How randomness affects the stability of node embeddings across different embedding dimensions?

Haotian Shen[*]
University of Cambridge
hs831@cam.ac.uk

Weiling Du[*]
University of Cambridge
wd301@cam.ac.uk

Abstract

We assess the impact of randomness on two node embedding algorithms, Node2Vec and GraphSAGE, across varying embedding dimensions. The effects are quantified by evaluating the geometric stability of different embeddings and their performance in downstream link prediction tasks. Two citation datasets of disparate sizes are employed to generate the embeddings. Our observations reveal that Node2Vec and GraphSAGE demonstrate distinct behaviors when confronted with randomness across diverse embedding dimensions, suggesting that the choice of algorithm should be taken into account when considering the influence of randomness. Our work offers insights into selecting an embedding dimension that exhibits greater stability towards randomness for a given embedding algorithm.

Statement of contribution

We, Haotian Shen and Weiling Du of the University of Cambridge, jointly declare that our work towards this project has been executed as follows:

- **Haotian Shen** contributed by designing and implementing a majority of the geometric and downstream stability metrics. He designed and executed all the experiments related to the Node2Vec embedding algorithm, and conducting corresponding analysis. Additionally, he authored half of the paper.
- **Weiling Du** contributed by generating the test set of negative edges for Cora and Citation2, computing the second-order cosine similarity, building and implementing all experiments related to the GraphSAGE algorithm, designing functions to save metrics and visualizing them, and writing the half of the paper.

We both independently submit identical copies of this paper, certifying this statement to be correct.

GitHub repository with commit log

The companion source code for our project may be found at: https://github.com/ClouddShen/L45.

1 Introduction

Many state-of-the-art node embedding algorithms inherently encode randomness in various aspects, including parameter initialization and optimization. This randomness can impact the generated embeddings, as an algorithm may produce different embeddings with identical parameter settings and input graphs when using different random seeds. Prior research [1] has explored the effects of randomness on the stability of node embeddings. However, this study fixes the node embedding dimension to 128, neglecting the possibility that the influence of randomness on node embeddings

[*]Equal contribution.

Haotian Shen & Weiling Du, How randomness affects the stability of node embeddings across different embedding dimensions?. *Representation Learning on Graphs and Networks (L45), University of Cambridge*.

may vary across different embedding dimensions. To address this research gap, we conduct a study evaluating the effect of randomness on node embeddings for various embedding dimensions.

Research Objective. Our objective is to analyze the sensitivity of node embeddings with different embedding dimensions to randomness. Specifically, we assess the geometric stability of node embeddings and downstream stability, with the former measuring changes in the embedding space and the latter capturing differences in link prediction results. By identifying the relationship between embedding dimensions and the sensitivity of randomness encoded in the aforementioned stability metrics, we aim to gain a deeper understanding of the impact of randomness on node embeddings.

Approach. We select Node2Vec and GraphSAGE as the target node embedding generation algorithms and choose a range of embedding dimensions. For each dimension, both algorithms produce multiple node embeddings using identical parameter settings on the same dataset while utilizing different random seeds. The datasets employed are two citation datasets of varying sizes. Initially, we investigate the impact of randomness on different embedding dimensions by directly evaluating the node embeddings. Subsequently, we analyze the effect on downstream link prediction task stability to further understand the influence of randomness when dimensions vary.

Result. We observe that Node2Vec and GraphSAGE exhibit distinct behaviors when facing randomness across different embedding dimensions, with more details available in Sections 4 and 5. Additionally, this study yields a by-product, revealing a positive relationship between the geometric stability of node embeddings and downstream task performance for a specific embedding dimension.

2 Stability Measures

2.1 Basic Definitions

A graph $G = (V, E)$ consists of a collection of vertices $V = \{v_1, \ldots, v_N\}$ and a set of edges $E = \{e_1, \ldots, e_M\}$. We define a node embedding as a function $\varphi : V \to \mathbb{R}^{N \times d}$, mapping nodes to embedding vectors of length d. We compare the similarity between the different node embeddings with the same length d. Therefore, we use upper indices $\varphi^{(g,d)}(v_i) = z_i^{(g,d)}, \varphi^{(m,d)}(v_i) = z_i^{(m,d)}$ to denote different node embeddings with the same length d. We also denote a node embedding $\varphi^{(g,d)}(V)$ by its embedding matrix $Z^{(g,d)} = \left[z_1^{(g,d)}, \ldots, z_N^{(g,d)}\right]^T \in \mathbb{R}^{N \times d}$ in which the i-th row corresponds to the embedding $z_i^{(g,d)}$ of node v_i under $\varphi^{(g,d)}$. When we use node embeddings matrix $Z^{(g,d)}$ to do downstream tasks, the downstream model achieves link predictions, $L^{(g,d)} = \left[l_1^{(g,d)}, \ldots, l_{M'}^{(g,d)}\right]^T \in \mathbb{R}^{M_1+M_2}$, where M_1 is the number of positive edges in the test set, the M_2 is the number of negative edges in the test set, and the i-th element corresponds to the link prediction of the i-th edge. Among link predictions, $TP^{(g,d)}$ refers to the edges that are accurately identified as positive in $L^{(g,d)}$. $FP^{(g,d)}$ refers to the edges that are inaccurately identified as positive in $L^{(g,d)}$. $TN^{(g,d)}$ refers to the edges that are accurately identified as negative in $L^{(g,d)}$. $FN^{(g,d)}$ refers to edges that are inaccurately identified as negative in $L^{(g,d)}$.

2.2 Geometric Stability

We calculate the similarity value between two node embeddings of the same dimension, generated by a node embedding algorithm using two different random seeds. We then compute the average similarity value and the standard deviation across all random seed pairs, which serves as the geometric stability of the node embedding algorithm for that specific dimension. To measure the similarity value between two node embeddings, we utilize three metrics: Procrustes Similarity, $\boldsymbol{k}$-NN Jaccard Similarity, and $\boldsymbol{k}$-NN second-order cosine similarity.

Procruste Similarity [2], [3]. This measure calculates the global similarity between two node embeddings after addressing the Procrustes problem. We initially normalize both node embeddings using Procrustes normalization, ensuring that $tr(Z^{(g,d)} Z^{(g,d)^T}) = 1$, $tr(Z^{(m,d)} Z^{(m,d)^T}) = 1$, and that $Z^{(g,d)}$ and $Z^{(m,d)}$ are centered around the origin. Subsequently, we compute the optimal rotation matrix $Q^{(g,m,d)} \in \mathbb{R}^{d \times d}$, which most closely maps $Z^{(g,d)}$ to $Z^{(m,d)}$, in order to solve the orthogonal

Procrustes problem. Using $Q^{(g,m,d)}$, we define the Procruste disparity between two embeddings $Z^{(g,d)}$ and $Z^{(m,d)}$ as the sum of the squares of the node-wise differences between these two aligned embeddings. We then define the Procruste similarity as:

$$s^{\text{ps}}\left(Z^{(g,d)}, Z^{(m,d)}\right) := 1 - \text{Procruste-sim}\left(Z^{(g,d)}Q^{(g,m,d)}, Z^{(m,d)}\right).$$

***k*-NN Jaccard Similarity [4] .** This measure is based on the local neighbourhood information. For a given node i, we first compute the k nearest neighbours with respect to cosine similarity in both embedding vectors of the same dimension d. Following this, we compute the Jaccard similarity between the two sets of k nearest neighbours for node i. Formally, let $\mathcal{N}_k^{(g,d)}(v_i)$ denote the generated set of k nearest neighbors $\{v_{n_1}, \ldots, v_{n_k}\}$ of node v_i. The ***k*-NN Jaccard similarity between two embeddings $z_i^{(g,d)}$ and $z_i^{(m,d)}$ of a node v_i can be computed as:

$$s_k^{\text{NN}}\left(z_i^{(g,d)}, z_i^{(m,d)}\right) := \frac{\left|\mathcal{N}_k^{(g,d)}(v_i) \cap \mathcal{N}_k^{(m,d)}(v_i)\right|}{\left|\mathcal{N}_k^{(g,d)}(v_i) \cup \mathcal{N}_k^{(m,d)}(v_i)\right|}.$$

***k*-NN Second-Order Cosine Similarity .** The second-order cosine similarity [5] also focuses on the local neighbourhood but in a more fine-grained way. We first generate the ordered set $\{u_1, \ldots, u_K\} := \mathcal{N}_k^{(g,d)}(v_i) \cup \mathcal{N}_k^{(m,d)}(v_i)$, which contains the k nearest neighbours of v_i in each of the two embeddings under the current node embedding dimension d. For each u_j, we calculate the cosine similarity of its embedding and the embedding of v_i. Subsequently, for all u_j, we store the resulting cosine similarities in vectors $s^{(g,d)}(v_i), s^{(m,d)}(v_i)$, representing the k-NN first-order cosine similarities of node v_i under two node embedding algorithms and the node embedding dimension d, where

$$\begin{aligned} s_j^{(g,d)}(v_i) &:= \text{cos-sim}\left(z_i^{(g,d)}, \varphi^{(g,d)}(u_j)\right) \quad , \\ s_j^{(m,d)}(v_i) &:= \text{cos-sim}\left(z_i^{(m,d)}, \varphi^{(m,d)}(u_j)\right) \quad , \\ &\forall u_j \in \mathcal{N}_k^{(g,d)}(v_i) \cup \mathcal{N}_k^{(m,d)}(v_i). \end{aligned}$$

Therefore, the k-NN second-order cosine similarity between the embeddings $z_i^{(g,d)}, z_i^{(m,d)}$ of v_i is defined as

$$s_k^{\cos}\left(z_i^{(g,d)}, z_i^{(m,d)}\right) := \text{cos-sim}\left(s^{(g,d)}(v_i), s^{(m,d)}(v_i)\right).$$

2.3 Downstream Stability

2.3.1 Accuracy Score

We first employ the accuracy score to evaluate the performance of the downstream task. We use the node embedding matrix as input features and test the model to achieve $L^{(g,d)}$. Based on the link predictions, we calculate three accuracy scores $Acc_{pos}^{(g,d)}$, $Acc_{neg}^{(g,d)}$, and $Acc_{all}^{(g,d)}$. We compute the standard deviation of the resulting accuracy scores across random seeds as the downstream stability measure for this node embedding algorithm of the node embedding dimension d. $Acc_{pos}^{(g,d)}$, $Acc_{neg}^{(g,d)}$, and $Acc_{all}^{(g,d)}$ are computed as:

$$Acc_{pos}^{(g,d)} := \frac{\left|TP^{(g,d)}\right|}{M_1}$$

$$Acc_{neg}^{(g,d)} := \frac{\left|TN^{(g,d)}\right|}{M_2}$$

$$Acc_{all}^{(g,d)} := \frac{\left|TP^{(g,d)}\right| + \left|TN^{(g,d)}\right|}{M_1 + M_2}$$

2.3.2 Link Prediction Similarity

To explore the downstream stability in a more fine-grained way, we further investigate the stability of link prediction outcomes. Specifically, we use two different node embeddings $Z^{(g,d)}$ and $Z^{(m,d)}$to train the downstream model separately, resulting in link predictions outcomes $L^{(g,d)}$ and $L^{(m,d)}$. Subsequently, we calculate their link-wise similarity of $L^{(g,d)}$ and $L^{(m,d)}$ for each pair of node embeddings. In terms of the link-wise similarity, we propose three metrics, the link prediction overlap rate metric 1, the link prediction overlap rate metric 2, and the link prediction overlap rate metric 3.

Link Prediction Overlap Rate Metric 1. The first metric is that we calculate the ratio of overlapping predictions (i.e., when both models predict an edge exists or does not exist between two nodes). We define $metric1_{pos}^{(g,d)}$, $metric1_{neg}^{(g,d)}$ and $metric1_{all}^{(g,d)}$ as follows:

$$metric1_{pos}^{(g,d)} := \frac{\left|(TP^{(g,d)} \cup FN^{(g,d)}) \cap (TP^{(m,d)} \cup FN^{(m,d)})\right|}{M_1},$$

$$metric1_{neg}^{(g,d)} := \frac{\left|(TN^{(g,d)} \cup FP^{(g,d)}) \cap (TN^{(m,d)} \cup FP^{(m,d)})\right|}{M_2},$$

$$metric1_{all}^{(g,d)} := \frac{\left|L^{(g,d)} \cap L^{(m,d)}\right|}{M_1 + M_2}.$$

Link Prediction Overlap Rate Metric 2. One drawback of the previous metric is that overlapping cases also include instances where both predictors produce the same, but incorrect result. Since the focus is more on the correctly classified cases, we propose the correct prediction link-wise stability metric. The overlapping predictions now only cover cases where both predictors make the same prediction, and this prediction aligns with the ground truth. We compute $metric2_{pos}^{(g,d)}$, $metric2_{neg}^{(g,d)}$ and $metric2_{all}^{(g,d)}$ as follows:

$$metric2_{pos}^{(g,d)} := \frac{\left|TP^{(g,d)} \cap TP^{(m,d)}\right|}{M_1}$$

$$metric2_{neg}^{(g,d)} := \frac{\left|TN^{(g,d)} \cap TN^{(m,d)}\right|}{M_2}$$

$$metric2_{all}^{(g,d)} := \frac{\left|(TP^{(g,d)} \cup TN^{(g,d)}) \cap (TP^{(m,d)} \cup TN^{(m,d)})\right|}{M_1 + M_2}$$

Link Prediction Overlap Rate Metric 3. However, the previous metrics are all link-wise metrics, which can be problematic in certain scenarios. For example, if the link-wise stability of two pairs of node embeddings is very low, but all other pairs have high stability, these link-wise metrics cannot capture this information and will still report relatively high stability. To address this issue, we propose a comprehensive metric that directly computes the link overlap ratio across all node embeddings under different random seeds. We decide to use this metric as the primary metric for the downstream stability task.

$$metric3_{pos}^{(g,d)} := \frac{\left|\cap_{g \in G}(TP^{(g,d)} \cup FN^{(g,d)})\right|}{M_1},$$

$$metric3_{neg}^{(g,d)} := \frac{\left|\cap_{g \in G}(TN^{(g,d)} \cup FP^{(g,d)})\right|}{M_2},$$

$$metric3_{all}^{(g,d)} := \frac{\left|\cap_{g \in S}(L^{(g,d)})\right|}{M_1 + M_2},$$

where S is the set of node embeddings with the same node embedding dimension under different random seeds.

Table 1: Statistics of empirical graph datasets

Data Set	$\|V\|$	$\|E\|$	Node Feature Dimension
Cora	2708	5429	1433
ogbl-citation2	2927963	60703760	128

3 Experiment Framework

3.1 Node Embedding Algorithms

We consider Node2Vec and GraphSAGE as the chosen embedding algorithms in our study. These algorithms were selected as they represent two distinct paradigms in the realm of node embedding techniques. Node2Vec employs a random walk-based approach to create embeddings and optimizes the objective function for learning embeddings utilizing the skip-gram techniques derived from Word2Vec [6]. This approach is also employed in LINE and DeepWalk[7, 8]. Moreover, Node2Vec is a transductive, unsupervised learning technique that does not rely on any explicit node or edge labels. In contrast, GraphSAGE operates on the graph neural network (GNN) paradigm, aggregating neighborhood information to generate embeddings [9]. It is an inductive, supervised node embedding generation algorithm that creates embeddings using node features.

Both Node2Vec and GraphSAGE are susceptible to the influence of randomness. Node2Vec's sensitivity to randomness arises primarily from its initialization and optimization processes, whereas GraphSAGE's susceptibility originates from its neighbour sampling strategy and neural network weight initialization and optimization.

3.2 Graph Datasets

We aim to investigate the stability of node embedding algorithms on datasets of varying sizes. To accomplish this, we utilize two empirical graph datasets: the Cora dataset and the ogbl-citation2 dataset. Both are citation networks, but they differ in size: the former is smaller and the latter is larger. Statistics for each graph can be found in Table 1.

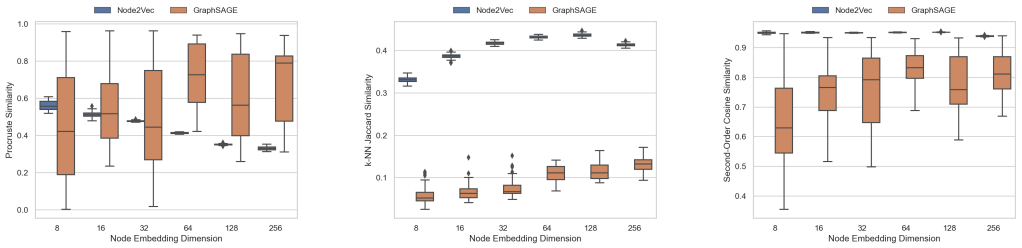
Cora [10]. In the Cora citation network, each scientific paper is depicted as a node, and a directed edge represents that the originating node cites the destination node. In addition, each paper is connected to a category representing its research subject.

ogbl-citation2. The ogbl-citation2 dataset is a directed graph representing the citation relationships between a selection of papers taken from MAG [11]. In this graph, each node represents a paper and is characterized by 128-dimensional word2vec features [12], summarising its title and abstract, while each directed edge indicates that one paper cites the other. We call this dataset as Citation2 in the following description.

3.3 Experimental Setup

For both embedding algorithms, we investigate six embedding dimensions: 8, 16, 32, 64, 128, and 256. In the Cora dataset, for each dimension, we sample ten random seeds and generate ten corresponding embeddings. In the Citation2 dataset, we generate five embeddings per dimension due to constraints on running time and computational resources. Both algorithms are executed with the default parameters provided in the PyG [13] library for the Cora dataset, and for Citation2, the default parameters in the OGB [14] codebase are used.

Geometric Stability. Geometric stability is assessed by calculating node-wise similarity values using metrics discussed in Section 2.2. We employ k = 20 for the k-nearest neighborhood-related metrics to select an appropriate neighborhood size [1]. It is worth noting that the Citation2 dataset comprises over two million nodes, rendering the computation of actual geometric stability infeasible due to high time and space complexity. To approximate geometric stability, we sample a set of nodes from the original graph and utilize their embeddings to evaluate the metrics. We compute metrics with 1,000, 2,000, 4,000, 8,000, and 16,000 nodes, respectively.



(a) Procruste Similarity **(b)** k-NN Jaccard Similarity **(c)** Second-Order Cosine Similarity

Figure 1: Geometric Stability of Node2Vec and GraphSAGE on Cora

Downstream Stability. We conduct a link prediction task to evaluate downstream stability. In the link prediction task, we provide a pair of nodes' embeddings for a predictor to decide whether an edge exists between the pair of nodes. It is worth noting that the test sets for both datasets exclusively comprise positive edges, i.e., cases where an edge exists between a pair of nodes. In order to perform an accurate evaluation of link prediction performance, we sample an equal number of negative edges, i.e., cases where no edge exists between a pair of nodes, for each dataset. Specifically, the number of negative edges sampled is identical to the number of positive edges present in each respective dataset. The evaluation metric employed for this task is the test set accuracy. The Citation2 test dataset is provided by the OGB codebase, while Cora lacks a preexisting test set. We obtain the largest connected component of Cora and partition 10% of its edges to form a test set, ensuring the subgraph remains connected. The downstream classifier used is the multi-layer perceptron (MLP). For the Cora dataset, the embeddings generated by both algorithms are evaluated using the same two-layer MLP, with the hidden dimension equal to the embedding dimension. ReLU activation units introduce non-linearity. For the Citation2 dataset, we utilize the Link Predictor (an MLP with an alternative configuration) provided by the OGB codebase for evaluation. For each embedding across dimensions and datasets, an MLP is trained independently. The training epochs for the Cora dataset are determined by monitoring the model's performance plateau; we set Node2Vec-based embeddings' training epochs to 100 and GraphSAGE-based embeddings to 10. For the Citation2 dataset, due to long training time and computational resource limitations, we train both Node2Vec and GraphSAGE-based embeddings for ten epochs.

4 Results

4.1 Geometric Stability

To assess the geometric stability within node embeddings, three metrics are employed to compute their similarity. Specifically, the mean of all pairs of embeddings' similarities is utilized as a measure of this stability. A higher mean value indicates the presence of a stable geometric state across multiple random seeds, within a given metric. Additionally, we explore the standard deviation of similarity values for each metric, as it offers insights into the stability of the observed geometric stability itself.

Concerning the Node2Vec algorithm, as illustrated in Figure 1 the standard deviation of similarities between pairs of node embeddings exhibits stability across all dimension sizes, metrics, and datasets, as these deviations are all very small. Therefore, much importance should be attached to the mean value of similarity.

For the Procrustes similarity, the mean value diminishes as the dimension increases, a trend observed in both the Cora and Citation2 datasets (Figure 1a and Figure 2). In Figure 1b, the mean value of k-NN Jaccard similarity for the Cora dataset portrays a different pattern: the similarity initially increases with the dimension size, but eventually decreases once the dimension reaches 256. Similarly, the second-cos similarity displays a different scenario: all mean similarity values are remarkably high and remain constant across varying dimension sizes (Figure 1c). However, this is not the case for the citation network (Figure 2), as considerable discrepancies exist between different dimensions.

In our study of GraphSAGE performance on the Cora dataset, as illustrated in Figure 1, we observed that using node embedding dimensions of 8, 16, or 32 generally yielded lower mean values for three

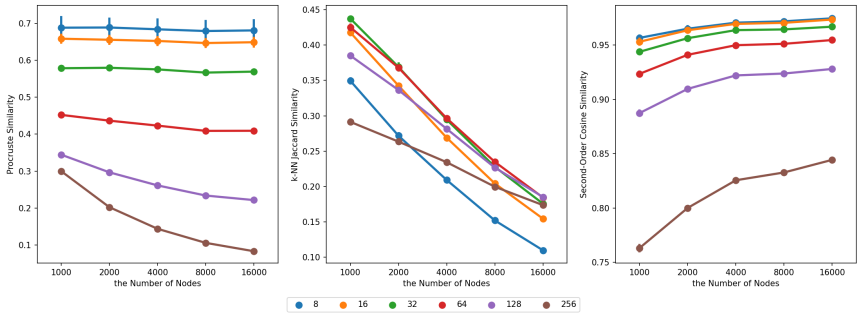


Figure 2: Geometric Stability of Node2Vec on Citation2.

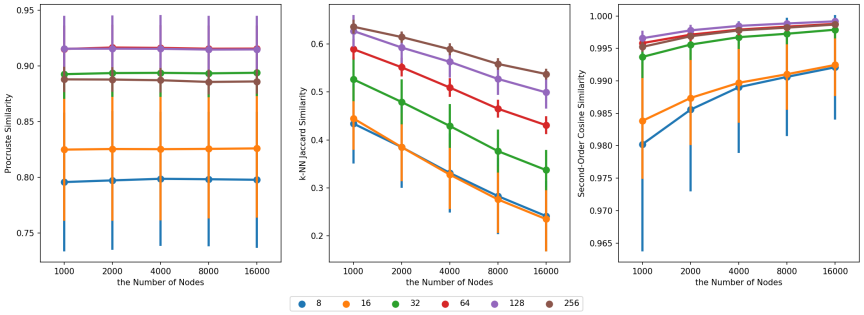


Figure 3: Geometric Stability of GraphSAGE on Citation2.

geometric stability measures compared to dimensions of 64, 128, or 256. We will refer to the former dimensions as the "smaller dimension group" and the latter as the "larger dimension group" in our following analysis. This trend is also evident in the results for the Citation2 dataset. We suppose that this outcome occurs because, during training, GraphSAGE can capture more relevant information and transfer it to the predictor for supervised learning than using lower-dimensional node embeddings. Specifically, for the Cora dataset, GraphSAGE achieves $\boldsymbol{k}$-NN Jaccard similarities slightly above zero for the smaller dimension group and around 0.1 for the larger dimension group. In the case of the Citation2 dataset (Figure 3), when the number of nodes increases from 1,000 to 16,000, the Procrustes similarity and the second-order cosine similarity remain relatively stable and maintain higher values, while the $\boldsymbol{k}$-NN Jaccard similarity has a significant decline.

Regarding standard deviation, the Cora dataset (Figure 1) exhibits different patterns across the three measures of geometric stability. The standard deviation of the Procrustes similarity does not display a clear trend based on the node embedding dimension. The larger dimension group has a higher standard deviation for the $\boldsymbol{k}$-NN Jaccard similarity while exhibiting lower results for the second-order cosine similarity. On the other hand, in the case of the Citation2 dataset (Figure 3), the larger dimension group demonstrates a lower standard deviation in all three measures compared to the smaller dimension group, which aligns with our earlier observations.

In general, we note that GraphSAGE yields lower mean values and higher standard deviation in the three measures of stability compared to Node2Vec. This suggests that, compared to Node2Vec, GraphSAGE is more unstable in generating node embeddings when encountering randomness.

4.2 Downstream Stability

4.2.1 Accuracy

In our analysis, we examine the downstream stability of the Node2Vec and GraphSage algorithms. Given that Node2Vec generates node embeddings solely based on the graph's topology information, we introduce an additional experimental setting by concatenating the generated node embeddings

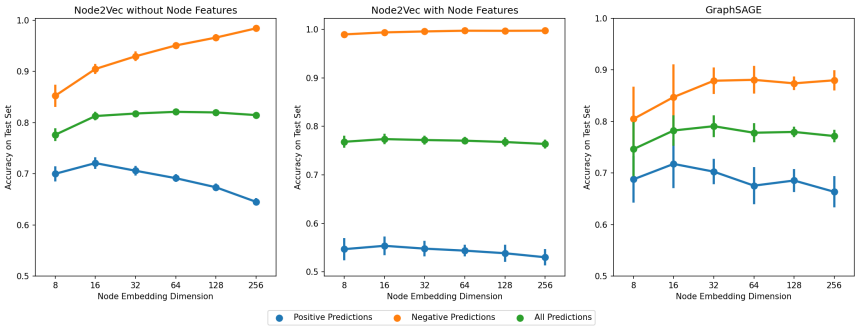


Figure 4: Accuracy on Cora.

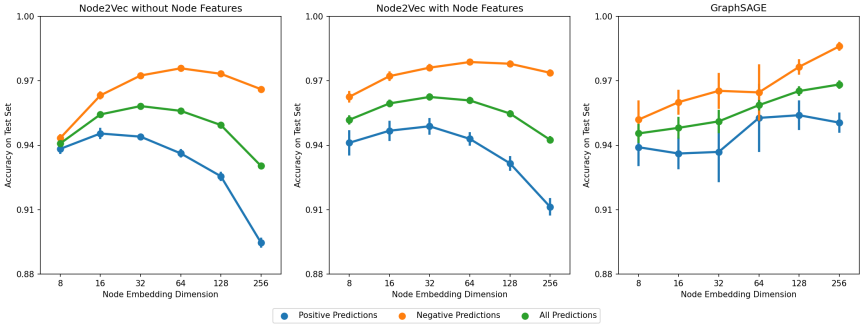


Figure 5: Accuracy on Citation2.

with the original node features to form the input features for the downstream predictor (MLP). We call this new experiment as "concat MLP" in short. We investigate the accuracy of the MLP for various embedding dimensions. In particular, we analyze the standard deviation of accuracy for different runs using the corresponding node embeddings to investigate the downstream stability.

As illustrated in the Figure 4 and Figure 5, whatever on Cora or Citation2, Node2Vec and GraphSAGE exhibit higher accuracy when predicting negative edges. One possible explanation is that Cora and Citation2 are both sparse networks, where negative edges represent a larger proportion of the total possible edges, and thus easier to be identified correctly. In order to evaluate the overall performance, we focus on the accuracy of overall predictions.

As the node embedding dimension increases, GraphSAGE consistently achieves high mean accuracy on both the Cora and Citation2 datasets, reaching approximately 75% and 95% respectively. Meanwhile, we observe a decrease in standard deviation with the increase in node embedding dimensions. This indicates that, when the node embedding dimension is higher, the model is less sensitive to the random seed. Concerning Node2Vec, the standard deviation is too small to investigate. To explore more detailed downstream stability of models, we will further discuss three kinds of link prediction overlap rates in Section 4.2.2.

4.2.2 Link Prediction Overlap Rate

Regarding Node2Vec, the analysis of overlap rate metrics indicates a positive correlation between embedding dimension and overlap rates for the Cora dataset (Figure 6, Figure 7 and Figure 8). However, in Figure 9, Figure 10 and Figure 11, the link prediction overlap rates for the Citation2 dataset demonstrate an initial increase followed by a subsequent decrease as embedding dimension increases, with a peak value observed at an embedding dimension of 32. Furthermore, a comparison of two experimental settings for Node2Vec reveals that the "concat MLP" approach is associated with greater stability in link prediction overlap rates across three metrics and two datasets. Notably, the "concat MLP" approach displays a similar trend to the original MLP.

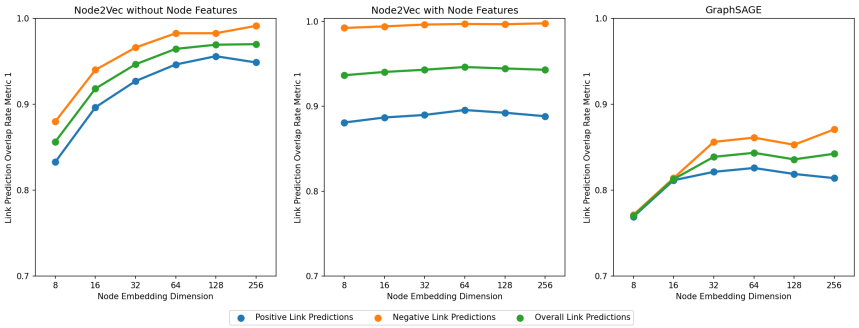


Figure 6: Link Prediction Overlap Rate Metric 1 on Cora.

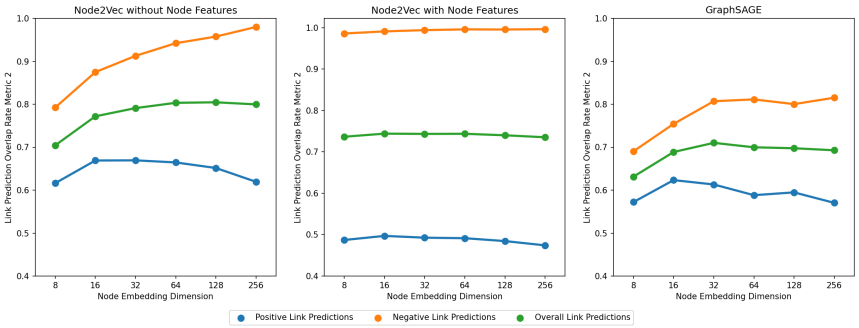


Figure 7: Link Prediction Overlap Rate Metric 2 on Cora.

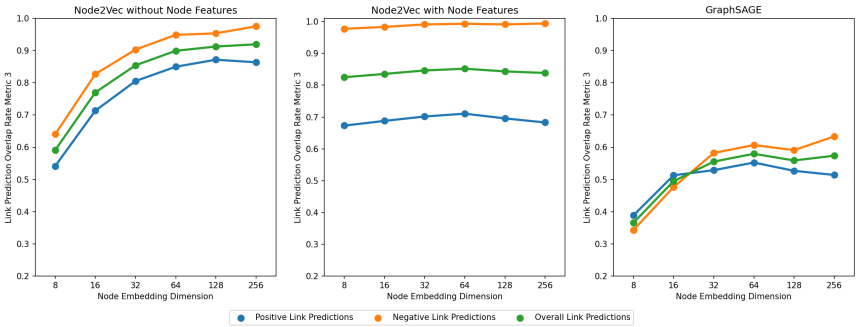


Figure 8: Link Prediction Overlap Rate Metric 3 on Cora.

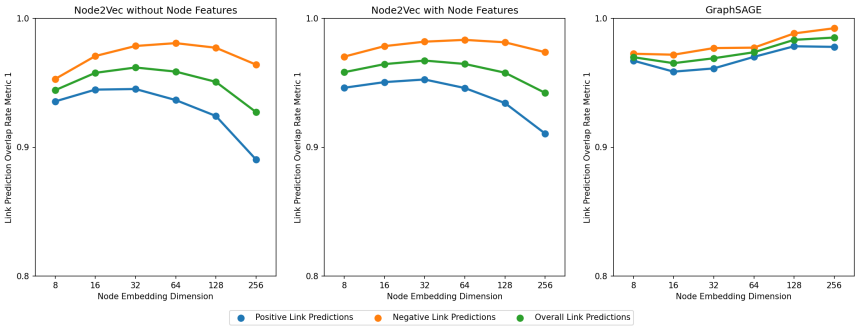


Figure 9: Link Prediction Overlap Rate Metric 1 on Citation2.

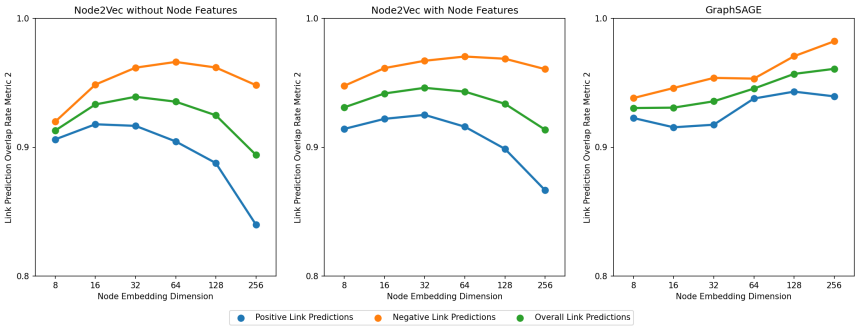


Figure 10: Link Prediction Overlap Rate Metric 2 on Citation2.

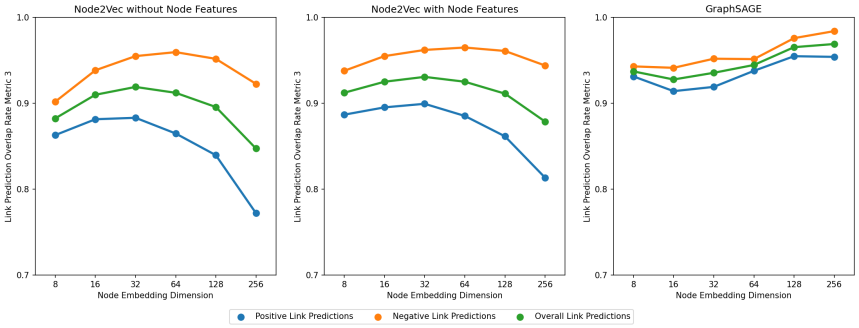


Figure 11: Link Prediction Overlap Rate Metric 3 on Citation2.

For the GraphSAGE results on both Cora and Citation2 datasets, the three metrics generally increase as the node embedding dimension rises. This observation aligns with our analysis in Section 4.2.1, a decrease in standard deviation with the increase in node embedding dimensions. More specifically, the metrics for Citation2 remain consistently high (above 0.9) across all dimensions. In contrast, Cora displays a different pattern: for instance, at a dimension of 256, Metric 1 reaches approximately 0.84, Metric 2 is about 0.70, and Metric 3 stands at around 0.55. This finding suggests that GraphSAGE can generate more stable downstream predictions on Citation2 compared to the Cora dataset, which can also be detected in Node2Vec.

5 Discussion

In this discussion, we present findings derived from the experimental results of the Node2Vec and GraphSAGE algorithms applied to the Cora and Citation2 datasets. With regard to the Node2Vec algorithm on the Cora dataset, three distinct geometric stability measures yield disparate outcomes. The Procrustes similarity metric exhibits a decline in stability, indicating that embeddings generated by Node2Vec become globally more susceptible to randomness as the embedding dimension increases. Conversely, local neighborhood-based metrics suggest that the stability of embeddings remains consistent or even improves with an increase in embedding dimension.

To ascertain the relative importance of global versus local neighborhood-based metrics for Node2Vec, we utilize downstream stability results to inform our decision. In the Cora dataset, trends in link prediction overlap rate metrics align closely with those of the kNN Jaccard similarity metric, revealing an increase in stability with increasing dimension for smaller dimensions, followed by a plateau or minor decline for larger dimensions. A similar pattern emerges in the Citation2 dataset, where the approximated geometric stability exhibits an increase-decrease trend with respect to the embedding dimension, which is same as the pattern observed in the downstream task. Consequently, we conclude that for the Node2Vec algorithm, local-based metrics, particularly the kNN Jaccard similarity metric,

should be prioritized as they offer valuable insights into the behavior of downstream stability with varying input embedding dimensions subject to randomness.

For the GraphSAGE algorithm, all three geometric stability metrics reveal greater resistance to randomness in the "larger dimension group," corroborating the downstream stability trends that generally increase with embedding dimension. Thus, for GraphSAGE, all three geometric stability metrics provide valuable insights into downstream stability.

It is pertinent to note that due to time and space constraints, the geometric stability in the Citation2 dataset is approximated by sampling a subset of nodes rather than employing the entire node set. We anticipate that approximated metric values for each dimension will exhibit greater stability as the sample size increases. This holds true for the Procrustes similarity and $\boldsymbol{k}$-NN second-order cosine similarity metrics, as their values remain relatively constant when the sample size increases from 8,000 to 16,000 nodes. However, this is not the case for the $\boldsymbol{k}$-NN Jaccard similarity metric due to the influence of "fake neighbors" in randomly selected node subsets: In a randomly selected subset of nodes from the original graph, it is improbable that the K nearest neighbors for a particular node are part of that node's neighborhood in the original graph. These so-called "fake neighbors" may be topologically distant from the target node and exhibit minimal message passing during the embedding creation process. Nonetheless, within the node subset, these fake neighbors exhibit the highest cosine similarity with the target node by coincidence, resulting in their selection as the nearest neighbors. Comparing these fake neighbors of a pair of nodes in the $\boldsymbol{k}$-NN Jaccard similarity metric is inconsequential and devoid of meaningful interpretation.

Additionally, we identify a relationship between downstream accuracy and link-prediction overlap rates, whereby higher overlap rates (indicating greater stability in the face of randomness) correspond to higher accuracy at specific embedding dimensions. For instance, in the Citation2 dataset for the Node2Vec algorithm, both link-prediction overlap rates and downstream task accuracy peak at an embedding dimension of 32. This contradicts the intuitive notion that link-prediction overlap rates are solely related to the standard deviation of accuracy. Given the established connection between embedding geometric stability and downstream stability (represented by link-prediction overlap rates), this novel finding may suggest a potential relationship between embedding geometric stability and downstream link prediction task accuracy across different embedding dimensions.

6 Future Work

Our study is limited to the stability of node embeddings in small and large citation networks, and we were unable to compute the factual geometric stability metrics of node embeddings in Citation2 due to resource limitations. However, in future studies, we will investigate whether our findings hold across other types of datasets with different graph properties such as size, density, node feature dimension, and context.

Moreover, it's important to note that our work is primarily an empirical study, where we observe the stability (or instability) of node embeddings generated by different algorithms and provide some possible explanations for the observed behaviour. However, we acknowledge that our study does not offer a systematic or principled theoretical explanation for the underlying causes of stability. Thus, future research could focus on more theoretical analysis, which would be valuable for developing more stable node embedding algorithms.

7 Conclusion

In this paper, we aim to the impact of randomness on node embeddings for various embedding dimensions. We propose some metrics to evaluate the geometric stability and the downstream stability of node embeddings generated by Node2Vec and GraphSAGE on Cora and Citation2. Our study reveal distinct behaviors of Node2Vec and GraphSAGE with respect to randomness across different embedding dimensions. We also observe a positive relationship between the geometric stability of node embeddings and downstream task performance for a specific embedding dimension.

References

[1] Tobias Schumacher, Hinrikus Wolf, Martin Ritzert, Florian Lemmerich, Martin Grohe, and Markus Strohmaier. The effects of randomness on the stability of node embeddings. In *Machine Learning and Principles and Practice of Knowledge Discovery in Databases: International Workshops of ECML PKDD 2021, Virtual Event, September 13-17, 2021, Proceedings, Part I*, pages 197–215. Springer, 2022. 1, 5

[2] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40:33–51, 1975. 2

[3] Wojtek J Krzanowski and Wojtek Krzanowski. *Principles of multivariate analysis*. Oxford University Press, 2000. 2

[4] Maria Antoniak and David Mimno. Evaluating the stability of embedding-based word similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119, 2018. 3

[5] William L Hamilton, Jure Leskovec, and Dan Jurafsky. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the conference on empirical methods in natural language processing. Conference on empirical methods in natural language processing*, volume 2016, page 2116. NIH Public Access, 2016. 3

[6] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 5

[7] Miguel Angel Lozano, Francisco Escolano, Manuel Curado, and Edwin R Hancock. Network embedding from the line graph: Random walkers and boosted classification. *Pattern Recognition Letters*, 143:36–42, 2021. 5

[8] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 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*, pages 459–467, 2018. 5

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 5

[10] Lovro Šubelj and Marko Bajec. Model of complex networks based on citation dynamics. In *Proceedings of the 22nd international conference on World Wide Web*, pages 527–530, 2013. 5

[11] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 1(1):396–413, 2020. 5

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. 5

[13] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 5

[14] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020. 5