Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2025-0160

Vol. **40**, No. 1, pp. 176-198
July 2026

# Network Embedding Using Sparse Approximations of Random Walks

Paula Mercurio[1,*] and Di Liu[2]

[1] *Department of Mathematics, Hamline University, Minneapolis, MN, US.*
[2] *Department of Mathematics, Michigan State University, East Lansing, MI, US.*

Received 24 June 2025; Accepted (in revised version) 2 September 2025

**Abstract.** In this paper, we propose an efficient numerical implementation of Network Embedding based on commute times, using sparse approximation of a diffusion process on the network obtained by a modified version of the diffusion wavelet algorithm. The node embeddings are computed by optimizing the cross entropy loss via the stochastic gradient descent (SGD) method with sampling of low-dimensional representations of green functions. We demonstrate the efficacy of this method for data clustering and multi-label classification through several examples, and compare its performance over existing methods in terms of efficiency and accuracy. Theoretical issues justifying the scheme are also discussed.

**AMS subject classifications**: 05C81, 68R10, 05C62

**Key words**: Network Embedding, commute times, diffusion wavelets, SVD.

## 1 Introduction

Networks and graphs provide versatile and intuitive representations for many real world situations. A wide variety of relational data, e.g., social networks, protein-protein interactions, bio-chemical reacting systems, citation networks, and many others, can be represented as graphs and networks. As a result, methods for analyzing these networks have consistently garnered much interest. *Network Embedding*, or *Feature Learning for Graphs*, which provides a major framework for network analysis, seeks to map each node in a given graph to a point in a low-dimensional vector space such that proximity information from the original graph is preserved. In particular, the embeddings of nodes that are closely related in the original network through edge or path connections, mutual neighbors, overlapping proximities, will be close to each other in the embedding space.

*Corresponding author. *Email addresses:* pmercurio01@hamline.edu (P. Mercurio), liudi1@msu.edu (D. Liu)

 

In many machine learning applications, the lower dimensional vectors representations are more convenient to exploit for downstream tasks than the original network. In this paper, we focus on node embedding methods, instead of embedding edges or whole networks, for their benefit of being more adaptive and applicable for a variety of practical situations such as node classification, clustering, and link prediction [5]. Node classification assigns class labels to unlabeled nodes based on samples of labeled nodes, while clustering algorithms group representations of similar nodes together in the target vector space, and link prediction is used to predict edges based on data containing edge information.

Most node embedding methods can be categorized according to the general techniques used to compute the embedding function including 1.) matrix factorization, 2.) random walk sampling and 3.) deep learning [12]. Matrix factorization methods, such as Laplacian Eigenmaps [3], Locally Linear Embedding [23] and Graph Factorization [1], involve factoring a matrix representation of the network (e.g. adjacency matrix, Laplacian, transition probabilities) by eigenvalue or singular value decompositions to determine embedding maps. Due to its ability to model nonlinear structural information, Deep Neural Networks have also been used recently in Network Embedding, in which the embedding and decoding functions consist of multiple stacked neural network layers.

Random walk methods collect information about conditional probabilities of node observations by running short random walk trials on the network, then obtain embeddings via Stochastic Gradient Descent (SGD) minimization of cross entropies. Like many of the Monte Carlo techniques, random walk methods apply to large high-dimensional networks with few limitations. In DeepWalk [20], based on the SkipGram language model [18, 19], closely correlated nodes are embedded near each other according to the frequency of co-occurrence within unbiased, fixed length random walks on the graph. Node2Vec [13] replaces the simple unbiased random walks with walks controlled by adaptive parameters to either backtrack and stay near the starting node, or explore higher degree neighbors. A recent generalization for directed networks was proposed in [7] with applications to the study of transition pathways of metastable chemical reacting networks.

In this paper, we propose an improved implementation of random walk methods for node embeddings preserving the *commute time* [14], based upon a sparse approximation of random walks on the graph. The approach achieves efficiency by avoiding Monte Carlo simulations of finite time random walks, and takes into account statistics of paths of arbitrary length between different nodes. The commute time is defined as the expected time for a random walk to travel from one node $u$ to another node $v$, and then back to the starting node $u$. It can also be viewed as a time integration of the *diffusion distance* introduced in [9], which is defined through the probability of the random walk traveling from $u$ to $v$ in fixed time $t$. One advantage of utilizing the commute time, rather than the diffusion distance, is that it removes the need to choose an appropriate value for the parameter of time $t$.

Numerically, the commute time between two nodes can be computed from the Green