HARP

Hierarchical Representation Learning for Networks

ONE Network Embedding

TWO Algorithm: HARP

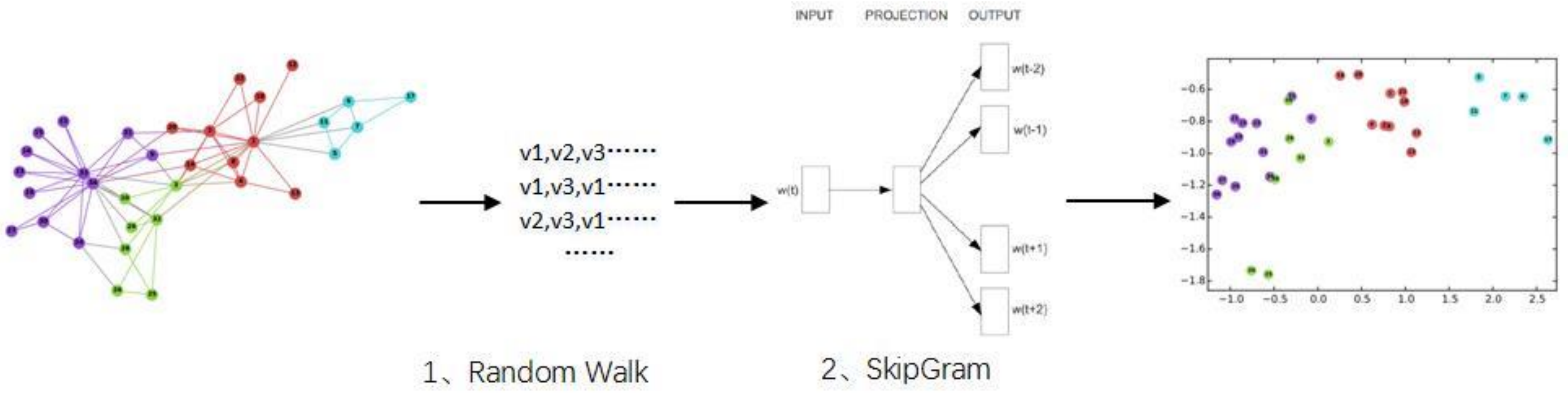
THREE Experiment

PART 1

Network Embedding

- DeepWalk
- Line
- Node2vec

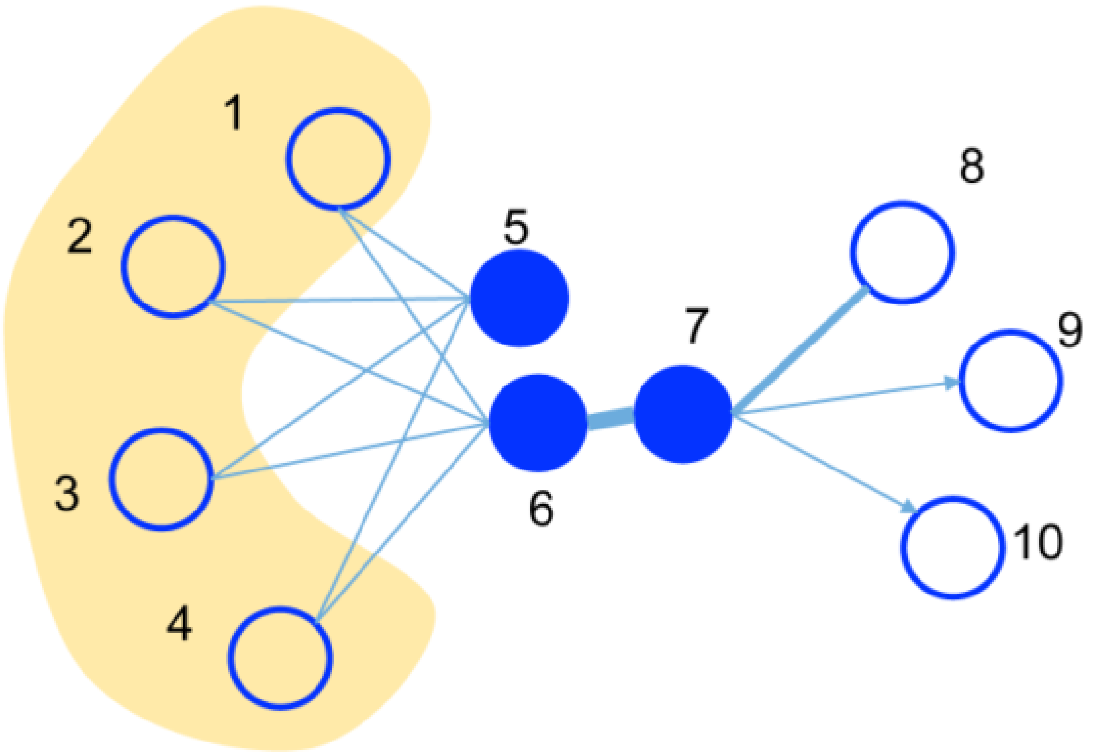
DeepWalk



steps:

- Network/graph ------->random walk --------->representation mapping--------> skip-gram model---
  ------>output：representation

Line



- First-order proximity

$$p_1(v_i, v_j) = \frac{1}{1+exp(-\vec{u}_i^T \cdot \vec{u}_j)}$$

- Second-order proximity

$$p_1(v_j|v_i) = \frac{exp(\vec{u}_j'^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} exp(\vec{u}_k'^T \cdot \vec{u}_i)}$$

Node2vec

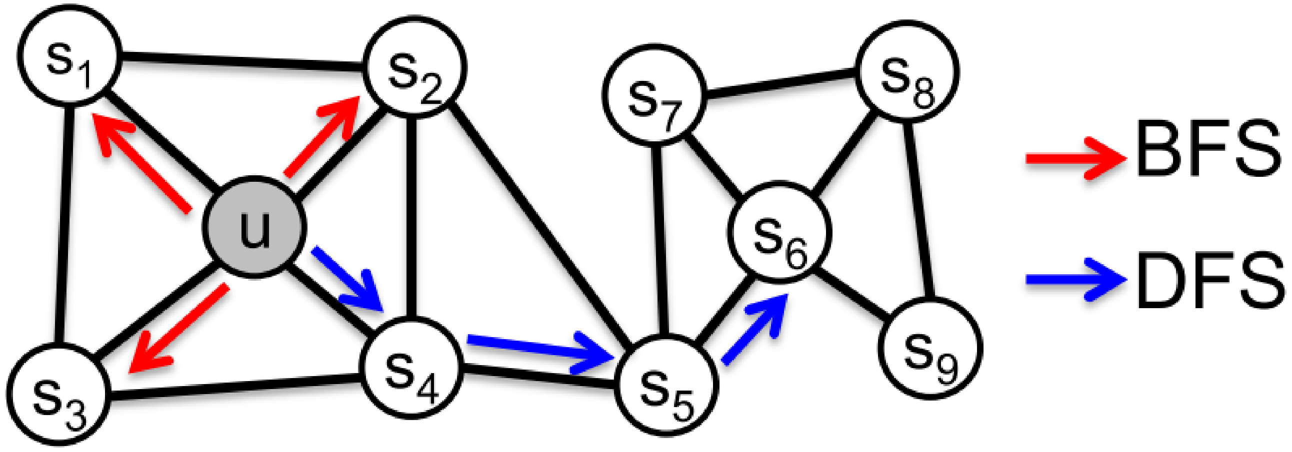
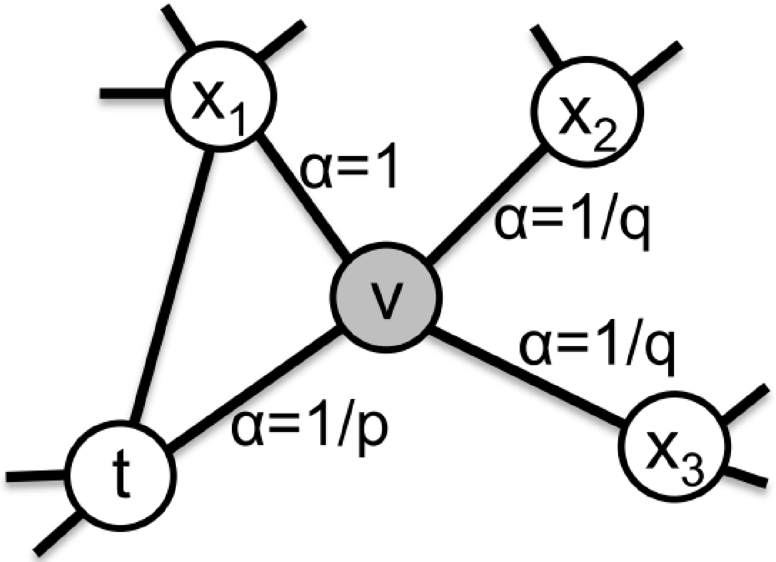


Figure 1: BFS and DFS search strategies from node u ($k = 3$).



- DeepWalk: p=1、q=1

PART 2

Algorithm: HARP

The main idea

HARP Algorithm

- graph coarsening
- graph embedding
- representation refinement

Main idea

problems:

- higher-order graph structural information is not modeled
- stochastic optimization can fall victim to poor initialization

methods:

- coalesces the nodes and edges ——>smaller graphs(original graph’s global structure)
- G_i’s representation——>G_{i-1}’s initialization

output:

- HARP(DW)，HARP(LINE)，HARP(N2V)

Algorithm 1

Algorithm 1 HARP(G, $Embed()$)

Input:

graph $G(V, E)$

arbitrary graph embedding algorithm EMBED()

Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

1: $G_0, G_1, \cdots, G_L \leftarrow \text{GRAPHCOARSENING}(G)$

2: Initialize Φ'_{G_L} by assigning zeros

3: $\Phi_{G_L} \leftarrow \text{EMBED}(G_L, \Phi'_{G_L})$

4: **for** $i = L - 1$ to 0 **do**

5: $\quad \Phi'_{G_i} \leftarrow \text{PROLONGATE}(\Phi_{G_{i+1}}, G_{i+1}, G_i)$

6: $\quad \Phi_{G_i} \leftarrow \text{EMBED}(G_i, \Phi'_{G_i})$

7: **end for**

8: **return** Φ_{G_0}

steps:

- Graph Coarsening (line 1)
- Graph Embedding on the Coarsest Graph (line 2-3)
- Graph Representation Prolongation and Refinement (line 4-7)
- Graph Embedding of the Original G raph (line 8)

Algorithm 2

Algorithm 2 GraphCoarsening(G)

Input: graph $G(V, E)$
Output: Series of Coarsened Graphs $G_0, G_1, \cdots, G_L$

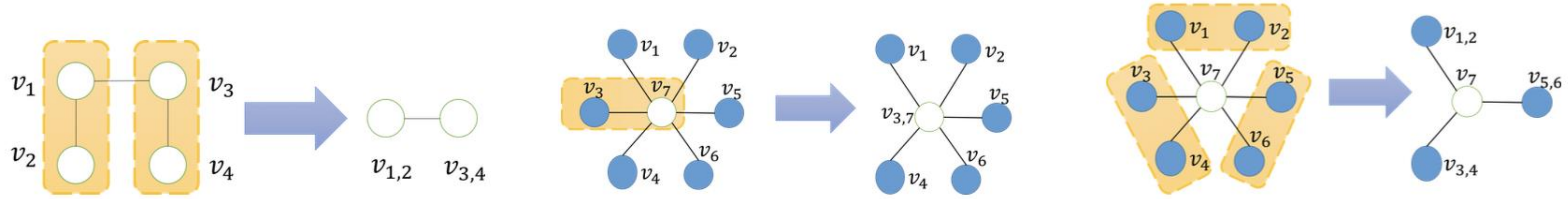
$L \leftarrow 0$
$G_0 \leftarrow G$
while $|V_L| \geq threshold$ **do**
    $L \leftarrow L + 1$
    $G_L \leftarrow \textsc{EdgeCollapse}(\textsc{StarCollapse}(G))$
end while
return $G_0, G_1, \cdots, G_L$

Edge Collapsing :

- preserving first-order proximity

Star Collapsing :

- preserving second-order proximity



(a) Edge Collapsing. (b) Edge Collapsing fails to collapse stars. (c) Star Collapsing.

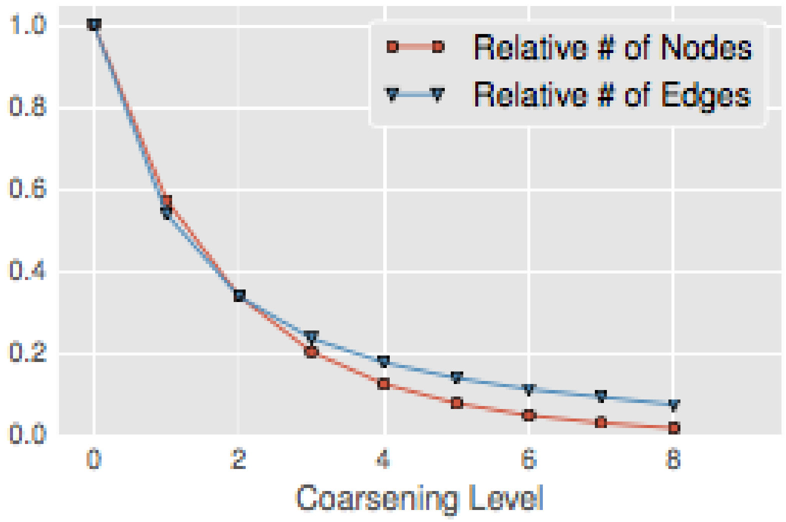
PART 3

Experiment

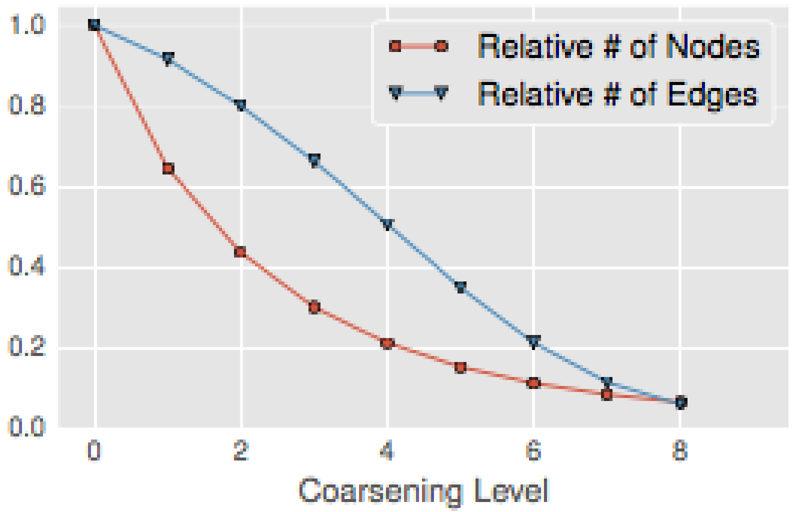
- Datasets & Graph Coarsening
- Visualization

Datasets& Graph Coarsening

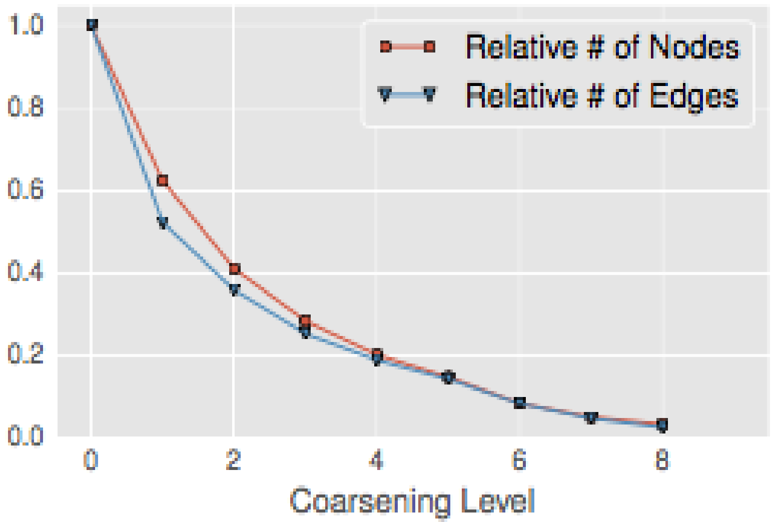
Name	DBLP	Blogcatalog	CiteSeer
# Vertices	29,199	10,312	3,312
# Edges	133,664	333,983	4,732
# Classes	4	39	6
Task	Classification	Classification	Classification



DBLP



Blogcatalog



CiteSeer

Visualization

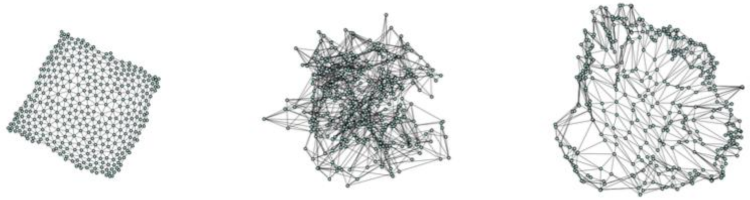
- HARP(LINE)
  - Level 7 —— smallest graph
  - Level 0 —— original graph.
  - Input—— force-direct graph drawing

(a) Level 7 (b) Level 6 (c) Level 5

(d) Level 4 (e) Level 3 (f) Level 2

(g) Level 1 (h) Level 0 (i) Input

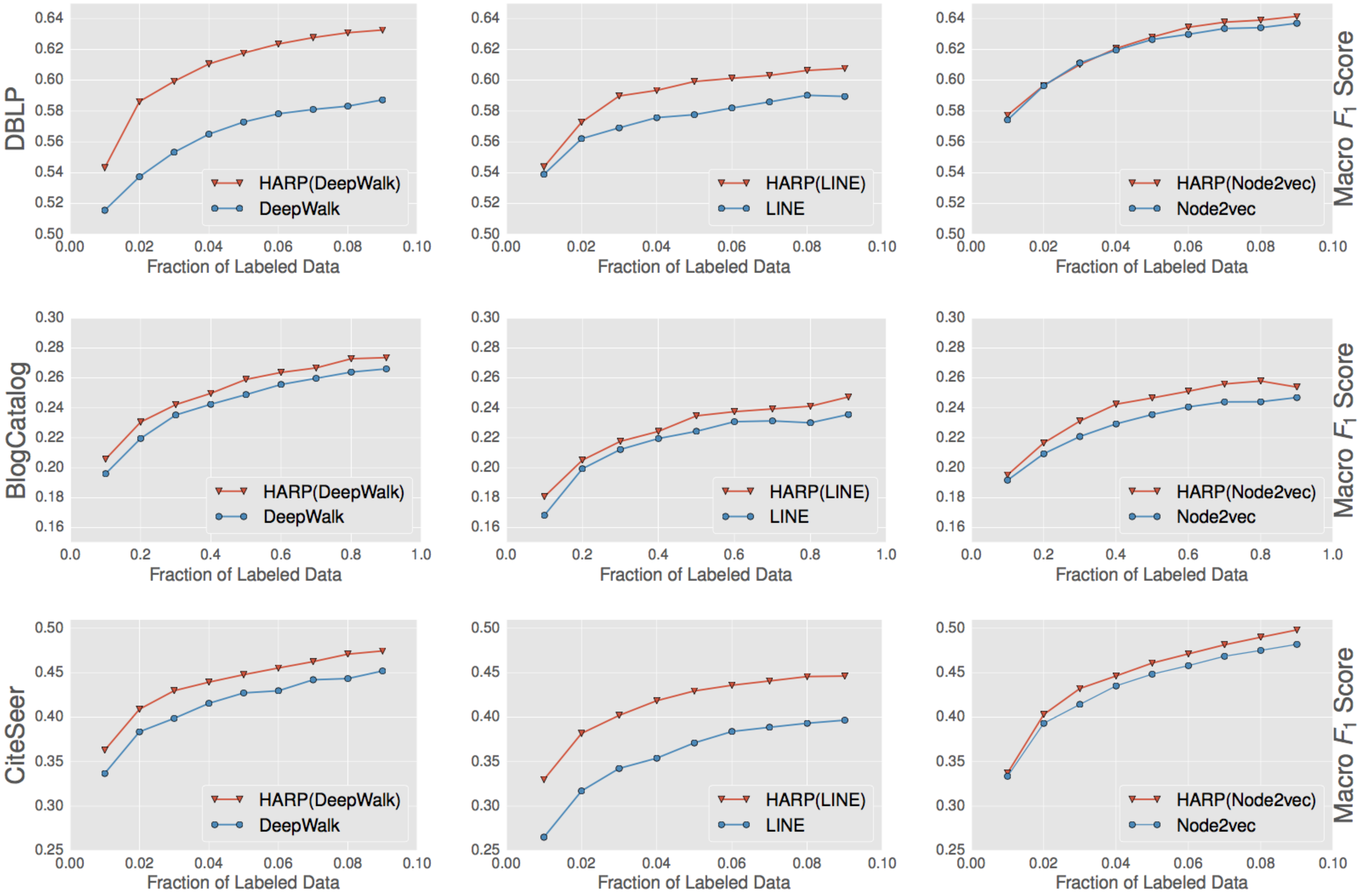
(a) Can_187 (b) LINE (c) HARP



(d) Poisson 2D (e) LINE (f) HARP

Multi-label Classification

Algorithm	Dataset		
	DBLP	BlogCatalog	CiteSeer
DeepWalk	57.29	24.88	42.72
HARP(DW)	**61.76***	**25.90***	**44.78***
Gain of HARP[%]	**7.8**	**4.0**	**4.8**
LINE	57.76	22.43	37.11
HARP(LINE)	**59.51***	**23.47***	**42.95***
Gain of HARP[%]	**3.0**	**4.6**	**13.6**
Node2vec	62.64	23.55	44.84
HARP(N2V)	**62.80**	**24.66***	**46.08***
Gain of HARP[%]	**0.3**	**4.7**	**2.8**


DBLP
BlogCatalog
CiteSeer
Macro F_1 Score
Macro F_1 Score
Macro F_1 Score
Fraction of Labeled Data
HARP(DeepWalk)
DeepWalk
HARP(LINE)
LINE
HARP(Node2vec)
Node2vec

END

THANK YOU