Survey on graph embeddings and their applications to machine learning problems on graphs

Ilya Makarov[1,2], Dmitrii Kiselev[1], Nikita Nikitinsky[3] and Lovro Subelj[2]

[1] HSE University, Moscow, Russia
[2] Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia
[3] Big Data Research Center, National University of Science and Technology MISIS, Moscow, Russia

ABSTRACT

Dealing with relational data always required significant computational resources, domain expertise and task-dependent feature engineering to incorporate structural information into a predictive model. Nowadays, a family of automated graph feature engineering techniques has been proposed in different streams of literature. So-called graph embeddings provide a powerful tool to construct vectorized feature spaces for graphs and their components, such as nodes, edges and subgraphs under preserving inner graph properties. Using the constructed feature spaces, many machine learning problems on graphs can be solved via standard frameworks suitable for vectorized feature representation. Our survey aims to describe the core concepts of graph embeddings and provide several taxonomies for their description. First, we start with the methodological approach and extract three types of graph embedding models based on matrix factorization, random-walks and deep learning approaches. Next, we describe how different types of networks impact the ability of models to incorporate structural and attributed data into a unified embedding. Going further, we perform a thorough evaluation of graph embedding applications to machine learning problems on graphs, among which are node classification, link prediction, clustering, visualization, compression, and a family of the whole graph embedding algorithms suitable for graph classification, similarity and alignment problems. Finally, we overview the existing applications of graph embeddings to computer science domains, formulate open problems and provide experiment results, explaining how different networks properties result in graph embeddings quality in the four classic machine learning problems on graphs, such as node classification, link prediction, clustering and graph visualization. As a result, our survey covers a new rapidly growing field of network feature engineering, presents an in-depth analysis of models based on network types, and overviews a wide range of applications to machine learning problems on graphs.

INTRODUCTION

Many instances in the real world can be modeled as graphs or networks. Some of the typical examples include social interactions, biological data, such as protein interactions or neural connections, links between websites on the Internet, etc. One of the main goals of graph modeling is to formulate a general technique capable of processing structural data including relations between objects, which may also have some domain-specific information. For example, given a social network, we might be interested in predicting whether a pair of users are friends, or in identifying communities of interconnected users. The former leads to a link prediction problem on the graph, while the latter describes a node clustering problem.

We focus on graph representation theory, aiming to automatically learn low-dimensional vector features for the simplest graph motifs, such as nodes and edges, in a way that would enable efficiently solve machine learning problems on graphs including node classification, link prediction, node clustering, while also tackling approaches for graph similarity and classification, and general aspects of graph visualization.

Before the emergence of the area, the extraction of important features for predictive tasks on graphs had to be manually engineered. It required a lot of efforts from the domain experts. For example, many approaches for graph representation rely on extracting summary statistics, such as vertex degrees or clustering coefficients (*Bhagat, Cormode & Muthukrishnan, 2011*) popular in social sciences, graph kernels (*Vishwanathan et al., 2010*) particularly used in computational biology to compute inner product similarities between graphs, or specifically designed features to measure neighborhood similarity (*Liben-Nowell & Kleinberg, 2007*). In addition to the time-consuming feature engineering, such summaries were very inflexible, task/data-dependent, and did not generalize well across different prediction tasks on graphs. An alternative methodology is to learn feature representations automatically as an optimization problem. The goal is to design objective cost functions that capture dependencies and similarities in a graph while preserving high quality in relational machine learning tasks and constructing graph embeddings under efficiency constraints over time and memory.

Today, there exists a large variety of graph embeddings automatically extract vector representation for networks (*Moyano, 2017*; *Hamilton, Ying & Leskovec, 2017b*; *Cai, Zheng & Chang, 2017*; *Cui et al., 2018*; *Goyal & Ferrara, 2017*; *Chen et al., 2018a*; *Wu et al., 2019b*), knowledge graphs (*Nickel et al., 2016*) and biological data (*Su et al., 2020*). Some of these algorithms only work with structural information, such as popular Node2vec (*Grover & Leskovec, 2016*), LINE (*Tang et al., 2015*), DeepWalk (*Perozzi, Al-Rfou & Skiena, 2014*), while others like GCN (*Kipf & Welling, 2016a*), GraphSAGE (*Hamilton, Ying & Leskovec, 2017a*), VGAE (*Kipf & Welling, 2016b*) also use node attributes. The methods also differ based on whether a given graph is (un)directed, (un)weighted, (non-)attributed, (dis)assortative, if it changes over time in terms of adding/deleting nodes/edges, and whether they use a transductive or inductive approach for learning network dynamics inference. All of these models have their advantages and shortcomings, but what unifies them is the unique pipeline to verify the network embedding model in terms of the quality

of machine learning tasks on benchmark datasets. In addition, authors measure construction and inference time efficiency, memory consumption, and a possibility to include graph dynamics in the model.

Most surveys on graph embeddings provide a simple taxonomy for graph models based on how the model is fitted and only show applications within the graph domain, for example, node classification or link prediction (*Moyano, 2017*; *Hamilton, Ying & Leskovec, 2017b*). *Goyal & Ferrara (2017)* provide experiments and study the influence of hyperparameters on different tasks. Some works focus on a specific field such as attention models (*Lee et al., 2019*) and graph neural networks (*Wu et al., 2019b*; *Chen et al., 2018a*; *Zhang, Cui & Zhu, 2018*). *Cui et al. (2018)* compare models in terms of what information they preserve: structure and properties or side information. Neural network approaches are usually classified by the core architecture, for example, recurrent neural networks (RNN) or convolutional neural networks (CNN), and losses for different tasks, such as cross-entropy for link prediction and node classification and reconstruction loss for unsupervised representation learning. *Chen et al. (2018a)* provides meta-strategies for choosing embedding models, but examine only deep learning based methods. *Lee et al. (2019)* follow the classification of *Cai, Zheng & Chang (2017)* and separate attention models by type of input and output, deriving recommendations for working with different graphs (heterogeneity, multi-view, directed acyclic graphs) and on different tasks (node classification, clustering, ranking, alignment, link prediction). *Zhang, Cui & Zhu (2018)* is quite similar to other GNN surveys, but also provides an overview of modern models and tasks like reinforcement learning on graphs, analyses techniques for better representation learning like sampling strategies, skip connections, inductive learning and adversarial training.

In contrast, our work tries to generalize the advances of previous surveys. Our survey is not limited to specific model types and provides an overview from different angles: training process, input graph properties, specific tasks and applications in a non-graph domain, and open problems, etc.

The paper is structured as follows. We start with a brief explanation of general approaches to learn network embedding and introduce to a reader the core ideas of graph representation models. Next, we describe different models adapted to specific types of networks. Then, we state the most crucial machine learning problems on graphs and solutions to them based on network embeddings. To cover the use of overviewed models, we provide applications to other machine learning domains. We finalize review sections with the listing of open problems in the field of network representation learning.

Finally, we provide our experiments to understand in practice, how different graph embeddings perform on benchmark network datasets and interpret, why the chosen graph embedding model with a given training setting result in good or bad quality on a given benchmark dataset and how it is related to the method behind the model. Our experiment section aims to show how one can choose the best graph embedding by the nature of the model construction and network descriptive statistics, which is one the most interesting problems for practical applications of graph embeddings for machine learning frameworks.

PRELIMINARIES

Before describing any methods we need to introduce some definitions. We will use V as a set of graph vertices, E as a set of graph edges, A as graph adjacency matrix and $G(V, E)$ as graph description. The procedure on constructing vector representation of a graph we are interested in is called graph embedding.

Definition 1 (Graph embedding) *is a mapping from a collection of substructures (most commonly either all nodes, or all edges, or certain subgraphs) to $\mathbb{R}^d$. We will mostly consider* **node embeddings***: $f : V \to \mathbb{R}^d,\ d \ll |V|$.*

For many graph-based tasks, the most natural task formulation is unsupervised learning: this is the case when we need to learn embeddings using only the adjacency matrix A containing information on structural similarity and possibly **attributed features** X, but without task-specific loss part. It is also possible that there are labels available for some substructures of the graph, and we wish to recover missing labels in a semi-supervised approach. One example of this is node classification, in which all nodes are available from the outset, but only a fraction is labeled.

Now let us clarify what is meant by a good embedding. By the embedding procedure, one should aim to compress the data, while retaining most of the essential information about similarities and simultaneously, extract important features from the structural information. What counts as essential may vary depending on an intended application; most common properties we want to capture in a graph are termed as **node proximity** and **structural similarity** (neighbourhood information and structural role, respectively).

Definition 2 (First and second order proximities) *The first-order proximity describes the pairwise proximity between vertices. For any vertices, the weight a_{ij} (possibly zero) of the edge between v_i and v_j characterizes the first-order proximity between these vertices, thus representing adjacency matrix $A = (a_{ij})_{i,j=1}^{n}$. A neighborhood of vertex v_i is defined as a set of adjacent vertices $N_{v_i} = \{v_k | a_{ik} > 0,\ k \neq i\}$ thus meaning that vertex itself is not included in its neighborhood. The second-order proximity between a pair of vertices v_i and v_j describes the similarity measure between their neighborhood structures Nv_i and Nv_j with respect to a selected proximity measure.*

METHODS FOR CONSTRUCTING GRAPH EMBEDDING

We briefly describe graph embedding methods of three general categories, corresponding to the perspective they take on embedding graphs: matrix factorizations, node sequence methods and deep learning based methods. These are, of course, not mutually exclusive, but it is more convenient to adhere to their primary features. We also cover a specific type of embeddings based on embedding space metric.

We select papers from several curated lists and major conferences on network science, artificial intelligence, machine learning and data mining, as well as core research publishers and indexing services. Paper sources are referred in Table 1. We used the following keywords: graph/network embeddings, graph/network representation, graph neural networks, graph convolutional networks, graph convolution, graph attention,

Table 1 Paper sources.

Name	Link	Description
Curated lists		
by Chen	https://github.com/chihming/	awesome-network-embedding
by Rozemberczki	https://github.com/benedekrozemberczki/	awesome-graph-classification
by Rebo	https://github.com/MaxwellRebo/	awesome-2vec
by Soru	https://gist.github.com/mommi84/	awesome-kge
Conferences		
Complex Networks	https://complexnetworks.org/	International Conference on Complex Networks and their Applications
The Web	https://www2020.thewebconf.org/	The Web Conference is international conference on the World Wide Web.
WSDM	http://www.wsdm-conference.org/	Web-inspired research involving search and data mining
IJCAI	https://www.ijcai.org/	International Joint Conferences on Artificial Intelligence
AAAI	https://www.aaai.org/	Association for the Advancement of Artificial Intelligence
ICML	https://icml.cc/	International Conference on Machine Learning
SIGKDD	https://www.kdd.org/	Special Interest Group in Knowledge Discovery and Databases
Domain conferences		
ACL	http://www.acl2019.org/	Association for Computational Linguistics
CVPR	http://cvpr2019.thecvf.com/	Conference on Computer Vision and Pattern Recognition
Publishers		
ACM DL	https://dl.acm.org/	Full-text articles database by Association for Computing Machinery
IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp	Research published by Institute of Electrical and Electronics Engineers
Link Springer	https://link.springer.com/	Online collection of scientific journals, books and reference works
Indexing services		
Scopus	https://www.scopus.com/	Abstract and citation database
Web of Science	https://www.webofknowledge.com/	Citation Indexer
Scholar Google	https://scholar.google.com/	Web search engine for indexing full-text papers or its metadata

graph/network classification/link prediction/clustering, deep learning for graphs, geometric deep learning, GCN, GNN, GAT.

Historically the first graph embedding methods were factorization based, which generally try to approximate a large matrix with a low-rank matrix factorized into a product of two matrices containing representations, thus modeling each entry of the original matrix with an inner product of representations. Sequence-based embeddings linearize the graph using random walks or diffusion and maximize the probability of observing the neighborhood (context) of a node given its embedding. Deep learning-based models learn a function mapping a graph in the numeric form to a low-dimensional embedding by optimizing over a broad class of expressive neural network functions.

Dimensionality reduction (matrix factorization) methods

Definition 3 (Matrix factorization) *is a decomposition of a matrix to the product of matrices. In this sense, the first matrix in series is named self node representation and the last matrix refers to node context.*

Factorization models are common techniques in different machine learning domains to receive meaningful low-dimensional representation. Moreover, a lot of methods use similarity matrix between observations, which can also be reformulated as the graph similarity matrix.

Factorization techniques can be applied to a different graph representations and optimize different objectives. Some methods directly decompose the adjacency matrix A, for example, MDS (*Kruskal & Wish, 1978*) reconstructs it by minimizing MSE between element a_{ij} and euclidean distance between vectors u_i and u_j of manifold U. We can rewrite this with expression $\sum_{i=1}^{N} \sum_{j=1}^{N} \left(a_{ij} - \|u_i - u_j\|_2^2 \right)^2$. LSI (*Deerwester et al., 1990*) simply applies singular value decomposition to A *Golub & Reinsch (1971)*. In *Wold, Esbensen & Geladi (1987)* the manifolds are learned by maximizing variance for linear mixture. It is extended by LDA *Martinez & Kak (2001)*.

Another way to use dimensionality reduction is to build proximity matrix of the graph. For example, IsoMap (*Tenenbaum, De Silva & Langford, 2000*) use shortest path matrix D and apply MDS to learn embeddings. LLE (*Roweis & Saul, 2000*) learns node similarity by reconstructing weights matrix W with which neighboring nodes affect each other: $\|X - W^T U\|_2^2$ and repeats that procedure to learn manifold U with achieved matrix W. LPP (*He & Niyogi, 2004*) estimates the weighted matrix W as heat kernel and learn manifold U by reduction of W with Laplacian Eigenmaps technique. IsoMap and LLE were proposed to model global structure while preserving local distances or sampling from the local neighborhood of nodes. The lower bound for methods complexity was quadratic in the number of vertices, still making them inappropriate for large networks.

Definition 4 (Graph Laplacian) If matrix D is the diagonal degree matrix, that is $D = \text{diag}(\Sigma_j A_{ij})$, then Laplacian matrix can be defined as L = D − A.

Another approach for spectral graph clustering (*Chung & Graham, 1997*) was suggested in *Belkin & Niyogi (2002)* named Laplacian eigenmaps (LE), representing each node by graph Laplacian eigenvectors associated with its first k nontrivial eigenvalues. The goal for Laplacian Eigenmaps class of models lies in preserving first-order similarities. Thus, a model gives a larger penalty using graph Laplacian if two nodes with larger similarity are embedded far apart in the embedding space. Laplacian objective function is symmetric in each pair (i, j), and thus it cannot capture edge orientations. Kernel Eigenmaps (*Brand, 2003*) extends this approach to nonlinear cases. In contrast to LE, which preserved nodes dissimilarity, Cauchy embedding (*Luo et al., 2011*) proposes optimization condition modification which preserves the similarity between vertices. Structure Preserving Embedding (SPE) (*Shaw & Jebara, 2009*) aims to use LE combined with preserving spectral decomposition representing the cluster structure of the graph. It introduces a new graph kernel and applies SVD to it.

Graph Factorization (GF) (*Ahmed et al., 2013*) try to solve the scalability issue of factorization methods by decreasing node neighborhood via graph partitioning and utilizing distributed computation.

The models in this class can be either symmetric and obtain final representations only from embedding matrix. GraRep (*Cao, Lu & Xu, 2015*) consider k-hop neighborhood (A^k)

using SVD decomposition of A^k. HOPE (*Ou et al., 2016*) is specific asymmetric transitivity preserving graph embedding. It is found that most asymmetric similarity measures can be formulated as $S = M_g^{-1} M_l$. Katz index refers to $M_g = I - \beta A$, $M_l = \beta A$. Rooted PageRank can be stated as $M_g = I - \alpha P$, $M_l = (1 - \alpha) P$. Common neighbors is represented by $M_g = I$, $M_l = A^2$, and Adamic-Adar with $M_g = I$, $M_l = A \cdot D \cdot A$. To avoid calculation of similarity matrix authors propose to use generalized SVD and directly estimate matrices M_g and M_l. *Abu-El-Haija, Perozzi & Al-Rfou (2017)* proposed to use concatenation of two node representations capturing in- and out-connections. Authors of *Wang et al. (2017d)* proposed a Modularized Nonnegative Matrix Factorization (M-NMF) model to preserve the community structure in network representation. In ATP model (*Sun et al., 2019*) authors embed directed graph constructing two vectors for each node via factorization framework. *Kefato, Sheikh & Montresor (2020)* propose multi-objective framework for preserving directed nature of graph. SDNE (*Wang, Cui & Zhu, 2016*) uses autoencoders (as neural network based dimension reduction technique) to capture non-linear dependencies in local proximity.

Factorization based models are the best-studied theoretically and provide a well-known general framework for graph embedding optimization (*Liu et al., 2019*), however, they suffer from high computational complexity for large graphs and often capture only a small-order proximity *Perozzi, Al-Rfou & Skiena (2014)*.

Sequence-based approaches

Definition 5 (Random walk on graph) *is a sequence of nodes obtained from the random process of node sampling. Usually, probability of choice of node j after node i is proportional to $A_{i,j}$.*

Motivated by drawbacks of the matrix factorization approach, another approach emerged that attempts to preserve local neighborhoods of nodes and their properties based on random walks (*Newman, 2005*; *Pirotte et al., 2007*). More specifically, the main idea is to maximize the probability of observing the neighborhood of a node given its embedding, following the line of Skip-gram model initiated in NLP applications by *Mikolov et al. (2013)*, *Pennington, Socher & Manning (2014)*. An objective of this type can be efficiently optimized with stochastic gradient descent on a single-layer neural network, and hence has lower computational complexity.

Definition 6 (Skip-gram) *is method to learn sequence element i representation via maximization of probability of elements in context of i based on representation of i.*

Two prominent examples of models in this class are node2vec (*Grover & Leskovec, 2016*) and DeepWalk (*Perozzi, Al-Rfou & Skiena, 2014*). DeepWalk performs a random walk over a graph and then uses sampled sequences to learn embeddings, using the Skip-gram objective (while having modifications for other NLP based sequence models, such as using Glove from *Brochier, Guille & Velcin (2019)*). Its predecessor LINE (*Tang et al., 2015*) is equivalent to DeepWalk when the size of vertices' contexts is set

to one. Node2vec extends the random walk with biasing parameters of BFS or DFS parameters. Another way of sampling based on diffusion was presented in diff2vec (*Rozemberczki & Sarkar, 2018*). By virtue of sampling being more centered around source nodes, it provides robust embeddings while being less flexible.

Walklets (*Perozzi, Kulkarni & Skiena, 2016*) as a generalization of GraRep (*Cao, Lu & Xu, 2015*) use weighted combination of embeddings of powers of adjacency matrix A, A^2, …, A^k to reduce the bias of Deepwalk for low-order proximities, and approximates computing A^i by skipping nodes using short random walks (*Perozzi et al., 2017*).

The focus on the local structure and non-convex optimization requiring the use of stochastic gradient descent and proper initialization limit random walk based methods in capturing the hierarchical structure of a graph. HARP (*Chen et al., 2018b*) proposes a meta-strategy for graph embedding under recursive construction of nodes and edges into condensed graphs with similar global structure. These graphs are used as source initializations for embedding detailed graphs, resulting in the end in proper node and edge embeddings, which can be adopted for improving DeepWalk (*Perozzi, Al-Rfou & Skiena, 2014*), LINE (*Tang et al., 2015*), and Node2vec (*Grover & Leskovec, 2016*) algorithms. It was further generalized for community preserving using Modularity Maximization (*Tang & Liu, 2009*) and supporting large free-scale networks (*Feng et al., 2018*).

Alternatively, Struct2vec (*Ribeiro, Saverese & Figueiredo, 2017*) uses structural similarity without using node or edge attributes but considering graph hierarchy to measure similarity at different scales. *Liu et al. (2020c)* uses rooted substructures of a graph to preserve structural similarity. Diffusion wavelet model to capture structural proximity was suggested in *Donnat et al. (2018)*. Another approach to control hyper-parameters in random-walk methods is Graph Attention (*Abu-El-Haija et al., 2018*) learning multi-scale representation over adjacency matrix powers with the probabilistic approach for learning balancing weights for each power. It was further generalized to its deep learning analog in *Veličković et al. (2017)* and *Liu et al. (2018a)*, see also *Lee et al. (2019)* for details on attention models on graphs.

Extension of Deepwalk to heterogeneous networks was suggested in Metapath2vec (*Dong, Chawla & Swami, 2017*). Modifications of random-walk based methods using node attribute concepts and node proximities were suggested in GenVector (*Yang, Tang & Cohen, 2016b*). With GEMSEC (*Rozemberczki et al., 2018*), the authors extend sequence-based methods with additional K-means objective encouraging clustering structure-preserving in the embedding space and improving overall performance. Discriminative Deep Random Walk (DDRW) (*Li, Zhu & Zhang, 2016*) was suggested for the task of attributed network classification. *Çelikkanat & Malliaros (2019)* generalizes random walk based methods to the case of the exponential family of distributions for sampling strategies.

Sequence-based models, such as node2vec, can obtain high-quality embeddings of structural input graph by sampling node sequences and learning context-consistent embeddings but are not able to capture additional node/edge features while being transductive by their nature.

Deep learning: graph convolutions

Complex non-regular graphs structure makes graph filtering not as simply defined as on images. In the past decades, researchers have been working on the graph signal processing methods including filtering, wavelets, Fourier transformations using graph spectral domain. The studies on these methods can be found in *Shuman et al. (2013)*, *Ortega et al. (2018a)*.

Advances in deep learning have led to a new field of studies devoted to applying neural networks to graph data (*Scarselli et al., 2009*; *Li et al., 2014a*, *2014b*). Recently, SDNE (*Wang, Cui & Zhu, 2016*) and DNGR (*Cao, Lu & Xu, 2016*) use deep autoencoder to capture non-linearity in graphs and simultaneously apply dimension reduction for constructing graph embedding. SDNE use autoencoder preserving first order proximity and Laplacian Eigenmaps for penalizing long distances for embedding vectors of similar vertices. DGNR uses stacked denoising autoencoders over positive pointwise mutual information matrix obtained from similarity information based on random surfing. Both methods use global information and thus are not appropriate for large networks.

Kipf & Welling (2016a) propose Graph Convolutional Layer that offers a further simplified approximation to spectral convolution and achieves better computational efficiency for semi-supervised multi-class node classification is applicable for the other machine learning tasks. A model of several such convolutions is referred to as Graph Convolutional Network (GCN). Improvements over speed and optimization methods of training GCNs were suggested in *Chen, Zhu & Song (2017)*, *Chen, Ma & Xiao (2018)*. Stochastic approaches for network embedding optimization were briefly over-viewed in *Lei, Shi & Niu (2018)*.

Assume the graph $G(V,E)$, adjacency matrix A and feature matrix X of size (N_{nodes}, $N_{features}$), where N_{nodes} refers to number of vertices and $N_{features}$ to number of node attributes. Then, GCN can be defined as set of hidden layers $H^i = \sigma(A H^{i-1} W^{i-1})$ where H^0 is equal to matrix X, W^i is learnable weight matrix. At the next hidden layer, these features are aggregated using the same propagation rule. It means that graph convolutions aggregate feature information of its neighbors based on the adjacency matrix. The idea of graph convolutions using spatial convolutions (operating with adjacency matrix) or spectral graph methods (operating with graph Laplacian) was proposed in *Bruna et al. (2013)*, *Duvenaud et al. (2015)*, *Henaff, Bruna & LeCun (2015)*, *Niepert, Ahmed & Kutzkov (2016)*, *Defferrard, Bresson & Vandergheynst (2016)*, *Levie et al. (2017)*, while extending the GCN idea to recurrent models *Li et al. (2015c)*, *Monti, Bronstein & Bresson (2017)*, mixture models of CNNs *Monti et al. (2017)*; *Fey et al. (2018)*, diffusion convolutions *Atwood & Towsley (2016)*; *Li et al. (2017c)*, and models suitable for dynamic graphs under inductive learning paradigm *Natarajan & Dhillon (2014)*; *Hamilton, Ying & Leskovec (2017a)*. All the methods suggest semi-supervised embedding, however, choosing unique labels for each vertex one may obtain an unsupervised version of network embedding. The GraphSAINT (*Zeng et al., 2019*) provides a solution for scalability problem in training graph neural networks. It compares different topology-based sampling algorithms

(node, edge and random walks) in terms of bias and variance of learned GCN model. It also introduces unbiased estimator for node aggregation.

Another idea is to use deep autoencoders to learn compressed representations that capture the essence of the graph structure. An autoencoder includes two nonlinear functions, an encoder and a decoder, and attempts to minimize reconstruction loss. One such model specifically designed for graphs is GAE, which consists of a GCN encoder (one or two stacked GCN layers in most use cases) that produces embeddings and an inner product decoder that reconstructs the adjacency matrix ($\hat{A} = \sigma(UU^T)$, where σ is non-linearity like sigmoid function and U is embedding matrix of nodes). The weights of the model are trained by backpropagating the reconstruction loss, which is usually Mean Squared Error (MSE).

VGAE (*Kipf & Welling, 2016b*) is a probabilistic counterpart of GAE. It introduces a distribution over latent variables Z, with these variables being conditionally independent Gaussians given A and X with means (μ) and diagonal covariances (σ) being parameterized by two GCN encoders (*Kingma & Welling, 2013*). As in the case of images, VGAE just adds KL-divergence term between conditional distribution $q(Z|X,A)$ and unconditional $p(Z) \sim N(0,1)$ to the loss. After node embeddings are reconstructed via random normal distribution sampling, that is, $Z = \mu + \sigma\varepsilon$. Then adjacency matrix is decoded using inner product of achieved vector Z as in simple GAE.

In very recent work, authors of GraphSAGE (*Hamilton, Ying & Leskovec, 2017a*) offer an extension of GCN for inductive unsupervised representation learning and offer to use trainable aggregation functions instead of simple convolutions applied to neighborhoods in GCN. GraphSAGE learns aggregation functions for a different number of hops that are applied to sampled neighborhoods of different depths, which then are used for obtaining node representations from initial node features. PinSage (*Ying et al., 2018a*) extends the previous algorithm with the importance sampling based on random walks. Importance score is calculated simply as visit counts. It provides better scalability and quality. GAT (*Veličković et al., 2017*) use masked self-attention layers for learning weights balancing impact of neighbors on node embedding, and supporting both, inductive and transductive learning settings. In *Liu et al. (2018a)*, authors suggested specific layers controlling the aggregation of the local neighborhood over BFS and DFS sampling, thus generalizing Node2vec (*Grover & Leskovec, 2016*) model to graph neural networks. Similar to GCN, GAT contains several hidden layers $H^i = f(H^{i-1}, A)$, where H_0 is a graph node features. In each hidden layer linear transformation of input is firstly calculated with the learnable matrix W. The authors replace the adjacency matrix by learnable self-attention in form of a fully-connected layer with activation and further normalization with softmax. Generalization of gated recurrent graph neural networks (*Li et al., 2015c*) was suggested in Message Passing Neural Network (MPNN) (*Gilmer et al., 2017*) providing a differentiable way to combine information from neighbours.

Nowadays, many advanced deep neural network models are adapted to graph data. Graph generative adversarial networks were suggested in *Ding, Tang & Zhang (2018)* and *Yu et al. (2018)*. In *You et al. (2018)*, recurrent graph neural network was suggested for the task of graphs generation. Pooling operators for graphs were used in

Defferrard, Bresson & Vandergheynst (2016), *Ying et al. (2018b)*. *Yuan & Ji (2020)* modernize classic pooling to account graph structure using Conditional Random Fields. Adversarially regularized variational graph autoencoder (ARVGA) was suggested in *Pan et al. (2019)*. *Zhu et al. (2020a)* develop the DGGAN model that jointly learns source and target vectors for the directed graphs employing adversarial techniques. *Liu (2020)* builds Anonymized GCN with adversarial training to be robust to the noise attacks. *Hettige et al. (2020)* propose the RASE model, that applies Gaussian denoising attribute autoencoder for achieving robustness of received embedding, while *Laakom et al. (2020)* catches the uncertainty by learning probability Gaussian distributions over embedding space. *Weng, Zhang & Dou (2020)* employs adversarial training for variational graph autoencoder. *Zhu et al. (2020b)* use node feature smoothing for learn better embeddings. *Jing et al. (2020)* designs variable heat kernel to learn robust representations.

Deep Learning models are now a study of vulnerability to adversarial attacks, in particular, it relates to structural data. The first approaches for detection of node/edge add/remove mechanisms were studied in *Bojcheski & Günnemann (2018)*, *Chen et al. (2018c)*, while other researchers focused on methods for unsupervised (*Sun et al., 2018b*), semi-supervised (*Chen et al., 2018e*) and supervised (*Zügner, Akbarnejad & Günnemann, 2018*) scenarios of graph embedding construction, and application for ML problems. The black-box approach was formulated in *Dai et al. (2018)* and further covered in general overview for the problem of graph data poisoning (*Chen et al., 2019b*) and its applications to social media data (*Zhou et al., 2018*) and knowledge graphs (*Zhang et al., 2019b*). A survey of methods for defense from adversarial attacks on graphs was suggested in *Sun et al. (2018a)*.

The deep learning models propose a new way of approximation for classic graph convolutions and kernels, which allows extracting embeddings faster. A mixture of it with semi-supervised techniques gives the state-of-the-art results in terms of scalability, speed and quality on downstream tasks.

Hyperbolic (non-Euclidean) embeddings

The Euclidean space is not the best for structures like graphs, because has the low descriptive ability for hierarchical and scale-free structures. So, researchers have considered other space, that can successfully represent it in a comparatively low number of dimensions, saving the basic properties like angles. It allows using classical machine learning methods in down-streamed tasks.

In certain cases, embedding into non-Euclidean spaces may be beneficial for model performance (*Kleinberg, 2007*; *Shavitt & Tankel, 2008*; *Krioukov et al., 2009*). LEs were also used for constructing embedding in hyperbolic space (*Alanis-Lobato, Mier & Andrade-Navarro, 2016*). Deep learning approach was applied for hyperbolic embedding in *Chamberlain, Clough & Deisenroth (2017)*.

There is no exact research on the properties of embedding spaces, but researchers mostly pay attention to preserving low dimensional space, catching graph properties and model quality trade-off.

SPECIFIC EMBEDDINGS BASED ON NETWORK TYPES

In this section, we show specific embedding models generalizing core methods of network representation to a certain domain of networks and applications based on the network type.

Attributed networks

Real-world networks are often accompanied with additional features for nodes and edges, such as labels, texts, images. These attributes tend to be correlated for close graph structures and could affect network embedding by adding additional information for the similarity of nodes. The attributes are usually represented by high-dimensional vectors of features (which are sparse for just label attributes). Once the attributes are represented by their embeddings, the task is to incorporate them in network embedding model (under unsupervised or semi-supervised framework).

The authors of TADW (*Yang et al., 2015*) represent DeepWalk model as matrix factorization and incorporate text attributes into factorization framework. PLE (*Ren et al., 2016*) jointly learns the representations of entity types and links together with text features. In *Le & Lauw (2014)*, a generative model for document network embedding was suggested based on topic modeling of documents using Relational Topic Model (RTM) (*Chang & Blei, 2009*) and the relationships between the documents. In *Ganguly et al. (2016)*, authors combine text and network features for co-authorship recommendations.

Augmented Relation Embedding (ARE) (*Lin, Liu & Chen, 2005*) adds content-based features for images using graph-Laplacian spectral embedding modification. In *Geng et al. (2015)*, *Zhang et al. (2015*, *2017*), authors suggested to embed images, textual and network information for modeling user-image interaction.

In addition to structural similarity, in certain cases feature similarity may be also important. Two-layered network embedding for node-to-node and text-to-text similarities was suggested in *Sun et al. (2016)*. In *Zhang et al. (2016b)*, the authors proposed the HSCA model, embedding homophily, network topological structure and node features simultaneously. In DeepBrowse (*Chen, Anantharam & Skiena, 2017*), the authors suggested using DeepWalk-based node similarity together with priority ranking for recommender system based on an interaction graph. Label preserving attribute node embedding was suggested in Tri-party Deep Network Representation (*Pan et al., 2016*). Modifications of random-walk based methods using node attribute concepts and node proximities were suggested in GenVector (*Yang, Tang & Cohen, 2016b*).

Label attributes are also an important part for such problems as classification of nodes and edges, or community information (assigning each node a community label). Community preserving network embeddings were suggested in *Shaw & Jebara (2009)*, *Wang et al. (2017d)* and *Rozemberczki et al. (2018)*. Incorporating group information was presented in GENE model (*Chen, Zhang & Huang, 2016b*) under a supervised framework.

Semi-supervised frameworks for learning network embedding under loss constraints for labeled data were suggested in Planetoid (*Yang, Cohen & Salakhutdinov, 2016a*) Max-margin Deep Walk (*Tu et al., 2016*) and LANE (*Huang, Li & Hu, 2017*).

Heterogeneous networks

A heterogeneous network presents a different concept of graph representation, in which nodes and edges may have different types (or even multiple edges).

The heterogeneous network embeddings either learn embeddings in the same vector space (*Li, Ritter & Jurafsky, 2015*; *Zhao, Liu & Sun, 2015*), or construct the embeddings separately for each modality and then aggregate them into one space, such as HNE model (*Chang et al., 2015*) and *Tang & Liu (2011)*, or even aggregate over multiple network layers (*Xu et al., 2017*) or different relation features (*Huang, Li & Hu, 2017*).

Random-walk based approach for different node types based on DeepWalk was presented in Metapath2vec (*Dong, Chawla & Swami, 2017*). Similar approaches based on meta-path random walks for graph embedding were suggested in *Huang & Mamoulis (2017)*, *Chen & Sun (2017)*. *Jacob, Denoyer & Gallinari (2014)* use heterogeneous network embedding for node classification across different node types. A similar problem was posed for author identification on double-blind review scenario (*Chen & Sun, 2017*). Study by *Jiang et al. (2020)* provides a framework for efficient task-oriented skip-gram based embeddings. *Hu, Fang & Shi (2019)* utilizes the generative adversarial networks, which learn node distributions for efficient negative sampling. *Shi et al. (2020)* proposes a method for automatic meta-path construction.

Cao et al. (2020) use the graph attention mechanism for heterogeneous graph embedding task. MAGNN architecture (*Fu et al., 2020*) extends simple attention mechanism with several levels: node attributes, inter meta-path information and intra meta-path semantic information. DyHAN (*Yang et al., 2020*) presents the model for dynamic heterogeneous graphs with hierarchical attention. Another way to use the attention mechanism in dynamic heterogeneous networks is the *Li et al. (2020b)*. It employs three types of attention: structural, semantic and temporal.

Heterogeneous graph embeddings are widely used in real-world applications. *Hong et al. (2020)* estimates the arrival time for transportation networks, *Ragesh et al. (2020)* use it in text classification. *Chen & Zhang (2020)*, *Li et al. (2020a)* utilizes HIN embedding for multi-modal data fusion task. *Zhang et al. (2020a)* preserves the relationships in HIN. A survey on heterogeneous networks can be found in *Wang et al. (2020a)*.

Signed networks

In a signed network, each edge is associated with its weight, taking values from the set $\{1, -1\}$, which usually represents belief or opinion sentiment for different relations types. These networks are specifically considered apart from Heterogeneous networks as important objects for social network analysis, although they are still just a specific type of such networks. One of the tasks on such networks is predicting links and their signs (*Liu et al., 2015a*).

SiNE (*Wang et al., 2017c*) is a DNN model aiming at close relationships with friends (positive weight) rather than with foes (negative weight). For highly positive social networks a virtual node with negative relation is proposed to use in the model, which uses pairwise similarities optimization under constraint mentioned above. In *Yuan, Wu & Xiang (2017)*, the authors propose a local neighborhood aggregation model SNE for each

type of positive and negative relations. *Kim et al. (2018)* propose random-walks based model SIDE for signed directed networks. Also, they provide socio-psychological interpretation for each term in the loss function. SIGnet (*Islam, Prakash & Ramakrishnan, 2018*) develops new target node sampling for more efficient learning. In oppose to previous works, *Lu et al. (2019)* provides signed network embedding powered by Status Theory (*Leskovec, Huttenlocher & Kleinberg, 2010*). It natively works with directed networks by preserving node ranking except direct node similarity.

Multi-layer networks

Multi-layer networks are used to model complex systems with different levels of interaction between nodes, for example, whole Airline network with different carriers. Each layer in such networks corresponds to different types of relationships.

Liu et al. (2017a) compare three aggregation methods for single-layer network embedding models: merging of different layers in one network, single-layer vectors concatenation and between-layer random walks. The best results show the last method named layer co-analysis because it allows learning between-layer interactions. In *Xu et al. (2017)* authors provide an example of coupling into joint space two separately learned heterogeneous networks embeddings. IONE (*Liu et al., 2016*) preserves users similarity based on their followers and followees for several social networks. A hierarchy-aware unsupervised node feature learning approach for multi-layer networks was proposed in *Zitnik & Leskovec (2017)*. In *Li et al. (2018)* authors develop the single optimization framework for both within-layer and between-layer communication. It exploits spectral embedding and the block model.

Temporal networks

A lot of real-world networks are evolving over-time. Most of the described above methods concentrate on the static embeddings, so it works poorly in the temporal scenario.

Haddad et al. (2019) propose the adaptation of Node2vec model to the dynamic case. Authors also introduce the task-specific temporal embeddings. *Rossi et al. (2020a)* provide the generic framework named Temporal Graph networks for deep learning on dynamic graphs. *Fathy & Li (2020)* apply the graph attention to the temporal networks. *Zhong, Qiu & Shi (2020)* develop the model for efficient community mining. *Rokka Chhetri & Al Faruque (2020)* present the model for dynamic physics graphs. CTGCN model (*Liu et al., 2020a*) generalizes graph convolution networks with feature transformation and aggregation. It builds the hierarchical representation of the graph with K-cores and applies GCN to it. *Goyal, Chhetri & Canedo (2020)* use the recurrent neural networks to catch the dynamics. There is one more specific graph type: temporal interaction networks, such as user-item interactions in the recommender systems. *Zhang et al. (2020b)* creates the embedding approach for such graph utilizing coupled memory networks.

Nowadays, methods based on smart neighborhood aggregation, such as limiting random walks over clusters *Chiang et al. (2019)* and precomputing diffusion-based neighborhoods for one-layer GCN *Rossi et al. (2020b)* show great performance over existing approaches, thus combining advances in deep learning and neighborhood sampling methodology.

Large graphs

We have already mentioned that random walks and graph neural networks were proposed as the approximations for the different classic matrix factorization techniques. So in this section, we will discuss approaches to scale up GNN training.

The basic idea implemented in different papers is a sampling. GraphSAGE (*Hamilton, Ying & Leskovec, 2017a*) learns trainable aggregations for sampled node neighbourhood. This approach was further improved with fixed-length random walk based importance sampling of the neighborhood in *Ying et al. (2018a)*. GraphSAGE also provides the idea of minibatch training for GNNs. A similar idea was proposed in the *Chen, Ma & Xiao (2018)*. *Salha, Hennequin & Vazirgiannis (2020)* propose to use linear aggregation over direct neighbors to simplify computations. The GraphSAINT (*Zeng et al., 2019*) compares different topology-based sampling algorithms (node, edge and random walks) in terms of bias and variance of learned GCN model. It also introduces unbiased estimator for aggregation of node and normalizes propagation by this value, that solves the scalability problem.

Nie, Zhu & Li (2020) is based on the idea of Locality Preserving Projection. It works with anchor-based proximity matrices and calculates these anchors via Balanced and Hierarchical K-means. Such an approach allow to reduce complexity from n^2d to ndm where n is a number of samples, d is embedding dimension and m is a number of anchors. *Akyildiz, Aljundi & Kaya (2020)* extends the VERSE (*Tsitsulin et al., 2018*) with graph partitioning and coarsening to provide fast embedding computation on the GPU. *Atahan Akyildiz, Alabsi Aljundi & Kaya (2020)* analyzes effects of graph coarsening on different embeddings in comparison to GOSH. Another distributed training framework was presented in *Zheng et al. (2020a)*. It also provides efficient graph partitioning schemes for reducing between-machine communication. *Gallicchio & Micheli (2020)* keeps the graph embedding as the dynamical systems and study the embedding stability issue. Authors found that stable initialization allows to left weights untrained in deep sparse networks. *Lu & Chang (2020)* use softmax clustering for modularity maximization. They show that such a method is a linear approximation for main eigenvectors.

APPLICATION OF GRAPH EMBEDDINGS TO MACHINE LEARNING PROBLEMS

Here, we aim to overview core machine learning problems involving structural data. We start with problems related to small graph motifs such as nodes and edges, while further going to the problems connected to subgraphs and graphs as a whole.

Node classification

Definition 7 (Node classification) *For a given graph G(V, E) with known labels for some of nodes from V, node classification is the task of predicting missing labels for existing or newly added nodes.*

Node classification deals with assigning class labels to nodes based on labeled nodes data (*Zhu et al., 2007*; *Bhagat, Cormode & Muthukrishnan, 2011*). The structural information is used in a context that "similar" nodes should have the same/similar labels. The original

framework uses label propagation based on random walks statistics (*Xiaojin & Zoubin, 2002*; *Azran, 2007*; *Baluja et al., 2008*). In an unsupervised framework, each node is embedded in a low-dimensional space following by training a classifier on the set of labeled node embedding vectors (*Lu & Getoor, 2003*; *Bhagat, Cormode & Rozenbaum, 2009*). Authors use such machine learning models as logistic regression (*Perozzi, Al-Rfou & Skiena, 2014*; *Pimentel, Veloso & Ziviani, 2017*), SVM (*Wang, Cui & Zhu, 2016*; *Wang et al., 2017d*), kNN (*Le & Lauw, 2014*; *Wilson et al., 2014*), random forest and xgboost (*Makarov et al., 2018*; *Makarov et al., 2019c*); the choice is usually made based on the size of training data, interpretability of features and embedding dimension.

In semi-supervised framework, node embeddings are learned via loss function containing regularization for labeled data predictions, penalizing "similar" nodes to have different labels (*Li, Zhu & Zhang, 2016*; *Yang, Cohen & Salakhutdinov, 2016a*; *Tu et al., 2016*; *Kipf & Welling, 2016a*; *Monti et al., 2017*). *Zhang, Zhou & Li (2020)* proposes hierarchical GCN and pseudo-labeling technique for learning in scarce of annotated data. *Liu et al. (2020b)* proposes a sampling strategy and model compression for handling sparsity of labels. *Chen et al. (2020)* employs contrastive learning techniques to achieve semi-supervised parametrized fusion of graph topology and content information. *Zhu et al. (2020c)* also use metric learning approach but applies it to corrupted graph substructures. *Nozza, Fersini & Messina (2020)* use two-phase optimization for attributed graph embedding. *Shi, Tang & Zhu (2020)* aligns topology of attribute content network to the corresponding graph to simultaneously learn good embeddings. *Wang et al. (2020b)* propose two models for the imbalanced scenarios. A survey on classic techniques for node classification can be found in *Bhagat, Cormode & Muthukrishnan (2011)*.

Link prediction

Definition 8 (Link prediction problem (LPP)) *is a task of completing missing edges in noisy graphs or predicting new edges in temporal network structures. Formally, LPP for given graph G(V, E) with adjacency matrix A is a task of learning such function f that reconstruct or predict next adjacency matrix A based on different graph features such as metrics (e.g., Jaccard, Adamic-Adar), graph embeddings.*

Network science approach to the problem of predicting collaborations results in the link prediction (LP) problem (*Liben-Nowell & Kleinberg, 2007*) for temporal networks and missing edges reconstruction in noisy network data. Basically, it is a method to apply standard machine learning framework for graph data considering feature space consisting of pairs of nodes and their features.

One of the interesting research questions is in the way of constructing edge embedding in a non-direct combination of node embeddings, as it was suggested in component-wise embeddings (*Grover & Leskovec, 2016*) or bi-linear combination of compressed node embeddings suggested in *Abu-El-Haija, Perozzi & Al-Rfou (2017)*. Certain practical applications for drug combinations was suggested in *Zitnik, Agrawal & Leskovec (2018)*.

HARP *Chen et al. (2018b)* incorporates several hierarchical layers while transmitting information from edge embedding to node embedding. Other systems of directly

incorporating edge features and labels were suggested in CANE (*Tu et al., 2017*) and LANE (*Huang, Li & Hu, 2017*). Models of joint node and edge structure learning were proposed in Dual-Primal GCN (*Monti et al., 2018*) and ELAINE (*Goyal et al., 2018*). A model for embedding event graphs in which event is described by several edges was presented in HEBE (*Gui et al., 2016*). *Wu et al. (2020)* presents random walk with restart index. *Phuc, Yamada & Kashima (2020)* embeds several graphs with similar structural properties to boost link prediction accuracy. *Keser et al. (2020)* employs skip-connections in VGAE.

Link prediction models are applied in web linking (*Adafre & de Rijke, 2005*), social dating services (*Backstrom & Leskovec, 2011*) and paper recommender system for digital libraries (*He et al., 2010*). The reader can found an up-to-date survey in *Srinivas & Mitra (2016)*.

LPP was specifically formulated in *Liben-Nowell & Kleinberg (2007)* based on nodes pairwise similarity measures. Approaches for link prediction include similarity based methods (*Adamic & Adar (2003)*), maximum likelihood models (*Clauset, Moore & Newman, 2008*), and probabilistic models (*Getoor & Taskar, 2007*; *Heckerman, Meek & Koller, 2007*). In *Tang & Liu (2012)*, authors are suggesting unsupervised approach for LP problem. *Gao, Denoyer & Gallinari (2011)*, *Gao et al. (2015)* suggested temporal link prediction based on matrix factorization technique and noise reduction in large networks. Attribute-based link formation in social networks was studied in *McPherson, Smith-Lovin & Cook (2001)*, *Robins et al. (2007)*, while deep learning approaches were presented in *Liu et al. (2013)*, *Zhai & Zhang (2015)* and *Berg, Kipf & Welling (2017)*. Heterogeneous graph link prediction for predicting links of certain semantic type was suggested in *Liu et al. (2017b, 2018b)*. An evaluation of link prediction models based on graph embeddings for biological data was presented in *Crichton et al. (2018)*.

Two surveys on link prediction methods describing core approaches for feature engineering, that is, Bayesian approach and dimensionality reduction were presented in *Hasan & Zaki (2011)* and *Lü & Zhou (2011)*. Survey on link prediction was published in *Wang et al. (2015)*.

Node clustering

Definition 9 (Node clustering or community detection or graph partitioning) *is the task of the partitioning of a graph G(V, E) into several subgraphs $G_i(V_i, E_i)$ with a dense connection within groups and sparse connection between clusters.*

Node clustering (also known as community detection in social network analysis) aims to find such a grouping (labelling) of nodes so that nodes in the same group are closer to each other rather than to the nodes from outside of the group (*Malliaros & Vazirgiannis, 2013*). No labels are provided on initial step due to unsupervised type of the problem. Methods use attribute (*Zhou, Cheng & Yu, 2009*) or structural information. The latter methods of graph clustering are usually based on either community detection (*Newman & Girvan, 2004*; *Fortunato, 2010*) or structural equivalence (*Xu et al., 2007*). In community detection (*Shi & Malik, 2000*; *Ding et al., 2001*), the cluster is defined as

dense subgraph with a high number of edges inside subgraph, and a low number of edges between subgraph and the rest of a graph.

The general idea is to use node embeddings as a compressed representation of sparse graph adjacency matrix and then apply standard clustering algorithms, such as K-means or DBScan, for vectorized data (*White & Smyth, 2005*; *Tian et al., 2014*; *Cao, Lu & Xu, 2015*; *Chen et al., 2015b*; *Cao, Lu & Xu, 2016*; *Nie, Zhu & Li, 2017*). Going further, joint optimization of clustering and node embedding was suggested in *Tang, Nie & Jain (2016)*, *Wei et al. (2017)*. Efficient iterative community aware network embedding was proposed in *Wang et al. (2017d)* and several others (*Zheng et al., 2016*; *Cavallari et al., 2017*).

Teng & Liu (2020) propose multi-objective evolutionary algorithm for community detection. *Zhang, Shang & Jiao (2020)* use multi-objective matrix factorization over several shortest path graphs and utilizes (MOEA) to find community structure. *Salim, Shiju & Sumitra (2020)* train the embeddings on different views for preserving many properties of a given network. *Quiring & Vassilevski (2020)* employs hierarchical coarsening of the graph to better extract clusters.

Subgraph (and graph) embedding

While studying network embedding, one may think of a way to aggregate or generalize low-level node feature representation to the whole network representation, thus stating the problem of embedding the whole graph (*Song, 2018*). Such vector is required for the graph-level tasks like graph classification, similarity and clustering. It considers the whole network as one structural unit in the training dataset.

The task is relevant to chemistry or biology domains (*Nikolentzos, Meladianos & Vazirgiannis, 2017*; *Zhang et al., 2016a*; *Duvenaud et al., 2015*; *Dai, Dai & Song, 2016*; *Niepert, Ahmed & Kutzkov, 2016*; *Kearnes et al., 2016*). They can also be applied for graph reasoning (*Li et al., 2015c*) or computer vision tasks (*Bruna et al., 2013*).

In *Duvenaud et al. (2015)*, the sum based approach over network embedding was suggested. Following by it, in *Dai, Dai & Song (2016)*, authors proposed neural network aggregation for constructing network embedding which is an argument for summing over subgraph nodes. Improvement of these methods was later suggested in *Bronstein et al. (2017)* based on approximations of spectral graph decompositions. Ordered-based (*Niepert, Ahmed & Kutzkov, 2016*) and fuzzy-based (*Kearnes et al., 2016*) approaches based on aggregating features from convolutional approaches further improved subgraph embedding models. *Sun, Hoffmann & Tang (2019)* maximize the mutual information between embedding and different graph substructures.

The general approach of *Gilmer et al. (2017)* as well as other convolutional approaches can be generalized by pooling-aggregation models or, as was suggested in *Scarselli et al. (2009)*, by adding super-node for whole graph embedding. The attention mechanism was applied to the graph classification task (*Lee, Rossi & Kong, 2018*).

Definition 10 (Line (dual) graph) *For a **graph** G = (V, E) defined as a set of vertices V and a set of edges E ⊆ V × V without loops and multi-edges we denote by G⋆ = (V⋆, E⋆) a **dual (Line) graph** the nodes of which are the edges of G and edges are nodes, in the*

sense that two adjacent nodes are connected by an edge if corresponding edges have a common node incident to them.

In graph-level tasks, specific network properties play a major role. So vectors reconstructing sophisticated similarity metrics closely related to the problem of graph isomorphism was studied in several works (*Shervashidze et al., 2011*; *Niepert, Ahmed & Kutzkov, 2016*; *Mousavi et al., 2017*; *Yanardag & Vishwanathan, 2015*; *Narayanan et al., 2016*). GL2VEC (*Chen & Koga, 2019*) extends *Narayanan et al. (2016)* model with edge features by utilizing the line graph. The works on matching node embedding and graph kernels were suggested in *Johansson & Dubhashi (2015)*, *Nikolentzos, Meladianos & Vazirgiannis (2017)*. In *Donnat & Holmes (2018)* authors analyze graph-based distance methods for a temporal graph of bio-medical surveys.

Hierarchical clustering and fusion of different network representations were overviewed in *Yang & Wang (2018)*. Usually, this kind tasks require fusion of different similarity representations of a network as different graphs (*Serra, Greco & Tagliaferri, 2015*; *Xue et al., 2015*), preserving graph structure (*Hou et al., 2017*) or simultaneously performing semi-supervised classification and clustering with adaptive kNN model (*Nie, Cai & Li, 2017*). Different domain network clustering was suggested in *Cheng et al. (2013)* and improved in the following works suggesting fusion of different not-synchronized networks with different structures (*Ni et al., 2016*), cross-domain associations (*Liu et al., 2015b*) or multi-view spectral clustering (*Li et al., 2015b*). *Khasahmadi et al. (2020)* propose a memory layer for graphs, that can efficiently learn graph hierarchical representations. *Tsitsulin, Munkhoeva & Perozzi (2020)* propose an algorithm for efficient calculation of spectral distances for large graphs. *Kolouri et al. (2020)* suggest the embedding preserving Wasserstein distance with linear complexity. *Qin et al. (2020)* presents one more graph pooling technique that uniformly aggregates neighborhood. *Baldini, Martino & Rizzi (2020)* embeds maximal cliques to preserve structural similarities between graphs. *Yan & Wang (2020)* states the problem of transfer learning suggesting the framework for graph alignment and further adaptation learning for GNNs.

Network visualization

Definition 11 (Graph visualization) *is a way to map a graph to a low (2D, 3D) dimensional space.*

All nodes are either directly embedded as 2D vectors (*Le & Lauw, 2014*; *Wang, Cui & Zhu, 2016*; *Cao, Lu & Xu, 2016*; *Tu et al., 2016*; *Niepert, Ahmed & Kutzkov, 2016*; *Pan et al., 2016*) or first embedded to certain dimension, and then compressed via PCA (*Herman, Melançon & Marshall, 2000*) or t-SNE (*Maaten & Hinton, 2008*) (or other dimension reduction frameworks, see for, for example, *Tenenbaum, De Silva & Langford, 2000*, *De Oliveira & Levkowitz, 2003*) in order to plot in 2D space. If there are labels or communities representative for network dataset, the nodes are usually visualized with different colors for each label in order to verify whether similar nodes are embedded closer to each other. Such models, as *Perozzi, Al-Rfou & Skiena (2014)*, *Grover & Leskovec (2016)*, *Tang et al. (2015)*, *Ou et al. (2016)*, *Wang, Cui & Zhu (2016)* demonstrated proper

performance on the task of network visualization for unsupervised graph embedding models. Evaluation of graph embeddings for large structural data visualization can be found in *Tang et al. (2016a)*. Graph visualization techniques beyond planar mappings can be found in *Didimo, Liotta & Montecchiani (2019)*.

Network compression

Definition 12 (Network compression, simplification or sparsification) *is a task of reducing the number of nodes and edges in a graph, for further efficient application of graph algorithms.*

The concept of network compression was first introduced in *As Feder & Motwani (1991)* under the idea of reducing the number of stored graph edges while achieving a faster performance of certain algorithms on graphs. The compression was made by grouping nodes and edges into partitions of bipartite cliques and then replacing these cliques with trees. Similar ideas of dividing the graph into groups of nodes and edges and encoding them were proposed in several studies (*Pardalos & Xue, 1994*; *Tian, Hankins & Patel, 2008*; *Toivonen et al., 2011*). Minimum Description Length (MDL) (*Rissanen, 1978*) was used in *Navlakha, Rastogi & Shrivastava (2008)* to construct graph summary adjusted with edge correction algorithm.

Graph embeddings support compact graph representation, reducing memory storage from $O(|V| \times |V|)$ to $O(d \times |V|)$, where embedding dimension $d \ll n$ below 200 was shown to be enough for qualitative network reconstruction for second-order preserving proximity models (e.g., link prediction), such as *Ou et al. (2016)* and *Wang, Cui & Zhu (2016)*. They also suit for various graph optimization task providing useful tools for constructing graph-based heuristics (*Khalil et al., 2017*).

APPLICATIONS TO REAL-WORLD PROBLEMS

In this section, we are interested in how graph embeddings appear in many other computer science fields, in which graphs are not directly expressed in the data, but relations between the objects can be efficiently described by graphs, and so, graph embeddings.

Computer vision

Image classification can be solved with classic CNN models considering the images as a grid-like structure. Recently, graph convolutional network models can take into account different neighboring relations, thus going beyond the nearest pixels as the only features for convolutions. Especially interesting results were obtained for 3D shape reconstruction (*Monti et al., 2017*) and video action recognition.

There are four main ideas of using graph neural networks for computer vision tasks: working with the interaction of objects on video and images, feature similarity graph, label graph, that is, images with the same label are connected, and internal graph-structured image data.

One of the main problems with CNN is that they should be deep enough to account interaction information between object, so *Chen et al. (2019c)* propose GloRe unit that applies GCNs over interaction data. It helps to efficiently solve relational reasoning task. In

Wang et al. (2018) relation graph of image objects was built for localizing object instance from natural language expression. Graph representation is also useful for representing in-label object interaction like in metric learning. It successfully applied to face clustering task (*Yang et al., 2019*; *Wang et al., 2019b*). Also such graph was exploited by *Kim et al. (2019)* for few-shot learning classification. Graph Convolutions are widely used in skeleton-based action recognition. It applies different graph network models to human skeleton graph (*Shi et al., 2019*; *Si et al., 2019*; *Li et al., 2019a*). GNNs are used for video tracking and classification tasks (*Zhang et al., 2018a*; *Gao, Zhang & Xu, 2019*; *Zhong et al., 2019*).

Natural language processing

NLP is highly correlated to graph tasks. Here similar sequential methods are used, while data have hierarchical structure from different views.

In *Marcheggiani & Titov (2017)*, authors assign semantic roles by encoding sentences with the graph convolutional network. In *Marcheggiani, Bastings & Titov (2018)*, *Zhao et al. (2019)* graph convolutional network models were applied for machine translation. *Sevgili, Panchenko & Biemann (2019)* use the Wikipedia link graph between entities to improve the quality of entity disambiguation task on unstructured text data. Graph models are widely used in NLP to extract syntactic and semantic information (*Luo et al., 2019*; *Vashishth et al., 2019*; *Veyseh, Nguyen & Dou, 2019*). The main approach is to extract the dependency graph and learn node (word) embeddings using GCN. Another approach is to examine each sentence as a complete graph with adjacency weighted by attention.

Graph neural networks also help in sequence tagging task, because it natively exploits information about the connection between different entities. *Zhu et al. (2019)* propose the Generated Parameters GNN for the Relation extraction task. It also builds a complete graph of entities in the sentence via encoding of the sentence with any sequence model. After that, GNN is applied to solve the node classification task. A prominent application of GNNs is to encode dependency tree information. Such an approach is exploited by *Guo, Zhang & Lu (2019)*, they apply Graph Attention Models. *Sahu et al. (2019)* also use dependency graph for relation extraction tasks, but their model accounts for inter-sentence dependencies.

Question answering, comment generation and dialog systems are highly dependent on domain knowledge-base. Such knowledge-base usually can be depicted as knowledge graphs. *Banerjee & Khapra (2019)*, *Kim, Kim & Kwak (2018)* applies GNN to encode knowledge and account to it in these tasks. *Li et al. (2019b)* also use graph models based on news interaction graphs.

The transformer-based language models (*Vaswani et al., 2017*) works in a similar way to graph attention networks. It models a sentence as a complete graph and calculates new word representation weighting previous vectors with self-attention. The BERT model (*Devlin et al., 2018*) is a special case of transformer-based models. It learns the vector by predicting masked words. Such tasks can be formulated as link prediction between context and masked words.

Knowledge graph completion

Knowledge graph embedding aims to learn vectors for entities and multi-dimensional vectors for entity relations. Knowledge graph completion solves link prediction between entities in knowledge graphs thus predicting ordered triples of entity-relation-entity (*Lin et al., 2015*). Knowledge graph (KG) embedding presents a knowledge base as a collection of triples "head-relation-tail" and consider them training samples. Structured Embedding (*Bordes et al., 2011*) learns two separate entity-relation representations for head and tail, while Semantic Matching Energy (*Bordes et al., 2012*), Latent Factor Model (*Jenatton et al., 2012*) and Neural Tensor Network (*Socher et al., 2013*) embed entities and relations, and use models to capture correlations between them.

A survey on KG embeddings *Wang et al. (2017a)* considers translation-based models, such as TransE (*Bordes et al., 2013*), TransH (*Wang et al., 2014*), TransM (*Fan et al., 2014*), TransR/CTransR (*Lin et al., 2015*), TransC (*Lv et al., 2018*), TransD (*Ji et al., 2015*), TranSparse (*Ji et al., 2016*), KG2E (*He et al., 2015*), and semantic matching models, based on RESCAL (*Nickel, Tresp & Kriegel, 2011*) tensor factorization framework, such as DistMult (*Yang et al., 2014*), HolE (*Nickel, Rosasco & Poggio, 2015*) and ComplEx (*Trouillon et al., 2017*) with comparison paper for the latter two in *Trouillon & Nickel (2017)*.

Question answering via knowledge graph embeddings was suggested in *Huang et al. (2019)*. Weighted attention for supporting triple in KG link prediction problem was presented in *Mai, Janowicz & Yan (2018)*.

Data mining

Ye et al. (2019) proposed method that models relations between different entities in Android logs (API, apps, device, signature, affiliation) using a hierarchical graph. Then they classify nodes of such graphs for real-time malware classification. Graph neural networks are widely used to utilize the social network information. *Wu et al. (2019a)*, *Song et al. (2019)*, *Chen et al. (2019a)* use such models to account for social effects in recommender systems. *Zhang, Ren & Urtasun (2019)* propose Graph HyperNetworks for neural architecture search. It learns topology of architecture and infers weights for it.

Recommender systems

The basic approach for recommending top K nodes of interest for a given node is usually based on certain similarity metric (*Pirotte et al., 2007*; *Zhao et al., 2013*; *Gui et al., 2016*; *Zhou et al., 2017*; *Ou et al., 2016*). There are various situations in which one need to provide node recommender system *Zhang & Wang (2016)*, in particular, for items to customers via APP model (*Zhou et al., 2017*), documents matching a given query (*Xiong, Power & Callan, 2017*), community-based question answering (*Zhao et al., 2016*; *Fang et al., 2016*), music recommendations via user preference embedding for query answering (*Chen et al., 2015a*, *2016a*), location recommendations (*Xie et al., 2016*), and many other real-world scenarios.

Matrix completion approach based on graph embeddings was provided in *Monti, Bronstein & Bresson (2017)*. Large scale recommender system was presented in *Ying et al. (2018a)*. Explainable recommendations were studied in *Zhang & Chen (2018)*.

In *Zhang, Wang & Zhang (2019d)* authors represents product search as a graph of co-clicked answers. They mix network embedding, term item vectors and term query vector using MLP to predict the probability of click on the item in certain query. This score is used to rank products.

STAR-GCN (*Zhang et al., 2019c*) is used over user-item interaction graph to learn user and item vectors. This approach is also suitable for inductive learning only using several interactions of users and items. This helps to solve the cold-start problem in recommender systems. *Shang et al. (2019)* use graphs for encoding hierarchical structure of health diseases. Next, achieved embeddings are integrated into BERT model for visit-based user recommendation.

The classical specific case of using network science in recommendations is the link prediction in collaborator networks (*Chen, Li & Huang, 2005*; *Liu & Kou, 2007*; *Li & Chen, 2009*; *Cho & Yu, 2018*). *Kong et al. (2018)* developed a scientific paper recommender system based on citation networks, which uses text information embeddings to find papers of similar research interest and structural network embedding. The combined embedding model was then applied for constructing article vector representations. A combination of network and knowledge graphs was proposed in *Yang, Tang & Cohen (2016b)*. In *Makarov et al. (2019a, 2019b, 2019c)* authors show that two-level architecture can improve the recommendation results. Firstly it predicts the collaboration itself and further estimates its quantity/quality. A survey on co-authorship and citation recommender systems may be found in *Ortega et al. (2018b)*.

Biomedical data science

The large variety of data in biomedicine can be represented as networks. *Le, Yapp & Yeh (2019)* applies embedding techniques to electron transport chains. *Do, Le & Le (2020)* utilizes it for detection of specific proteins. *Lin et al. (2020b)* exploits the dynamic graph embedding for detecting changes in functional connectivity in the brain network.

Computational drug design is an attractive direction because it reduces the costs of development of new drugs. The prominent field is drug repositioning. It usually works with networks of drug interaction with other entities: target, disease, gene or another drug. The main idea of such task is to predict possible relations between drug and other entities *Su et al. (2020)*. For example, drug-disease interaction networks can predict the possible treatment of new disease with existing drugs. So, it is a similar statement to the link prediction problem. *Yamanishi et al. (2008)*, *Cobanoglu et al. (2013)*, *Ezzat et al. (2017)* find drug-target pairs via proximity over matrix factorization based embeddings. *Zheng et al. (2013)*, *Yamanishi et al. (2014)*, *Ezzat et al. (2016)* try to add external data to the drug-interaction network embeddings. *Luo et al. (2017)*, *Zong et al. (2017)*, *Alshahrani et al. (2017)* build heterogeneous networks of different drug-related interaction and apply network embedding methods to it. *Wang et al. (2019a)* embeds heterogeneous gene graph to predict drug response.

Another important field in medicine design is the adverse drug reaction (ADR) analysis. Some articles (*Zitnik & Zupan, 2016*; *Zitnik, Agrawal & Leskovec, 2018*) focus on similar drug–drug and drug–target interaction prediction. *Wang (2017)*, *Abdelaziz et al. (2017)*

use the knowledge graph based on biomedical texts. *Stanovsky, Gruhl & Mendes (2017)* also works with KG embedding, but over ADRs mentions in social media.

Network science is also applied to the molecule structure. *Li et al. (2017b)* proposes a prediction of pathogenic human genes using network embedding. Network embedding is very popular method in protein–protein interaction assessment and function prediction (*Kulmanov, Khan & Hoehndorf, 2018*; *Su et al., 2020*; *Wang, Qu & Peng, 2017b*). *Shen et al. (2017)* and *Li et al. (2017a)* applies to miRNA-disease interaction network to associate genes with complex diseases. The detailed survey of biomedical network embedding applications is presented by *Su et al. (2020)*.

Reinforcement learning

Reinforcement learning (RL) is a popular approach to solve combinatorial optimization problems. *Zheng, Wang & Song (2020)* provides the open-sourced environment for graph optimization problems using reinforcement learning and graph embeddings. *Hayashi & Ohsaki (2020)* use RL for a similar task, such as binary topology optimization of trusses. It utilizes graph convolution networks for feature extraction and further usage in RL optimization. A similar concept was used in *Yan et al. (2020)* to solve automatic embedding problem using actor-critic models for optimization and graph embeddings for representation learning.

Waradpande, Kudenko & Khosla (2020) suggests encoding states in Markov's decision process with graph embedding models. *Lin, Ghaddar & Nathwani (2020a)* follows this idea and utilizes GNN for parametrization of the stochastic policy in electric vehicle routing problem. *Zhou et al. (2020)* solves the interactive recommender system problem enhancing it with knowledge graphs. It describes states using GCN over knowledge graph.

OPEN PROBLEMS

Here we mention the most interesting open problems in graph representation theory, which are far from good results applicable for any given scenarios.

Many real-world graphs are dynamic: nodes and edges can appear and vanish over time. Despite a large number of recent papers, this field is far from benchmark well-performing models as of now. One of the approaches for it is inductive learning, which is strongly correlated with graph dynamics problem. Inductive methods allow finding embedding for newly added nodes without refitting the whole model. It is important in real-world applications and partially solve the scalability issue.

Edge attributes aware network embedding is poorly studied field. There is a low number of models. Such models usually depend on a Line graph, which has a dramatically larger number of nodes. So such models have a problem with scalability. Edge attributes are important in such tasks as context-aware recommender systems or transportation networks optimization.

They are an only little number of works about subgraph embedding. Such models should represent complex structures like triangles or hierarchy. The application of non-euclidean spaces to the embedding task is a promising method solving this issue, but also poorly studied.

Recent advances in the distributed and batch training for graph neural networks looks promising. However, most of the methods are not theoretically grounded, so it could be hard to understand the issues of poor quality of results. Only *Zeng et al. (2019)* provides some bias-variance analysis of node and edge sampling approaches. However, *Akyildiz, Aljundi & Kaya (2020)* provides a much faster and powerful method for large scale embedding.

Another field that is currently under the control of many papers is the heterogeneous graph embedding. Such graphs are very common in real-world scenarios. The graph attention-based methods look promising in that field. It allows us to catch different aggregation levels like in *Fu et al. (2020)* and *Li et al. (2020b)*.

As can be seen from our survey, most embedding models catch specific graph attributes and there is no general model, thus, raising a problem of selection and recommendation of different models for specific use-cases.

It is also an interesting point to develop meta-strategies for embedding mixture, that will preserve different graph properties. Such meta-models could solve the problem of knowledge generalization and reduce costs for deploy of application.

As in the other fields like NLP and CV, graph neural networks are poorly interpretable, apart from an initial study in *Ying et al. (2019)*.

These and many other research questions lead to a vast amount of open research directions, which will benefit the field and lead to many applications in other computer science domains.

In our study, we focus on another interesting question regarding the fact that there are almost no general studies that compare the performance of models based on graph properties, most of the models are created for specific graph use-case. Below, we provide our insights on real-world networks as well as interpretations on such findings.

MODEL COMPARISON

This paper focuses on the four most popular tasks on graphs: node classification, link prediction, node clustering and network visualization. These tasks cover most of the real-world applications, in which a graph is used to unify information on nodes and their properties.

Data

We use four benchmark datasets for comparison of different models: CORA (*Sen et al., 2008*), Citeseer (*Lim & Buntine, 2016*), HSE coauthor network (*Makarov et al., 2018*), and Microsof Academic Graph Computer Science (MAG CS) (*Sinha et al., 2015*).

First two datasets are citation networks. This type of networks is very common for evaluating the quality of network embeddings. Also, these datasets are convenient for comparison of models, because they have interesting label and feature structure. The last dataset is a co-authorship network. It has heterogeneous edges and large size. General puspose graph embedding models work only with homogeneous graphs, so we merge all the edges between nodes in one edge. A brief overview of the datasets statistics is provided in Table 2.

Table 2 Datasets description.

	Assortativity	Label modularity	#Nodes	#Edges	#Features	#Classes
CORA	0.7711	0.8061	2708	10,556	1,433	7
CITESEER	0.6754	0.8872	3327	9,228	3,703	6
HSE	–	–	4181	12,004	0	0
MAG CS	0.7845	0.6989	18333	16,3788	6,805	15

Metrics

We use standard classification metrics for node classification and link prediction tasks.

- Accuracy is the rate of the right answer of a classifier.
- Precision is the rate of true positive answers relative to the number of all positive answers of a classifier.
- Recall is the rate of true positive answers relative to the number of all positive examples in the data.
- F1 is a harmonic mean of precision and recall.
- Area Under ROC-curve shows the probability that a random negative example sampled from a uniform distribution is ranked lower than randomly sampled positive.
- Average precision is the average of all possible precision values weighted by the recall for different probability thresholds.

We calculate the standard deviation with the following procedure:

1. Generate subsample of data with 90% volume of a given dataset.
2. Train model on it
3. Estimate quality of the trained model on the test set.
4. Repeat previous steps nine more times.

Described bootstrap (*Efron (1992)*) procedure allows to easily calculate standard error and confidence intervals for any statistics. Confidence intervals are required to understand the significance of the difference between models.

Node clustering was evaluated with two metrics: silhouette coefficient and modularity.

- Silhouette score shows the average similarity between each example and its cluster in comparison with the closest another cluster, so it measures the overall cluster separation relative to the distance measure. In our study, we use the Euclidean distance.
- Modularity score works pretty similarly but for computing inter- and intra-cluster quality, it measures the density of connections between clusters, respectively to its density inside clusters.

We also evaluate the quality of node clustering and network visualization with a visual comparison of how clusters are grouped in the UMAP (*McInnes, Healy & Melville, 2018*) projection of embeddings. UMAP (Uniform Manifold Approximation and Projection)

is the dimensionality reduction technique based on topology and Riemannian geometry. Firstly, it builds the weighted nearest neighbors graph according to elements feature vectors (embeddings). Then, it initializes layout using spectral embedding and optimizes it using SGD minimizing fuzzy-set cross-entropy.

The UMAP works much faster then TSNE and gives at least the same quality of the projection. Interpretation of received plot is simple: similar samples in initial space (for e.g., nodes with the same labels) should lie closely in the 2D plane.

Evaluation pipeline

The node classification task is native multi-class classification. Link prediction task can be also solved as classification, but with two classes depicting edge existence. The basic approach on validating such methods is to use delayed sample. So, before any model was trained, we create a train-test split for all the datasets, in order to compare all the models on similar subsets. We use simple 50% test, 50% train random split for node classification, following other papers on graph embeddings.

The problem with link prediction is that described graphs are high-imbalanced because there are much more unconnected node pairs. Large imbalance leads to poor training because even simple constant prediction will give high-scores. One of the methods for working with this problem is to the under-sample larger class. To keep the classification task harder, it is convenient to use a negative sampling technique. We select the most similar pairs of nodes which are not connected in the same amount as existent edges. The used proximity metric is cosine similarity, which is a normalized dot product of feature vectors. For features, we use the adjacency matrix.

Because basic classification models do not work with discrete graph data, after developing train and test samples, we need to generate meaningful features. Here we use the unsupervised graph embeddings (128 dimensions as commonly used in different papers and surveys). Graph neural networks were also trained in an unsupervised way with reconstruction loss over the graph adjacency matrix. Reconstruction loss calculates with binary cross-entropy between adjacency matrix and its estimation achieved by inner-product decoding from the embedding matrix.

Now, we can solve downstream tasks like classification or clustering. For that part, we use three different classifiers: logistic regression (LR), random forest (RF) and gradient boosting (GBM). Logistic regression is a linear model: it calculates the weighted average of object features and normalizes it with sigmoid to receive probability. Linear models are fast, interpretable and easily tunable because of their simplicity. Random forest is the ensemble of decision trees built on bootstrapped subsamples in both dimensions features and observations. Gradient boosted machines are another approach to learn decision tree ensemble. It determines each next tree by sequential optimization of the previous learner error-term. The main advantage of the tree-based models is that they could recover non-linear dependencies. But for this flexibility, we pay with a strong dependance on hyperparameter selection. Scikit-learn implementation with default hyperparameters was used. In the link prediction, we simply concatenate node vectors to achieve edge embedding. For the clustering, we use the K-means model from Scikit-learn.

Remark. The common way to use graph neural networks is semi-supervised training. Such an approach gives a bias towards the usage of that model, because embedding learns not only graph structure and feature transformations, but supervised information about labels on the other hand. So we train graph neural networks in an unsupervised way because our study is aimed to understand how different properties of embedding models can help in considered tasks.

Models

We select several models of different types mentioned in "Methods for Constructing Graph Embedding" that preserve different properties of a graph. The general idea is to compare models of different fitting approaches with respect to network properties.

Matrix factorization based models:

- GraRep is symmetric and preserves high-proximity. The default K-hop order is 5, the number of SVD iterations is 20, a random seed is 42.
- HOPE directly models asymmetric similarities.
- M-NMF preserves community structure. The default number of clusters is 20, clustering penalty is 0.05, modularity regularization penalty is 0.05, similarity mixing parameter is 5, the number of power-iterations is 200, early stopping step is 3.

Random-walks based models:

- Node2vec is a baseline for sequential methods which efficiently trade-offs between different proximity levels. The default walk length is 80, the number of walks per node is 10, return hyper-parameter is 1, in-out hyper-parameter is 1.
- Diff2vec use diffusion to sample random walks. The default number of nodes per diffusion tree is 80, the number of diffusions per source node is 10, context-size is 10, the number of ASGD iterations is 1, the learning rate is 0.025.
- Walklets allow to model different levels of community structure and generalize GraRep model. Default random walk length is 80, the number of random walks per source node is 5, the window size is 5, the minimal number of appearances is 1, the order of random walk is first, return hyper-parameter is 1, in-out hyper-parameter is 1.
- GEMSEC directly cluster nodes. Default random walk length is 80, the number of random walks per source node is 5, the window size is 5, the minimal number of appearances is 1, the order of random walk is first, return hyper-parameter is 1, in-out hyper-parameter is 1, distortion is 0.75, negative samples number is 10, the initial learning rate is 0.001, annealing factor for learning rate is 1, initial clustering weight coefficient is 0.1, final clustering weight coefficient is 0.5, smoothness regularization penalty is 0.0625, the number of clusters is 20, normalized overlap weight regularization.

Deep learning models:

- GCN is a baseline for deep learning models. The default number of epochs is 200, dropout is 0.3, the learning rate is 0.01, weight decay is 0.0005, the number of hidden layers is 1.

- GraphSage (GS) improves GCN by reducing the number of neighbors while weighting the node vectors. The dropout is 0.1, aggregation is GCN, the number of epochs is 200, the learning rate is 0.01, weight decay is 0.0005.
- GAT utilizes an attention mechanism. The number of epochs is 200, in-dropout is 0.1, attention dropout is 0.1, the learning rate is 0.005, the negative slope is 0.2, weight decay is 0.0005, the number of hidden layers is 1.

RESULTS

The current section has the following structure. We start the analysis from node clustering tasks because it also helps to understand the performance of graph embeddings on the other tasks. Further, we describe node classification task and link prediction followed by network visualization. We also conducted experiments on random graphs to study the difference of graph embeddings on real-world networks and simulated ones.

Node clustering

The results on the node clustering task are presented in Table 3. Rows depict different models, which are grouped by model type: matrix factorization, random walks, graph neural networks with and without features. On the columns, we can see results on different datasets. For each dataset, we calculate two metrics: modularity and silhouette score. Highlighted results are the best.

In node clustering task, results are pretty obvious: the embeddings, which work with community structure, perform the best in terms of modularity. GEMSEC directly penalizes embeddings for low modularity score with K-Means objective, Walklets catches this information by accounting for several levels of node neighborhood. Importance of such information could be proven by the comparatively high value of GraRep model, that works pretty similar to Walklets.

Graph neural networks with features give comparatively better results, meaning that node content helps to describe graph structure. GraphSAGE and GAT efficiently utilize the local structure of the network. The main difference is that GAT aggregates over the entire neighborhood, but GraphSAGE aggregates only over a fix-sized sample.

In the case of MAG CS graph (Table 4) the best results show GAT and GCN. It means that in the case of large, heterogeneous graph features play a major role. Interesting, that GAT without features works much better than other structural models. It could refer to the attention, that selects only the most important neighbors in node embedding construction. It seems that the attention mechanism helps in this case to distinguish heterogeneous edge nature.

GNN models trained in unsupervised fashion give poor results because they highly rely on the features when constructing embeddings even for learning graph structure.

The clustering results show that specific losses can dramatically increase quality on a specific task. As we will see further, such losses are also helpful in the node classification task preserving important graph properties.

Table 3 Results of model validation on node clustering task (both metrics lie between (−1,1) and higher value means better results). Bold corresponds to the best metric for each dataset.

	CORA		CITESEER		HSE	
	Modularity	Silhouette	Modularity	Silhouette	Modularity	Silhouette
GRAREP (*Cao, Lu & Xu, 2015*)	0.2249	0.1902	0.0320	0.3159	0.2320	0.3163
HOPE (*Ou et al., 2016*)	0.1222	0.2593	0.1748	0.5492	0.0027	0.6684
NODE2VEC (*Grover & Leskovec, 2016*)	0.0106	0.1000	−0.0018	0.0464	0.0419	0.5576
DIFF2VEC (*Rozemberczki & Sarkar, 2018*)	0.1289	0.5412	0.0292	0.5422	0.1155	0.5429
GEMSEC (*Rozemberczki et al., 2018*)	**0.7684**	0.2280	**0.7555**	0.1508	**0.7710**	0.1678
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.7353	0.0812	0.7263	0.0566	0.7593	0.0667
GCN *Kipf & Welling (2016a)*	0.3800	0.3336	0.3754	0.4215	–	–
GRAPHSAGE (*Hamilton, Ying & Leskovec, 2017a*)	0.6455	0.3311	0.5774	0.4757	–	–
GAT (*Veličković et al., 2017*)	0.7209	0.3477	0.7367	0.3797	–	–
GCN (NF)	−0.0096	0.3979	0.0360	0.4999	0.0008	0.6837
GRAPHSAGE (NF)	0.0212	**0.7672**	0.0960	**0.9442**	0.0552	**0.8381**
GAT (NF)	0.1335	0.2001	0.2968	0.3641	0.1400	0.6390

Table 4 Results of model validation on node clustering task for MAG-CS dataset (both metrics lie between (−1,1) and higher value means better results).

	HOPE	NODE2VEC	GRAREP	WALKLETS	GCN	GAT	GCN (NF)	GAT (NF)
Modularity	−0.0001	−0.0037	0.0027	0.0025	0.3462	0.3446	0.0112	0.1951
Silhouette	0.6548	0.0771	0.2348	0.0441	0.2369	−0.0261	0.4654	0.0411

Node classification

The results on node classification task are presented in Table 5. Rows show different types of models and columns show different datasets. For each dataset, we calculate accuracy for three different models: gradient boosted machines, logistic regression and random forest. Highlighted results are the best.

Models that have good performance in node clustering task also show high score in node classification. Labels in given datasets show different topics of articles, as soon as usually authors are dedicated to specific topics, so natural communities are constructed within these labels. This can also be proven by high modularity and assortativity coefficients of label communities for all graphs. In the classification task, it is also important to have a good separation of clusters, that could be measured by the silhouette coefficient. We can see those models that keep both high modularity and high silhouette work better.

Linear models show the comparatively lower score, but for random walk based embeddings, this difference is much less severe. Most of considered random walk models are based on Skip-Gram approach, which is a log-linear model. It reduces expression quality of the model but allows to learn vectors that perform well in linear models.

Results for MAG CS are presented in Table 6. Firstly, we compare fewer models, because we were not able to compute some embeddings for such a large graph, so we choose the

Makarov et al. (2021), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.357

30/62

Table 5 Results of model validation on node classification task (accuracy metric lies between (0,1) and higher value means better results). Bold corresponds to the best metric for each dataset.

	GBM	LR	RF
CORA			
GRAREP (*Cao, Lu & Xu, 2015*)	0.7610 ± 0.0434	0.7503 ± 0.0323	0.7751 ± 0.0254
HOPE (*Ou et al., 2016*)	0.7518 ± 0.0333	0.3024 ± 0.0308	0.7614 ± 0.0289
M-NMF (*Wang et al., 2017d*)	0.2596 ± 0.0250	0.2799 ± 0.0324	0.2633 ± 0.0239
NODE2VEC (*Grover & Leskovec, 2016*)	0.2522 ± 0.0200	0.2441 ± 0.0273	0.2441 ± 0.0257
DIFF2VEC (*Rozemberczki & Sarkar, 2018*)	0.2212 ± 0.0635	0.2843 ± 0.0387	0.2500 ± 0.0293
GEMSEC (*Rozemberczki et al., 2018*)	0.8338 ± 0.0326	0.8153 ± 0.0390	**0.8634 ± 0.0251**
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.8142 ± 0.0252	0.8124 ± 0.0317	0.8327 ± 0.0326
GCN (*Kipf & Welling, 2016a*)	0.7803 ± 0.0378	0.6588 ± 0.0448	0.7718 ± 0.0380
GRAPHSAGE (*Hamilton, Ying & Leskovec, 2017a*)	0.8083 ± 0.0358	0.7385 ± 0.0391	0.8168 ± 0.0316
GAT (*Veličković et al., 2017*)	0.8194 ± 0.0304	0.7455 ± 0.0420	0.8264 ± 0.0324
GCN (NF)	0.3021 ± 0.0204	0.2969 ± 0.0238	0.2888 ± 0.0194
GRAPHSAGE (NF)	0.3017 ± 0.0298	0.3021 ± 0.0305	0.3017 ± 0.0298
GAT (NF)	0.3021 ± 0.0305	0.3021 ± 0.0305	0.3021 ± 0.0305
CITESEER			
GRAREP (*Cao, Lu & Xu, 2015*)	0.5582 ± 0.0577	0.5110 ± 0.0443	0.5834 ± 0.0453
HOPE (*Ou et al., 2016*)	0.5468 ± 0.0346	0.2663 ± 0.0443	0.5489 ± 0.0378
M-NMF (*Wang et al., 2017d*)	0.1767 ± 0.0220	0.1978 ± 0.0241	0.1909 ± 0.0311
NODE2VEC (*Grover & Leskovec, 2016*)	0.1815 ± 0.0253	0.1806 ± 0.0165	0.1867 ± 0.0237
DIFF2VEC (*Rozemberczki & Sarkar, 2018*)	0.2035 ± 0.0373	0.2239 ± 0.0281	0.1930 ± 0.0287
GEMSEC (*Rozemberczki et al., 2018*)	0.6754 ± 0.0343	0.5867 ± 0.0427	**0.7175 ± 0.0247**
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.6291 ± 0.0280	0.6243 ± 0.0228	0.6480 ± 0.0277
GCN (*Kipf & Welling, 2016a*)	0.6312 ± 0.0210	0.5092 ± 0.0272	0.6342 ± 0.0209
GRAPHSAGE (*Hamilton, Ying & Leskovec, 2017a*)	0.6450 ± 0.0228	0.5425 ± 0.0192	0.6586 ± 0.0309
GAT (*Veličković et al., 2017*)	0.6733 ± 0.0238	0.5582 ± 0.0443	0.6763 ± 0.0220
GCN (NF)	0.2729 ± 0.0272	0.2317 ± 0.0336	0.2792 ± 0.0260
GRAPHSAGE (NF)	0.1996 ± 0.0409	0.1996 ± 0.0409	0.1996 ± 0.0409
GAT (NF)	0.1996 ± 0.0409	0.1996 ± 0.0409	0.1996 ± 0.0409

fastest ones with good performance in other experiments. HOPE outperforms other classic non-attributed embeddings. It is also interesting that the linear mixture of embedding elements works much better for this embedding then others. One of the reasons for such behavior is that in the large graphs, embeddings could be noisy and simple models could give better quality. It could be also a reason for the worse performance of K-hop based embeddings (Walklets and GraRep). But the main driver in node classification quality is the features of nodes. It could be seen from the high results of GCN and GAT models.

HOPE has a dramatic difference between linear and non-linear models because it estimates Katz centrality, which has non-linear nature. Also, we use HOPE implementation from GEM *Goyal & Ferrara (2017)*, where node embedding is achieved as a concatenation of its self- and context-representations. The non-linear model helps to reconstruct the dot

Table 6 Results of model validation on node classification task for MAG-CS dataset (accuracy metric lies between (0,1) and higher value means better results). Bold corresponds to the best metric for each dataset.

	GBM	LR	RF
GRAREP (*Cao, Lu & Xu, 2015*)	0.1915 ± 0.0162	0.1404 ± 0.0217	0.1737 ± 0.0169
HOPE (*Ou et al., 2016*)	0.1985 ± 0.0233	0.2255 ± 0.021	0.1665 ± 0.0184
NODE2VEC (*Grover & Leskovec, 2016*)	0.1882 ± 0.034	0.2048 ± 0.0332	0.168 ± 0.0195
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.1866 ± 0.0171	0.1527 ± 0.0084	0.1886 ± 0.0189
GCN (*Kipf & Welling, 2016a*)	0.752 ± 0.0177	0.7317 ± 0.0166	**0.7568 ± 0.0176**
GAT (*Veličković et al., 2017*)	0.6272 ± 0.0188	0.6424 ± 0.0202	0.6095 ± 0.0208
GCN (NF)	0.2089 ± 0.0154	0.2255 ± 0.021	0.1738 ± 0.0117
GAT (NF)	0.2255 ± 0.021	0.2255 ± 0.021	0.2255 ± 0.021

product decoder. A similar argument can explain diversity in neural network models, but it has less variance because of high clustering efficiency.

Link prediction

Table 7 shows results for link prediction task. It is separated into three groups by datasets. Rows represent different graph embedding models and columns show different second-level classification models: gradient boosted machines, logistic regression and random forest. Highlighted results are the best.

In the link prediction task, we can also see the importance of clustering. Links are more likely to occur within one community. The high-proximity models also work much better, because in that task we need to understand how similar are non-connected nodes.

The performance of the HOPE model in this task is more significant. HOPE model concentrates on preserving asymmetric transitivity. The older paper can not cite the newer one.

GCN without features performs much better than other graph neural networks. It accounts for the whole neighborhood and directly uses the adjacency matrix to train embeddings.

Results for MAG CS (Table 8) are consistent with these findings. However, despite the good quality on the community clustering task, GAT without features shows pure performance on the Link prediction task. However, GCN without features is close to the GAT with features. It means that in this task it is necessary to account the whole neighborhood.

A dramatic difference in the quality of linear and non-linear models can be explained by the objective of the link prediction task. It requires to model the proximity between to nodes. Such metrics are non-linear. So for reconstructing it from concatenated vectors of nodes, we need some non-linear transformations.

Graph visualization

We present results of node clustering jointly with network visualization using UMAP technique. The results for three different datasets are shown at Fig. 1 for Cora, Fig. 2 for Citeseer, and Fig. 3 for HSE datasets, respectively.

Table 7 Results of model validation on link prediction task (accuracy metric lies between (0,1) and higher value means better results). Bold corresponds to the best metric for each dataset.

	GBM	LR	RF
CORA			
GRAREP (*Cao, Lu & Xu, 2015*)	0.8766 ± 0.0056	0.7585 ± 0.0037	0.9143 ± 0.0021
HOPE (*Ou et al., 2016*)	0.9422 ± 0.0039	0.6706 ± 0.0032	0.9478 ± 0.0020
M-NMF (*Wang et al., 2017d*)	0.6507 ± 0.0038	0.6252 ± 0.0018	0.6618 ± 0.0022
NODE2VEC (*Grover & Leskovec, 2016*)	0.7047 ± 0.0039	0.6185 ± 0.0037	0.7060 ± 0.0042
DIFF2VEC (*Rozemberczki & Sarkar, 2018*)	0.7780 ± 0.0049	0.7508 ± 0.0045	0.7413 ± 0.0029
GEMSEC (*Rozemberczki et al., 2018*)	**0.9692 ± 0.0030**	0.6512 ± 0.0052	0.9653 ± 0.0011
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.9153 ± 0.0058	0.7073 ± 0.0022	0.9574 ± 0.0017
GCN (*Kipf & Welling (2016a*)	0.8784 ± 0.0028	0.7094 ± 0.0041	0.8978 ± 0.0022
GRAPHSAGE (*Hamilton, Ying & Leskovec, 2017a*)	0.8988 ± 0.0050	0.5668 ± 0.0049	0.9111 ± 0.0028
GAT (*Veličković et al., 2017*)	0.9127 ± 0.0047	0.5666 ± 0.0063	0.9337 ± 0.0021
GCN (NF)	0.7852 ± 0.0060	0.7084 ± 0.0033	0.8014 ± 0.0024
GRAPHSAGE (NF)	0.5459 ± 0.0043	0.5033 ± 0.0021	0.5459 ± 0.0043
GAT (NF)	0.5033 ± 0.0021	0.5033 ± 0.0021	0.5033 ± 0.0021
CITESEER			
GRAREP (*Cao, Lu & Xu, 2015*)	0.8786 ± 0.0046	0.7198 ± 0.0049	0.9254 ± 0.0031
HOPE (*Ou et al., 2016*)	0.8985 ± 0.0074	0.6358 ± 0.0052	0.9119 ± 0.0029
M-NMF (*Wang et al., 2017d*)	0.5926 ± 0.0049	0.5685 ± 0.0033	0.6215 ± 0.0031
NODE2VEC (*Grover & Leskovec, 2016*)	0.6895 ± 0.0050	0.6315 ± 0.0056	0.6934 ± 0.0046
DIFF2VEC (*Rozemberczki & Sarkar, 2018*)	0.7553 ± 0.0038	0.7258 ± 0.0038	0.7206 ± 0.0060
GEMSEC (*Rozemberczki et al., 2018*)	**0.9827 ± 0.0031**	0.6151 ± 0.0096	0.9726 ± 0.0026
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.8688 ± 0.0066	0.6672 ± 0.0040	0.9429 ± 0.0024
GCN (*Kipf & Welling, 2016a*)	0.8863 ± 0.0033	0.6910 ± 0.0032	0.9052 ± 0.0024
GRAPHSAGE (*Hamilton, Ying & Leskovec, 2017a*)	0.8952 ± 0.0037	0.6082 ± 0.0036	0.8998 ± 0.0034
GAT (*Veličković et al., 2017*)	0.9175 ± 0.0030	0.6136 ± 0.0051	0.9306 ± 0.0025
GCN (NF)	0.7892 ± 0.0039	0.6881 ± 0.0044	0.8100 ± 0.0034
GRAPHSAGE (NF)	0.5181 ± 0.0039	0.5037 ± 0.0026	0.5181 ± 0.0039
GAT (NF)	0.5037 ± 0.0026	0.5037 ± 0.0026	0.5037 ± 0.0026
HSE			
GRAREP (*Cao, Lu & Xu, 2015*)	0.9202 ± 0.0068	0.7956 ± 0.0032	0.9332 ± 0.0022
HOPE (*Ou et al., 2016*)	0.6590 ± 0.0050	0.6062 ± 0.0055	0.7022 ± 0.0038
M-NMF (*Wang et al. (2017d*)	0.6824 ± 0.0058	0.6277 ± 0.0041	0.7467 ± 0.0032
NODE2VEC (*Grover & Leskovec, 2016*)	0.7257 ± 0.0049	0.6634 ± 0.0034	0.7592 ± 0.0039
DIFF2VEC (*Rozemberczki & Sarkar, 2018*)	0.7850 ± 0.0040	0.7505 ± 0.0037	0.7795 ± 0.0034
GEMSEC (*Rozemberczki et al., 2018*)	**0.9724 ± 0.0035**	0.7065 ± 0.0043	0.9671 ± 0.0013
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.9484 ± 0.0028	0.7730 ± 0.0035	0.9615 ± 0.0022
GCN (NF) (*Kipf & Welling, 2016a*)	0.8178 ± 0.0021	0.7867 ± 0.0031	0.8214 ± 0.0030
GRAPHSAGE (NF) (*Hamilton, Ying & Leskovec, 2017a*)	0.5071 ± 0.0026	0.5039 ± 0.0030	0.5071 ± 0.0026
GAT (NF) (*Veličković et al., 2017*)	0.5039 ± 0.0030	0.5039 ± 0.0030	0.5039 ± 0.0030

Makarov et al. (2021), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.357

33/62

Table 8 Results of model validation on link prediction task for MAG-CS dataset (accuracy metric lies between (0,1) and higher value means better results). Bold corresponds to the best metric for each dataset.

|	GBM	LR	RF
GRAREP (*Cao, Lu & Xu, 2015*)	0.5986 ± 0.0047	0.5626 ± 0.0016	0.5998 ± 0.0025
HOPE (*Ou et al., 2016*)	0.566 ± 0.0017	0.5275 ± 0.0025	0.6007 ± 0.0027
NODE2VEC (*Grover & Leskovec, 2016*)	0.578 ± 0.0023	0.5425 ± 0.0015	0.6137 ± 0.0031
WALKLETS (*Perozzi, Kulkarni & Skiena, 2016*)	0.5798 ± 0.0024	0.5647 ± 0.0015	0.6077 ± 0.0027
GCN (*Kipf & Welling, 2016a*)	0.8486 ± 0.0022	0.6553 ± 0.0014	**0.8772 ± 0.0012**
GAT (*Veličković et al., 2017*)	0.7293 ± 0.0041	0.5632 ± 0.0024	0.7524 ± 0.0027
GCN (NF)	0.7253 ± 0.0012	0.697 ± 0.0014	0.7261 ± 0.0017
GAT (NF)	0.5015 ± 0.0009	0.5015 ± 0.0009	0.5015 ± 0.0009
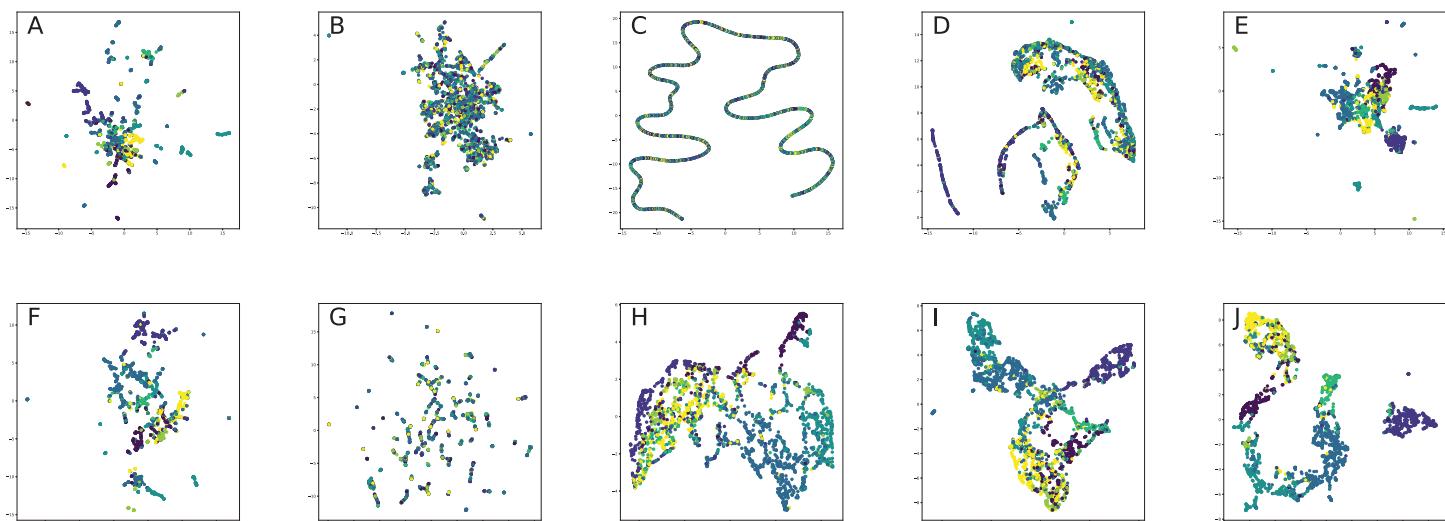
Figure 1 UMAP projection of CORA embeddings: (A) HOPE. (B) Node2Vec. (C) Diff2Vec. (D) GraRep. (E) Walklets. (F) GEMSEC. (G) M-NMF. (H) GCN. (I) GraphSage. (J) GAT. Full-size 🖼 DOI: 10.7717/peerj-cs.357/fig-1

The best visualization in terms of community structure seems to be Walklets and GraRep models, which give nicely distinguishable clusters in all the cases. Both models work in the same way with *k*-hop similarity of vertices. GEMSEC also provides separate cluster picture but creates a lot of noisy points.

Interestingly, that HOPE also split graphs into several parts, but we can see by the modularity score, such parts are not correlated with node communities. Such an effect could occur because HOPE embedding has non-homogeneous structure due to concatenation of self- and context-representations.

In the case of graph neural networks, except for GAT, all clusters have poor separation. Such effect occurs because GNN weights neighborhood node attributes, so boundary nodes will be close. GAT allows mitigating this problem because utilizes the attention mechanism, which weights meaningless node neighbors to zero.
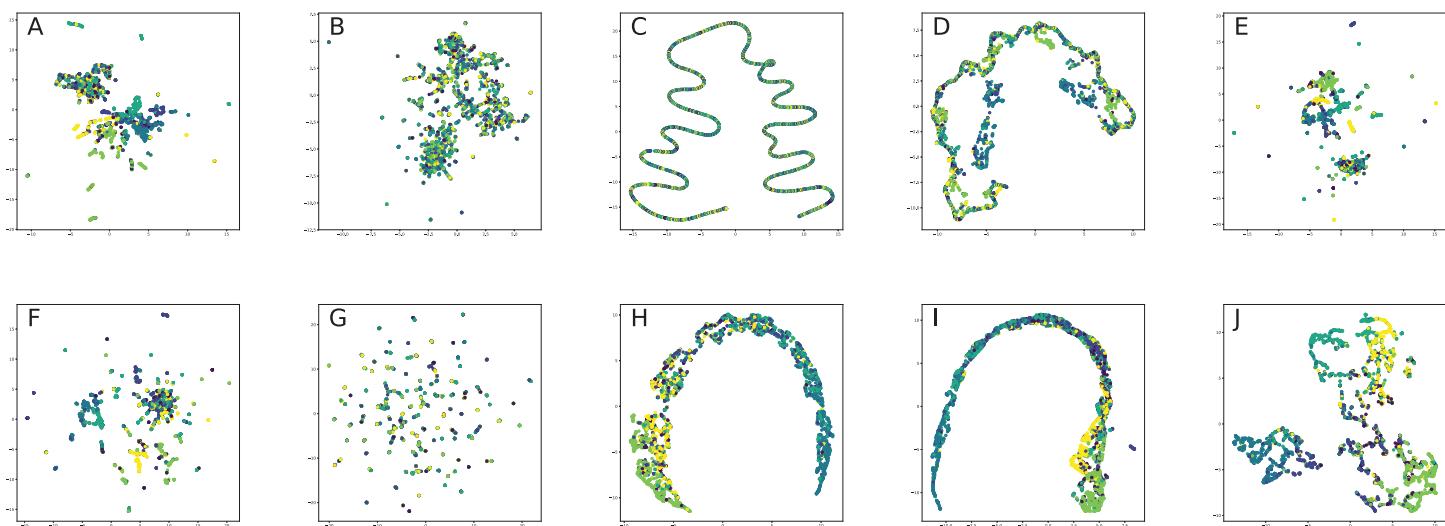
Figure 2 UMAP projection of Citeseer embeddings: (A) HOPE. (B) Node2Vec. (C) Diff2Vec. (D) GraRep. (E) Walklets. (F) GEMSEC. (G) M-NMF. (H) GCN. (I) GraphSage. (J) GAT. Full-size ⬓ DOI: 10.7717/peerj-cs.357/fig-2

It seems that one of the most important graph property in the studied tasks is the community structure. So, most of the methods, that works with it directly allow to achieve the best scores. It is also connected to the level of proximity because it is indirectly connected with the community structure. The graph neural networks allow to easily catch node attributes, but miss most of graph properties information, so it performs on the level of baseline models without it.

Random graphs

In order to understand robustness of graph embeddings, we decided to test how modeling real-world network with random graphs impact on the quality of graph embeddings for simulated networks.

Firstly, we should explain the formation of random graphs in different models. *Erdös & Rényi (1959)* builds the graphs using a simple binomial rule for creating an edge with a given density of graph. *Barabási & Albert (1999)* starts from a small graph and sequentially adds a new node with a given density and connects existing nodes using preferential attachment rule. In the *Watts & Strogatz (1998)* model, the regular lattice is firstly constructed followed by edge rewiring procedure.

We build random graphs regarding the properties of real-world graphs. To build ER graph one need to have a number of nodes and edges in the graph. For the BA graph construction, it is required to have a number of nodes and number of edges for the newly added node at each iteration. It is a small integer, so we just select the best to fit the number of edges of benchmarks. The parameters of WS graphs were chosen based on the number of nodes, edges and average clustering of graphs following formulae: the number of edges in starting lattice is equal to $k = [2 \# \text{edges} \# \text{nodes}]$, the rewriting probability is equal to $p = 1 - \sqrt[3]{4(k-1)/3(k-2) \cdot (\text{average clustering})}$ (*Barrat & Weigt, 2000*).
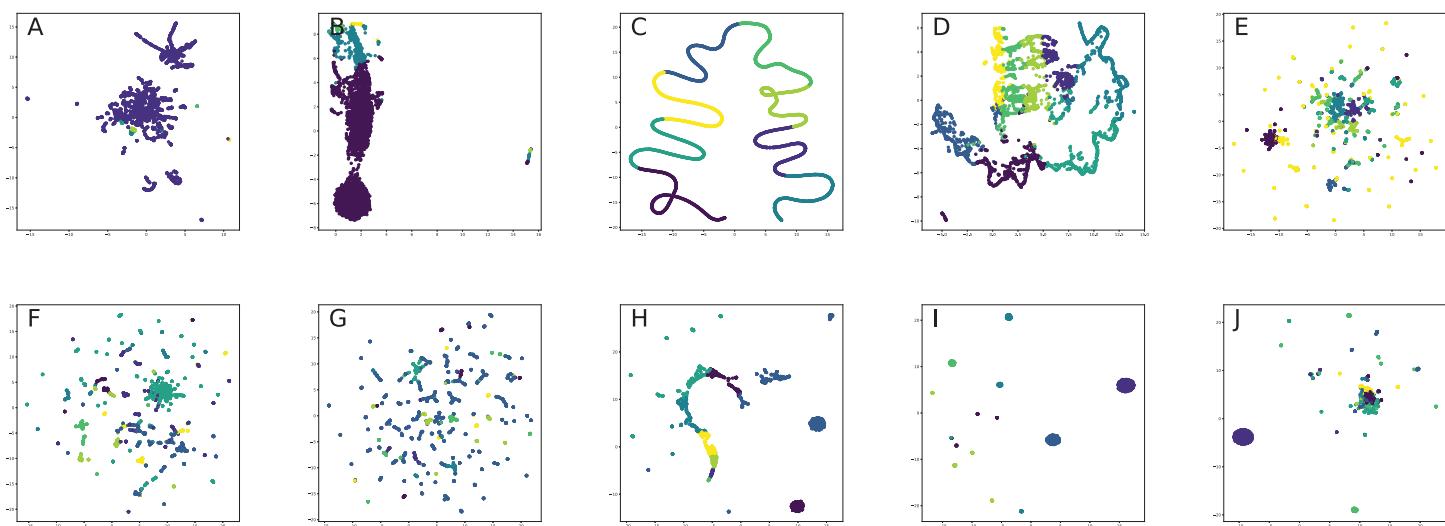
Figure 3 UMAP projection of HSE embeddings: (A) HOPE. (B) Node2Vec. (C) Diff2Vec. (D) GraRep. (E) Walklets. (F) GEMSEC. (G) M-NMF. (H) GCN. (I) GraphSage. (J) GAT.
Full-size ☒ DOI: 10.7717/peerj-cs.357/fig-3

Table 9 Results of model validation on node clustering task for random graphs (both metrics lie between (−1,1) and higher value means better results. Bold corresponds to the best metric for each dataset.

	HOPE		WALKLETS	
	Modularity	Silhouette	Modularity	Silhouette
CORA (Original)	0.1222	0.2593	0.7353	0.0812
CORA (Barabási-Albert)	0.0005	**0.1807**	0.1465	0.0046
CORA (Erdős-Rényi)	−0.0022	0.0216	0.0184	0.0080
CORA (Watts-Strogatz)	0.0629	0.1180	**0.5251**	0.0212
CITESEER (Original)	0.1748	0.5492	0.7263	0.0566
CITESEER (Barabási-Albert)	−0.0008	**0.3731**	0.0397	−0.0006
CITESEER (Erdős-Rényi)	0.0031	0.1495	−0.0040	0.0085
CITESEER (Watts-Strogatz)	0.0344	0.1270	**0.4941**	0.0155

One of the main properties of all random graph models is the giant connected component. So embedding models learn it on the train part and works better than random in link prediction task in all cases. Additionally, in the BA model, there are some nodes with much larger density than others, so it is easier to predict missed links. Watts–Strogatz also has a straightforward mechanism of edge construction, where the shortest path is small. In both BA and WS models it is also possible to reproduce community structure. We can see it by large modularity metric in the clustering task.

Random graph modeling is one of the efficient methods in network science for evaluation of different model properties. For example, comparison of real-graph with its random analog could help to understand how good is the received quality for the specific task. We follow this idea and compare two embeddings models Walklets and HOPE.

Table 10 Results of model validation on link prediction task for random graphs (accuracy metric lies between (0,1) and higher value means better results). Bold corresponds to the best metric for each dataset.

	RF	LR	GBM
Walklets			
CORA Original	0.8142 ± 0.0252	0.8124 ± 0.0317	0.8327 ± 0.0326
CORA-ER	0.617 ± 0.0081	0.5658 ± 0.013	0.5904 ± 0.0141
CORA-BA	0.7216 ± 0.0113	0.6928 ± 0.0103	0.7271 ± 0.0136
CORA-WS	0.6511 ± 0.0329	0.5168 ± 0.0075	**0.7442 ± 0.0762**
CITESEER Original	0.6291 ± 0.0280	0.6243 ± 0.0228	0.6480 ± 0.0277
CITESEER-ER	0.5505 ± 0.0062	0.5335 ± 0.0071	0.5411 ± 0.0076
CITESEER-BA	0.6807 ± 0.0071	0.662 ± 0.0123	**0.6871 ± 0.018**
CITESEER-WS	0.571 ± 0.0142	0.5232 ± 0.022	0.6121 ± 0.031
HOPE			
CORA Original	0.7518 ± 0.0333	0.3024 ± 0.0308	0.7614 ± 0.0289
CORA-ER	0.5936 ± 0.0042	0.5114 ± 0.0055	0.5734 ± 0.0063
CORA-BA	**0.6521 ± 0.0071**	0.5559 ± 0.0144	0.6312 ± 0.007
CORA-WS	0.5115 ± 0.0048	0.51 ± 0.0052	0.5132 ± 0.0071
CITESEER Original	0.5468 ± 0.0346	0.2663 ± 0.0443	0.5489 ± 0.0378
CITESEER-ER	0.5509 ± 0.015	0.5066 ± 0.0029	0.5439 ± 0.01
CITESEER-BA	**0.6304 ± 0.0116**	0.5422 ± 0.0071	0.6096 ± 0.0056
CITESEER-WS	0.5169 ± 0.0057	0.5093 ± 0.0058	0.521 ± 0.0088
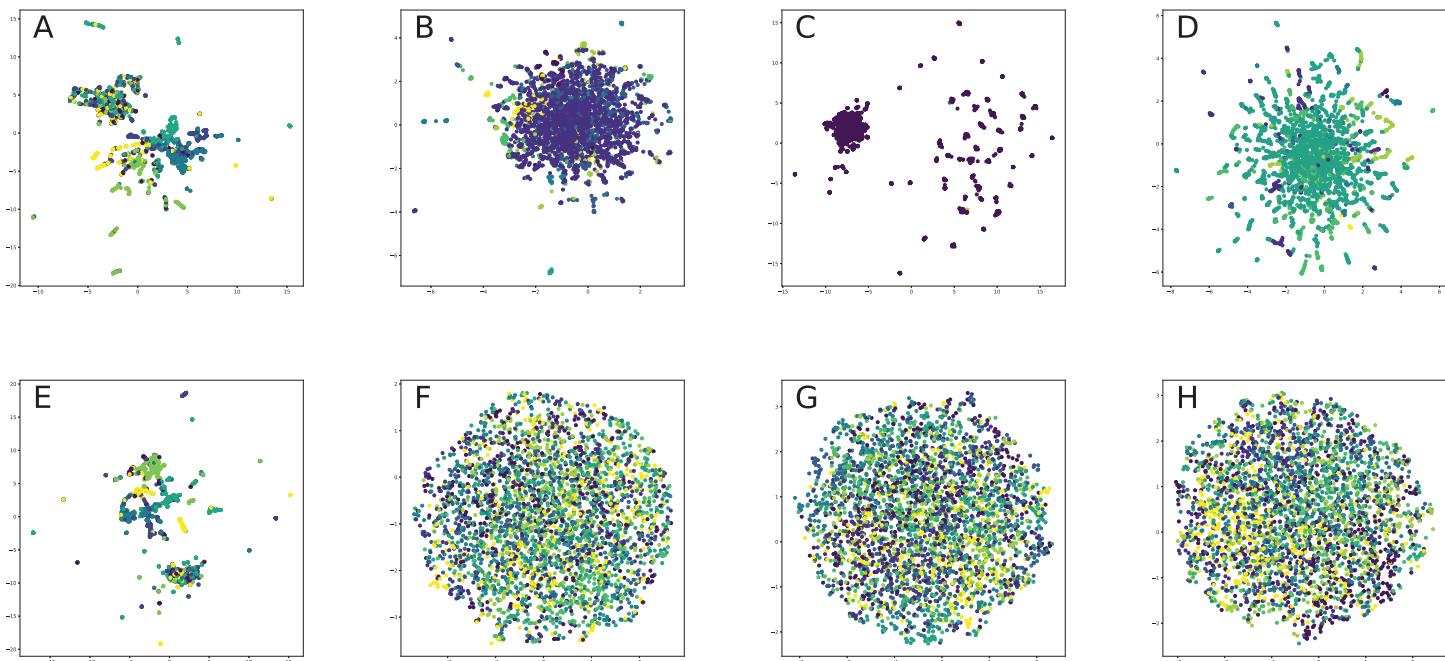
Figure 4 UMAP projection of Citeseer based random graph embeddings: (A) Original graph, HOPE. (B) Erdős-Rényi, HOPE. (C) Barabási-Albert, HOPE. (D) Watts-Strogatz, HOPE. (E) Original graph, Walklets. (F) Erdős-Rényi, Walklets. (G) Barabási-Albert, Walklets. (H) Watts-Strogatz, Walklets.

Full-size 🖼 DOI: 10.7717/peerj-cs.357/fig-4

Makarov et al. (2021), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.357

37/62
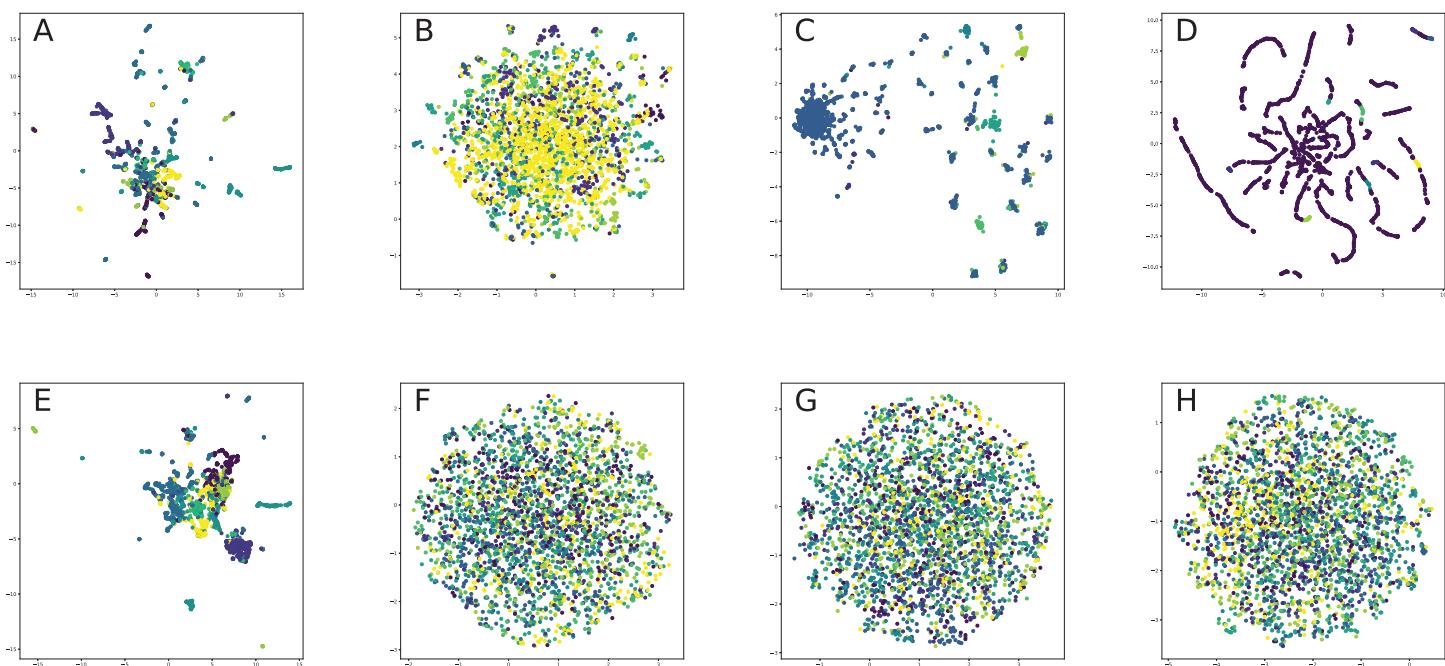
Figure 5 UMAP projection of CORA based random graph embeddings: (A) Original graph, HOPE. (B) Erdős-Rényi, HOPE. (C) Barabási-Albert, HOPE. (D) Watts-Strogatz, HOPE. (E) Original graph, Walklets. (F) Erdős-Rényi, Walklets. (G) Barabási-Albert, Walklets. (H) Watts-Strogatz, Walklets. Full-size ☑ DOI: 10.7717/peerj-cs.357/fig-5

We select these embeddings because it is non-context, show good performance and saves different properties. Walklets preserves K-hop similarity and HOPE preserves asymmetric transitivity.

Similarly to the experiments on the real-world graphs, Walklets shows superior performance in comparison to the HOPE in Clustering (Table 9) and LPP (Table 10) tasks.

However, visually (Figs. 4 and 5) it better separates dense clusters. The Walklets visualization of random graphs differs from real-world cases. Random graphs give much sparser and visually harder to distinguish structure.

The results on random graphs and real networks differ sufficiently. It means that embedding models could really learn graph structure and its properties. Also, such citation networks are poorly described by random graph models.

CONCLUSION

In the current work, we present a comprehensive survey of graph embedding techniques. The work overviews different types of graph embeddings with respect to methods, network types, their applications to computer science domains.

One of the main achievements at the moment are the scalable models. The GNN could be trained in batch and distributed fashion. Such methods allow using powerful attribute-aware models for real-world large graphs. However, only a few works analyze the proper strategies for batch sampling and its effect on the final results in terms of bias-variance trade-off. Another way to accelerate GNN training is to coarse a graph, but it could affect

dramatically the final quality of the model. So, the understanding and further developing of coarsening and sampling techniques is the promising direction.

One of the most popular technique for graph embedding at the current time is the attention mechanism. It helps to account for some significant properties of a graph like temporality and heterogeneity by introducing attention in different dimensions: time, different levels of edge and nodes types. However, this method could be exhaustive in terms of computation, so it should be used with acceleration techniques.

The other research direction that grows rapidly is the stability of graph embeddings. The popular practices are to use variational graph autoencoders with Gaussian denoising or adversarial training.

The development of scalable and high-quality graph neural networks leads to an increase in the number of applications to the non-graph domain. The most common application of it is the modeling of similarity or nearest neighbors graphs. Such approaches are presented in natural language processing, computer vision and recommender systems. However, in many fields, structures could be natively presented as graphs in terms of labels (samples of one type are connected), interaction, knowledge or relation graphs.

Our survey covers the most complete of methods and application in different computer science domains related to machine learning problems on relational data.

In addition, in the experiment part of the study we provide results on training best graph embedding models for node classification, link prediction, node clustering and network visualization tasks for different types of models and graphs to understand why certain graph embedding perform better than others on benchmark datasets under different training settings. Our experiments explain how different embeddings work with different properties uncovering graph inner properties and descriptive statistics impact on the models performance. As one of the most interesting findings, we show that structural embeddings with proper objectives achieve competitive quality vs graph neural networks.

Still, it could be hard to apply such methods to large graphs. Firstly, there is a problem with high computational complexity. Graph neural networks solve this issue by using batch training and sampling techniques. Another problem is that learned structural embeddings for large graphs could be noisy. However, adding the node attributes helps to concentrate on the specific important properties. Modern models focus on accounting for node attributes, but it was found that more important question is how to balance a trade-off between node attributes and network structure. Our work will be helpful in the further development of such generalization methods to answer this question. Such methods will allow to easily apply graph models in different domains.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Grant Disclosures

Competing Interests

The authors declare that they have no competing interests.

Author Contributions

- Ilya Makarov conceived and designed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Dmitrii Kiselev conceived and designed the experiments, performed the experiments, analyzed the data, performed the computation work, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.
- Nikita Nikitinsky conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.
- Lovro Subelj conceived and designed the experiments, analyzed the data, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability

The following information was supplied regarding data availability:

Metric evaluation code for each of the presented tasks is available in the Supplemental Files.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj-cs.357#supplemental-information.

REFERENCES

Abdelaziz I, Fokoue A, Hassanzadeh O, Zhang P, Sadoghi M. 2017. Large-scale structural and textual similarity-based mining of knowledge graph to predict drug-drug interactions. *Journal of Web Semantics* **44**:104–117 DOI 10.1016/j.websem.2017.06.002.

Abu-El-Haija S, Perozzi B, Al-Rfou R. 2017. Learning edge representations via low-rank asymmetric projections. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* ACM, 1787–1796.

Abu-El-Haija S, Perozzi B, Al-Rfou R, Alemi AA. 2018. Watch your step: learning node embeddings via graph attention. In: *Advances in Neural Information Processing Systems.* 9198–9208.

Adafre SF, de Rijke M. 2005. Discovering missing links in wikipedia. In: *Proceedings of the 3rd International Workshop on Link Discovery, LinkKDD '05.* New York: ACM, 90–97.

Adamic LA, Adar E. 2003. Friends and neighbors on the web. *Social networks* **25(3)**:211–230 DOI 10.1016/S0378-8733(03)00009-1.

Ahmed A, Shervashidze N, Narayanamurthy S, Josifovski V, Smola AJ. 2013. Distributed large-scale natural graph factorization. In: *Proceedings of the 22nd International Conference on World Wide Web.* ACM, 37–48.

Akyildiz TA, Aljundi AA, Kaya K. 2020. Gosh: Embedding big graphs on small hardware. In: *49th International Conference on Parallel Processing—ICPP, ICPP '20.* New York: Association for Computing Machinery.

Alanis-Lobato G, Mier P, Andrade-Navarro MA. 2016. Efficient embedding of complex networks to hyperbolic space via their laplacian. *Scientific Reports* **6(1)**:30108 DOI 10.1038/srep30108.

Alshahrani M, Khan MA, Maddouri O, Kinjo AR, Queralt-Rosinach N, Hoehndorf R. 2017. Neuro-symbolic representation learning on biological knowledge graphs. *Bioinformatics* **33(17)**:2723–2730 DOI 10.1093/bioinformatics/btx275.

As Feder T, Motwani R. 1991. Clique partitions, graph compression, and speeding-up algorithms. In: *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing.* New York: ACM, 123–133.

Atahan Akyildiz T, Alabsi Aljundi A, Kaya K. 2020. Understanding coarsening for embedding large-scale graphs. arXiv preprint arXiv:2009.04925.

Atwood J, Towsley D. 2016. Diffusion-convolutional neural networks. In: *Advances in Neural Information Processing Systems.* 1993–2001.

Azran A. 2007. The rendezvous algorithm: Multiclass semi-supervised learning with markov random walks. In: *Proceedings of the 24th international conference on Machine learning.* New York: ACM, 49–56.

Backstrom L, Leskovec J. 2011. Supervised random walks: predicting and recommending links in social networks. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, WSDM '11.* New York: ACM, 635–644.

Baldini L, Martino A, Rizzi A. 2020. Exploiting cliques for granular computing-based graph classification. In: *2020 International Joint Conference on Neural Networks (IJCNN).* Piscataway: IEEE, 1–9.

Baluja S, Seth R, Sivakumar D, Jing Y, Yagnik J, Kumar S, Ravichandran D, Aly M. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In: *Proceedings of the 17th International Conference on World Wide Web.* New York: ACM, 895–904.

Banerjee S, Khapra MM. 2019. Graph convolutional network with sequential attention for goal-oriented dialogue systems. *Transactions of the Association for Computational Linguistics* **7(1)**:485–500 DOI 10.1162/tacl_a_00284.

Barabási A-L, Albert R. 1999. Emergence of scaling in random networks. *Science* **286(5439)**:509–512 DOI 10.1126/science.286.5439.509.

Barrat A, Weigt M. 2000. On the properties of small-world network models. *European Physical Journal B-Condensed Matter and Complex Systems* **13(3)**:547–560 DOI 10.1007/s100510050067.

Belkin M, Niyogi P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *Advances in NIPS.* Cambridge: MIT press, 585–591.

Berg Rv d, Kipf TN, Welling M. 2017. Graph convolutional matrix completion. arXiv preprint arXiv:1706.02263.

Bhagat S, Cormode G, Muthukrishnan S. 2011. Node classification in social networks. In: *Social Network Data Analytics.* Berlin: Springer, 115–148.

Bhagat S, Cormode G, Rozenbaum I. 2009. Applying link-based classification to label blogs. In: *Advances in Web Mining and Web Usage Analysis.* Berlin: Springer, 97–117.

Bojcheski A, Günnemann S. 2018. Adversarial attacks on node embeddings. arXiv preprint arXiv:1809.01093.

Bordes A, Glorot X, Weston J, Bengio Y. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In: *Artificial Intelligence and Statistics.* Cambridge: PMLR, MIT Press, 127–135.

Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. 2013. Translating embeddings for modeling multi-relational data. In: *Advances in Neural Information Processing Systems.* Cambridge: MIT press, 2787–2795.

Bordes A, Weston J, Collobert R, Bengio Y. 2011. Learning structured embeddings of knowledge bases. In: *Twenty-Fifth AAAI Conference on Artificial Intelligence.* AAAI, 25–27.

Brand M. 2003. Continuous nonlinear dimensionality reduction by kernel eigenmaps. In: *IJCAI.* 547–554.

Brochier R, Guille A, Velcin J. 2019. Global vectors for node representations. arXiv preprint arXiv:1902.11004.

Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34(4)**:18–42 DOI 10.1109/MSP.2017.2693418.

Bruna J, Zaremba W, Szlam A, LeCun Y. 2013. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203.

Cai H, Zheng VW, Chang KC-C. 2017. A comprehensive survey of graph embedding: problems, techniques and applications. arXiv preprint arXiv:1709.07604.

Cao M, Ma X, Zhu K, Xu M, Wang C. 2020. Heterogeneous information network embedding with convolutional graph attention networks. In: *2020 International Joint Conference on Neural Networks (IJCNN).* Piscataway: IEEE, 1–8.

Cao S, Lu W, Xu Q. 2015. Grarep: learning graph representations with global structural information. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management.* New York: ACM, 891–900.

Cao S, Lu W, Xu Q. 2016. Deep neural networks for learning graph representations. In: *AAAI.* 1145–1152.

Cavallari S, Zheng VW, Cai H, Chang KC-C, Cambria E. 2017. Learning community embedding with community detection and node embedding on graphs. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management.* New York: ACM, 377–386.

Çelikkanat A, Malliaros FD. 2019. Exponential family graph embeddings. arXiv preprint arXiv:1911.09007.

Chamberlain BP, Clough J, Deisenroth MP. 2017. Neural embeddings of graphs in hyperbolic space. arXiv preprint arXiv:1705.10359.

Chang J, Blei D. 2009. Relational topic models for document networks. In: *Artificial Intelligence and Statistics.* 81–88.

Chang S, Han W, Tang J, Qi G-J, Aggarwal CC, Huang TS. 2015. Heterogeneous network embedding via deep architectures. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 119–128.

Chen C, Min Z, Liu Y, Ma S. 2019a. Social attentional memory network: modeling aspect- and friend-level differences in recommendation. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* New York: ACM, 177–185.

Chen C-M, Chien P-C, Lin Y-C, Tsai M-F, Yang Y-H. 2015a. Exploiting latent social listening representations for music recommendations. In: *Proceedings of the Ninth ACM International Conference Recommender Systems Poster.* 1–2.

Chen C-M, Tsai M-F, Lin Y-C, Yang Y-H. 2016a. Query-based music recommendations via preference embedding. In: *Proceedings of the 10th ACM Conference on Recommender Systems.* New York: ACM, 79–82.

Chen H, Anantharam AR, Skiena S. 2017. Deepbrowse: similarity-based browsing through large lists. In: *International Conference on Similarity Search and Applications.* Berlin: Springer, 300–314.

Chen H, Koga H. 2019. Gl2vec: graph embedding enriched by line graphs with edge features. In: Gedeon T, Wong KW, Lee M, eds. *Neural Information Processing.* Cham: Springer International Publishing, 3–14.

Chen H, Li X, Huang Z. 2005. Link prediction approach to collaborative filtering. In: *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '05).* New York: ACM, 141–142.

Chen H, Perozzi B, Al-Rfou R, Skiena S. 2018a. A tutorial on network embeddings. arXiv preprint arXiv:1808.02590.

Chen H, Perozzi B, Hu Y, Skiena S. 2018b. Harp: hierarchical representation learning for networks. In: *Thirty-Second AAAI Conference on Artificial Intelligence.* 2121–2134.

Chen J, Ma T, Xiao C. 2018. Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint arXiv:1801.10247.

Chen J, Shi Z, Wu Y, Xu X, Zheng H. 2018c. Link prediction adversarial attack. arXiv preprint arXiv:1810.01110.

Chen J, Wu Y, Lin X, Xuan Q. 2019b. Can adversarial network attack be defended? arXiv preprint arXiv:1903.05994.

Chen J, Wu Y, Xu X, Chen Y, Zheng H, Xuan Q. 2018e. Fast gradient attack on network embedding. arXiv preprint arXiv:1809.02797.

Chen J, Zhang A. 2020. Hgmf: heterogeneous graph-based fusion for multimodal data with incompleteness. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20.* New York: Association for Computing Machinery, 1295–1305.

Chen J, Zhang Q, Huang X. 2016b. Incorporate group information to enhance network embedding. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management.* New York: ACM, 1901–1904.

Chen J, Zhu J, Song L. 2017. Stochastic training of graph convolutional networks with variance reduction. arXiv preprint arXiv:1710.10568.

Chen M, Tsang IW, Tan M, Cham TJ. 2015b. A unified feature selection framework for graph embedding on high dimensional data. *IEEE Transactions on Knowledge and Data Engineering* **27(6)**:1465–1477 DOI 10.1109/TKDE.2014.2382599.

Chen T, Sun Y. 2017. Task-guided and path-augmented heterogeneous network embedding for author identification. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* New York: ACM, 295–304.

Chen Y, Rohrbach M, Yan Z, Shuicheng Y, Feng J, Kalantidis Y. 2019c. Graph-based global reasoning networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Piscataway: IEEE.

Chen Y, Sun K, Pu J, Xiong Z, Zhang X. 2020. Grapasa: parametric graph embedding via siamese architecture. *Information Sciences* **512**:1442–1457 DOI 10.1016/j.ins.2019.10.027.

Cheng W, Zhang X, Guo Z, Wu Y, Sullivan PF, Wang W. 2013. Flexible and robust co-regularized multi-domain graph clustering. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 320–328.

Chiang W-L, Liu X, Si S, Li Y, Bengio S, Hsieh C-J. 2019. Cluster-gcn: an efficient algorithm for training deep and large graph convolutional networks. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 257–266.

Cho H, Yu Y. 2018. Link prediction for interdisciplinary collaboration via co-authorship network. *Social Network Analysis and Mining* **8(1)**:25 DOI 10.1007/s13278-018-0501-6.

Chung FR, Graham FC. 1997. *Spectral graph theory.* Vol. 92. Rhode Island: American Mathematical Soc.

Clauset A, Moore C, Newman ME. 2008. Hierarchical structure and the prediction of missing links in networks. *Nature* **453(7191)**:98–101 DOI 10.1038/nature06830.

Cobanoglu MC, Liu C, Hu F, Oltvai ZN, Bahar I. 2013. Predicting drug-target interactions using probabilistic matrix factorization. *Journal of Chemical Information and Modeling* **53(12)**:3399–3409 DOI 10.1021/ci400219z.

Crichton G, Guo Y, Pyysalo S, Korhonen A. 2018. Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. *BMC Bioinformatics* **19(1)**:176 DOI 10.1186/s12859-018-2163-9.

Cui P, Wang X, Pei J, Zhu W. 2018. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* **31(5)**:833–852 DOI 10.1109/TKDE.2018.2849727.

Dai H, Dai B, Song L. 2016. Discriminative embeddings of latent variable models for structured data. In: *International conference on machine learning.* New York: ACM, 2702–2711.

Dai H, Li H, Tian T, Huang X, Wang L, Zhu J, Song L. 2018. Adversarial attack on graph structured data. arXiv preprint arXiv:1806.02371.

De Oliveira MF, Levkowitz H. 2003. From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics* **9(3)**:378–394 DOI 10.1109/TVCG.2003.1207445.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41(6)**:391–407 DOI 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

Defferrard M, Bresson X, Vandergheynst P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In: *Advances in Neural Information Processing Systems.* 3844–3852.

Devlin J, Chang M-W, Lee K, Toutanova K. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Didimo W, Liotta G, Montecchiani F. 2019. A survey on graph drawing beyond planarity. *ACM Computing Surveys* **52(1)**:1–37 DOI 10.1145/3301281.

Ding CH, He X, Zha H, Gu M, Simon HD. 2001. A min-max cut algorithm for graph partitioning and data clustering. In: *Proceedings 2001 IEEE International Conference on Data Mining.* Piscataway: IEEE, 107–114.

Ding M, Tang J, Zhang J. 2018. Semi-supervised learning on graphs with generative adversarial nets. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* New York: ACM, 913–922.

Do DT, Le TQT, Le NQK. 2020. Using deep neural networks and biological subwords to detect protein s-sulfenylation sites. *Briefings in Bioinformatics* **14**:1049 DOI 10.1093/bib/bbaa128.

Dong Y, Chawla NV, Swami A. 2017. Metapath2vec: scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 135–144.

Donnat C, Holmes S. 2018. Tracking network dynamics: a survey using graph distances. *Annals of Applied Statistics* **12(2)**:971–1012 DOI 10.1214/18-AOAS1176.

Donnat C, Zitnik M, Hallac D, Leskovec J. 2018. Learning structural node embeddings via diffusion wavelets. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* **24**:1320–1329.

Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams RP. 2015. Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in Neural Information Processing Systems.* Cambridge: MIT press, 2224–2232.

Efron B. 1992. Bootstrap methods: another look at the jackknife. In: *Breakthroughs in Statistics.* Berlin: Springer International Publishing, 569–593.

Erdös P, Rényi A. 1959. On random graphs publ. *Publicationes Mathematicae Debrecen* **6**:290–297.

Ezzat A, Wu M, Li X-L, Kwoh C-K. 2017. Drug-target interaction prediction using ensemble learning and dimensionality reduction. *Methods* **129**:81–88 DOI 10.1016/j.ymeth.2017.05.016.

Ezzat A, Zhao P, Wu M, Li X-L, Kwoh C-K. 2016. Drug-target interaction prediction with graph regularized matrix factorization. *IEEE/ACM Transactions On Computational Biology and Bioinformatics* **14(3)**:646–656 DOI 10.1109/TCBB.2016.2530062.

Fan M, Zhou Q, Chang E, Zheng TF. 2014. Transition-based knowledge graph embedding with relational mapping properties. *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing* **28**:328–337.

Fang H, Wu F, Zhao Z, Duan X, Zhuang Y, Ester M. 2016. Community-based question answering via heterogeneous social network learning. In: *Thirtieth AAAI Conference on Artificial Intelligence.* 122–128.

Fathy A, Li K. 2020. Temporalgat: attention-based dynamic graph representation learning. In: Lauw HW, Wong RC-W, Ntoulas A, Lim E-P, Ng S-K, Pan SJ, eds. *Advances in Knowledge Discovery and Data Mining.* Cham: Springer International Publishing, 413–423.

Feng R, Yang Y, Hu W, Wu F, Zhang Y. 2018. Representation learning for scale-free networks. In: *Thirty-Second AAAI Conference on Artificial Intelligence.*

Fey M, Eric Lenssen J, Weichert F, Müller H. 2018. Splinecnn: fast geometric deep learning with continuous b-spline kernels. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* Piscataway: IEEE, 869–877.

Fortunato S. 2010. Community detection in graphs. *Physics Reports* **486(3–5)**:75–174 DOI 10.1016/j.physrep.2009.11.002.

Fu X, Zhang J, Meng Z, King I. 2020. Magnn: metapath aggregated graph neural network for heterogeneous graph embedding. In: *Proceedings of the Web Conference 2020, WWW '20.* New York: Association for Computing Machinery, 2331–2341.

Gallicchio C, Micheli A. 2020. Fast and deep graph neural networks. In: *AAAI.* 3898–3905.

Ganguly S, Gupta M, Varma V, Pudi V. 2016. et al.Author2vec: learning author representations by combining content and link information. In: *Proceedings of the 25th International Conference Companion on World Wide Web.* 49–50.

Gao F, Musial K, Cooper C, Tsoka S. 2015. Link prediction methods and their accuracy for different social networks and network metrics. *Scientific Programming* **2015(3)**:1–13 DOI 10.1155/2015/172879.

Gao J, Zhang T, Xu C. 2019. Graph convolutional tracking. In: *CVPR*.

Gao S, Denoyer L, Gallinari P. 2011. Temporal link prediction by integrating content and structure information. In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management, CIKM '11.* New York: ACM, 1169–1174.

Geng X, Zhang H, Bian J, Chua T-S. 2015. Learning image and user features for recommendation in social networks. In: *Proceedings of the IEEE International Conference on Computer Vision.* Piscataway: IEEE, 4274–4282.

Getoor L, Taskar B. 2007. Statistical relational learning. https://mitpress.mit.edu/books/introduction-statistical-relational-learning.

Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. 2017. Neural message passing for quantum chemistry. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70.* 1263–1272.

Golub GH, Reinsch C. 1971. Singular value decomposition and least squares solutions. In: *Linear Algebra.* Berlin: Springer, 134–151.

Goyal P, Chhetri SR, Canedo A. 2020. Dyngraph2vec: capturing network dynamics using dynamic graph representation learning. *Knowledge-Based Systems* **187**:104816 DOI 10.1016/j.knosys.2019.06.024.

Goyal P, Ferrara E. 2017. Graph embedding techniques, applications, and performance: a survey. arXiv preprint arXiv:1705.02801.

Goyal P, Hosseinmardi H, Ferrara E, Galstyan A. 2018. Embedding networks with edge attributes. In: *Proceedings of the 29th on Hypertext and Social Media.* New York: ACM, 38–42.

Grover A, Leskovec J. 2016. Node2vec: scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD IC on KDD* **22**:855–864.

Gui H, Liu J, Tao F, Jiang M, Norick B, Han J. 2016. Large-scale embedding learning in heterogeneous event data. In: *2016 IEEE 16th International Conference on Data Mining (ICDM).* Piscataway: IEEE, 907–912.

Guo Z, Zhang Y, Lu W. 2019. Attention guided graph convolutional networks for relation extraction. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence: Association for Computational Linguistics, 241–251.

Haddad M, Bothorel C, Lenca P, Bedart D. 2019. Temporalnode2vec: temporal node embedding in temporal networks. In: *International Conference on Complex Networks and Their Applications.* Berlin: Springer International Publishing, 891–902.

Hamilton W, Ying Z, Leskovec J. 2017a. Inductive representation learning on large graphs. In: *Advances in NIPS.* Cambridge: MIT press, 1025–1035.

Hamilton WL, Ying R, Leskovec J. 2017b. Representation learning on graphs: methods and applications. arXiv preprint arXiv:1709.05584.

Hasan MA, Zaki MJ. 2011. *A survey of link prediction in social networks, chapter 9.* Boston: Springer US, 243–275.

Hayashi K, Ohsaki M. 2020. Reinforcement learning and graph embedding for binary truss topology optimization under stress and displacement constraints. *Frontiers in Built Environment* **6**:59 DOI 10.3389/fbuil.2020.00059.

He Q, Pei J, Kifer D, Mitra P, Giles L. 2010. Context-aware citation recommendation. In: *Proceedings of the 19th International Conference on World Wide Web, WWW '10.* New York: ACM, 421–430.

He S, Liu K, Ji G, Zhao J. 2015. Learning to represent knowledge graphs with gaussian embedding. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* **24**:623–632.

He X, Niyogi P. 2004. Locality preserving projections. In: *Advances in Neural Information Processing Systems.* 153–160.

Heckerman D, Meek C, Koller D. 2007. Probabilistic entity-relationship models, prms, and plate models. In: *Introduction to Statistical Relational Learning.* 201–238.

Henaff M, Bruna J, LeCun Y. 2015. Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163.

Herman I, Melançon G, Marshall MS. 2000. Graph visualization and navigation in information visualization: a survey. *IEEE Transactions on Visualization and Computer Graphics* **6(1)**:24–43 DOI 10.1109/2945.841119.

Hettige B, Wang W, Li Y-F, Buntine W. 2020. Robust attribute and structure preserving graph embedding. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Berlin: Springer, 593–606.

Hong H, Lin Y, Yang X, Li Z, Fu K, Wang Z, Qie X, Ye J. 2020. Heteta: heterogeneous information network embedding for estimating time of arrival. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20.* New York: Association for Computing Machinery, 2444–2454.

Hou C, Nie F, Tao H, Yi D. 2017. Multi-view unsupervised feature selection with adaptive similarity and view weight. *IEEE Transactions on Knowledge and Data Engineering* **29(9)**:1998–2011 DOI 10.1109/TKDE.2017.2681670.

Hu B, Fang Y, Shi C. 2019. Adversarial learning on heterogeneous information networks. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19.* New York: Association for Computing Machinery, 120–129.

Huang X, Li J, Hu X. 2017. Label informed attributed network embedding. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* New York: ACM, 731–739.

Huang X, Zhang J, Li D, Li P. 2019. Knowledge graph embedding based question answering. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.* New York: ACM. Vol. 12. 105–113.

Huang Z, Mamoulis N. 2017. Heterogeneous information network embedding for meta path based proximity. arXiv preprint arXiv:1701.05291.

Islam MR, Prakash BA, Ramakrishnan N. 2018. Signet: Scalable embeddings for signed networks. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Berlin: Springer International Publishing, 157–169.

Jacob Y, Denoyer L, Gallinari P. 2014. Learning latent representations of nodes for classifying in heterogeneous social networks. In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining.* New York: ACM, 373–382.

Jenatton R, Roux NL, Bordes A, Obozinski GR. 2012. A latent factor model for highly multi-relational data. In: *Advances in Neural Information Processing Systems.* Cambridge: MIT press, 3167–3175.

Ji G, He S, Xu L, Liu K, Zhao J. 2015. Knowledge graph embedding via dynamic mapping matrix. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*

and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), volume 53. 687–696.

Ji G, Liu K, He S, Zhao J. 2016. Knowledge graph completion with adaptive sparse transfer matrix. In: *Thirtieth AAAI Conference on Artificial Intelligence.* New York: AAAI, 30.

Jiang Z, Gao Z, Lan J, Yang H, Lu Y, Liu X. 2020. Task-oriented genetic activation for large-scale complex heterogeneous graph embedding. In: *Proceedings of the Web Conference 2020, WWW '20.* New York: Association for Computing Machinery, 1581–1591.

Jing Y, Wang H, Shao K, Huo X, Zhang Y. 2020. Unsupervised graph representation learning with variable heat kernel. *IEEE Access* **8**:15800–15811 DOI 10.1109/ACCESS.2020.2966409.

Johansson FD, Dubhashi D. 2015. Learning with similarity functions on graphs using matchings of geometric embeddings. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 467–476.

Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. 2016. Molecular graph convolutions: moving beyond fingerprints. *Journal of Computer-Aided Molecular Design* **30(8)**:595–608 DOI 10.1007/s10822-016-9938-8.

Kefato ZT, Sheikh N, Montresor A. 2020. Which way? direction-aware attributed graph embedding. arXiv preprint arXiv:2001.11297.

Keser R, Nallbani I, Calik N, Ayanzadeh A, Töreyin B. 2020. Graph embedding for link prediction using residual variational graph autoencoders. In: *The 28th IEEE Conference on Signal Processing and Communications Applications.* Piscataway: IEEE.

Khalil E, Dai H, Zhang Y, Dilkina B, Song L. 2017. Learning combinatorial optimization algorithms over graphs. In: *Advances in Neural Information Processing Systems.* 6348–6358.

Khasahmadi AH, Hassani K, Moradi P, Lee L, Morris Q. 2020. Memory-based graph networks. In: *International Conference on Learning Representations..*

Kim D, Kim S, Kwak N. 2018. Textbook question answering with knowledge graph understanding and unsupervised open-set text comprehension. arXiv preprint arXiv:1811.00232.

Kim J, Kim T, Kim S, Yoo CD. 2019. Edge-labeling graph neural network for few-shot learning. arXiv preprint arXiv:1905.01436.

Kim J, Park H, Lee J-E, Kang U. 2018. Side: Representation learning in signed directed networks. In: *Proceedings of the 2018 World Wide Web Conference, WWW '18.* Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 509–518.

Kingma DP, Welling M. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kipf TN, Welling M. 2016a. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Kipf TN, Welling M. 2016b. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308.

Kleinberg R. 2007. Geographic routing using hyperbolic space. In: *IEEE Infocom 2007-26th IEEE International Conference On Computer Communications.* Piscataway: IEEE, 1902–1909.

Kolouri S, Naderializadeh N, Rohde GK, Hoffmann H. 2020. Wasserstein embedding for graph learning. arXiv preprint arXiv:2006.09430.

Kong X, Mao M, Wang W, Liu J, Xu B. 2018. Voprec: vector representation learning of papers with text information and structural identity for recommendation. In: *IEEE Transactions on Emerging Topics in Computing..*

Krioukov D, Papadopoulos F, Boguñá M, Vahdat A. 2009. Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces. *ACM SIGMETRICS Performance Evaluation Review* **37(2)**:15–17 DOI 10.1145/1639562.1639568.

Kruskal J, Wish M. 1978. *Multidimensional Scaling.* New York: SAGE Publications.

Kulmanov M, Khan MA, Hoehndorf R. 2018. Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* **34(4)**:660–668 DOI 10.1093/bioinformatics/btx624.

Laakom F, Raitoharju J, Passalis N, Iosifidis A, Gabbouj M. 2020. Graph embedding with data uncertainty. arXiv preprint arXiv:2009.00505.

Le NQK, Yapp EKY, Yeh H-Y. 2019. Et-gru: using multi-layer gated recurrent units to identify electron transport proteins. *BMC Bioinformatics* **20(1)**:377 DOI 10.1186/s12859-019-2972-5.

Le TM, Lauw HW. 2014. Probabilistic latent document network embedding. In: *2014 IEEE International Conference on Data Mining (ICDM).* Piscataway: IEEE, 270–279.

Lee JB, Rossi R, Kong X. 2018. Graph classification using structural attention. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* New York: ACM, 1666–1674.

Lee JB, Rossi RA, Kim S, Ahmed NK, Koh E. 2019. Attention models in graphs: a survey. *ACM Transactions on Knowledge Discovery from Data* **13(6)**:1–25 DOI 10.1145/3363574.

Lei M, Shi Y, Niu L. 2018. The applications of stochastic models in network embedding: a survey. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI).* Piscataway: IEEE, 635–638.

Leskovec J, Huttenlocher D, Kleinberg J. 2010. Signed networks in social media. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '10.* New York: Association for Computing Machinery, 1361–1370.

Levie R, Monti F, Bresson X, Bronstein MM. 2017. Cayleynets: graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing* **67(1)**:97–109 DOI 10.1109/TSP.2018.2879624.

Li B, Pi D, Lin Y, Khan IA, Cui L. 2020a. Multi-source information fusion based heterogeneous network embedding. *Information Sciences* **534**:53–71 DOI 10.1016/j.ins.2020.05.012.

Li G, Luo J, Xiao Q, Liang C, Ding P, Cao B. 2017a. Predicting microrna-disease associations using network topological similarity based on deepwalk. *IEEE Access* **5**:24032–24039 DOI 10.1109/ACCESS.2017.2766758.

Li J, Chen C, Tong H, Liu H. 2018. Multi-layered network embedding. In: *Proceedings of the 2018 SIAM International Conference on Data Mining.* SIAM, 684–692.

Li J, Ritter A, Jurafsky D. 2015. Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks. arXiv preprint arXiv:1510.05198.

Li J, Zhu J, Zhang B. 2016. Discriminative deep random walk for network classification. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1.* 1004–1013.

Li K, Gao J, Guo S, Du N, Li X, Zhang A. 2014a. Lrbm: a restricted boltzmann machine based approach for representation learning on linked data. In: *2014 IEEE International Conference on Data Mining.* Piscataway: IEEE, 300–309.

Li M, Chen S, Chen X, Zhang Y, Wang Y, Tian Q. 2019a. Actional-structural graph convolutional networks for skeleton-based action recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Piscataway: IEEE.

Li Q, Shang Y, Qiao X, Dai W. 2020b. Heterogeneous dynamic graph attention network. In: *2020 IEEE International Conference on Knowledge Graph (ICKG).* Piscataway: IEEE, 404–411.

Li W, Xu J, He Y, Yan S, Wu Y, Sun X. 2019b. Coherent comments generation for Chinese articles with a graph-to-sequence model. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence: Association for Computational Linguistics, 4843–4852.

Makarov et al. (2021), *PeerJ Comput. Sci.*, DOI 10.7717/peerj-cs.357

49/62

Li X, Chen H. 2009. Recommendation as link prediction: a graph kernel-based machine learning approach. In: *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries, JCDL '09.* New York: ACM, 213–216.

Li X, Chen W, Chen Y, Zhang X, Gu J, Zhang MQ. 2017b. Network embedding-based representation learning for single cell rna-seq data. *Nucleic Acids Research* **45(19)**:e166 DOI 10.1093/nar/gkx750.

Li X, Du N, Li H, Li K, Gao J, Zhang A. 2014b. A deep learning approach to link prediction in dynamic networks. In: *Proceedings of the 2014 SIAM International Conference on Data Mining.* SIAM, 289–297.

Li Y, Nie F, Huang H, Huang J. 2015b. Large-scale multi-view spectral clustering via bipartite graph. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Volume 29.* 2750–2756.

Li Y, Tarlow D, Brockschmidt M, Zemel R. 2015c. Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493.

Li Y, Yu R, Shahabi C, Liu Y. 2017c. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.

Liben-Nowell D, Kleinberg J. 2007. The link-prediction problem for social networks. *Journal of the Association for Information Science and Technology* **58(7)**:1019–1031.

Lim KW, Buntine WL. 2016. Bibliographic analysis with the citation network topic model. arXiv preprint arXiv:1609.06826.

Lin B, Ghaddar B, Nathwani J. 2020a. Deep reinforcement learning for electric vehicle routing problem with time windows. arXiv preprint arXiv:2010.02068.

Lin Y, Hou J, Laurienti PJ, Wu G. 2020b. Detecting changes of functional connectivity by dynamic graph embedding learning. In: Martel AL, Abolmaesumi P, Stoyanov D, Mateus D, Zuluaga MA, Zhou SK, Racoceanu D, Joskowicz L, eds. *Medical Image Computing and Computer Assisted Intervention—MICCAI 2020.* Cham: Springer International Publishing, 489–497.

Lin Y, Liu Z, Sun M, Liu Y, Zhu X. 2015. Learning entity and relation embeddings for knowledge graph completion. *Twenty-Ninth AAAI Conference on Artificial Intelligence* **39**:2181–2187.

Lin Y-Y, Liu T-L, Chen H-T. 2005. Semantic manifold learning for image retrieval. In: *Proceedings of the 13th Annual ACM International Conference on Multimedia.* New York: ACM, 249–258.

Liu A. 2020. Anonymized gcn: A novel robust graph embedding method via hiding node position in noise. arXiv preprint arXiv:2005.03482.

Liu F, Liu B, Sun C, Liu M, Wang X. 2013. Deep learning approaches for link prediction in social network services. In: *International Conference on Neural Information Processing.* Berlin: Springer International Publishing, 425–432.

Liu F, Liu B, Sun C, Liu M, Wang X. 2015a. Deep belief network-based approaches for link prediction in signed social networks. *Entropy* **17(4)**:2140–2169 DOI 10.3390/e17042140.

Liu J, Xu C, Yin C, Wu W, Song Y. 2020a. K-core based temporal graph convolutional network for dynamic graphs. arXiv preprint arXiv:2003.09902.

Liu L, Cheung WK, Li X, Liao L. 2016. Aligning users across social networks using network embedding. In: *IJCAI.* 1774–1780.

Liu R, Cheng W, Tong H, Wang W, Zhang X. 2015b. Robust multi-network clustering via joint cross-domain cluster alignment. In: *2015 IEEE International Conference on Data Mining.* Piscataway: IEEE, 291–300.

Liu W, Chen P-Y, Yeung S, Suzumura T, Chen L. 2017a. Principled multilayer network embedding. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW).* Piscataway: IEEE, 134–141.

Liu X, Li Q, Shen C, Peng X, Zhou Y, Guan X. 2020b. Learning by sampling and compressing: efficient graph representation learning with extremely limited annotations. arXiv preprint arXiv:2003.06100.

Liu X, Murata T, Kim K-S, Kotarasu C, Zhuang C. 2019. A general view for network embedding as matrix factorization. In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Volume 12 of WSDM '19.* New York: ACM, 375–383.

Liu Y, Kou Z. 2007. Predicting who rated what in large-scale datasets. *ACM SIGKDD Explorations Newsletter* **9(2)**:62–65 DOI 10.1145/1345448.1345462.

Liu Y, Zhang X, Liu L, Li G. 2020c. Graph embedding based on characteristic of rooted subgraph structure. In: Li G, Shen HT, Yuan Y, Wang X, Liu H, Zhao X, eds. *Knowledge Science, Engineering and Management.* Cham: Springer International Publishing, 28–39.

Liu Z, Chen C, Li L, Zhou J, Li X, Song L, Qi Y. 2018a. Geniepath: graph neural networks with adaptive receptive paths. arXiv preprint arXiv:1802.00910.

Liu Z, Zheng VW, Zhao Z, Zhu F, Chang KC-C, Wu M, Ying J. 2017b. Semantic proximity search on heterogeneous graph by proximity embedding. In: *AAAI.* 154–160.

Liu Z, Zheng VW, Zhao Z, Zhu F, Chang KC-C, Wu M, Ying J. 2018b. Distance-aware dag embedding for proximity search on heterogeneous graphs. In: *Thirty-Second AAAI Conference on Artificial Intelligence.* New York: AAAI, 2355–2362.

Lu C, Jiao P, Liu H, Wang Y, Xu H, Wang W. 2019. Ssne: status signed network embedding. In: Yang Q, Zhou Z-H, Gong Z, Zhang M-L, Huang S-J, eds. *Advances in Knowledge Discovery and Data Mining.* Cham: Springer International Publishing, 81–93.

Lü L, Zhou T. 2011. Link prediction in complex networks: a survey. *Physica A: Statistical Mechanics and its Applications* **390(6)**:1150–1170 DOI 10.1016/j.physa.2010.11.027.

Lu P-E, Chang C-S. 2020. Explainable, stable, and scalable graph convolutional networks for learning graph representation. arXiv preprint arXiv:2009.10367.

Lu Q, Getoor L. 2003. Link-based classification. In: *Proceedings of the 20th International Conference on Machine Learning (ICML-03).* New York: ACM, 496–503.

Luo D, Nie F, Huang H, Ding CH. 2011. Cauchy graph embedding. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11).* 553–560.

Luo H, Li Y, Fu J, Glass J. 2019. Language modeling with graph temporal convolutional networks. https://openreview.net/forum?id=HJlYzhR9tm.

Luo Y, Zhao X, Zhou J, Yang J, Zhang Y, Kuang W, Peng J, Chen L, Zeng J. 2017. A network integration approach for drug-target interaction prediction and computational drug repositioning from heterogeneous information. *Nature Communications* **8(1)**:1–13 DOI 10.1038/s41467-016-0009-6.

Lv X, Hou L, Li J, Liu Z. 2018. Differentiating concepts and instances for knowledge graph embedding. arXiv preprint arXiv:1811.04588.

Maaten Lv d, Hinton G. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* **9(Nov)**:2579–2605.

Mai G, Janowicz K, Yan B. 2018. Support and centrality: learning weights for knowledge graph embedding models. In: *European Knowledge Acquisition Workshop.* Berlin: Springer International Publishing, 212–227.

Makarov I, Gerasimova O, Sulimov P, Korovina K, Zhukov L. 2019a. Joint node-edge network embedding for link prediction. In: *Springer Proceedings in Mathematics and Statistic (to appear)*. Berlin: Springer International Publishing, 1–12.

Makarov I, Gerasimova O, Sulimov P, Zhukov L. 2019b. Co-authorship network embedding and recommending collaborators via network embedding. In: *Springer Proceedings in Mathematics and Statistic (to appear)*. Berlin: Springer International Publishing, 1–6.

Makarov I, Gerasimova O, Sulimov P, Zhukov LE. 2018. Recommending co-authorship via network embeddings and feature engineering. In: *Proceedings of the 18th ACM/IEEE JCDL*. New York: ACM, 1–2.

Makarov I, Gerasimova O, Sulimov P, Zhukov LE. 2019c. Dual network embedding for representing research interests in the link prediction problem on co-authorship networks. *PeerJ Computer Science* **5(9)**:e172 DOI 10.7717/peerj-cs.172.

Malliaros FD, Vazirgiannis M. 2013. Clustering and community detection in directed networks: a survey. *Physics Reports* **533(4)**:95–142 DOI 10.1016/j.physrep.2013.08.002.

Marcheggiani D, Bastings J, Titov I. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. arXiv preprint arXiv:1804.08313.

Marcheggiani D, Titov I. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. arXiv preprint arXiv:1703.04826.

Martinez AM, Kak AC. 2001. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23(2)**:228–233 DOI 10.1109/34.908974.

McInnes L, Healy J, Melville J. 2018. Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426.

McPherson M, Smith-Lovin L, Cook JM. 2001. Birds of a feather: homophily in social networks. *Annual Review of Sociology* **27(1)**:415–444 DOI 10.1146/annurev.soc.27.1.415.

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. 2013. Distributed representations of words and phrases and their compositionality. In: *Advances in NIPS*. Cambridge: MIT press, 3111–3119.

Monti F, Boscaini D, Masci J, Rodola E, Svoboda J, Bronstein MM. 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5115–5124.

Monti F, Bronstein M, Bresson X. 2017. Geometric matrix completion with recurrent multi-graph neural networks. In: *Advances in Neural Information Processing Systems*. 3697–3707.

Monti F, Shchur O, Bojchevski A, Litany O, Günnemann S, Bronstein MM. 2018. Dual-primal graph convolutional networks. arXiv preprint arXiv:1806.00770.

Mousavi SF, Safayani M, Mirzaei A, Bahonar H. 2017. Hierarchical graph embedding in vector space by graph pyramid. *Pattern Recognition* **61**:245–254 DOI 10.1016/j.patcog.2016.07.043.

Moyano LG. 2017. Learning network representations. *European Physical Journal Special Topics* **226(3)**:499–518 DOI 10.1140/epjst/e2016-60266-2.

Narayanan A, Chandramohan M, Chen L, Liu Y, Saminathan S. 2016. Subgraph2vec: learning distributed representations of rooted sub-graphs from large graphs. arXiv preprint arXiv:1606.08928.

Natarajan N, Dhillon IS. 2014. Inductive matrix completion for predicting gene-disease associations. *Bioinformatics* **30(12)**:i60–i68 DOI 10.1093/bioinformatics/btu269.

Navlakha S, Rastogi R, Shrivastava N. 2008. Graph summarization with bounded error. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. New York: ACM, 419–432.

Newman ME. 2005. A measure of betweenness centrality based on random walks. *Social Networks* **27(1)**:39–54 DOI 10.1016/j.socnet.2004.11.009.

Newman ME, Girvan M. 2004. Finding and evaluating community structure in networks. *Physical Review E* **69(2)**:026113 DOI 10.1103/PhysRevE.69.026113.

Ni J, Cheng W, Fan W, Zhang X. 2016. Self-grouping multi-network clustering. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. Piscataway: IEEE, 1119–1124.

Nickel M, Murphy K, Tresp V, Gabrilovich E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* **104(1)**:11–33 DOI 10.1109/JPROC.2015.2483592.

Nickel M, Rosasco L, Poggio T. 2015. Holographic embeddings of knowledge graphs. arXiv preprint arXiv:1510.04935.

Nickel M, Tresp V, Kriegel H-P. 2011. A three-way model for collective learning on multi-relational data. In: *ICML, Volume 11*. New York: ACM, 809–816.

Nie F, Cai G, Li X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In: *Thirty-First AAAI Conference on Artificial Intelligence*. New York: AAAI, 31.

Nie F, Zhu W, Li X. 2017. Unsupervised large graph embedding. In: *AAAI*. 2422–2428.

Nie F, Zhu W, Li X. 2020. Unsupervised large graph embedding based on balanced and hierarchical k-means. *IEEE Transactions on Knowledge and Data Engineering* **PP(99)**:1.

Niepert M, Ahmed M, Kutzkov K. 2016. Learning convolutional neural networks for graphs. In: *International Conference on Machine Learning*. 2014–2023.

Nikolentzos G, Meladianos P, Vazirgiannis M. 2017. Matching node embeddings for graph similarity. In: *AAAI*. 2429–2435.

Nozza D, Fersini E, Messina E. 2020. Cage: constrained deep attributed graph embedding. *Information Sciences* **518**:56–70 DOI 10.1016/j.ins.2019.12.082.

Ortega A, Frossard P, Kovacevic J, Moura JMF, Vandergheynst P. 2018a. Graph signal processing: overview, challenges, and applications. *Proceedings of the IEEE* **106(5)**:808–828 DOI 10.1109/JPROC.2018.2820126.

Ortega F, Bobadilla J, Gutierrez A, Hurtado R, Li X. 2018b. Artificial intelligence scientific documentation dataset for recommender systems. *IEEE Access* **6**:48543–48555 DOI 10.1109/ACCESS.2018.2867731.

Ou M, Cui P, Pei J, Zhang Z, Zhu W. 2016. Asymmetric transitivity preserving graph embedding. *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **22**:1105–1114.

Pan S, Hu R, Fung S-f, Long G, Jiang J, Zhang C. 2019. Learning graph embedding with adversarial training methods. arXiv preprint arXiv:1901.01250.

Pan S, Wu J, Zhu X, Zhang C, Wang Y. 2016. Tri-party deep network representation. *Network* **11(9)**:12.

Pardalos PM, Xue J. 1994. The maximum clique problem. *Journal of Global Optimization* **4(3)**:301–328 DOI 10.1007/BF01098364.

Pennington J, Socher R, Manning C. 2014. Glove: global vectors for word representation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

Perozzi B, Al-Rfou R, Skiena S. 2014. Deepwalk: online learning of social representations. *Proceedings of the 20th ACM SIGKDD IC on KDD* **20**:701–710.

Perozzi B, Kulkarni V, Chen H, Skiena S. 2017. Don't walk, skip!: online learning of multi-scale network embeddings. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. New York: ACM, 258–265.

Perozzi B, Kulkarni V, Skiena S. 2016. Walklets: multiscale graph embeddings for interpretable network classification. arXiv preprint arXiv:1605.02115.

Phuc LH, Yamada M, Kashima H. 2020. Link prediction on multiple graphs with graph embedding and optimal transport. In: *Proceedings of the National Conference of the Society for Artificial Intelligence, 34th National Conference.* JSAI, 4Rin189.

Pimentel T, Veloso A, Ziviani N. 2017. Unsupervised and scalable algorithm for learning node representations. In: *Proceedings of ICLR Workshop Track.* 1–4.

Pirotte A, Renders J-M, Saerens M. 2007. et al.Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge & Data Engineering* **19(3)**:355–369 DOI 10.1109/TKDE.2007.46.

Qin J, Liu L, Shen H, Hu D. 2020. Uniform pooling for graph networks. *Applied Sciences* **10(18)**:6287 DOI 10.3390/app10186287.

Quiring B, Vassilevski PS. 2020. Multilevel graph embedding. Epub ahead of print 23 September 2020. *Numerical Linear Algebra with Applications* DOI 10.1002/nla.2326.

Ragesh R, Sellamanickam S, Iyer A, Bairi R, Lingam V. 2020. Hetegcn: heterogeneous graph convolutional networks for text classification. arXiv preprint arXiv:2008.12842.

Ren X, He W, Qu M, Voss CR, Ji H, Han J. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 1825–1834.

Ribeiro LF, Saverese PH, Figueiredo DR. 2017. Struc2vec: learning node representations from structural identity. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **23**:385–394.

Rissanen J. 1978. Modeling by shortest data description. *Automatica* **14(5)**:465–471 DOI 10.1016/0005-1098(78)90005-5.

Robins G, Snijders T, Wang P, Handcock M, Pattison P. 2007. Recent developments in exponential random graph (p*) models for social networks. *Social Networks* **29(2)**:192–215 DOI 10.1016/j.socnet.2006.08.003.

Rokka Chhetri S, Al Faruque MA. 2020. *Dynamic graph embedding.* Cham: Springer International Publishing, 209–229.

Rossi E, Chamberlain B, Frasca F, Eynard D, Monti F, Bronstein M. 2020a. Temporal graph networks for deep learning on dynamic graphs. arXiv preprint arXiv:2006.10637.

Rossi E, Frasca F, Chamberlain B, Eynard D, Bronstein M, Monti F. 2020b. Sign: scalable inception graph neural networks. arXiv preprint arXiv:2004.11198.

Roweis ST, Saul LK. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290(5500)**:2323–2326 DOI 10.1126/science.290.5500.2323.

Rozemberczki B, Davies R, Sarkar R, Sutton C. 2018. Gemsec: graph embedding with self clustering. arXiv preprint arXiv:1802.03997.

Rozemberczki B, Sarkar R. 2018. Fast sequence-based embedding with diffusion graphs. In: *International Workshop on Complex Networks.* Berlin: Springer International Publishing, 99–107.

Sahu SK, Christopoulou F, Miwa M, Ananiadou S. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence: Association for Computational Linguistics, 4309–4316.

Salha G, Hennequin R, Vazirgiannis M. 2020. Simple and effective graph autoencoders with one-hop linear models. arXiv preprint arXiv:2001.07614.

Salim A, Shiju S, Sumitra S. 2020. Design of multi-view graph embedding using multiple kernel learning. *Engineering Applications of Artificial Intelligence* **90**:103534 DOI 10.1016/j.engappai.2020.103534.

Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. 2009. The graph neural network model. *IEEE Transactions on Neural Networks* **20(1)**:61–80 DOI 10.1109/TNN.2008.2005605.

Sen P, Namata GM, Bilgic M, Getoor L, Gallagher B, Eliassi-Rad T. 2008. Collective classification in network data. *AI Magazine* **29(3)**:93–106 DOI 10.1609/aimag.v29i3.2157.

Serra A, Greco D, Tagliaferri R. 2015. Impact of different metrics on multi-view clustering. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. Piscataway: IEEE, 1–8.

Sevgili Ö, Panchenko A, Biemann C. 2019. Improving neural entity disambiguation with graph embeddings. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Florence: Association for Computational Linguistics, 315–322.

Shang J, Ma T, Xiao C, Sun J. 2019. Pre-training of graph augmented transformers for medication recommendation. arXiv preprint arXiv:1906.00346.

Shavitt Y, Tankel T. 2008. Hyperbolic embedding of internet graph for distance estimation and overlay construction. *IEEE/ACM Transactions on Networking* **16(1)**:25–36 DOI 10.1109/TNET.2007.899021.

Shaw B, Jebara T. 2009. Structure preserving embedding. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. New York: ACM, 937–944.

Shen Z, Zhang Y-H, Han K, Nandi AK, Honig B, Huang D-S. 2017. mirna-disease association prediction with collaborative matrix factorization. *Complexity* **2017(9)**:1–9 DOI 10.1155/2017/2498957.

Shervashidze N, Schweitzer P, Leeuwen EJv, Mehlhorn K, Borgwardt KM. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research* **12(Sep)**:2539–2561.

Shi J, Malik J. 2000. Normalized cuts and image segmentation. *Departmental Papers (CIS)* **22**:107.

Shi L, Zhang Y, Cheng J, Lu H. 2019. Skeleton-based action recognition with directed graph neural networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway: IEEE.

Shi M, Tang Y, Zhu X. 2020. Topology and content co-alignment graph convolutional learning. arXiv preprint arXiv:2003.12806.

Shi R, Liang T, Peng H, Jiang L, Dai Q. 2020. Heam: Heterogeneous network embedding with automatic meta-path construction. In: Li G, Shen HT, Yuan Y, Wang X, Liu H, Zhao X, eds. *Knowledge Science, Engineering and Management*. Cham: Springer International Publishing, 304–315.

Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. 2013. The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine* **30(3)**:83–98 DOI 10.1109/MSP.2012.2235192.

Si C, Chen W, Wang W, Wang L, Tan T. 2019. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway: IEEE.

Sinha A, Shen Z, Song Y, Ma H, Eide D, Hsu B-JP, Wang K. 2015. An overview of microsoft academic service (mas) and applications. In: *Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion*. New York: Association for Computing Machinery, 243–246.

Socher R, Chen D, Manning CD, Ng A. 2013. Reasoning with neural tensor networks for knowledge base completion. In: *Advances in Neural Information Processing Systems.* Cambridge: MIT press, 926–934.

Song L. 2018. Structure2vec: deep learning for security analytics over graphs. https://www.usenix.org/conference/scainet18/presentation/song.

Song W, Xiao Z, Wang Y, Charlin L, Zhang M, Tang J. 2019. Session-based social recommendation via dynamic graph attention networks. arXiv preprint arXiv:1902.09362.

Srinivas V, Mitra P. 2016. *Applications of link prediction, chapter 5.* Cham: Springer International Publishing, 57–61.

Stanovsky G, Gruhl D, Mendes P. 2017. Recognizing mentions of adverse drug reaction in social media using knowledge-infused recurrent models. In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers.* Valencia: Association for Computational Linguistics, 142–151.

Su C, Tong J, Zhu Y, Cui P, Wang F. 2020. Network embedding in biomedical data science. *Briefings in Bioinformatics* **21(1)**:182–197 DOI 10.1093/bib/bby117.

Sun F-Y, Hoffmann J, Tang J. 2019. Infograph: unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000.

Sun J, Bandyopadhyay B, Bashizade A, Liang J, Sadayappan P, Parthasarathy S. 2019. Atp: directed graph embedding with asymmetric transitivity preservation. *Proceedings of the AAAI Conference on Artificial Intelligence* **33**:265–272 DOI 10.1609/aaai.v33i01.3301265.

Sun L, Wang J, Yu PS, Li B. 2018a. Adversarial attack and defense on graph data: a survey. arXiv preprint arXiv:1812.10528.

Sun M, Tang J, Li H, Li B, Xiao C, Chen Y, Song D. 2018b. Data poisoning attack against unsupervised node embedding methods. arXiv preprint arXiv:1810.12881.

Sun X, Guo J, Ding X, Liu T. 2016. A general framework for content-enhanced network representation learning. arXiv preprint arXiv:1610.02906.

Tang J, Liu H. 2012. Unsupervised feature selection for linked social media data. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12.* New York: ACM, 904–912.

Tang J, Liu J, Zhang M, Mei Q. 2016a. Visualizing large-scale and high-dimensional data. *Proceedings of the 25th International Conference on World Wide Web* **25**:287–297.

Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q. 2015. Line: large-scale information network embedding. *Proceedings of the 24th IC on WWW* **24**:1067–1077.

Tang L, Liu H. 2009. Relational learning via latent social dimensions. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* **15**:817–826.

Tang L, Liu H. 2011. Leveraging social media networks for classification. *Data Mining and Knowledge Discovery* **23(3)**:447–478 DOI 10.1007/s10618-010-0210-x.

Tang M, Nie F, Jain R. 2016. Capped lp-norm graph embedding for photo clustering. In: *Proceedings of the 2016 ACM on Multimedia Conference.* New York: ACM, 431–435.

Tenenbaum JB, De Silva V, Langford JC. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* **290(5500)**:2319–2323 DOI 10.1126/science.290.5500.2319.

Teng X, Liu J. 2020. Atrributed graph embedding based on multiobjective evolutionary algorithm for overlapping community detection. In: *2020 IEEE Congress on Evolutionary Computation (CEC).* Piscataway: IEEE, 1–8.

Tian F, Gao B, Cui Q, Chen E, Liu T-Y. 2014. Learning deep representations for graph clustering. In: *AAAI.* 1293–1299.

Tian Y, Hankins RA, Patel JM. 2008. Efficient aggregation for graph summarization. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data.* New York: ACM, 567–580.

Toivonen H, Zhou F, Hartikainen A, Hinkka A. 2011. Compression of weighted graphs. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 965–973.

Trouillon T, Dance CR, Gaussier É, Welbl J, Riedel S, Bouchard G. 2017. Knowledge graph completion via complex tensor factorization. *Journal of Machine Learning Research* **18(1)**:4735–4772.

Trouillon T, Nickel M. 2017. Complex and holographic embeddings of knowledge graphs: a comparison. arXiv preprint arXiv:1707.01475.

Tsitsulin A, Mottin D, Karras P, Müller E. 2018. Verse: versatile graph embeddings from similarity measures. In: *Proceedings of the 2018 World Wide Web Conference.* 539–548.

Tsitsulin A, Munkhoeva M, Perozzi B. 2020. Just slaq when you approximate: accurate spectral distances for web-scale graphs. In: *Proceedings of the Web Conference 2020.* 2697–2703.

Tu C, Liu H, Liu Z, Sun M. 2017. Cane: context-aware network embedding for relation modeling. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* **1**:1722–1731.

Tu C, Zhang W, Liu Z, Sun M. 2016. Max-margin deepwalk: discriminative learning of network representation. In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), Volume 2016.* 3889–3895.

Vashishth S, Bhandari M, Yadav P, Rai P, Bhattacharyya C, Talukdar P. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence: Association for Computational Linguistics, 3308–3318.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. 2017. Attention is all you need. In: *Advances in Neural Information Processing Systems.* Cambridge: MIT press, 5998–6008.

Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.

Veyseh APB, Nguyen TH, Dou D. 2019. Graph based neural networks for event factuality prediction using syntactic and semantic structures. arXiv preprint arXiv:1907.03227.

Vishwanathan SVN, Schraudolph NN, Kondor R, Borgwardt KM. 2010. Graph kernels. *Journal of Machine Learning Research* **11(Apr)**:1201–1242.

Wang D, Cui P, Zhu W. 2016. Structural deep network embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference On Knowledge Discovery And Data Mining.* New York: ACM, 1225–1234.

Wang M. 2017. Predicting rich drug-drug interactions via biomedical knowledge graphs and text jointly embedding. arXiv preprint arXiv:1712.08875.

Wang P, Wu Q, Cao J, Shen C, Gao L, van den Hengel A. 2018. Neighbourhood watch: referring expression comprehension via language-guided graph attention networks. arXiv preprint arXiv:1812.04794.

Wang P, Xu B, Wu Y, Zhou X. 2015. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences* **58(1)**:1–38.

Wang Q, Mao Z, Wang B, Guo L. 2017a. Knowledge graph embedding: a survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* **29(12)**:2724–2743 DOI 10.1109/TKDE.2017.2754499.

Wang S, Huang E, Cairns J, Peng J, Wang L, Sinha S. 2019a. Identification of pathways associated with chemosensitivity through network embedding. *PLOS Computational Biology* **15(3)**: e1006864 DOI 10.1371/journal.pcbi.1006864.

Wang S, Qu M, Peng J. 2017b. Prosnet: Integrating homology with molecular networks for protein function prediction. In: *Pacific Symposium on Biocomputing 2017.* World Scientific, 27–38.

Wang S, Tang J, Aggarwal C, Chang Y, Liu H. 2017c. Signed network embedding in social media. In: *Proceedings of the 2017 SIAM International Conference on Data Mining.* SIAM, 327–335.

Wang X, Bo D, Shi C, Fan S, Ye Y, Yu PS. 2020a. A survey on heterogeneous graph embedding: methods, techniques, applications and sources. arXiv preprint arXiv:2011.14867.

Wang X, Cui P, Wang J, Pei J, Zhu W, Yang S. 2017d. Community preserving network embedding. In: *AAAI.* 203–209.

Wang Z, Ye X, Wang C, Cui J, Yu P. 2020b. Network embedding with completely-imbalanced labels. Epub ahead of print 3 February 2020. *IEEE Transactions on Knowledge and Data Engineering* DOI 10.1109/TKDE.2020.2971490.

Wang Z, Zhang J, Feng J, Chen Z. 2014. Knowledge graph embedding by translating on hyperplanes. In: *AAAI, Volume 14.* 1112–1119.

Wang Z, Zheng L, Li Y, Wang S. 2019b. Linkage based face clustering via graph convolution network. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Piscataway: IEEE.

Waradpande V, Kudenko D, Khosla M. 2020. Deep reinforcement learning with graph-based state representations. arXiv preprint arXiv:2004.13965.

Watts DJ, Strogatz SH. 1998. Collective dynamics of 'small-world' networks. *Nature* **393(6684)**:440–442 DOI 10.1038/30918.

Wei X, Xu L, Cao B, Yu PS. 2017. Cross view link prediction by learning noise-resilient representation consensus. In: *Proceedings of the 26th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 1611–1619.

Weng Z, Zhang W, Dou W. 2020. Adversarial attention-based variational graph autoencoder. *IEEE Access* **8**:152637–152645.

White S, Smyth P. 2005. A spectral clustering approach to finding communities in graphs. In: *Proceedings of the 2005 SIAM International Conference on Data Mining.* SIAM, 274–285.

Wilson RC, Hancock ER, Pekalska E, Duin RP. 2014. Spherical and hyperbolic embeddings of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36(11)**:2255–2269 DOI 10.1109/TPAMI.2014.2316836.

Wold S, Esbensen K, Geladi P. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* **2(1–3)**:37–52 DOI 10.1016/0169-7439(87)80084-9.

Wu C, Zhou Y, Tan L, Teng C. 2020. Link prediction based on graph embedding method in unweighted networks. In: *2020 39th Chinese Control Conference (CCC).* 736–741.

Wu Q, Zhang H, Gao X, He P, Weng P, Gao H, Chen G. 2019a. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. arXiv preprint arXiv:1903.10433.

Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. 2019b. A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596.

Xiaojin Z, Zoubin G. 2002. Learning from labeled and unlabeled data with label propagation. Tech. Rep., Technical Report CMU-CALD-02–107. Carnegie Mellon University. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.3864&rep=rep1&type=pdf.

Xie M, Yin H, Wang H, Xu F, Chen W, Wang S. 2016. Learning graph-based poi embedding for location-based recommendation. *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* **25**:15–24.

Xiong C, Power R, Callan J. 2017. Explicit semantic ranking for academic search via knowledge graph embedding. In: *Proceedings of the 26th International Conference on World Wide Web.* International World Wide Web Conferences Steering Committee, 1271–1279.

Xu L, Wei X, Cao J, Yu PS. 2017. Embedding of embedding (eoe): joint embedding for coupled heterogeneous networks. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining.* New York: ACM, 741–749.

Xu X, Yuruk N, Feng Z, Schweiger TA. 2007. Scan: a structural clustering algorithm for networks. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 824–833.

Xue Z, Li G, Wang S, Zhang C, Zhang W, Huang Q. 2015. Gomes: a group-aware multi-view fusion approach towards real-world image clustering. In: *2015 IEEE International Conference on Multimedia and Expo (ICME).* Piscataway: IEEE, 1–6.

Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M. 2008. Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* **24(13)**:i232–i240 DOI 10.1093/bioinformatics/btn162.

Yamanishi Y, Kotera M, Moriya Y, Sawada R, Kanehisa M, Goto S. 2014. Dinies: drug-target interaction network inference engine based on supervised analysis. *Nucleic Acids Research* **42(W1)**:W39–W45 DOI 10.1093/nar/gku337.

Yan B, Wang C. 2020. Graphae: adaptive embedding across graphs. In: *2020 IEEE 36th International Conference on Data Engineering (ICDE).* Piscataway: IEEE, 1958–1961.

Yan Z, Ge J, Wu Y, Li L, Li T. 2020. Automatic virtual network embedding: a deep reinforcement learning approach with graph convolutional networks. *IEEE Journal on Selected Areas in Communications* **38(6)**:1040–1057 DOI 10.1109/JSAC.2020.2986662.

Yanardag P, Vishwanathan S. 2015. Deep graph kernels. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 1365–1374.

Yang B, Yih W-t, He X, Gao J, Deng L. 2014. Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575.

Yang C, Liu Z, Zhao D, Sun M, Chang E. 2015. Network representation learning with rich text information. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence.* 2111–2117.

Yang L, Xiao Z, Jiang W, Wei Y, Hu Y, Wang H. 2020. Dynamic heterogeneous graph embedding using hierarchical attentions. In: Jose JM, Yilmaz E, Magalhães J, Castells P, Ferro N, Silva MJ, Martins F, eds. *Advances in Information Retrieval.* Cham: Springer International Publishing, 425–432.

Yang L, Zhan X, Chen D, Yan J, Loy CC, Lin D. 2019. Learning to cluster faces on an affinity graph. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Piscataway: IEEE.

Yang Y, Wang H. 2018. Multi-view clustering: a survey. *Big Data Mining and Analytics* **1(2)**:83–107 DOI 10.26599/BDMA.2018.9020003.

Yang Z, Cohen WW, Salakhutdinov R. 2016a. Revisiting semi-supervised learning with graph embeddings. arXiv preprint arXiv:1603.08861.

Yang Z, Tang J, Cohen WW. 2016b. Multi-modal bayesian embeddings for learning social knowledge graphs. In: *IJCAI*. 2287–2293.

Ye Y, Hou S, Chen L, Lei J, Wan W, Wang J, Xiong Q, Shao F. 2019. Out-of-sample node representation learning for heterogeneous graph in real-time android malware detection. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 4150–4156.

Ying R, Bourgeois D, You J, Zitnik M, Leskovec J. 2019. Gnn explainer: a tool for post-hoc explanation of graph neural networks. arXiv preprint arXiv:1903.03894.

Ying R, He R, Chen K, Eksombatchai P, Hamilton WL, Leskovec J. 2018a. Graph convolutional neural networks for web-scale recommender systems. arXiv preprint arXiv:1806.01973.

Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J. 2018b. Hierarchical graph representation learning with differentiable pooling. In: *Advances in Neural Information Processing Systems*. Cambridge: MIT press, 4805–4815.

You J, Ying R, Ren X, Hamilton WL, Leskovec J. 2018. Graphrnn: a deep generative model for graphs. arXiv preprint arXiv:1802.08773.

Yu W, Zheng C, Cheng W, Aggarwal CC, Song D, Zong B, Chen H, Wang W. 2018. Learning deep network representations with adversarially regularized autoencoders. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Cambridge: ACM, 2663–2671.

Yuan H, Ji S. 2020. Structpool: structured graph pooling via conditional random fields. In: *International Conference on Learning Representations.*.

Yuan S, Wu X, Xiang Y. 2017. Sne: signed network embedding. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Berlin: Springer, 183–195.

Zeng H, Zhou H, Srivastava A, Kannan R, Prasanna VK. 2019. Graphsaint: graph sampling based inductive learning method. arXiv preprint arXiv:1907.04931.

Zhai S, Zhang Z. 2015. Dropout training of matrix factorization and autoencoder for link prediction in sparse graphs. In: *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 451–459.

Zhang C, Ren M, Urtasun R. 2019. Graph hypernetworks for neural architecture search. In: *International Conference on Learning Representations. ICLR.*.

Zhang D, Dai X, Wang X, Wang Y, Davis LS. 2018a. MAN: moment alignment network for natural language moment retrieval via iterative graph adjustment. arXiv preprint arXiv:1812.00087.

Zhang D, Yin J, Zhu X, Zhang C. 2016a. Collective classification via discriminative matrix factorization on sparsely labeled networks. In: *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. New York: ACM, 1563–1572.

Zhang D, Yin J, Zhu X, Zhang C. 2016b. Homophily, structure, and content augmented network representation learning. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*. Piscataway: IEEE, 609–618.

Zhang H, Shang X, Luan H, Wang M, Chua T-S. 2017. Learning from collective intelligence: feature learning using social images and tags. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **13(1)**:1.

Zhang H, Shang X, Luan H, Yang Y, Chua T-S. 2015. Learning features from large-scale, noisy and social image-tag collection. In: *Proceedings of the 23rd ACM International Conference on Multimedia.* New York: ACM, 1079–1082.

Zhang H, Zheng T, Gao J, Miao C, Su L, Li Y, Ren K. 2019b. Towards data poisoning attack against knowledge graph embedding. arXiv preprint arXiv:1904.12052.

Zhang H, Zhou J, Li R. 2020. Enhanced unsupervised graph embedding via hierarchical graph convolution network. *Mathematical Problems in Engineering* **2020(6)**:1–9 DOI 10.1155/2020/5702519.

Zhang J, Shi X, Zhao S, King I. 2019c. STAR-GCN: stacked and reconstructed graph convolutional networks for recommender systems. arXiv preprint arXiv:1905.13129.

Zhang Q, Wang H. 2016. Not all links are created equal: an adaptive embedding approach for social personalized ranking. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval.* New York: ACM, 917–920.

Zhang W, Fang Y, Liu Z, Wu M, Zhang X. 2020a. Mg2vec: learning relationship-preserving heterogeneous graph representations via metagraph embedding. *IEEE Transactions on Knowledge and Data Engineering* 1. https://ieeexplore.ieee.org/abstract/document/9089251.

Zhang W, Shang R, Jiao L. 2020. Complex network graph embedding method based on shortest path and moea/d for community detection. *Applied Soft Computing* **97**:106764.

Zhang Y, Chen X. 2018. Explainable recommendation: a survey and new perspectives. arXiv preprint arXiv:1804.11192.

Zhang Y, Wang D, Zhang Y. 2019d. Neural IR meets graph embedding: a ranking model for product search. arXiv preprint arXiv:1901.08286.

Zhang Z, Bu J, Ester M, Zhang J, Yao C, Li Z, Wang C. 2020b. Learning temporal interaction graph embedding via coupled memory networks. In: *Proceedings of the Web Conference 2020, WWW '20.* New York: Association for Computing Machinery, 3049–3055.

Zhang Z, Cui P, Zhu W. 2018. Deep learning on graphs: a survey. arXiv preprint arXiv:1812.04202.

Zhao G, Li J, Wang L, Qian X, Fu Y. 2019. Graphseq2seq: graph-sequence-to-sequence for neural machine translation. https://openreview.net/forum?id=B1fA3oActQ.

Zhao X, Chang A, Sarma AD, Zheng H, Zhao BY. 2013. On the embeddability of random walk distances. *Proceedings of the VLDB Endowment* **6(14)**:1690–1701 DOI 10.14778/2556549.2556554.

Zhao Y, Liu Z, Sun M. 2015. Representation learning for measuring entity relatedness with rich information. In: *Twenty-Fourth International Joint Conference on Artificial Intelligence.* 1412–1418.

Zhao Z, Yang Q, Cai D, He X, Zhuang Y. 2016. Expert finding for community-based question answering via ranking metric network learning. In: *IJCAI.* 3000–3006.

Zheng D, Ma C, Wang M, Zhou J, Su Q, Song X, Gan Q, Zhang Z, Karypis G. 2020a. Distdgl: distributed graph neural network training for billion-scale graphs. arXiv preprint arXiv:2010.05337.

Zheng VW, Cavallari S, Cai H, Chang KC-C, Cambria E. 2016. From node embedding to community embedding. arXiv preprint arXiv:1610.09950.

Zheng W, Wang D, Song F. 2020. Opengraphgym: a parallel reinforcement learning framework for graph optimization problems. In: Krzhizhanovskaya VV, Závodszky G, Lees MH, Dongarra JJ, Sloot PMA, Brissos S, Teixeira J, eds. *Computational Science—ICCS 2020.* Cham: Springer International Publishing, 439–452.

Zheng X, Ding H, Mamitsuka H, Zhu S. 2013. Collaborative matrix factorization with multiple similarities for predicting drug-target interactions. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York: ACM, 1025–1033.

Zhong J, Li N, Kong W, Liu S, Li TH, Li G. 2019. Graph convolutional label noise cleaner: train a plug-and-play action classifier for anomaly detection. arXiv preprint arXiv:1903.07256.

Zhong J, Qiu H, Shi B. 2020. Dynamics-preserving graph embedding for community mining and network immunization. *Information-An International Interdisciplinary Journal* **11(5)**:250.

Zhou C, Liu Y, Liu X, Liu Z, Gao J. 2017. Scalable graph embedding for asymmetric proximity. In: *AAAI.* 2942–2948.

Zhou K, Michalak TP, Rahwan T, Waniek M, Vorobeychik Y. 2018. Adversarial link prediction in social networks. arXiv preprint arXiv:1809.08368.

Zhou S, Dai X, Chen H, Zhang W, Ren K, Tang R, He X, Yu Y. 2020. Interactive recommender system via knowledge graph-enhanced reinforcement learning. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20.* New York: Association for Computing Machinery, 179–188.

Zhou Y, Cheng H, Yu JX. 2009. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment* **2(1)**:718–729 DOI 10.14778/1687627.1687709.

Zhu H, Lin Y, Liu Z, Fu J, Chua T-S, Sun M. 2019. Graph neural networks with generated parameters for relation extraction. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* Florence: Association for Computational Linguistics, 1331–1339.

Zhu S, Li J, Peng H, Wang S, Yu PS, He L. 2020a. Adversarial directed graph embedding. arXiv preprint arXiv:2008.03667.

Zhu S, Yu K, Chi Y, Gong Y. 2007. Combining content and link for classification using matrix factorization. In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* New York: ACM, 487–494.

Zhu S, Zhou L, Pan S, Zhou C, Yan G, Wang B. 2020b. Gssnn: graph smoothing splines neural networks. In: *AAAI.* 7007–7014.

Zhu Y, Xu Y, Yu F, Liu Q, Wu S, Wang L. 2020c. Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131.

Zitnik M, Agrawal M, Leskovec J. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* **34(13)**:i457–i466 DOI 10.1093/bioinformatics/bty294.

Zitnik M, Leskovec J. 2017. Predicting multicellular function through multi-layer tissue networks. *Bioinformatics* **33(14)**:i190–i198 DOI 10.1093/bioinformatics/btx252.

Zitnik M, Zupan B. 2016. Collective pairwise classification for multi-way analysis of disease and drug data. In: *Biocomputing 2016: Proceedings of the Pacific Symposium.* World Scientific, 81–92.

Zong N, Kim H, Ngo V, Harismendy O. 2017. Deep mining heterogeneous networks of biomedical linked data to predict novel drug-target associations. *Bioinformatics* **33(15)**:2337–2344 DOI 10.1093/bioinformatics/btx160.

Zügner D, Akbarnejad A, Günnemann S. 2018. Adversarial attacks on classification models for graphs. arXiv preprint arXiv:1805.07984.