Article

An Information-Explainable Random Walk Based Unsupervised Network Representation Learning Framework on Node Classification Tasks

Xin Xu ¹ , Yang Lu ¹, Yupeng Zhou ¹, Zhiguo Fu ¹, Yanjie Fu ^{2,*} and Minghao Yin ^{1,*}

- ¹ Department of Computer Science, College of Information Science and Technology, Northeast Normal University, Changchun 130117, China; xux894@nenu.edu.cn (X.X.); luy179@nenu.edu.cn (Y.L.); zhouyp605@nenu.edu.cn (Y.Z.); fuzg432@nenu.edu.cn (Z.F.)
- ² Department of Computer Science, College of Engineering and Computer Science, University of Central Florida, Orlando, FL 32816, USA
- * Correspondence: yanjie.fu@ucf.edu (Y.F.); ymh@nenu.edu.cn (M.Y.)

Abstract: Network representation learning aims to learn low-dimensional, compressible, and distributed representational vectors of nodes in networks. Due to the expensive costs of obtaining label information of nodes in networks, many unsupervised network representation learning methods have been proposed, where random walk strategy is one of the widely utilized approaches. However, the existing random walk based methods have some challenges, including: 1. The insufficiency of explaining what network knowledge in the walking path-samplings; 2. The adverse effects caused by the mixture of different information in networks; 3. The poor generality of the methods with hyper-parameters on different networks. This paper proposes an information-explainable random walk based unsupervised network representation learning framework named Probabilistic Accepted Walk (PAW) to obtain network representation from the perspective of the stationary distribution of networks. In the framework, we design two stationary distributions based on nodes' self-information and local-information of networks to guide our proposed random walk strategy to learn representational vectors of networks through sampling paths of nodes. Numerous experimental results demonstrated that the PAW could obtain more expressive representation than the other six widely used unsupervised network representation learning baselines on four real-world networks in single-label and multi-label node classification tasks.

Keywords: network representation learning; random walk; stationary distributions; unsupervised learning; network embedding

Citation: Xu, X.; Lu, Y.; Zhou, Y.; Fu, Z.; Fu, Y.; Yin, M. An Information-Explainable Random Walk Based Unsupervised Network Representation Learning Framework on Node Classification Tasks. *Mathematics* **2021**, *9*, 1767. <https://doi.org/10.3390/math9151767>

Academic Editor: Massimiliano Ferrara

Received: 22 June 2021

Accepted: 22 July 2021

Published: 27 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Network Representation Learning (NRL) plays a crucial role in many real-world network analysis applications, such as the protein-protein interaction [1], the community detection [2], and the evaluation of doctors [3]. The NRL aims to learn low-dimensional, compressible, and distributed representational vectors of nodes in networks. Although supervised methods [4,5] are more adaptable for specific tasks to obtain more expressive representation, the problematic and expensive costs of getting nodes' labels cause that many unsupervised network representation learning methods [6–11] have been proposed for NRL. Among unsupervised network representation learning methods, random walk is one of the widely used strategies.

The main idea of the random walk based methods is to design various transition strategies (transition matrices) to sample walking paths, and then feed those paths into the skip-gram model [12] to obtain representations. In previous studies, the random walk based methods package all prior information, such as self-information (nodes' attribute) and local-information (knowledge from neighbors), into a transition matrix to sample

node walking paths. In 2014, Deepwalk was proposed to learn NRL by utilizing a uniform random walk strategy (transfer to any one of neighbors with equal probability) for sampling node walking paths [6]. The node2vec algorithm was proposed to sample node walking paths by employing two parameters to adjust the proportion of the deep-first sampling and the breadth-first sampling [7].

The random walk based methods are frequently used in limited time costs or training data because they can obtain the higher-quality representations with the lower required training time (easy implementation of distributed computation) and the lower required training data. Besides, the random walk strategy can improve the performance of the neural network based representation learning methods by depicting local structure patterns of networks [13]. Therefore, it is meaningful that researching random walk strategies to improve the expressive ability of learned representational vectors by including more accurate network structure information.

Although random walk based methods have been widely researched in many studies, there are some challenges to be addressed. First, it is insufficient for explaining what network knowledge is included in the node walking paths. Previous works which explain the specific knowledge of networks in the sampled walking paths are few. They generally claim that the local context of nodes is embedded into paths. Take the Deepwalk as an example. Besides the local context of nodes, the knowledge of nodes' degree is also included in walking paths, which means nodes with a higher degree are sampled under a higher probability, while some complicated random walk strategies cannot explain the specific network knowledge embedded in paths. Second, the mixture information could bring adverse effects when network representational vectors are learned from the sampled walking paths with various knowledge. Similar to the heterogeneity [14], different knowledge decides different contexts of nodes in sampled walking paths. Learning representational vectors by meta-knowledge can obtain exact ones. Third, due to the poor generality of random walk based approaches with hyper-parameters on different networks, the methods have to adapt their hyper-parameters to obtain the best representational vectors. The reason for the poor generality, essentially, is that the methods try to explore a strategy to embed the mixture information of all network knowledge into the sampled walking paths. If a method aims to extract one kind of specific knowledge, there is no need for a complicated sampling strategy with hyper-parameters. Facing those challenges, this paper tries to answer the following questions:

1. How to explicitly explain what network knowledge in sampled walking paths;
2. How to learn more expressive representational vectors of networks by designing one random walk strategy without hyper-parameters to sample the walking paths with each specific network information individually.

Since each random walk strategy has one certain transition matrix (in most cases, the matrix is not easily known.), an arbitrary random walk obeys only one specific stationary distribution when the sampling is on a connected aperiodic network [15]. In other words, a random walk strategy implicitly samples walking paths of nodes under a certain stationary distribution on a connected aperiodic networks (Since any unconnected networks can be transformed into connected ones by adding edges having minimal transition probabilities between nodes [16] and are generally aperiodic in the real world; when this paper mentions networks, it means connected aperiodic networks). This brings a novel idea of designing a random walk based method for NRL from the perspective of stationary distributions. We can first explicitly provide a stationary distribution based on a specific network knowledge, then design a random walk strategy that obeys this distribution to embed the knowledge into sampled walking paths.

Unlike the implicit sampling in the existing random walk based methods, the explicit consideration of stationary distributions brings two advantages. On the one hand, we could choose a specific network knowledge and extract the knowledge by sampling walking paths. For example, when a random walk strategy follows a stationary distribution designed by the nodes' own information (like the degree of nodes), the crucial nodes (with a high

degree) will be frequently sampled and often appear in walking paths. When the walking paths are sampled by a random walk obeying a stationary distribution designed by the neighborhood information of nodes (like the degrees of neighbors), the nodes that have more important neighbors (with a high degree) are probably sampled into walking paths. In this way, we could explicitly explain what network knowledge is embedded in sampled walking paths.

On the other hand, explicit consideration of various stationary distributions can alleviate the adverse effects of a mixture of information. Many studies have been proposed to address the adverse effects of mixed data in various applications [17–19]. The networks with abundant information in nodes or edges could provide various stationary distributions based on the network knowledge. According to every one of the distributions to sample a set of walking paths, the walking paths include single knowledge of networks. The representational vectors of the network learned by the walking paths with single network knowledge could reduce the adverse effects of the mixture of network knowledge.

In this paper, we proposed an explainable random walk based network representation learning framework to obtain more expressive representational vectors of networks. Specifically, we first explicitly designed two stationary distributions to extract specific network knowledge individually. Second, we proposed a general random walk strategy without hyper-parameters called Probabilistic Acceptive Walk (PAW), which could obey an arbitrary stationary distribution to sample walking paths. Third, we utilized the PAW to sample walking paths under each given stationary distribution and put the paths into the skip-gram model [12] to obtain two kinds of representational vectors with specific network information. After that, the Principal Component Analysis (PCA) [20] method is used to fuse the two kinds of representational vectors into a unified feature space and a low-dimension latent representation of networks would be reselected. Numerous experimental results demonstrate that the proposed framework dominates six unsupervised network representation learning methods on four real-world datasets with two kinds of node classification tasks.

Concretely, our contributions are following:

1. We proposed a novel random walk based network representation learning framework from the perspective of stationary distributions, explicitly explaining what network knowledge is embedded in sampled walking paths;
2. Individual extraction of different network knowledge could alleviate the adverse effects of a mixture of network knowledge;
3. We proposed a random walk strategy that can obey the arbitrary given stationary distribution;
4. Numerous experimental results demonstrated that the proposed framework could obtain more expressive representation than six popular used unsupervised network representational learning methods on four real-world datasets with two kinds of node classification tasks.

2. Related Work

This section will briefly introduce some methods related to this paper and roughly divide them into two categories. On one hand, the random walk is an important strategy for unsupervised network representation learning. The random walk based methods design various transfer strategies to obtain walking paths, then feed those paths into the skip-gram model [12] to learn representational vectors. The Deepwalk method [6] utilized a uniform random walk strategy for sampling walking paths of nodes to learn the representational vectors. The node2vec [7] was proposed to sample walking paths of nodes for capturing the structure information by utilizing two hyper-parameters of adjusting the deep-first search and the breadth-first research. In random walk studies, the stationary distribution is an important property that has been utilized in many works [21–23]. Random walk strategies designed by various explicit stationary distributions can learn representational vectors from sampled walking paths of nodes from the perspectives of specific network

information. It can help researchers easily extend random walk strategies and extract network information as much as possible. On the other hand, the second category of methods is based on the similarity between nodes, such as the LINE [8], the struc2vec [10], the SDNE [9], and the NetMF [11]. The first three methods utilized the similarity between vertices in the network to learn network representation. The NetMF pre-processes the network into an adjacency matrix then decomposes the adjacency matrix to obtain the embedding matrix.

3. Preliminary

In this section, we will introduce some important concepts mentioned in this paper.

Connected Aperiodic Network: A connected aperiodic network is a network with the following properties:

- For all pairs of nodes (u, v) , there exists a path that starts at node u and ends at node v ;
- The greatest common divisor of the lengths of all walk paths starting from a node and returning the same node is 1.

Stationary Distribution: The stationary distribution is defined as a probability vector π^t satisfying

$$\pi^t = \pi^{t-1} * P, \quad (1)$$

where P is a transition probability matrix. π^t is the probability vector at the t step. In a connected aperiodic network, no matter what the initial distribution π^0 of the random walk is, the probability distribution will converge to a unique stationary distribution when a transition probability matrix is given [24].

Detailed Balance Condition: For a random walk on a connected aperiodic network $G(V, E, P)$ with the probability P on edges $e \in E$, if the vector π satisfies

$$\pi_u * p_{uv} = \pi_v * p_{vu}, \quad (2)$$

for all node u and node v , $e_{uv} \in E$, $p_{uv}(p_{vu})$ is the transition probability from node $u(v)$ to node $v(u)$, $\sum_{x \in V} \pi_x = 1$, then the π is the stationary distribution of the random walk. This condition is called Detailed Balance Condition [24].

Self-distribution: From the perspective of nodes' properties, we design the self-distribution for guiding the PAW method to sample the properties' walking paths. For a node $u \in V$, we define the self-distribution of the node u as

$$\pi_u^{self} = \frac{\deg(u)}{2|E|}, \quad (3)$$

where $\deg(u)$ is the degree of the node u , and $|E|$ is the number of edges. The π_u^{self} is a distribution as it satisfies $\pi_u^{self} \in [0, 1]$, and $\sum_{u \in V} \pi_u^{self} = 1$. The nodes with high degrees have a high probability of being sampled in walking paths, aligning with the actual situation that the nodes with higher degrees are more accessible.

Neighbor-distribution: From the perspective of the neighborhood's properties, we design the neighbor-distribution for guiding the PAW method to sample the properties' walking paths. The Neighbor-distribution of a node u is defined as

$$\pi_u^{local} = \frac{\sum_{v \in N(u)} \deg(v)}{\sum_{x \in V} \deg(x)^2}, \quad (4)$$

where $N(u)$ is the set of neighbors of the node u , $\pi_u^{local} \in [0, 1]$, and $\sum_{u \in V} \pi_u^{local} = 1$. The neighbor-distribution could be used to describe the situation of neighbors of nodes. The nodes with the high value of neighbor-distribution will be frequently sampled into walking paths because they have some neighbors who are visited with high probability from other nodes. It is similar to the idea in the Pagerank [16] that when a node has

important neighbors (frequently be visited), the node should be important (frequently be visited as well from its neighbors).

Intuitively, the two kinds of network information could be utilized to distinguish the local status of nodes. The four typical local status are illustrated in Figure 1. The learned representation of nodes can depict the local status through the walking paths sampled under the two distributions, respectively.

Figure 1. Four kinds of local status. Combinations of the self-distribution and the neighbor-distribution can depict the different local status. The nodes with the Internal structure have the high values of two distributions; the nodes with the Internal-outer structure have the high value of the self-distribution and the medium value of the neighbor-distribution; the nodes with the Outer structure have the low values of both two distributions; the nodes with the Outer-internal structure have the low value of the self-distribution and the medium value of the neighbor-distribution.

4. Methodology

This section will, first, present each part of our proposed framework in detail. Second, we introduced the Probabilistic Accepted Walk (PAW) strategy and provided proof that the PAW can obey an arbitrary given stationary distribution. Third, we utilize the Principal Component Analysis (PCA) method to fuse two kinds of representational vectors into a unified space to obtain low-dimensional latent representations of nodes.

4.1. Framework

The proposed framework is illustrated in Figure 2. First, two kinds of stationary distributions were designed based on nodes' self-properties and neighborhood properties individually in the framework. Second, the PAW strategy sampled walking paths under each of the proposed stationary distributions individually, then feed the paths containing different information into two skip-gram models to learn vector-based representations separately. Third, two representational vectors of nodes were fused into a unified embedding space by the PCA algorithm to reselect the low-dimensional latent representational vectors.

Figure 2. Framework Overview. We first propose two stationary distributions. According to each of the proposed stationary distributions, the PAW strategy starts with a random node (node 5) and then uniformly takes one of its neighbors (node 3) as a selected target. If the transfer is accepted, the walk will move to the neighbor (node 3 is sampled into paths). Otherwise, the walk will stay at the current position (node 5) and reselect a neighbor (node 1). The random walk continuously repeats the above process until the specified path length is reached. Next, we separately feed the paths into two skip-gram models to obtain vector-based representations. Finally, we utilize the principal component analysis algorithm to fuse two representational vectors of nodes into a unified space and obtain low-dimensional latent representations.

The pseudo-code of our framework is shown in Algorithm 1. Formally, a network $G = (V, E)$, where V denotes the set of nodes, and E denotes the set of edges. Line 1 is for initialization of two representations. Line 2 is for the calculation of two stationary distributions. Lines 3–11 display that the PAW strategy obtains different sampled walking paths under different stationary distributions (the next subsection will propose the PAW strategy in detail). Lines 12–13 show that two sets of walking paths are fed into two skip-gram models separately to obtain vector-based representations. The skip-gram model is described in [6]. Line 14 connects Φ^{self} and Φ^{local} in sequence and fuses the two representations into a unified feature space by the PCA algorithm.

Algorithm 1 Framework

Require: network $G(V, E)$; window size w ; embedding size $d^{separate}$, d^{merge} ; walks per vertex γ ; walk length t ;

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

- 1: Initialization the representations: $\Phi^{self}, \Phi^{local}$;
- 2: Calculate the stationary distributions π^{self}, π^{local} according to Equations (3) and (4);
- 3: **for** $i = 0$ to γ **do**
- 4: $\mathbf{O} = Shuffle(V)$;
- 5: **for each** $v_i \in \mathbf{O}$ **do**
- 6: $w_{vi}^{self} = PAW(G, v_i, t, \pi^{self})$;
- 7: Add w_{vi}^{self} into W^{self}
- 8: $w_{vi}^{local} = PAW(G, v_i, t, \pi^{local})$;
- 9: Add w_{vi}^{local} to W^{local}
- 10: **end for**
- 11: **end for**
- 12: $SkipGram(\Phi^{self}, W^{self}, w, d^{separate})$;
- 13: $SkipGram(\Phi^{local}, W^{local}, w, d^{separate})$;
- 14: $\Phi = PCA(\Phi^{self}, \Phi^{local}, d^{merge})$;
- 15: **return** Φ ;

4.2. PAW: Probabilistic Accepted Walk

In this subsection, we introduce the PAW strategy in detail. The strategy is inspired by the probabilistic acceptance of sampling in the Metropolis-Hastings algorithm [25].

We propose an acceptance probability to decide whether to accept the transfer from the current position to the neighbor node. The acceptance probability from node u to node v is defined as:

$$\alpha_{uv} = \min\left(1, \frac{\pi_v * p_{vu}}{\pi_u * p_{uv}}\right), \quad (5)$$

where π_u is the stationary distribution of node u , π_v is the stationary distribution of node v , p_{uv} is the transfer probability from node u to node v , $p_{uv} = \frac{1}{\deg(u)}$, p_{vu} is the transfer probability from node v to node u , $p_{vu} = \frac{1}{\deg(v)}$.

The pseudo-code of the PAW strategy is shown in Algorithm 2. In line 1, one node is selected as the start node. Lines 3–8 show the mechanism of random jumping with a minimal probability to maintain the connected property of graphs. In Line 9, one neighbor of the current position becomes the target of the transfer by the uniform selection. In lines 10–11, we calculate the acceptance probability, and then compare it with a random number. If a random number is lower than the acceptance probability, the step of the walk is accepted, and then the walk moves to the neighbor (Lines 13–15). Otherwise, the step of the walk is rejected. The walk stays in the current position and then reselects a neighbor (Line 17). To sample a node path, the walk repeats the above processes until the path satisfies the required length of paths.

Algorithm 2 PAW: Probabilistic Accepted Walk

Require: network $G(V, E)$; start node s ; walk length t ; stationary distribution π ; probability of random jumping β

Ensure: a walk path W ;

```

1:  $len = 0, u = s, W = [s]$ ;
2: while  $len < t$  do
3:   if  $\text{random}() < \beta$  then
4:     uniform Random Select a node  $v$ ;
5:     Add  $v$  into  $W$ ;
6:      $len = len + 1$ 
7:   continue;
8:   end if
9:   uniform Random Select a neighbor  $v$  of node  $u$ ;
10:  calculate  $\alpha_{uv}$  according to Equation 5;
11:   $r = \text{random}(), r \in [0, 1]$ ;
12:  if  $r < \alpha_{uv}$  then
13:    Add  $v$  into  $W$ ;
14:     $u = v$ ;
15:     $len = len + 1$ ;
16:  else
17:    continue;
18:  end if
19: end while
20: return  $W$ ;
```

To state the PAW strategy obeying an arbitrary stationary distribution, inspired by [25], we prove the following Proposition 1.

Proposition 1. *Supposing π is an arbitrary distribution, the PAW strategy obeys the stationary distribution π .*

Proof. For the distribution π , the distribution of the node u is π_u , the distribution of the node v is π_v . The node v is one of the neighbors of node u . The transition probability from

node u to node v is p_{uv} , and from node v to node u is p_{vu} . The acceptance probability from node u to node v is α_{uv} , and from node v to node u is α_{vu} . We have

$$\begin{aligned}\pi_u * p_{uv} * \alpha_{uv} &= \pi_u * p_{uv} * \min\left(1, \frac{\pi_v p_{vu}}{\pi_u * p_{uv}}\right) \\ &= \pi_v * p_{vu} * \min\left(1, \frac{\pi_u * p_{uv}}{\pi_v * p_{vu}}\right) \\ &= \pi_v * p_{vu} * \alpha_{vu}.\end{aligned}\quad (6)$$

We consider $p'_{uv} = p_{uv} * \alpha_{uv}$ as the transition probability from node u to node v , and $p'_{vu} = p_{vu} * \alpha_{vu}$ as transition probability from node v to node u . We can rewrite Equation (6) as

$$\pi_u * p'_{uv} = \pi_v * p'_{vu}.\quad (7)$$

According to the detailed balance condition, the PAW strategy obeys the stationary distribution π . $\square$

4.3. Fusion of Feature Spaces

Principle Component Analysis (PCA) is widely used for extracting low-dimensional latent features in a unified embedding space. Since the PAW samples the walking paths from two stationary distributions, the learned representational vectors belong to different embedding spaces. Simply sequential connection of the two kinds of features need to be fused into a unified embedding space. After obtaining the sequential connection of the two kinds of features in one line, we utilize the PCA algorithm to fuse the vectors into one unified embedding space for obtaining high-quality low-dimensional latent vectors.

5. Experiments

In this section, we evaluated the proposed framework on four real-world datasets. We assessed our method both on the multi-label and single-label classification tasks, and the experimental results demonstrated improvements over baselines in most cases.

5.1. Datasets

We employed four real-world datasets to evaluate the proposed method's performance comprehensively, and the detailed statistics of these datasets are listed in Table 1. In the table, $|V|$ is the number of nodes; $|E|$ is the number of edges; the 'Yes' of Multi-label means the dataset is for multi-label classification tasks; otherwise, it is for single-label classification tasks; the 'Labels' means the number of labels of nodes.

Table 1. Statistics of Datasets.

Dataset	$ V $	$ E $	Multi-label	# Labels
BlogCatalog	10,312	333,983	Yes	39
PPI	3890	76,584	Yes	50
Hamilton	2312	96,393	No	7
Mich	3745	81,901	No	10

BlogCatalog [26] is a social relationship network of online bloggers. The vertex labels represent the interests of online users.

Protein-Protein Interactions (PPI) [27] is a subgraph of the PPI network representing relationships between proteins for Homo Sapiens. The labels obtained from the hallmark gene sets represent biological states.

Hamilton, Mich [28] are the social friendship networks extracting from Facebook networks at two American institutions, respectively, and they consist of nodes and edges representing people's relationships. The labels of nodes are the majors of users.

5.2. Compared Algorithms

Six widely used methods are utilized as baselines compared with our framework on the single-label and multi-label classification tasks.

DeepWalk [6] randomly selects one neighbor with equal probability and moves to the neighbor, repeating the selecting and moving until meeting the length of paths. It treats walking paths as sentences equally and uses the skip-gram model [12] to learn the latent representations.

LINE [8] proposes a new definition of similarity, consisting of the first-order similarity and second-order similarity to represent the closeness between vertices. This method optimizes a designed objective function, which preserves both the global and local graph structures. **node2vec** [7] extends the DeepWalk and proposes a biased random walk, capturing biased randomness and structural equivalence. It comprehensively considers depth-first search and breadth-first search.

SDNE [9] proposes a deep structure based model to capture the highly non-linear graph structure, which has multiple layers of non-linear functions, and to preserve the network structure by jointly exploiting the first-order and second-order proximity.

struc2vec [10] measures the node similarity at different scales using a hierarchy and encodes structural similarities by constructing a multilayer graph and generating structural context for nodes.

NetMF [11] regards random walk strategies as various matrix factorization methods. It connected the skip-gram based graph embedding algorithms and the theory of graph Laplacian, then presents the algorithm for computing network embedding.

We proposed two variants of our method ('PAW128' and 'PAW256') with the number of walks as 80, the walk length as 40, the window size as 10, and $d^{separate} = 128$ for both of them. For the PAW128, we set $d^{merge} = 128$. For the PAW256, we set $d^{merge} = 256$. The probability of a random jump to other nodes sets 0.001. For baselines, we set the dimension of vector-based representations in all the baselines as 128; the window size in the NetMF as 'Large'; set LINE with the 2-step representation; other parameters are set as default in corresponding papers.

5.3. Experiment Results

To demonstrate that the proposed framework could learn more expressive representational vectors on node classification tasks, we compare our method with other baselines on four real-world datasets (two datasets for multi-label classification tasks and the other two datasets for single-label classification tasks). We utilize a one-vs-rest logistic regression [6] to evaluate the learned representational quality of all methods. We randomly sample a certain fraction of the nodes during the training phase as the training set, and the remaining nodes are used for testing. We repeat this process 10 times to calculate the average values on the F1 metric.

5.3.1. Multi-label Classification

We employ two widely utilized multi-label datasets for our experiments. In these datasets, each node is assigned one or more labels.

BlogCatalog: In this experiment, we increase the training proportion of datasets from 10% to 90%. Our proposed method is significantly stronger than other baselines on the Micro-F1 metric shown in Figure 3, the Macro-F1 metric shown in Figure 4, and the Weighted-F1 metric shown in Figure 5. In addition, our method is particularly prominent on the Macro-F1 metric, which illustrates the effectiveness of classifying the rare categories on the BlogCatalog dataset. The struc2vec method is not shown, because it has poor performance on all metrics from 10% to 90% training proportion (Micro-F1:0.1-0.14; Macro-F1:0.04-0.05; Weight-F1:0.09-0.1) on the BlogCatalog dataset. The gap of experimental results between PAW128 and PAW256 is not apparent, which means that the dimension of representational vectors has not much impact on results.

PPI: As is shown in Table 2, in the high proportion of training data (50% or higher), PAW256 has the best performance than other methods. In the low proportion of training data (lower than 50%), the proposed method loses the best performance (only lower than the NetMF method) on the PPI dataset. In the low proportion of training data, the drawback of the proposed method probably is caused by not enough training in the skip-gram model. When we focus on comparing the proposed method with other random walk based methods (DeepWalk, node2vec), we could find our method performs better than the random walk based methods. The reason for the disastrous performance of node2vec is that its default hyper-parameters are not adapted for this dataset. This illustrated the poor generalize of random walk based methods with hyper-parameters from one side.

In the above multi-label classification missions, the random walk based methods generally perform better than the node-similarity based methods. We guess the node-similarity based methods may need specific node-similarity to improve their performance in corresponding missions.

Figure 3. Micro-F1 of multi-label classification on the BlogCatalog dataset on varying the proportion of training data.

Figure 4. Macro-F1 of multi-label classification on the BlogCatalog dataset on varying the proportion of training data.

Figure 5. Weighted-F1 of multi-label classification on the BlogCatalog dataset on varying the proportion of training data.

Table 2. Results of multi-label classification on the PPI dataset with the ratio of training data from 10% to 90% on the Micro-F1 evaluation (%).

Data	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
PPI	DeepWalk	16.35	18.46	19.82	20.76	21.48	22.09	22.65	23.19	23.61
	struct2vec	16.19	17.65	18.62	19.24	19.64	19.95	20.22	20.42	20.57
	node2vec	9.60	11.35	12.44	13.38	14.08	14.67	15.14	15.56	15.90
	LINE	8.41	9.35	9.89	10.34	10.56	10.82	10.95	11.06	11.14
	SDNE	8.54	9.95	11.06	11.97	12.68	13.18	13.69	14.09	14.46
	NetMF	18.19	20.51	21.87	22.69	23.16	23.76	24.25	24.67	24.75
	PAW128	17.11	19.88	21.44	22.35	22.99	23.62	24.04	24.45	24.73
	PAW256	17.15	19.98	21.47	22.38	23.38	23.86	24.29	24.79	25.29

5.3.2. Single-Label Classification

In this experiment, we randomly select the proportion of nodes from 10% to 90% as the training set and use the rest of the nodes to evaluate the performance. Our results are shown in Table 3, and we bold the highest results in each column of each dataset.

The PAW256 got the best performances in the two datasets with all proportions of training data. Although the proposed methods (PAW128, PAW256) performed a slight advance from some baselines (especially the NetMF), compared with the random walk based methods (DeepWalk, node2vec), the proposed approaches are better with apparent improvements. Comparing the different dimensions of representational vectors, we found that low-dimension (128) representation performs slightly weak than the high-dimension (256), which is caused by the low-dimension losing network information.

5.3.3. Limitations

In some cases, like the description in the above subsection (Multi-label Classification tasks), there is not enough training for the skip-gram model in our proposed method with a lower proportion of training data. In this paper, the skip-gram model in our proposed framework compares with the other random walk based methods (DeepWalk, node2vec) to demonstrate the effectiveness of the individual extraction of network knowledge. If we apply a model that required less training data instead of the skip-gram model, our method might avoid the performing drawbacks in a lower proportion of training data. The methods based on the similarity of nodes (LINE, SDNE) did not get satisfactory performance in the above classification tasks, which is caused, in our opinion, by the default similarity not for specific missions. The method (struct2vec) aims to find nodes with similar local structures leading to the nodes are close to each other in the embedding space, even if they are far

apart in the original network. Therefore, it is probable that the methods are not suitable for the classification tasks in this paper.

Overall, the above experimental results demonstrated that the representational vectors from the sampled walking paths with individual-specific network information could obtain more expressive representation on four real-world networks with single-label and multi-label node classification tasks. Furthermore, the probability accepted walk strategy (PAW) had demonstrated a bright application prospect where it could be instead of the random walk part of all random walk based methods.

Table 3. Results of single-label classification using the Micro-F1 evaluation (%) on the Hamilton and the Mich datasets.

Data	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%
Hamilton	DeepWalk	26.02	27.28	28.13	28.88	29.59	30.17	30.66	31.11	31.39
	struct2vec	20.90	21.60	22.04	22.38	22.73	23.16	23.55	23.56	23.76
	node2vec	26.82	27.69	28.25	28.78	29.03	29.39	29.63	30.18	30.40
	LINE	23.32	24.03	25.52	26.69	27.45	28.36	29.09	29.84	30.38
	SDNE	26.95	28.13	28.75	29.31	29.83	30.12	30.12	30.46	30.75
	NetMF	27.11	28.08	28.80	29.39	29.78	30.21	30.50	30.92	30.99
	PAW128	27.13	28.22	28.98	29.58	30.16	30.35	30.59	30.78	31.23
	PAW256	27.17	28.49	29.17	29.81	30.23	30.73	31.12	31.33	31.58
Mich	DeepWalk	36.58	39.31	40.94	42.15	42.89	43.47	43.99	44.52	44.70
	struct2vec	28.04	29.40	30.40	30.96	31.43	31.72	31.82	32.03	32.25
	node2vec	36.92	39.37	41.02	42.04	42.86	43.53	44.28	44.47	44.88
	LINE	32.94	35.44	36.90	38.08	39.04	40.09	40.53	41.22	41.46
	SDNE	34.41	36.40	37.49	38.44	38.97	39.40	40.10	40.48	40.39
	NetMF	37.26	39.78	41.22	42.28	43.06	43.67	44.37	44.70	45.03
	PAW128	37.58	40.23	41.75	42.64	43.45	43.90	44.63	44.95	45.01
	PAW256	37.92	40.47	42.02	43.07	43.78	44.30	44.64	45.01	45.13

6. Conclusions

In this paper, we proposed an unsupervised network representation learning framework of an information-explainable random walk strategy from the perspective of stationary distributions. First, two stationary distributions based single network knowledge were proposed for guiding random walk to sample walking paths. Individual extraction of different network knowledge alleviates the adverse effects of a mixture of network knowledge, thus improving representational vectors' express ability. Second, we proposed a random walk strategy named Probability Accept Walk to adapt multiple distributions. The strategy makes random walk obey an arbitrary given stationary distribution. Third, we utilized the principal component analysis algorithm to fuse two original feature spaces into a united space and reselect the latent representation.

The proposed framework could explain the specific knowledge of networks embedded in sampled walking paths. Independent learning representational vectors based on each kind of network knowledge could alleviate the adverse effects of mixed network knowledge. To adapt various distributions, the probabilistic accepted walk (PAW) algorithm has the ability to obey arbitrary given stationary distributions. The probability accepted walk strategy (PAW256) outperformed in four real-life data compared to all other six unsupervised network representation learning methods. The results achieved are 0.4311 (Micro-F1), 0.2985 (Macro-F1), 0.4094 (Weighted-F1) for BLOGCATALOG, 25.29 (Micro-F1) for PPI, 31.58 (Micro-F1) for Hamilton, and 45.13 (Micro-F1) for Mich at 90% of training data for PAW 256. In the future, we would like to explore designing various stationary distributions to extract various specific information of networks and attempt novel learning models instead of the skip-gram model to improve the performance of representation.

Author Contributions: Investigation: X.X., Y.L.; Methodology: X.X., Y.F., Z.F. and M.Y.; Supervision: Y.F. and M.Y.; Writing—original draft: X.X.; Writing—review&editing: X.X., Y.L., Y.Z., M.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Fundamental Research Funds for the Central Universities 2412018QD022, NSFC (under Grant No. 61976050, 61972384) and Jilin Provincial Science and Technology Department under Grant No. 20190302109GX.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study.

References

1. Cannistraci, C.V.; Alanis-Lobato, G.; Ravasi, T. Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics* **2013**, *29*, i199–i209. [CrossRef] [PubMed]
2. Tu, C.; Wang, H.; Zeng, X.; Liu, Z.; Sun, M. Community-enhanced network representation learning for network analysis. *arXiv* **2016**, arXiv:1611.06645.
3. Xu, X.; Fu, Y.; Xiong, H.; Jin, B.; Li, X.; Hu, S.; Yin, M. Dr. right!: Embedding-based adaptively-weighted mixture multi-classification model for finding right doctors with healthcare experience data. In Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM), Singapore, 17–20 November 2018; pp. 647–656.
4. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
5. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
6. Perozzi, B.; Al-Rfou, R.; Skiena, S. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 701–710.
7. Grover, A.; Leskovec, J. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 855–864.
8. Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web, Florence, Italy, 18–22 May 2015; pp. 1067–1077.
9. Wang, D.; Cui, P.; Zhu, W. Structural deep network embedding. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1225–1234.
10. Ribeiro, L.F.; Saverese, P.H.; Figueiredo, D.R. struc2vec: Learning node representations from structural identity. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 385–394.
11. Qiu, J.; Dong, Y.; Ma, H.; Li, J.; Wang, K.; Tang, J. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, 5–9 February 2018; pp. 459–467.
12. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
13. Jin, Y.; Song, G.; Shi, C. *GraLSP: Graph Neural Networks with Local Structural Patterns*; AAAI: Menlo Park, CA, USA, 2020; pp. 4361–4368.
14. Higgins, J.; Thompson, S.G. Quantifying heterogeneity in a meta-analysis. *Stat. Med.* **2002**, *21*, 1539–1558. [CrossRef] [PubMed]
15. Lambiotte, R.; Delvenne, J.C.; Barahona, M. Random walks, Markov processes and the multiscale modular organization of complex networks. *IEEE Trans. Netw. Sci. Eng.* **2014**, *1*, 76–90. [CrossRef]
16. Page, L.; Brin, S.; Motwani, R.; Winograd, T. *The PageRank Citation Ranking: Bringing Order to the Web*; Technical Report; Stanford InfoLab: Stanford, CA, USA, 1999. Available online: <http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf> (accessed on 17 June 2021).
17. Caruso, G.; Gattone, S.A. Waste Management Analysis in Developing Countries through Unsupervised Classification of Mixed Data. *Soc. Sci.* **2019**, *8*, 186 [CrossRef]
18. Donnat, C.; Miolane, N.; Bunbury, D.S.P.F.; Kreindler, J. A Bayesian Hierarchical Network for Combining Heterogeneous Data Sources in Medical Diagnoses. In Proceedings of the Machine Learning for Health NeurIPS Workshop, 2020; PMLR 136:53–84. Available online: <http://proceedings.mlr.press/v136/donnat20a.html> (accessed on 17 June 2021).
19. Hu, P.; Huang, Y.A.; Chan, C.C.K.; You, Z.H. Learning Multimodal Networks from Heterogeneous Data for Prediction of lncRNA-miRNA Interactions. *IEEE-ACM Trans. Comput. Biol. Bioinform.* **2020**, *17*, 1516–1524. [CrossRef] [PubMed]
20. Tipping, M.E.; Bishop, C.M. Probabilistic principal component analysis. *J. R. Stat. Soc. Ser. B* **1999**, *61*, 611–622. [CrossRef]
21. Hastings, W.K. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **1970**, *57*, 97–109 [CrossRef]
22. Green, P.J. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **1995**, *82*, 711–732. [CrossRef]

-
23. Carpenter, B.; Gelman, A.; Hoffman, M.D.; Lee, D.; Goodrich, B.; Betancourt, M.; Brubaker, M.; Guo, J.; Li, P.; Riddell, A. Stan: A probabilistic programming language. *J. Stat. Softw.* **2017**, *76*, 1–32. [[CrossRef](#)]
 24. Blum, A.; Hopcroft, J.; Kannan, R. Foundations of data science. *Vorabversion Eines Lehrb.* **2016**, 76–86. Available online: <https://www.cs.cornell.edu/jeh/book.pdf> (accessed on 17 June 2021)
 25. Chib, S.; Greenberg, E. Understanding the metropolis-hastings algorithm. *Am. Stat.* **1995**, *49*, 327–335.
 26. Tang, L.; Liu, H. Relational learning via latent social dimensions. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, Paris, France, 28 June 2009; pp. 817–826
 27. Stark, C.; Breitkreutz, B.J.; Chatr-Aryamontri, A.; Boucher, L.; Oughtred, R.; Livstone, M.S.; Nixon, J.; Van Auken, K.; Wang, X.; Shi, X.; et al. The BioGRID interaction database: 2011 update. *Nucleic Acids Res.* **2010**, *39*, D698–D704. [[CrossRef](#)] [[PubMed](#)]
 28. Traud, A.L.; Mucha, P.J.; Porter, M.A. Social structure of facebook networks. *Phys. A Stat. Mech. Appl.* **2012**, *391*, 4165–4180. [[CrossRef](#)]