IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 2, 2022, accepted March 18, 2022, date of publication April 4, 2022, date of current version April 12, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164670

Toward an Adaptive Skip-Gram Model for Network Representation Learning

I-CHUNG HSIEH AND CHENG-TE LI, (Member, IEEE)
Institute of Data Science, National Cheng Kung University, Tainan 70101, Taiwan

Corresponding author: Cheng-Te Li (reliefli@gmail.com)

This work was supported by the Ministry of Science and Technology (MOST) of Taiwan under Grant 110-2221-E-006-136-MY3, Grant 110-2221-E-006-001, and Grant 110-2634-F-002-051.

ABSTRACT The random walk process on network data is a widely-used approach for network representation learning. However, we argue that the sampling of node sequences and the subsampling for the Skip-gram's contexts have two drawbacks. One is less possible to precisely find the most correlated context nodes for every central node with only uniform graph search. The other is not easily controlled due to the expensive cost of hyperparameter tuning. Such two drawbacks lead to higher training cost and lower accuracy due to abundant and irrelevant samples. To solve these problems, we compute the adaptive probability of random walk based on Personalized PageRank (PPR), and propose an *Adaptive SKip-gram* (ASK) model without using complicated sampling process and negative sampling. We utilize k-most important neighbors for positive samples selection, and attach their corresponding PPR probability into the objective function. Based on benchmark datasets with three citation networks and three social networks, we demonstrate the improvement of our ASK model for network representation learning in tasks of link prediction, node classification, and embedding visualization. The results achieve more effective performance and efficient learning time.

INDEX TERMS Network embedding, random walk, personalized pagerank, context co-occurrence, representation learning, node classification, link prediction, visualization.

I. INTRODUCTION

Network data is getting much attention due to modern issues like social media analytics, disease infection, and knowledge database. *Graph representation learning* (GRL) [1] is an essential task to distill latent features from network data. While a network consists of a collection of links between nodes in a non-Euclidean space, the common purpose of GRL is to convert the highly complex network structure to a low-dimensional and explicit vector for each node, which is termed *node embedding*. Eventually, the embedding vectors can be used for downstream network analysis tasks, such as link prediction [2], node classification [3], and community detection [4]. Furthermore, network embedding can offer more information to solve real-world problems. In the recommender systems with user attributes and their interactions with items, to learn essential features, GNN-SoR [5] generates the embeddings based on social influence and user preference, and leverages them with items' content for producing recommendation outcomes based on matrix factorization. Besides, SIoT-SR [6] constructs the recommender system in the context of Internet-of-Things. By collecting and learning from multiple feedback of different items, and SIoT-SR can generate embeddings of users and items for effective recommendation. A generative network embedding model Graph-GAN [7] can be utilized for rumor event detection. Graph-GAN models posts' content via network embedding, and adopts adversarial learning to improve the utility of the derived embeddings.

To represent nodes in the context of network structure, the typical approaches can be divided into matrix-based, edge-based, and random walk-based. While the matrix-based approach like network matrix factorization (NetMF) [8], is costly in terms of computational efficiency. The edge-based model like the proximity preservation method LINE [9] is shallow resulting in less efficacy. The random walk-based methods, such as DeepWalk [10], extract correlated neighbors of the target by random walk, and utilize neural network

The associate editor coordinating the review of this manuscript and approving it for publication was Le Hoang Son.

learning to generate node embeddings. However, the random walk-based methods require sampling abundant paths to approximate the degree of correlation among nodes, and would be influenced by several hand-crafted hyperparameters [10]–[12]. Besides, the hyperparameters optimization is an expensive task corresponding to the model quality, and a larger number of hyperparameters make the model configuration space more complex [13], [14]. That is, we argue that a random walk-based sampling-based method is easily influenced by random noise and has high cost of hyperparameter tuning, such as the number of walks per node, walk length, and context size. These factors lead to the requirement of more learning samples, tedious hyperparameter tuning, and most importantly, the selection of irrelevant context nodes for every central node. In this work, we aim to revisit the Skip-gram model with the concept of the random walk, and show a simplified implementation based on PPR can effectively replace the original random walk with performance improvement on downstream tasks.

It is because that the truncated context with the fixed length is not capable to depict the topological correlation (e.g., proximity) between the central node and each context node. Besides, the sampling frequency distribution of nodes occurring in the target's context would be less precise as the number of the samples is not enough. If we try to sample more samples to improve accuracy, we have to create the additional cost during training model. Therefore, we need a more precise mechanism to select representative neighbors for every node. This would be achieved by the estimation of adaptive probability in the random walk process, along with some incorporation into the Skip-gram model.

To deal with the aforementioned issues, we leverage Personalized PageRank [15] (PPR) that represents the convergent probability from a root (central node) to any other nodes along a randomly sampled path. PPR can be considered as a re-arrangement of the random walk without given any sampling steps [16]. Therefore, we can consider such a probability as the degree of correlation between two nodes as well as the exact node frequency in sampled node sequences. To be specific, by combining the PPR probability and the random walk process, we can derive the adaptive random walk probability indicating the structural correlation between two nodes so that we can accordingly select the most significant context nodes for every central node. Eventually, by incorporating PPR into Skip-gram model, we develop the *Adaptive SKip-gram* (ASK) model.

We summarize the contributions of this work as follows.

- First, we simplify the complex random walk process by the probability of personalized PageRank. The hyperparameters of the original random walk in Skip-gram model can be combined as one.
- Second, technically, we improve the Skip-gram model via the estimated probability by proposed Adaptive SKip-gram model, which emphasizes and exploits the correlation between nodes. Our model would precisely learn the correlation, and does not require the negative sampling that could lead to misleading embeddings and increase computational cost.
- Third, the experiments conducted on three citation networks and three social networks in GRL tasks exhibit the improvement of our Adaptive SKip-gram (ASK) model in link prediction, node classification, and embedding visualization. We also suggest an approximated version of the Adaptive Skip-gram model that can be used to achieve efficient but similar performance in the limited environment.

II. RELATED WORK

In this section, we discuss the existing random walk-based method for the graph representation learning. First, DeepWalk [10] adopts the random walk mechanism and the Skip-gram model to efficiently learn node embeddings. The main idea comes from the language model in word2vec [17]. Based on the random surfer that walks through highly correlated local neighbors surrounded by each target node, and Skip-gram model is able to truncate a context with inter-correlated words and updates node embeddings. node2vec [18] presents a biased random walk controlled by the hyperparameters of depth-first and breadth-first search. GENE [19] considers the group labels from the random walk's neighbors to preserve more information in node embeddings. DDRW [20] jointly optimizes the classification objective and the objective of random-walk-based embedding entities for better node classification. Walklets [21] discusses multi-scale meanings in the real world graph and proposes subsampling process to skip the random walk path for extracting the embedding for specific scales. Besides, Struct2Vec [22] constructs the multilayers graph for different hierarchical levels and follows node2vec to learn the representation for each layer. DRRW [23] analyzes the convergence of random path and proposes an exploration score to guide the path toward less-visited nodes for better distribution learning.

Extended studies further aim at learning node embeddings in attributed networks, in which ANRL [24], RWR-GAE [25], and ARWR-GE [26] are random walk-based approaches that also incorporate the Skip-gram model as a component for the graph structure preservation. On the other hand, some methods such as DANE [27], GraphRNA [28], and wGCN [29] utilize the random walk to extract the graph structure and help the representation learning via random path and co-currency. In short, one research direction of GRL is incorporating the Skip-gram model with random walk, which is widely validated as being useful.

While the random walk mechanism takes high sampling cost and has imprecise estimation of node's context, PPR can precisely depict graph diffusion without any specific sampling process. Some graph applications like [30] employ graph neural network with PPR to improve information propagation for node classification. Lasagne [31] utilizes PPR to find important neighbors in the large-scale community, and C_PPR [32] is designed for community detection by

PPR measurement of node proximity globally. However, few studies properly apply PPR to the Skip-gram model.

III. METHODOLOGY

A. PERSONALIZED PAGERANK

Given a network $\boldsymbol{G} = (V, \boldsymbol{A})$, where V is the node set with n nodes ($|V| = n$), $\boldsymbol{A}$ is the adjacency matrix. A personalized PageRank [15] (PPR) value can be seen as the probability from a certain root r to another node v via a random walk-like process. The probability updating equation of personalized PageRank is given by

$$\boldsymbol{\pi}_r^{(n)} = (1-\alpha)\boldsymbol{H}\boldsymbol{\pi}_r^{(n-1)} + \alpha \boldsymbol{e}_r, \tag{1}$$

where $\boldsymbol{\pi}_r^{(n)}$ is the probability vector from the root r to each node at n-th step, $\boldsymbol{H} = \boldsymbol{D}^{-1}\boldsymbol{A}$ is the normalized adjacency matrix based on $\boldsymbol{A}$ and the degree matrix $\boldsymbol{D}$.

In addition, $\alpha \in [0, 1]$ is the restart probability, and $\boldsymbol{e}_r$ is the one-hot encoding vector for the root. After some reformulation, the PPR matrix $\boldsymbol{\Pi}$ can be described as

$$\boldsymbol{\Pi} = \alpha(I - (1-\alpha)H)^{-1}, \tag{2}$$

where $\boldsymbol{\Pi}_{ij}$ means the probability of going to the node j from the root i. Note that we will use ''root,'' ''central node'' and ''target'' interchangeably throughout this work.

B. ADAPTIVE SKIP-GRAM MODEL

Typical network representation learning methods with the Skip-gram model and random walk, such as node2vec [18] and DeepWalk [10], have three common phases. It contains sampling node sequences by random walk, generating contexts, and the Skip-gram model. The second is composing contexts of every node by setting central nodes and neighboring nodes from left to right in the derived node sequences. The third is applying the Skip-gram model. We argue such a process cannot precisely extract significant contexts for each node. It is because the random walk is not personally performed to generate the contexts for a central node. That said, the contexts, sampled via random walk, may be correlated with the central node. To be more specific, for nodes with high proximity scores to each other in a densely-connected community, they may not be each other's context. Repeated independent sampling via random walk from any nodes lead to such kind of outcome.

We aim at exploiting the probability values derived from personalized PageRank to generate the contexts of every node. Since PPR values reflect the proximity degree from a root node to any other nodes in the network, we propose to leverage PPR for generating more representative contexts so that the Skip-gram model can be constructed to produce better node embeddings. We will generate representative contexts by selecting top-k neighbors that possess the highest proximity values to the root/central node. In addition, we also want to simplify the process by allowing only one hyperparameter, rather than three typical hyperparameters, including context size, number of walks, and length of walk. The context size (i.e., number of contexts) can be regarded as the demand of the number of contexts to explain the central node. It should be proportional to the density and size of the central node's neighborhood. Hence, we make the parameter k play a role representing the maximum needed context size for learning a central node's embedding.

To estimate k, we need to figure out the occurrence frequency of every node in all random walk generated sequences. Because the derivation of PPR is according to the iteration of the node transition probability Eq. (1), the results for n iterations can represent the probabilities of the n-th node that we would sample from the given root. Thus, the PPR values derived from the case where n achieves the infinite, and PPR can also be considered as the probability of sampling a node of any generated infinite-length sequence from the root. The summation of the scaled probability from all nodes to any node j can be simply regarded as the node frequency in all sequences, given by

$$\boldsymbol{f}_j = \sum_{i=1}^{n} (\boldsymbol{\Pi}_{ij})/n. \tag{3}$$

Given the average context size a_e as a hyperparameter used to obtain k, the total number of contexts for all nodes would be $a_e \times n$. Then the expected context size for each node can be derived as a vector:

$$a_e \times n \times \boldsymbol{f}_j = a_e \sum_{i=1}^{n} (\boldsymbol{\Pi}_{ij}). \tag{4}$$

We choose k to be the maximum expected context size for each node, given by

$$max_j(a_e \sum_{i=1}^{n} \boldsymbol{\Pi}_{ij}). \tag{5}$$

The next step is to attach the subsampling mechanism into the derivation of k. The subsampling in the original Skip-gram model utilizes the discarding probability [11]

$$1 - (t_0(\boldsymbol{f}_w)^{-1} + \sqrt{t_0}(\boldsymbol{f}_w)^{-0.5}), \tag{6}$$

where t_0 is a chosen threshold (typically 10^{-5}), and $\boldsymbol{f}_w$ is frequency vector of each word in all sentences.

Based on the node frequency in Eq. 3, the subsampling probability would be

$$\boldsymbol{p}_{sub} = t_0(\boldsymbol{f}_j)^{-1} + \sqrt{t_0}(\boldsymbol{f}_j)^{-0.5}, \tag{7}$$

which smooths the sampling probability of highly-frequent nodes. As a result, the maximum expected context size with subsampling is given by

$$k = max_j(a_e \boldsymbol{f}_j \odot \boldsymbol{p}_{sub}), \tag{8}$$

where $\odot$ is Hadamard product. Such selection of k-most significant context nodes, along with PPR, simplifies the context generation and its hyperparameters.

We incorporate the Skip-gram model with the derived expected context size:

$$a_e \boldsymbol{f}_c \odot \boldsymbol{p}_{sub}. \tag{9}$$

Consider the target node t and its k-most significant context nodes, we reconstruct the Skip-gram model to model the importance of each of its neighbors through PPR. The objective function is given by:

$$\sum_{c \in context(t)} log(\sigma(\boldsymbol{v}_t^T \boldsymbol{v}_c)), \tag{10}$$

for a pair of target t and its context node set $context(t)$, where $\boldsymbol{v}_t$ is the embedding for node t, and σ is the logit function. We replace original context nodes with nodes possessing k highest values in the subsampling PPR value matrix, given by

$$\{\boldsymbol{\Pi}_{sub}\}_{t*} = \{a_e \boldsymbol{Diag}(\boldsymbol{f}_c \odot \boldsymbol{p}_{sub})\boldsymbol{\Pi}\}_{t*}, \tag{11}$$

where $\boldsymbol{Diag}(\boldsymbol{v})$ is a diagonal matrix with diagonal entries equal to a vector $\boldsymbol{v}$. In other words, the values in PPR matrix is used in the objective function to point out which are significant neighbors. In short, our model is learned by k context nodes of each central node. The proposed PPR-enhanced objective not only emphasizes the importance of context nodes without additional cost, but alleviates the problem of choosing irrelevant neighbors as contexts. Hence, less correlated nodes in terms of proximity could be pushed away from one another in the learned embedding space. To some extent, such an effect is originally generated through *negative sampling*, and as a by-product in our model. Therefore, we choose not to perform negative sampling in our model.

C. AN APPROXIMATED APPROACH FOR PPR

Since the derivation of PPR matrix requires $O(n^3)$ time complexity, our Adaptive SKip-gram model may be less efficient when the network is large scale. Hence, we aim to provide an efficient alternative for the estimation of PPR matrix. Consider the inverse part of PPR matrix:

$$(I - (1-\alpha)\boldsymbol{H})^{-1} = \boldsymbol{P}^{-1}. \tag{12}$$

The normalized matrix with bounded row sum:

$$\sum_j (1-\alpha)\boldsymbol{H}_{ij} < 1, \tag{13}$$

satisfies $||(1-\alpha)\boldsymbol{H}|| < 1$. Therefore, $\boldsymbol{P}$ can be approximated by the convergent sum of Neumann series:

$$\lim_{m \to \infty} \sum_i^m ((1-\alpha)\boldsymbol{H})^i. \tag{14}$$

Given a small m, the complexity of the approximated PPR matrix would be decreased a lot due to the sparsity of $\boldsymbol{H}$. Besides, $((1-\alpha)\boldsymbol{H})^i$ can be regarded as the i-order proximity. Therefore, the approximated PPR matrix with a small m is capable to cover most of information for modeling.

IV. EXPERIMENTS

A. EXPERIMENT SETTINGS

We conduct experiments to evaluate the effectiveness of our Adaptive SKip-gram model for network representation learning. Three citation networks, including Cora, Citeseer, and Pubmed,[1] are employed. These three citation networks contain the relationships of paper citations as edges, and they are benchmark ones that are widely utilized to evaluate the quality of network embedding models [24]–[27], [29]. In addition, three social networks of Twitch users [33] from different countries with mutual follower-followee interactions as edges, including Twitch-EN, Twitch-RU, and Twitch-PT,[2] are also considered. The dataset sizes in (#nodes, #edges, #density) are (2708, 5278, 0.0014), (3312, 4460, 0.0008), (19717, 44327, 0.0002), (7126, 35324, 0.0014), (4385, 37304, 0.0039), (1912, 31299, 0.0171) for Cora, Citeseer, Pubmed, Twitch-EN, Twitch-RU, and Twitch-PT, respectively. We randomly choose 70%, 10%, and 20% edges as the training, validation, and testing sets. We select the best model according to the performance of the validation set. We also ensure the network is connected. The tasks include link prediction, node classification, and embedding visualization. These tasks follow the typical procedure of accessing the quality of node embeddings [8]–[10], [31]. We do perform the experiments on both node-level and path-level tasks. Node classification is the node-level task that examines whether network features can be encoded to distinguish nodes with labels from one another, while link prediction is the path-level task that exhibits whether the network structure can be reflected by the derived node embeddings. We adopt the commonly-used classification evaluation metrics. Specifically, the Area Under the Curve (AUC) score[3] is used for link prediction. Micro-F1 and Macro-F1 scores[4] are employed for node label classification. The higher score means better performance. Besides, embedding visualization can display how nodes with same labels are grouped together in the embedding space. We expect nodes with different labels are well separated from each other.

We compare the performance for the original SKip-gram model (SK) with biased random walk [18], the graph first- and second-order proximities preserving method, LINE [9], our Adaptive SKip-gram model (ASK), and PPR-Approximated Adaptive SKip-gram model (AASK(m)), where the order m of the Neumann series is given by three different sizes {5, 10, 20}.

The dimensionality of the node embedding vector is set 128 for all methods, and all models are trained by Adam optimizer with a learning rate $=$ 0.001. For the setting of SK, we set length window size $=$ 5, the number of repeating walks $=$ 1, and the walk length $=$ 80 for random walk process. The number of negative samples is 20 for Cora and Citeseer and 5 for Pubmed, these settings follow the tuned values obtained from the original word2vec work [11]. For the settings of ASK and AASK, we set the default expected average context size $a_e = 25$, and the restart probability of PPR is set

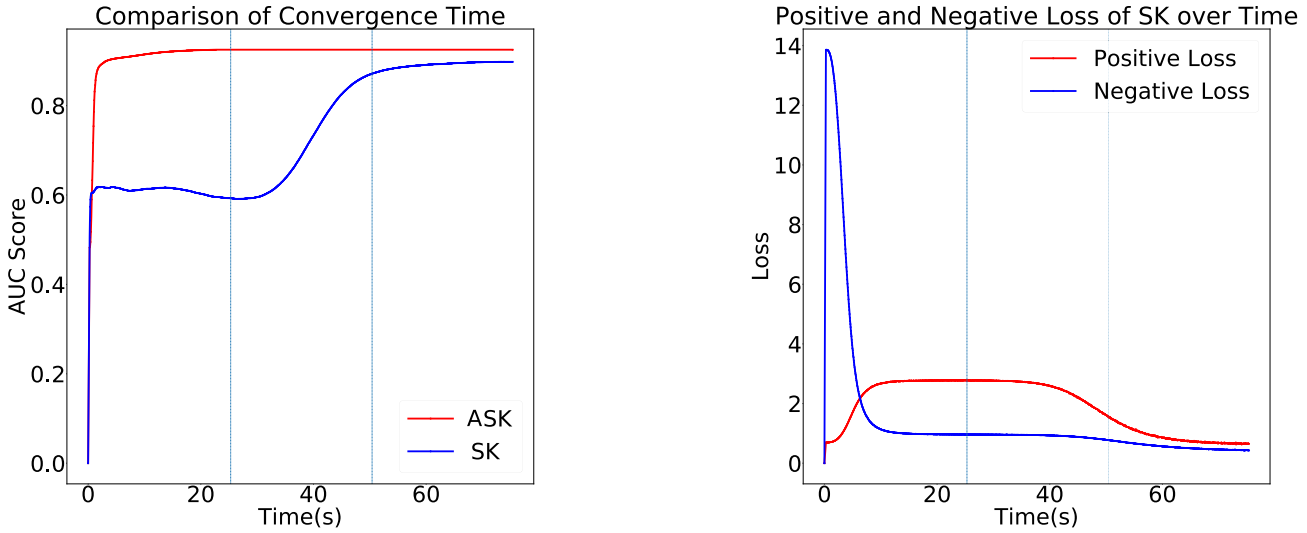
[1]Three citation datasets are available via https://linqs.soe.ucsc.edu/data
[2]Three social datasets are available via https://graphmining.ai/datasets/
[3]https://en.wikipedia.org/wiki/Receiver_operating_characteristic
[4]https://en.wikipedia.org/wiki/F-score

TABLE 1. AUC scores and time cost (seconds) for link prediction. (Cora, Citeseer and Pubmed).

	Cora	Citeseer	Pubmed
LINE	0.6505±0.0152 (85.00)	0.5465±0.0083 (85.00)	0.5382±0.0129 (**121.00**)
SK	0.8902±0.0093 (123.00)	0.9135±0.0080 (147.30)	0.9340±0.0030 (264.01)
ASK	**0.9262**±0.0053 (19.53)	**0.9387**±0.0065 (22.04)	**0.9399**±0.0023 (242.94)
AASK (5)	0.8965±0.0091 (**18.18**)	0.9015±0.0101 (21.09)	0.9276±0.0027 (162.36)
AASK (10)	0.9110±0.0073 (19.73)	0.9168±0.0053 (21.44)	0.9360±0.0015 (242.46)
AASK (20)	0.9196±0.0035 (22.83)	0.9310±0.0094 (22.70)	0.9395±0.0015 (433.79)

TABLE 2. AUC scores and time cost (seconds) for link prediction. (Twitch-EN, Twitch-RU, and Twitch-PT).

	Twitch-EN	Twitch-RU	Twitch-PT
LINE	0.8261± 0.0034 (102.00)	0.6983± 0.0118 (92.00)	0.7502± 0.0121 (84.00)
SK	0.6592±0.0057 (50.4474)	0.7177± 0.0045 (42.8637)	0.6646± 0.0050 (22.9353)
ASK	**0.8377**±0.0039 (**10.7264**)	**0.8673**± 0.0036 (**4.3091**)	**0.8086**± 0.0079 (**1.1066**)
AASK (5)	0.6184±0.0031 (12.9422)	0.6857±0.0121 (6.532)	0.62±0.0127 (1.8953)
AASK (10)	0.7094±0.0068 (26.6396)	0.7875±0.0067 (12.3042)	0.686±0.0105 (3.1824)
AASK (20)	0.7899±0.0047 (53.974)	0.8404±0.0032 (23.7233)	0.7605±0.0061 (5.6923)



(a) AUC scores vs. run time

(b) Loss of SK over time

FIGURE 1. Convergence Analysis of SK and ASK.

TABLE 3. #(Positive pair) and the detailed time cost for link prediction. (Cora, Citeseer and Pubmed).

	Cora	PT(s)	TT(s)	Citeseer	PT(s)	TT(s)	Pubmed	PT(s)	TT(s)
SK	3.6E+05	8.68	114.33	4.8E+05	8.88	138.39	7.0E+06	63.79	200.22
ASK	3.8E+05	1.32	18.21	3.4E+05	1.62	20.42	2.4E+06	114.52	128.42
AASK (5)	**3.5E+05**	**0.88**	**17.30**	**3.1E+05**	**0.91**	**20.18**	**2.1E+06**	**33.20**	129.15
AASK (10)	3.7E+05	1.91	17.82	3.2E+05	1.20	20.24	2.2E+06	114.97	**127.49**
AASK (20)	3.8E+05	4.64	18.18	3.3E+05	2.25	20.45	2.3E+06	305.21	128.58

as $\alpha = 0.05$ for Cora and Citeseer and 0.07 for Pubmed. After obtaining the node embeddings, we use Hadamard product to derive the embedding vectors of node pairs. Then, we utilize logistic regression as the classifier and the area under the ROC curve (i.e., AUC score) as the evaluation metric.

B. RESULTS

The results on link prediction are shown in Table 1, Table 2, Table 3 and Table 4 for six datasets (three citation networks and three social networks). Table 1 and Table 2 further exhibit both AUC scores and time cost in seconds. Table 3 and Table 4 present the number of training pairs without negative

TABLE 4. #(Positive pair) and the detailed time cost for link prediction. (Twitch-EN, Twitch-RU, and Twitch-PT).

	Twitch-EN	PT(s)	TT(s)	Twitch-RU	PT(s)	TT(s)	Twitch-PT	PT(s)	TT(s)
SK	9.4E+06	30.67	19.78	5.8E+06	29.35	13.51	1.9E+06	17.64	5.29
ASK	8.6E+05	**9.53**	**1.20**	5.5E+05	**3.46**	0.85	1.5E+05	**0.61**	**0.50**
AASK (5)	**8.1E+05**	11.65	1.29	**5.0E+05**	5.69	**0.84**	**1.4E+05**	1.39	0.51
AASK (10)	8.3E+05	25.41	1.23	5.2E+05	11.43	0.87	1.4E+05	2.68	**0.50**
AASK (20)	8.3E+05	52.78	1.19	5.4E+05	22.86	0.86	1.5E+05	5.20	**0.50**

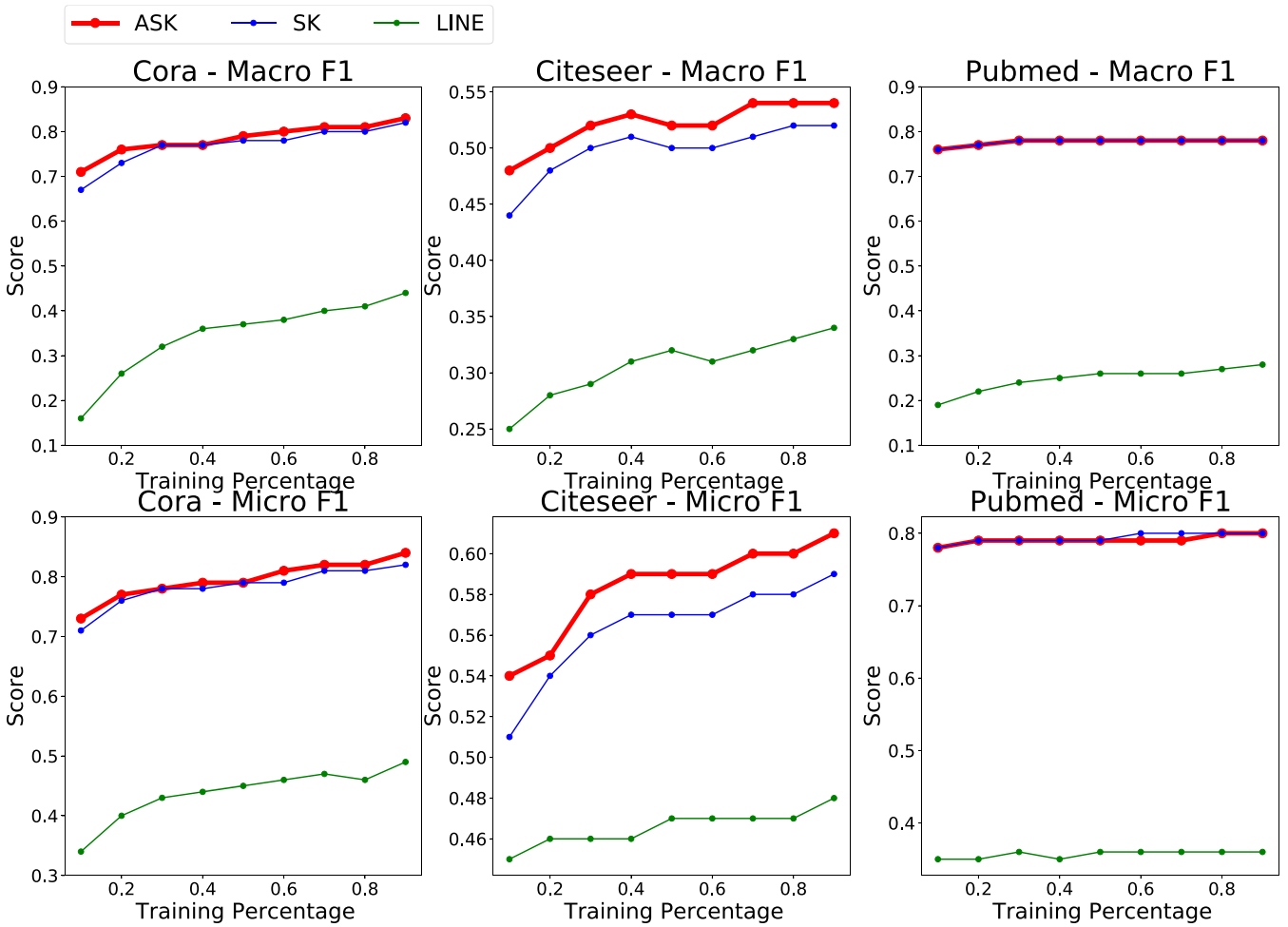


FIGURE 2. Results of node classification for SK, ASK, and LINE.

samples, and the detailed time in processing time (PT) and training time (TT). PT is the time cost of random walk process or PPR computation, and TT records the time from the first epoch to the epoch where the loss is convergent.

For link prediction on citation networks, as exhibited in Table 1, the results show both ASK and AASK with higher m lead to better performance on AUC scores than SK and LINE. We think it is because our ASK can consider PPR to select representative contexts while SK and LINE cannot precisely capture the neighborhood information. Regarding the AASK, the time cost would increase dramatically and surpass ASK because the iteration matrix is getting non-sparse. It suggests that AASK with $m = 5$ or 10 can be more appropriate than ASK when there is some requirement on run time. For link prediction on social networks, Table 2 shows that ASK leads to higher scores and lower training cost than the competing methods. Though the social networks contain highly-dense user connections, ASK and AASK with $m = 20$ can detect and exploit the most crucial substructure to learn node embeddings. Moreover, we can find that ASK has better training efficiency (i.e., lower time cost) than AASK due to that the latter requires heavier matrix computation in non-sparse network structures. The social networks with high structure density also prohibit LINE from learning well. SK employs the same size of sampled random walk paths, but it is hard to capture enough information

TABLE 5. Total time cost in seconds, and the detailed time cost for node classification.

	Cora	PT(s)	TT(s)	Citeseer	PT(s)	TT(s)	Pubmed	PT(s)	TT(s)
SK	107.06	8.23	98.83	130.1	9.46	120.64	230.32	71.38	158.94
ASK	4.78	1.24	3.53	5.57	1.57	4	143.42	120.77	22.65

In Table 3 and Table 4, it clearly demonstrates that random walk in ASK can efficiently find crucial structural contexts for nodes, especially for larger networks (i.e., PT on Pubmed). Such results imply that the performance of the random-walk sampling model is highly dependent on the number of repetitive sampling. That is, we find that SK requires higher time cost to sample and learn as the network gets dense. Besides, it also affects the time cost of the following training steps. Instead, ASK utilizes top-k selecting and the PPR probability weighting in the objective so that the learning volume of each epoch can be reduced.

C. CONVERGENCE ANALYSIS FOR SK AND ASK

We analyze the convergence of SK and ASK. We also discuss the disadvantages of SK that our ASK can overcome. In Figure 1, the testing AUC scores for link prediction on Cora data, and the loss of ASK and SK are displayed in (a) and (b), respectively. The vertical lines in the figures indicate the timestamps of the epoch of SK at 25.3 (sec) and 50.4 (sec) as the beginning of the 2-nd epoch and the 3-rd epoch. In Figure 1a, we can clearly observe that the AUC score increases over time. However, the convergence time of ASK is less than one epoch of SK but SK would not start growing until the 2-nd epoch. We think that SK needs to balance the effect between the positive loss and negative one, as first shown in Figure 1b. In the 1-st epoch, the model makes the negative loss decrease, but the positive loss is retained at the same level, and then focuses on reducing the positive loss in the next epochs. In other words, since the correlated nodes are still far away from each other, the accuracy would not be raised at the beginning. Though negative sampling help estrange the non-correlated nodes, it still has a trade-off in delaying the training efficiency. Our ASK utilizes a more precise selection of positive samples, and therefore avoiding the undesired effect of negative sampling.

D. LABEL CLASSIFICATION OF SK AND ASK

We also conduct the node label classification task for SK, ASK and LINE. The number of labels for Cora, Citeseer and Pubmed are 7, 6 and 4, respectively. We first learn node embeddings from the network, and then employ a one-vs-rest logistic regression classifier with L2 regularization on randomly select training and testing samples. The percentage of the training set is varied from 10% to 90%. We utilize Micro-F1 and Macro-F1 as the evaluation metrics. Higher scores indicate better performance. For the experiments conducted for the task of node classification, as shown in Figure 2, the proposed ASK has significant performance improvement over LINE. We think LINE cannot produce higher scores because they cannot effectively explore and exploit the neighboring substructure surrounded by each node to learn node embeddings. Besides, according to the scores are shown in Figure 2, ASK has a slight improvement in accuracy for small networks, and the performance of ASK and SK on Pubmed are close [5] because the sampling distributions for larger networks would be more well-approximating.

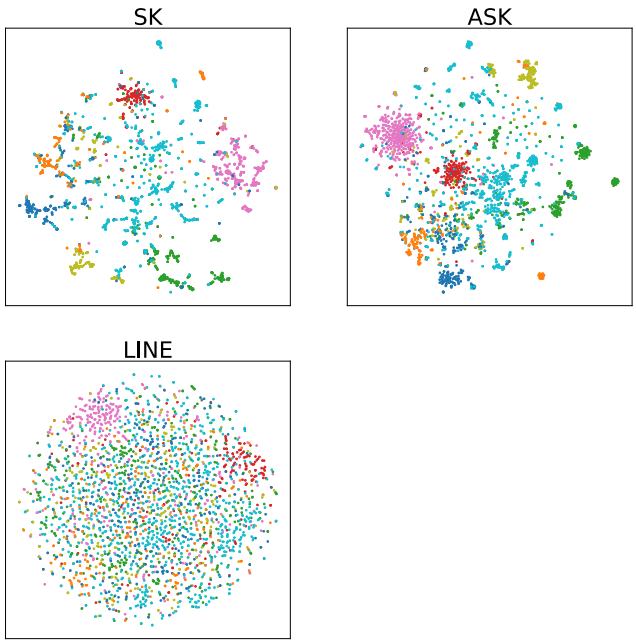


FIGURE 3. Visualizing embeddings of each method for Cora data.

Besides, we summarize the time cost of ASK and SK in Table 5. It can also be apparently found that the run time of our ASK is significantly less than SK. Such results again prove the efficiency of ASK. In detail, during the training, the time cost of ASK and SK are dropped. We think that classification is the uncomplicated version of link prediction, which only needs to model the correlation between nodes and rare labels. Therefore, the model can recognize the labels by learning the shallow structure. Especially, our PPR scores can offer more significant candidates, so the time cost is clearly decreased.

E. EMBEDDING VISUALIZATION

To present the properties of node embeddings generated by different models, i.e., to exhibit whether similar nodes are close in the embedding space, we employ t-SNE [34] to

[5]The differences |ASK-SK| between the results of ASK and SK are smaller than 0.005 from label percentage 0.1 to 0.9 for Macro and Micro, and therefore their lines are almost overlapping.

visualize node embeddings using the Cora dataset. t-SNE can reduce the embedding vector of each node to two dimensions, and generate the corresponding visualization plot of the embedding space. The results are shown in Figure 3, in which each node is colored based on its labels. It can be found that both ASK and SK have more compact and well-separated clusters than LINE, with respect to labels. By looking into the details, ASK can well separate nodes with different labels into various groups, which explains its outstanding performance on both node classification and link prediction. It is worthwhile noticing that ASK learns to separate dissimilar nodes from each other without applying negative sampling, which has been adopted by SK and brings heavier computational cost as shown in Table 1 and Table 2.

V. CONCLUSION AND DISCUSSION

In this paper, we design a more efficient and effective Skip-gram model ASK that requires no random walk for network representation learning. ASK overcomes the problems of the cost of hyperparameter decision and imprecise learning for the Skip-gram model with random walk. Since the hyperparameters, such as the number of walks, and walk length, increase the training complexity, we derive the adaptive probability based on PPR, which is equivalent to the random walk process, to avoid the inefficient sampling process. Then, the Adaptive SKip-gram model via the estimated probability of k-most significant nodes would precisely make the highly-correlated nodes close, and therefore the objective function can quickly achieve the convergence without negative sampling and even have better performance. We also consider an approximated method as a light version of Adaptive SKip-gram model using a small m, which has an efficient performance when the running environment is limited. The proposed Adaptive SKip-gram model can be seamlessly used for random walk Skip-gram based network representation learning models, such as node2vec and DeepWalk so that the efficiency and the effectiveness can get boosted.

According to the derivation and the experiments, we can depict three novel insights obtained by this work. First, we create the connection between neighborhood sampling and node correlation estimation based on PPR. We accordingly develop the ASK model, which demonstrates that PPR derivation can generate high-quality node embeddings for different downstream tasks. Second, the original skip-gram model cannot adaptively arrange and utilize the node correlation in the process of embedding learning, and thus it requires negative sampling to distinguish the differences between nodes. Our experimental results show negative sampling is not necessary, and a proper design of adaptive context discovery mechanism with PPR can simultaneously boost the performance and reduce the computational cost. Third, some potential redundant sampling and biased estimation used by SK and LINE can mislead the embedding quality, which further affects not only performance but also time cost.

We discuss the strength and weakness of the proposed ASK in the following. The strength of this work is three-fold. (1) ASK reduces the number of tuning hyperparameters, which facilitates the training of network embeddings. (2) ASK requires no negative sampling for precise embedding learning and low computational cost, comparing with the original Skip-gram model that needs negative sampling. (3) With a light neural network structure, ASK still outperforms Skip-gram models across two tasks (node classification and link prediction) and six datasets (citation and social graphs). The major limitation of our ASK model lies in its shallow model architecture. ASK is designed to preserve few-hop neighborhood. However, deeper implicit correlation between neighbors and even between local clusters cannot be encoded by ASK. In addition, currently ASK is devised for preserving graph structure in node embeddings, rather than modeling node attributes. One needs to come up with attribute-aware random walk [35] so that ASK can receive adaptive neighbors for generating node embeddings in attributed graphs.

Finally, we summarize three-folds future directions to improve work. First, we aim to exploit these insights and to adaptively find key neighbors for end-to-end node representation learning in graphs, i.e., extending the adaptive neighborhood to the realm of graph neural networks. Second, while both ASK and SK focus on learning node embeddings in simple graphs, it is worthwhile to incorporate node attributes into adaptive neighborhood sampling and representation learning. Third, we believe the idea of our proposed PPR-based adaptive mechanism can be used to not only simple graphs, but also bipartite graphs. By exploiting to generate collaborative neighbors in user-item bipartite graphs, we will examine to construct a recommender system without negative sampling.

REFERENCES

[1] W. L. Hamilton, "Graph representation learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 14, no. 3, pp. 1–159, 2020.

[2] N. N. Daud, S. H. A. Hamid, M. Saadoon, F. Sahran, and N. B. Anuar, "Applications of link prediction in social networks: A review," *J. Netw. Comput. Appl.*, vol. 166, Sep. 2020, Art. no. 102716.

[3] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social Network Data Analytics*. Boston, MA, USA: Springer, 2011, pp. 115–148.

[4] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. Comput. Appl.*, vol. 108, pp. 87–111, Apr. 2018.

[5] Z. Guo and H. Wang, "A deep graph neural network-based mechanism for social recommendations," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2776–2783, Apr. 2021.

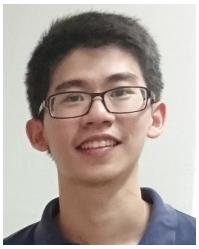
[6] Z. Guo, K. Yu, Y. Li, G. Srivastava, and J. C.-W. Lin, "Deep learning-embedded social Internet of Things for ambiguity-aware social recommendations," *IEEE Trans. Netw. Sci. Eng.*, early access, Jan. 5, 2021, doi: 10.1109/TNSE.2021.3049262.

[7] Z. Guo, K. Yu, A. Jolfaei, A. K. Bashir, A. O. Almagrabi, and N. Kumar, "A fuzzy detection system for rumors through explainable adaptive learning," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 12, pp. 3650–3664, Dec. 2021.

[8] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec," in *Proc. 11th ACM Int. Conf. Web Search Data Mining (WSDM)*, Feb. 2018, pp. 459–467.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, May 2015, pp. 1067–1077.

[10] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 701–710.
[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.
[12] S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, and A. Alemi, "Watch your step: Learning node embeddings via graph attention," 2017, *arXiv:1710.09599*.
[13] F. Hutter, L. Kotthoff, and J. Vanschoren, *Automated Machine Learning: Methods, Systems, Challenges*. Cham, Switzerland: Springer, 2019.
[14] W. Jia, C. Xiu-Yun, Z. Hao, X. Li-Dong, L. Hang, and D. Si-Hao, "Hyper-parameter optimization for machine learning models based on Bayesian optimization," *J. Electron. Sci. Technol.*, vol. 17, no. 1, pp. 26–40, 2019.
[15] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., 1999.
[16] F. Chung and W. Zhao, "Pagerank and random walks on graphs," in *Fete of Combinatorics and Computer Science*. Berlin, Germany: Springer, 2010, pp. 43–62.
[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[18] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.
[19] J. Chen, Q. Zhang, and X. Huang, "Incorporate group information to enhance network embedding," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1901–1904.
[20] J. Li, J. Zhu, and B. Zhang, "Discriminative deep random walk for network classification," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1004–1013.
[21] B. Perozzi, V. Kulkarni, H. Chen, and S. Skiena, "Don't walk, skip!: Online learning of multi-scale network embeddings," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2017, pp. 258–265.
[22] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "Struc2vec: Learning node representations from structural identity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 385–394.
[23] W. Xiao, H. Zhao, V. W. Zheng, and Y. Song, "Vertex-reinforced random walk for network embedding," in *Proc. SIAM Int. Conf. Data Mining*, 2020, pp. 595–603.
[24] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "ANRL: Attributed network representation learning via deep neural networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3155–3161.
[25] Vaibhav, P.-Y. Huang, and R. Frederking, "RWR-GAE: Random walk regularization for graph auto encoders," 2019, *arXiv:1908.04003*.
[26] W. Dou, W. Zhang, Z. Weng, and Z. Xia, "Graph embedding framework based on adversarial and random walk regularization," *IEEE Access*, vol. 9, pp. 1454–1464, 2021.
[27] H. Gao and H. Huang, "Deep attributed network embedding," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3364–3370.
[28] X. Huang, Q. Song, Y. Li, and X. Hu, "Graph recurrent networks with attributed random walks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 732–740.
[29] X. Li, W. Wei, X. Feng, and Z. Zheng, "Representation learning of reconstructed graphs using random walk graph convolutional network," 2021, *arXiv:2101.00417*.
[30] J. Klicpera, A. Bojchevski, and S. Günnemann, "Predict then propagate: Graph neural networks meet personalized PageRank," 2018, *arXiv:1810.05997*.
[31] E. Faerman, F. Borutta, K. Fountoulakis, and M. W. Mahoney, "LASAGNE: Locality and structure aware graph node embedding," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2018, pp. 246–253.
[32] Y. Zhang, X. Xia, X. Xu, F. Yu, H. Wu, Y. Yu, and B. Wei, "Robust hierarchical overlapping community detection with personalized pagerank," *IEEE Access*, vol. 8, pp. 102867–102882, 2020.
[33] B. Rozemberczki, C. Allen, and R. Sarkar, "Multi-scale attributed node embedding," *J. Complex Netw.*, vol. 9, no. 2, p. cnab014, 2021.
[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.
[35] I.-C. Hsieh and C.-T. Li, "CoANE: Modeling context co-occurrence for attributed network embedding," *IEEE Trans. Knowl. Data Eng.*, early access, May 14, 2021, doi: 10.1109/TKDE.2021.3079498.



I-CHUNG HSIEH received the master's degree in science and statistical science from the National Chung Cheng University, Chiayi, Taiwan, in 2017. He is now a Research Assistant at Networked Aritficial Intelligence Laboratory, National Cheng Kung University, Tainan, Taiwan, since 2018. His research interests include privacy protection on graph, graph representation learning, data mining, and deep learning. Graph research has been accepted and presented at NeurIPS GRL 2019. Recently, two papers in the graph field have been published in the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

CHENG-TE LI (Member, IEEE) received the Ph.D. degree from the Graduate Institute of Networking and Multimedia, National Taiwan University, in 2013. He is currently an Associate Professor with the Department of Statistics, Institute of Data Science, National Cheng Kung University (NCKU), Tainan, Taiwan. Before joining NCKU, he was an Assistant Research Fellow at CITI, Academia Sinica, from 2014 to 2016. His research interests include machine learning, deep learning, data mining, social networks and social media analysis, recommender systems, and natural language processing. He has a number of papers published at top conferences, including KDD, TheWebConf (WWW), ICDM, CIKM, SIGIR, IJCAI, ACL, EMNLP, NAACL, and ACM-MM. He leads the Networked Artificial Intelligence Laboratory (NetAI Lab), NCKU.