2022 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Evaluation of Joint Modeling Techniques for Node Embedding and Community Detection on Graphs

Simon Hiel
KU Leuven, Belgium
simon.hiel@brightwolves.eu

Lore Nicolaers
KU Leuven, Belgium
lore.nicolaers@student.kuleuven.be

Carlos Ortega Vázquez
KU Leuven, Belgium
carloseduardo.ortegavazquez@kuleuven.be

Sandra Mitrović
IDSIA, Switzerland
sandra.mitrovic@idsia.ch

Bart Baesens
KU Leuven, Belgium
University of Southampton, United Kingdom
bart.baesens@kuleuven.be

Jochen De Weerdt
KU Leuven, Belgium
jochen.deweerdt@kuleuven.be

***Abstract*—Novel joint techniques capture both the microscopic context and the mesoscopic structure of networks by leveraging two previously separated fields of research: node representation learning (NRL) and community detection (CD). However, several limitations exist in the literature. First, a comprehensive comparison between these joint NRL-CD techniques is nonexistent. Second, baseline techniques, datasets, evaluation metrics, and classification algorithms differ significantly between each method. Thirdly, the literature lacks a synchronized experimental approach, thus rendering comparison between these methods strenuous. To overcome these limitations, we present a unified experimental setup mutually comparing six joint NRL-CD techniques and comparing them with corresponding NRL/CD baselines in three different settings: non-overlapping and overlapping CD and node classification. Our results show that joint methods underperform on the node classification task but achieve relatively solid results for overlapping community detection. Our research contribution is two-fold: first, we show specific weaknesses of selected joint techniques in different tasks and data sets; and second, we suggest a more thorough experimental setup to benchmark joint techniques with simpler NRL and CD techniques.**

***Index Terms*—Node Representation Learning, Community Detection, Joint Modeling Techniques**

I. Introduction

Graphs are flexible data structures encoding intricate relations between their nodes. They are exploited in many fields (e.g., cheminformatics, biology, transportation, social media, and communication). The usefulness of graphs drives the importance of graph analytics and its two well-known tasks: Node Representation Learning (NRL) and Community Detection (CD).

NRL techniques aim to efficiently transform nodes into a low-dimensional representation that serves as features in predictive tasks. NRL excels at preserving local graph structure [1] while global community structure, a quintessential feature of modern graphs, is mostly ignored [2]. However, it is demonstrated that NRL improves numerous applications, such as node classification, link prediction, and visualization [3]. On the other hand, CD identifies communities (clusters) of nodes that are similar to each other, where intra-group connections are denser than inter-group connections [4]. Communities define a mesoscopic description of network structures and are, therefore, recognized as an essential feature of a network [5].

While NRL techniques cannot consider mesoscopic network structures, CD algorithms have trouble integrating local network structures. Joint modeling techniques recently emerged to overcome the drawbacks of NRL and CD techniques by combining microscopic and mesoscopic network structures in the embedding process [6]. These novel joint techniques show promising results compared to baseline NRL or CD methods. However, as seen from Table I, the literature lacks harmonization: there is no consensus in the selection of baseline techniques, neither in terms of NRL or CD baselines nor among baseline joint NRL-CD methods. Each work presents a different set of baselines. Most joint methods, except for DANMF, indeed use ComE [7] as a baseline joint technique; however, mutual comparisons of joint methods are otherwise almost non-existent. An exception is M-NMF which is confronted with DANMF, whereas GEMSEC is compared to M-NMF and DANMF. Other aspects where the research diverges are the choice of evaluation metrics, datasets, and classification algorithms. Moreover, classification techniques are omitted or differ significantly. For example, both CommunityGan and DANMF do not test classification performance.

We bridge this literature gap by evaluating six joint NRL-CD methods and comparing them with different NRL and CD baselines on three tasks: overlapping CD, non-overlapping CD, and node classification. Our experiments demonstrate that joint techniques are not competitive with GraRep and NetMF on node classification and non-overlapping CD. On the other hand, vGraph and CommunityGAN outperform all other baselines for overlapping community detection.

II. Related NRL, CD and joint NRL-CD works

NRL techniques can be categorized depending on the approach: random walk, matrix factorization, or deep neural network [1]. Over the years, the number of NRL techniques has dramatically increased. Nevertheless, the most relevant

IEEE/ACM ASONAM 2022, November 10-13, 2022
978-1-6654-5661-6/22/$31.00 © 2022 IEEE

TABLE I: Overview of joint NRL-CD methods (right) and their corresponding baselines (left) demonstrating the presence of literature gap. Notation: CGAN: CommunityGAN, DW: DeepWalk, SC: Spectral Clustering; NA: Not Applicable.

Baseline		Joint NRL-CD methods					
Category	Method	ComE	GEMSEC	vGraph	CGAN	M-NMF	DANMF
NRL methods	DW [8]	✓	✓	✓		✓	
	LINE [9]	✓	✓	✓		✓	✓
	Node2Vec [10]	✓	✓	✓	✓	✓	✓
	Walklets [11]		✓				
	NetMF [12]			✓			
	GraRep [13]	✓				✓	
	GraphGAN [14]				✓		
CD methods	k-means		✓	✓			
	SC	✓					
	BigClam [15]			✓			✓
	CESNA [16]			✓			
	Circles [17]			✓			
	SVI [18]			✓			
	MMSB [19]				✓		
	CPM [20]				✓		
	AGM [21]				✓		
Joint methods	ComE [7]	NA	✓	✓	✓	✓	
	GEMSEC [22]	✓	NA			✓	✓
	vGraph [23]	✓		NA			
	CGAN [24]	✓			NA		
	M-NMF [5]	✓	✓			NA	✓
	DANMF [25]		✓			✓	NA

for this work are the following: random walk-based methods DeepWalk [8] and Node2Vec [10] hypothesizing that nodes co-occurring in short random walks should have similar embeddings; their contemporary method LINE [9] preserving the first and second order proximity; GraRep [13] using multiple powers of the adjacency matrix to cover higher-order similarities; Walklets [11] aiming to speed up random walks-based learning by occasionally skipping nodes; NetMF [12] performing explicit matrix factorization and thus unifying DeepWalk, Node2Vec, LINE, and GraphGAN [14].

CD methods are even more numerous and, in general, can be divided into those detecting: non-overlapping, overlapping, and both types of communities [26]. For our work, from a non-overlapping CD perspective, the most relevant methods are well-known techniques of k-means (a partitional clustering method) and spectral clustering (applying k-means on the eigenvalues of the Laplacian matrix of a graph). For overlapping CD, the most relevant works are BigClam [15], assuming that overlaps of communities exhibit denser connections, and CPM [20], presuming that intra-community cliques are more common than inter-community cliques.

TABLE II: Summary of joint NRL-CD methods. Notation: CGAN: CommunityGAN, NNMF: Non-negative Matrix Factorization, Prob: Probabilistic method, NN: Neural Network method, PBH: Population-based Heuristic, NO: Non-Overlapping, O: Overlapping, F1: F1-Score, Acc.: Accuracy, AUC: AUC-ROC, n: number of nodes, m: representation's dimension size, r: number of communities

Method	Approach	Community Type	Eval. Measure	Time Complexity
M-NMF	NNMF	NO	Acc	$\mathcal{O}(n^2 \cdot m + n^2 \cdot k)$
DANMF	NNMF	NO+O	ARI, NMI, Acc	$\mathcal{O}(p(t_p + t_f)(n^2 \cdot r + n \cdot r^2))$
vGraph	Prob	NO+O	F1, NMI	$\mathcal{O}(\mid E \mid \cdot d \cdot max(M, K))$
CGAN	NN	O	F1, NMI	Unavailable
GEMSEC	Prob	O	F1, NMI	$\mathcal{O}((w \cdot k + \mid C \mid) \cdot l \cdot d \cdot \mid V \mid \cdot N)$
ComE	Prob	NO+O	F1, NMI	Unavailable
CNRL	Prob	NO+O	AUC	Unavailable
MemeRep	PBH	NO	F1, NMI	Unavailable
EA-NE	PBH	NO+O	Acc, AUC, NMI	Unavailable

Within existing joint techniques, we discern four main research streams: probabilistic methods, matrix factorization, neural network approaches, and population-based heuristics, as outlined in Table II. Apart from the methods we consider for further evaluation, three additional joint methods are worth mentioning. First, CNRL [2] is a probabilistic method that simultaneously discovers community distributions for every node and learns embeddings for both nodes and communities, allowing for nodes to belong to multiple communities. The other two techniques, MemeRep [27] and EA-NE [28] fall into the category of population-based heuristics. MemeRep utilizes a genetic algorithm combined with a two-level learning strategy (first, from neighbors, second, from communities) to accelerate the optimization of a population of solutions. EA-NE employs a classical genetic algorithm using crossover and mutation operators to optimize an objective function with node embedding and community detection components.

III. Joint NRL-CD Techniques

The advantage of techniques combining NRL and CD is preserving both the local and global graph structure. Here, we focus on the mechanism behind each joint NRL-CD method and its strengths and weaknesses.

A. ComE: A Closed Loop Approach

ComE (COMmunity Embedding framework) [7] jointly solves CD and NRL by introducing community embedding and applying a closed loop approach whereby the idea is that a good node embedding facilitates community detection, which enforces better community embedding. Consequently, the obtained community embedding allows node embeddings to be optimized. Furthermore, using community-aware learning (via community embeddings), the method aims to enforce the preservation of high-order proximity. More precisely, embeddings of nodes belonging to the same community should be closer to each other and additionally close to that community embedding. Given that community, by its definition, represents a group of densely linked nodes, ComE defines community embedding as a multivariate Gaussian distribution $\mathcal{N}(\psi_k, \Sigma_k)$ in a low-dimensional space (here ψ_k denotes the mean vector of the distribution and Σ_k denotes a covariance matrix). This approach allows to perform CD and learn community embedding together in a single objective function based on a Gaussian mixture model:

$$O_{high}(\Phi, \Pi, \Psi, \Sigma) = \frac{-\beta}{K} \sum_{i=1}^{|V|} \log \left(\sum_{k=1}^{K} \pi_{ik} \mathcal{N}(\phi_i | \psi_k, \Sigma_k) \right),$$

where ϕ_i denotes embedding of node $v_i \in V$, and π_{ik} is the probability of node v_i belonging to community k, with K being the total number of communities and $\sum_{k=1}^{K} \pi_{ik} = 1$.

The final objective function $\mathcal{L}$ combines first- and second-order proximity preserving objective functions (defined the same way as in LINE and DeepWalk but with negative sampling) with the high-order preserving O_{high}. The optimization of $\mathcal{L}$ is divided into two phases. Firstly, the parameters (Π, Ψ, Σ) are inferred with a constrained minimization given $(\Phi, \Phi‘)$ using expectation-minimization. Secondly, parameters $(\Phi, \Phi‘)$ are inferred with an unconstrained minimization given (Π, Ψ, Σ) using stochastic gradient descent.

1) Evaluation of ComE: ComE is evaluated on three tasks: Node Classification, CD, and Graph Visualization. The only joint technique used for comparison was M-NMF. When evaluated on node classification using Macro-F1 and Micro-F1 scores, ComE outperforms most other baselines (DeepWalk, LINE, Node2vec, M-NMF), except for GraRep when the percentage of labeled data is low, and the dataset is small. This suboptimal result can be explained by a smaller set of sampled paths with smaller datasets. Concerning CD, ComE consistently outperforms all baseline techniques on both NMI and conductance measures for every dataset.

ComE appears to be an excellent technique for presenting community structure. ComE particularly stands out in identifying nodes that may belong to multiple communities. The visualizations of both node and community embeddings show a clear overlap between communities, while other methods cannot detect this overlap. However, ComE struggles with similar communities with considerable overlap.

B. GEMSEC: Graph Embedding with Self Clustering

GEMSEC (Graph Embedding with Self Clustering) [22], builds on top of the SkipGram method [29] including a so-called clustering cost. It relies on the concept that vertices in the same community have a pronounced overlap between their neighborhoods; however, GEMSEC does not detect overlapping communities. Hence, the GEMSEC objective function:

$$\min_{f,\mu} \mathcal{L} = \sum_{v\in V}\left[\ln\left(\sum_{u\in V}\exp(f(v)\cdot f(u))\right) - \sum_{n_i\in N_S(v)} f(n_i)\cdot f(v)\right]$$
$$+\gamma\cdot\sum_{v\in V}\min_{c\in C}\|f(v)-\mu_c\|_2$$
$$+\lambda\cdot\sum_{(v,u)\in E_S} w_{(v,u)}\cdot\|f(v)-f(u)\|_2$$

combines the standard SkipGram embedding cost (first two terms) with clustering cost (third term) and the regularization term, which serves to smooth the clustering. The first term is the partition function which enforces nodes to be embedded in a low volume space around the origin, while the second term forces nodes with similar sampled neighborhoods to be embedded close to each other [22]. A partition function is approximated with negative sampling for tractability of computation performed by Adam optimizer; this is a form of noise contrastive estimation [30]. The clustering cost minimizes the distance from each node to its nearest cluster center, and its importance is weighted by hyper-parameter γ.

The embedding is computed with prior initialization of model weights, based on the number of vertices, embedding dimensions, and clusters, followed by N repetitions whereby from a shuffled node list, the node sequences are generated, next clustering hyper-parameter γ and learning rate α are updated and accordingly the model weights as well. In the initialization phase, the weight for the clustering cost γ is very small (close to zero) and increases exponentially to 1 at the end of the process (updating by the exponential annealing rule). The initialized weights follow the same uniform distribution to avoid empty clusters.

1) Evaluation of GEMSEC: GEMSEC is evaluated on: clustering quality (by modularity), computational efficiency (by the log of the mean run-time (in seconds)), and node classification (by F1-score). Interestingly, the performance drops for graphs with low transitivity. The transitivity denotes the relative number of triangles in a graph and implies the number of cliques in a graph (e.g., the friend of a friend is also my friend). The computational efficiency shows significant improvements compared to ComE and M-NMF but is on par with DeepWalk. More importantly, the run-time increases linearly with the number of vertices in the graph. This advantage allows the method to run on massive datasets.

On a multi-label node classification as a downstream task, GEMSEC outperforms, on average, all baseline methods. Aside from better performance, GEMSEC offers superior control over community sensitivity in the objective function. Furthermore, it does not depend on the neighborhood sampling method. These extra benefits make GEMSEC applicable to a wider variety of datasets both in industrial and academic setups.

C. vGraph: A Generative Model for Joint CD and NRL

vGraph [23] is a probabilistic generative model based on the assumption that each node can have multiple contexts or communities, and each community is defined as a multinomial distribution over nodes. vGraph generates neighbors for each node, considering the communities as latent variables of the graph. For each node w, a community assignment z is generated following a prior belief about the community structure $p_{\phi,\psi}(z|w)$, where ϕ_w denotes the embedding of node w and ψ_w denotes the embedding of z^{th} community. This community expresses the neighborhood context of node w. Next, w's neighbor c is generated conditional on this generated context z through $c \sim p_{\psi,\varphi}(c|z)$, φ_w denotes the embedding of node w. This two-step process defines the simplified generation process of a graph with edges (w, c) and can be expressed as: $p_{\phi,\varphi,\psi}(c|w) = \sum_z p_{\psi,\varphi}(c|z)p_{\phi,\psi}(z|w)$.

Formally, vGraph aims to maximize the log-likelihood of the observed edges, $\log p_{\phi,\varphi,\psi}(c|w)]$. As this is computationally intractable, the objective is instead shifted to optimizing evidence lower bound (ELBO) and approximating true posterior distribution $p_{\phi,\varphi,\psi}(z|c, w)$ with variational distribution $q(z|c, w)$, which actually represents the community membership of the edge (w, c). Therefore, the ultimate objective function of vGraph is defined as:

$$\min_{\phi,\varphi,\psi} \mathcal{L} = -\log p_{\phi,\varphi,\psi}(z|c,w) =$$
$$-E_{z\sim q(z|c,w)}[\log p_{\psi,\varphi}(c|z)] + KL(q_{\phi,\psi}(z|c,w)\|p_{\phi,\psi}(z|w))$$
$$+\mathcal{L}_{reg}, \text{ where } \mathcal{L}_{reg} = \lambda \sum_{(w,c)\in\epsilon} \alpha_{w,c}\cdot d(p(z|c,p(z|w)))$$

is a smoothness regularization term that encourages the learned community distributions of linked nodes to be similar (for details, please see the original paper), and $KL(\cdot||\cdot)$ represents the Kullback-Leibler divergence between two distributions. Community membership is then determined based on $p_{\phi,\psi}(z|w)$, which is approximated by variational distribution.

1) Evaluation of vGraph: vGraph is evaluated on three tasks: overlapping CD, non-overlapping CD, and node classification. Twenty different datasets and a fairly wide variety of baseline methods are used. For overlapping CD, F1-score and Jaccard similarity metrics are used. By virtue of the parametrization, vGraph should be highly efficient, thereby allowing its use on large graphs. vGraph outperforms all baselines apart from BigClam and relatives. For non-overlapping CD, performance is assessed in terms of Normalized Mutual Information (NMI) and modularity. Here, vGraph proves to outperform most baseline techniques as well as ComE. Finally, by considering global graph structure through community-aware embedding, vGraph presents improvements in performance for node classification in terms of Micro-F1 and Macro-F1 scores.

D. CommunityGAN: CD with Generative Adversarial Nets

CommunityGAN [24] performs NRL while jointly detecting overlapping communities via a specifically constructed Generative Adversarial Net (GAN) based on motifs, a well-established feature for CD. CommunityGAN's embeddings follow the AGM (Affiliation Graph Model), showing the strength of a node's membership in a community.

CommunityGAN framework learns two models, the generator $G(s|v_c;\Theta_G)$, which generates subsets of the set of vertices V that cover v_c and who are most likely real motifs, and the Discriminator $D(s,\Theta_D)$, which estimates the probability that node subset s is a real motif. Then, those two models are combined in a minimax game, where the generator G tries to deceive the discriminator D by trying to perfectly approximate $p_{true}(m|v_c)$ and also generates the most likely node subsets that are similar to the real motifs that cover v_c. Discriminator D, in its turn, tries to differentiate those motifs that belong to $p_{true}(m|v_c)$ and the motifs generated by the generator. This results in the following objective:

$$\min_{\Theta_G}\max_{\Theta_D} V(G,D) = \sum_{c=1}^{V} \mathbb{E}_{m\sim p_{true}(\cdot|v_c)}[logD(m;\theta_D)]$$
$$+\sum_{c=1}^{V} \mathbb{E}_{s\sim G(s|v_c;\Theta_G)}[log(1-D(s;\Theta_D))]$$

The discriminator D is trained with the positive samples from $p_{true}(\cdot|v_c)$ and the negative samples coming from the generator $G(\cdot|v_c;\Theta_G)$. The generator is updated by the policy gradient method under the guidance of the discriminator.

Furthermore, to overcome the problems of dense overlapping, computational intractability, and neglecting neighborhood information which all occur when using sigmoid and softmax functions for discriminator and generator, respectively, CommunityGAN extends AGM from edge to motif generation. Furthermore, both discriminator and generator are redefined using AGM, discriminator as:

$$D(s) = 1 - exp(-\bigodot(d_{v_1},...,d_{v_m})),$$

where $d_v \in \mathbb{R}^C$ denotes the nonnegative C-dimensional representation vectors of vector v and $\bigodot$ means the sum of the entrywise product of d vectors, while Generator $G_v(v_{s_m}|v_{s_1},...,v_{s_{m-1}})$ is redefined as:

$$G_v = \frac{1-exp(-\bigodot(g_{v_{s_1}},...,g_{v_{s_m}}))}{\sum_{v\notin(v_{s_1},...,v_{s_m})} 1-exp(-\bigodot(g_{v_{s_1}},...,g_{v_{s_{m-1}}},g_v))}.$$

where $g_v \in \mathbb{R}^C$ represents the non-negative C-dimensional representation vectors of node v for the Generator, and the union of all those representation vectors g_v is denoted by θ_G. For further redefinition of the Generator and the improvement of remaining shortcomings, we point the interested reader to the original paper.

1) Evaluation of CommunityGAN: CommunityGAN framework is compared to three NRL techniques (Node2Vec, LINE, GraphGan), three CD techniques (MMSB, CPM, AGM), and one joint (ComE). The evaluation is performed using synthetic and real-world data. In the case of the synthetic data, the tasks defined were: solving a dense overlapping problem, efficient motif generation, and discrimination. With the first task, CommunityGAN outperformed its competitors regarding the average F1-score. With the second, it was noted that the size of cliques influences the quality of learned community representations (optimal: below 6).

With real-world datasets, community detection and clique prediction tasks were considered. Using datasets containing ground-truth communities (Amazon, YouTube, DBLP), CommunityGAN outperformed other baselines regarding NMI and F1-score. For clique prediction, datasets without ground-truth communities (arXiv-AstroPh, arXiv-GrQc) were used to show that CommunityGAN has superior performance in case of 3-clique and 4-clique prediction but falls behind its competitors in case of 2-clique prediction.

E. M-NMF: Community Preserving Network Embedding

M-NMF (Modularized Nonnegative Matrix Factorization) [5] exploits the idea that when the microscopic structure link is weak due to sparsity issues, the mesoscopic structure will ensure that nodes within the same community are more similar than nodes from different communities. M-NMF models the mesoscopic community structure by employing the modularity maximization-based community detection method [31]. To generalize to more than two communities, modularity is defined as $Q = tr(H^TBH), s.t.tr(H^TH) = n,$

where $H \in \mathbb{R}^{nxk}$ is the community membership indicator (having one column per community and only one element =1 per row, which results in the constraint $tr(H^TH) = n$), $B \in \mathbb{R}^{nxn}$ is the modularity matrix defined as $B_{ij} = A_{ij} - \frac{k_ik_j}{2e}$ and $tr(X)$ denotes the trace of matrix X.

For modeling the microscopic structure, the similarity matrix S is defined by $S = S^{(1)} + \eta S^{(2)}$, where $S^{(1)}$ is the first-order similarity and $S^{(2)}$ is the second-order similarity defined by:

Here $||X||$ denotes the norm of vector X and $N_i = (S^{(1)}_{i,1}, ..., S^{(1)}_{i,n})$ is the first order proximity between the node i and all the other nodes. Hence, the second-order proximity is defined by the similarities between N_i and N_j. Considering only the first-order similarity would not suffice as not having an edge between the two nodes does not imply that they do not have similarities [9]. Next, the method uses non-negative matrix factorization to approximate the similarity matrix S, which results in the following objective function: $\min ||S - MU^T||_F^2,\ s.t.M \geq 0, U \geq 0.$

Finally, to jointly optimize micro- and mesoscopic objectives, a community representation (an auxiliary non-negative matrix) $C \in \mathbb{R}^{kxm}$ is introduced, leading to the final objective:

$$\min_{M,U,H,C} ||S - MU^T||_F^2 + \alpha||H - UC^T||_F^2 - \beta tr(H^TBH),$$
$$s.t.\ M \geq 0, U \geq 0, H \geq 0, C \geq 0, tr(H^TH) = n,$$

with α and β positive parameters for defining the contribution of the corresponding terms [5].

1) Evaluation of M-NMF: M-NMF is evaluated on two different tasks: node clustering and node classification. Nine different networks are used of variable sizes, and M-NMF is compared with multiple NRL baseline models. The evaluation of node clustering is executed by applying k-means to the learned node embeddings and assessed in terms of accuracy. M-NMF outperforms DeepWalk and LINE on all nine networks. However, there is only one dataset where GraRep outperforms M-NMF and only one other dataset where Node2Vec outperforms M-NMF. N-NMF outperformed all but one method for node classification on all nine networks (LINE2 outperformed M-NMF on only one of the nine networks).

F. DANMF: Using a deep autoencoder for joint learning

DANMF (Deep Autoencoder-like Nonnegative Matrix Factorization) [25] relies on the non-negative matrix factorization (NMF) and encoder-decoder architecture, whereby the encoder transforms the network into the community membership space. At the same time, the decoder reconstructs the network from the community membership space. Given that these transformations can contain rather complex hierarchical and structural information with implicit lower-level hidden attributes, they propose using multiple layers of abstraction and, consequently, deep architecture (deep autoencoder). Concretely, instead of factorizing the adjacency matrix $\mathbf{A}$ of a given network into two non-negative matrices $\mathbf{U}$ (whose columns correspond to communities) and $\mathbf{V}$ (capturing community membership of node), the paper proposes factorization into p + 1 non-negative factor matrices $\mathbf{A} \approx \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_p\mathbf{V}_p$, hoping that each of the factors would add an extra layer of abstraction. The hierarchy of p levels of abstraction is given by the following factorizations:

$$\begin{aligned} \mathbf{V}_{p-1} &\approx \mathbf{U}_p\mathbf{V}_p, \\ &\cdots \\ \mathbf{V}_2 &\approx \mathbf{U}_3 \cdots \mathbf{U}_p\mathbf{V}_p, \\ \mathbf{V}_1 &\approx \mathbf{U}_2 \cdots \mathbf{U}_p\mathbf{V}_p. \end{aligned} \tag{1}$$

where $\mathbf{V}_p \geq 0$.

The objective function of the deep autoencoder-like NMF model is defined as:

$$\min_{U_iV_p} \mathcal{L} = \mathcal{L}_D + \mathcal{L}_E + \lambda\mathcal{L}_{ref} =$$
$$||\mathbf{A} - \mathbf{U}_1\mathbf{U}_2 \cdots \mathbf{U}_p\mathbf{V}_p||_F^2 + ||\mathbf{V}_p - \mathbf{U}_p^T \cdots \mathbf{U}_2^T\mathbf{U}_1^T\mathbf{A}||_F^2 +$$
$$\lambda tr(\mathbf{V}_p\mathbf{L}\mathbf{V}_p^T), s.t.\mathbf{V}_p \geq 0, \mathbf{U}_i \geq 0, \forall i = 1, 2, ..., p.$$

where $\mathcal{L}_D$ is decoder loss, $\mathcal{L}_E$ is encoder loss, $\mathcal{L}_{reg}$ is regularization factor with λ being regularization parameter and $\mathbf{L} = D - A$ the Laplacian matrix.

1) Evaluation of DANMF: DANMF is compared to shallow NMF-based methods, BigClam, and three network embedding methods: LINE, Node2Vec, and M-NMF. Both datasets with disjoint communities and datasets with overlapping communities are considered for evaluation.

For the disjoint communities case, five different datasets with variable sizes were evaluated on Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and Accuracy (Acc.). Using three hidden layers, DANMF outperforms all the competitors on all the datasets in terms of ARI, except for the Wiki dataset. Regarding NMI and Acc, DANMF outperforms all the other metrics on all the datasets.

For overlapping communities, the LFR toolkit [32] is used to generate synthetic networks (5000 nodes, an average degree of 20) with overlapping structures where community size varies between 100 and 250. As a result, DANMF outperforms all the other frameworks for all the synthetic networks in the Overlapping NMI (ONMI) metric.

IV. Experimental Setup

A. Data Sets & Computational Limitations

We select eight well-known data sets that have been used in the literature. For the selection of data sets, we consider four factors to analyze the performance of algorithms on a diverse set of graphs: the size of the graph, the density of the graph; whether the communities are overlapping or non-overlapping; and the availability of the ground truth. Table III shows the characteristics of the data sets. The number of communities and if they are overlapping determine which classification task

TABLE III: Summary of data sets. Notation: # of Comm.: Number of communities, GIT: Github, TWI: Twitch, CIT: Citeseer, WEB: Web Spam Detection, YOU: YouTube, DBL: DBLP, AMA: Amazon, DEE: Deezer

Dataset	Nodes	Edges	Density	Overlapping	# of Comm.	Class. Task
GIT	37700	289003	0.00041	No	2	Binary
TWI	7126	35324	0.00139	No	2	Binary
CIT	3300	4500	0.00085	No	6	Multi-class
WEB	9072	473854	0.01152	No	3	Multi-class
YOU[1]	8022	40334	0.00125	No	10	Multi-label
DBL[1]	57360	234169	0.00008	Yes	100	Multi-label
AMA[1]	3225	10288	0.00198	Yes	100	Multi-label
DEE	47538	222887	0.00020	Yes	80	Multi-label

[1] The data set is sampled based on a random selection of communities.

is more appropriate. GIT and TWI data sets have two non-overlapping communities, which lead to a standard binary classification. In contrast, four data sets (YOU, DBL, AMA, and DEE) contain each several overlapping communities; hence a more suitable task is multi-label classification.

We conduct a sampling strategy in three data sets (YOU, DBL, and AMA) to decrease the computational cost of the experiments. The sampling consists of randomly selecting communities instead of nodes. Since joint NRL-CD techniques incorporate community information in the node embeddings and community detection is an essential task in this work, we find this approach most suitable for our analysis of these tasks. For each data set, we perform three repetitions of a holdout validation that splits the data into a training set (70%) and a test set (30%).

The Azure ML environment was used to run the experiments with an Nvidia Tesla K80 and 56 GB of RAM. Unfortunately, this infrastructure posed memory limitations when running GraRep and vGraph on larger datasets.

B. Baselines

For the NRL baselines, we select DeepWalk [8], NetMF [12], and GraRep [33]. The selected NRL baselines are popular in works that propose joint techniques. NRL baselines require a downstream classifier for node classification: a logistic regression with $L2$ regularization is used. For multi-class node classification, logistic regression allows for one-vs-rest strategy in `scikit-learn`. For the multi-label setting, the binary relevance method is used [34]. This method transforms the problem into multiple binary settings, creating a new dataset for each label each time. Likewise, for non-overlapping community detection, NRL baselines use k-means to create the predicted communities from the node embeddings.

Spectral Clustering [35], BigClam [15], and CPM [20] are selected for the CD baselines. Spectral Clustering and BigClam, together with the NRL baselines, are used for non-overlapping community detection. For overlapping community detection, only CPM and BigClam can be used.

C. Joint Modelling Techniques

The selection of joint modeling techniques is based on their relevance in the literature and the availability for implementation. Furthermore, some algorithm implementations require an extra threshold function. Both vGraph and CommunityGAN have a default setting to detect overlapping clusters. To implement these algorithms, the threshold function below is used [24]. Unlike vGraph and CommunityGAN, BigClam and DANMF have a default setting to detect non-overlapping clusters. BigClam and DANMF can find overlapping clusters, but the implementations lack a threshold function. This threshold function alters the found embeddings to represent community memberships. For the metrics, baselines, and two joint techniques (M-NMF and DANMF), we utilized the work by [36], [37] and [38] to implement the code. To implement GemSec, vGraph, and CommunityGAN, we utilize the source code from [39], [40], and [41], respectively.

TABLE IV: Normalized Mutual Information for Non-Overlapping CD

Joint Mod. Techniques	CIT	GIT	TWI	WEB
M-NMF	0.036	0.043	0.007	**0.060**
GemSec	0.048	0.012	0.005	0.051
DANMF	0.137	**0.071**	0.006	0.048
vGraph	0.038	-	0.002	-
CommunityGAN	0.005	0.000	0.000	0.000
Baselines				
Deepwalk + k-means	0.093	0.000	**0.009**	0.031
NetMF + k-means	0.103	0.001	0.001	0.053
GraRep + k-means	**0.197**	-	0.006	0.038
Spectral Clustering	0.006	0.000	0.000	0.000
BigClam	0.002	0.000	0.000	0.001

Best performing model is in **bold and underlined** per dataset.
"-" stands for out-of-memory error

V. Results and Discussion

A. Non-overlapping Community Detection

The metrics NMI and ARI are shown in Table IV and V respectively, in which the best performing method is shown in bold and underlined. DANMF, M-NMF, and GraRep + k-means are the techniques that have been more times the best-performing model in terms of NMI and ARI. We can also observe that joint modeling techniques do not dominate the baselines. However, in GIT, the largest data set with non-overlapping communities, joint techniques substantially outperform the baselines: DAMNF and M-NMF obtained the highest score in NMI and ARI.

B. Overlapping Community Detection

Three metrics (ONMI, Omega index, and F1-score) are considered for graphs with overlapping communities. Notice the ONMI index and Omega index are the extensions of NMI and ARI index, respectively. Table VI, Table VII, and Table VIII show the results. In this task, joint techniques generally dominate the baselines. vGraph and CommunityGAN are the most successful joint techniques as they have outperformed six times the rest of the techniques.

TABLE V: Adjusted Rand Index for Non-Overlapping CD

Joint Mod. Techniques	CIT	GIT	TWI	WEB
M-NMF	0.030	**0.056**	0.009	0.017
GemSec	0.001	0.012	0.004	0.001
DANMF	**0.102**	0.031	0.005	0.002
vGraph	0.030	-	0.004	-
CommunityGAN	0.001	0.000	0.000	0.000
Baselines				
Deepwalk + k-means	0.036	0.002	0.004	0.011
NetMF + k-means	0.003	0.010	0.002	**0.024**
GraRep + k-means	0.070	-	**0.010**	0.013
Spectral Clustering	0.003	0.000	0.000	0.004
BigClam	0.000	0.000	0.000	0.001

Best performing model is in **bold and underlined** per dataset.
"-" stands for out-of-memory error

TABLE VI: Overlapping Normalized Mutual Information for Overlapping CD

Joint Mod. Techniques	YOU	DBL	AMA	DEE
DANMF	0.036	0.001	0.013	**0.001**
vGraph	**0.358**	-	**0.289**	-
CommunityGAN	0.002	**0.283**	0.024	0.000
Baselines				
CPM	0.006	0.000	0.152	**0.001**
BigClam	0.002	0.000	0.005	0.000

Best performing model is in **bold and underlined** per dataset.
"-" stands for out-of-memory error

TABLE VII: Omega Index for Overlapping CD

Joint Mod. Techniques	YOU	DBL	AMA	DEE
DANMF	0.021	0.033	0.021	0.018
vGraph	**0.806**	-	0.952	-
CommunityGAN	0.754	**0.927**	**0.953**	0.175
Baselines				
CPM	0.772	0.925	**0.953**	**0.182**
BigClam	0.380	0.435	0.216	0.113

Best performing model is in **bold and underlined** per dataset.
"-" stands for out-of-memory error

TABLE VIII: F1-Score for Overlapping CD

Joint Mod. Techniques	YOU	DBL	AMA	DEE
DANMF	0.097	0.175	0.032	0.212
vGraph	**0.616**	-	**0.530**	-
CommunityGAN	0.276	0.080	0.386	**0.439**
Baselines				
CPM	0.052	0.004	0.003	0.006
BigClam	0.215	**0.199**	0.058	0.236

Best performing model is in **bold and underlined** per dataset.
"-" stands for out-of-memory error

TABLE IX: AUC-ROC of Node Classification in Binary, Multi-Class, and Multi-Label Tasks

	Binary Task		Multi-Class Task		Multi-Label Task			
Joint Methods	GIT	TWI	CIT	WEB	YOU	DBL	AMA	DEE
M-NMF	0.703	0.582	0.787	0.657	0.944	0.919	0.985	0.501
GemSec	0.641	0.521	0.741	0.650	0.941	0.823	0.945	0.498
DANMF	0.530	0.551	0.638	0.527	0.907	0.603	0.990	0.499
vGraph	-	**0.834**	**0.997**	-	0.947	-	0.987	-
CommunityGAN	0.500	0.503	0.537	0.521	0.939	0.868	0.690	**0.950**
Baselines								
Deepwalk + LR	0.697	0.584	0.712	0.649	0.919	0.808	0.963	0.503
NetMF + LR	**0.706**	0.582	0.787	0.651	0.944	0.919	0.985	0.501
GrapRep + LR	-	0.568	0.813	**0.661**	**0.952**	**0.937**	**0.995**	-

Best performing model is in **bold and underlined** per dataset
"-" stands for out-of-memory error

C. Node Classification

For node classification, Table IX shows the techniques' performance in terms of AUC-ROC. Joint techniques are dominated by GraRep in the multi-label task. However, in the other tasks, the results are more nuanced. In smaller networks, such as TWI and CIT, joint methods show a significant advantage because the highest-scoring baseline is outperformed by a difference larger than 10%. Similar to overlapping community detection, vGraph and CommunityGAN are the best-performing joint techniques.

VI. CONCLUSION & FUTURE WORK

We contribute to the literature by introducing a more comprehensive benchmark comparing six joint techniques to well-known baselines across different learning tasks and datasets. This benchmark study highlights some weaknesses of NRL-CD joint techniques that were not addressed in previous works. More specifically, we found that the joint techniques outperform the baselines only in overlapping community detection and fail to meet expectations in node classification and non-overlapping community detection. These empirical results prove the necessity of conducting more comprehensive benchmarks. Our work is a first attempt to provide a more systematic benchmark study for joint techniques. Despite the ability of the joint techniques to capture both the microscopic and mesoscopic structure of networks, we encourage the practitioner to compare with simpler baselines.

A possible direction for future work can be to compare joint techniques in more complex settings that include temporal and heterogeneous networks instead of purely homogeneous networks.

REFERENCES

[1] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation Learning on Graphs: Methods and Applications," pp. 1–24, 2017. [Online]. Available: http://arxiv.org/abs/1709.05584

[2] C. Tu, X. Zeng, H. Wang, Z. Zhang, Z. Liu, M. Sun, B. Zhang, and L. Lin, "A Unified Framework for Community Detection and Network Representation Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 6, pp. 1051–1065, 2019.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[4] M. Girvan and M. E. Newman, “Community structure in social and biological networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–7826, 6 2002. [Online]. Available: www.pnas.orgcgidoi10.1073pnas.122653799

[5] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu, and S. Yang, “Community preserving network embedding,” *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pp. 203–209, 2017.

[6] S. Fortunato and D. Hric, “Community detection in networks: A user guide,” *Physics Reports*, vol. 659, pp. 1–44, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2016.09.002

[7] S. Cavallari, V. W. Zheng, H. Cai, K. C. C. Chang, and E. Cambria, “Learning community embedding with community detection and node embedding on graphs,” *International Conference on Information and Knowledge Management, Proceedings*, vol. Part F1318, pp. 377–386, 2017.

[8] B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, 2014.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: Large-scale information network embedding,” *WWW 2015 - Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, 2015.

[10] A. Grover and J. Leskovec, “Node2Vec,” pp. 855–864, 2016.

[11] B. Perozzi, V. Kulkarni, H. Chen, and S. Skiena, “Don’t walk, skip! online learning of multi-scale network embeddings,” *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2017*, pp. 258–265, 2017.

[12] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, “Network Embedding as Matrix Factorization,” pp. 459–467, 2018.

[13] S. Cao, W. Lu, and Q. Xu, “GraRep: Learning graph representations with global structural information,” *International Conference on Information and Knowledge Management, Proceedings*, vol. 19-23-Oct-, pp. 891–900, 2015.

[14] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, “Graphgan: Graph representation learning with generative adversarial nets,” *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pp. 2508–2515, 2018.

[15] J. Yang and J. Leskovec, “Overlapping community detection at scale: A nonnegative matrix factorization approach,” *WSDM 2013 - Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pp. 587–596, 2013.

[16] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” in *2013 IEEE 13th international conference on data mining*. IEEE, 2013, pp. 1151–1156.

[17] J. McAuley and J. Leskovec, “Discovering social circles in ego networks,” *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, pp. 1–30, 2014.

[18] P. K. Gopalan and D. M. Blei, “Efficient discovery of overlapping communities in massive networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 110, no. 36, pp. 14 534–14 539, 2013.

[19] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, “Mixed membership stochastic blockmodels,” *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, vol. 9, no. 2008, pp. 34–41, 2009.

[20] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” *Nature*, vol. 435, no. 7043, pp. 814–818, 2005.

[21] J. Yang and J. Leskovec, “Community-affiliation graph model for overlapping network community detection,” *Proceedings - IEEE International Conference on Data Mining, ICDM*, vol. 1, pp. 1170–1175, 2012.

[22] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton, “GemSec: Graph embedding with self clustering,” *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2019*, pp. 65–72, 2019.

[23] F. Y. Sun, M. Qu, J. Hoffmann, C. W. Huang, and J. Tang, “vGraph: A generative model for joint community detection and node representation learning,” *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, pp. 1–11, 2019.

[24] Y. Jia, Q. Zhang, W. Zhang, and X. Wang, “CommunityGan: Community detection with generative adversarial nets,” *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, pp. 784–794, 2019.

[25] F. Ye, C. Chen, and Z. Zheng, “Deep autoencoder-like nonnegative matrix factorization for community detection,” *International Conference on Information and Knowledge Management, Proceedings*, pp. 1393–1402, 2018.

[26] E. S. Negara, R. Andryani, and U. B. Darma, “A review on overlapping and non-overlapping community detection detection algorithms for social network analytics,” vol. 18, no. 1, pp. 1–27, 2018.

[27] M. Gong, S. Member, C. Chen, Y. Xie, and S. Wang, “Community Preserving Network Embedding Based on Memetic Algorithm,” vol. 4, no. 2, pp. 108–118, 2020.

[28] M. Li, J. Liu, P. Wu, and X. Teng, “Evolutionary Network Embedding Preserving Both Local Proximity and Community Structure,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 523–535, 2020.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp. 1–12, 2013.

[30] M. Gutmann and A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[31] M. E. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, no. 3, 2006.

[32] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, 2009.

[33] J. Cao, D. Jin, L. Yang, and J. Dang, “Incorporating network structure with node contents for community detection on large networks using deep learning,” *Neurocomputing*, vol. 297, pp. 71–81, 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2018.01.065

[34] O. Luaces, J. Díez, J. Barranquero, J. J. del Coz, and A. Bahamonde, “Binary relevance efficacy for multilabel classification,” *Progress in Artificial Intelligence*, vol. 1, no. 4, pp. 303–313, 2012.

[35] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[36] F. Pedregosa, V. Michel, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, J. Vanderplas, D. Cournapeau, F. Pedregosa, G. Varoquaux, A. Gramfort, B. Thirion, O. Grisel, V. Dubourg, A. Passos, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” Tech. Rep. 85, 2011. [Online]. Available: http://scikit-learn.sourceforge.net.

[37] B. Rozemberczki, O. Kiss, and R. Sarkar, “Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs,” *International Conference on Information and Knowledge Management, Proceedings*, pp. 3125–3132, 3 2020. [Online]. Available: http://arxiv.org/abs/2003.04819

[38] G. Rossetti, L. Milli, and R. Cazabet, “CDLIB: a python library to extract, compare and evaluate communities from complex networks,” *Applied Network Science*, vol. 4, no. 1, 12 2019.

[39] B. Rozemberczki, “GitHub - benedekrozemberczki/GEMSEC: The TensorFlow reference implementation of ’GEMSEC: Graph Embedding with Self Clustering’ (ASONAM 2019).” 2019. [Online]. Available: https://github.com/benedekrozemberczki/GEMSEC

[40] F. Y. Sun, “GitHub - fanyun-sun/vGraph: Code for NeurIPS paper ”vGraph: A Generative Model for Joint CommunityDetection and Node Representation Learning”,” 2019. [Online]. Available: https://github.com/fanyun-sun/vGraph

[41] S. Jia, “GitHub - SamJia/CommunityGAN: A tensorflow implementation of CommunityGAN (Community Detection with Generative Adversarial Nets),” 2019. [Online]. Available: https://github.com/SamJia/CommunityGAN