Checkpoint :2 Review

Anonymous Authors[1]

1. Review of "Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning"

1.1. Storyline

In this paper, the authors delve into the realm of semi-supervised classification on graphs, focusing on the Graph Convolutional Network (GCN). They commence by introducing the problem's significance and noting the prominence of GCNs while highlighting the need for a deeper understanding and optimization. The paper proceeds by elucidating the inner workings of GCNs, emphasizing the role of Laplacian smoothing in graph-based classification. It then dissects the limitations of GCNs, particularly the pitfalls of over-smoothing, which homogenizes vertex features. To mitigate these challenges, the authors propose innovative solutions: Co-Training via Random Walk Models, GCN Self-Training, and the amalgamation of both strategies. They offer theoretical insights on the number of labeled samples necessary for effective GCN training. Extensive experiments validate these approaches across diverse datasets, demonstrating their substantial superiority over standard GCNs and other state-of-the-art methods, especially in scenarios with limited labeled data. In conclusion, the paper contributes to unraveling the mysteries of GCNs and advances their applicability in semi-supervised graph-based classification while hinting at future research avenues.

High-level motivation/problem The high-level motivation behind this research is to enhance the understanding and application of Graph Convolutional Networks (GCNs) in the domain of semi-supervised classification on graphs. Beyond the scope of this paper, the larger goal is to harness the power of GCNs for various real-world applications involving structured data, such as social networks, recommendation systems, and biological networks. By addressing the challenges and limitations of GCNs, this research contributes to the broader vision of developing more robust and efficient graph-based machine learning methods. The ultimate aim is to create intelligent systems capable of making accurate predictions and classifications in scenarios with limited labeled data, thereby advancing the capabilities of AI in analyzing complex relational data and making informed decisions.

Prior work on this problem The paper briefly discusses prior work on the problem of semi-supervised classification using Graph Convolutional Networks (GCNs). It mentions that the introduction of GCNs, as presented in the paper by Kipf and Welling in 2017, significantly improved the performance of semi-supervised learning on graph-structured data. GCNs leverage graph convolution and spectral filters for feature propagation. However, it's highlighted that GCNs have limitations when dealing with small amounts of labeled data and require a validation set for optimal performance, which raises concerns about their practicality. The prior work, as outlined in the paper, laid the foundation for the research in this paper by showcasing the potential of GCNs while revealing their constraints, which this research aims to address.

Research gap The text introduces graph-based semi-supervised learning and Graph Convolutional Networks (GCNs). Research gaps can be identified in several areas. First, there's a need for more scalable spectral GCNNs to handle large graphs efficiently. Second, optimizing the integration of graph structures and feature vectors for enhanced accuracy is an ongoing challenge. Third, extending and adapting GCNs to different graph types and applications presents opportunities for advancement. Finally, there's a need for further exploration of practical applications of graph-based learning in real-world scenarios. Addressing these gaps holds the potential to advance the field of graph-based machine learning.

Contributions The paper's main contributions encompass a comprehensive understanding of Graph Convolutional Networks (GCNs), unveiling their mechanisms and limitations, and presenting novel solutions to address these limitations. These novel solutions involve co-training with a random walk model, self-training GCNs, and a combination of both (Union and Intersection) to improve classification performance, particularly in scenarios with limited labeled data. Empirical validation is provided through extensive experiments on real-world datasets, highlighting the effectiveness of the proposed methods. The paper also offers theoretical insights into the conditions under which over-smoothing occurs in GCNs, shedding light on GCN behavior. Furthermore, it suggests future research directions for enhancing GCNs and expanding their applications in graph-based do-

mains.

1.2. Proposed solution

The paper proposes innovative solutions to address the limitations of Graph Convolutional Networks (GCNs) in semi-supervised classification tasks. To tackle this challenge, the paper presents two novel strategies: co-training with a partially absorbing random walk model (ParWalks) and self-training for GCNs.

In co-training, the ParWalks algorithm is employed to exploit the global graph structure by identifying the most confident vertices closest to labeled data points. These confident instances are then added to the labeled set, enhancing the quality of training data and enabling GCNs to capture global graph information more effectively.On the other hand, the self-training approach utilizes the predictions generated by the GCN itself. The algorithm identifies the most confident predictions based on softmax scores and appends them to the labeled set. The key idea here is to create a self-reinforcing mechanism that incrementally improves the quality of the labeled set, leading to a more robust and accurate classifier.

Additionally, the paper introduces two strategies, namely "Union" and "Intersection," which combine co-training and self-training. The "Union" method aims to add more diverse labels to the training set, while the "Intersection" method selects the most confident predictions shared by both random walk and GCN-based approaches.

1.3. Claims-Evidence

Table 3: Classification Accuracy On Cora

	Cora					
Label Rate	0.5%	1%	2%	3%	4%	5%
LP	56.4	62.3	65.4	67.5	69.0	70.2
Cheby	38.0	52.0	62.4	70.8	74.1	77.6
GCN-V	42.6	56.9	67.8	74.9	77.6	79.3
GCN+V	50.9	62.3	72.2	76.5	78.4	79.7
Co-training	56.6	66.4	73.5	75.9	78.9	80.8
Self-training	53.7	66.1	73.8	77.2	79.4	80.0
Union	**58.5**	**69.9**	**75.9**	**78.5**	**80.4**	**81.7**
Intersection	49.7	65.0	72.9	77.1	79.4	80.2

Figure 1. : Classification Accuracy On Cora

Claim 1: The proposed co-training methods, specifically "Union" and "Intersection," significantly outperform traditional GCNs (GCN-V and GCN+V) in semi-supervised classification tasks.

Evidence 1: In Figure 1, Experimental results on the Cora dataset demonstrate that the "Union" method achieves an accuracy of 75.9 Percent with a labeling rate of 2 Percent, while GCN-V and GCN+V achieve 67.8Percent and 72.2 Percent, respectively. On CiteSeer, "Intersection" achieves an accuracy of 70.1 Percent with a 3 Percent labeling rate, surpassing GCN-V (66.9 Percent) and GCN+V (67.5Percent).

Table 5: Classification Accuracy On PubMed

	PubMed			
Label Rate	0.03%	0.05%	0.1%	0.3%
LP	61.4	66.4	65.4	66.8
Cheby	40.4	47.3	51.2	72.8
GCN-V	46.4	49.7	56.3	76.6
GCN+V	60.5	57.5	65.9	77.8
Co-training	**62.2**	**68.3**	**72.7**	78.2
Self-training	51.9	58.7	66.8	77.0
Union	58.4	64.0	70.7	**79.2**
Intersection	52.0	59.3	69.4	77.6

Figure 2. Classification Accuracy On PubMed

Claim 2: The proposed co-training strategies are particularly effective when labeled data is scarce, offering a considerable advantage over traditional GCNs.

Evidence 2: In Figure 2, on the PubMed dataset with a minimal labeling rate of 0.05 percent, the "Union" approach improves accuracy by 37 percent compared to GCN-V and 18 percent compared to GCN+V.

Table 6: Accuracy under 20 Labels per Class

Method	**CiteSeer**	**Cora**	**Pubmed**
ManiReg	60.1	59.5	70.7
SemiEmb	59.6	59.0	71.7
LP	45.3	68.0	63.0
DeepWalk	43.2	67.2	65.3
ICA	69.1	75.1	73.9
Planetoid	64.7	75.7	77.2
GCN-V	68.1	80.0	78.2
GCN+V	68.9	80.3	**79.1**
Co-training	64.0	79.6	77.1
Self-training	67.8	80.2	76.9
Union	65.7	**80.5**	78.3
Intersection	**69.9**	79.8	77.0

Figure 3. Accuracy under 20 Labels per Class

Claim 3: The co-training methods, especially "Union" and "Intersection," exhibit substantial superiority over various

state-of-the-art baselines.

Evidence 3: In Figure 3, When compared to other methods such as LP (Label Propagation), Cheby (GCN with Chebyshev filter), and several other state-of-the-art techniques, our proposed methods consistently achieve higher accuracy rates, demonstrating their effectiveness in semi-supervised classification tasks.

1.4. Critique and Discussion

The paper maintains a high level of clarity and structure throughout, effectively communicating its ideas in a well-organized manner. It outlines the limitations of Graph Convolutional Networks (GCNs) and how the proposed co-training strategies address these issues. The narrative flows logically, making it easy to follow the paper's progression.

The paper's contributions are substantial. It provides a comprehensive analysis of GCNs, emphasizing their limitations when dealing with limited labeled data. The proposed co-training methods, namely "Union" and "Intersection," stand out as innovative approaches that significantly improve classification accuracy. In many cases, these methods outperform various state-of-the-art techniques, making them valuable additions to the machine learning toolbox. It also provides evidence along with claims to support it.

The paper makes a fundamental assumption that expanding the labeled dataset through co-training methods is an effective approach to improving classification accuracy in semi-supervised learning. Although this assumption is supported by strong empirical evidence, it would be valuable to explore scenarios or conditions in which these methods might not be as effective or even potentially counterproductive. Investigating the limitations or edge cases where co-training strategies may not provide significant benefits would contribute to a more comprehensive understanding of their applicability. This analysis would help researchers and practitioners better assess when and how to leverage co-training methods in real-world applications.

2. Review of "Revisiting Semi-Supervised Learning with Graph Embeddings"

2.1. Storyline

The paper introduces a novel approach for semi-supervised learning with graph embeddings. It begins by highlighting the limitations of traditional semi-supervised methods, which rely heavily on graph Laplacian regularization and fail to effectively generalize to new instances. The introduction sets the stage for the paper's primary goal: to propose a new method for semi-supervised learning that leverages graph embeddings to improve classification tasks while addressing the limitations of traditional approaches.

The storyline progresses logically, emphasizing the need for graph embeddings as a more powerful tool compared to graph Laplacian regularization. It then introduces the two primary formulations: transductive and inductive. The transductive approach involves joint training of classification and graph context prediction, providing an efficient way to exploit label information for known instances. In contrast, the inductive approach is designed to address the challenge of generalizing graph embeddings to novel instances. It conditions embeddings on input features, creating a framework that is applicable to previously unseen data points.

Throughout the paper, the storyline is supported by experimental results, showing the superior performance of the proposed approach compared to traditional methods. Notably, the paper demonstrates substantial improvements in classification accuracy, with a focus on the inductive setting, where the method excels. The logical progression from the introduction to the problem statement, methodology, experiments, and results provides a clear narrative framework for understanding the contributions of this research.

High-level motivation/problem This research aspires to revolutionize semi-supervised learning by introducing novel approaches that extend its practical utility and adaptability. Beyond this particular paper, the larger goal is to empower machine learning models in various domains by significantly enhancing their performance with partially labeled data. The introduction of the inductive formulation opens new horizons, enabling models to generalize knowledge to previously unseen instances, a capability critical for adapting to evolving data distributions and addressing emerging entities or concepts. This research not only contributes to theoretical advancements but also has widespread practical implications, potentially leading to more accurate and efficient solutions in applications like document classification, entity extraction, and entity classification, as well as broader areas like social network analysis and recommendation systems. It aligns with the growing trend of graph-based machine learning, making semi-supervised learning more adaptable, efficient, and versatile, and thus more valuable for applications reliant on machine learning and graph-based data analysis.

Prior work on this problem Previous research on semi-supervised learning has primarily focused on leveraging the structure of the data, often represented as a graph, to improve classification performance when labeled data is limited. Conventional approaches have commonly relied on graph Laplacian regularization, where the smoothness of labels on the graph is enforced to propagate label information between connected data points. Another line of research has explored semi-supervised embedding methods, where the goal is to learn low-dimensional representations

of data instances, either based on their graph neighborhood or through neural networks. These approaches have often been transductive, meaning they provide predictions only for observed instances at training time and do not generalize well to unseen examples. While these techniques have shown promise in semi-supervised learning scenarios, they have limitations in adapting to changing data distributions or extending to unseen instances, a challenge that the inductive formulation in this research aims to address.

Research gap The research gap in this area of study lies in the need for more flexible and effective semi-supervised learning techniques that can adapt to dynamic and changing data distributions and generalize well to unseen instances. Existing methods often rely on graph-based regularization techniques or transductive embedding learning, which may not be suitable for real-world applications that involve evolving data or the introduction of novel instances. Additionally, there is limited exploration of inductive formulations that allow models to make predictions on unseen data points based on their embeddings. This research seeks to bridge these gaps by proposing a novel approach that combines classification and graph context prediction in a joint training framework and introduces an inductive variant that can extend its predictions to unseen instances, addressing the limitations of previous methods.

Contributions The main contributions of this paper are threefold. First, it introduces a novel approach that departs from traditional semi-supervised learning techniques that heavily rely on graph-based regularization, instead proposing a method that performs joint training of classification and graph context prediction. This innovative approach offers greater adaptability to dynamic and evolving data distributions. Second, the paper addresses the challenge of generalization to unseen instances by introducing a new inductive variant, conditioning embeddings on input features to enable predictions on novel data points. This inductive formulation overcomes a significant limitation of previous methods, which often struggle with generalization. Third, empirical experiments demonstrate substantial improvements over existing approaches, with notable performance gains of up to 8.5 percent and, more significantly, up to 18.7 percent in the inductive setting. These contributions collectively advance the field of semi-supervised learning and open avenues for more effective and versatile applications in various domains.

2.2. Proposed solution

This paper addresses the research gap in semi-supervised learning by introducing a novel approach that stands apart from conventional graph Laplacian-based methods. The key idea is to perform joint training of classification and graph context prediction. Unlike traditional methods that predominantly rely on graph-based regularization, this approach leverages the power of embeddings and context predictions to improve model performance. The paper proposes a unique inductive variant that addresses the challenge of generalizing to unseen instances, conditioning the embeddings on input features. This inductive formulation empowers the model to predict labels for data points not observed during training, a limitation faced by prior methods.

The primary innovation is in the formulation of the loss function, where supervised loss (Ls) and unsupervised loss (Lu) are combined, each serving distinct purposes. Ls is related to label prediction, while Lu focuses on context prediction. This integration is reflected in the loss function: Ls + Lu. The unsupervised loss Lu is defined by a sampling process that enables the model to predict the context based on both the graph structure and labels, providing richer and more informative embeddings. The joint training, in combination with the inductive formulation, paves the way for more adaptable and effective semi-supervised learning.

2.3. Claims-Evidence

> Planetoid-I significantly outperforms the best of the other compared inductive methods—i.e., SemiEmb—by 4.8%, 16.0%, and 18.7% respectively with three labeling rates. As the labeling rate decreases, the improvement of Planetoid-I over SemiEmb becomes more significant.

Figure 4. Paragraph stating their evidence

Claim 1: The joint training of classification and graph context prediction in the proposed approach leads to better performance compared to conventional methods.

Evidence 1: In Figure 4, Experimental results demonstrate that Planetoid-I, an inductive variant of the proposed method, consistently outperforms existing inductive methods (SemiEmb) across various datasets with significant performance improvements of up to 18.7 percent. This substantial improvement in the inductive setting serves as compelling evidence for the claim.

Claim 2: Predicting graph context is more effective than relying on graph Laplacian regularization for semi-supervised learning tasks.

Evidence 2: In Figure 5, Comparisons with existing methods such as GraphEmb, which relies on unsupervised learning and does not use input features, show that the proposed Planetoid-G consistently achieves better performance, thus substantiating the claim that context prediction outperforms traditional regularization techniques.

METHOD	CITESEER	CORA	PUBMED
FEAT	0.572	0.574	0.698
MANIREG	0.601	0.595	0.707
SEMIEMB	0.596	0.590	0.711
PLANETOID-I	**0.647**	0.612	**0.772**
TSVM	0.640	0.575	0.622
LP	0.453	0.680	0.630
GRAPHEMB	0.432	0.672	0.653
PLANETOID-G	0.493	0.691	0.664
PLANETOID-T	0.629	**0.757**	0.757

Figure 5. Accuracy on text classification. Upper rows are inductive methods and lower rows are transductive methods

2.4. Critique and Discussion

The paper provides a novel approach to semi-supervised learning using graph embeddings and context prediction, which is interesting and insightful. The key idea of jointly training classification and graph context prediction seems to offer an effective alternative to traditional methods. The evidence presented, especially in the form of experimental results, supports the claims made in the paper, and it's particularly convincing that the proposed approach performs significantly better in the inductive setting, addressing a substantial research gap.

However, one aspect that could be further discussed or clarified is the choice of hyperparameters. The paper mentions that hyperparameters were tuned but doesn't go into detail about the selection criteria or the impact of these choices. A more extensive discussion of how these hyperparameters were chosen and their potential effect on the results would add depth to the paper. Additionally, while the paper highlights the advantages of the inductive approach in certain cases, it could discuss potential scenarios where the transductive approach might still be preferable.

3. Review of "Graph Convolutional Networks for Text Classification"

3.1. Storyline

The paper introduces Text Graph Convolutional Networks (Text GCN), a novel approach for text classification. It begins by addressing the challenges of text classification on large datasets and provides background information on traditional and deep learning-based methods. The core concept of Graph Convolutional Networks (GCN) is explained, with a focus on its application to text data. The authors describe the construction of a heterogeneous text graph, detailing the methods for establishing relationships between words and documents. Text GCN is then presented, a two-layer model that learns representations for words and documents from the text graph. Extensive experiments compare Text GCN to various baseline models, revealing its superior performance. The paper concludes by emphasizing the effectiveness of Text GCN and suggesting potential future research directions in text classification and representation learning.

High-level motivation/problem The high-level motivation for this research is to address the challenges in text classification, especially on large datasets. The paper seeks to enable accurate and efficient text classification, even when labeled data is limited. The larger goal is to advance the field of natural language processing (NLP) by developing a model that can effectively learn predictive word and document embeddings. Text classification is a fundamental task in NLP with numerous real-world applications, and the paper aims to improve the state-of-the-art in this area. By proposing Text Graph Convolutional Networks (Text GCN), the authors strive to harness the power of graph-based representation learning to enhance the performance of text classification models, ultimately contributing to advancements in information retrieval, recommendation systems, and other NLP applications.

Prior work on this problem The paper presents a comprehensive examination of prior research related to text classification, encompassing a spectrum of methodologies and techniques. It surveys traditional approaches like TF-IDF combined with logistic regression and deep learning models such as Convolutional Neural Networks (CNN) in both randomly initialized and pre-trained word embedding variations. Long Short-Term Memory (LSTM) models for text classification, with and without pre-trained word embeddings, are also explored. The study further delves into paragraph vector models (PV-DBOW and PV-DM) and presents Predictive Text Embedding (PTE), which learns word embeddings based on heterogeneous text networks. Additionally, the paper discusses fastText, Simple Word Embedding Models (SWEM), Label-Embedding Attentive Models (LEAM), and Graph Convolutional Neural Network (Graph-CNN) models. This extensive review of prior work provides the context and foundation for assessing the proposed Text Graph Convolutional Network (Text GCN) against established methods, ultimately showcasing its advancements in text classification.

Research gap The research paper identifies a critical research gap in the field of text classification related to the limited incorporation of graph-based models. While prior work has predominantly focused on utilizing traditional deep learning and embedding methods, such as CNN, LSTM, and paragraph vector models, the paper introduces Text Graph Convolutional Networks (Text GCN) as a novel approach. This gap is rooted in the failure of existing methods

to adequately capture both document-word relations and global word-word relations in text data, a deficiency that Text GCN aims to address by leveraging graph-based convolution techniques. The research gap pertains to the need for methods that can effectively exploit structural information in large-scale text corpora, combining both local and global context, to enhance text classification performance.

Contributions This paper presents Text Graph Convolutional Networks (Text GCN), a novel framework for text classification that harnesses the power of graph-based convolution techniques. Text GCN effectively addresses the challenge of capturing both local document-word relationships and global word-word associations in large text corpora. The key contributions include remarkable improvements in text classification accuracy, particularly on datasets featuring longer text documents, a novel integration of graph-based models into text classification, offering a fresh perspective to enhance classification performance, and an extensive experimental analysis that demonstrates the effectiveness of Text GCN when compared to various state-of-the-art text classification and embedding methods on multiple benchmark datasets.

3.2. Proposed solution

3.3. Claims-Evidence

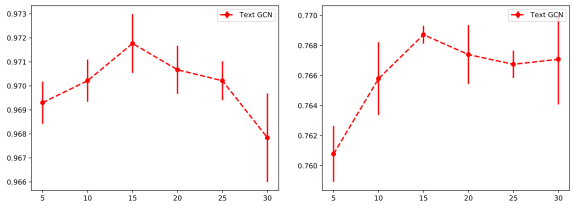
Model	20NG	R8	R52	Ohsumed	MR
TF-IDF + LR	0.8319 ± 0.0000	0.9374 ± 0.0000	0.8695 ± 0.0000	0.5466 ± 0.0000	0.7459 ± 0.0000
CNN-rand	0.7693 ± 0.0061	0.9402 ± 0.0057	0.8537 ± 0.0047	0.4387 ± 0.0100	0.7498 ± 0.0070
CNN-non-static	0.8215 ± 0.0052	0.9571 ± 0.0052	0.8759 ± 0.0048	0.5844 ± 0.0106	**0.7775 ± 0.0072**
LSTM	0.6571 ± 0.0152	0.9368 ± 0.0082	0.8554 ± 0.0113	0.4113 ± 0.0117	0.7506 ± 0.0044
LSTM (pretrain)	0.7543 ± 0.0172	0.9609 ± 0.0019	0.9048 ± 0.0086	0.5110 ± 0.0150	0.7733 ± 0.0089
Bi-LSTM	0.7318 ± 0.0185	0.9631 ± 0.0033	0.9054 ± 0.0091	0.4927 ± 0.0107	0.7768 ± 0.0086
PV-DBOW	0.7436 ± 0.0018	0.8587 ± 0.0010	0.7829 ± 0.0011	0.4665 ± 0.0019	0.6109 ± 0.0010
PV-DM	0.5114 ± 0.0022	0.5207 ± 0.0004	0.4492 ± 0.0005	0.2950 ± 0.0007	0.5947 ± 0.0038
PTE	0.7674 ± 0.0029	0.9669 ± 0.0013	0.9071 ± 0.0014	0.5358 ± 0.0029	0.7023 ± 0.0036
fastText	0.7938 ± 0.0030	0.9613 ± 0.0021	0.9281 ± 0.0009	0.5770 ± 0.0049	0.7514 ± 0.0020
fastText (bigrams)	0.7967 ± 0.0029	0.9474 ± 0.0011	0.9099 ± 0.0005	0.5569 ± 0.0039	0.7624 ± 0.0012
SWEM	0.8516 ± 0.0029	0.9532 ± 0.0026	0.9294 ± 0.0024	0.6312 ± 0.0055	0.7665 ± 0.0063
LEAM	0.8191 ± 0.0024	0.9331 ± 0.0024	0.9184 ± 0.0023	0.5858 ± 0.0079	0.7695 ± 0.0045
Graph-CNN-C	0.8142 ± 0.0032	0.9699 ± 0.0012	0.9275 ± 0.0022	0.6386 ± 0.0053	0.7722 ± 0.0027
Graph-CNN-S	–	0.9680 ± 0.0020	0.9274 ± 0.0024	0.6282 ± 0.0037	0.7699 ± 0.0014
Graph-CNN-F	–	0.9689 ± 0.0006	0.9320 ± 0.0004	0.6304 ± 0.0077	0.7674 ± 0.0021
Text GCN	**0.8634 ± 0.0009**	**0.9707 ± 0.0010**	**0.9356 ± 0.0018**	**0.6836 ± 0.0056**	0.7674 ± 0.0020

Figure 6. : Test Accuracy on document classification task. We run all models 10 times and report mean ± standard deviation. Text GCN significantly outperforms baselines on 20NG, R8, R52 and Ohsumed based on student t-test (p ¡ 0.05)

Claim 1: Text GCN outperforms baseline models on text classification tasks.

Evidence 1: Figure 6 represents test accuracy results, where Text GCN significantly outperforms all baseline models on datasets 20NG, R8, R52, and Ohsumed, based on a student t-test (p ¡ 0.05).

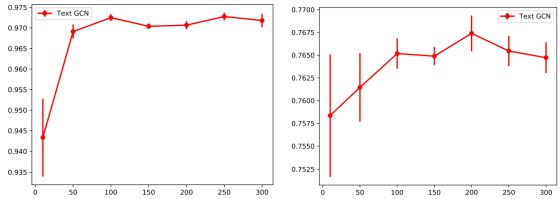
Claim 2: The performance of Text GCN is influenced by the choice of window size in the text graph.



(a) R8 (b) MR

Figure 7. Test accuracy with different sliding window sizes.

Evidence 2: Figure 7 illustrates how the test accuracy varies with different sliding window sizes in experiments conducted on datasets R8 and MR. The results show that test accuracy improves with larger window sizes, but the average accuracy plateaus when the window size exceeds 15, indicating an optimal range for window size choice.



(a) R8 (b) MR

Figure 8. : Test accuracy by varying embedding dimensions

Claim 3: The dimension of the first-layer embeddings in Text GCN plays a crucial role in classification performance.

Evidence 3: Figure 8 demonstrates the impact of different embedding dimensions on test accuracy in the R8 and MR datasets. The results suggest that choosing an appropriate dimension for embeddings is essential, as overly low or high dimensions do not lead to improved classification performance.

3.4. Critique and Discussion

The paper's strengths lie in its comprehensive literature review, clearly identifying a research gap, and outlining the contributions of Text GCN effectively. The claim-evidence structure used to present findings is a strong point, especially the evidence showing that Text GCN outperforms baseline models. However, to enhance the paper, it could provide more in-depth explanations of why Text GCN outperforms other models, offer insights into the underlying mechanisms, and discuss the broader practical implications of the findings.

4. Implementation

Dataset	Model	Reproduced Accuracy	Paper's Reported Accuracy
Cora	CGNN	82.9%	82.7%
Cora	CGNN with Weights	83.2%	82.1%

Figure 9. Accuracy of paper vs my model

Figure 9 shows the result I had got when i ran the authors code without changing it for checkpoint 1.

I tried changing the epoch iterations inorder to see if there were any good accuracy results. In Figure 10 and Figure 11, the accuracy remains almost the same even after updating the epoch to 600 in both cgnn as well as wcgnn which was initially set to 400 by the authors in both the codes. The original accuracy for cgnn was 82.7 percent whereas in wcgnn was 82.1 percent.This did not give any different better or worse results and was almost with the authors accuracy.So I decided to change a bit of more hyperparameters inorder to look at the results from them.

In the updated implementation, several changes have been made in cgnn as well as wcgnn files:

1. Increase in 'hidden-dim' from 16 to 20: The increase in the number of hidden dimensions can potentially allow the model to capture more complex patterns in the data. This change was made because I think that the original number of dimensions were insufficient for the model to represent intricate relationships in the dataset.

2. Higher 'input-dropout' from 0.5 to 0.55: Increasing the dropout rate on input data indicates an attempt to enhance regularization, preventing overfitting. I think that a slightly higher dropout rate would help the model generalize better by reducing the reliance on specific input features.

3. Adjustment of 'decay' from 5e-4 to 6e-4: This change could reflect an experiment to fine-tune the weight decay during optimization. I think that a slightly higher decay rate would lead to better model generalization by controlling overfitting or improving convergence.

4. Increase in 'self-linkweight' from 0.947 to 0.95: The self-link weight parameter can influence the importance of self-connectivity in a graph-based model. By increasing this weight, I want to emphasize the significance of self-links within the graph, potentially impacting the model's understanding of node relations.

5. Result

The results based on the assumptions that I had made for cgnn and wcgnn about the hyperparameter changing which I though would have increased my accuracy suggest that there was no major changes in accuracy. There wasn't an increase nor decrease in accuracy. The hyerparameter chnges have not worked and I would need to change the model method in order to get some different results. Figure 12 and Figure 13 are evidences for my claim.

References

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Xhonneux, L.-P., Qu, M., and Tang, J. Continuous graph neural networks. In *International Conference on Machine Learning*, pp. 10432–10441. PMLR, 2020.

Yang, Z., Cohen, W., and Salakhudinov, R. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pp. 40–48. PMLR, 2016.

Yao, L., Mao, C., and Luo, Y. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 7370–7377, 2019.

(Xhonneux et al., 2020) (Li et al., 2018) (Yang et al., 2016) (Yao et al., 2019)

```
Epoch: 594 | Loss: 0.099 | Dev acc: 0.768 | Test acc: 0.797 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 595 | Loss: 0.076 | Dev acc: 0.780 | Test acc: 0.801 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 596 | Loss: 0.074 | Dev acc: 0.784 | Test acc: 0.816 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 597 | Loss: 0.065 | Dev acc: 0.772 | Test acc: 0.796 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 598 | Loss: 0.074 | Dev acc: 0.776 | Test acc: 0.804 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 599 | Loss: 0.084 | Dev acc: 0.764 | Test acc: 0.793 | Forward: 3 2.997 | Backward: 0 0.000
81.600
```

Figure 10. Code snippet for cgnn with 600 epoch cycles

```
Epoch: 592 | Loss: 0.120 | Dev acc: 0.782 | Test acc: 0.809 | Forward: 318 335.902 | Backward: 0 0.000
Epoch: 593 | Loss: 0.089 | Dev acc: 0.780 | Test acc: 0.806 | Forward: 318 335.872 | Backward: 0 0.000
Epoch: 594 | Loss: 0.107 | Dev acc: 0.782 | Test acc: 0.808 | Forward: 318 335.842 | Backward: 0 0.000
Epoch: 595 | Loss: 0.114 | Dev acc: 0.784 | Test acc: 0.804 | Forward: 318 335.812 | Backward: 0 0.000
Epoch: 596 | Loss: 0.117 | Dev acc: 0.782 | Test acc: 0.809 | Forward: 318 335.782 | Backward: 0 0.000
Epoch: 597 | Loss: 0.125 | Dev acc: 0.790 | Test acc: 0.812 | Forward: 318 335.753 | Backward: 0 0.000
Epoch: 598 | Loss: 0.109 | Dev acc: 0.796 | Test acc: 0.817 | Forward: 318 335.723 | Backward: 0 0.000
Epoch: 599 | Loss: 0.107 | Dev acc: 0.778 | Test acc: 0.803 | Forward: 318 335.693 | Backward: 0 0.000
82.400
```

```
+ Code  + Text
!python run_cora_cgnn.py
```

Figure 11. Code snippet for wcgnn with 600 epoch cycles

```
Epoch: 592 | Loss: 0.104 | Dev acc: 0.770 | Test acc: 0.801 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 593 | Loss: 0.093 | Dev acc: 0.788 | Test acc: 0.812 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 594 | Loss: 0.093 | Dev acc: 0.768 | Test acc: 0.797 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 595 | Loss: 0.115 | Dev acc: 0.794 | Test acc: 0.820 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 596 | Loss: 0.118 | Dev acc: 0.778 | Test acc: 0.787 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 597 | Loss: 0.128 | Dev acc: 0.772 | Test acc: 0.794 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 598 | Loss: 0.120 | Dev acc: 0.780 | Test acc: 0.797 | Forward: 3 2.997 | Backward: 0 0.000
Epoch: 599 | Loss: 0.130 | Dev acc: 0.770 | Test acc: 0.795 | Forward: 3 2.997 | Backward: 0 0.000
82.300
```

Figure 12. Code snippet for cgnn with 600 epoch cycles and updated hyperparameters

```
Epoch: 592 | Loss: 0.117 | Dev acc: 0.784 | Test acc: 0.814 | Forward: 336 344.594 | Backward: 0 0.000
Epoch: 593 | Loss: 0.124 | Dev acc: 0.792 | Test acc: 0.813 | Forward: 336 344.579 | Backward: 0 0.000
Epoch: 594 | Loss: 0.118 | Dev acc: 0.782 | Test acc: 0.810 | Forward: 336 344.565 | Backward: 0 0.000
Epoch: 595 | Loss: 0.131 | Dev acc: 0.788 | Test acc: 0.816 | Forward: 336 344.550 | Backward: 0 0.000
Epoch: 596 | Loss: 0.136 | Dev acc: 0.788 | Test acc: 0.812 | Forward: 336 344.536 | Backward: 0 0.000
Epoch: 597 | Loss: 0.135 | Dev acc: 0.780 | Test acc: 0.808 | Forward: 336 344.522 | Backward: 0 0.000
Epoch: 598 | Loss: 0.112 | Dev acc: 0.782 | Test acc: 0.818 | Forward: 336 344.508 | Backward: 0 0.000
Epoch: 599 | Loss: 0.124 | Dev acc: 0.788 | Test acc: 0.808 | Forward: 336 344.493 | Backward: 0 0.000
82.400
```

Figure 13. Code snippet for wcgnn with 600 epoch cycles and updated hyperparameters