Complex & Intelligent Systems (2024) 10:7333–7348
https://doi.org/10.1007/s40747-024-01545-6

ORIGINAL ARTICLE

Unsupervised Graph Representation Learning with Inductive Shallow Node Embedding

Richárd Kiss[1] · Gábor Szűcs[1]

Received: 26 March 2024 / Accepted: 21 June 2024 / Published online: 12 July 2024

Abstract
Network science has witnessed a surge in popularity, driven by the transformative power of node representation learning for diverse applications like social network analysis and biological modeling. While shallow embedding algorithms excel at capturing network structure, they face a critical limitation—failing to generalize to unseen nodes. This paper addresses this challenge by introducing Inductive Shallow Node Embedding—as a main contribution—pioneering a novel approach that extends shallow embeddings to the realm of inductive learning. It has a novel encoder architecture that captures the local neighborhood structure of each node, enabling effective generalization to unseen nodes. In the generalization, robustness is essential to avoid degradation of performance arising from noise in the dataset. It has been theoretically proven that the covariance of the additive noise term in the proposed model is inversely proportional to the cardinality of a node's neighbors. Another contribution is a mathematical lower bound to quantify the robustness of node embeddings, confirming its advantage over traditional shallow embedding methods, particularly in the presence of parameter noise. The proposed method demonstrably excels in dynamic networks, consistently achieving over 90% performance on previously unseen nodes compared to nodes encountered during training on various benchmarks. The empirical evaluation concludes that our method outperforms competing methods on the vast majority of datasets in both transductive and inductive tasks.

Keywords Node embedding · Graph representation learning · Inductive learning · Dynamic networks · Node2Vec · GraphSAGE

Introduction

Network science has experienced a surge in interest, fueled by the transformative potential of node representation learning [1] for diverse applications like bioinformatics [2, 3], chemoinformatics [4, 5], recommendation systems [6–8], social network analysis [9], and more. At the heart of this field lies the ability to capture meaningful representations of nodes within complex networks [10].

Shallow node embedding techniques, exemplified by the popular Node2Vec [11] algorithm, play a crucial role in this endeavor [12, 13]. These methods excel at capturing the structural information within networks. However, they are inherently transductive, limiting their ability to generalize to unseen nodes, a significant drawback in dynamic networks or scenarios with incomplete initial knowledge.

GraphSAGE (Graph SAmple and aggreGatE) [14], a pioneering work in inductive node representation learning, addresses this limitation by employing Message Passing Neural Networks (MPNN). However, the performance of MPNNs is heavily reliant on the quality and availability of node features. In scenarios where node features are missing, unreliable, or not informative for the task at hand, GraphSAGE's effectiveness can be diminished.

Our research addresses a notable gap in the field: solving inductive tasks in graph analysis where GraphSAGE struggles, particularly because it relies heavily on node features for effective learning. We sought to develop a solution capable of learning without these features and making predictions on new nodes, achieving true inductive capabilities, unlike the transductive nature of Node2Vec. Additionally, our study aims to fill a void in the literature by providing a robust theo-

✉ Gábor Szűcs
szucs@tmit.bme.hu

Richárd Kiss
richard.kiss@edu.bme.hu

[1] Department of Telecommunications and Artificial Intelligence, Budapest University of Technology and Economics, Műegyetem rkp. 3., Budapest 1111, Hungary

retical analysis of the effects of noise, which has been largely overlooked in previous research. The aim of this work was to bridge these gaps and advance the understanding and application of graph-based machine learning.

To address these limitations, a novel approach, so called Inductive Shallow Node Embedding (ISNE) was introduced, that extends shallow node embeddings to inductive learning. Unlike traditional methods, ISNE employs a unique encoder architecture that captures the local neighborhood structure of each node, enabling it to generalize effectively to unseen nodes. This makes ISNE particularly valuable for dynamic networks where the network structure evolves over time.

The main contributions of this work are as follows:

- *Novel encoder design*: The introduction of a new encoder that captures local neighborhood structures for inductive learning is a significant advancement. This encoder moves beyond the limitations of traditional lookup tables and message-passing frameworks.
- *Inductive learning for shallow embeddings*: Extending shallow embeddings to inductive learning creates a new category of algorithms that combine the simplicity and effectiveness of shallow methods with the generalizability of inductive approaches.
- *Theoretical insights into robustness*: Providing a mathematical proof of the robustness of the embeddings under parameter noise offers a new understanding of how embeddings can remain reliable in noisy and dynamic environments. This can influence future research on the robustness of machine learning models.
- *Hybrid embedding and attribute utilization*: Demonstrating the effective combination of structural embeddings with node attributes creates a hybrid approach that leverages the strengths of both. This knowledge can inform the design of future algorithms that need to balance structural and attribute information.
- *Comparative analysis on new nodes*: The ability of ISNE to adapt to dynamic networks is demonstrated, consistently achieving over 90% performance on previously unseen nodes compared to nodes encountered during training across various benchmarks.
- *Comprehensive empirical evaluation*: Extensive experiments are conducted on multiple datasets to validate ISNE's performance in both transductive and inductive settings, showcasing its superiority over traditional methods and state-of-the-art inductive algorithms like GraphSAGE.

By addressing the inherent limitations of traditional shallow embedding methods and advancing the capabilities of inductive learning, ISNE represents a significant step forward in the field of network science. This work lays the foundation for more robust and adaptable node representation learning techniques, capable of handling the complexities of dynamic and evolving networks.

In the remainder of this paper, the related works are reviewed in Sect. "Related Work", the node representation learning framework with shallow encoders is presented in Sect. "Node representation learning with shallow encoders. A novel perspective that addresses the limitations and a method called Inductive Shallow Node Embedding are introduced in Sect. "Inductive Shallow Node Embedding". A theoretical analysis of the robustness of the proposed method is provided in Sect. "Theoretical analysis of robustness". In Sect. "Empirical results", the empirical performance of the proposed method on both transductive and inductive tasks is evaluated, demonstrating promising results. Finally, the last section concludes the paper with remarks on future research directions.

Related work

Shallow embedding methods

Traditional shallow embedding methods excel at capturing network structure through techniques like random walks (DeepWalk [15], Node2Vec [11]) or preserving proximity relationships (LINE—Large-scale Information Network Embedding [16], NetMF—NETwork embedding as Matrix Factorization [17], GraRep—GRAph REPresentations [18], PTE—Predictive Text Embedding [19]). However, these methods rely on a lookup table encoder architecture, limiting their ability to generalize to unseen nodes not encountered during training. Applications include recommendation systems [10, 20], financial fraud detection [21], learning text representations [19], and predicting miRNA-disease associations [22, 23].

Inductive node representation learning

GraphSAGE [14] represents a significant advancement by introducing a message-passing framework for unsupervised inductive node representation learning. This approach allows GraphSAGE to effectively handle unseen nodes, making it particularly valuable for dynamic networks. GraphSAGE has been successfully applied to various tasks, including knowledge graph completion [24], recommendation systems [25], intrusion detection [26], prediction of molecular toxicity [27], financial portfolio optimization [28], and traffic speed forecasting [29]. However, GraphSAGE's effectiveness depends on the quality and availability of node attributes, which can be a limitation in certain scenarios.

ISNE builds upon these approaches by introducing a novel inductive encoder that captures network structure without relying on a static lookup table encoder or requiring high-quality node features. This enables ISNE to effectively generalize to unseen nodes in dynamic networks, even when node attributes are unavailable or unreliable.

Node representation learning with shallow encoders

Notation

To enhance clarity and understanding, consistent notation throughout the paper is established:

Variables

- $\mathbf{A}$: Matrix, $\mathbf{A}[i, j]$ denotes the element in the i-th row and j-th column
- $\mathbf{A}^2$: Element-wise square of matrix $\mathbf{A}$, $\mathbf{A}^2[i, j] = (\mathbf{A}[i, j])^2$
- v_n: Specific version of vector v (subscript denotes version)
- $v_n^\top$: Transpose of vector v_n
- $\mathbb{1}$: Vector with all elements equal to 1
- $\mathbb{0}$: Vector with all elements equal to 0

Graph Properties

- $G = (V, E)$: Graph with nodes V and undirected edges $E \in V \times V$
- $\mathcal{N}_i$: Set of neighboring nodes for node i
- $|\mathcal{N}_i|$: Number of neighbors for node i
- $\mathcal{N}_{i,j}$: Intersection of neighbors between nodes i and j
- $|\mathcal{N}_{i,j}|$: Number of common neighbors of i and j

Embedding and Similarity

- $f : V \rightarrow \mathbb{R}^D$: Function mapping nodes to D-dimensional representations (encoder)
- $f(i)$: Embedding vector of node i
- $\tilde{f}(i)$: Embedding vector of node i with a noise augmented encoder function f
- $s_f : V \times V \rightarrow \mathbb{R}$: Node similarity function based on encoder f
- $s_{\tilde{f}} : V \times V \rightarrow \mathbb{R}$: Node similarity function based on the noise augmented encoder function $\tilde{f}$

The node representation learning framework

Node representation learning aims to discover low-dimensional vector representations (embeddings) for nodes in a network. These embeddings capture the inherent structure and relationships within the network, allowing them to be readily utilized in various downstream tasks such as node classification, link prediction, and community detection.

The core concept of this learning process revolves around learning an encoder function with learnable weights. This function takes a node as input and maps it to its corresponding embedding in a low-dimensional space. Different algorithms employ distinct encoder functions, similarity metrics in the embedding space, and methods for defining node similarity within the graph.

Ideally, this process leads to embeddings where geometric relationships between nodes in the low-dimensional space accurately reflect the structural relationships within the original network. Nodes that exhibit higher similarity within the network should be positioned closer together in the embedding space, and vice versa.

Shallow encoders

Shallow Encoder Algorithms originally represent nodes using a lookup table, which assigns a unique, pre-allocated embedding vector to each node. This function, denoted as f, essentially maps a node v in the network to its corresponding embedding vector in the low-dimensional space: $f(v) = \theta_v$. Algorithms that utilize lookup table encoders suffer from two key limitations:

1. *Transductivity*: Due to its reliance on a pre-defined lookup table, Node2Vec [11] is inherently transductive. This means the model cannot generalize to unseen nodes, which were not present during the training process. This limitation hinders its applicability in scenarios involving dynamic networks or tasks requiring predictions for new nodes.
2. *Static Embeddings*: The learned representations generated by Node2Vec are static. Any changes to the network structure, such as adding or removing edges, do not trigger updates to the existing node embeddings. This lack of adaptability can be problematic in real-world networks that often exhibit dynamic changes.

Inductive shallow node embedding

Inductive Shallow Node Embedding (ISNE) offers a novel perspective that addresses the limitations explained in Sect. "The node representation learning framework". ISNE leverages a novel encoder function that overcomes the challenges associated with both unseen nodes and dynamic network structures. This novel design empowers ISNE to:

- *Generalize to unseen nodes*: Unlike transductive methods, ISNE can effectively represent even nodes not present during training.

- *Adapt to dynamic networks*: ISNE representations can adjust to changes in the network structure, making them suitable for evolving network scenarios.
- *Function independently of node attributes*: ISNE embeddings are constructed solely based on the network structure, eliminating the dependency on potentially unreliable or unavailable attribute information.

Furthermore, ISNE retains the flexibility to incorporate node attributes by simply concatenating them to the existing ISNE embeddings. This allows users to leverage the strengths of both network structure and node attributes, potentially leading to even more robust and informative representations. The following section delves deeper into the details of the proposed ISNE method, including its novel encoder function and its theoretical properties.

Methodology

Unlike traditional shallow embedding methods that rely on lookup tables, the proposed Inductive Shallow Node Embedding (ISNE) method leverages a novel encoder function to construct node embeddings. This function operates based on the immediate neighbors of each node, as captured by the neighborhood set denoted by $\mathcal{N}_v$. The core equation for the ISNE encoder is presented as follows:

$$h(v) = \frac{1}{|\mathcal{N}_v|} \sum_{n \in \mathcal{N}_v} \theta_n \tag{1}$$

In this equation, $h(v)$ represents the embedding vector of node v, and the summation iterates through all neighbors n within its neighborhood set. This design ensures that the embedding of a node is informed by the parameters of its immediate neighbors, effectively capturing the local network structure around each node.

This approach offers several key advantages:

- *Dynamic updates*: Whenever a new edge is added to the network, the neighborhood set of affected nodes (i.e., $\mathcal{N}_j$ for specific nodes j) is updated accordingly. By recalculating $h(v)$ for these nodes, the ISNE embeddings automatically reflect the latest network structure changes within their local neighborhoods.
- *Handling unseen nodes*: The ISNE framework is capable of generating embeddings for previously unseen nodes, provided their connections are known. By incorporating these connections into the neighborhood set during the encoding process (i.e., adding them to $\mathcal{N}_v$ for the unseen node), ISNE can effectively estimate their embeddings.
- *Inductive learning*: This unique design empowers ISNE to perform inductive learning tasks. By relying solely on the network structure and generalizing from known information, the model can infer embeddings for unseen and modified data points, significantly expanding its applicability in dynamic network settings and demonstrating adaptability to evolving graph structures.

In essence, the ISNE encoder overcomes the limitations of lookup tables by enabling dynamic updates, handling unseen nodes, and facilitating inductive learning tasks, thereby establishing itself as a valuable tool for various network analysis applications.

Theoretical analysis of robustness

This section investigates the robustness of ISNE compared to the traditional lookup table encoder in the presence of parameter noise. This can be achieved by introducing zero-mean additive noise, denoted by z_n, into the model parameters θ_n. Each z_n is independently and identically distributed (i.i.d.) following a common multivariate Gaussian distribution with zero mean and covariance matrix Σ. We denote the noise-corrupted versions of the lookup table and ISNE encoders as $\tilde{f}$ and $\tilde{h}$, respectively:

$$\tilde{f}(i) = \theta_i + z_i \tag{2}$$

$$\begin{aligned}\tilde{h}(i) &= \frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} \theta_n + z_n \\ &= \underbrace{\frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} \theta_n}_{h(i)} + \underbrace{\frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} z_n}_{\epsilon_i} \qquad (3)\end{aligned}$$

Here, $\epsilon_i = \frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} z_n$ captures the additive noise introduced by the neighborhood aggregation in the ISNE encoder. Since ϵ_i is the sum of i.i.d. Gaussian random variables, it also follows a multivariate Gaussian distribution with zero mean. However, it's important to note that the noise vectors ϵ_{n_1} and ϵ_{n_2} for different nodes n_1 and n_2 might not be independent. This is because the aggregation in ϵ_i involves noise terms z from potentially overlapping neighborhoods, i.e., $\mathcal{N}_{n_1} \cap \mathcal{N}_{n_2} \neq \emptyset$.

Representation robustness

Covariance analysis is a valuable tool for evaluating the robustness of node representations. This analysis focuses on the inherent noise level within the ISNE embeddings.

Theorem 1 *The covariance of the additive noise term ϵ_i in the ISNE model is inversely proportional to the cardinality of the neighbors of i:*

$$\mathbb{E}\left(\epsilon_i \epsilon_i^{\top}\right) = \frac{1}{|\mathcal{N}_i|}\Sigma \tag{4}$$

Proof A formal proof of this theorem is provided in Appendix B.1.3 □

A consequence of Theorem 1 is that nodes with a greater number of neighbors tend to have lower noise levels in their representations. This is because the averaging effect inherent in processing information from a larger neighborhood helps to reduce the impact of individual noise components. Consequently, nodes with robust representations possess higher reliability and perform better in downstream tasks that utilize these representations.

Bias and variance of representation similarity

Many downstream tasks in network analysis rely on the similarity between node representations, rather than the node representations themselves. In this section, the bias and variance of the similarity functions $s_{\tilde{f}}$ and $s_{\tilde{h}}$ are investigated, which measure the dot-product similarity between representations obtained with noise-corrupted encoders.

Bias

Theorem 2 *The embedding similarity function $s_{\tilde{f}}(i, j)$ obtained from the noise-corrupted lookup table encoder is unbiased if $i \neq j$.*

Proof A formal proof of this theorem is provided in Appendix B.2 □

Theorem 3 *The embedding similarity function $s_{\tilde{h}}(i, j)$ obtained from the noise-corrupted ISNE encoder exhibits bias proportional to the number of common neighbors between nodes i and j:*

$$\mathbb{E}\left(s_{\tilde{h}}(i, j)\right) = s_h(i, j) + \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i||\mathcal{N}_j|} Tr(\Sigma) \tag{5}$$

Proof A formal proof of this theorem is provided in Appendix B.4 □

Here, $|\mathcal{N}_{i,j}|$ denotes the number of common neighbors between nodes i and j, and $\mathrm{Tr}(\Sigma)$ represents the trace of the covariance matrix Σ associated with the noise.

While the bias term introduced in Theorem 3 causes the similarity score to deviate from the exact similarity measure of ISNE, it does not necessarily invalidate its utility as an indicator of node proximity. Nodes with a higher number of shared neighbors tend to exhibit a stronger similarity under this bias. The empirical results in Sect. "Evaluation of transductive task performance" demonstrate that this bias does not adversely affect the performance of downstream tasks.

Variance

Next, the variance of the representation similarity functions is analyzed, which measures the spread of the similarity scores around their expected values.

Theorem 4 (Variance Bound) *The variance of the representation similarity function $s_{\tilde{h}}$ is upper-bounded by a constant factor multiplied by the variance of the similarity obtained using the noise-corrupted lookup table encoder, $s_{\tilde{f}}$, for the same nodes:*

$$\mathrm{Var}\left(s_{\tilde{h}}(i, j)\right) \leq \frac{3}{K} \mathrm{Var}\left(s_{\tilde{f}}(i, j)\right) \tag{6}$$

where, $K = \min\{|\mathcal{N}_i|, |\mathcal{N}_j|\}$ represents the minimum number of neighbors between nodes i and j.

Proof A formal proof of this theorem is provided in Appendix B.6 □

Theorem 4 establishes a valuable relationship between the variances of the similarity functions. It shows that the variance of the similarity obtained using ISNE is guaranteed to be less than or equal to a constant factor multiplied by the variance of the similarity obtained using the lookup table encoder, for any two nodes in the network. This upper bound serves as a tool for assessing the robustness of downstream tasks that rely on representation similarity, such as Information Retrieval. In the subsequent section, the robustness of such tasks in the presence of noise is explored.

Robustness of information retrieval

Impact of noise on retrieval

When seeking similar entities in a network, the similarity between their representations often serves as a crucial metric for retrieving relevant items. Understanding how noise affects the retrieval process sheds light on the robustness of the order of node similarities in the presence of parameter noise.

Let q denote a query node, and r and n represent candidate nodes that are relevant and non-relevant to the query, respectively. Information Retrieval (IR) aims to retrieve the relevant item, meaning we want $s(q, r) > s(q, n)$, where s denotes the similarity function. The effect of noise can be modelled on the similarity scores by using zero-mean Gaussian noise vectors $\delta_r \sim N(0, \sigma_r^2)$ and $\delta_n \sim N(0, \sigma_n^2)$ for the relevant and non-relevant nodes, respectively. The robustness of the retrieval process can be assessed by calculating the probability of retrieving the wrong node, given by Eq. 7.

$$P(s_{\tilde{f}}(q, r) \leq s_{\tilde{f}}(q, n)) = \Phi\left(\frac{s_f(q, n) - s_f(q, r)}{\sqrt{\sigma_r^2 + \sigma_n^2}}\right) \tag{7}$$

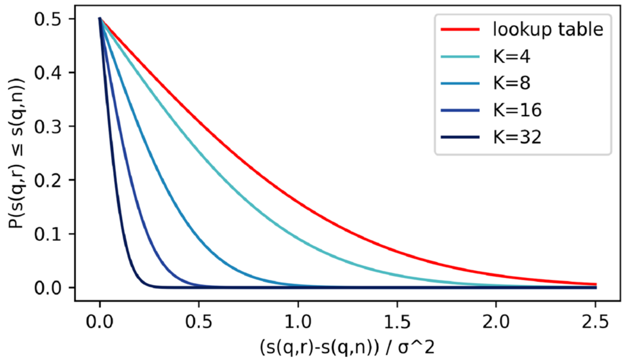

Fig. 1 Misclassification probability comparison between lookup table (red) and ISNE models (blue). The *x*-axis represents the normalized true similarity difference between relevant and non-relevant nodes. The *y*-axis indicates the probability of incorrect retrieval

Here, Φ epresents the cumulative distribution function (CDF) of the standard normal distribution. The denominator follows from the fact $\delta_n - \delta_r = \delta \sim N(0, \sigma_r^2 + \sigma_n^2)$.

Advantage of ISNE in retrieval

Using Theorem 4, the probability of retrieving the wrong node in the ISNE model can be rewritten as:

$$P(s_{\tilde{h}}(q,r) \leq s_{\tilde{h}}(q,n)) = \Phi\left(\frac{s_h(q,n) - s_h(q,r)}{\sqrt{\frac{3}{K}}\sqrt{\sigma_r^2 + \sigma_n^2}}\right) \quad (8)$$

Fig. 1 compares the misclassification probabilities for the lookup table and ISNE models. The *x*-axis denotes the normalized true similarity difference between relevant and non-relevant nodes, while the *y*-axis illustrates the probability of incorrect retrieval. As the minimum node degree (K) increases, the ISNE model demonstrates a significant improvement in retrieval robustness compared to the lookup table model. This is evident in the steeper decrease in misclassification probability for the ISNE model with increasing K.

The ISNE model's advantage stems from its lower noise variance, which scales inversely with K compared to the constant noise variance in the lookup table model. In simpler terms, the noise in ISNE has a lesser impact on the final similarity score as the number of neighbors (K) increases.

It is important to note that the derivation intentionally disregarded the bias term introduced in Theorem 3. This omission is justified because the underlying assumption—that relevant items tend to have a higher degree of overlap compared to non-relevant ones—is generally valid. If this assumption holds, the true difference between $s(q,r)$ and $s(q,n)$ would be even greater, further bolstering the ISNE model's robustness. Consequently, neglecting the bias term doesn't invalidate our conclusion that the ISNE model exhibits superior robustness in noisy environments.

Table 1 Properties of the datasets used in the experiments

Name	Nodes	Edges	Classes	Features
ARXIV [30]	169,343	1,166,243	40	128
CORA [31]	19,793	126,842	70	8710
PUBMED [31]	19,717	88,648	3	500
BLOGCATALOG [32]	5196	343,486	6	8189
WIKICS [33]	11,701	431,726	10	300

Empirical results

This section delves into the performance of Inductive Shallow Node Embedding (ISNE) through a series of experiments designed to address the following key research questions:

1. *Performance in Transductive Tasks:* Do ISNE embeddings maintain performance comparable to the traditional lookup table encoder in transductive tasks, where all test nodes are seen during training?
2. *Inductive Reasoning:* Can ISNE maintain comparable classification accuracy on unseen nodes to its performance on nodes seen during training?
3. *Comparison with Inductive Algorithms:* How does ISNE compare to other state-of-the-art inductive algorithms designed specifically for handling unseen data?

By addressing these research questions through carefully designed experiments, we aim to gain a comprehensive understanding of the effectiveness of composite embeddings across various task settings.

Datasets

To comprehensively evaluate the effectiveness and generalizability of Inductive Shallow Node Embedding (ISNE), a diverse range of datasets were employed. These datasets, summarized in Table 1, encompass various network types and exhibit distinct structural characteristics.

- ARXIV [30]: The dataset represents the citation network of papers on arXiv.org, covering different subject areas. Nodes represent papers, edges represent citations, and each node is associated with a feature vector derived from the paper's text. The task is to classify papers into 40 different subject areas.
- CORA [31]: The dataset is a well-known benchmark for citation network analysis. It comprises scientific publications classified into 70 different research areas. Nodes represent papers, and edges denote citation relationships between them. Each node has a feature vector based on

Table 2 Average Accuracy Scores (%) in unattributed Transductive Node Classification

Blog					
	Arxiv	Catalog	Cora	PubMed	WikiCS
Node2Vec	0.475	0.518	0.489	0.706	0.574
LINE	**0.614**	0.633	0.528	0.759	0.758
ISNE (ours)	0.557	**0.657**	**0.569**	**0.774**	**0.762**

Bold values present the best performers and underlined values highlight the statistically significant best performer with a confidence level of 95%

the paper's abstract, consisting of a bag-of-words representation.

- PubMed [31]: The dataset contains a citation network of scientific publications in the biomedical domain. Nodes represent papers, and edges indicate citation links. The task involves classifying papers into three classes related to different diseases. Node features are derived from the Term Frequency-Inverse Document Frequency (TF-IDF) of words in the paper abstracts.
- BlogCatalog [32]: The dataset is a social network where nodes represent users, and edges represent the friendship relationships between them. The classification task is to assign users to one of six predefined categories. Each user has a descriptor vector as node feature.
- WikiCS [33]: The dataset consists of a citation network of computer science articles from Wikipedia. Nodes represent articles, and edges denote hyperlinks between them. The classification task involves categorizing articles into 10 different computer science topics. Each node has a feature vector representing the article's content, captured through a pre-trained language model.

Evaluation of transductive task performance

This section addresses research question 1, which investigates the performance of the proposed Inductive Shallow Node Embedding (ISNE) method compared to traditional lookup table encoders in transductive tasks. Transductive tasks involve training and testing on the same set of nodes, aiming to evaluate the models' ability to capture inherent network structure and perform well on tasks like node classification.

The detailed experimental setup and model configurations are provided in Appendix A.1 for reference.

Table 2 shows that ISNE significantly outperforms both LINE and Node2Vec on the BlogCatalog, Cora, PubMed and WikiCS datasets. This superior performance demonstrates ISNE's ability to effectively capture network structure and produce robust embeddings for node classification tasks in these environments.

However, on the Arxiv dataset, ISNE falls short compared to LINE (while still performing significantly better than Node2Vec). This dataset is characterized by extreme sparsity, being an order of magnitude sparser than the other datasets considered. Additionally, the Arxiv dataset exhibits very low node homophily (42%). These factors degrade ISNE's performance, as the method relies on the information from neighboring nodes. High sparsity results in a low number of neighbors, and low homophily means that these neighbors are often less informative about the node's class, impacting the overall effectiveness of ISNE in such conditions.

In this experiment, ISNE parameters were set to capture a more global node similarity through long random walks, while LINE preserves only 1st and 2nd proximities. It is possible that if the ISNE parameters were adjusted to focus more on local structures (such as LINE), it could achieve a similar level of performance to LINE in datasets with high sparsity and low homophily like Arxiv.

Evaluation of inductive task performance

This section tackles research questions 2 and 3, focusing on the performance of ISNE in inductive tasks involving unforeseen nodes. Inductive tasks require models to generalize their knowledge and perform well on unseen data, making them particularly challenging. To comprehensively assess ISNE's effectiveness and generalizability, evaluations on both attributed and unattributed node classification tasks were conducted. The specific details of the experimental setup and model configurations can be found in Appendix A.2 for reference.

Unattributed node classification

This section addresses the ability of ISNE to handle unseen nodes in unattributed node classification tasks. Evaluating performance on unseen nodes is crucial to assess the generalizability and extrapolation capabilities of the model.

Table 3 compares the average accuracy scores achieved by ISNE on both training nodes and unseen nodes across different datasets. The last row presents the relative performance of unseen nodes compared to training nodes.

As shown in Table 3, ISNE demonstrates remarkable generalization capabilities for unseen nodes. The model consistently achieves accuracy exceeding 90% across all datasets, showcasing its ability to effectively construct representations for unseen data points. Notably, the minimal performance drop ($< 2\%$) in the BlogCatalog, PubMed, WikiCS datasets further emphasizes the robustness and generalizability of ISNE.

These findings highlight the effectiveness of ISNE in handling unseen nodes, making it a valuable tool for tasks requiring models to perform well on new and evolving data.

Table 3 Average accuracy scores (%) in unattributed inductive node classification

Blog					
	Arxiv	Catalog	Cora	PubMed	WikiCS
Training nodes	0.557	0.657	0.569	0.774	0.762
Unseen nodes	0.508	0.648	0.529	0.764	0.751
Relative	91.2%	98.6%	92.9%	98.7%	98.6%

Table 4 Average accuracy scores (%) in Attributed Inductive Node Classification

Blog					
	Arxiv	Catalog	Cora	PubMed	WikiCS
Baseline	0.557	0.834	0.545	0.844	0.775
GraphSAGE	0.583	0.783	0.553	0.830	0.808
ISNE (ours)	**0.585**	**0.873**	**0.605**	**0.865**	**0.816**

Bold values present the best performers and underlined values highlight the statistically significant best performer with a confidence level of 95%

Attributed node classification

This section delves into the performance of ISNE on attributed node classification tasks. In this setting, both the model and the benchmark method, GraphSAGE, utilize the same information: ISNE embeddings concatenated with node attributes and GraphSAGE embeddings, respectively. This ensures a fair comparison by eliminating bias introduced by different attribute usage. Additionally, a baseline utilizing node attributes only is included to assess the inherent predictive power of attributes, independent of embedding techniques.

Table 4 summarizes the average accuracy scores achieved by the baseline, GraphSAGE, and ISNE on attributed inductive node classification tasks across different datasets.

As shown in Table 4, ISNE consistently outperforms the baseline and GraphSAGE across all datasets, with statistically significant improvements observed in BlogCatalog, Cora, and PubMed. These findings highlight the effectiveness of combining ISNE embeddings and node attributes for classification tasks.

While GraphSAGE integrates node attributes into its embedding generation process, its performance does not always surpass the baseline utilizing attributes alone. This suggests that the unsupervised learning approach used by GraphSAGE may not consistently extract optimal information for node classification as it needs to strike a balance between preserving the original node attribute information and capturing structural relationships within the network.

In contrast, ISNE effectively captures structural information through its encoder function, and the undistorted node attributes can be seamlessly incorporated for downstream tasks. This allows ISNE to leverage both the inherent structural patterns within the network and the rich information encapsulated within the node attributes.

In conclusion, the experimental results demonstrate the competitive advantage of ISNE over both GraphSAGE and attribute-based methods. The significant performance improvements achieved by ISNE showcase its potential as a powerful tool for attributed node classification tasks, offering a valuable alternative to existing approaches.

Limitations

While the proposed method demonstrates significant advancements in handling unseen nodes and adapting to dynamic network structures, several limitations must be acknowledged. It is important to note that these limitations are not unique to ISNE but are inherent to any method for unattributed node representation learning. When additional information is unavailable, there are inherent constraints on performance.

Dependence on neighboring nodes

ISNE embeds new nodes based on the parameter vectors of their neighbors which are learned ahead of time. If a large number of new nodes are introduced to the network and they primarily form edges among themselves rather than with previously existing nodes, the new nodes' embeddings may lack richness and informativeness. This can lead to degraded performance in scenarios where new nodes are densely interconnected but sparsely connected to the existing network, as the embeddings of new nodes may not capture the broader network structure effectively. However, in many real-world graphs, the phenomenon of preferential attachment is observed, where new nodes tend to connect to high-degree nodes [34]. This natural tendency helps mitigate the issue, as connections to well-established, high-degree nodes can enrich the embeddings of new nodes, ensuring they reflect the broader network structure.

Limited initial information

ISNE relies on the neighborhood information of new nodes for embedding. When new nodes have limited initial connections, especially in the early stages of their introduction, the embeddings generated may be suboptimal. This can result in reduced accuracy and effectiveness of the embeddings in capturing the true position and role of new nodes within the network, particularly when node attributes are sparse or unavailable.

These limitations are inherent to unattributed node representation learning methods. Without additional information such as node attributes or external context, it is challenging to achieve better performance.

Conclusion

This paper introduced Inductive Shallow Node Embedding (ISNE), a novel approach that addresses the limitations of existing shallow embedding methods for learning node representations in graphs. Unlike traditional methods that rely on lookup tables, ISNE utilizes an encoder specifically designed to capture the local neighborhood structure of each node. This approach enables ISNE to effectively generalize to unseen nodes, making it particularly valuable for dynamic network settings.

Comprehensive evaluation across various tasks and datasets showcases the effectiveness of ISNE:

1. *Competitive performance in transductive tasks*: ISNE achieves comparable or better performance compared to traditional methods like Node2Vec and LINE in transductive node classification tasks, demonstrating its ability to capture inherent network structure.
2. *Superior performance in handling unseen nodes*: ISNE exhibits remarkable generalization capabilities, maintaining high accuracy on unseen nodes in inductive tasks. This highlights its ability to construct high-quality representations for new data points.
3. *Effective utilization of node attributes*: When combined with node attributes, ISNE consistently outperforms the state-of-the-art method, GraphSAGE, in attributed node classification tasks. This demonstrates the effectiveness of ISNE in leveraging both structural information and node attributes for improved performance.

Beyond empirical findings, we also presented a theoretical analysis of the robustness of ISNE to parameter noise:

1. *Covariance of Additive Noise*: One of our contributions is the finding that the covariance of the additive noise term in the ISNE model is inversely proportional to the cardinality of a node's neighbors. This implies that nodes with more neighbors experience lower noise levels in their representations due to the averaging effect of having a larger neighborhood.
2. *Biased Node Similarity*: Another theoretical contribution pertains to the bias and variance analysis. The embedding similarity function obtained from the noise-corrupted ISNE encoder exhibits bias proportional to the common neighbors of two nodes. This can often be helpful by increasing embedding similarity within nodes that have high neighborhood overlap.
3. *Variance of Node Similarity*: Our analysis shows that the variance of the representation similarity function in ISNE is upper-bounded by a factor inversely proportional to the minimum number of neighbors of the nodes involved. This means that ISNE embeddings tend to have lower variance, leading to more stable and reliable similarity measures, especially for nodes with larger neighborhoods.

In conclusion, ISNE establishes itself as a versatile and robust approach for inductive node representation learning. Its ability to handle unseen nodes, effectively utilize node attributes, and achieve strong theoretical guarantees positions ISNE as a promising tool for various graph mining applications, particularly in dynamic and evolving networks.

Our results have the potential for multiple applications, future research directions include Explainable Artificial Intelligence, recommendation systems, social network analysis, citation networks, graph convolutional networks in computer vision [35], combinatorial optimization [36], and robot swarm control [37].

Appendix A: Experiment details

This section details the experimental setup and configuration parameters used for training the embedding models.

A.1 Transductive setting

To assess model performance, we employed a 5-fold cross-validation strategy. After training the model and generating the embeddings, we utilized a K-Nearest Neighbors (KNN) classifier for label prediction on the test nodes. This involved a separate train/test split on the embeddings themselves. The KNN classifier utilized dot product similarity to identify the 15 nearest neighbors within the training set embeddings.

A.2 Inductive setting

In the inductive setting, the model is evaluated on unseen nodes. We split the nodes into training and test sets, where the test set comprises nodes not present during training. Similar to the transductive setting, we employed a 5-fold cross-validation approach with a KNN classifier.

For attributed inductive node classification tasks, the ISNE embeddings are augmented with the inherent node features.

Table 5 One-sided related T-test P-values for ISNE's higher accuracy scores across various datasets

Dataset	Other method	P value
BLOGCATALOG	N2V	0.000976%
BLOGCATALOG	LINE	0.878058%
ARXIV	N2V	0.000592%
ARXIV	LINE	99.999557%
CORA	N2V	0.000424%
CORA	LINE	0.014846%
PUBMED	N2V	2.876472%
PUBMED	LINE	0.149597%
WIKICS	N2V	0.000278%
WIKICS	LINE	7.716635%

Table 6 One-Sided Related T test P values for ISNE's higher accuracy scores across various datasets

Dataset	Model 1	P value
BLOGCATALOG	GraphSAGE	0.001334%
BLOGCATALOG	Baseline	0.026882%
ARXIV	GraphSAGE	22.032741%
ARXIV	Baseline	0.006854%
CORA	GraphSAGE	0.033234%
CORA	Baseline	0.000788%
PUBMED	GraphSAGE	0.001162%
PUBMED	Baseline	0.008831%
WIKICS	GraphSAGE	3.289872%
WIKICS	Baseline	0.006987%

This is achieved by concatenating the embeddings and features along their dimension. To balance the influence of these two information sources, we introduce an α parameter that determines the relative weight given to each component in the final representation.

A.3 Model configurations

Our models were trained with the following configurations:

- *Embedding Dimensionality* The dimensionality of the embedding space varied depending on the dataset. For ArXiv, we used 128 dimensions, while BlogCatalog, Cora, PubMed, and WikiCS all employed a 64-dimensional embedding space. Notably, all models inherit their embedding dimensionality from the chosen value specified here.
- *LINE* We opted for a factorization-based implementation of LINE as described in [17].
- *Node2Vec and ISNE* These models share several hyperparameters. They utilize a context size of 5, a negative sample ratio of 1, and are trained for 200 epochs.
- *GraphSAGE* This model leverages the dimensionality of the node features as the size of its input layer. It possesses two hidden layers, each with a dimensionality of 512, and is trained for 50 epochs.

Appendix B: Derivation of results

B.1 Properties of ϵ

B.1.1 Expectation of ϵ

$$\mathbb{E}\left(\epsilon_i\right) = 0 \tag{B1}$$

Proof Expressing ϵ_i in terms of z and leveraging the linearity of expectation, we can establish the validity of the given statement:

$$\mathbb{E}\left(\epsilon_i\right) = \mathbb{E}\left(\frac{1}{|\mathcal{N}_i|}\sum_{n\in\mathcal{N}_i} z_n\right) = \frac{1}{|\mathcal{N}_i|}\sum_{n\in\mathcal{N}_i}\underbrace{\mathbb{E}\left(z_n\right)}_{0} = 0 \tag{B2}$$

□

B.1.2 Expectation of the inner product of ϵ

$$\mathbb{E}\left(\epsilon_i^\top \epsilon_j\right) = \underbrace{\frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i||\mathcal{N}_j|}}_{Q}\operatorname{Tr}\left(\Sigma\right) \tag{B3}$$

Proof Initially, we write ϵ_i and ϵ_j in terms of z as follows:

$$\mathbb{E}\left(\epsilon_i^\top \epsilon_j\right) = \mathbb{E}\left(\left(\frac{1}{|\mathcal{N}_i|}\sum_{n_1\in\mathcal{N}_i} z_{n_1}\right)\left(\frac{1}{|\mathcal{N}_j|}\sum_{n_2\in\mathcal{N}_j} z_{n_2}\right)\right) \tag{B4}$$

We begin by factoring out the constants, consolidating the product of sums into a sum of products. Utilizing the linearity of expectation, we subsequently move the expectation operation inward:

$$= \frac{1}{|\mathcal{N}_i||\mathcal{N}_j|}\sum_{n_1\in\mathcal{N}_i}\sum_{n_2\in\mathcal{N}_j}\mathbb{E}\left(z_{n_1}^\top z_{n_2}\right) \tag{B5}$$

By expressing the inner product in summation form and interchanging the order of summation with the expectation, we obtain:

$$= \frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} \sum_{n_1 \in \mathcal{N}_i} \sum_{n_2 \in \mathcal{N}_j} \sum_{k=1}^{d} \mathbb{E}\left(z_{n_1}[k]\, z_{n_2}[k]\right) \tag{B6}$$

If $n_1 \neq n_2$, the expectation is equal to 0 owing to the independence of z_{n_1} and z_{n_2}. Consequently, we can combine the first two summations into a single summation, considering the case where $n_1 = n_2$. Such an occurrence is only possible when summing over the intersection of $\mathcal{N}_i \cap \mathcal{N}_j$. The resulting expression takes the following form:

$$= \frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} \sum_{n \in \mathcal{N}_i \cap \mathcal{N}_j} \overbrace{\sum_{k=1}^{d} \underbrace{\mathbb{E}\left((z_n[k])^2\right)}_{\Sigma[k,k]}}^{\mathrm{Tr}(\Sigma)} \tag{B7}$$

As the quantity $\mathrm{Tr}(\Sigma)$ remains independent of n, it can be extracted and placed outside the summation. Consequently, the sum simplifies to the cardinality of the intersection: $|\mathcal{N}_i \cap \mathcal{N}_j|$. Using all this we can rewrite B7 as follows:

$$= \frac{|\mathcal{N}_i \cap \mathcal{N}_j|}{|\mathcal{N}_i||\mathcal{N}_j|} \mathrm{Tr}(\Sigma) \tag{B8}$$

A special case of B3 emerges when j = i. In this particular scenario, the expectation of the inner product can be written in the following form:

$$\mathbb{E}\left(\epsilon_i^\top \epsilon_i\right) = \frac{\cancel{\overbrace{|\mathcal{N}_i \cap \mathcal{N}_i|}^{|\mathcal{N}_i|}}}{|\mathcal{N}_i|\cancel{|\mathcal{N}_i|}} \mathrm{Tr}(\Sigma) = \frac{1}{|\mathcal{N}_i|} \mathrm{Tr}(\Sigma) \tag{B9}$$

□

B.1.3 Expectation of the outer product of ϵ

$$\mathbb{E}\left(\epsilon_i \epsilon_j^\top\right) = \frac{|\mathcal{N}_i \cap \mathcal{N}_j|}{|\mathcal{N}_i||\mathcal{N}_j|} \Sigma \tag{B10}$$

Proof By writing $\mathbb{E}\left(\epsilon_i \epsilon_j^\top\right)$ in terms of z we get:

$$\mathbb{E}\left(\epsilon_i \epsilon_j^\top\right) = \frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} \sum_{n_1 \in \mathcal{N}_i} \sum_{n_2 \in \mathcal{N}_j} \mathbb{E}\left(z_{n_1} z_{n_2}^\top\right) \tag{B11}$$

If $n_1 \neq n_2$ then $\mathbb{E}\left(z_{n_1} z_{n_2}^\top\right) = \mathbb{E}\left(z_{n_1}\right) \mathbb{E}\left(z_{n_2}^\top\right) = 0$ due to independence. If $n_1 = n_2$ then $\mathbb{E}\left(z_{n_1} z_{n_1}^\top\right) = \Sigma$ by definition. Similarly to the expectation of the inner product in B8, $n_1 = n_2$ is only possible in the intersection, thus the expectation can be equivalently rewritten as:

$$\frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} \sum_{n \in \mathcal{N}_i \cap \mathcal{N}_j} \Sigma = \frac{|\mathcal{N}_i \cap \mathcal{N}_j|}{|\mathcal{N}_i||\mathcal{N}_j|} \Sigma \tag{B12}$$

A specific case of B10 emerges when $j = i$. In this particular scenario, the expectation of the outer product can be written in the following form:

$$\mathbb{E}\left(\epsilon_i \epsilon_i^\top\right) = \frac{\cancel{\overbrace{|\mathcal{N}_i \cap \mathcal{N}_i|}^{|\mathcal{N}_i|}}}{|\mathcal{N}_i|\cancel{|\mathcal{N}_i|}} \Sigma = \frac{1}{|\mathcal{N}_i|} \Sigma \tag{B13}$$

□

B.1.4 Expectation of the cubic form of ϵ

$$\mathbb{E}\left(\epsilon_i \epsilon_i^\top \epsilon_j\right) = 0 \tag{B14}$$

Proof We begin by writing ϵ_i and ϵ_j in terms of z:

$$\mathbb{E}\left(\epsilon_i \epsilon_i^\top \epsilon_j\right) = \frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|} \mathbb{E}\left(\left(\sum_{n_1 \in \mathcal{N}_i} z_{n_1}\right)\left(\sum_{n_2 \in \mathcal{N}_i} z_{n_2}^\top\right)\left(\sum_{n_3 \in \mathcal{N}_j} z_{n_3}\right)\right) \tag{B15}$$

$$= \frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|} \sum_{n_1 \in \mathcal{N}_i} \sum_{n_2 \in \mathcal{N}_i} \sum_{n_3 \in \mathcal{N}_j} \mathbb{E}\left(z_{n_1} z_{n_2}^\top z_{n_3}\right) \tag{B16}$$

In the terms where the condition $n_1 = n_2 = n_3$ does not hold, the expected value becomes zero, a consequence of the assumed independence. The sum can be rewritten by only considering terms where $n_1 = n_2 = n_3$. This circumstance is only possible when all variables are within the intersection of $\mathcal{N}_i$ and $\mathcal{N}_j$. Exploiting this condition, we arrive at the following expression:

$$= \frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|} \sum_{n \in \mathcal{N}_i \cap \mathcal{N}_j} \mathbb{E}\left(z_n z_n^\top z_n\right) \tag{B17}$$

By expressing the dot product in summation form and representing the outcome as a vector, we obtain:

$$= \frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|} \begin{bmatrix} \sum_{k=1}^{d} \mathbb{E}\left(z_n[1]\,(z_n[k])^2\right) \\ \vdots \\ \sum_{k=1}^{d} \mathbb{E}\left(z_n[d]\,(z_n[k])^2\right) \end{bmatrix} \overset{\text{(Isserlis)}}{=} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \tag{B18}$$

Isserlis's theorem states that the expectation of a product involving an odd number of zero-mean Gaussian random variables is always zero. Consequently, this theorem provides a conclusive proof for our statement. □

B.1.5 Expectation of the quartic form of ϵ

$$\mathbb{E}\left(\epsilon_i^\top \epsilon_j \epsilon_j^\top \epsilon_i\right) = \frac{|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} 1^\top \Sigma^2 1 + \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \operatorname{Tr}(\Sigma)^2 \tag{B19}$$

Proof Following a similar approach as before, we start by expressing ϵ_i and ϵ_j in terms of z. Additionally, we interchange the order of summations and expectations, while extracting the constants to the front:

$$\mathbb{E}\left(\epsilon_i^\top \epsilon_j \epsilon_j^\top \epsilon_i\right) = \frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \sum_{i_1 \in \mathcal{N}_i} \sum_{i_2 \in \mathcal{N}_i} \sum_{j_1 \in \mathcal{N}_j} \sum_{j_2 \in \mathcal{N}_j} \underbrace{\mathbb{E}\left(z_{i_1}^\top z_{j_1} z_{j_2}^\top z_{i_2}\right)}_{EV} \tag{B20}$$

If at least one term is independent in the expectation, the expected value becomes zero. Therefore, we will exclusively focus on cases where there are no terms that are independent of the other three terms. This implies that there will be two pairs that correspond to the same nodes—so there is no independent z. We denote these pairs as μ and ρ. Taking this into consideration B20 can be rewritten as:

$$\frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \sum_{\mu \in \mathcal{N}_i} \sum_{\rho \in \mathcal{N}_j} \mathbb{E}\left(z_\mu^\top z_\rho z_\rho^\top z_\mu\right) \tag{B21}$$

$$\mathbb{E}\left(z_\mu^\top z_\rho z_\rho^\top z_\mu\right) = \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \mathbb{E}\left(z_\mu[k_1] z_\rho[k_1] z_\rho[k_2] z_\mu[k_2]\right) \tag{B22}$$

The expectation can be divided into two cases: one where $\mu \neq \rho$ and another where $\mu = \rho$. In the subsequent analysis, we will compute the expectation for each case separately and then reconstruct the overall sum by counting the occurrences of each case.

Case 1: $\mu \neq \rho$ Leveraging the independence of z_μ and z_ρ (given that $\mu \neq \rho$), equation B22 can be expressed as:

$$\mathbb{E}\left(z_\mu^\top z_\rho z_\rho^\top z_\mu\right) = \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \underbrace{\mathbb{E}\left(z_\mu[k_1] z_\mu[k_2]\right)}_{\Sigma[k_1,k_2]} \underbrace{\mathbb{E}\left(z_\rho[k_1] z_\rho[k_2]\right)}_{\Sigma[k_1,k_2]} \tag{B23}$$

$$= \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \Sigma^2[k_1,k_2] = 1^\top \Sigma^2 1 \tag{B24}$$

Case 2: $\mu = \rho$ Using $\mu = \rho$ we can rewrite B22 as follows:

$$\mathbb{E}\left(z_\mu^\top z_\rho z_\rho^\top z_\mu\right) = \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \mathbb{E}\left(z_\mu[k_1] z_\mu[k_1] z_\mu[k_2] z_\mu[k_2]\right) \tag{B25}$$

$$= \sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \underbrace{\mathbb{E}\left(\left(z_\mu[k_1]\right)^2 \left(z_\mu[k_2]\right)^2\right)}_{\substack{\text{(Isserlis)} \\ \Sigma[k_1,k_1]\Sigma[k_2,k_2]+2\Sigma^2[k_1,k_2]}} \tag{B26}$$

$$= \underbrace{\sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \Sigma[k_1,k_1]\,\Sigma[k_2,k_2]}_{\left(\sum_{k_1=1}^{d} \Sigma[k_1,k_1]\right)\left(\sum_{k_2=1}^{d} \Sigma[k_2,k_2]\right)=(\operatorname{Tr}(\Sigma))^2} + 2 \underbrace{\sum_{k_1=1}^{d} \sum_{k_2=1}^{d} \Sigma^2[k_1,k_2]}_{1^\top \Sigma^2 1} \tag{B27}$$

$$= (\operatorname{Tr}(\Sigma))^2 + 2 \cdot 1^\top \Sigma^2 1 \tag{B28}$$

Now that we have the expectation for $\mu \neq \rho$ and $\mu = \rho$, we just need to count how many times each appears in the sum in B20.

- Case 1: there are $|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_i \cap \mathcal{N}_j|^2 - 3|\mathcal{N}_i \cap \mathcal{N}_j|$ terms at total such that $\mu \neq \rho$, composed of the following two disjoint components:
  - There are $|\mathcal{N}_i||\mathcal{N}_j| - |\mathcal{N}_i \cap \mathcal{N}_j|^2$ possible combinations for μ and ρ such that $\mu \notin \mathcal{N}_i \cap \mathcal{N}_j$ and $\rho \notin \mathcal{N}_i \cap \mathcal{N}_j$. These combinations can be constructed from i_1, i_2, j_1, j_2 only if $i_1, i_2 \in \mathcal{N}_i \backslash \mathcal{N}_i \cap \mathcal{N}_j$ and $j_1, j_2 \in \mathcal{N}_j \backslash \mathcal{N}_i \cap \mathcal{N}_j$ and this further implies that $i_1 = i_2$ and $j_1 = j_2$, resulting in 1 term per μ, ρ pair.
  - When $\mu, \rho \in \mathcal{N}_i \cap \mathcal{N}_j$, there are exactly $|\mathcal{N}_i \cap \mathcal{N}_j|^2 - |\mathcal{N}_i \cap \mathcal{N}_j|$ terms when $\mu \neq \rho$. However this time there are 3 distinct ways that construct μ, ρ from i_1, i_2, j_1, j_2:
    - $\ast$ $\mu = i_1 = i_2$ and $\rho = j_1 = j_2$
    - $\ast$ $\mu = i_1 = j_2$ and $\rho = j_1 = i_2$
    - $\ast$ $\mu = j_1 = j_2$ and $\rho = i_1 = i_2$

    resulting in $3\left(|\mathcal{N}_i \cap \mathcal{N}_j|^2 - |\mathcal{N}_i \cap \mathcal{N}_j|\right)$ such terms.
- Case 2: if $\mu = \rho$, it follows that both μ and ρ must belong to the intersection of the neighborhoods $\mathcal{N}_i \cap \mathcal{N}_j$, implying the existence of a total of $|\mathcal{N}_i \cap \mathcal{N}_j|$ such terms. Since $\mu = \rho$ can only be true if $i_1 = i_2 = j_1 = j_2$, the total number of the terms corresponding to Case 2 is $|\mathcal{N}_i \cap \mathcal{N}_j|$.

Putting the results altogether we get

$$\begin{aligned}\mathbb{E}\left(\epsilon_i^\top \epsilon_j \epsilon_j^\top \epsilon_i\right) &= \frac{\overbrace{\left(|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - 3|\mathcal{N}_{i,j}|\right) 1^\top \Sigma^2 1}^{\text{(Case 1)}} + \overbrace{|\mathcal{N}_{i,j}|\left(\mathrm{Tr}\left(\Sigma\right)^2 + 2\cdot 1^\top \Sigma^2 1\right)}^{\text{(Case 2)}}}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \\ &= \frac{|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} 1^\top \Sigma^2 1 + \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2}\mathrm{Tr}\left(\Sigma\right)^2 \end{aligned} \tag{B29}$$

and thus we arrived to the desired form of the quartic expectation.

□

B.2 Expectation of the lookup table embedding similarity ($s_{\tilde{f}}$)

If $i \neq j$ then

$$\mathbb{E}\left(s_{\tilde{f}}(i,j)\right) = s_f(i,j) \tag{B30}$$

Proof

$$\begin{aligned}\mathbb{E}\left(s_{\tilde{f}}(i,j)\right) &= \mathbb{E}\left(\tilde{f}(i)^\top \tilde{f}(j)\right) \\ &= \mathbb{E}\left((\theta_i + z_i)^\top\left(\theta_j + z_j\right)\right) \qquad \text{(B31)} \\ &= \mathbb{E}\left(\theta_i^\top \theta_j + \theta_i^\top z_j + \theta_j^\top z_i + z_i^\top z_j\right) \\ &= \theta_i^\top \theta_j + \theta_i^\top \underbrace{\mathbb{E}\left(z_j\right)}_{0} + \theta_j^\top \underbrace{\mathbb{E}\left(z_i\right)}_{0} + \mathbb{E}\left(z_i^\top z_j\right) \qquad \text{(B32)}\end{aligned}$$

Using $i \neq j$ we get

$$\begin{aligned}&= \theta_i^\top \theta_j + \underbrace{\mathbb{E}\left(z_i^\top\right)}_{0}\underbrace{\mathbb{E}\left(z_j\right)}_{0} \\ &= \theta_i^\top \theta_j = f(i)^\top f(j) = s_f(i,j)\end{aligned} \tag{B33}$$

□

B.3 Variance of the lookup table embedding similarity ($s_{\tilde{f}}$)

If $i \neq j$ then

$$\mathrm{Var}\left(s_{\tilde{f}}(i,j)\right) = f(i)^\top \Sigma f(i) + f(j)^\top \Sigma f(j) + 1^\top \Sigma^2 1 \tag{B34}$$

Proof

$$\mathrm{Var}\left(s_{\tilde{f}}(i,j)\right) = \mathbb{E}\left(\left(\mathbb{E}\left(s_{\tilde{f}}(i,j)\right) - s_{\tilde{f}}(i,j)\right)^2\right) \tag{B35}$$

$$\begin{aligned}&= \mathbb{E}\left(\left(s_f(i,j) - s_{\tilde{f}}(i,j)\right)^2\right) \\ &= \mathbb{E}\left(\left(\cancel{\theta_i^\top \theta_j} - \cancel{\theta_i^\top \theta_j} - \theta_i^\top z_j - \theta_j^\top z_i - z_i^\top z_j\right)^2\right)\end{aligned} \tag{B36}$$

$$\begin{aligned}&= \underbrace{\mathbb{E}\left(\theta_i^\top z_j \theta_i^\top z_j\right)}_{\theta_i^\top \Sigma \theta_i} + \underbrace{\mathbb{E}\left(\theta_i^\top z_j \theta_j^\top z_i\right)}_{0} \\ &\quad + \underbrace{\mathbb{E}\left(\theta_i^\top z_j z_i^\top z_j\right)}_{0} \quad (i \neq j) \qquad \text{(B37)} \\ &\quad + \underbrace{\mathbb{E}\left(\theta_j^\top z_i \theta_i^\top z_j\right)}_{0} + \underbrace{\mathbb{E}\left(\theta_j^\top z_i \theta_j^\top z_i\right)}_{\theta_j^\top \Sigma \theta_j} \\ &\quad + \underbrace{\mathbb{E}\left(\theta_j^\top z_i z_i^\top z_j\right)}_{0} \quad (i \neq j) \qquad \text{(B38)} \\ &\quad + \underbrace{\mathbb{E}\left(z_i^\top z_j \theta_i^\top z_j\right)}_{0} + \underbrace{\mathbb{E}\left(z_i^\top z_j \theta_j^\top z_i\right)}_{0} \\ &\quad + \underbrace{\mathbb{E}\left(z_i^\top z_j z_i^\top z_j\right)}_{1^\top \Sigma^2 1} \quad (i \neq j) \qquad \text{(B39)}\end{aligned}$$

$$\begin{aligned}&= \theta_i^\top \Sigma \theta_i + \theta_j^\top \Sigma \theta_j + 1^\top \Sigma^2 1 \\ &= f(i)^\top \Sigma f(i) + f(j)^\top \Sigma f(j) + 1^\top \Sigma^2 1\end{aligned} \tag{B40}$$

□

B.4 Expectation of the ISNE embedding similarity ($s_{\tilde{h}}$)

$$\mathbb{E}\left(s_{\tilde{h}}(i,j)\right) = s_h(i,j) + \underbrace{\frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i||\mathcal{N}_j|}}_{Q}\mathrm{Tr}\left(\Sigma\right) \tag{B41}$$

Proof

$$s_{\tilde{h}}(i,j) = \tilde{h}(i)^\top \tilde{h}(j) = \left(h(i)+\epsilon_i\right)^\top \left(h(j)+\epsilon_j\right) \tag{B42}$$

$$= h(i)^\top h(j) + h(i)^\top \epsilon_j + h(j)^\top \epsilon_i + \epsilon_i^\top \epsilon_j \tag{B43}$$

$$\mathbb{E}\left(s_{\tilde{h}}(i,j)\right) = \mathbb{E}\left(h(i)^\top h(j) + h(i)^\top \epsilon_j + h(j)^\top \epsilon_i + \epsilon_i^\top \epsilon_j\right) \tag{B44}$$

$$= \mathbb{E}\left(h(i)^\top h(j)\right) + h(i)^\top \underbrace{\mathbb{E}\left(\epsilon_j\right)}_{0} + h(j)^\top \underbrace{\mathbb{E}\left(\epsilon_i\right)}_{0} + \underbrace{\mathbb{E}\left(\epsilon_i^\top \epsilon_j\right)}_{Q\mathrm{Tr}(\Sigma)} \quad (B1,\ B3) \tag{B45}$$

$$= \underbrace{h(i)^\top h(j)}_{s_h(i,j)} + \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i||\mathcal{N}_j|}\mathrm{Tr}\left(\Sigma\right) \tag{B46}$$

□

B.5 Variance of the ISNE embedding similarity ($s_{\tilde{h}}$)

$$\mathrm{Var}\left(s_{\tilde{h}}(i,j)\right) = \frac{1}{|\mathcal{N}_j|}h(i)^\top \Sigma h(i) + 2Qh(i)^\top \Sigma h(j) + \frac{1}{|\mathcal{N}_i|}h(j)^\top \Sigma h(j) \tag{B47}$$

$$+ \frac{1}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2}\left[\left(|\mathcal{N}_{i,j}| - |\mathcal{N}_{i,j}|^2\right)\mathrm{Tr}\left(\Sigma\right)^2 + \left(|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|\right)1^\top \Sigma 1\right] \tag{B48}$$

where $Q = \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i||\mathcal{N}_j|}$.

Proof

$$\mathrm{Var}\left(s_{\tilde{h}}(i,j)\right) = \mathbb{E}\left(\left(\mathbb{E}\left(s_{\tilde{h}}(i,j)\right) - s_{\tilde{h}}(i,j)\right)^2\right) \tag{B49}$$

$$= \mathbb{E}\left(\left(Q\mathrm{Tr}\left(\Sigma\right) - h(i)^\top \epsilon_j - h(j)^\top \epsilon_i - \epsilon_i^\top \epsilon_j\right)^2\right) \tag{B50}$$

$$= Q\mathrm{Tr}\left(\Sigma\right)\left(\cancel{Q\mathrm{Tr}\left(\Sigma\right)} - h(i)^\top \underbrace{\mathbb{E}\left(\epsilon_j\right)}_{0} - h(j)^\top \underbrace{\mathbb{E}\left(\epsilon_i\right)}_{0} - \underbrace{\cancel{\mathbb{E}\left(\epsilon_i^\top \epsilon_j\right)}}_{Q\mathrm{Tr}(\Sigma)}\right) \tag{B51}$$

$$(B1,\ B3) \tag{B52}$$

$$+ h(i)^\top \underbrace{\mathbb{E}\left(\epsilon_j \epsilon_j^\top\right)}_{\frac{1}{|\mathcal{N}_j|}\Sigma} h(i) + h(i)^\top \underbrace{\mathbb{E}\left(\epsilon_j \epsilon_i^\top\right)}_{Q\Sigma} h(j) + h(i)^\top \underbrace{\mathbb{E}\left(\epsilon_j \epsilon_i^\top \epsilon_j\right)}_{0} \tag{B53}$$

$$(B1,\ B13,\ B10,\ B14) \tag{B54}$$

$$+ h(j)^\top \underbrace{\mathbb{E}\left(\epsilon_i \epsilon_j^\top\right)}_{Q\Sigma} h(i) + h(j)^\top \underbrace{\mathbb{E}\left(\epsilon_i \epsilon_i^\top\right)}_{\frac{1}{|\mathcal{N}_i|}\Sigma} h(j) + h(j)^\top \underbrace{\mathbb{E}\left(\epsilon_i \epsilon_i^\top \epsilon_j\right)}_{0} \tag{B55}$$

$$(B1,\ B13,\ B10,\ B14) \tag{B56}$$

$$- \underbrace{\mathbb{E}\left(\epsilon_i^\top \epsilon_j\right)}_{Q\mathrm{Tr}(\Sigma)} Q\mathrm{Tr}\left(\Sigma\right) + \underbrace{\mathbb{E}\left(\epsilon_i^\top \epsilon_j \epsilon_j^\top\right)}_{0} h(i) + \underbrace{\mathbb{E}\left(\epsilon_i^\top \epsilon_j \epsilon_i^\top\right)}_{0} h(j) + \underbrace{\mathbb{E}\left(\epsilon_i^\top \epsilon_j \epsilon_i^\top \epsilon_j\right)}_{*} \tag{B57}$$

$$(B3,\ B14,\ B19) \tag{B58}$$

$$= \frac{1}{|\mathcal{N}_j|}h(i)^\top \Sigma h(i) + \underbrace{Qh(i)^\top \Sigma h(j) + Qh(j)^\top \Sigma h(i)}_{2Qh(i)^\top \Sigma h(j)} + \frac{1}{|\mathcal{N}_i|}h(j)^\top \Sigma h(j) \tag{B59}$$

$$- Q^2\mathrm{Tr}\left(\Sigma\right)^2 + \underbrace{\frac{|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2}1^\top \Sigma^2 1 + \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2}\mathrm{Tr}\left(\Sigma\right)^2}_{*} \tag{B60}$$

$$\begin{aligned}
&= \frac{1}{|\mathcal{N}_j|} h(i)^\top \Sigma h(i) + 2Q h(i)^\top \Sigma h(j) \\
&\quad + \frac{1}{|\mathcal{N}_i|} h(j)^\top \Sigma h(j)
\end{aligned} \tag{B61}$$

$$\begin{aligned}
&\quad + \frac{|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} 1^\top \Sigma^2 1 \\
&\quad + \frac{|\mathcal{N}_{i,j}| - |\mathcal{N}_{i,j}|^2}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \mathrm{Tr}\,(\Sigma)^2
\end{aligned} \tag{B62}$$

□

B.6 Variance bound

$$\mathrm{Var}\left(s_{\tilde{h}}(i,j)\right) \le \frac{3}{K} \mathrm{Var}\left(s_{\tilde{f}}(i,j)\right) \tag{B63}$$

Proof

$$\begin{aligned}
&K = \min\{|\mathcal{N}_i|, |\mathcal{N}_j|\}; \quad \frac{1}{|\mathcal{N}_i|} \le \frac{1}{K}; \frac{1}{|\mathcal{N}_i|} \le \frac{1}{K}; \\
&\frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} \le \frac{1}{K}
\end{aligned} \tag{B64}$$

$$\begin{aligned}
&|\mathcal{N}_{i,j}| \le K; \quad Q = \frac{|\mathcal{N}_{i,j}|}{|\mathcal{N}_i||\mathcal{N}_j|} \le \frac{K}{|\mathcal{N}_i||\mathcal{N}_j|} \\
&\le \frac{K}{K^2} \le \frac{1}{K}
\end{aligned} \tag{B65}$$

$$\begin{aligned}
T_1 &= \frac{1}{|\mathcal{N}_j|} h(i)^\top \Sigma h(i) + 2Q h(i)^\top \Sigma h(j) \\
&\quad + \frac{1}{|\mathcal{N}_i|} h(j)^\top \Sigma h(j)
\end{aligned} \tag{B66}$$

$$T_1 \le \frac{1}{K} h(i)^\top \Sigma h(i) + \frac{2}{K} h(i)^\top \Sigma h(j) + \frac{1}{K} h(j)^\top \Sigma h(j) \tag{B67}$$

The $h(i)^\top \Sigma h(j)$ term can always be upper-bounded by $\max\{h(i)^\top \Sigma h(i), h(j)^\top \Sigma h(j)\}$. We will assume, without loss of generality, that $h(i)^\top \Sigma h(i) \ge h(j)^\top \Sigma h(j)$. The derivation proceeds by bounding $h(i)^\top \Sigma h(j)$ with $h(i)^\top \Sigma h(i)$. However, the proof follows an analogous structure in the alternative case where $h(j)^\top \Sigma h(j)$ is larger.

$$B67 \le \frac{3}{K} h(i)^\top \Sigma h(i) + \frac{1}{K} h(j)^\top \Sigma h(j) \tag{B68}$$

$$\le \frac{3}{K} \left(h(i)^\top \Sigma h(i) + h(j)^\top \Sigma h(j)\right) = \hat{T}_1 \tag{B69}$$

$$\begin{aligned}
T_2 &= \frac{|\mathcal{N}_i||\mathcal{N}_j| + 2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} 1^\top \Sigma^2 1 \\
&\quad + \frac{|\mathcal{N}_{i,j}| - |\mathcal{N}_{i,j}|^2}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \mathrm{Tr}\,(\Sigma)^2
\end{aligned} \tag{B70}$$

$$\begin{aligned}
&= \left(\frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} + \frac{2|\mathcal{N}_{i,j}|^2 - |\mathcal{N}_{i,j}|}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2}\right) 1^\top \Sigma^2 1 \\
&\quad + \frac{|\mathcal{N}_{i,j}| - |\mathcal{N}_{i,j}|^2}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2} \mathrm{Tr}\,(\Sigma)^2
\end{aligned} \tag{B71}$$

$$\begin{aligned}
&\le \left(\frac{1}{|\mathcal{N}_i||\mathcal{N}_j|} + \frac{2|\mathcal{N}_{i,j}|^2}{|\mathcal{N}_i|^2|\mathcal{N}_j|^2}\right) 1^\top \Sigma^2 1 \\
&(\text{Using } |\mathcal{N}_{i,j}| \ge 0)
\end{aligned} \tag{B72}$$

$$\le \left(\frac{1}{K} + \frac{2}{K}\right) 1^\top \Sigma^2 1 = \frac{3}{K} 1^\top \Sigma^2 1 = \hat{T}_2 \tag{B73}$$

If we posit that both the ISNE and lookup table models are capable of achieving the same optimal solution (w.r.t. the Node2Vec loss function), then we can deduce that the (not noise augmented) embedding vector associated with any node v is identical in both models, denoted by $f(v) = h(v)$ (and is equal the optimal embedding). Using this we can get to the desired upper bound:

$$\mathrm{Var}\left(s_{\tilde{h}}(i,j)\right) = T_1 + T_2 \le \hat{T}_1 + \hat{T}_2 \tag{B74}$$

$$= \frac{3}{K}\left(h(i)^\top \Sigma h(i) + h(j)^\top \Sigma h(j) + 1^\top \Sigma^2 1\right) \tag{B75}$$

$$\begin{aligned}
&= \frac{3}{K}\left(f(i)^\top \Sigma f(i) + f(j)^\top \Sigma f(j) + 1^\top \Sigma^2 1\right) \\
&= \frac{3}{K} \mathrm{Var}\left(s_{\tilde{f}}(i,j)\right)
\end{aligned} \tag{B76}$$

□

Author Contributions All authors contributed equally to this work.

Funding Open access funding provided by Budapest University of Technology and Economics. The authors declare that no funds, grants, or other support were received during the preparation of this article.

Data Availibility All data generated or analyzed during this study are included in this article.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Hamilton WL, Ying R, Leskovec J (2017) Representation learning on graphs: methods and applications. arXiv preprint arXiv:1709.05584
2. Yi H-C, You Z-H, Huang D-S, Kwoh CK (2022) Graph representation learning in bioinformatics: trends, methods and applications. Briefings Bioinform 23(1):340
3. Kim M, Baek SH, Song M (2018) Relation extraction for biological pathway construction using node2vec. BMC Bioinform 19:75–84
4. Thafar MA, Olayan RS, Albaradei S, Bajic VB, Gojobori T, Essack M, Gao X (2021) Dti2vec: drug-target interaction prediction using network embedding and ensemble learning. J Cheminform 13(1):1–18
5. Wang Y, Li Z, Farimani AB (2023) In: Qu, C., Liu, H. (eds.) Graph neural networks for molecules, pp. 21–66. Springer, Cham
6. Wang M, Lin Y, Lin G, Yang K, Wu X-m (2020) M2grl: A multi-task multi-view graph representation learning framework for web-scale recommender systems. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2349–2358
7. Ge S, Wu C, Wu F, Qi T, Huang Y (2020) Graph enhanced representation learning for news recommendation. In: Proceedings of The Web Conference 2020, pp. 2863–2869
8. Liu Y, Tian Z, Sun J, Jiang Y, Zhang X (2020) Distributed representation learning via node2vec for implicit feedback recommendation. Neural Comput Appl 32:4335–4345
9. Tan Q, Liu N, Hu X (2019) Deep representation learning for social network analysis. Front Big Data 2:2
10. Li B, Pi D (2020) Network representation learning: a systematic literature review. Neural Comput Appl 32(21):16647–16679
11. 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
12. Ljubičić K, Merćep A, Kostanjčar Z (2023) Churn prediction methods based on mutual customer interdependence. J Comput Sci 67:101940
13. Thang DC, Dat HT, Tam NT, Jo J, Hung NQV, Aberer K (2022) Nature vs. nurture: feature vs. structure for graph neural networks. Pattern Recogn Lett 159:46–53
14. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. Adv Neural Inform Process Syst 30
15. 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
16. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, pp. 1067–1077
17. Qiu J, Dong Y, Ma H, Li J, Wang K, Tang J (2018) Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 459–467
18. Cao S, Lu W, Xu Q (2015) Grarep: learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pp. 891–900
19. Tang J, Qu M, Mei Q (2015) Pte: Predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1165–1174
20. Guo L, Cai X, Qin H, Hao F, Guo S (2022) A content-sensitive citation representation approach for citation recommendation. J Ambient Intell Hum Comput:1–12
21. Zhou H, Sun G, Fu S, Wang L, Hu J, Gao Y (2021) Internet financial fraud detection based on a distributed big data approach with node2vec. IEEE Access 9:43378–43386
22. Ha J, Park S (2022) Ncmd: Node2vec-based neural collaborative filtering for predicting mirna-disease association. IEEE/ACM Trans Comput Biol Bioinform 20(2):1257–1268
23. Ji B-Y, You Z-H, Cheng L, Zhou J-R, Alghazzawi D, Li L-P (2020) Predicting mirna-disease association from heterogeneous information network with grarep embedding model. Sci Rep 10(1):6658
24. Liang X, Si G, Li J, Tian P, An Z, Zhou F (2024) A survey of inductive knowledge graph completion. Neural Comput Appl 36(8):3837–3858
25. Tran DH, Sheng QZ, Zhang WE, Aljubairy A, Zaib M, Hamad SA, Tran NH, Khoa NLD (2021) Hetegraph: graph learning in recommender systems via graph convolutional networks. Neural Comput Appl:1–17
26. Lo WW, Layeghy S, Sarhan M, Gallagher M, Portmann M (2022) E-graphsage: a graph neural network based intrusion detection system for iot. In: NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9. IEEE
27. Liu J, Lei X, Zhang Y, Pan Y (2023) The prediction of molecular toxicity based on bigru and graphsage. Comput Biol Med 153:106524
28. Sun Q, Wei X, Yang X (2024) Graphsage with deep reinforcement learning for financial portfolio optimization. Expert Syst Appl 238:122027
29. Liu J, Ong GP, Chen X (2020) Graphsage-based traffic speed forecasting for segment network with sparse data. IEEE Trans Intell Transp Syst 23(3):1755–1766
30. Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J (2020) Open graph benchmark: datasets for machine learning on graphs. Adv Neural Inform Process Syst 33:22118–22133
31. Bojchevski A, Günnemann S (2017) Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815
32. Tang L, Liu H (2009) Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 817–826
33. Mernyei P, Cangea C (2020) Wiki-cs: A wikipedia-based benchmark for graph neural networks. arXiv preprint arXiv:2007.02901
34. Jeong H, Néda Z, Barabási A-L (2003) Measuring preferential attachment in evolving networks. Europhys Lett 61(4):567
35. Bukumira M, Antonijevic M, Jovanovic D, Zivkovic M, Mladenovic D, Kunjadic G (2022) Carrot grading system using computer vision feature parameters and a cascaded graph convolutional neural network. J Electron Imaging 31(6):061815–061815
36. Schuetz MJ, Brubaker JK, Katzgraber HG (2022) Combinatorial optimization with physics-inspired graph neural networks. Nat Mach Intell 4(4):367–377
37. Tolstaya E, Gama F, Paulos J, Pappas G, Kumar V, Ribeiro A (2020) Learning decentralized controllers for robot swarms with graph neural networks. In: Conference on Robot Learning, pp. 671–682. PMLR

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.