The Effects of Randomness on the Stability of Node Embeddings*

Tobias Schumacher[1,2] [0000−0003−3091−5095],
Hinrikus Wolf[1,2] [0000−0003−4579−3633],
Martin Ritzert[1,2] [0000−0002−5322−3684],
Florian Lemmerich[3] [0000−0001−7620−1376],
Martin Grohe[2] [0000−0002−0292−9142], and
Markus Strohmaier[2,4,5] [0000−0002−5485−5720]

[1] Equal contribution
[2] RWTH Aachen University, Aachen, Germany
{tobias.schumacher,markus.strohmaier}@cssh.rwth-aachen.de
{hinrikus,ritzert,grohe}@cs.rwth-aachen.de
[3] University of Passau
florian.lemmerich@uni-passau.de
[4] GESIS - Leibniz Institute for the Social Sciences
[5] Complexity Science Hub Vienna

Abstract. We systematically evaluate the (in-)stability of state-of-the-art node embedding algorithms due to randomness, i.e., the random variation of their outcomes given identical algorithms and networks. We apply five node embeddings algorithms—HOPE, LINE, node2vec, SDNE, and GraphSAGE—to assess their stability under randomness with respect to their performance in downstream tasks such as node classification and link prediction. We observe that while the classification of individual nodes can differ substantially, the overall accuracy is mostly unaffected by the geometric instabilities in the underlying embeddings. In link prediction, we also observe high stability in the overall accuracy and a higher stability in individual predictions than in node classification. While our work highlights that the overall performance of downstream tasks is largely unaffected by randomness in node embeddings, we also show that individual predictions might be dependent solely on randomness in the underlying embeddings. Our work is relevant for researchers and engineers interested in the effectiveness, reliability, and reproducibility of node embedding approaches.

Keywords: node embedding, graph embedding, node classification, link prediction, reliability, representation learning, embedding stability

* This work is supported by the German research council (DFG) Research Training Group 2236 UnRAVeL, the Federal Ministry of Education and Research (BMBF), and the Ministry of Culture and Science of the German State of North Rhine-Westphalia (MKW). We thank Jan Bachmann, Max Klabunde, and Florian Frantzen for their help with implementing and running the experiments.

1 Introduction

Many state-of-the-art node embedding algorithms make explicit use of randomness in parameter initialization, edge sampling, or through stochastic optimization. Thus, the application of the same algorithm with identical parameters on the exact same graph data can lead to different embeddings.
Recent research [19] has provided an initial assessment of such instabilities, in particular with respect to the geometry of the embedding spaces. Yet, the impact of these instabilities on the outcomes of downstream tasks such as node classification and link prediction has not been systematically evaluated.
Research Objective. We investigate the effects of randomness on the stability of node embeddings. Towards this end, we specifically focus on assessing the *downstream stability* of node embeddings, i.e., the stability of outcomes from tasks such as node classification and link prediction.
Approach. We conduct experiments with five state-of-the-art embedding algorithms on empirical network datasets. For each embedding algorithm, we compute multiple node embeddings with the same parameters on the same networks but with different random seeds. Specifically, we apply HOPE [12], LINE [17], node2vec [4], SDNE [20], and GraphSAGE [6]. On the resulting embeddings, we then perform node classification and link prediction to quantify downstream stability with respect to these tasks. In that regard, we consider both stability in overall performance and stability of individual predictions.
Results and Implications. We find that despite substantial geometric instabilities, which have been reported in previous work [19] as well as our own preliminary experiments, the overall accuracy in node classification and link prediction is almost constant. This indicates a surprising stability in downstream tasks. At the same time, we show that the actual predicted classes of individual nodes can—and often do—differ between classifiers trained on embeddings based on different random seeds. For link prediction, we observe similar trends, although the stability of the single predictions is much higher than for node classification. This higher stability is however likely due to a higher overall accuracy in the considered scenarios for this task, which leaves less room for different misclassifications.

Overall, our work contributes towards a more fundamental understanding of the stability of node embeddings, and thereby opens up ways for more informed deployments and a better understanding of the effects of randomness on embedding-based predictions.

2 Related Work

Our paper extends a recent study by Wang et al. [19], who conducted their research independently and in parallel to ours. They provide an initial assessment of the issue of instability of node embeddings with an emphasis on geometric stability. Next to finding significant instabilities over most algorithms for both global and node-based stability, they perform a factor analysis to identify the

main sources of those instabilities. The factor analysis suggests that the impact of dataset-dependent features such as size and density, as well as node properties such as closeness centrality, have higher impact than algorithmic parameters. They identified a correlation between embedding stability and node classification accuracy with SVMs. However, they did not investigate to which extent the accuracy of repeated downstream tasks varies and how far individual predictions differ in these downstream tasks.

In a set of preliminary experiments (cf. Appendix B), we confirmed the fundamental geometric instabilities which Wang et al. [19] have reported. However, we could not confirm the impact of network size and density as well as node centrality. Our paper complements their work by providing a thorough analysis on the impact of instability on downstream predictions, i.e., predictions occurring when combining embeddings with other machine learning algorithms.

Aside from the study by Wang et al. [19], there has not been any additional previous study on the stability of node embeddings. However, the issue of embedding instability has been thoroughly studied in the context word embeddings, which has also influenced our work.

The first work to point out instabilities in word embeddings has been conducted by Hellrich and Hahn [7]. They discovered that neighborhoods of words in the embedding space change significantly even under fixed corpora. These instabilities have been confirmed and further investigated by Antoniak and Minmo [1]. Both studies [7, 1] report significant instabilities of skip-gram-based word embedding methods with respect to local neighborhood similarities. To investigate which word properties influence stability, Wendlandt et al. [22] and Pierrejean and Tanguy [13] conducted regression-based factor analyses. They correlated the stability of a word embedding with semantic features such as a word's part of speech, as well as algorithmic parameters such as the dimensionality of the embedding space. Finally, Leszczynski et al. [10] specifically analyzed the relationship between geometric stability of word embeddings and the resulting instabilities in downstream tasks. They introduced an Eigenspace instability measure to quantify geometric instability, and proved that this measure theoretically determines the expected downstream disagreement on linear regression tasks. In our study, we directly measure the variance of downstream tasks with non-linear classifiers, such that this instability measure is not applicable.

3 Experimental Framework

Our main set of experiments quantifies the downstream stability of five state-of-the-art node embedding algorithms. We start with a short description of the algorithms and datasets and then describe the experiments. The code for our experiments is published on GitHub.[6]

We consider the following five node embedding algorithms as representatives of the spectrum of currently existing approaches. The spectral embedding algorithm *HOPE* [12] factorizes the Katz similarity matrix. *LINE* [17] embeds the

[6] All code available on `https://github.com/SGDE2020/embedding_stability`

local and global neighborhood structures separately and combines the resulting embeddings. *node2vec* [4] applies the word embedding algorithm *word2vec* on random walks generated from the network. *SDNE* [20] computes embeddings based on the encoder-decoder principle. The inductive node embedding algorithm *GraphSAGE* [6] applies a GNN to compute its embeddings.

We investigate the downstream stability of node embeddings on four graph datasets, which cover a broad spectrum of commonly used empirical graphs: the social graph BlogCatalog [23], the citation graph Cora [16], as well as the datasets Protein [15] and Wikipedia [11]. Statistics for each graph can be found in Table 1 in Appendix A.1.

Overview of Experiments. To analyze the impact of randomness in node embeddings on the outcomes of downstream predictions, we computed for each dataset 30 embeddings with each embedding algorithm, all with the embedding dimension of 128 and mostly standard parameters. We consider the two most common downstream tasks, node classification and link prediction. We evaluate two types of downstream stability, first the *stability of performance* and second the *stability of single predictions*. In stability of performance, we measure the variance of general performance scores such as micro-F1 of the classification on a holdout set. To quantify the stability of single predictions, we train (i) multiple classifiers on the same embedding and (ii) multiple classifiers on multiple embeddings of the same network produced by the same embedding algorithm. Differences in the classifications in (i) indicate the stability of the classification algorithm itself due to random elements in the classification algorithm, independent of the embedding. Such random elements naturally occur in most learning algorithms. Comparing outcomes of classifiers trained on different embeddings (ii) provides an indication of the combined stability of the embedding algorithm and the classifier. Thus, the difference between the outcomes of (i) and (ii) corresponds to the influence of the instability of the embeddings on the stability of the classification. To measure differences in the outcome of classifiers, we use general performance scores (such as micro-F1 of the classification on a holdout set) as well as the *stable core*, i.e., the ratio of nodes that are assigned to the same class in at least 90% of the classifier runs.

Since different machine learning algorithms have very different characteristics, we use multiple classifiers, namely AdaBoost, decision trees, random forests, and feedforward neural networks. For node classification, we performed a 10-fold cross-validation with 10 repetitions. For link prediction, we were able to generate a sufficient amount of training data and thus left out the cross-validation. More details on the parameterization of both the embedding and classification algorithms can be found in Appendix A.2.

4 Results

In a set of preliminary experiments (cf. Appendix B), we have found that all embedding algorithms except HOPE, which yields near-constant embeddings, display substantial geometric instabilities. These results are in line with results

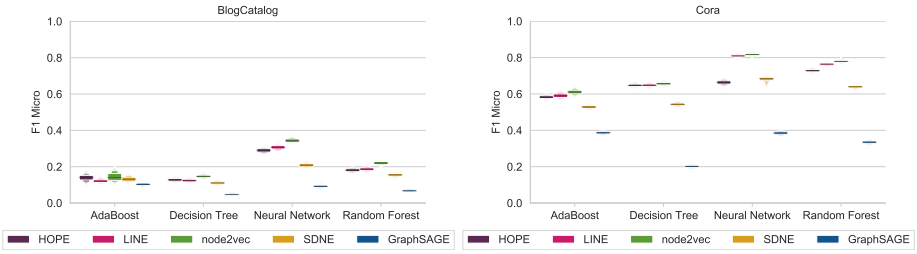


Fig. 1: Stability of classification performance of BlogCatalog and Cora.

from Wang et al. [19]. In this section, we present the results from our experiments on the instability of downstream tasks as described in Section 3. We begin with the results for node classification and then continue with the stability of link prediction.

Node Classification. We first analyze the *stability of performance* in the node classification task. Due to limited space, we only present and discuss the results on BlogCatalog and Cora here. Results for the other datasets can be found in Appendix C. Figure 1 depicts the micro-F1 scores of the predictions. Each box in the figure aggregates the different micro-F1 scores of the repeated predictions on the 30 embedding per algorithm and dataset. We observe that the F1 scores of all classification tasks vary only marginally. Aside from stability, we observe a strong dependence of the micro-F1 scores on the classification algorithm, but not so much on the embedding algorithm, except for GraphSAGE which is always lower in performance.

Next, we investigate the stability of individual *node-wise predictions*. For that purpose, we determine the stable core of predictions over multiple classification runs, i.e., the ratio of nodes which are classified to have the same labels in 90% of all predictions. To distinguish between (i) instability originating from the classifiers and (ii) instability originating from the underlying embeddings, we compute the stable cores in two distinct settings. For (i), we train each classifier ten times on a fixed embedding and averaged the sizes of the resulting stable cores over five embeddings, for (ii), we trained each classifier once on all 30 embeddings. The results are shown in Figure 2 where the stable cores from (i) are depicted in saturated colors and the stable cores from (ii) are shown in light colors. Compared to the stability of performance, the picture of stability in node-wise predictions is more mixed. Our first observation is that since the embeddings generated by HOPE are almost identical, also the stables cores from (i) and (ii) are about the same size over all datasets and classifiers. For the remaining embedding algorithms, there is no clear trend recognizable. Next to the embedding algorithm, the choice of classifier seems to have a high impact on the stability of individual predictions. For AdaBoost, we observe almost stable predictions under fixed embeddings, while for varying embeddings, stability is highly dependent on the embedding algorithm. In contrast, the individual predictions of the decision trees are relatively unstable, in particular under varying

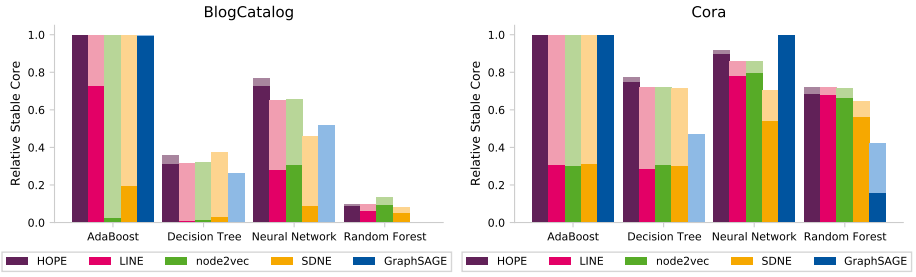


Fig. 2: Stability in node-wise predictions. We depict the mean stable core of predictions under varying embeddings in saturated colors, and the mean stable core of multiple predictions on fixed embeddings in lighter colors.

embeddings. For random forests, we observe that the observed instabilities are mostly due to the classification algorithm itself. However, the degree of instability strongly varies over the datasets. Finally, for neural networks we observe that the degree of instability varies over both datasets and there is no clear trend on whether varying classifiers or varying embeddings have a stronger impact on the stability of individual predictions.

In general, we observe that both the chosen embedding algorithm as well as the selected classifier have a high influence on the stability of individual predictions. The impact of classifier and embedding algorithm varies over different datasets.

Link Prediction. For simplicity, we show the results of our link prediction experiments only on BlogCatalog, which are depicted in Figure 3. The results on the other datasets can be found in Appendix C. The first observation is that in this binary task, the accuracies are naturally much higher than for multi-class and multi-label node classification. Further, we see low variances in those accuracies. In terms of the stability of individual link predictions, we also observe more stable individual predictions than in node classification. AdaBoost is almost perfect in repeating the task on the same input data, although the accuracy varies between 0.5 and 0.95 depending on the embedding. Decision trees also achieve highly reproducible predictions for varying embeddings, whereas for neural networks and in particular random forests there is a stronger dependence on the embeddings. For LINE and SDNE embeddings, we observe that most predictions stay the same independently of the underlying embedding, despite their geometric instability (see [19] and Appendix B). For node2vec and GraphSAGE, there is, however, a relatively high fluctuation in the predictions that results from instabilities in the embeddings. For HOPE, we confirm the stability of the predictions that is to be expected given its almost fully stable embeddings.

As expected, we see a strong dependence between performance and the individual predictions, i.e., higher performance in terms of accuracy corresponds to larger stable cores. Again, we observe a high impact of the embedding algorithm on the stability of individual predictions.

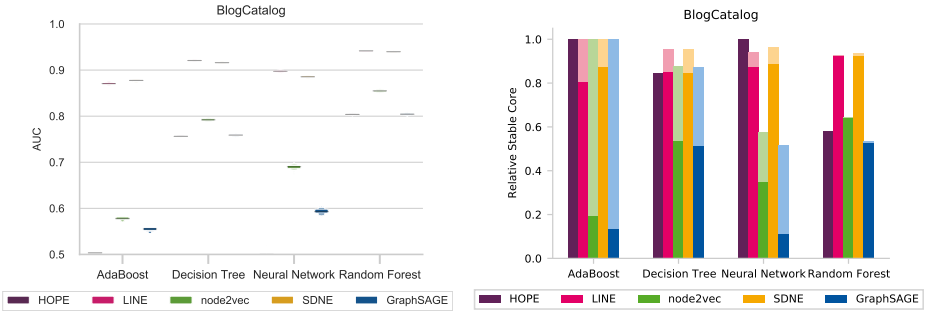


Fig. 3: Stability in link prediction on BlogCatalog. Left: Stability of accuracy. Right: Stability of individual predictions as difference between the mean stable core over all embeddings (saturated colors) and the mean stable core over repeatedly trained classifiers on fixed embeddings (lighter colors).

5 Discussion

Next, we discuss our experimental results, including potential explanations, relationship to prior research, implications, and limitations of our work.

Summary of Results. Our results show that the overall classification performance in both node embedding and link prediction is mostly unaffected by random variations in the embeddings, which were observed in preliminary experiments (cf. Appendix B) and previous work [19]. However, the actual predicted classes for single nodes vary depending on the embedding that the classifier was trained on, i.e., due to the randomness in the embeddings, different classifications are produced. To a lesser extent, this effect was also observed when analyzing the predictions of individual links.

Potential Explanations for Results. A potential explanation for the surprising stability in the overall classification performance is that classifiers seem to be able to extract and utilize local structural information from embeddings even if their global structure changes. This means that even in very different embeddings, the necessary information for a model that generalizes well is contained in each of those embeddings. Since the classifications of single nodes or edges are not nearly as stable as the overall performance, we conclude that for different underlying embeddings, the learning algorithm chooses to focus on different parts of the embedding. On other hand, the fluctuations in individual predictions fit with the geometric instabilities. Further, when the overall classification performance is high, there is not much room for variations in individual predictions, which we especially observed on the easier link prediction task.

Relation to Existing Stability Results. Overall, the results from our work complement the findings by Wang et al. [19] on embedding instability. In a set of preliminary experiments, we have confirmed the substantial geometric instabilities which they pointed out in their work. However, we did not observe a strong

impact on downstream performance, which they have reported in a smaller experiment. Only when considering individual predictions, we observed substantial instabilities. For link prediction, we observed a relatively high downstream stability, again contrasting the results by Wang et al. [19].

Implications. In the authors' opinion, the outcomes of this paper have significant impact on the research of node embeddings. Since node embeddings vary just based on their internal random processes, great care must be taken in their evaluation and, if possible, experiments should be repeated several times in order to estimate and limit the influence of randomness and enable reproducibility of results. In settings in which unstable predictions are not problematic, for example for product recommendations, node embedding algorithms can safely be applied since the overall predictive performance is not influenced by the geometric stability. However, reproducibility of algorithms has emerged as a key factor for building trust in algorithmic decisions, which requires a high stability of predictions. This is especially important for high-stakes real-world decisions based on node embeddings. Practitioners should be aware that node embeddings add another level of uncertainty to individual (e.g., classification) decisions.

Limitations. The stability of the investigated algorithms might be strongly influenced by their concrete implementations. In that regard, we picked reference implementations from the respective research papers or—if that was not possible—established code bases for the different algorithms. However, we cannot rule out that some (in-)stabilities we observed are a consequence of implementation details. Since the chosen implementations are widely used, our results are still highly relevant for researchers and practitioners. In our experiments, we did not aim for optimal performance, but for a comparable standard setting. Thus, we did not perform extensive hyperparameter optimization for each individual task, but relied on default parameters for each algorithm. We expect a slightly higher stability with optimized hyperparameters due to higher accuracies.

6 Conclusion

In this work, we analyzed the effects of instabilities in node embeddings on the predictions in downstream tasks. Despite substantial variations in the geometry of the embedding space, which have been pointed out in previous work [19] and confirmed in our own preliminary experiments (cf. Appendix B), we found that the overall performance in the downstream tasks node classification and link prediction only displays small deviations. However, we found considerable variations when looking the classifications of single nodes, and, to a smaller extent, in the prediction of single links.

In the future, we anticipate investigations of stability and robustness of node embedding algorithms towards an in-depth study of the effects of different embedding sizes and graph modifications such as deletions or additions of nodes or edges. Furthermore, we see an opportunity for developing measures that will allow to estimate the potential instability of an embedding without computing it multiple times.

References

1. Antoniak, M., Mimno, D.: Evaluating the stability of embedding-based word similarities. Transactions of the Association for Computational Linguistics **6**, 107–119 (2018)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. science **286**(5439), 509–512 (1999)
3. Goyal, P., Ferrara, E.: GEM: A Python package for graph embedding methods. J. Open Source Software **3**, 876 (2018)
4. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
5. Hamilton, W.L., Leskovec, J., Jurafsky, D.: Diachronic word embeddings reveal statistical laws of semantic change. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1489–1501 (2016)
6. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. pp. 1024–1034 (2017)
7. Hellrich, J., Hahn, U.: Bad company—neighborhoods in neural embedding spaces considered harmful. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2785–2796 (2016)
8. Kunegis, J.: Konect: the koblenz network collection. In: Proceedings of the 22nd International Conference on World Wide Web. pp. 1343–1350 (2013)
9. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data` (Jun 2014)
10. Leszczynski, M., May, A., Zhang, J., Wu, S., Aberger, C., Re, C.: Understanding the downstream instability of word embeddings. In: Proceedings of Machine Learning and Systems 2020. pp. 262–290 (2020)
11. Mahoney, M.: Large text compression benchmark (2011)
12. Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W.: Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1105–1114. ACM (2016)
13. Pierrejean, B., Tanguy, L.: Predicting word embeddings variability. In: Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics. pp. 154–159 (2018)
14. Rozemberczki, B., Davies, R., Sarkar, R., Sutton, C.: Gemsec: Graph embedding with self clustering. arXiv preprint arXiv:1802.03997 (2018)
15. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: Biogrid: a general repository for interaction datasets. Nucleic acids research **34**(suppl_1), D535–D539 (2006)
16. Šubelj, L., Bajec, M.: Model of complex networks based on citation dynamics. In: Proceedings of the 22nd international conference on World Wide Web. pp. 527–530. ACM (2013)
17. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067–1077. International World Wide Web Conferences Steering Committee (2015)
18. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 817–826. ACM (2009)

19. Wang, C., Rao, W., Guo, W., Wang, P., Liu, J., Guan, X.: Towards understanding the instability of network embedding. IEEE Transactions on Knowledge and Data Engineering pp. 1–1 (2020)
20. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1225–1234. ACM (2016)
21. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. nature **393**(6684), 440 (1998)
22. Wendlandt, L., Kummerfeld, J.K., Mihalcea, R.: Factors influencing the surprising instability of word embeddings. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2092–2102 (2018)
23. Zafarani, R., Liu, H.: Social computing data repository at ASU (2009), `http://socialcomputing.asu.edu`

A Experimental Setup

A.1 Datasets

We provide some more details on the graphs datasets that were used in our experiments. Note that the Facebook dataset has only been used in our preliminary experiments on embedding geometry (cf. Appendix B), as it does not provide any node labels. Statistics for each graph can be found in Table 1.

- **BlogCatalog**: This graph models the relationships among the users of the BlogCatalog website. Each user is represented by a node and two nodes are connected if the respective users are friends. Each user additionally has one or more labels which correspond to the news category their blog belongs to.
- **Cora** [16]: In the well-known Cora citation network each scientific paper is represented by a node, and a directed edge indicates that the outgoing node cites the target node. Each paper is associated with a category that refers to its research topic.
- **Facebook** [14]: The Facebook government dataset models the social network structure of verified government sites on Facebook. Each site is represented by a node and nodes are connected by an edge if both sites like each other.
- **Protein [15]**: This biological network models protein interactions in human beings. Each node represents a protein and two nodes are connected if the corresponding proteins interact with each other. Additionally, each node is associated with one or more labels that represent biological states.
- **Wikipedia** [11]: This network represents the co-occurrence of words within a dump of Wikipedia articles. Each word corresponds to a node, and weighted edges represent the number of times two words occur in the same context. Additionally, each node has one or more labels that encode its part of speech.

We used the Cora dataset from the KONECT graph repository [8] and BlogCatalog from the ASU Social computing repository [23]. The other empirical datasets were taken from the SNAP graph repository [9].

Table 1: *Statistics of empirical graph datasets.* We show number of nodes (—V—) and edges (—E—), density, and number of node labels. MC indicates multi class, ML multi label problems.

Data Set	$\|V\|$	$\|E\|$	Density	# Labels
BlogCatalog	10,312	333,983	0.00628	39 (ML)
Cora	23,166	91,500	0.00034	10 (MC)
Facebook	7,057	89,455	0.00359	-
Protein	3,890	76,584	0.01012	50 (ML)
Wikipedia	4,777	184,812	0.01620	40 (ML)

A.2 Implementations and Parameter Settings

To complement Section 3, in the following we give a more detailed overview on the chosen implementations and parameter settings of the node embedding algorithms, as well as the experimental setups of the downstream classification tasks that we used in our experiments.

Node Embedding Algorithms. For every algorithm from Section 3 we use the reference implementation except for HOPE, for which no reference implementation was published. Thus we resorted to the HOPE implementation from the GEM library [3]. We run the algorithms with default parameters from the given implementations whenever possible and compute embedding vectors of length $d = 128$. We adapted SDNE to use only a single intermediate layer and for larger graphs increased the weight on the reconstruction error and the regularization term, as otherwise SDNE maps all nodes onto the same vector.

Downstream Classification. For both node classification and link prediction, we use AdaBoost, decision trees, random forests, and feedforward neural networks as downstream classification algorithms. For all classifiers we used the standard methods with default parameters from scikit-learn (AdaBoost, decision tree, random forest) and TensorFlow (neural networks). In the case of neural networks, we use a network with a single hidden layer of width 100 with ReLu activation and an output layer with softmax or sigmoid activation depending on the classification type. Deeper and wider networks did not improve performance which is why we worked with this very simple architecture.

In node classification we predict either the class of a node, e.g., top-level research category in Cora, or a set of labels of a node, e.g., the news categories in BlogCatalog. In the latter case of multi label classification, we assume that we know the number l of labels and thus predict the l most probable labels. This approach leads to more stable predictions and is common in literature [18].

For the link prediction task, we considered subgraphs of each network where we removed 10% of the original edges at random while ensuring that the residual graph is still connected. For each reduced network, we computed 10 embeddings per algorithm. We then interpreted link prediction as a binary classification task on the Hadamard product of two embedding vectors. The removed edges are then the positive examples for the link prediction, and we chose as many non-edges at random as negative examples for training the classifier.

For the stability of performance, we compute the variance of micro-F1 scores over one classifier computed on each of the 30 embeddings per graph and embedding algorithm in node classification, and each of the 10 embeddings per graph and embedding algorithm in link prediction. In both experiments, macro-F1 yields very similar results such that we only report micro-F1.

For the stability of single classifications, we have to separate inherent instability of the classifiers from the influence of different embeddings. We estimate the instability of a classifier by running it 10 times on a single embedding, averaged over 5 embeddings. The total variance in individual predictions is computed on the results of one classifier trained on each of the 30 embeddings using 75% of the nodes for training and 25% for evaluation.

B Experiments on Geometric Stability

In this section, we present our preliminary experiments on the geometric stability of node embeddings. We first give a brief description of the measures for geometric stability, and then present the results.

B.1 Measures for Geometric Stability

To quantify geometric instability of node embeddings, we use two measures which have been introduced in related literature on word embeddings, namely *aligned cosine similarity* [5] and *k-NN Jaccard similarity* [1].

The aligned cosine similarity computes the node-wise cosine similarity between two embeddings after aligning the axes of the corresponding embedding spaces. To obtain the optimal alignment, we normalize all embedding vectors and solve the Procrustes problem: Given two embedding matrices $Z^{(1)}, Z^{(2)}\mathbb{R}^{N\times d}$, with N denoting the number of nodes in a given network, and d denoting the embedding dimension, we determine the transformation matrix $Q \in \mathbb{R}^{d\times d}$ by solving the minimization problem

$$Q \coloneqq \underset{Q^TQ=I}{\operatorname{argmin}} \left\| Z^{(1)}Q - Z^{(2)} \right\|_F .$$

The k-NN Jaccard similarity measure compares the local neighborhoods of nodes between different embeddings. In both embedding spaces, we compute for a node u the k nearest neighbors with respect to cosine similarity. We then calculate the Jaccard similarity of the two nearest-neighbor sets of u.

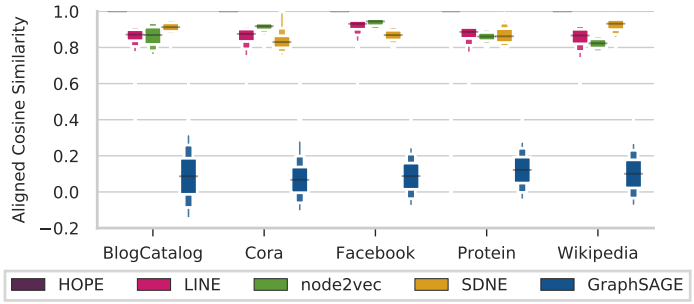
Each of those two measures computes a score for a single node in two embeddings. In order to obtain a score for an embedding space to compare different algorithms, we average over all pairs of embeddings and all nodes.

B.2 Experimental Results

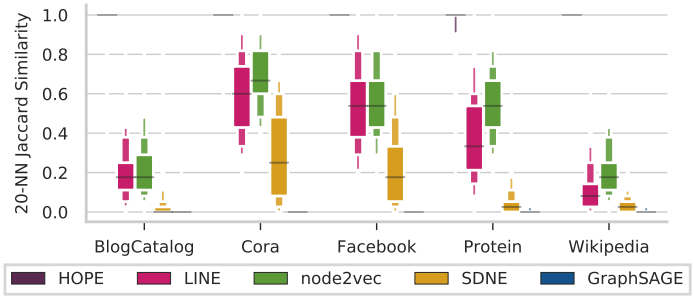
In our experiments on geometric stability, we used the same algorithmic parameter settings and datasets that have been introduced in Appendix A. Next to the overall stability of the embeddings, we also look into the *influence of node centrality*, and the *influence of network size and density* on the stability of node embeddings.

Geometric Stability. We start our analysis by computing 30 embeddings per dataset with every algorithm. We then compute node-wise stability measures averaged over all pairs of embeddings computed per graph and embedding algorithm. Figure 4 shows the distributions of (a) aligned cosine similarity and (b) k-NN Jaccard similarity over the nodes of each graph.

For the aligned cosine similarity, we observe that GraphSAGE achieves similarities that are generally only slightly above zero and sometimes even negative. Negative values correspond to angle differences of more than 90 degrees between two embeddings of the same node. Thus, even after aligning axes, embedding



(a) Variability of aligned cosine similarity.

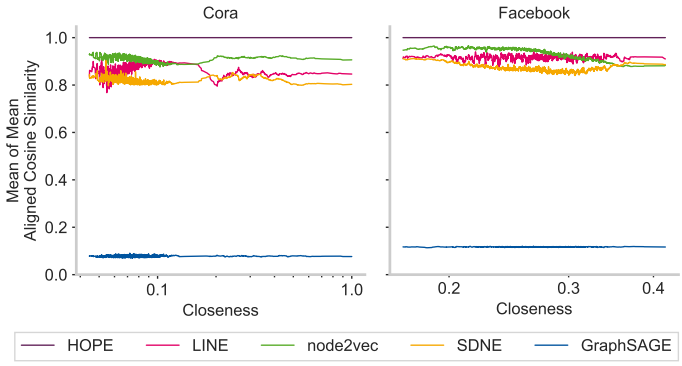


(b) Variability of $\boldsymbol{k}$-NN Jaccard similarity.

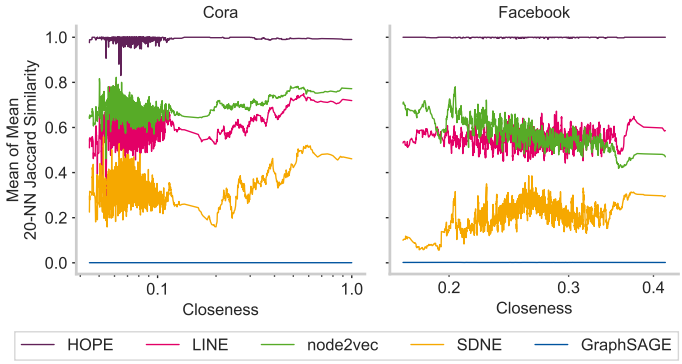
Fig. 4: *Geometric stability.* Each letter-value plot shows the node-wise similarity values resulting from 30 runs per algorithm and graph. In (a) we use aligned cosine similarity, in (b) 20-NN Jaccard similarity.

vectors of the same node are mostly close to orthogonal to each other. In contrast, HOPE yields near-constant embeddings (not shown) and shows hardly any instability. The algorithms SDNE, node2vec and LINE achieve aligned cosine similarities in the interval $(0.8, 0.9)$ with low variances. These values correspond to angles between 25 and 35 degrees such that corresponding embedding vectors roughly point in the same direction after aligning the embedding spaces. Thus, the latter algorithms exhibit a moderate, but significant degree of instability in their embeddings.

Results for the k-NN Jaccard similarity, as shown in Figure 4(b), generally confirm these findings. For HOPE, we observe perfectly matching neighborhoods, while for GraphSAGE the neighborhoods are completely disjoint. This matches our observations for aligned cosine similarity. For the other three algorithms, the resulting similarities seem to be highly dependent on the dataset, with quite large variances. Generally, node2vec appears most stable among these algorithms, though only by a slight margin over LINE. SDNE appears to the significantly less stable than node2vec and LINE with respect to Jaccard similarity, with similarity values close to zero on BlogCatalog, Protein and Wikipedia. This



(a) Node-wise aligned cosine similarity against closeness centrality.



(b) Node-wise 20-NN Jaccard similarity against closeness centrality.

Fig. 5: *Influence of node centrality.* The moving averages of the node-wise (a) aligned cosine similarities and (b) 20-NN Jaccard similarities resulting from 30 embeddings per graph are plotted against each node's closeness centrality.

contrasts the results with respect to aligned cosine similarity, where SDNE appeared as stable as the other two algorithms.

Influence of Node Centrality. Now, we analyze whether nodes that are central in their graph have more stable embeddings. Closeness centrality has been identified to be one of the top influence factors for stability in the analysis of Wang et al. [19]. Also, from the definition of node2vec we expect this algorithm, among others, to produce more stable central node embeddings since central nodes occur more often in random walks. In Figure 5, for the Cora and Facebook datasets we plot each node's closeness centrality against a moving average with window size 25 of their average node-wise (a) k-NN Jaccard similarity, and (b) k-NN angle divergence, aggregated over all 30 embeddings per network and algorithm. First of all, the (in)stability of the extreme cases HOPE and Graph-

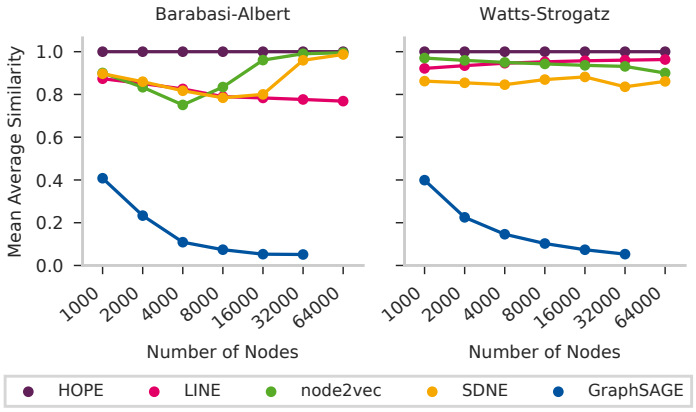
SAGE appears invariant of the centrality of the node, both in (a) and (b). For SDNE, we observe that stability with respect to k-NN Jaccard similarity appears to increase with growing closeness centrality. This trend however is not visible when considering aligned cosine similarity. For LINE and node2vec, there is no simple trend visible with respect to any of the two measures, their similarity scores look rather arbitrary. Overall, we see that although closeness centrality is ranked high in the factor analysis of Wang et al. [19], there are no clear signs that more central nodes have more stable embeddings.

Influence of Graph Properties. To evaluate the impact of graph properties on the stability of the embeddings, we generated synthetic graphs with varying sizes and densities. More precisely, we utilized two network models, namely Barabasi-Albert networks [2] and Watts-Strogatz [21] networks. For each model, we generate two sets of networks, in which we either fixed the network's size at $n = 8000$ nodes and varied its density, or fixed the densities at $D = 0.01$ and varied their size. The results of this analysis can be found in Figure 6, where we plot the average aligned cosine similarities over all nodes and embeddings per graph and algorithm against (a) graph size and (b) graph density. Figure 6(a) contains missing data points that result from terminating the embedding computation after a maximum of 72 hours per embedding.

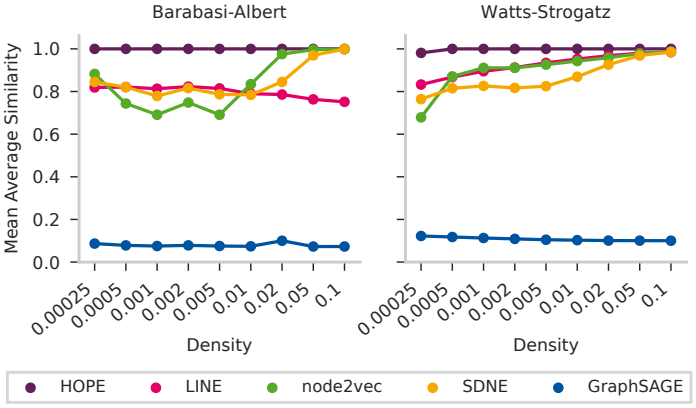
Considering the impact of network size, we see that for GraphSAGE, the already low stability rapidly drops with larger graph size on both synthetic models, whereas for HOPE, the near-perfect stability seems invariant of graph size. In between, LINE, SDNE and node2vec show similar stabilities like in our experiments on empirical graphs, however there is no consistent trend regarding the impact of network size on their stability. This finding contrasts results from Wang et al. [19], who stated that the stability of DeepWalk and node2vec primarily depends on the size of the input graphs.

For the dependence on network *density* plotted in Figure 6(b), we see that the embedding stability of SDNE and node2vec seems to increase when graphs get more dense. HOPE is once again consistent in its high stability, whereas GraphSAGE shows consistently low stability that is unaffected by network size. Finally, LINE does nor display any clear trend as it diverges between the two synthetic models.

Summary. Our results indicate clear differences in the geometric stability between the embedding algorithms, which is also in line with the results by Wang et al. [19]. HOPE consistently yields near-constant embeddings, whereas GraphSAGE was shown to be very volatile. In between, the other algorithms (LINE, node2vec, and SDNE) exhibit a moderate, but significant degree of instability. When checking possible influence factors for stability, we found for none of them a strong and general trend. In particular, we observed that the influence of node centrality, graph size, and graph density have a rather small to negligible influence on the stability of node embeddings. This does not match the high ranking of the node and graph properties in the factor analysis by Wang et al. [19]. In contrast, stability is dominated by the choice of the embedding algorithm, which overshadows the aforementioned influences.



(a) Mean average aligned cosine similarity over varying sizes.



(b) Mean average aligned cosine similarity over varying densities.

Fig. 6: *Influence of graph properties.* In (a) synthetic graphs with varying size at fixed density 0.01 and in (b) synthetic graphs with varying density and 8000 nodes are used to measure the influence of those graph properties on stability. Each data point represents the average node-wise similarity over all nodes per graph and all 435 embedding pairs resulting from 30 runs of the corresponding algorithm.

C Additional Results on Downstream Stability

In the following we present additional plots from the experiments that we conducted on downstream stability, which we left out due to space limitations in the main part.

C.1 Node Classification

We first present our results on the node classification task. Figure 7 depicts the stability of classification performance on all datasets. We observe that over all algorithms and datasets, the resulting accuracies vary only marginally, and higher variances appear to depend on the datasets rather than embedding techniques.

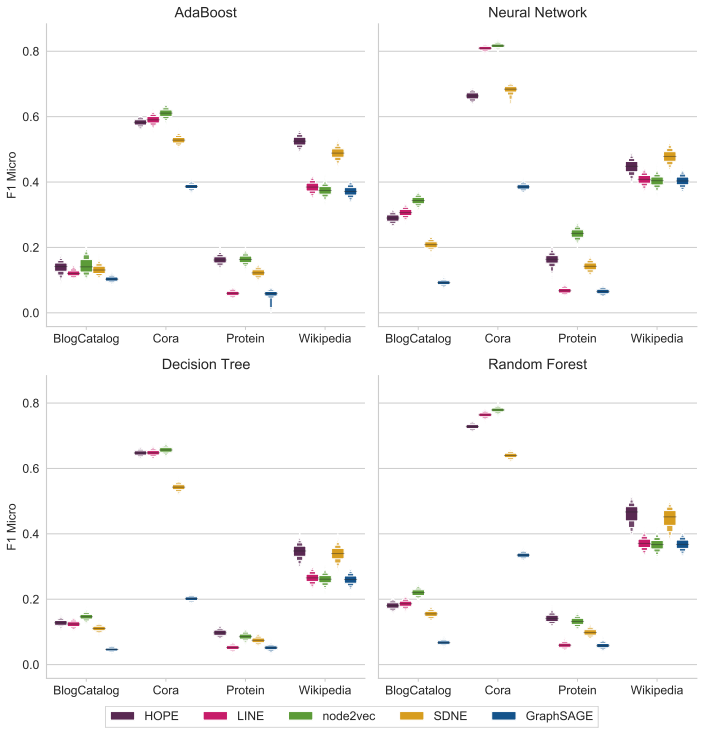


Fig. 7: *Stability of classification performance.* Stability of the micro-F1 score of the used classification methods is plotted against the used embedding algorithms. Each box corresponds to the prediction of 30 embeddings with 10 repetitions.

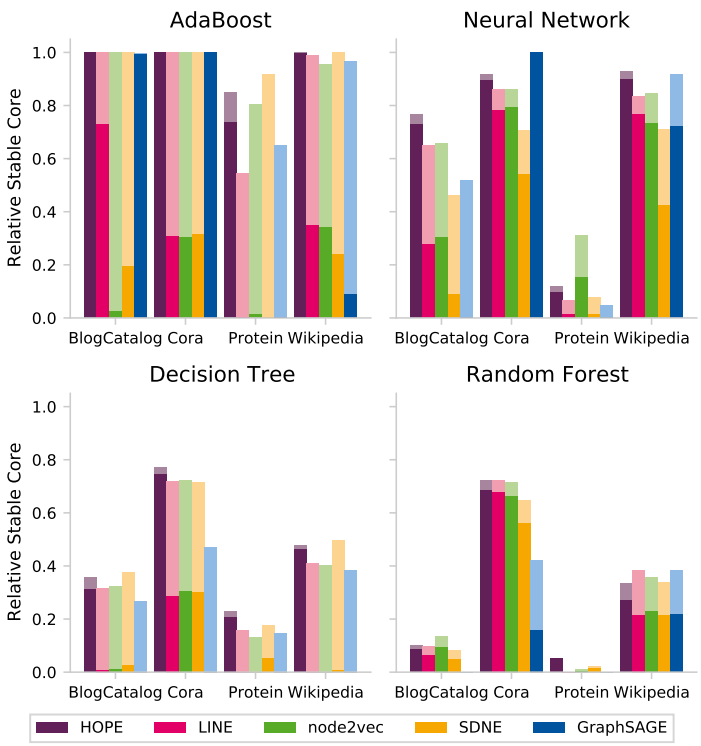


Fig. 8: *Stability in node-wise predictions.* This figure shows the stability of the classifiers as ratios of nodes which are always predicted to be in the same class. Saturated colors represent the mean stable core of all 30 embeddings and lighter colors the mean stable core of five randomly sampled embeddings with 10 repetitions each.

Our results regarding the *stability of single predictions* are shown in Figure 7. The results on Wikipedia are mostly in line with the results that were obtained on BlogCatalog and Cora and discussed in the main part. For Protein, where we have already obtained the overall lowest accuracies in node classification, we observe an overall much lower stability in individual predictions compared to the other datasets.

C.2 Link Prediction

We close with the results regarding the stability of link prediction performance on all datasets, which are shown in Figure 9. We observe that once again, the performance differences between different embeddings are negligible, except for neural networks on HOPE embeddings of the Protein network.

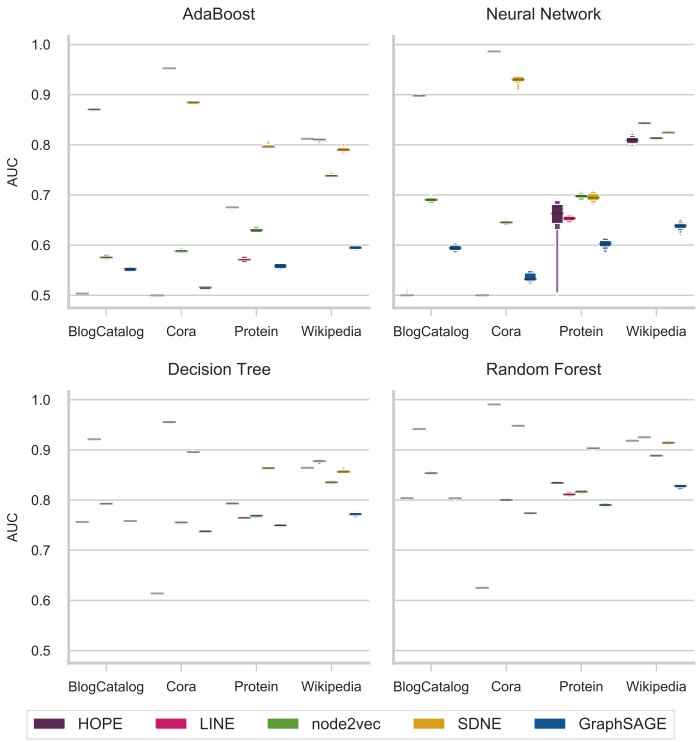


Fig. 9: *Stability of link prediction performance.* Stability of the link prediction accuracy in Area Under Curve of the used machine learning algorithms is plotted against the used embeddings algorithms. Each box corresponds to the prediction of 10 embeddings with 10 repetitions.