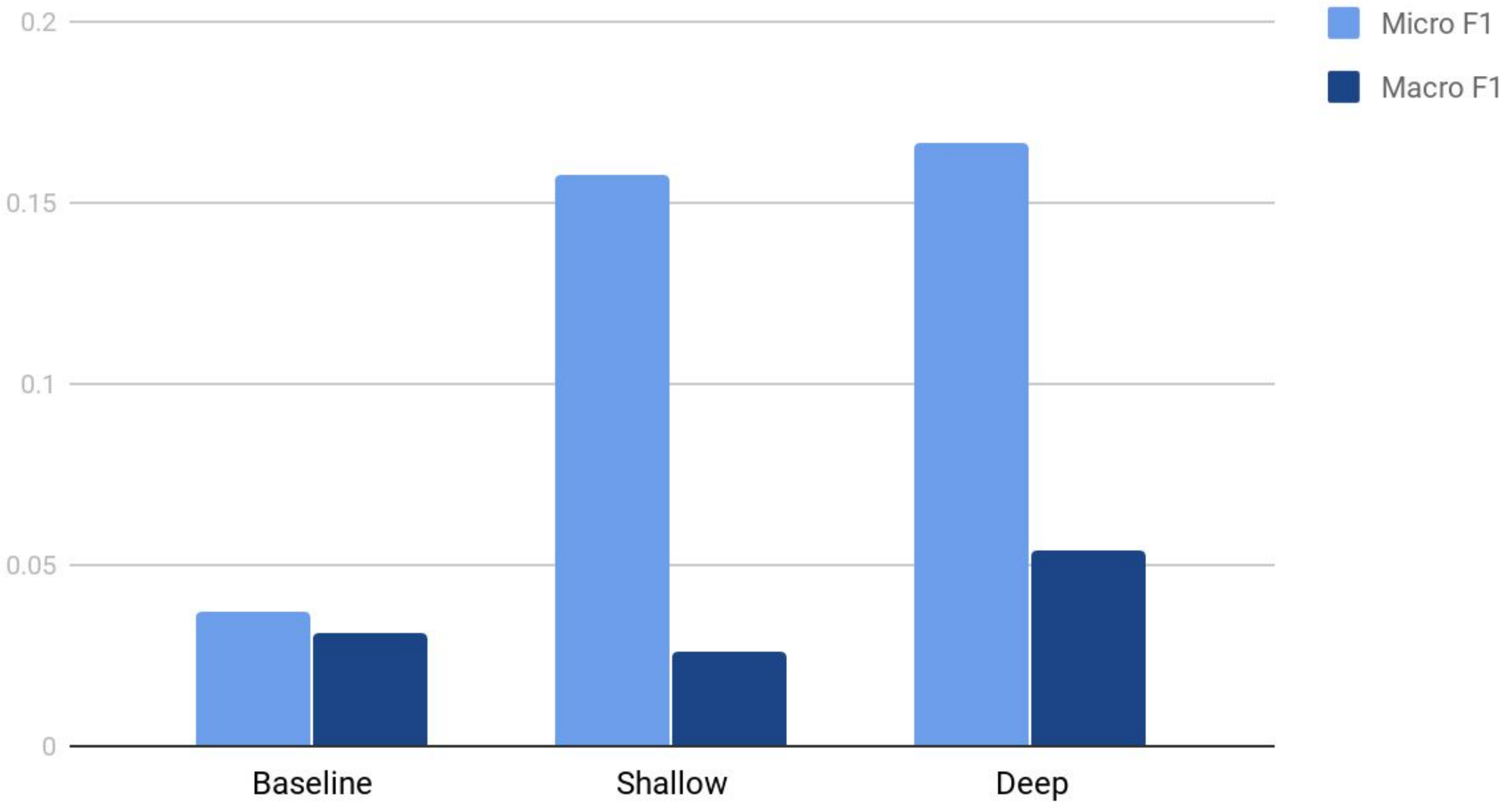
Node Induction



Performance on Node Induction Task



Node Classification F1 Score (25 Classes)



Future Directions

Next Steps

Add dynamic attributes

Benchmark against more common solutions

Scale to larger graphs

Incorporate more attributes & edge types

Thank You

www.cbbruss.com