Learning Node Embeddings with Exponential Family Distributions

Abdulkadir Çelikkanat
Université Paris-Saclay
CentraleSupélec, Inria
abdcelikkanat@gmail.com

Fragkiskos D. Malliaros
Université Paris-Saclay
CentraleSupélec, Inria
fragkiskos.me@gmail.com

Abstract

Representing networks in a low dimensional latent space is a crucial task with many interesting application in graph learning problems, such as link prediction and node classification. A widely applied network representation learning paradigm is based on the combination of random walks with the traditional *Skip-Gram* approach, modeling center-context node relationships. In this paper, we emphasize on exponential family distributions to capture rich interaction patterns between nodes in random walk sequences. We introduce the generic *exponential family graph embedding* (EFGE) model, that generalizes random walk-based network representation learning techniques to exponential family conditional distributions. Our experimental evaluation demonstrates that the proposed technique outperforms well-known baseline methods in two downstream machine learning tasks.

1 Introduction

Graphs or networks have become ubiquitous as data from diverse disciplines can naturally be represented as graph structures. A recent paradigm in network analysis, known as *network representation learning* (NRL) [5], aims at finding vector representations of nodes (i.e., *node embeddings*), in such a way that the structure of the network and its various properties are preserved in the lower dimensional representation space. That way, after obtaining the embeddings, the learned features can further be used in any downstream machine learning task, such as classification and prediction.

Random walk-based methods have become a prominent line of research [12, 4], being inspired from the field of natural language processing (NLP) [9]. Typically, those methods sample a set of random walks on the input graph, treating them as the equivalent of sentences in natural language, while the nodes visited by the walk are considered as words. Then, widely used NLP models, such as the *Skip-Gram* model [9], are used to learn node latent representations, examining *simple co-occurrence relationships* of nodes within the set of random walk sequences.

Nevertheless, *Skip-Gram* models the conditional distribution of nodes within a random walk based on the *softmax* function, which might prohibit to capture richer types of interaction patterns among nodes that co-occur within a random walk. In this work, we argue that considering more expressive *conditional probability models* to relate nodes within a random walk sequence, might lead to more informative representations. We capitalize on *exponential family distribution* models to capture interactions between nodes. The main contributions of the paper can be summarized as follows:

- We introduce a novel approach, referred to as EFGE, which generalizes classical *Skip-Gram*-based models to exponential family distributions.
- We show that the objectives of existing models, including word embedding in NLP [9] and overlapping community detection [18], can be reinterpreted under the EFGE model.

Workshop on Graph Representation Learning
33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

- We demonstrate that the proposed exponential family graph embedding models generally outperform widely used baseline approaches in various learning tasks on graphs.

2 Preliminary Concepts

Random walk-based methods. For a graph $G = (\mathcal{V}, \mathcal{E})$, a sequence of nodes $\mathbf{w} = (w_1, ..., w_L) \in \mathcal{W}$ will be called a *walk* if $(w_{l-1}, w_l) \in \mathcal{E}$ for every $2 \leq l \leq L$ and $\mathcal{W}$ will represent the set of walks of length L. The *context* sequence of a node w_l, referred to as *center*, in the random walk $\mathbf{w} \in \mathcal{W}$ is defined as $\mathcal{N}_\gamma^{\mathbf{w}}(w_l) := (w_{l-\gamma}, \ldots, w_{l-1}, w_{l+1}, \ldots, w_{l+\gamma})$. Random walk-based node embedding methods [12, 4, 10] generate a set of node sequences by simulating various (e.g., uniform, biased) random walk strategies. Node representations are then learned by optimizing the relationship between each center and context node pair in the generated sequences under the *Skip-Gram* model. We typically learn two embedding vectors $\alpha[v]$ and $\beta[v]$ for each node $v \in \mathcal{V}$, where $\beta[v]$ corresponds to the vector if the node is interpreted as *center* and $\alpha[v]$ denotes the vector if it is considered as *context* (in the experiments, we will only consider $\alpha[v]$ to represent the embedding vector of v).

Exponential families. A class of probability distributions is called *exponential family*, if they can be expressed as $p(x) = h(x) \exp\left(\eta T(x) - A(\eta)\right)$, where h is the *base measure*, η are the *natural parameters*, T is the *sufficient statistic* and $A(\eta)$ is the *log-partition* function [1]. Different choices of base measure and sufficient statistics lead us to obtain different distributions. For instance, the base measure and sufficient statistic of *Poisson* distribution are $h(x) = 1/x!$ and $T(x) = x$. We use the natural parameter $\eta_{v,u}$ to design a set of network representation learning models [15]; it is defined as the product of context and center vectors, $\eta_{v,u} := f\left(\alpha[u]^\top \beta[v]\right)$, where f is called the *link function*.

3 Proposed Approach

We define the generic objective function to learn node embeddings in the following way:

$$\mathcal{L}(\alpha, \beta) := \arg\max_{\Omega} \sum_{\mathbf{w} \in \mathcal{W}} \sum_{1 \leq l \leq L} \sum_{v \in \mathcal{V}} \log p(y^l_{w_l,v}; \Omega), \tag{1}$$

where $y^l_{w_l,v}$ is the observed value indicating the relationship between center w_l and context node v. Instead of restricting ourselves to the *Sigmoid* or *Softmax* functions in order to model the conditional probability in the objective function of Eq. (1), we assume that each $y_{w_l,v}$ follows an exponential family distribution. That way, the objective to learn node embeddings $\Omega = (\alpha, \beta)$ can be rewritten as:

$$\arg\max_{\Omega} \sum_{\mathbf{w} \in \mathcal{W}} \sum_{1 \leq l \leq L} \sum_{v \in \mathcal{V}} \Bigg[\log h(x_{w_l,v}) + \eta_{w_l,v} T(x_{w_l,v}) - A(\eta_{w_l,v}) \Bigg]. \tag{2}$$

As we can observe, Eq. (2) generalizes *Skip-Gram*-based models to exponential family distributions. Therefore, the EFGE models have the additional flexibility to utilize a wide range of exponential distributions, allowing them to capture more complex types of node interactions.

We examine three particular instances of the EFGE model, in particular, we utilize the Bernoulli, Poisson, and Normal distributions leading to the corresponding EFGE-Bern, EFGE-Pois and EFGE-Norm models. For illustration, two dimensional embeddings of *Dolphins* network [8] are depicted in Fig. 1 in the Appendix. As we can observe, the proposed EFGE-Bern and EFGE-Pois models learn representations differentiating nodes with respect to their communities.

3.1 The EFGE-Bern Model

Our first model is the EFGE-Bern model, in which we assume that each $y_{w_l,v}$ follows a Bernoulli distribution, which is equal to 1 if node v appears in the context set of w_l in the walk $\mathbf{w} \in \mathcal{W}$. It can be written as $y_{w_l,v} = x^{l-\gamma}_{w_l,v} \vee \cdots \vee x^{l-1}_{w_l,v} \vee x^{l+1}_{w_l,v} \vee \cdots \vee x^{l+\gamma}_{w_l,v}$, where $x^{t+l}_{w_l,v}$ indicates the appearance of v in the context of w_l at the position t $(-\gamma \leq t \leq \gamma)$. We express the objective function of the EFGE-Bern model, $\mathcal{L}_B(\alpha, \beta)$, by dividing Eq. (2) into two parts with respect to the values of $x_{w_l,v}$:

$$\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\Bigg[\sum_{v\in\mathcal{N}_{\gamma}(w_l)}\log p(y_{w_l v})+\sum_{v\notin\mathcal{N}_{\gamma}(w_l)}\log p(y_{w_l v})\Bigg]=\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\Bigg[\sum_{\substack{-\gamma\leq j\leq\gamma\\ u:=w_j}}\log p(x^{l+j}_{w_l u})+\sum_{\substack{-\gamma\leq j\leq\gamma\\ u:\neq w_j}}\log p(x^{l+j}_{w_l u})\Bigg]$$

$$=\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\Bigg[\sum_{\substack{-\gamma\leq j\leq\gamma\\ u:=w_j}}\log\sigma(\eta_{w_l u})+\sum_{\substack{-\gamma\leq j\leq\gamma\\ u:\neq w_j}}\log\sigma(-\eta_{w_l u})\Bigg],$$

where $\sigma(\eta)$ is the sigmoid function defined as $1/(1-\exp(-\eta))$ and the identity map is chosen for the link function, so $\eta_{v,u}$ is the product of vectors $\alpha[v]$ and $\beta[u]$.

Relationship to negative sampling. In Lemma 1, we show that the log-likelihood $\mathcal{L}_B(\alpha,\beta)$ of the EFGE-BERN model in fact converges to the objective function of negative sampling given in Eq. (3).

Lemma 1. *For large values of k, the log-likelihood function $\mathcal{L}_B$ converges to*

$$\sum_{\boldsymbol{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{-\gamma\leq j\leq\gamma}\Big[\log p(x^{l+j}_{w_l,w_{l+j}})+\sum_{s=1}^{k}\mathbb{E}_{u\sim q^{-}}\big[\log p(x^{l+j}_{w_l,u})\big]\Big]. \tag{3}$$

Proof. Please see the appendix. □

3.2 The EFGE-POIS Model

Let $y_{w_l,v}$ be a value indicating the number of occurrences of node v in the context of w_l. We assume that $y_{w_l,v}$ follows a Poisson distribution, with the mean value $\tilde{\lambda}_{w_l,v}$ being the number of appearances of node v in the context $\mathcal{N}^{\mathbf{w}}_{\gamma}(w_l)$. Similar to the previous model, it can be expressed as $y_{w_l,v} = x^{l-\gamma}_{w_l,v} + \cdots + x^{l-1}_{w_l,v} + x^{l+1}_{w_l,v} + \cdots + x^{l+\gamma}_{w_l,v}$, where $x^{l+t}_{w_l,v} \sim Pois(\lambda_{w_l,v})$ for $-\gamma\leq t\leq\gamma$. That way, we obtain $\tilde{\lambda}_{w_l,v}=\sum_{j=-\gamma}^{\gamma}\lambda^{l+j}_{w_l,v}$, since the sum of independent Poisson random variables is also Poisson. By plugging the exponential form of the Poisson distribution into Eq. (1), and following a similar strategy as in the EFGE-BERN model, the equation can be split into two parts for the cases where $y_{w_l,v}>0$ and $y_{w_l,v}=0$. That way, the negative sampling strategy (given in Eq. (3)) can be adopted:

$$\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{\substack{-\gamma\leq j\leq\gamma\\ u:=w_j}}\Big[-\log(x_{w_l,u}!)+\eta_{w_l,u}x_{w_l,u}-\exp(\eta_{w_l,u})\Big]+\sum_{\substack{-\gamma\leq j\leq\gamma\\ u:\neq w_j}}\Big[-\exp(\eta_{w_l,u})\Big].$$

Relationship to overlapping community detection. It can be seen that the objective function of the widely used BIGCLAM overlapping community detection method by Yang and Leskovec [18], can be obtained by unifying the objectives of the EFGE-BERN and EFGE-POIS models.

Lemma 2. *Let $Z_{w_l,v}$ be independent random variables following Poisson distribution with natural parameter $\eta_{w_l,v}$ defined as $\log(\beta[w_l]\cdot\alpha[v])$. Then, if the model parameter $\pi_{w_l,v}$ is defined as $p(Z_{w_l,v}>0)$, the objective function of* EFGE-BERN *model becomes equal to*

$$\sum_{\boldsymbol{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\Bigg[\sum_{v\in\mathcal{N}_{\gamma}(w_l)}\log\Big(1-\exp\big(-\beta[w_l]\cdot\alpha[v]\big)\Big)-\sum_{v\notin\mathcal{N}_{\gamma}(w_l)}\beta[w_l]\cdot\alpha[v]\Bigg].$$

Proof. Please see the appendix. □

3.3 The EFGE-NORM Model

We consider each $y_{w_l,v}$ as an edge weight indicating the relationship between w_l and v. We assume that $x^{l+t}_{w_l,v}\sim\mathcal{N}(1,\sigma^2_{+})$ if $v\in\mathcal{N}_{\gamma}(w_l)$, and $x^{l+t}_{w_l,v}\sim\mathcal{N}(0,\sigma^2_{-})$ otherwise. Hence, we obtain that

$y_{w_l,v} \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma^2})$, where $\tilde{\mu}$ is the number of occurrences of v in the context if we follow a similar assumption $y_{w_l,v} = \sum_{j=-\gamma}^{\gamma} x_{w_l,v}^{l+j}$ as in the previous models. The base measure of the model is $\exp(-x_{w_l,u}^2/2\sigma^2)/\sqrt{2\pi\sigma}$ for known variance, and $\eta_{w_l,u}$ is defined as $\exp(-\alpha[u]^\top \beta[w_l])$:

$$\sum_{\mathbf{w}\in\mathcal{W}} \sum_{1\leq l\leq L} \sum_{\substack{-\gamma\leq j\leq\gamma \\ u:=w_j}} \left[\log h(x_{w_l u}) + x_{w_l u}\frac{\eta_{w_l u}}{\sigma^+} - \frac{\eta_{w_l u}^2}{2}\right] + \sum_{\substack{-\gamma\leq j\leq\gamma \\ u:\neq w_j}} \left[\log h(x_{w_l u}) + x_{w_l u}\frac{\eta_{w_l u}}{\sigma^-} - \frac{\eta_{w_l u}^2}{2}\right].$$

4 Experimental Evaluation

The performance of the proposed models is evaluated in node classification and link prediction tasks. The details of the experimental set-up, baseline methods and datasets are provided in the Appendix.

Node classification. Table 1 shows the classification performance on three different networks. The experiments show that the proposed EFGE-Pois and EFGE-Norm models perform quite well, outperform most baselines especially on limited training data. The percentage gain for Micro-F_1 score of our best model with respect to the highest baseline score varies from 0.61% up to 5.33% for *CiteSeer* and from 0.22% to 2.44% for *DBLP*. The highest gain of EFGE-Norm model against the best performing baseline is around 3.80% over *Cora*.

These results can qualitatively be explained by the fact that, the exponential family distribution models enable to effectively capture the number of occurrences of a node within the context of another one, while learning the embedding vectors. Of course, the structural properties of the network, such as the existence of community structure, might affect the performance as shown in the toy example of Fig. 1 in the Appendix. The existence of well defined communities at the *Dolphins* network, allows the EFGE-Pois model to learn more discriminative embeddings with respect to the underlying communities.

	4%	**6%**	**8%**	**10%**	**30%**	**50%**	**70%**	**90%**
DeepWalk	0.460	0.489	0.505	0.517	0.566	0.584	0.595	0.592
Node2Vec	0.491	0.517	0.530	0.541	0.585	0.597	0.601	0.599
LINE	0.387	0.423	0.451	0.466	0.532	0.551	0.560	0.564
HOPE	0.205	0.210	0.204	0.219	0.256	0.277	0.299	0.320
NetMF	0.496	0.526	0.540	0.552	0.590	0.603	0.604	0.608
EFGE-Bern	0.493	0.517	0.536	0.549	0.588	0.603	0.609	0.609
EFGE-Pois	0.514	0.537	0.551	**0.562**	0.595	**0.606**	0.611	0.613
EFGE-Norm	**0.525**	**0.542**	**0.553**	0.561	**0.596**	**0.606**	**0.612**	**0.616**

a: *CiteSeer*

	4%	**6%**	**8%**	**10%**	**30%**	**50%**	**70%**	**90%**
DeepWalk	0.689	0.715	0.732	0.747	0.802	0.819	0.826	0.833
Node2Vec	0.714	0.743	0.757	0.769	0.815	0.831	0.839	0.841
LINE	0.544	0.590	0.633	0.661	0.746	0.765	0.774	0.775
HOPE	0.302	0.299	0.302	0.302	0.301	0.302	0.303	0.302
NetMF	0.716	0.748	0.767	0.773	**0.821**	**0.834**	**0.841**	**0.844**
EFGE-Bern	0.720	0.743	0.759	0.767	0.808	0.823	0.834	0.838
EFGE-Pois	0.733	0.746	0.759	0.765	0.802	0.814	0.820	0.825
EFGE-Norm	**0.743**	**0.760**	**0.770**	**0.780**	0.810	0.824	0.827	0.839

b: *Cora*

	4%	**6%**	**8%**	**10%**	**30%**	**50%**	**70%**	**90%**
DeepWalk	0.585	0.600	0.608	0.613	0.626	0.628	0.628	0.633
Node2Vec	0.600	0.611	0.619	0.622	0.636	0.638	0.639	0.639
LINE	0.580	0.590	0.597	0.603	0.618	0.621	0.623	0.623
HOPE	0.378	0.379	0.379	0.379	0.379	0.379	0.378	0.380
NetMF	0.589	0.596	0.601	0.605	0.617	0.620	0.623	0.623
EFGE-Bern	0.598	0.610	0.617	0.622	0.634	0.638	0.638	0.638
EFGE-Pois	0.605	0.614	0.620	0.624	0.635	0.637	0.636	0.638
EFGE-Norm	**0.614**	**0.622**	**0.624**	**0.628**	**0.637**	**0.640**	**0.642**	**0.641**

c: *DBLP*

Table 1: Micro-F_1 scores for node classification.

Link prediction. Table 2 shows the area under curve (AUC) scores for the link prediction task. Since the networks used in the node classification experiments consist of disconnected components, we perform the link prediction over the largest connected component. As it can be seen, the EFGE-Norm model is performing quite well on almost all different types of networks. Although Node2Vec is quite effective having the same performance in two datasets, EFGE-Norm performs quite better in the remaining networks, with gain ranging from 0.04% up to 18.29%.

	DeepWalk	*Node2Vec*	*LINE*	*HOPE*	*NetMF*	*EFGE-Bern*	*EFGE-Pois*	*EFGE-Norm*
Citeseer	0.770	0.780	0.717	0.744	0.742	0.815	**0.834**	0.828
Cora	0.739	0.757	0.686	0.712	0.755	0.769	0.797	**0.807**
DBLP	0.919	0.954	0.933	0.873	0.930	0.950	0.950	**0.955**
AstroPh	0.911	0.969	0.971	0.931	0.897	0.963	0.922	**0.973**
HepTh	0.843	0.896	0.854	0.836	0.882	**0.898**	0.885	0.896
Facebook	0.980	**0.992**	0.986	0.975	0.987	0.991	0.991	**0.992**
GrQc	0.921	**0.940**	0.909	0.902	0.928	0.938	0.937	**0.940**

Table 2: AUC scores for link prediction.

5 Conclusions

We introduced the EFGE models, proposing three instances (EFGE-Bern, EFGE-Pois, and EFGE-Norm) that generalize random walk approaches to exponential family. The benefit of these models stems from the fact that they allow to utilize conditional distributions over center-context node pairs, going beyond simple co-occurrence relationships. The experimental results have demonstrated that the proposed models are able to outperform widely used baseline methods.

References

[1] Erling Andersen. Sufficiency and exponential families for discrete sample spaces. *Journal of the American Statistical Association.*, 65(331):1248–1255, 1970.

[2] Leon Bottou. Stochastic gradient learning in neural networks. In *Proceedings of Neuro-Nimes. EC2*, 1991.

[3] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. HARP: Hierarchical representation learning for networks. In *AAAI*, 2018.

[4] Aditya Grover and Jure Leskovec. Node2Vec: Scalable feature learning for networks. In *KDD*, pages 855–864, 2016.

[5] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017.

[6] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.

[7] Jure Leskovec and Julian J. Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 539–547. Curran Associates, Inc., 2012.

[8] David Lusseau, Karsten Schneider, Oliver J. Boisseau, Patti Haase, Elisabeth Slooten, and Steve M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, Sep 2003.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[10] Duong Nguyen and Fragkiskos D. Malliaros. BiasedWalk: Biased sampling for representation learning on graphs. In *Big Data*, pages 4045–4053, 2018.

[11] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *KDD*, pages 1105–1114, 2016.

[12] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. In *KDD*, pages 701–710, 2014.

[13] Bryan Perozzi, Vivek Kulkarni, Haochen Chen, and Steven Skiena. Don't walk, skip!: Online learning of multi-scale network embeddings. In *ASONAM*, pages 258–265, 2017.

[14] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and Node2Vec. In *WSDM*, pages 459–467, 2018.

[15] Maja Rudolph, Francisco Ruiz, Stephan Mandt, and David Blei. Exponential family embeddings. In *NIPS*, pages 478–486. Curran Associates, Inc., 2016.

[16] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 2008.

[17] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[18] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *WSDM*, pages 587–596, 2013.

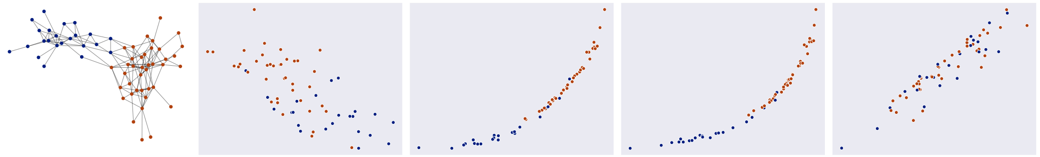
A Appendix

A.1 Source Code

The C+ implementation of EFGE models can be reached at: `https://github.com/abdcelikkanat/EFGE`

A.2 Visualization

For the illustration purposes of the proposed models, Fig. 1 depicts two dimensional embedding vectors of the *Dolphins* network composed by two communities.



(a) Network (b) DEEPWALK (c) EFGE-BERN (d) EFGE-POIS (e) EFGE-NORM

Figure 1: The *Dolphins* network composed by 2 communities and learned embeddings for $d = 2$.

A.3 Proofs of Lemmas

In this section, the proof of lemmas are given.

Lemma 1. *For large values of k. the log-likelihood function $\mathcal{L}_B$ converges to*

$$\sum_{w\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{-\gamma\leq j\leq\gamma}\Big[\log p(x^{l+j}_{w_l,w_{l+j}}) + \sum_{s=1}^{k}\mathbb{E}_{u\sim q^-}\big[\log p(x^{l+j}_{w_l,u})\big]\Big].$$

Proof. Let $q^-(\cdot|w_l)$ be the true conditional distribution of a random walk method for generating *non-context* nodes defined over $\mathcal{V}$. Then, it can be written that

$$\begin{aligned}
&\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{-\gamma\leq j\leq\gamma}\log p(x^{l+j}_{w_l,v_{l+j}}) + \sum_{s=1}^{k}\mathbb{E}_{u\sim q^-}\big[\log p(x^{l+j}_{w_l,u})\big]\\
\approx&\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{-\gamma\leq j\leq\gamma}\log p(x^{l+j}_{w_l,v_{l+j}}) + k\frac{1}{k}\sum_{\substack{s=1\\u_s\sim q^-}}^{k}\log p(x^{l+j}_{w_l,u_s})\\
=&\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{-\gamma\leq j\leq\gamma}\log p(x^{l+j}_{w_l,v_{l+j}}) + \sum_{\substack{s=1\\u_s\sim q^-}}^{k}\log p(x^{l+j}_{w_l,u_s})\\
\approx&\sum_{\mathbf{w}\in\mathcal{W}}\sum_{1\leq l\leq L}\sum_{\substack{-\gamma\leq j\leq\gamma\\u:=v_{l+j}}}\log p(x^{l+j}_{w_l,u}) + \sum_{\substack{-\gamma\leq j\leq\gamma\\u:\neq w_{l+j}}}\log p(x^{l+j}_{w_l,u})\\
=&\mathcal{L}_B(\alpha,\beta),
\end{aligned}$$

where the second line follows from the law of large numbers for the sample size of k and k is selected as $|\mathcal{V}| - 1$ in the fourth line. □

Lemma 2. *Let $Z_{w_l,v}$ be independent random variables following Poisson distribution with natural parameter $\eta_{w_l,v}$ defined as* $\log(\beta[w_l]\cdot\alpha[v])$. *Then, the objective function of* EFGE-BERN *model becomes equal to*

$$\sum_{w\in\mathcal{W}}\sum_{1\leq l\leq L}\Bigg[\sum_{v\in\mathcal{N}_\gamma(w_l)}\log\Big(1-\exp\big(-\beta[w_l]\cdot\alpha[v]\big)\Big) - \sum_{v\notin\mathcal{N}_\gamma(w_l)}\beta[w_l]\cdot\alpha[v]\Bigg],$$

if the model parameter $\pi_{w_l,v}$ is defined by $p(Z_{w_l,v} > 0)$.

Proof. Let $y_{w_l,v}$ follow a Bernoulli distribution with parameter $\pi_{w_l,v}$ and it is equal to 1 if $v \in \mathcal{N}_\gamma(w_l)$, and 0 otherwise. Then, the objective function $\mathcal{L}_B(\alpha, \beta)$ can be divided into parts as follows:

$$
\begin{aligned}
\mathcal{L}_{\mathcal{B}} &= \sum_{\mathbf{w}\in\mathcal{W}} \sum_{1\le l\le L} \left[\sum_{v\in\mathcal{N}_\gamma(w_l)} \log p(y_{w_l,vj}) + \sum_{v\notin\mathcal{N}_\gamma(w_l)} \log p(y_{w_l,v}) \right] \\
&= \sum_{\mathbf{w}\in\mathcal{W}} \sum_{1\le l\le L} \left[\sum_{v\in\mathcal{N}_\gamma(w_l)} \log\big(1 - p(z_{w_l,v} = 0)\big) + \sum_{v\notin\mathcal{N}_\gamma(v_i)} \log p(z_{w_l,v} = 0) \right] \\
&= \sum_{\mathbf{w}\in\mathcal{W}} \sum_{1\le l\le L} \left[\sum_{v\in\mathcal{N}_\gamma(w_l)} \log\Big(1 - \exp\big(-\exp(\eta_{w_l,v})\big)\Big) + \sum_{v\notin\mathcal{N}_\gamma(w_l)} \exp(-\eta_{w_l,v}) \right] \\
&= \sum_{\mathbf{w}\in\mathcal{W}} \sum_{1\le l\le L} \left[\sum_{v\in\mathcal{N}_\gamma(w_l)} \log\Big(1 - \exp\big(-\beta[w_l]\cdot\alpha[v]\big)\Big) - \sum_{v\notin\mathcal{N}_\gamma(w_l)} \beta[w_l]\cdot\alpha[v] \right].
\end{aligned}
$$

□

A.4 Description of Datasets

Here we provide a detailed description of networks that we have used in our study and (Table 3 shows various statistics of the networks):

- *CiteSeer* [3] is a citation network obtained from the *CiteSeer* library, in which each node corresponds to a paper and the edges indicate reference relationships among papers. The labels represent the subjects of the paper.
- *Cora* [16] is another citation network constructed from the publications in the machine learning area; the documents are classified into seven categories.
- *DBLP* [13] is a co-authorship graph, where an edge exists between nodes if two authors have co-authored at least one paper. The labels represent the research areas.
- *AstroPh* [6] is another collaboration network built from the papers submitted to the *ArXiv* repository for the Astro Physics subject area, from January 1993 to April 2003.
- *HepTh* [6] network is constructed in a similar way from the papers submitted to *ArXiv* for the *High Energy Physics - Theory* category.
- *GrQc* [6] is our last collaboration network which has been constructed from the e-prints submitted to the category of *General Relativity and Quantum Cosmology*.
- *Facebook* [7] is a social network extracted from a survey conducted via a *Facebook* application.

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{K}\rvert$	$\lvert\mathcal{C}\rvert$	**Avg. Degree**	**Density**
CiteSeer	3,312	4,660	6	438	2.814	0.0009
Cora	2,708	5,278	7	78	3.898	0.0014
DBLP	27,199	66,832	4	2,115	4.914	0.0002
AstroPh	17,903	19,7031	-	1	22.010	0.0012
HepTh	8,638	24,827	-	1	5.7483	0.0007
Facebook	4,039	88,234	-	1	43.6910	0.0108
GrQc	4,158	13,428	-	1	6.4589	0.0016

Table 3: Statistics of networks used in the experiments. $|\mathcal{V}|$: number of nodes, $|\mathcal{E}|$: number of edges, $|\mathcal{K}|$: number of labels and $|\mathcal{C}|$: number of connected components.

A.5 Baseline Methods

We evaluate the three proposed EFGE models against five state-of-the-art NRL techniques:

- DEEPWALK [12] generates a set of node sequences by choosing a node uniformly at random from the neighbours of the node it currently resides.
- NODE2VEC [4] relies on a biased random walk strategy, introducing two additional parameters which are used to determine the behaviour of the random walk in visiting nodes close to the one currently residing at. We simply set these parameters to 1.0.
- LINE [17] learns embeddings that are based on first-order and second-order proximity (each one of length $d/2$).
- HOPE [11] is a matrix factorization method which aims at extracting feature vectors by preserving higher order patterns of the network (in our experiments, we have used the *Katz* index).
- NETMF [14] aims at factorizing the matrix approximated by the pointwise mutual information of center and context pairs.

A.6 Experimental Setup

For node classification, we aim at predicting the correct labels of nodes having access to a limited number of training labels (i.e., nodes with known label). In our experiments, we split the learned embedding vectors into varying sizes of training and test sets, from 4% up to 90% in order to better evaluate the models. We perform our experiments applying an one-vs-rest logistic regression classifier with L_2 regularization.

In the link prediction task, the goal is to predict the missing edges or to estimate possible future connections between nodes. For this experiment, we randomly remove half of the edges of a given network, keeping the residual network connected. Then, we learn node representations using the residual network. The removed edges as well as a randomly chosen set of same number node pairs form the testing set. For the training set, we sample the same number of non-existing edges following the same strategy to have negative samples, and the edges in the residual network are used as positive instances. Since we learn embedding vectors for the nodes of the graph, we use the extracted node representation to build edge feature vectors using *Hadamard* product operator. In all experiments, we have used the logistic regression classifier with L_2 regularization over the networks listed in Table 3.

In our experiments, we have chosen walk length $L = 10$, number of walks $N = 80$ and window size $\gamma = 10$ for all models and the variants of EFGE model are fed with the same node sequences produced by NODE2VEC. The size of embedding vectors are chosen as 128 for all methods.

A.7 Optimization

For the optimization step of our models, we adopt *Stochastic Gradient Descent* (SGD) [2] to learn latent representations $\Omega = (\alpha, \beta)$. Except for the large networks, we start the initial learning rate from 0.025, and then it is linearly decreased with respect to the number of nodes which have been processed so far. We set the minimum step size to 0.0001 and we do not allow it to fall below this value. Since it is computationally very expensive to compute gradients for each node pairs, we take advantage of the fact that we have formulated the objective function of each model in a such way that it could be divided into two parts according to the values of x_{ij}'s; thus, we adopt the negative sampling strategy with the sampling size $k = 5$ in all the experiments. We generate negative samples from the whole vertex set with respect to the number of occurrences of nodes in the generated walks raised to the power of 0.75, similar to Ref. [9].

A.8 Parameter Sensitivity

We have performed further sensitivity analysis experiments, to better understand the impact of the various parameters in the performance of the proposed models.

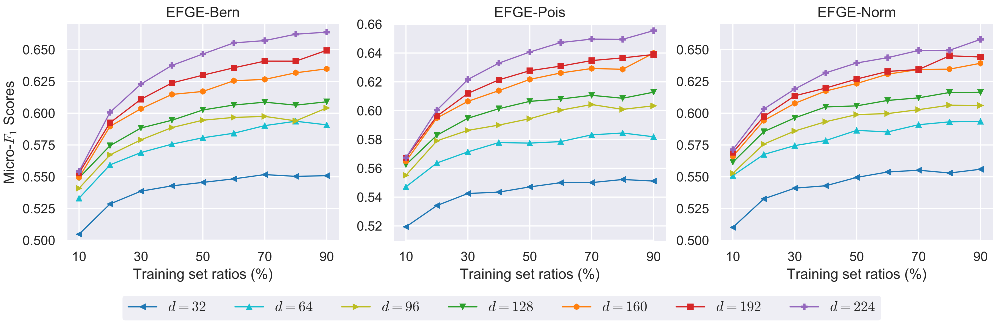


Figure 2: Influence of dimension size over *CiteSeer* network.

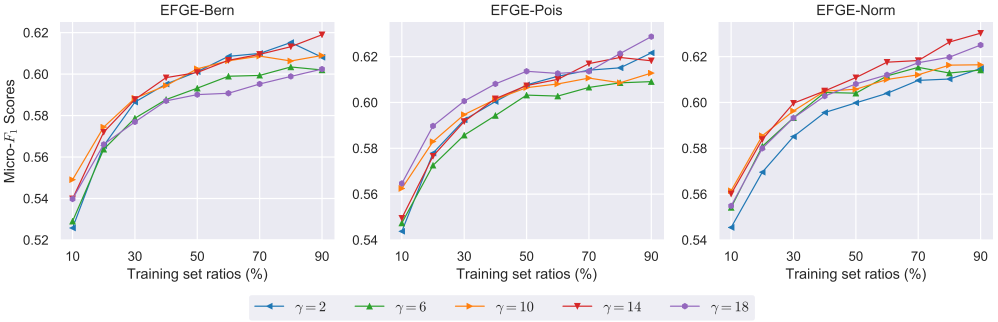


Figure 3: Influence of window size γ for the *CiteSeer* network.

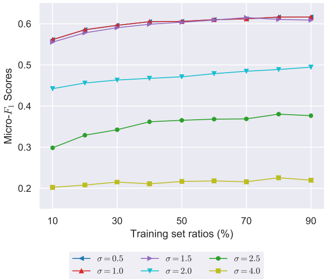


Figure 4: Influence of standard deviation for the EFGE-NORM model.

Influence of dimension and window sizes. We examine the effect of embedding dimension d and the effect of the window size γ used to sample context nodes on the different models over different training set ratios. The results are depicted in Figures 2 and 3.

The effect of standard deviation at the EFGE-NORM model. The EFGE-NORM model has an extra parameter σ which can influence the performance of the method. To examine the impact of σ, we have chosen six different values, performing experiments over *CiteSeer* network. Figure 4 depicts how the Micro-F_1 scores change for various training set ratios. The results clearly indicate that the model performs well for small values of σ — with the best results obtained for $\sigma = 1$. For this reason, we have set this value for all the experiments conducted in the node classification and link prediction tasks.