PAPER

Node-degree aware edge sampling mitigates inflated classification performance in biomedical random walk-based graph representation learning

Luca Cappelletti[1,*] Lauren Rekerle[2] Tommaso Fontana[1] Peter Hansen[2] Elena Casiraghi[1,3] Vida Ravanmehr[2] Christopher J Mungall[3] Jeremy Yang[4] Leonard Spranger[5] Guy Karlebach[2] J. Harry Caufield[3] Leigh Carmody[2] Ben Coleman[2,7] Tudor Oprea[4] Justin Reese[3] Giorgio Valentini[1,6] and Peter N Robinson[2,7,8,*]

[1]AnacletoLab, Dipartimento di Informatica, Università degli Studi di Milano, Via Celoria 18, 20133, Milano, Italy, [2]The Jackson Laboratory for Genomic Medicine, 10 Discovery Drive, 06032, CT, USA, [3]Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, 94710, CA, USA, [4]Department of Internal Medicine and UNM Comprehensive Cancer Center, UNM School of Medicine, Albuquerque, 87102, NM, USA, [5]Institute of Bioinformatics, Freie Universität Berlin, Arnimallee 3, 14195, Berlin, Germany, [6]ELLIS, European Laboratory for Learning and Intelligent Systems, Milan Unit, Italy, [7]Institute for Systems Genomics, University of Connecticut, Farmington, 06032, CT, USA and [8]Berlin Institute of Health, Charité – Universitätsmedizin Berlin, Charitéplatz 1, 10117, Berlin, Germany

[*]Corresponding author. peter.robinson@bih-charite.de

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Graph representation learning is a family of related approaches that learn low-dimensional vector representations of nodes and other graph elements called embeddings. Embeddings approximate characteristics of the graph and can be used for a variety of machine-learning tasks such as novel edge prediction. For many biomedical applications, partial knowledge exists about positive edges that represent relationships between pairs of entities, but little to no knowledge is available about negative edges that represent the explicit lack of a relationship between two nodes. For this reason, classification procedures are forced to assume that the vast majority of unlabeled edges are negative. Existing approaches to sampling negative edges for training and evaluating classifiers do so by uniformly sampling pairs of nodes. We show here that this sampling strategy typically leads to sets of positive and negative examples with imbalanced node degree distributions. Using representative heterogeneous biomedical knowledge graph and random walk-based graph machine learning, we show that this strategy substantially impacts classification performance. If users of graph machine-learning models apply the models to prioritize examples that are drawn from approximately the same distribution as the positive examples are, then performance of models as estimated in the validation phase may be artificially inflated. We present a degree-aware node sampling approach that mitigates this effect and is simple to implement.

Key words: Graph machine learning, knowledge graph, node2vec, graph representation learning

Introduction

Many problems in biology and medicine stand to benefit from machine learning (ML) approaches (Rajkomar et al., 2019). Biomedical data are often composed of entities from multiple different classes that are interconnected by different types of relation. Therefore, biological data are often represented computationally as knowledge graphs (KG), semantic networks that encode entities as nodes and relations between entities as edges. Typical ML tasks that leverage KGs involve node (entity) classification and prediction of novel relations between entities (edge prediction) (Nickel et al., 2016; Li et al., 2022). Graph machine learning methods have been applied to numerous biomedical classification tasks including protein function prediction, protein–protein interaction prediction and *in silico* drug discovery Muzio et al. (2021). Despite the great promise of ML in medicine, to date very few ML algorithms have contributed meaningfully to clinical care Deo (2015). One reason for this might be that published models not infrequently display methodological flaws or underlying biases (Zech et al., 2018; Wynants et al., 2020; Biderman and Scheirer, 2020; Roberts et al., 2021). It is therefore essential to understand

and ideally to mitigate sources of bias and error in ML in order to develop robust and accurate algorithms.

It was shown in 2011 that topological imbalances in biomedical KGs can result in densely-connected entities (i.e., high-degree nodes) being highly ranked no matter the context, suggesting that embedding models may be more influenced by node degree than by any biological information encoded within the relations (Gillis and Pavlidis, 2011; Bonner et al., 2022). Here, we show that the method by which negative edges are sampled for evaluation of results in the validation phase can contribute to this node-degree bias. We present an approach to mitigating the effect by node-degree aware sampling. We demonstrate our approach using two heterogeneous KGs.

Definitions

A **graph** $G = (\mathcal{V}, \mathcal{E})$ consists of nodes (a.k.a. vertices) $v \in \mathcal{V}$ and edges (a.k.a. links or relations) $e^r_{u,v} \in \mathcal{E}$ connecting nodes u and v via a relationship of type r.

A graph with a single node type and a single edge type is called **homogeneous**. For example, the protein-protein interaction graph described below is homogeneous because it has one type of node (a gene symbol that represents the proteins encoded by the gene) and one type of edge (an interaction between a pair of proteins). Graphs with two or more types of node, two or more types of edge, or both are called **heterogeneous**. For instance, the synthetic lethality graph described below is heterogeneous because it contains two types of edges, one for protein-protein interactions and another for synthetic lethality interactions.

A **knowledge graph** is a graph that uses nodes to represent real-world entities and edges to represent the relations between these entities.

A **random walk** is defined as an iterative walker's transition from its current node to a randomly selected neighbor starting at a given source node, s. In the experiments described here, we simulate random walks of a fixed path length $l = 128$.

With a **first-order random walk**, if the walker is at node n at step i, the next random step is chosen based on information solely from the immediate neighbors of node n. With a **second-order random walk**, the next random step is chosen based on information from the previous random walk step and the immediate neighbours of node n.

Graph representation learning (GRL) is a form of graph machine learning that applies various strategies to convert nodes, edges, or graphs into low-dimensional vectors called "embeddings" that preserve graph structural information and properties (Cai et al., 2018). Graph embeddings can be used to address downstream prediction tasks (Xu, 2021). In this work, we focus on random-walk based GRL methods that optimize node embeddings such that nodes have similar embeddings if they tend to co-occur on short random walks over the graph (Li et al., 2022; Hamilton et al., 2017).

Shallow embedding methods generate a vector representation for every node u that preserves the input graph structure information. The methods are called shallow to distinguish them from graph neural networks that can generate representations for any graph element by capturing both network structure and node attributes and metadata using deep learning techniques (Li et al., 2022). Numerous approaches have been developed to generate embeddings that reflect different aspects of graph structure (Perozzi et al., 2014; Grover and Leskovec, 2016; Tang et al., 2015; Mikolov et al., 2013b; Pennington et al., 2014).

The **Matthews correlation coefficient (MCC)** is calculated based on the counts of true-positive (TP), true-negative (TN), false-positive (FP) and false-negative (FN) classifications as follows:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

The MCC generates a high score only if the binary predictor was able to correctly predict the majority of positive data instances and the majority of negative data instances, with its values ranging from −1 for perfect misclassification to +1 for perfect classification. MCC=0 is the expected value for random classification (Chicco and Jurman, 2020).

Protein-Protein Associations

The STRING database is a comprehensive relational database of protein-protein associations (Szklarczyk et al., 2021). STRING (version 11.0) data for *H. sapiens* was used corresponding to `9606.protein.links.v11.5.txt.gz`. Associations were filtered to retain only those with a score of at least 700, and duplicate edges between node pairs were removed.

Synthetic lethal interaction data

We first analyzed data available in the supplemental data provided with ISLE (Lee et al., 2018) and files available from the SynLethDB resource (Guo et al., 2016) by comparing the curation with the original publications. To create the synthetic lethal interaction database (SLDB) resource, we manually reviewed publications cited in these resources and additional publications. The curated SLIs are available at https://github.com/monarch-initiative/syntheticLethalityNetwork. The tab-separated file (TSV) includes information about each pair of genes, the perturbations used for each gene, the assays used to measure synthetic lethality, a Cellosaurus id (Bairoch, 2018) (if applicable), and the PubMed identifier. To add additional information to SLDB, we integrated it with the STRING protein-protein interaction network by using the nodes (genes) in SLDB to also represent the proteins in STRING that the genes themselves encode.

For the experiments described in this work, we imported the SLDB resource directly using a utility function of GRAPE (Cappelletti et al., 2023). The SLDB graph was integrated with the STRING PPA graph. A Python script that implements the analysis is available in the project GitHub repository (runSli.py).

KG-IDG

The KG-IDG knowledge graph represents data from the NIH Common Fund's Illuminating the Druggable Genome (IDG) Consortium. The IDG aims to integrate current knowledge of proteins in order to study the function of specific understudied drug targets in three main druggable protein families: G-protein coupled receptors, ion channels and protein kinases. KG-IDG is intended to represent relationships between drugs, their protein targets, and disease. KG-IDG unifies structured data from 14 different sources concerning drugs, proteins, and diseases (Caufield et al., 2023b). For experiments in which we trained edge prediction models on KG-IDG, we used edges between biolink:ChemicalSubstance, biolink:ChemicalEntity, biolink:Drug nodes and biolink:Protein nodes for training and to test the performance of the edge prediction model.

Model	Epochs	Learning rate	Walk length	Window size	Max neighbors
DeepWalk CBOW	30	0.010	128	5	100
DeepWalk SkipGram	30	0.010	128	5	100
Walklets CBOW	30	0.010	128	4	100
Walklets SkipGram	30	0.010	128	4	100
First-order LINE	100	0.050	n/a	n/a	n/a
Second-order LINE	100	0.050	n/a	n/a	n/a
HOPE	n/a	n/a	n/a	n/a	n/a

Table 1. Parameters for the learning models used in this project. Each of the models shown here was run using uniform and node-based sampling of negative examples. In addition to the parameters shown in the table, some parameters are only relevant to a subset of models: `learning rate decay` was set to 0.9. `avoid false negatives` was set to false for First-order LINE and Second-order LINE. The `number of negative samples` was set to 10 for DeepWalk CBOW, DeepWalk SkipGram, Walklets CBOW, and Walklets SkipGram. `Iterations` was set to 100 for all models except First-order and second-order LINE.

Shallow Graph Representation Learning

The shallow graph representation learning experiments described here were performed using the GRAPE library for fast and scalable Graph Processing and Embedding, version 0.1.28. GRAPE provides a comprehensive library of Graph Representation Learning and inference models implemented in Rust with a Python interface (Cappelletti et al., 2023). Seven node embedding methods were used including four random-walk methods and three matrix factorization methods.

The random-walk methods combine methods for generating random walks with methods for sampling node contexts. The two random walk sampling mechanisms are DeepWalk (Perozzi et al., 2014), which samples first-order random walks, and Walklets (Perozzi et al., 2017), which samples first-order random walks and, for a given central node v, at each i-th sampling iteration, skips i nodes around the central node v. We used these training samples obtained with both DeepWalk and Walklets to train two different embedding models. The CBOW (Mikolov et al., 2013a) model, trains a shallow neural network to predict the central node of a random walk window given the remainder contextual nodes. The second model is SkipGram (Mikolov et al., 2013a), which analogously to CBOW trains a shallow neural network to predict the contextual nodes given the central node. In all of these models, the node embedding matrix is the (trained) weight matrix of the first hidden layer.

The matrix factorization methods included Large-scale Information Network Embedding (LINE) and High-Order Proximity preserved Embedding (HOPE). First and second-order LINE (Tang et al., 2015) trains a neural network with either one layer (first-order) or two layers (second-order) to predict whether a given tuple of nodes defines an existing edge. HOPE starts by computing a node-proximity matrix, where the proximity between two nodes may be defined in different ways, in our case by using the number of common neighbors. Then HOPE computes the singular vectors corresponding to the k most significant singular values of the proximity matrix and uses the left and right product of the singular values with the singular vectors as the embeddings of the source and destination nodes (Ou et al., 2016).

Each of the seven algorithms was used with GRAPE default parameters. Details are provided in Table 1.

Edge prediction

Edge embeddings were formed from the embeddings of the corresponding pair of nodes (u, v) by the binary Hadamard ($\boxdot$) operator, defined as the elementwise product of both vectors, i.e.,

$$[f(u) \boxdot f(v)]_i = f_i(u) \times f_i(v)$$

Edge prediction was performed by a Perceptron model. Positive edges for training and evaluation were derived from the SLDB KG, and negative edges were obtained as will be described in the Results.

Results

Here, we investigate the influence of negative-edge sampling in link prediction by graph ML. The relevant algorithms have three main stages, each of which uses a different type of negative sampling (Figure 1). As we will explain below, the sampling strategy that has been traditionally used in the third stage can artificially inflate the measured classification performance for new data that have a distribution similar to that of the positive training set. In this work, we present a degree-aware node sampling approach for sampling negative edge examples that mitigates this effect. Before we discuss our approach, we will present a brief explanation of the three phases.

In the **first (embedding) phase**, embeddings are generated from the input graph. The embedding procedure is a generalization of the word embedding procedure of the word2vec algorithm (Mikolov et al., 2013b), whereby each random walk is like a sentence and the nodes of the graph are like words. Much previous work has been invested in understanding the effect of negative sampling in the first phase (training the embedding model). The computational objective of the skipgram model is to maximize the mean log-probability of context words that occur in a window surrounding the input word. For instance, the node2vec algorithm scans over series of nodes encountered in random walks and attempts to predict nearby nodes (i.e., inside some context window) on the basis of a Skip-gram objective function.

However, the per-node partition function is expensive to compute for large networks since it involves every node of the graph, and so node2vec approximates it using negative sampling, whereby k negative nodes are sampled for each positive node according to the unigram distribution U (w) raised to the 3/4rd power (Grover and Leskovec, 2016; Mikolov et al., 2013b; Ahrabian et al., 2020; Yang et al., 2020; Wang et al., 2023). A number of methods have been developed to improve the selection of negative examples for creating embeddings (Armandpour et al., 2019; Zhang and Zweigenbaum, 2018) but there seems to be no single method that performs best for all datasets (Caselles-Dupré et al., 2018; Yang et al., 2020). The first phase concludes with the generation of edge embeddings from the node embeddings, which can be performed with several methods (Grover and Leskovec, 2016).

The method that we present in this work applies only to the second and third phases and is independent of this kind of negative sampling chosen for the first phase.

In the **second (classification) phase**, positive and labeled edge examples are extracted from the KG for training an edge classifier. For example, to classify protein-protein associations, positive edges can be obtained from the STRING knowledge base, and for the synthetic lethality graph, positive synthetic lethality interactions can be curated from the literature. In general, only a small proportion of all potential edges are labeled positive. For instance, the STRING graph has 16,812 nodes, corresponding to (16812 × 16811/2) potential edges between pairs of nodes, or roughly 141 million edges. Only 252,953 edges, or 0.18%, are labeled positive. To train machine-learning classifiers, it is recommended to use numbers of positive and negative examples that do not greatly differ. STRING does not include information about pairs of proteins that do not undergo interactions. For this reason, negative examples are sampled at random from the unlabeled set.

Many classification algorithms are suitable for the second phase. Figure 1 shows a Random Forest classifier. For the experiments shown here, we used a single layer neural network (i.e., Perceptron) for edge classification.

In the **third (evaluation) phase**, the performance of the classifier is evaluated. Similar to the second phase, a relatively balanced number of positive and negative examples are chosen for the evaluation.

The current work explores the consequences of two strategies for choosing negative examples in the third (evaluation) phase.

Input knowledge graph

We examined a heterogeneous graph of protein-protein-associations (PPAs) derived from the STRING resource (Szklarczyk et al., 2021) together with 2445 synthetic lethal interactions derived from the literature (SLDB; Methods). The classification task in the heterogeneous SLDB graph was to predict novel synthetic lethal interactions. The SLDB graph displayed a skewed node distribution. For instance, the mean degree of the top 20 nodes was 105.9, compared to a mean degree in the entire graph of 2.8. 2081 of the 2445 edges (85.1%) in the largest component of the SLDB graph involved at least one of the top 20 nodes (Figure 2 and Supplemental Figures S1-S2).

Two methods for sampling negative examples from graphs: UNS and DANS

The evaluation phase requires negative sampling to measure the generalization performance of the edge-prediction model.

A common approach for obtaining negative examples samples the source and destination nodes from a uniform distribution that randomly chooses an integer between 1 and $|\mathcal{V}|$ corresponding to the nodes of the graph (Figure 3A). We reasoned that this sampling strategy, which we term *edge sampling by uniform node sampling* (UNS) will produce negative examples whose node degree approximates the degree distribution of the entire KG but may differ from the node degree distribution of the positive examples in many relevant biomedical KGs, because typical biomedical KGs are generally characterized by a non-uniform node-degree distribution (Lima-Mendez and van Helden, 2009).

We therefore developed a different sampling mechanism that assigns a number of negative edges to each node proportional to its node degree. We term this method *edge sampling by degree-aware node sampling* (DANS). In this approach, we randomly sample two edges $e_1 = (s_1, d_1), e_2 = (s_2, d_2)$ from $\mathcal{U}\{1, |E|\}$, and build a new negative edge by connecting the source node of e_1 and the destination node of e_2 (Figure 3B).

In real-world graphs, there is a minimal likelihood of collisions between existent and non-existent edges using either the UNS or the DANS sampling strategy that can be trivially addressed by repeating the sampling.

To illustrate the effect of the two sampling strategies, we plot the distribution of the product of the node degrees of the two nodes forming edges chosen by the UNS and DANS. In this context, the product of node degrees has been referred to as "preferential node attachment ($\mathcal{PA}$)" (Zhou et al., 2009); that is, if the node degree of node u is $d(u)$, then the preferential attachment of edge (u, v) is calculated as $\mathcal{PA} = d(u) \times d(v)$. We sampled 100 times the number of positive edges in the SLI graph (100 × 2445) and plotted the distribution of $\mathcal{PA}$ for the UNS and DANS sampling approaches as well as for the original (positive) edges. It can be seen that the distribution of edges follow a strikingly different distribution compared to UNS or DANS, whereby the DANS distribution more closely resembles the distribution of the positive edges (Figure 3C). Indeed, DANS generates negative edges by randomly extracting edges according to a uniform distribution. In this way, source and destination nodes of the negative edges tend to have degrees similar to that of the nodes involved in positive edges, thus resulting in comparable $\mathcal{PA}$ distributions between positive SLI edges and negative edges randomly drawn according to DANS.

To investigate the influence of UNS and DANS sampling on a simple classification task, we trained a perceptron (a single layer neural network) to classify SLI interactions using only features derived from node degrees, without taking any other graph features into account. For each edge, we formed a two-dimension integer vector with the degree of each of the nodes that made up the edge. We then compared the results of classification whereby we used UNS and DANS both for the selection of negative edges to train the perceptron (the second phase) with UNS or DANS sampling for the evaluation (third) phase. An equal number of positive and negative examples was chosen. In each experiment, we performed ten-fold cross validation with training size of 0.75.

We present results of classification in terms of the Matthews correlation coefficient (MCC), which ranges from -1 for perfect misclassification to +1 for perfect classification, while MCC=0 is the expected value for a random classifier (Chicco and Jurman, 2020). Additional results are presented in Supplemental Figures S2-S5 for four other edge feature generation methods: Adamic-Adar index, Jaccard coefficient, Resource Allocation Index, and Preferential Attachment (Adamic and Adar, 2003; Zhou et al., 2009).

The results of this analysis demonstrate that node-degree bias operates in at least two phases in graph machine learning: in the phase in which the classifier is trained, and also in the phase in which the results of classification are evaluated. The classification models were created identically with the sole exception of the method for choosing negative examples, and yet the measured classification performance differs substantially. To our knowledge, our analysis is the first to show the effects of different sampling strategies on the evaluation phase. If the user of the model is interested in new examples drawn from the same distribution as the known

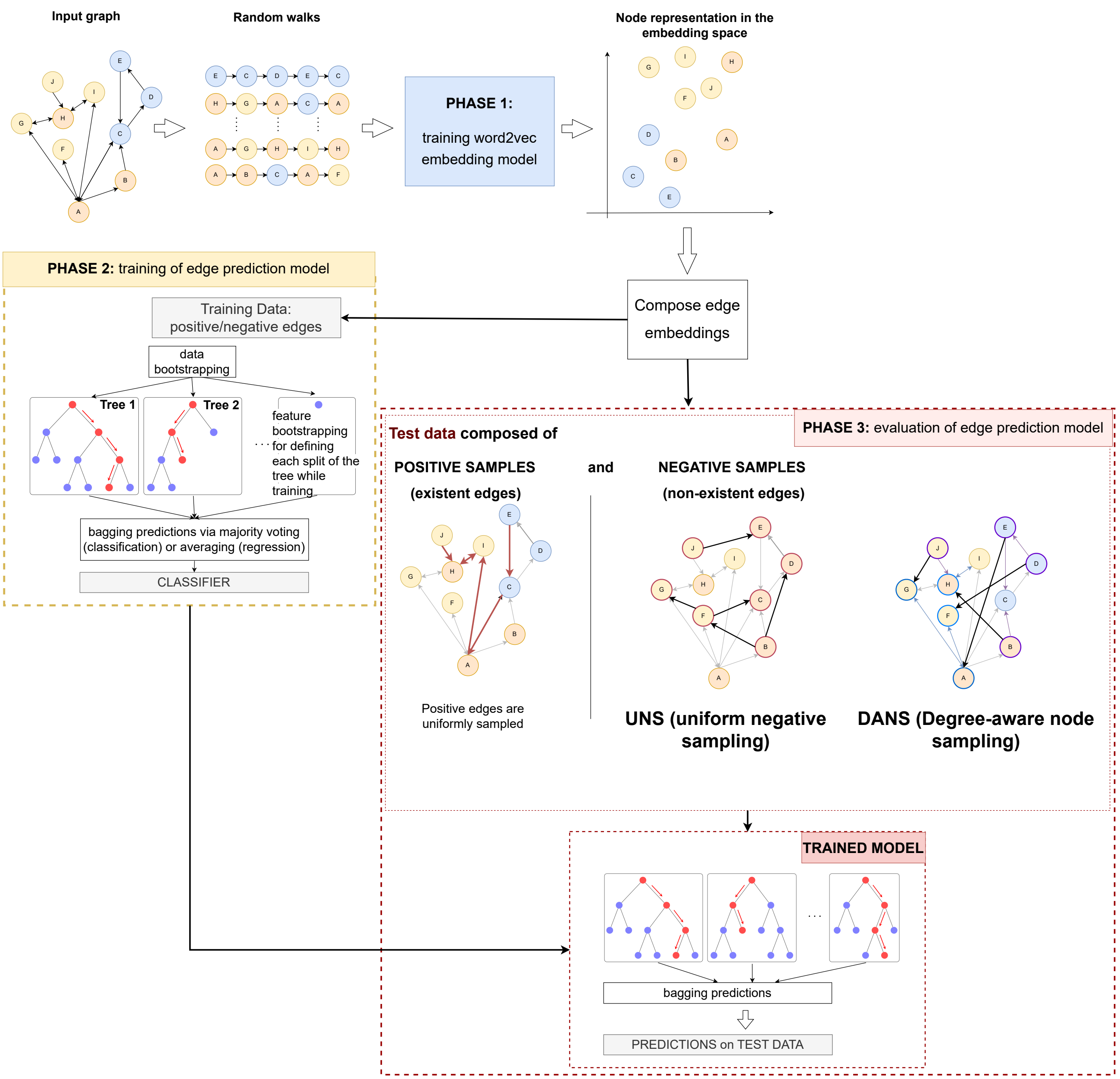
Input graph
Random walks
PHASE 1:
training word2vec embedding model
Node representation in the embedding space
Compose edge embeddings
PHASE 2: training of edge prediction model
Training Data: positive/negative edges
data bootstrapping
Tree 1
Tree 2
feature bootstrapping for defining each split of the tree while training
bagging predictions via majority voting (classification) or averaging (regression)
CLASSIFIER
PHASE 3: evaluation of edge prediction model
Test data composed of
POSITIVE SAMPLES (existent edges)
and
NEGATIVE SAMPLES (non-existent edges)
Positive edges are uniformly sampled
UNS (uniform negative sampling)
DANS (Degree-aware node sampling)
TRAINED MODEL
bagging predictions
PREDICTIONS on TEST DATA
A
B
C
D
E
F
G
H
I
J

Fig. 1. Graph representational learning and link prediction. There are three phases in which negative examples are sampled. In the first, embeddings are generated from the input graph using a variety of methods that sample the graph by random walks or related procedures. Edge embeddings are then generated from the node embeddings. In the second phase, a classifier (such as a random forest) is created using the positive and negative labels from a training set. In the third phase, the resulting classifier is *evaluated* on the basis of its performance on held-out data. In typical bioinformatics applications, we have knowledge about a subset of positive examples, and assume all unlabeled examples are negative. In the current work, we explore two strategies (**UNS** and **DANS**) for sampling from unlabeled examples to obtain negative examples for evaluation.

positive examples, then our experiments suggest that at least part of the signal obtained by the classifier using UNS sampling is spurious.

The influence of negative sampling strategies on GRL edge prediction

We then asked if a similar effect pertains to random walk-based GRL edge prediction. We applied seven different embedding approaches followed by perceptron-based classification of edges in the SLDB graph (Methods).

We tested the classification performance for the prediction of synthetic lethal interaction edges. The measured classification performance was consistently higher for UNS sampling than for DANS sampling. For instance, Walklets SkipGram displayed an MCC of 0.49 for UNS sampling but only 0.26 for DANS sampling (Figure 5; see also Supplemental Table S2 for AUROC, AUPRC, and F1 score analysis). For each comparison, the only difference was in the way the negative examples were selected.

The difference in the influence of negative sampling strategies on the SLDB graph is related to the different node

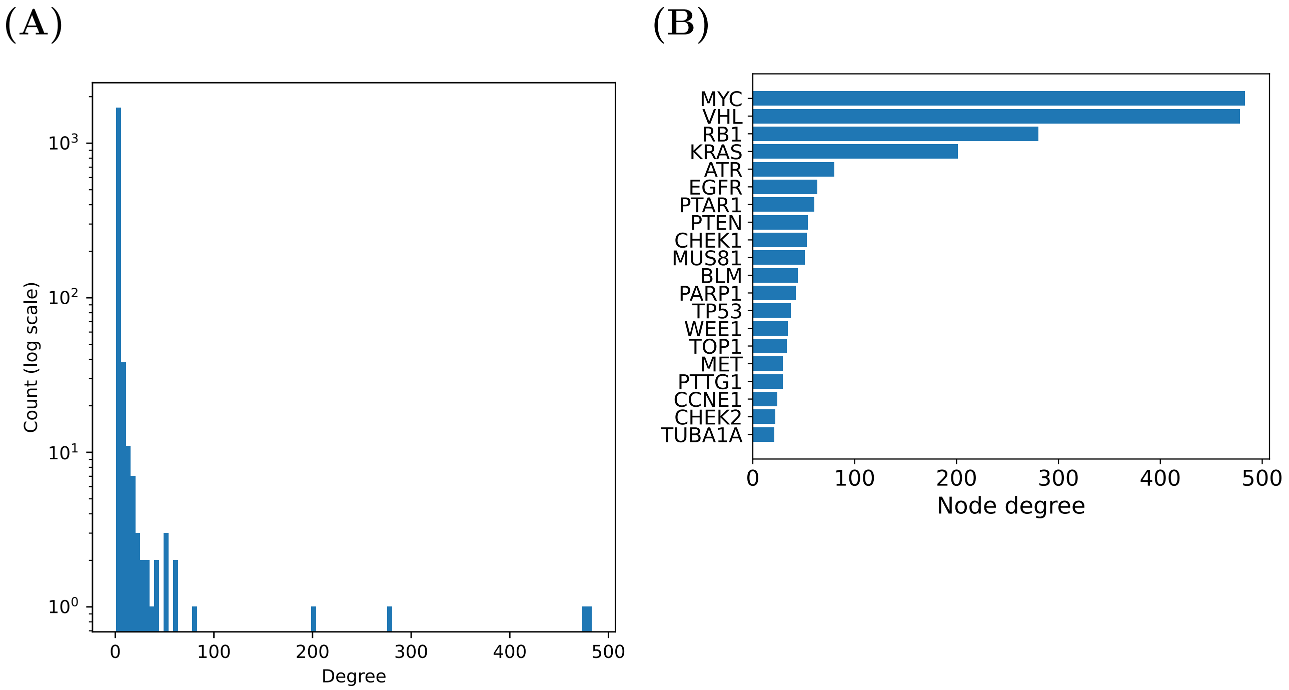

Fig. 2. Topology of the SLDB graph. A) Degree distribution of the synthetic lethality interaction edges in the SLDB graph. **B)** The 20 highest-degree nodes with synthetic lethality interactions of the SLDB network. The distribution of node degrees among the 20 most densely connected nodes of the SLI network.

A

```
Function uniform node sampling($\mathcal{G} = (V, E)$)
    $s \leftarrow \mathcal{U}\{1, |V|\}$
    $d \leftarrow \mathcal{U}\{1, |V|\}$
    return $(s, d)$
```

B

```
Function degree-aware node sampling($\mathcal{G} = (V, E)$)
    $e_1 = (s_1, d_1) \leftarrow \mathcal{U}\{1, |E|\}$
    $e_2 = (s_2, d_2) \leftarrow \mathcal{U}\{1, |E|\}$
    return $(s_1, d_2)$
```

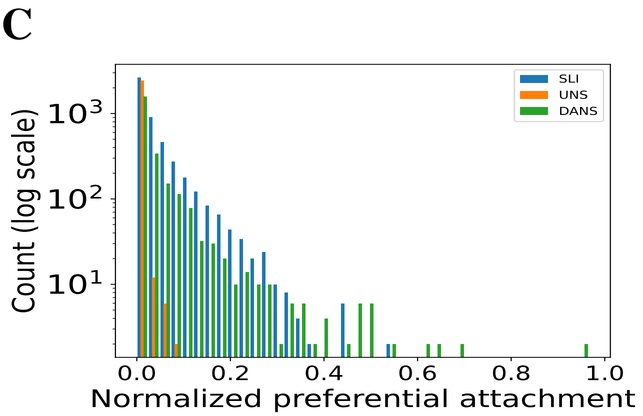

Fig. 3. Pseudocode for edge sampling strategies. Both strategies sample two nodes and return the edge that connects the two nodes, but the procedure used for sampling the nodes differs. **A. Uniform node sampling (UNS)**. The standard method for sampling samples two nodes uniformly to create a "random" edge for negative examples. **B. Degree-aware node sampling (DANS)**. The method presented here instead samples two edges uniformly to create a random edge from the source node of the first edge and the destination node of the second edge. **C. Preferential attachment** ($\mathcal{PA}$). The plot shows $\mathcal{PA}$ for positive examples from the SLI graph and edges sampling using the UNS and DANS approaches. See text for details. A Jupyter notebook that performs this analysis and generates panel C of this Figure is available on the project GitHub repository.

distributions or other differences in graph structure. Indeed, the SLDB graph shows a degree distribution that is approximately scale-free, with few nodes of very high degree and many low-degree nodes (Supplemental Figure S1-S2).

To confirm this result, we repeated this experiment on KG-IDG, a knowledge graph that integrates data related to drug repurposing (Caufield et al., 2023a). As before, we produced node embeddings using seven different approaches, and trained a perceptron-based edge prediction model. We then measured the performance of this model in predicting drug to protein edges from this graph. As with SLDB, the classification performance was higher for UNS compared to DANS. For example, the MCC of Walklets SkipGram was 0.90 using UNS sampling but only 0.17 using DANS sampling (Figure 6). The results from the other five node embeddings strategies were similar: in each case, the MCC using DANS sampling was lower than when using UNS sampling.

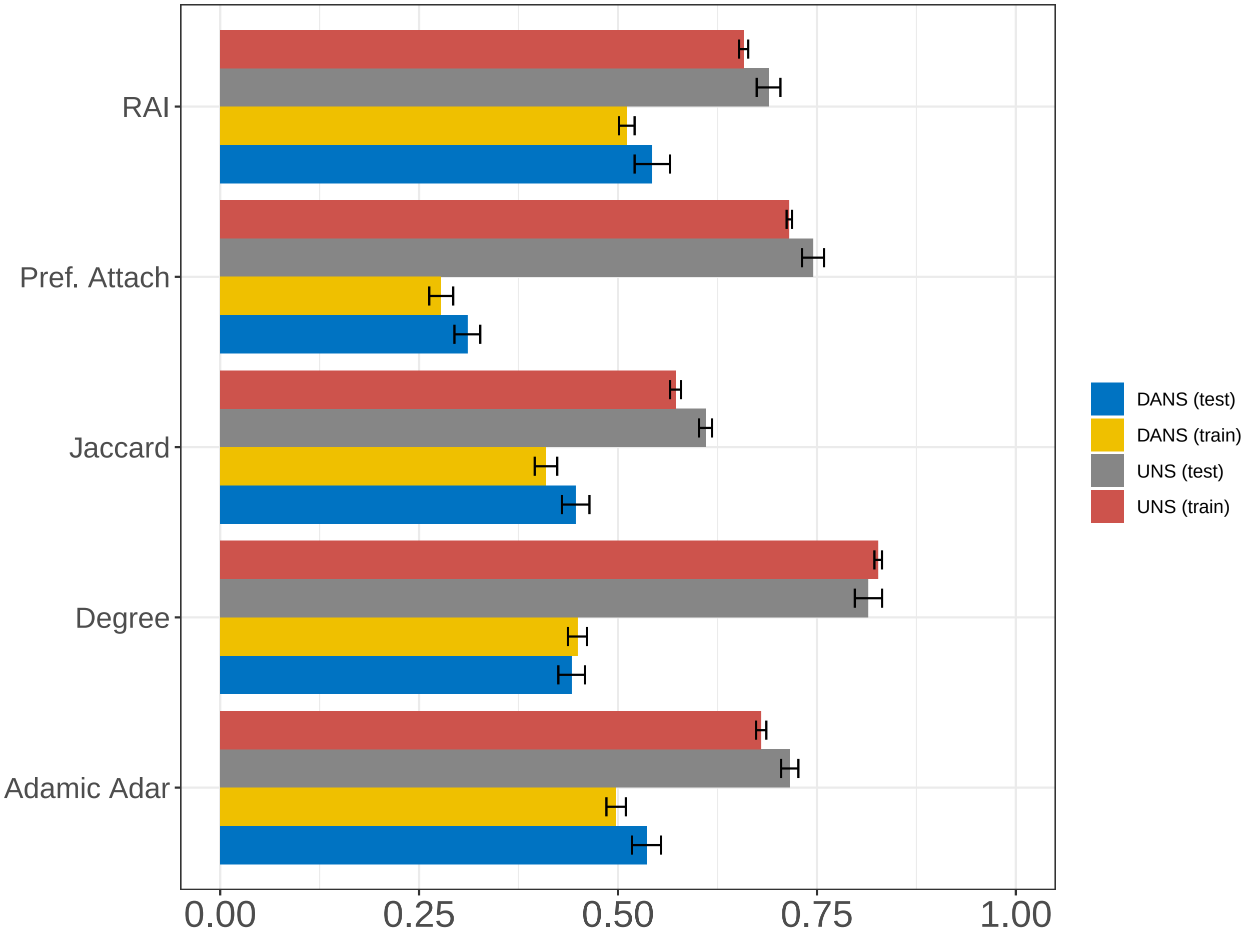

Fig. 4. Effects of UNS and DANS sampling on measured classification performance using two-dimensional degree-based features. A perceptron model was trained to predict novel SLI edges. DANS or UNS sampling was used in the evaluation phase. Abbreviations: RAI - Resource Allocation Index; Pref. Attach - Preferential Attachment; Jaccard - Jaccard Coefficient. The X-axis shows the Matthews Correlation Coefficient (MCC).

Discussion

Large graphs, including many graphs of interest for biomedical research, commonly follow an approximately scale-free power-law distribution, meaning that the probability $P(k)$ that a node in the network interacts with k other vertices decays as a power-law, following $P(k) \sim k^{-\gamma}$. This function indicates a high diversity of node degrees, with the lack of a typical degree in the graph motivating the characterization of these graphs as scale-free (Barabasi and Albert, 1999; Albert, 2005). Intuitively, scale-free networks contain a few hubs that are connected to many other nodes and many nodes with one or only a few connections. It has been known since 2011 that node degree can skew machine-learning predictions in biological graphs (Gillis and Pavlidis, 2011). In this work, we characterize the specific influence of negative sampling in generating such biases.

Constructing generalizable graph models of biomedical domains is challenging because most systems of scientific interest have multiple different classes of nodes and relations, and in many cases, our knowledge is so incomplete that comprehensive gold-standard data sets for training ML classifiers do not exist. An ML algorithm is said to be biased if its results are systematically wrong due to incorrect assumptions of the ML process. Biases can inflate the measured prediction performance of algorithms (Eid et al., 2021). In this work we have explored the relationship between sampling of negative examples and measured performance of shallow graph representation learners. Our results demonstrate that if the positive and negative samples have a different node degree distribution, then strategies for sampling negative examples for the evaluation that do not take node degree into account can greatly affect the estimate performance of classification algorithms.

While the classification performance of a given ML model depends on several crucial factors, including the amount of information in the training dataset, the training algorithm and its parameter settings, it is important to realize that the validity and reliability of the model assessment strongly depends on the distribution of the novel data the model is meant to classify. We note that the effect we have described here is not the same as overfitting. Rather, what we have shown is essentially that standard (UNS-based) approaches to evaluating classifiers in biomedical KGs risk comparing apples and oranges because the distribution of the positive examples in model training and evaluation is different. While the results that are estimated using UNS seem to be be accurate, they are derived from an artificially "easy" classification task that reflects differences in node degree between positive edges (typically

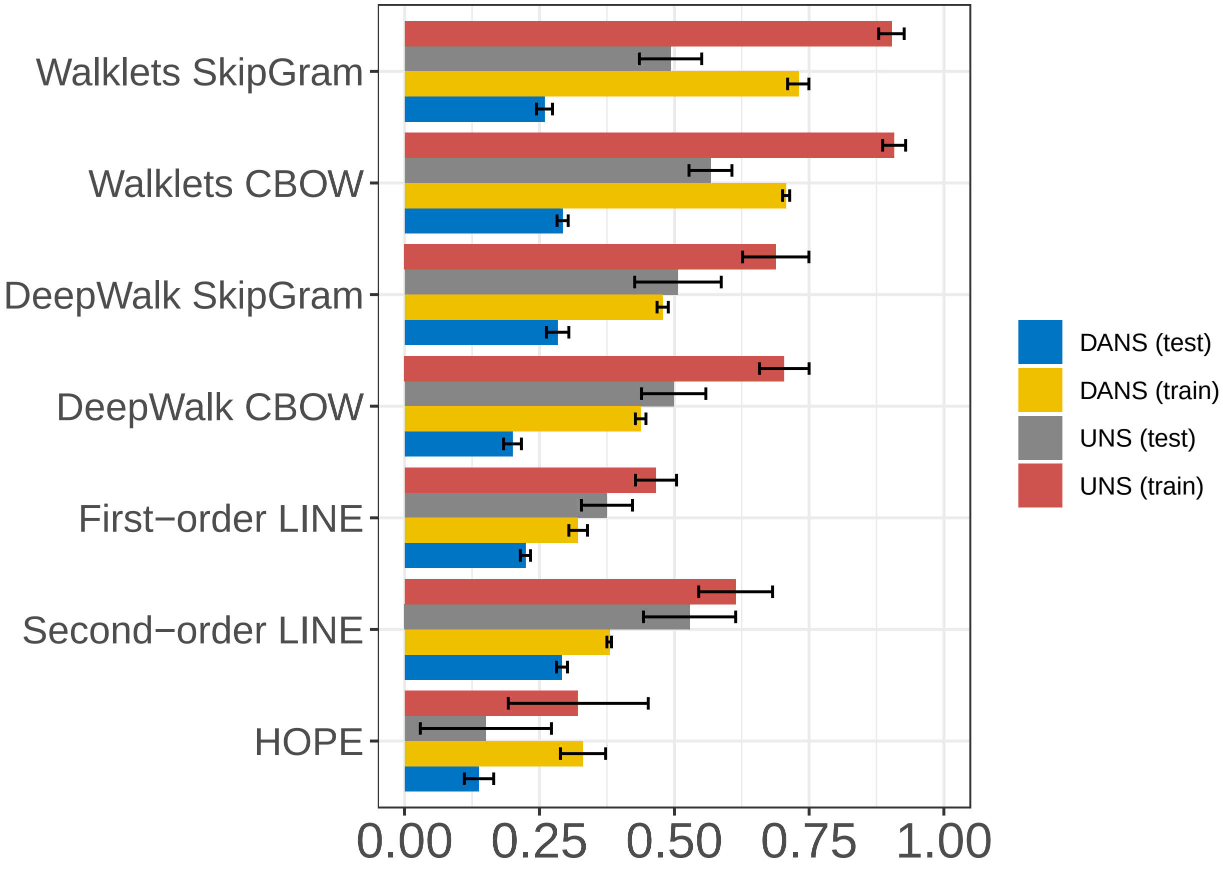

Fig. 5. Matthews correlation coefficient for the seven methods applied to edge prediction in the heterogeneous SLDBgraph. The bars show mean ± standard deviation. The X-axis shows the MCC.

high degree) and uniformly chosen edges (typically low degree). We argue that in many cases this does not reflect the actual biomedical problem, which is to identify novel edges of high degree nodes in an existing KG. We have shown that the DANS approach provides a more relevant estimation of performance of ML classifiers in this situation. For instance, if a biologist wants to find new SLI interactions that involve one of the genes shown in Figure 2C, the node degree of such candidates will be more similar to the node degree sampled by DANS than that from UNS.

Our review of other software packages for RW-GRL revealed that the uniform node sampling procedure is used during training by the gensim package (Řehůřek and Sojka, 2010) that is widely used as to develop algorithms to perform embedding such as node2vec (Grover and Leskovec, 2016). The methods used for sampling negative examples are rarely described in detail in the methods of many publications on the topic. Our results suggest that users should take this into account and report on the approach to negative sampling.

Limitations of our analysis include the restriction to random-walk-based GRL. We have not performed exhaustive parameter optimization; however, the parameters used in the results presented here are typical for the algorithms, and the effects described in this work do not pertain to the model training phase but instead to evaluation. It might be the case that some models, datasets, or parameter combinations obtain similar estimates of prediction accuracies when UNS and DANS are used. For instance, we observed no substantial difference in UNS and DANS inflated results, when the HOPE model is used for prediction of SLIs. However, we observe substantial differences for the vast majority of models tested. Therefore, we recommend that practitioners of random-walk based graph representation learning always investigate performance of their models using both UNS and DANS sampling to characterize potential differences. Further work will delve into the exploration of this issue for specific models and prediction tasks.

Conclusions

Negative samples should be meaningful for the classification task at hand, and inappropriate sampling mechanisms may lead to biased evaluation. Negative samples that differ substantially from the positive samples in one or more characteristics can lead to over-optimistic evaluations. Conversely, negative samples that are too similar to (or even collide with) the positive samples may lead to overly poor evaluations.

We recommend that practitioners of edge prediction in biological networks carefully evaluate the node degree distribution of samples chosen for the negative and positive examples. Results of training with the standard uniform node selection schema and the node-degree selection approach presented here should be compared.

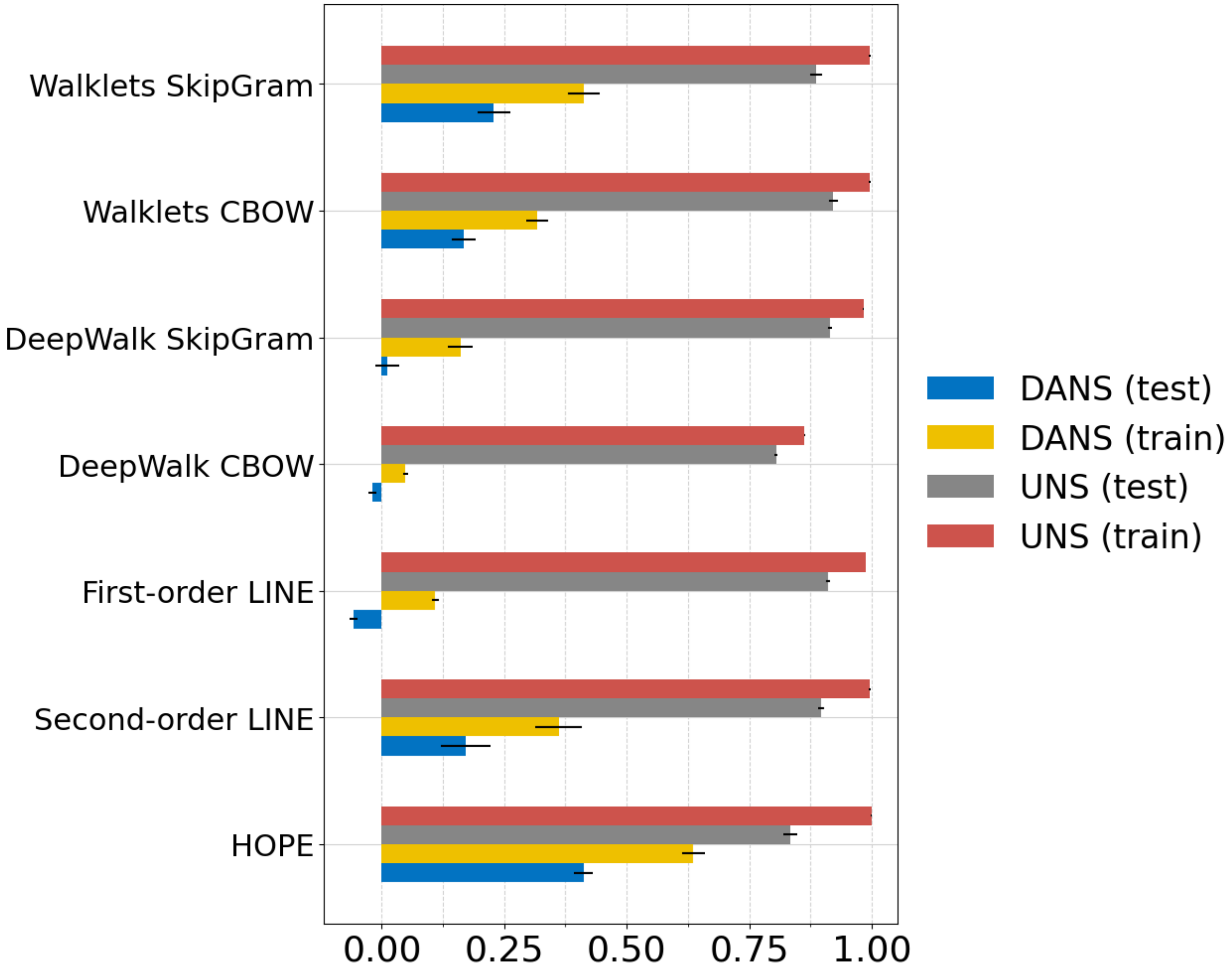

Fig. 6. Matthews correlation coefficient for the methods applied to edge prediction in the heterogeneous KG-IDG graph. The bars show mean ± standard deviation. The X-axis shows the MCC.

Competing Interests

The authors declare no competing financial or non-financial interests.

Code availability

The scripts used to generate the results described in this manuscript are available together with several illustrative Jupyter notebooks at https://github.com/monarch-initiative/negativeExampleSelection under the GNU General Public License version 3.

Data availability

No primary data was generated for this work. The graph data investigated here can be imported using the functions demonstrated in the illustrative Jupyter notebooks at https://github.com/monarch-initiative/negativeExampleSelection.

Author Contributions statement

L.Cap. designed the negative sampling algorithm. L.Cap., L.R., T.F., P.H., V.R., J.R. and P.N.R. performed the experiments. L.R., V.R., J.H.C., G.K., J.R., J.Y. contributed software. L.Car., B.C., L.S., and P.N.R. performed the biocuration to create the SLDB resource. T.O., E.C., C.J.M., J.R., G.V., and P.N.R. conceived the project, supervised all the experiments, and data analysis. L.Cap. and P.N.R. wrote the manuscript with assistance from all other authors.

Acknowledgments

This work was supported by the National Institutes of Health (NIH) [U01-CA239108-02] to P.N.R., C.J.M., and T.O.; Additional support was received from the National Cancer Institute (NCI) grant U24-CA224067 to P.N.R.

References

Adamic,L. A. and Adar,E. (2003) Friends and neighbors on the web. *Social Networks*, 25(3):211 – 230

Ahrabian,K. et al.(2020) Structure aware negative sampling in knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6093–6101,Association for Computational Linguistics.

Albert, R.(2005) Scale-free networks in cell biology. *J Cell Sci*, 118(Pt 21):4947–4957.

Armandpour,M. et al.(2019) Robust negative sampling for network embedding. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Bairoch,A.(2018) The Cellosaurus, a Cell-Line Knowledge Resource. *J Biomol Tech*, 29(2):25–38.

Barabasi,A. L. and Albert,R. et al.(1999) Emergence of scaling in random networks. *Science*, 286(5439):509–512.

Biderman,S. and Scheirer,W. J.(2020) Pitfalls in machine learning research: Reexamining the development cycle. *CoRR*, abs/2011.02832.

Bonner,S. et al.(2022) Implications of topological imbalance for representation learning on biomedical knowledge graphs. *Brief Bioinform*, 23(5):bbac279

Cai,H. et al.(2018) A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge & Data Engineering*, 30(09):1616–1637.

Cappelletti,L. et al.(2023) GRAPE for Fast and Scalable Graph Processing and Random Walk-based Embedding. *Nat. Comput. Sci.*, 3:552–568.

Caselles-Dupré,H. et al.(2018) Word2vec applied to recommendation: Hyperparameters matter. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 352–356, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450359016.

Caufield,J. H. et al.(2023a) KG-Hub—building and exchanging biological knowledge graphs. *Bioinformatics*, 39(7):btad418.

Caufield,J. H. et Al.(2023B) KG-Hub-building and exchanging biological knowledge graphs. *Bioinformatics*, 39(7):btad418.

Chicco,D. and Jurman,G. (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1):6.

DeO,R. C. (2015) Machine Learning in Medicine. *Circulation*, 132(20):1920–1930.

Eid, F. E. et al.(2021) Systematic auditing is essential to debiasing machine learning in biology. *Commun Biol*, 4(1):183.

Gillis,J. and Pavlidis,P. (2011) The impact of multifunctional genes on "guilt by association" analysis. *PLoS One*, 6(2):e17258.

Grover,A. and Leskovec,J. (2016) node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Guo,J. et al.(2016) SynLethDB: synthetic lethality database toward discovery of selective and sensitive anticancer drug targets. *Nucleic Acids Res*, 44(D1):D1011–1017.

Hamilton,W. L. et al.(2017) Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74.

Lee, J. S. et al.(2018) Harnessing synthetic lethality to predict the response to cancer treatment. *Nat Commun*, 9(1):2546.

Li, M. M. et al.(2022) Graph representation learning in biomedicine and healthcare. *Nat Biomed Eng*.

Lima-Mendez,G. and van Helden,J. (2009) The powerful law of the power law and other myths in network biology. *Mol Biosyst*, 5(12):1482–1493.

Mikolov, T. et al.(2013a) Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Mikolov, T. et al.(2013b) Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA.

Muzio,G. et al.(2021) Biological network analysis with deep learning. *Brief Bioinform*, 22(2):1515–1530.

Nickel,M. et al.(2016) A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Ou, M. et al.(2016) Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1105–1114, New York, NY, USA. Association for Computing Machinery.

Pennington, J. et al.(2014) Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perozzi, B. et al.(2014) Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710.

Perozzi, B. et al.(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*, ASONAM '17, page 258–265, New York, NY, USA. Association for Computing Machinery.

Rajkomar, A. et al.(2019) Machine Learning in Medicine. *N Engl J Med*, 380(14):1347–1358.

Řehůřek,R. and Sojka,P.(2010) Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Roberts, M. et al.(2021) Common pitfalls and recommendations for using machine learning to detect and prognosticate for COVID-19 using chest radiographs and CT scans. *Nature Machine Intelligence*, 3:199–217.

Szklarczyk,D. et al.(2021) The STRING database in 2021: customizable protein-protein networks, and functional characterization of user-uploaded gene/measurement sets. *Nucleic Acids Res*, 49(D1):D605–D612.

Tang, J. et al.(2015) LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1067–1077, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Wang, Y. et al.(2023) Adans: Adaptive negative sampling for unsupervised graph representation learning. *Pattern Recognition*, 136:109266.

L. Wynants, et al.(2020) Prediction models for diagnosis and prognosis of covid-19: systematic review and critical appraisal. *BMJ*, 369:m1328.

M. Xu(2021) Understanding graph embedding methods and their applications. *SIAM Review*, 63(4):825–853. doi: 10.1137/20M1386062.

Yang,Z. et al.(2020) Understanding negative sampling in graph representation learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1666–1676, New York, NY, USA. Association for Computing Machinery.

Zech, J. R. et al.(2018) Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Med*, 15(11): e1002683.

Zhang,Z. and Zweigenbaum,P.(2018) GNEG: Graph-based negative sampling for word2vec. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 566–571, Melbourne, Australia. Association for Computational Linguistics.

Zhou, T. et al.(2009) Predicting missing links via local information. *The European Physical Journal B*, 71(4): 623–630.