MEGAN: A Generative Adversarial Network for Multi-View Network Embedding

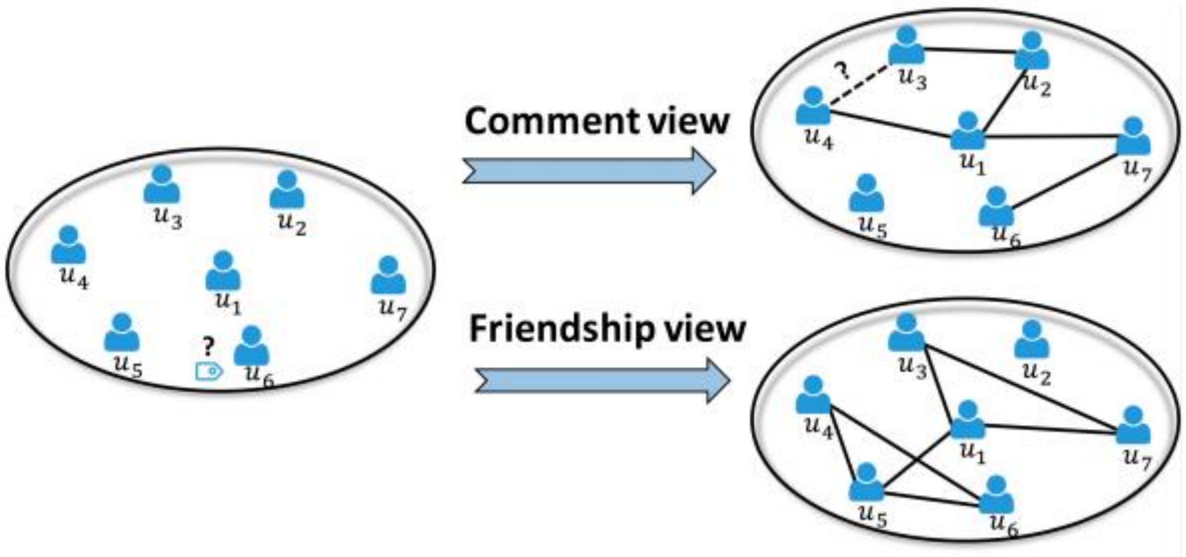
Yiwei Sun Suhang Wang Tsung-Yu Hsieh Xianfeng Tang
Vasant Honavar

IJCAI 2019

Contents

Introduction

Aims to learn low-dimensional, information preserving and typically non-linear repre

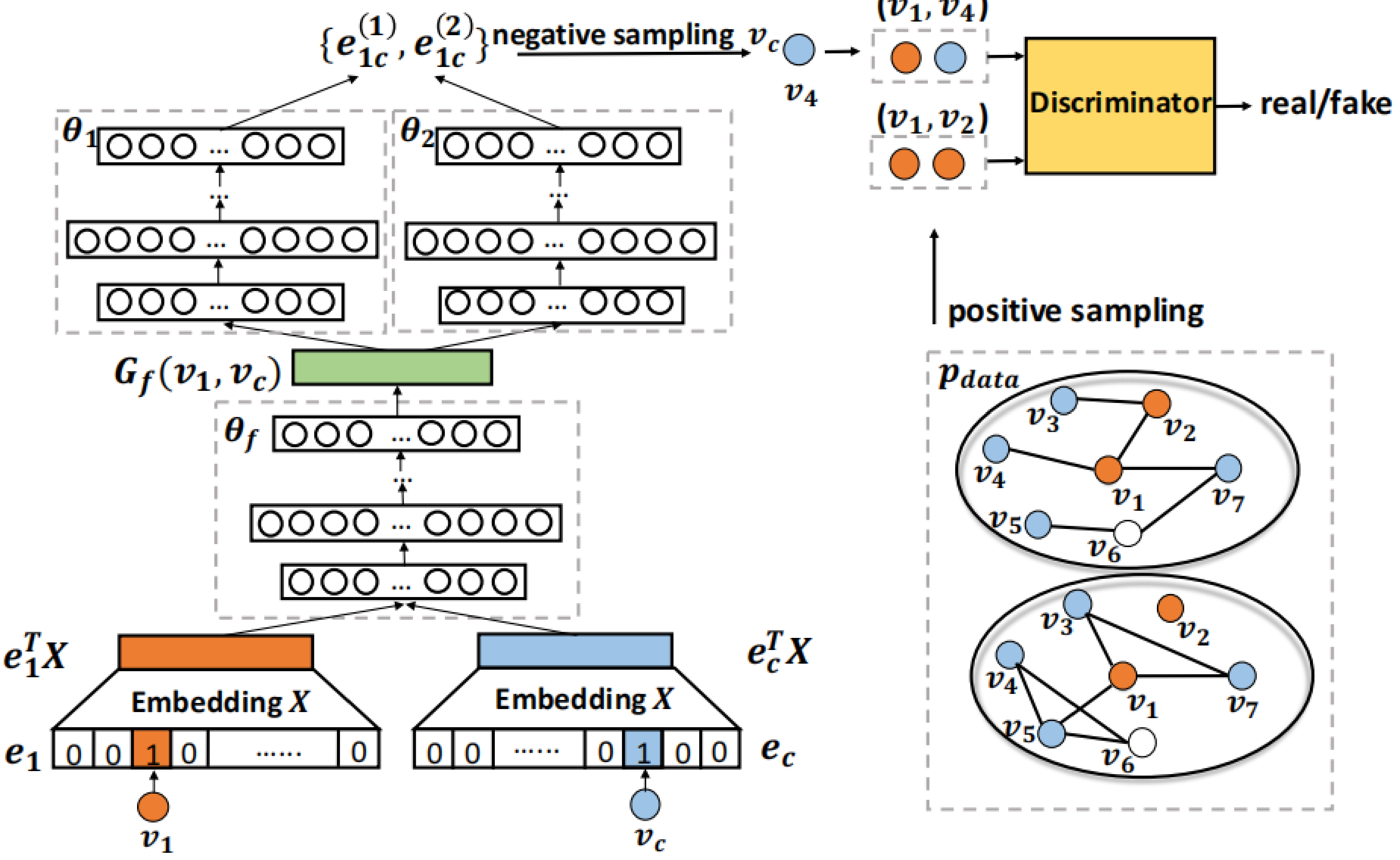


Most of the wor d primarily on single-view netw

Figure 1: The toy multi-view network containing 7 users (nodes) and comprised of comment view and friendship view

Most of the work on multi-view learning focuses on data that lack a network structure

Methods



$\{e_{1c}^{(1)}, e_{1c}^{(2)}\}$
negative sampling
v_c
v_4
(v_1, v_4)
(v_1, v_2)
Discriminator
real/fake
positive sampling
p_{data}
θ_1
θ_2
$G_f(v_1, v_c)$
θ_f
$e_1^T X$
$e_c^T X$
Embedding X
Embedding X
e_1
e_c
v_1
v_c
v_1 v_2 v_3 v_4 v_5 v_6 v_7

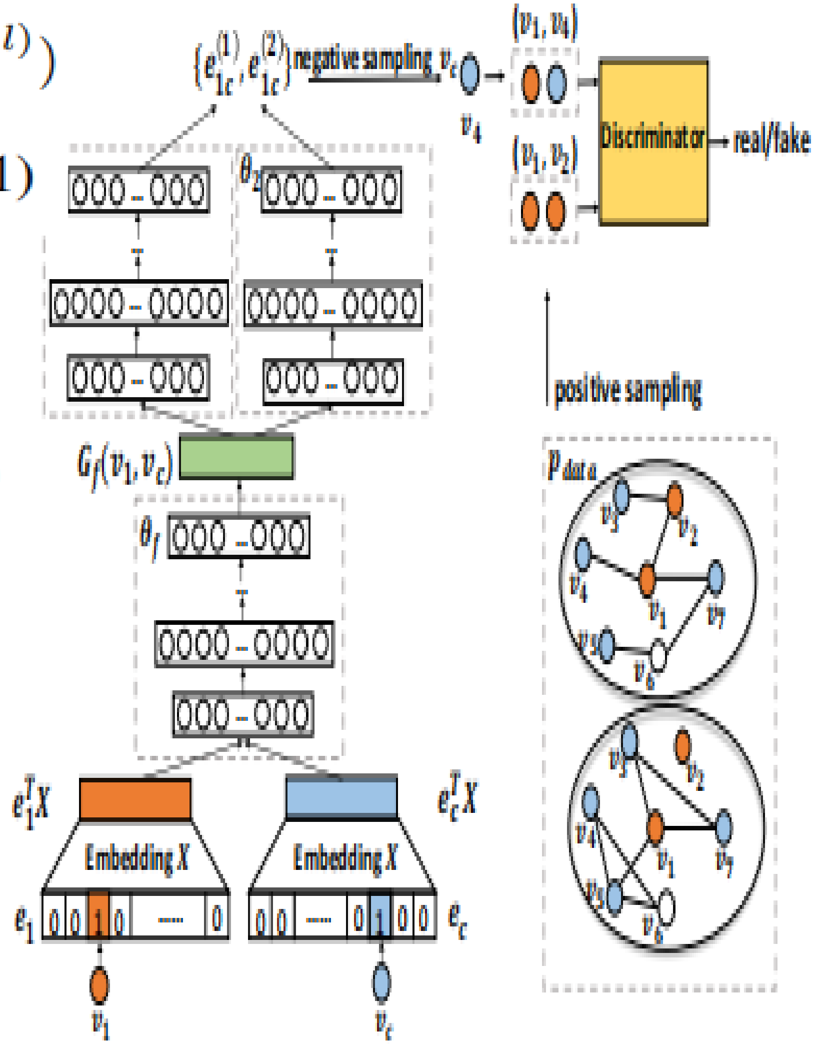
$$G(e_{ij}^{(1)}, \ldots, e_{ij}^{(k)} | v_i, v_j, \theta_G) = \prod_{l=1}^{k} G^{(l)}(e_{ij}^{(l)} | G_f(v_i, v_j; \mathbf{X}, \theta_f); \theta^{(l)})$$

(1)

$$G_f(v_i, v_j; \mathbf{X}, \theta_f)$$

Denote the fused representation

Capture the correlation between vi and vj



Choose the fake node vc:

$$\arg\max_{v_c \in \mathcal{V}} \prod_{l=1}^{k} G^{(l)}(e_{ic}^{(l)} = e_{ij}^{(l)} | v_i, v_j, G_f(v_i, v_j; \mathbf{X}, \theta_f) \quad (2)$$

Node Pair Discriminator

$$D(v_i, v_j) = \frac{\exp(\mathbf{d_i}^T \cdot \mathbf{d_j})}{1 + \exp(\mathbf{d_i}^T \cdot \mathbf{d_j})} \qquad (3)$$

Objective Function of MEGAN

$$\min_{\theta_G} \max_{\theta_D} V(G,D) = \sum_{i=1}^{n} (\mathbb{E}_{(v_i,v_j)\sim p_{\text{data}}}[\log D(v_i, v_j; \theta_D)]$$
$$+ \mathbb{E}_{(v_i,v_c)\sim p_g}[\log(1 - D(v_i, v_c; \theta_D))]) \quad (4)$$

Updating D

$$\nabla_{\theta_D} V(G,D) = \sum_{i=1}^{n} (\mathbb{E}_{(v_i,v_j)\sim p_{\text{data}}}[\nabla_{\theta_D} \log D(v_i, v_j)]$$
$$+ \mathbb{E}_{(v_i,v_c)\sim p_g}[\nabla \theta_D \log(1 - D(v_i, v_c))]) \quad (5)$$

Updating G

$$\nabla_{\theta_G} V(G,D) \longrightarrow \theta_G = \{\mathbf{X}, \theta_f, \theta^{(1)}, \ldots, \theta^{(k)}\}$$

$$= \nabla_{\theta_G} \sum_{i=1}^{n} \mathbb{E}_{(v_i,v_c)\sim p_g}[\log(1 - D(v_i, v_c; \theta_D)]$$

$$= \nabla_{\theta_G} \sum_{i=1}^{n} \sum_{c=1}^{n} G(\mathcal{K}_{ic}|v_i, v_c)[\log(1 - D(v_i, v_c; \theta_D)]$$

$$= \sum_{i=1}^{n} \mathbb{E}_{(v_i,v_c)\sim G}[\nabla_{\theta_G} \log G(\mathcal{K}_{ic}|v_i, v_c) \log(1 - D(v_i, v_c; \theta_D))]$$

$$(6)$$

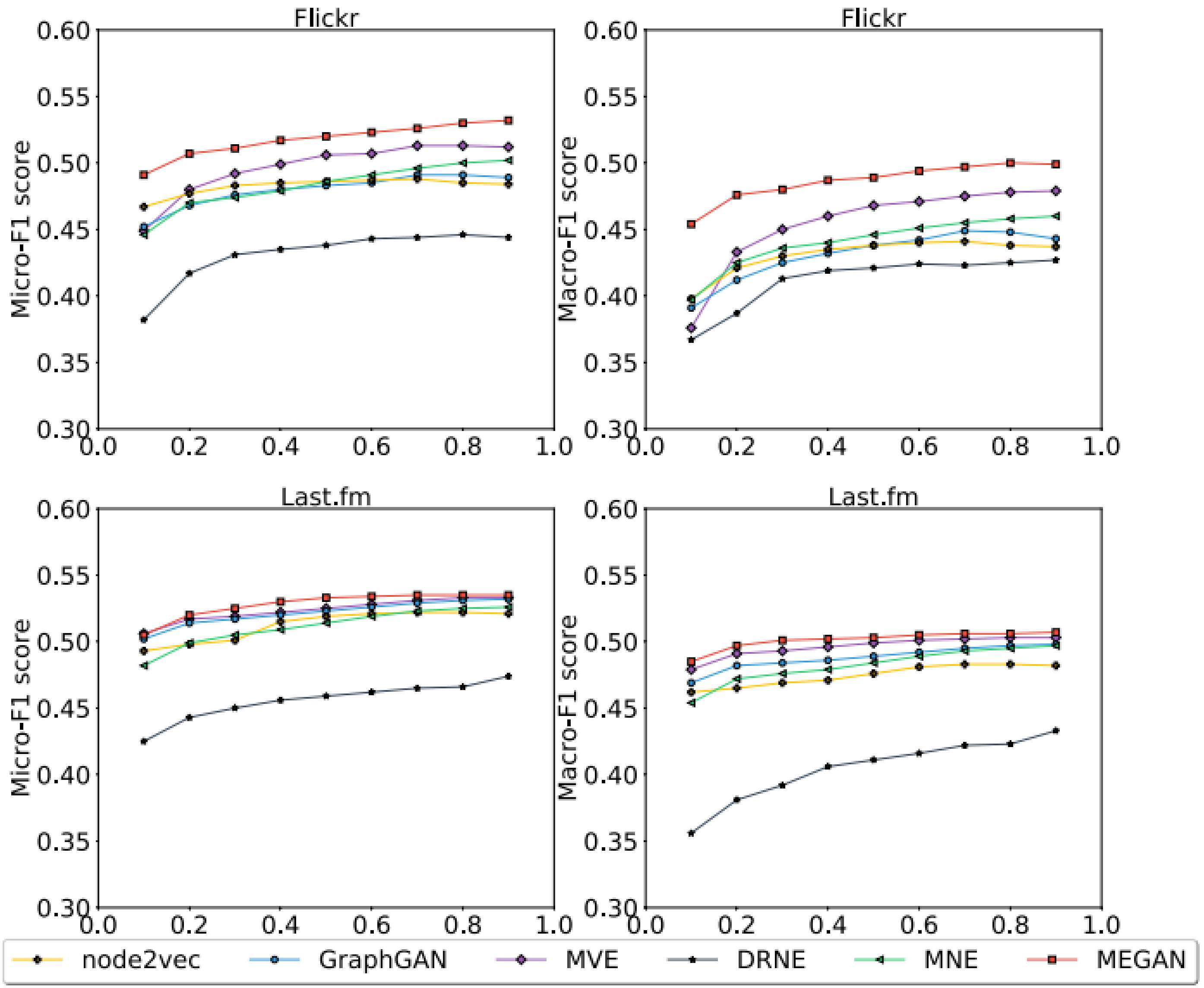
Algorithm 1 MVGAN framework

Require: embedding dimension d, size of discriminating samples t, generating samples s

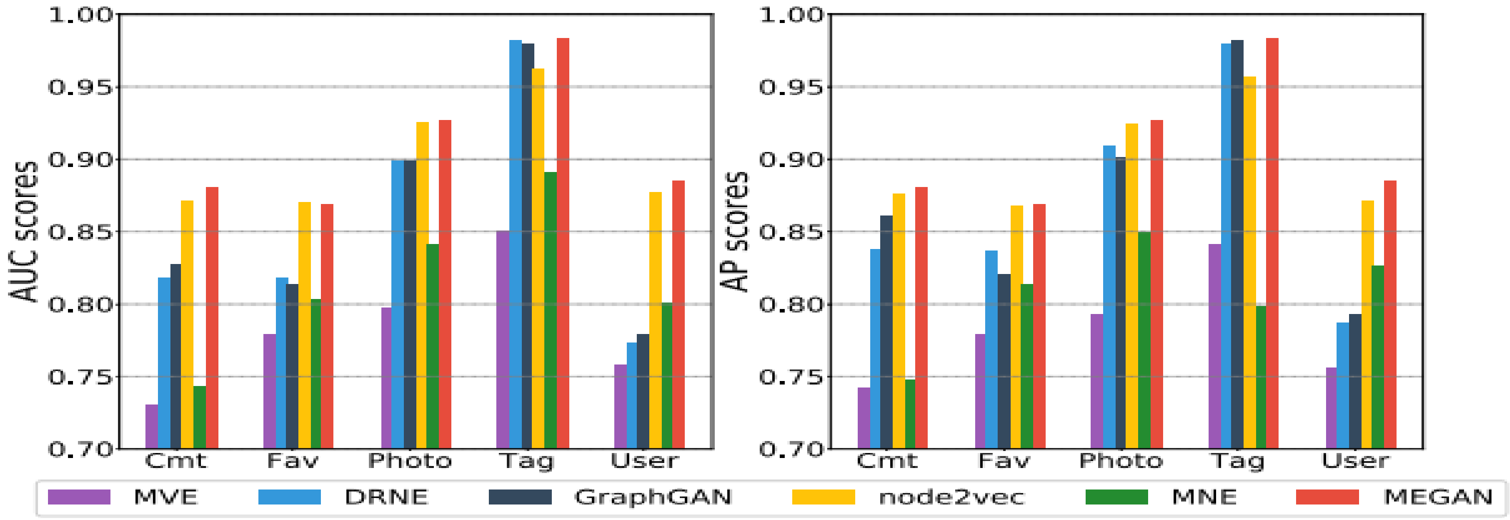
Ensure: θ_D, θ_G

```
Initialize and pre-train D and G
while MVGAN not converged do
    for G-steps do
        Sample s negative pairs of nodes (v_i, v_c) for the given positive pair of nodes (v_i, v_j)
        update θ_G according to Eq.(1) and Eq.(6)
    end for
    for D-steps do
        Sample t positive nodes pairs (v_i, v_j) and t negative node pairs (v_i, v_c) from p_g for each node v_i
        update θ_D according to Eq.(3) and Eq.(5)
    end for
end while
```

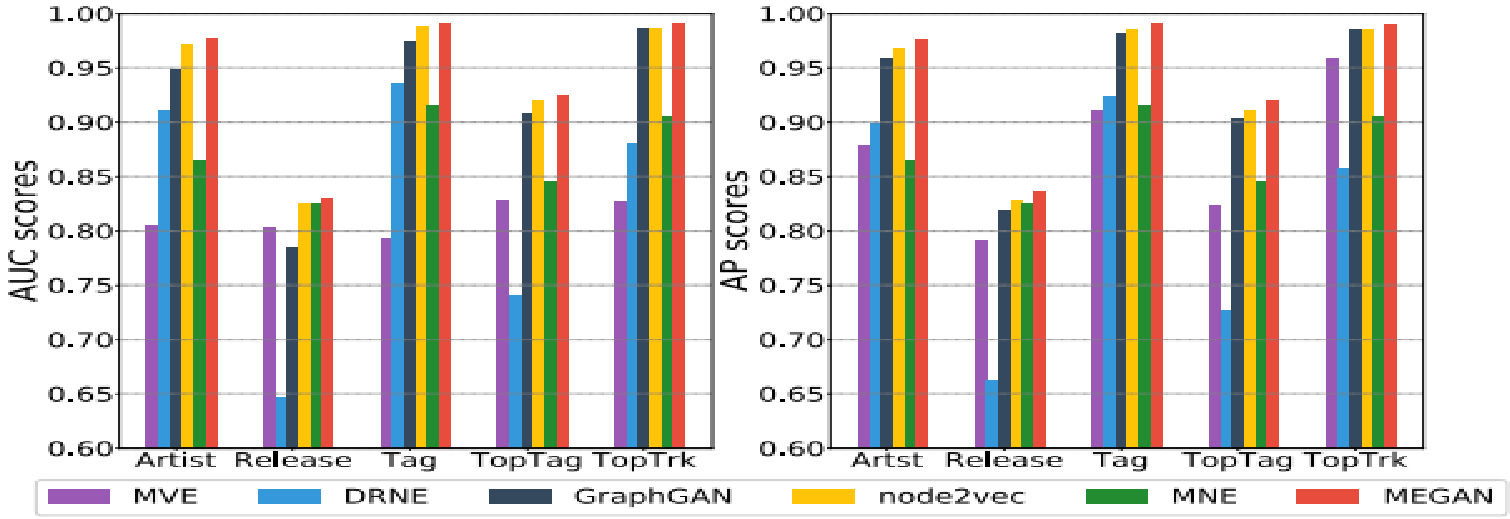
Node classifi-cation task



Link Prediction



(a) Flickr



(b) Last.fm

Network Visualization

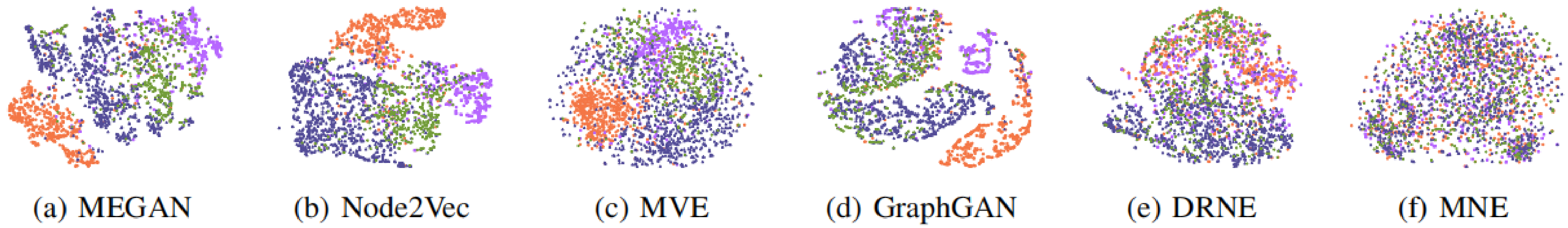
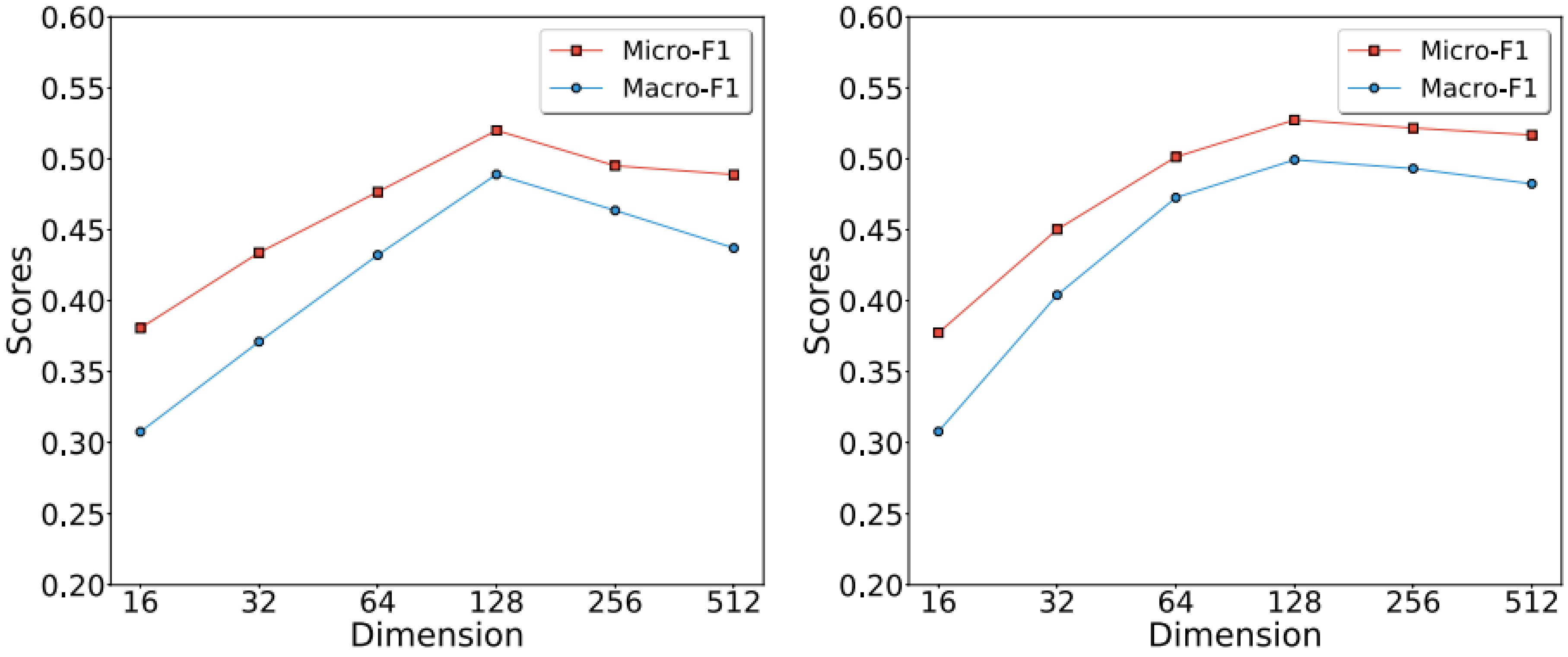


Figure 5: Visualization results of the Flickr. The users are projected into 2D space. Color of the node represents the categories of the user.

Impact of Embedding Dimension



(a) Flickr (b) Last

Figure 6: Parameter sensitivity for node classification on (a) Flickr and (b) Last.fm datasets with varying dimensions d.