Revista Colombiana de Estadística - Applied Statistics
July 2025, volume 48, issue 2, pp. 115–137
http://doi.org/10.15446/rce.v48n2.117558

A Unified Approach to Link Prediction in Collaboration Networks

Un enfoque unificado para la predicción de enlaces en redes de colaboración

Juan Sosa[a], Diego Martínez[b], Nicolás Guerrero[c]

Department of Statistics, Faculty of Science, Universidad Nacional de Colombia, Bogotá, Colombia

Abstract

This article investigates and compares three approaches to link prediction in colaboration networks, namely, an ERGM (Exponential Random Graph Model; Robins et al. 2007), a GCN (Graph Convolutional Network; Kipf & Welling 2017), and a Word2Vec+MLP model (Word2Vec model combined with a multilayer neural network; Mikolov, Chen, Corrado & Dean 2013 and Goodfellow et al. 2016). The ERGM, grounded in statistical methods, is employed to capture general structural patterns within the network, while the GCN and Word2Vec+MLP models leverage deep learning techniques to learn adaptive structural representations of nodes and their relationships. The predictive performance of the models is assessed through extensive simulation exercises using cross-validation, with metrics based on the receiver operating characteristic curve. The results clearly show the superiority of machine learning approaches in link prediction, particularly in large networks, where traditional models such as ERGM exhibit limitations in scalability and the ability to capture inherent complexities. These findings highlight the potential benefits of integrating statistical modeling techniques with deep learning methods to analyze complex networks, providing a more robust and effective framework for future research in this field.

Key words: Collaboration networks; Exponential random graph model; Graph convolutional network; Word2Vec; Social networks analysis.

[a]Ph.D. E-mail: jcsosam@unal.edu.co
[b]Statistician. E-mail: damartinezsi@unal.edu.co
[c]Statistician. E-mail: niguerreroc@unal.edu.co.

Resumen

Este artículo investiga y compara tres enfoques para la predicción de enlaces en redes de colaboración: un ERGM (*Exponential Random Graph Model*; Robins et al., 2007), una GCN (*Graph Convolutional Network*; Kipf & Welling, 2017) y un modelo Word2Vec+MLP (modelo Word2Vec combinado con una red neuronal multicapa; Mikolov, Chen, Corrado & Dean (2013), y Goodfellow et al. (2016)). El ERGM, basado en métodos estadísticos, se emplea para capturar patrones estructurales generales dentro de la red, mientras que los modelos GCN y Word2Vec+MLP utilizan técnicas de aprendizaje profundo para aprender representaciones estructurales adaptativas de los nodos y sus relaciones. El desempeño predictivo de los modelos se evalúa mediante extensos ejercicios de simulación con validación cruzada, utilizando métricas basadas en la curva característica operativa del receptor (ROC). Los resultados muestran claramente la superioridad de los enfoques de aprendizaje automático en la predicción de enlaces, particularmente en redes grandes, donde los modelos tradicionales como el ERGM presentan limitaciones en escalabilidad y en la capacidad de capturar complejidades inherentes. Estos hallazgos resaltan los posibles beneficios de integrar técnicas de modelado estadístico con métodos de aprendizaje profundo para analizar redes complejas, proporcionando un marco más robusto y efectivo para futuras investigaciones en este campo.

***Palabras clave*:** Redes de colaboración; Modelo exponencial de grafos aleatorios; Red de convolución sobre grafos; Word2Vec; Análisis de redes sociales.

1. Introduction

Social networks have captured the attention of researchers since the last century. Studying networks is essential, as we live in a connected world where understanding connections provides deeper insights into numerous phenomena (Kolaczyk & Csárdi, 2020). A specific area of interest within this field is collaboration networks. While some authors focus on studying these networks as relationships formed between actors who interact locally, resulting in biased and random components (Skvoretz, 1990), we consider a collaboration network as a fully structured complex system in which two scientists are connected if they have coauthored an article. This definition is reasonable, as most people who have written an article together have shared ideas over a certain period (Newman, 2001). Some researchers have analyzed collaboration graphs for scientists across various fields, using data from sources such as MEDLINE, the Los Alamos e-Print Archive, SPIRES, and NCSTRL (Newman, 2001), focusing primarily on descriptive studies and certain clustering processes.

In the context of collaboration networks, a primary objective is to predict new links within the network. Exponential Random Graph Models (ERGM; e.g., Snijders 2002, Robins et al. 2007 and Lusher et al. 2013) are commonly used for this purpose, although some applications have focused on economic collaboration networks, given that economic development has been driven by competitiveness (Lee et al., 2012). The use of these models is justified by their recognition as the

most powerful, flexible, and widely applied approach for constructing and testing statistical network models (Luke, 2015). However, machine learning models have advanced significantly, and their applications to networks have been extensively explored in recent years. This trend has led to two distinct perspectives: Models that prioritize the network's structure and models that represent networks as a set of vectors (e.g., Hamilton et al. 2017*b* and Zhang et al. 2020).

On the one hand, Graph Convolutional Network (GCN) models stand out (e.g., Kipf & Welling 2017, Hamilton et al. 2017*a*, and Wu et al. 2021). This approach extends the convolution operation to graphs, enabling the network to learn node representations by considering both node features and graph structure (Yao et al., 2019). Through convolutional layers, nodes aggregate information from their neighbors, allowing the model to capture local relationships and structural patterns within the graph. This capability makes GCNs particularly useful for tasks such as node classification (Kipf & Welling, 2017), link prediction, and clustering (Chiang et al., 2019), as they can infer and generalize the complex relationships inherent in relational data.

On the other hand, embedding algorithms are essential. The primary goal of graph embedding methods is to encode nodes within a latent vector space, effectively capturing each node's properties in a lower-dimensional vector form (Xu, 2021). This representation aims to preserve the overall structure of the graph, positioning related nodes with similar characteristics as "close" vectors. This approach is essentially similar to Hoff's latent space models (e.g., Hoff et al. 2002, Hoff 2007, and Sosa & Buitrago 2021). Typically, continuous-space language models are employed for this purpose, which focus on representing words as vectors (Mikolov, Yih & Zweig, 2013). Given the analogy between graphs and word sequences, these models can be adapted to network contexts by systematically traversing each edge and vertex in the graph (Skiena, 2008) and by categorizing the importance of nodes within the network (Skiena, 2017). This approach results in a set of vectors, enabling the application of all common properties and operations of a traditional vector space.

Despite growing interest in link prediction models, most studies have focused on evaluating statistical and machine learning approaches separately, with few direct comparisons between them. For instance, ERGMs have been widely applied to social and collaboration networks to analyze structural dependencies and infer link formation mechanisms (Snijders 2002; Robins et al. 2007; Lusher et al. 2013). These models provide interpretability but face scalability challenges in large networks. In contrast, GCNs have demonstrated strong predictive performance in network-related tasks, particularly in node classification and link prediction (Kipf & Welling 2017; Yao et al. 2019). While several studies have applied GCNs to collaboration networks (Chiang et al. 2019; Wu et al. 2021), they primarily focus on performance optimization rather than methodological comparisons with traditional statistical models.

Similarly, Word2Vec-based embeddings have gained traction in network science due to their ability to learn latent node representations from structural patterns. These methods have been effectively used for link prediction in various do-

mains, including citation networks and social graphs (Perozzi et al. 2014; Grover & Leskovec 2016). Some studies have explored their applicability to academic collaboration networks, demonstrating their capability to capture meaningful co-authorship structures (Xu, 2021). However, existing research tends to assess these methods in isolation rather than comparing them directly with ERGMs or GCNs.

Despite significant advancements in link prediction, existing methodologies tend to fall into two distinct paradigms: statistical models and deep learning-based approaches. On the one hand, statistical methods, such as ERGMs, are widely recognized for their ability to capture structural dependencies within networks, making them valuable for hypothesis-driven network analysis. However, these models face considerable computational challenges, particularly in large and dense networks, where estimation becomes intractable. On the other hand, machine learning and deep learning techniques, including Graph Neural Networks (GNNs) and embedding-based methods, have demonstrated superior predictive accuracy and scalability, leveraging data-driven representations to infer missing links. However, these methods often lack interpretability, making it difficult to discern the underlying mechanisms governing link formation.

This study aims to bridge this methodological gap by systematically comparing ERGMs, GNNs, and embedding-based approaches in the context of academic collaboration networks. Our analysis evaluates the trade-offs between interpretability, predictive performance, and computational efficiency, providing empirical insights into the suitability of each approach across different network structures. By offering a unified perspective on these methods, we contribute to a more nuanced understanding of link prediction, enabling researchers and practitioners to make informed methodological choices based on network characteristics and research objectives.

Thus, despite the popularity of the modeling approaches provided above, to the best of our knowledge, a comprehensive comparison of these tools for link prediction remains unavailable. Therefore, this study systematically compares the predictive performance of ERGMs, GNNs, and embedding models across five academic collaboration networks. The objective is to determine which approach offers the highest accuracy and computational efficiency and to provide clear recommendations for their application across various contexts. Although the analysis focuses on the specific case of the Astro-Ph (Astrophysics) network, as it is the largest and densest among the five networks, the findings are shown to be consistent across the other cases as well.

This study is crucial, as understanding the specific strengths and limitations of each modeling method is imperative for advancing network analysis and developing more effective techniques for predicting relationships in collaboration networks and other relational datasets with similar structures. The scope of this project encompasses a detailed review and careful implementation of these models (which is not a straightforward task!) followed by a comparison using specialized predictive metrics, and the generation of recommendations based on the results, with a focus on academic collaboration networks.

The remainder of the article is organized as follows. Section 2 provides a detailed overview of the fundamental aspects of the models under study. Section 3 offers a comprehensive comparison of the models using five well-known academic collaboration networks. Finally, Section 4 presents the findings along with several recommendations for future research.

2. Modeling

In this section, we present the most relevant details regarding the models to be evaluated for predictive purposes, namely, Exponential Random Graph Models (ERGMs), Graph Neural Networks (GNNs), and embedding models, including their corresponding theoretical and computational details.

Here's the translation and revision of your section into academic English, with enhanced clarity and structure:

2.1. Exponential Random Graph Model

The Exponential Random Graph Model (ERGM) is a statistical framework used to represent and analyze complex networks, allowing for the capture of dependencies between links and the characteristics of the nodes that compose the network. Unlike simpler models, such as the Erdős-Rényi model (Erdős & Rényi, 1960) and the generalized random graph model (Newman et al., 2001), ERGMs allow for the inclusion of structural dependencies, such as tendencies toward triangle formation or the preference for certain nodes to be more highly connected (Lusher et al., 2013). Fundamentally, ERGMs model the probability of observing specific link formations in a network based on a set of parameters associated with both nodal and network statistics. One clear advantage of this approach is the interpretability of its parameters, which aids in understanding the underlying factors influencing link formation in a particular network. However, in large networks, the computational cost of estimating ERGM parameters can be substantial, which complicates the process and often leads to the use of alternative algorithms (Handcock et al., 2008).

An ERGM is formally defined as

$$p\left(\mathbf{y} \mid \boldsymbol{\theta}\right) = \frac{1}{\kappa} \exp\left\{\boldsymbol{\theta}^{\mathsf{T}} \mathbf{g}(\mathbf{y})\right\},$$

where $\mathbf{y} = [y_{i,j}]$ represents the realization of a random adjacency matrix $\mathbf{Y}$, $\mathbf{g}(\mathbf{y})$ is a K-dimensional vector of network statistics (endogenous variables) and vertex characteristics (exogenous variables), $\boldsymbol{\theta}$ is a K-dimensional vector of unknown parameters, and $\kappa \equiv \kappa(\boldsymbol{\theta})$ is the normalizing constant ensuring that $p\left(\cdot \mid \boldsymbol{\theta}\right)$ is a proper probability distribution.

ERGM functionality involves specifying relevant network statistics and estimating the corresponding parameters, which indicate the relative significance of each statistic within the network's relational structure. Due to the complexity

of normalizing probabilities across all possible networks, parameter estimation is typically performed using Markov Chain Monte Carlo (MCMC) methods (see Gamerman & Lopes 2006 and Robins et al. 2007), which can be slow depending on the network size. ERGMs are particularly valuable when networks are assumed to form not only through random connections but also through significant structural patterns that can be captured and analyzed using both endogenous and exogenous variables.

ERGMs rely on a set of fundamental assumptions about network formation, particularly regarding sparsity and size. A key assumption is that network ties are not formed independently but rather exhibit dependencies that ERGMs attempt to model through network statistics, such as edges, triangles, and other structural configurations. However, the feasibility of ERGM estimation is strongly affected by network sparsity and size. Sparse networks, where the number of links is significantly lower than the number of possible connections, often lead to better model convergence, as the likelihood surface remains more tractable. In contrast, dense networks tend to pose severe challenges for ERGM estimation, as the inclusion of additional dependencies (e.g., higher-order interactions) leads to complex degeneracy issues, where the model assigns excessive probability mass to unrealistic network configurations.

Additionally, the computational burden of ERGM estimation scales dramatically with network size due to the intractability of the normalizing constant, requiring the use of MCMC methods to approximate the likelihood as discussed above. This results in significant limitations when applying ERGMs to large-scale networks, as demonstrated in this work. Despite these constraints, ERGMs remain advantageous in certain scenarios. They are particularly useful for analyzing small to moderately sized networks where structural dependencies are of primary interest. For example, in social science applications where understanding the role of homophily, transitivity, or preferential attachment in shaping network structures is essential, ERGMs provide interpretable parameter estimates that inform theoretical insights. Additionally, when the focus is on hypothesis testing rather than large-scale predictive modeling, ERGMs offer a statistically principled framework for evaluating structural tendencies in networks.

By explicitly modeling network dependencies and incorporating domain-specific knowledge through exogenous covariates, ERGMs remain a valuable tool despite their computational drawbacks. The interpretability of ERGM parameters allows researchers to make inferential claims about the underlying mechanisms governing network formation, distinguishing them from purely predictive approaches such as machine learning models. Consequently, while deep learning and embedding-based methods excel in scalability and predictive accuracy, ERGMs retain their relevance in fields where theoretical network modeling and hypothesis-driven analysis are the primary objectives.

Several approaches have been proposed to address scalabiloty challenges. Network decomposition methods, such as block-based modeling and subgraph sampling, reduce computational complexity by fitting ERGMs to smaller network partitions while preserving key structural properties. Another alternative is the use

of pseudolikelihood estimation (Strauss & Ikeda, 1990), which approximates the likelihood function by conditioning on local node neighborhoods, significantly improving computational efficiency. Additionally, variational inference techniques (Salter-Townshend & Murphy, 2013) have been explored as an alternative to MCMC, offering faster convergence with controlled approximation error.

2.2. Graph Convolutional Network

Graph Convolutional Networks (GCNs) are neural networks (Goodfellow et al., 2016) specifically designed to process graph-structured data directly, without transforming the graph. Unlike traditional convolutional networks, which operate on grid-based data like images, GCNs extend the convolution operation to graphs, capturing relationships between nodes and edges. This enables effective learning for tasks like node classification and link prediction in complex graph-structured data.

Graph convolution allows each node to aggregate and process information from not only its own features but also from those of its neighboring nodes, as illustrated in Figure 1. This capability is essential for capturing local relationships and structural patterns in the graph. In this context, the convolution operation is defined by (Kipf & Welling, 2017) as:

$$\mathbf{H}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-1/2}\,\tilde{\mathbf{A}}\,\tilde{\mathbf{D}}^{-1/2}\,\mathbf{H}^{(l)}\,\mathbf{W}^{(l)}\right),$$

where $\mathbf{H}^{(l)}$ is the feature matrix in layer l, $\tilde{\mathbf{A}} = \tilde{\mathbf{A}} + \mathbf{I}$ is the adjacency matrix with added self-connections ($\mathbf{I}$ is the identity matrix), $\tilde{\mathbf{D}}$ is the diagonal degree matrix of $\tilde{\mathbf{A}}$, $\mathbf{W}^{(l)}$ is the weight matrix in layer l, and σ is an activation function, such as ReLU, which provides non-linearity, helping the model capture complex relationships across nodes.

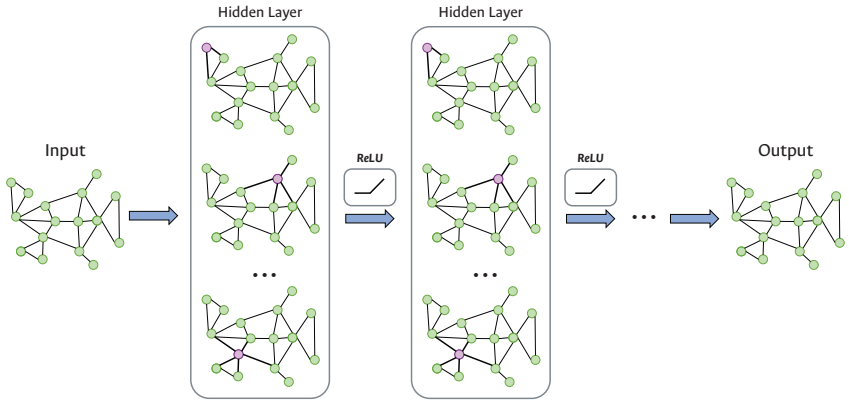

FIGURE 1: GCN Model: Adapted from Kipf & Welling (2017) and Xu (2021).

Graph-level outputs can be modeled by introducing pooling operations that aggregate node information across the graph structure (Duvenaud et al., 2015). By

calculating these representations as weighted combinations of the features of neighboring nodes, pooling operations help capture information from multiple scales within the graph, allowing the model to abstract details progressively. This enables the network to learn complex structures in a hierarchical and layered manner (Kipf & Welling, 2017), supporting tasks that require an understanding of the graph as a whole, such as graph classification or property prediction.

GCNs are often considered "black-box" models due to their deep learning nature. However, their interpretability can be improved by analyzing the learned node embeddings and understanding how different layers aggregate neighborhood information. GCNs function by iteratively updating node representations based on local connectivity patterns, which means that nodes with similar structural roles in the network tend to have similar embeddings. One way to interpret GCN outputs is through feature importance analysis, where trained models can reveal which node attributes or network connections contribute most to link formation. Additionally, attention mechanisms, when incorporated into GCN architectures, provide insights into the relative importance of neighboring nodes in learning representations.

Beyond direct feature analysis, dimensionality reduction techniques such as t-distributed Stochastic Neighbor Embedding (t-SNE) or Principal Component Analysis (PCA) can be used to visualize the learned node embeddings. When applied to the Astro-Ph collaboration network, these techniques reveal that nodes representing authors from the same research community tend to cluster together, demonstrating that GCN embeddings capture meaningful community structures.

2.3. Word2Vec

Word2Vec is a deep learning model widely employed in natural language processing (NLP; e.g., Amarasinghe et al. 2024) to learn continuous vector representations of words within a high-dimensional space. As its name suggests ("Word to Vector"), the model uses neural networks to transform *words* into *vectors*, positioning words with similar contexts close to each other in the vector space (Mikolov, Yih & Zweig, 2013). One commonly used method is the skip-gram approach (Silge & Robinson, 2017), which aims to predict the context surrounding a given word (Xu, 2021). The skip-gram model's loss function is defined as follows:

$$\mathcal{L}_{\text{Skip-gram}} = -\sum_{t=1}^{T} \sum_{-c \leq j \leq c} \log p(w_{t+j} \mid w_t)\,,$$

where w_t is the target word at time t, c is the context window size, and $p(w_O \mid w_I)$ is the probability of observing the word w_O given the context word w_I, modeled through a neural network.

This model has been widely applied in natural language processing (NLP) but it is also adaptable to social network analysis due to its ability to transform high-dimensional, non-Euclidean spaces into lower-dimensional vector spaces (Xu, 2021). In this context, *sentences* are represented by random walks on the target

graph, following existing edges, as illustrated in Figure 2. This process is repeated a set number of times to generate a sequence of nodes, resembling a sentence in NLP. Random walks capture both local structure (directly connected nodes) and broader patterns (nodes connected through multiple steps), making them efficient for handling large graphs.

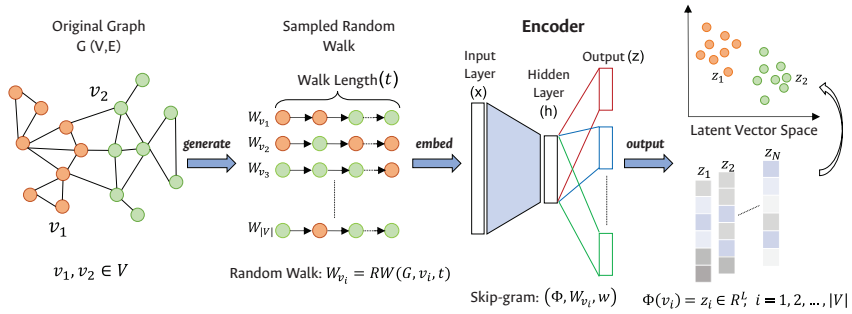

FIGURE 2: Embedding model: Adapted from Xu (2021).

In embedding models for graphs, positive and negative edges play a crucial role in training. Positive edges represent actual, observed connections between nodes, while negative edges signify pairs of nodes without connections. Negative sampling is important because it balances the dataset, preventing the model from overemphasizing positive connections (Xu, 2021). Since graphs usually contain far more possible node pairs than actual edges, random sampling of negative edges is used to reduce computational demands. This balance in edge types enables the model to learn meaningful distinctions between connected and unconnected nodes, which enhances its predictive power.

The embeddings produced by these models are vectors that encapsulate the relationships between nodes based on their connections. These vectors can then serve as input to other machine learning models—such as a multilayer perceptron (MLP; e.g., Rumelhart & McClelland 1986) or other traditional models—enabling predictions or classifications. The process allows for applying neural network-based models to graph data, leveraging learned structural patterns and providing insights into the relational data embedded in the graph.

Word2Vec-based methods generate low-dimensional vector representations of nodes by leveraging random walks that traverse the network structure. Despite being learned in an unsupervised manner, these embeddings encode important relational patterns, such as homophily (the tendency of similar nodes to connect) and structural equivalence (nodes with similar roles in the network having similar representations). Researchers can interpret these embeddings by analyzing node similarity scores—nodes with high cosine similarity in the embedding space are likely to be closely related in the original network.

Moreover, visualization techniques like t-SNE allow researchers to observe whether the learned embeddings naturally cluster nodes into meaningful groups, reflecting network communities. In our analysis, the embeddings generated by Word2Vec effectively capture co-authorship structures, with authors frequently

collaborating in the same research subfields appearing closer together in the latent space. This suggests that Word2Vec embeddings retain essential network properties, making them useful not only for link prediction but also for exploratory network analysis.

These interpretability techniques demonstrate that while deep learning models may initially appear less transparent than statistical approaches, careful analysis of learned embeddings can provide valuable insights into the underlying structure and dynamics of collaboration networks.

2.4. Model Comparison

To facilitate a clear comparison between the models analyzed in this study, Table 1 summarizes their key characteristics, including their primary purpose, task suitability, interpretability, scalability, and whether they incorporate node or edge attributes.

Table 1: Comparative summary of link prediction models.

Model	Purpose	Task type	Interpretability	Scalability	Attributes
ERGM	Statistical modeling of network structures	Link formation analysis, hypothesis testing	High (explicit parameter estimates)	Low (computationally expensive for large networks)	Yes (can incorporate exogenous covariates)
GCN	Learning node representations from graph structures	Link prediction, node classification	Moderate (depends on architecture, can include attention mechanisms)	High (efficient with mini-batching)	Yes (integrates node attributes)
Word2Vec	Learning low-dimensional embeddings based on random walks	Link prediction, clustering	Low (embeddings capture structure but lack direct interpretability)	High (scalable for large networks)	No (relies purely on graph topology)

ERGMs are particularly useful for understanding structural dependencies in small to moderate-sized networks, providing interpretable parameter estimates that describe the likelihood of link formation based on endogenous and exogenous network features. However, their computational complexity limits their applicability to large networks. In contrast, GCNs leverage node features and structural information to learn adaptive representations, making them well-suited for large-scale link prediction tasks. While their interpretability depends on the architecture, techniques such as attention mechanisms can provide insights into node relationships. Word2Vec-based embeddings, although less interpretable, offer an efficient and scalable approach to capturing latent structural patterns in networks. These distinctions highlight the trade-offs between statistical and machine learning-based approaches, reinforcing the need for a comparative evaluation across real-world collaboration networks.

2.5. Computational Complexity

Beyond empirical runtime analysis, understanding the theoretical computational complexity of each model provides further insight into their scalability across different network sizes. ERGMs are known for their high computational cost, primarily due to the intractability of the normalizing constant required for exact inference. In the worst-case scenario, the complexity of computing the likelihood of an ERGM scales exponentially with the number of nodes. Since exact estimation is infeasible for large networks, MCMC methods are commonly employed, which reduce the computational burden to approximately $O(n^3)$. However, this still presents significant scalability challenges, as the computational time increases rapidly with the number of nodes, making ERGMs impractical for large-scale networks.

GCNs, in contrast, are designed to efficiently process graph-structured data through iterative aggregation of node features from neighboring nodes. The computational complexity of GCNs depends on the number of nodes and edges in the graph. When using full-batch training, the complexity is $O(n^2)$, which can become computationally demanding for large networks. However, in practical implementations, GCNs often utilize mini-batch training and sparse matrix operations, reducing the complexity to approximately $O(n)$, making them significantly more scalable than ERGMs. Additionally, GCNs benefit from parallelization on GPUs, further improving their efficiency when applied to large graphs.

Word2Vec-based embeddings, combined with a MLP for link prediction, offer another computationally efficient approach. The complexity of learning node embeddings via Word2Vec primarily depends on the number of sampled random walks and their lengths. The training process typically scales as $O(n \log n)$, where the logarithmic term arises from the negative sampling technique used for optimization. This makes Word2Vec-based approaches highly scalable while still capturing meaningful structural patterns within the network.

This theoretical complexity analysis aligns with our empirical findings, where ERGMs exhibited the longest execution times due to their reliance on MCMC-based inference, GCNs demonstrated the fastest runtimes when employing mini-batch training, and Word2Vec-based embeddings provided a balance between computational efficiency and predictive accuracy. These results highlight the importance of considering both theoretical scalability and practical implementation strategies when selecting a link prediction method for large-scale networks.

3. Illustration

In this section, we analyze collaboration networks from Arxiv by comparing three link prediction models: ERGM, GCN, and Word2Vec. Our findings indicate that ERGM faces limitations with large networks, whereas the GCN model is the fastest, and the Word2Vec model offers the highest accuracy. These results provide empirical evidence that deep learning models are more effective for handling complex networks.

3.1. Data

To illustrate the methodologies presented in the previous section, we examine five collaboration networks representing scientific partnerships between authors of articles submitted to corresponding categories on the Arxiv platform:

- **Astro-Ph**: Astrophysics, with 198,110 edges and 18,772 nodes.
- **Cond-Mat**: Condensed Matter, with 93,497 edges and 23,113 nodes.
- **Gr-Qc**: General Relativity, with 14,496 edges and 5,242 nodes.
- **Hep-Ph**: High-Energy Physics, with 118,521 edges and 12,008 nodes.
- **Hep-Th**: Theoretical High-Energy Physics, with 25,998 edges and 9,877 nodes.

These networks are undirected and unweighted (binary), indicating only whether two authors collaborated, without specifying the strength of collaboration. Since author characteristics are not included in the dataset, the analysis focuses primarily on relational data. This article centers its analysis on the Astro-Ph network, as it has the most connections. However, results are generally consistent across the other networks, as demonstrated later.

3.2. Exploratory Analysis of the Astro-Ph Network

In this section, we examine the structure of the Astro-Ph network. The network contains a few authors with a high number of connections, acting as central hubs. Naturally, this characteristic is uncommon: Only 59 individuals have more than 400 connections, representing just 0.31% of all nodes. The average number of connections per author is approximately 18, and the maximum separation between two authors is 14 edges. The network's density of only 0.0022 indicates limited connectivity, which, in turn, suggests the presence of numerous cliques. The largest clique consists of 57 members, indicating the formation of distinct author groups, likely due to researchers' preference for collaborating with colleagues from the same institution rather than with external partners. Finally, we observe that the network has 290 components, with the giant component including 95.37% of the individuals.

Given the network's size, visualizing the complete graph is impractical, so we display a subgraph of the 50 individuals with the highest degree, as shown in Figure 3. First, we observe that a large portion of authors tends to collaborate with only a single co-author. Thus, researchers who work with multiple partners often choose collaborators within their immediate circle. This supports the conclusion that the collaborative network favors the formation of cliques. While many researchers collaborate with only one partner, those with more co-authors tend to work in more closed groups.

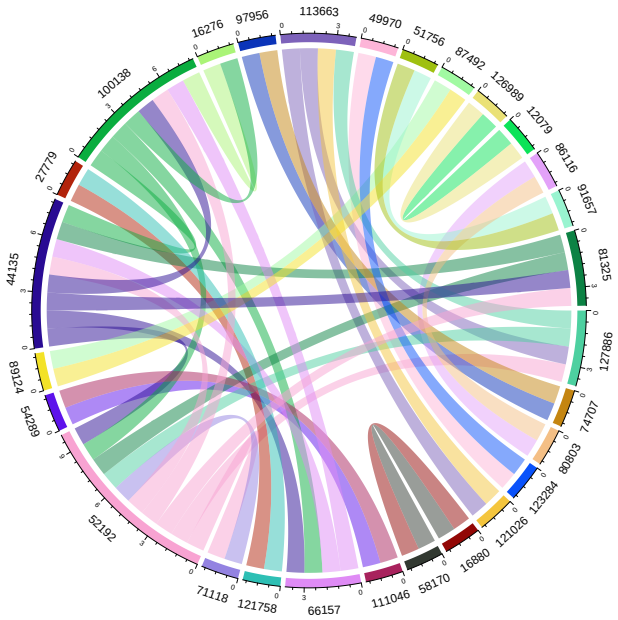

FIGURE 3: Chord diagram of a subset of authors in the Astro-Ph network.

3.3. Results

First, the ERGM model is fitted to the large Astro-Ph network using the most basic configuration, which considers only links between nodes. This simple approach enables the model to handle a large network, but at the expense of performance, as it does not account for other important network features, such as triangles or other complex structures. When more structural features are included in the model formulation, the fitting algorithms fail to converge, even with distributed parallel computing over 20 cores, highlighting a clear limitation of this type of model.

The ERGM coefficient for the existence of links between nodes in the full Astro-Ph network is -6.7895, which is highly statistically significant. In natural scale, this coefficient indicates a 0.4998 probability of interaction between two authors. For the restricted network, this coefficient is positive and significant, suggesting a high likelihood of node connections while holding other factors constant. In this case, we also consider the coefficient associated with triangle formation, estimated at 9.9544, which indicates a high likelihood of links completing triads. Similarly, the coefficient for four-node star configurations is both positive and significant.

These findings underscore one of the key strengths of ERGMs: their capacity for providing interpretable results. However, given the dimensions of the networks that we consider here, ERGM execution times are exceedingly high. For the largest network, approximately nine hours are needed to fit the model and make link predictions. Although ERGMs generally offer a good fit, their low computational efficiency reflects that they are not well-suited for handling high-dimensional data.

To train the two remaining models, we define a balanced set of positive and negative edges, as the number of negative edges is much higher due to the network's low density. The second model implemented is a Graph Convolutional Network (GCN) model with two convolution layers. The first layer projects the input node features into a 64-dimensional latent space, applying the ReLU activation function to introduce non-linearity. The second layer also projects these representations into a 64-dimensional space, and finally, the output layer produces the model's predictions, optimized using the Adam optimizer with a learning rate of 0.01. The binary link prediction loss is calculated using a combination of logarithmic losses for positive and negative edges. The model input consists of the edge indices, allowing the convolution to operate over the graph structure and capture topological information in the learned representations.

Finally, we fit the Word2Vec model using a DeepWalk-based approach to generate 100 random walks on the graph, each with a length of 30 nodes. The model is configured with a vector size of 32 dimensions, a contextual window of 10 nodes, a minimum count of 1, and a skip-gram approach, utilizing 4 workers to parallelize the process. The resulting node embeddings are then used as input for a multilayer perceptron (MLP). Figure 4 visualizes the Word2Vec model's results by projecting the original 12-dimensional embeddings into a two-dimensional space using t-distributed Stochastic Neighbor Embedding (t-SNE; e.g., van der Maaten & Hinton 2008). In this projection, nodes from the same community in the original graph tend to cluster together, indicating that the embeddings effectively capture community structure despite the reduction from the original 12 dimensions.

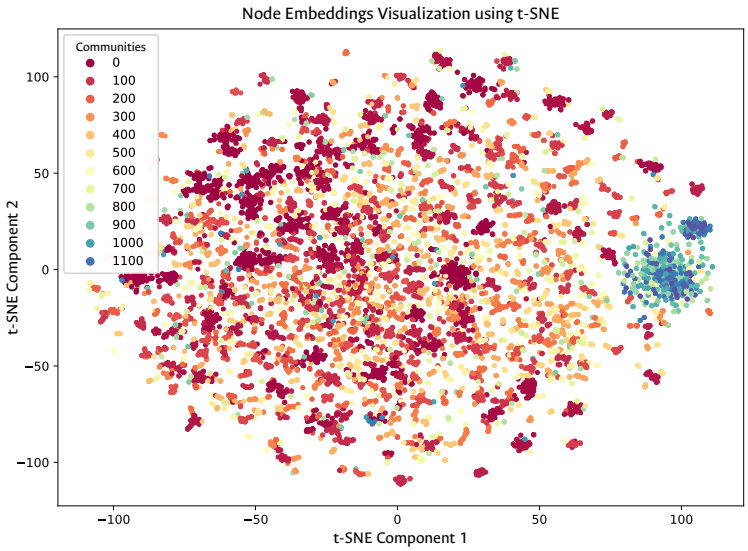

FIGURE 4: Visualization of embeddings and their corresponding clusters (using infomap) using t-SNE for the Astro-Ph network.

Continuing with this process, the MLP model takes the embeddings generated by Word2Vec as input. This MLP has a two-layer architecture: The first layer

contains 64 units and uses the ReLU activation function to introduce non-linearity, while the second layer projects to the appropriate output dimension. The model is optimized using the Adam optimizer with a learning rate of 0.01. As before, the binary link prediction loss is calculated using a combination of logarithmic losses for positive and negative edges.

To compare the models' performance, we use the area under the receiver operating characteristic (ROC) curve (AUC; e.g., Fawcett, 2006) as a metric to quantify each model's ability to distinguish between classes, with values closer to 1 indicating higher performance. Additionally, we use a confusion matrix (e.g., Sokolova & Lapalme, 2009) to observe the model's correct and incorrect predictions, categorized as true positives, false positives, true negatives, and false negatives, providing a detailed understanding of classification performance.

The results for the models using the Astro-Ph network are shown in Figure 5. For the ERGM, the AUC is 0.9797, indicating a high level of performance. The confusion matrix also shows that 96% of positive class samples were correctly classified, while 4% were incorrectly classified as negatives. Furthermore, this model had no false positives, which is advantageous for classification, and 100% of negative class samples were correctly classified. The high accuracy in negative class classification (100%), along with a high AUC and a significant percentage of false positives, suggests that the model generalizes well despite its substantial computational cost.

On the other hand, the GCN model achieves an AUC of 0.9590, which is quite good and competitive compared to the previous model. The confusion matrix reveals that 91% of negative class samples were correctly classified, while 9% were incorrectly classified as positives. These results indicate that the model performs well for negative edges (edges that do not exist in the network). For the positive class, 90% were correctly classified, while 10% were false negatives. These findings suggest that the model is effective in identifying positive edges (edges that do exist in the network). The AUC and confusion matrix show that the model performs well in distinguishing existing and non-existing connections within the network.

The Word2Vec model achieved an AUC of 0.9875, ranking it as the best-performing model among those evaluated. The confusion matrix shows that 95% of negative class samples were correctly classified, while 5% were incorrectly classified as positives. Additionally, for the positive class, 95% were correctly classified, with 5% classified as false negatives. These results demonstrate the model's high effectiveness in identifying both positive and negative edges, with a low error rate in both cases. The AUC and confusion matrix confirm this model as the top performer in predictive accuracy.

Finally, when comparing models in terms of computational efficiency, measured by the time taken to fit the models to the datasets and generate predictions, the ERGM exhibits the poorest performance. For the Astro-Ph network, the ERGM takes over nine hours to execute, even with parallel computing methods, while the GCN model takes less than 8 seconds in total. Meanwhile, the Word2Vec model takes just over half an hour, placing it between the two. This highlights the capability of modern machine learning models to handle large volumes of data efficiently.

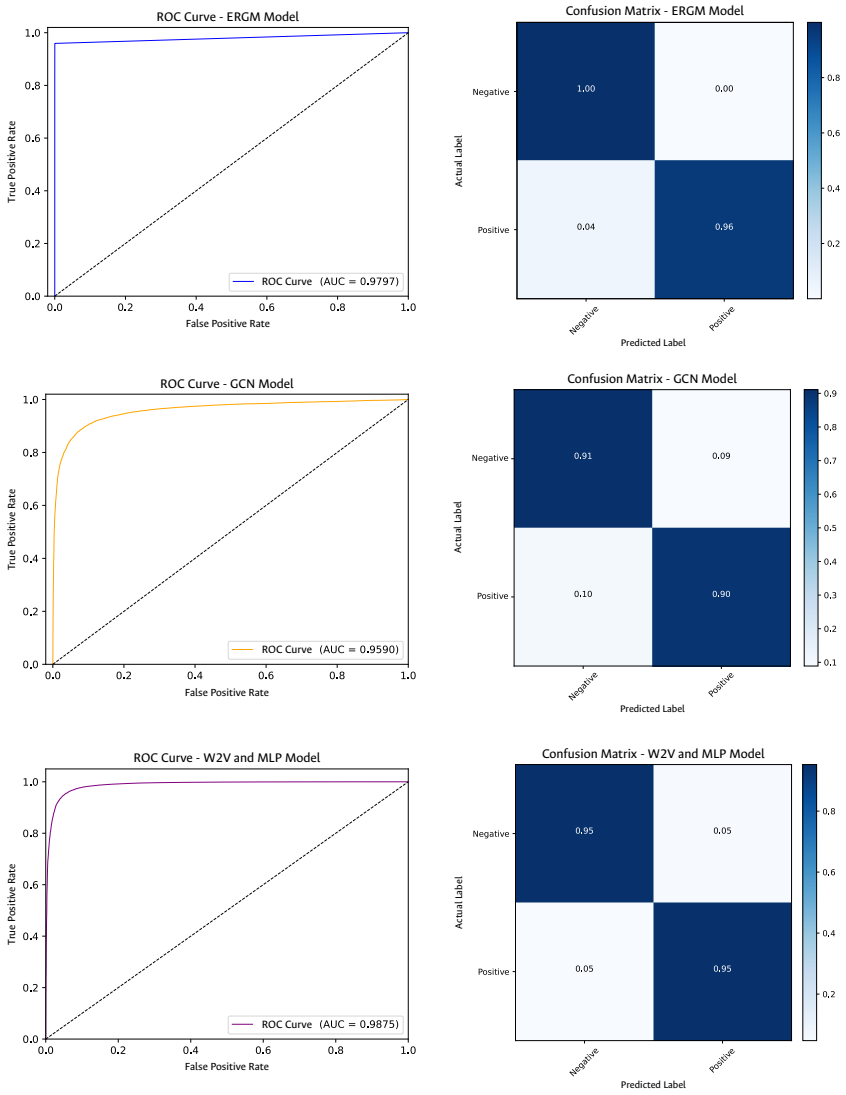

FIGURE 5: ROC curve and confusion matrix for the ERGM (first row), GCN model (second row), and Word2Vec model (third row) models.

The AUC and time taken for each network and model are reported in Table 2. The results show that, in terms of accuracy (AUC), the Word2Vec model achieves the highest scores, reaching 0.99 in most networks. Meanwhile, GCN also achieves high AUC values, particularly in the Hep-Ph and Gr-Qc networks, where its performance is similar to that of the Word2Vec model. In contrast, ERGM shows greater variability in the AUC values, with high values in some networks (such as Astro-Ph and Hep-Ph) and lower values in others (such as Gr-Qc). Regarding execution time, GCN is the fastest model, with significantly lower

times compared to ERGM and Word2Vec, which are much slower, especially in larger networks like Astro-Ph and Cond-Mat. Overall, Word2Vec stands out in terms of accuracy, while GCN is notably more efficient in execution time. The codes used to fit the models can be found at `https://github.com/damartinezsi/An-unified-approach-to-link-prediction-in-collaboration-networks`.

Table 2: AUC results and time (in seconds) for different networks and models.

	AUC			**Time**		
	ERGM	**GCN**	**Word2Vec**	**ERGM**	**GCN**	**Word2Vec**
Astro-Ph	0.98	0.96	0.98	31,284.0	7.9	2,174.1
Cond-Mat	0.96	0.91	0.99	18,360.0	4.7	2,594.4
Gr-Qc	0.78	0.89	0.99	3,307.8	1.1	472.2
Hep-Ph	0.97	0.95	0.99	11,844.0	2.9	1,257.7
Hep-Th	0.86	0.84	0.99	638.4	1.1	943.1

4. Discussion

The high computational cost associated with fitting more complex ERGMs on large networks poses a considerable challenge. The inclusion of additional terms and more complex structures, such as triangles and other dependency patterns, increases the difficulty of fitting the model, often preventing it from meeting convergence criteria. In this context, taking a relatively small sub-sample of nodes is not a viable alternative, as it results in the loss of connection patterns present in the complete network, leading to unreliable predictions. This situation highlights the limited capacity of ERGMs to handle large-scale networks and underscores the need to develop and apply alternative strategies for addressing large networks without compromising model fit quality.

Machine learning models demonstrated outstanding performance in link prediction, underscoring their ability to capture complex patterns in network data. Unlike traditional approaches such as ERGM, which rely on statistical assumptions and are more suitable for smaller networks, deep learning models like GCN and Word2Vec models are designed to efficiently scale with large data volumes. This is due to their capacity to process and learn from both global and local graph structures through deep layers and vector embeddings, making them powerful and flexible tools for analyzing large-scale, high-dimensional networks.

While GCN and Word2Vec-based models outperform ERGMs in scalability and predictive accuracy, they also have key limitations. A major drawback is interpretability—unlike ERGMs, which provide explicit statistical estimates, deep learning models generate latent representations that are harder to analyze (Ying et al., 2019). Although attention mechanisms and feature importance methods help, they do not fully explain how predictions are made. These models are also sensitive to hyperparameters and data availability. Their performance depends on architecture choices, learning rates, and embedding dimensions (You et al., 2020), making optimization non-trivial. GCNs, in particular, require node features for

effective learning, which limits their applicability when such attributes are missing or unreliable (Kipf & Welling, 2017).

Additionally, deep learning models struggle with dynamic networks, as they often require retraining when the network evolves (Rossi et al., 2020). While dynamic graph neural networks (Pareja et al., 2020) attempt to address this, handling evolving relationships remains an open challenge. Despite these limitations, GCNs and Word2Vec remain powerful tools for large-scale link prediction. Future research could explore hybrid approaches that integrate the interpretability of ERGMs with the adaptability of deep learning models.

The evaluation of link prediction models often relies on the area under the ROC curve (AUC), as it provides a robust measure of a model's ability to distinguish between positive and negative links. However, other performance metrics have been proposed in the literature to address specific challenges in network-based classification tasks. For instance, precision-recall AUC (PR-AUC) is particularly useful in highly imbalanced networks, where the number of non-existent links far exceeds the number of observed links (Davis & Goadrich, 2006). The F1-score has also been widely employed to balance precision and recall in network-based tasks (Sokolova & Lapalme, 2009), while mean average precision (MAP) is frequently used to assess ranking performance in recommendation and retrieval systems (Lu & Zhou, 2011).

Some studies have highlighted the advantages of using multiple evaluation metrics to gain a more nuanced understanding of model performance. For example, (Yang et al., 2015) examined PR-AUC and MAP in the context of social network link prediction, showing that different metrics may yield varying conclusions depending on network sparsity and density. Similarly, (Kivelä et al., 2014) discussed the effectiveness of recall-based measures in multilayer network analysis, emphasizing their importance in applications where missing links are of primary concern. These findings suggest that while AUC remains a standard metric, alternative measures can complement the evaluation process, particularly when networks exhibit significant class imbalance or when ranking-based link prediction is the primary objective.

Although AUC serves as the primary metric in our study for consistency across models, future research could explore additional evaluation measures to provide a more granular assessment of predictive performance in different network settings. This would allow for a deeper investigation into model strengths and weaknesses beyond binary classification accuracy.

While this study focuses on academic collaboration networks, the findings have broader implications for other types of complex networks, including citation networks and social media networks. The relative performance of ERGMs, GCNs, and Word2Vec-based embeddings is influenced by network characteristics such as density, node attribute availability, and the nature of link formation, which vary across different domains.

In citation networks, where nodes represent academic papers and edges denote citations, the structural patterns differ significantly from collaboration graphs. Unlike co-authorship networks, where ties are typically bidirectional and formed

through mutual collaboration, citation networks exhibit directed and acyclic properties, meaning that older papers cannot cite newer ones. Despite this structural difference, deep learning models such as GCNs and Word2Vec-based embeddings remain effective, as they can leverage citation proximity and topic similarity to predict future citations. However, ERGMs face additional challenges due to the directed nature and hierarchical dependencies inherent in citation networks, requiring model adaptations to handle asymmetric relationships.

In social media networks, where edges represent interactions such as friendships, follows, or message exchanges, link formation dynamics differ further. Social media networks often exhibit higher sparsity and dynamic growth, with time-dependent interactions playing a crucial role. While ERGMs can still be applied in small-scale settings with well-defined network snapshots, they may struggle with the evolving nature of social media graphs. In contrast, GCNs and Word2Vec-based embeddings perform well in these environments due to their ability to learn adaptive representations from evolving network structures. Additionally, node features such as user behavior, textual content, and engagement metrics can be incorporated into machine learning models to improve predictive accuracy, something that is less feasible with purely structural ERGMs. Overall, while the specific effectiveness of each approach depends on network characteristics, the insights gained from this study provide a foundational understanding of the trade-offs between interpretability, scalability, and predictive accuracy across different network types.

Future work could explore hybrid models that integrate the statistical rigor of ERGMs with the flexibility of deep learning architectures. For instance, latent variables inferred from ERGMs could serve as input features for GCNs, combining explicit structural modeling with adaptive representation learning to improve both interpretability and scalability. Another promising direction involves semi-supervised learning, where partially labeled data can enhance link prediction. Many real-world networks, such as social media or citation networks, contain known relationships that could be leveraged using contrastive learning or self-training GCNs. Incorporating statistical priors from ERGMs into semi-supervised frameworks could provide a principled approach to integrating structural dependencies with data-driven learning.

Additionally, latent space-based models offer an alternative for link prediction by embedding nodes into a continuous space where proximity reflects link probability. Comparing these models with ERGMs, GCNs, and Word2Vec-based embeddings could provide further insights into their relative advantages across different network types. Future research could also focus on scalability improvements for statistical models. While deep learning approaches efficiently handle large-scale networks, extending ERGMs to accommodate larger datasets remains a challenge. Techniques such as variational inference, scalable MCMC, or approximate likelihood estimation could improve computational feasibility for ERGMs in high-dimensional networks. Finally, applying these models to networks with nodal attributes would enhance predictive capability by capturing the heterogeneity and dynamism of complex networks. Extending the current analysis to heterogeneous, temporal, or multimodal networks could provide a deeper understanding of link formation mechanisms across various domains.

Statements and Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this article.

During the preparation of this work the authors used ChatGPT-4-turbo in order to improve language and readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

[Received: November 2024 — Accepted: March 2025]

References

Amarasinghe, S. et al. (2024), *Explainable Artificial Intelligence: Second World Conference, xAI 2024*, Springer. https://www.springer.com/

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', KDD '19, Association for Computing Machinery, New York, NY, USA, p. 257–266.

Davis, J. & Goadrich, M. (2006), The relationship between precision-recall and roc curves, *in* 'Proceedings of the 23rd International Conference on Machine Learning', ACM, pp. 233–240.

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A. & Adams, R. P. (2015), 'Convolutional Networks on Graphs for Learning Molecular Fingerprints'.

Erdős, P. & Rényi, A. (1960), 'On the evolution of random graphs', *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* **5**, 17–61.

Fawcett, T. (2006), 'An introduction to roc analysis', *Pattern Recognition Letters* **27**(8), 861–874.

Gamerman, D. & Lopes, H. F. (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, 2 edn, Chapman and Hall/CRC.

Goodfellow, I., Bengio, Y. & Courville, A. (2016), *Deep Learning*, MIT Press. http://www.deeplearningbook.org

Grover, A. & Leskovec, J. (2016), node2vec: Scalable feature learning for networks, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 855–864.

Hamilton, W. L., Ying, R. & Leskovec, J. (2017*a*), Inductive representation learning on large graphs, *in* 'Advances in Neural Information Processing Systems (NeurIPS)'.

Hamilton, W. L., Ying, R. & Leskovec, J. (2017*b*), 'Representation learning on graphs: Methods and applications', *IEEE Data Engineering Bulletin* **40**(3), 52–74.

Handcock, M., Hunter, D., Butts, C., Goodreau, S. & Morris, M. (2008), 'Statnet: Software tools for the representation, visualization, analysis and simulation of network data', *Journal of statistical software* **24**, 1548–7660.

Hoff, P. (2007), 'Modeling homophily and stochastic equivalence in symmetric relational data', *Advances in neural information processing systems* **20**.

Hoff, P. D., Raftery, A. E. & Handcock, M. S. (2002), 'Latent space approaches to social network analysis', *Journal of the american Statistical association* **97**(460), 1090–1098.

Kipf, T. N. & Welling, M. (2017), Semi-supervised classification with graph convolutional networks, *in* 'Proceedings of the International Conference on Learning Representations (ICLR)'.

Kivelä, M., Arenas, A., Barthelemy, M., Gleeson, J. P., Moreno, Y. & Porter, M. A. (2014), 'Multilayer networks', *Journal of Complex Networks* **2**(3), 203–271.

Kolaczyk, E. D. & Csárdi, G. (2020), *Statistical analysis of network data with R*, Use R!, 2nd ed edn, Springer, Cham.

Lee, Y., Lee, I. W. & Feiock, R. C. (2012), 'Interorganizational Collaboration Networks in Economic Development Policy: An Exponential Random Graph Model Analysis*', *Policy Studies Journal* **40**(3), 547–573.

Lu, L. & Zhou, T. (2011), 'Link prediction in complex networks: A survey', *Physica A: Statistical Mechanics and Its Applications* **390**(6), 1150–1170.

Luke, D. (2015), *A User's Guide to Network Analysis in R*, Use R!, Springer International Publishing, Cham.

Lusher, D., Koskinen, J. & Robins, G. (2013), *Exponential Random Graph Models for Social Networks: Theory, Methods, and Applications*, Cambridge University Press.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), Efficient estimation of word representations in vector space, *in* 'Proceedings of the International Conference on Learning Representations (ICLR)'.

Mikolov, T., Yih, W.-t. & Zweig, G. (2013), 'Linguistic regularities in continuous space word representations', *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* pp. 746–751.

Newman, M. E. J. (2001), 'The structure of scientific collaboration networks', *Proceedings of the National Academy of Sciences* **98**(2), 404–409.

Newman, M. E. J., Strogatz, S. H. & Watts, D. J. (2001), 'Random graphs with arbitrary degree distributions and their applications', *Physical Review E* **64**(2), 026118.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B. & Leiserson, C. E. (2020), Evolvegcn: Evolving graph convolutional networks for dynamic graphs, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence'.

Perozzi, B., Al-Rfou, R. & Skiena, S. (2014), Deepwalk: Online learning of social representations, *in* 'Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 701–710.

Robins, G., Pattison, P., Kalish, Y. & Lusher, D. (2007), 'An introduction to exponential random graph (p*) models for social networks', *Social Networks* **29**(2), 173–191.

Rossi, E., Kenlay, H., Gorinova, M., Bronstein, M. & Chamberlain, B. (2020), 'Temporal graph networks for deep learning on dynamic graphs', *arXiv preprint arXiv:2006.10637* .

Rumelhart, D. E. & McClelland, J. L. (1986), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, MIT Press, Cambridge, MA.

Salter-Townshend, M. & Murphy, T. B. (2013), 'Variational bayesian inference for the latent position cluster model for network data', *Computational Statistics & Data Analysis* **57**(1), 661–671.

Silge, J. & Robinson, D. (2017), *Text Mining with R: A Tidy Approach*, O'Reilly Media, Inc., Sebastopol, CA. https://www.oreilly.com/library/view/text-mining-with/9781491981658/

Skiena, S. S. (2008), *The Algorithm Design Manual*, Springer London, London.

Skiena, S. S. (2017), *The Data Science Design Manual*, Texts in Computer Science, Springer International Publishing, Cham.

Skvoretz, J. (1990), 'Biased net theory: Approximations, simulations and observations', *Social Networks* **12**(3), 217–238.

Snijders, T. A. B. (2002), 'Markov chain monte carlo estimation of exponential random graph models', *Journal of Social Structure* **3**(2), 1–40.

Sokolova, M. & Lapalme, G. (2009), 'A systematic analysis of performance measures for classification tasks', *Information Processing & Management* **45**(4), 427–437.

Sosa, J. & Buitrago, L. (2021), 'A review of latent space models for social networks', *Revista Colombiana de Estadística* **44**(1), 171–200.

Strauss, D. & Ikeda, M. (1990), 'Pseudolikelihood estimation for social networks', *Journal of the American Statistical Association* **85**(409), 204–212.

van der Maaten, L. & Hinton, G. (2008), 'Visualizing data using t-sne', *Journal of Machine Learning Research* **9**(Nov), 2579–2605.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Philip, S. Y. (2021), 'A comprehensive survey on graph neural networks', *IEEE Transactions on Neural Networks and Learning Systems* **32**(1), 4–24.

Xu, M. (2021), 'Understanding graph embedding methods and their applications', *SIAM Review* **63**(4), 825–853.

Yang, Z., Algesheimer, R. & Tessone, C. J. (2015), 'Evaluating link prediction methods', *Knowledge-Based Systems* **74**, 87–96.

Yao, L., Mao, C. & Luo, Y. (2019), Graph convolutional networks for text classification, *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'.

Ying, Z., Bourgeois, D., You, J., Zitnik, M. & Leskovec, J. (2019), Gnnexplainer: Generating explanations for graph neural networks, *in* 'Advances in Neural Information Processing Systems (NeurIPS)'.

You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z. & Shen, Y. (2020), Graph contrastive learning with augmentations, *in* 'Advances in Neural Information Processing Systems (NeurIPS)'.

Zhang, Z., Cui, P. & Zhu, W. (2020), 'Deep learning on graphs: A survey', *IEEE Transactions on Knowledge and Data Engineering* **34**(1), 249–270.