CPSC 583 Project Proposal
Evaluating Strategies for Incorporating Node Embeddings within GraphSAGE for Enhanced Binary Classification in Social Networks

Shurui Wang
shurui.wang@yale.edu

Weiyi You
weiyi.you@yale.edu

1 Introduction

Our project probes the effectiveness of various attributed node level embedding methods within Graph Neural Networks (GNNs) for binary classification on social network data, focusing on different embedding incorporation strategies: before, after, between, and parallel. Given the notable impact of node embedding and its incorporation on the performance of GNNs, understanding which strategy optimally harnesses the representation power of embeddings is pivotal. The study seeks to methodically evaluate and benchmark distinct embedding methods under varied incorporation contexts, employing metrics such as accuracy, to discern the optimal embedding-incorporation paradigm that enhances binary classification performance on social networks, providing valuable insights and guidelines for future research and application in graph-based learning tasks. By experimenting with 3 embedding methods with GraphSAGE in two databases, the results show that in general, all strategies could improve the performance of GraphSAGE in some extant and before incorporation strategy with FeatherNode embedding performs best in all combinations. The results show that with a carefully considered choice of embedding method and incorporation strategy based on the specific dataset characteristics, our embedding-incorporation model would provide notable improvement on the expressive power of GNNs.

2 Related Work

Our approach draws conceptual parallels with established practices in node embedding techniques and the GraphSAGE framework, alongside recent explorations into melding both deep and shallow node embedding methods with GNNs[6].

GraphSAGE GraphSAGE, Graph Sample and Aggregated Embeddings, is a method for learning continuous vector representations (embeddings) of nodes in a graph, which allows capturing structural and contextual information about each node[1]. The core concept of GraphSAGE involves inductive learning, enabling it to generate embeddings for nodes it hasn't seen during training. This is achieved by iterative sampling and aggregating information from a node's neighborhood, which helps capture both local and global graph properties.

While GraphSAGE offers several advantages in binary classification tasks on social network data, it also presents certain limitations. A primary advantage of GraphSAGE is its inductive learning capability, which allows it to generalize to unseen nodes, making it highly effective for dynamic social networks where new nodes frequently appear [1]. This adaptability is further enhanced by its ability to leverage node feature information, enabling nuanced and context-rich embeddings.

However, GraphSAGE's reliance on neighborhood sampling can be a double-edged sword. While this approach enables scalability to large networks, it may also lead to information loss, especially in sparsely connected regions of the graph [2]. This can result in sub-optimal embeddings for nodes with few connections or in networks with highly irregular structures. Furthermore, the iterative aggregation

process, although powerful, might lead to oversmoothing, where node embeddings in different classes become indistinguishable, reducing classification performance [2]. Thus, while GraphSAGE presents a robust framework for binary classification on social networks, its effectiveness can be contingent on the specific structural characteristics of the network in question.

Node Embedding methods: Methods like FEATHER-N [5], BANE [7], and ASNE [3] have emerged as pivotal tools for encapsulating structural and, at times, feature information of nodes within graph-structured data into dense vector representations. FEATHER-N offers an attributed node embedding, formulating representations based on both structural and nodal attribute information, aiming to intertwine these two informational realms to generate a more comprehensive node portrayal. However, it may be potentially less effective in attribute-sparse scenarios [4]. BANE introduces a novel approach by binarizing the embedding space. It efficiently captures both the network structure and node attributes while significantly reducing the dimensionality of the embeddings, which enhances computational efficiency and preserves essential information for network analysis, yet this might lead to some loss of information due to reduced dimensionality. ASNE (Attributed Social Network Embedding) adopts a different strategy, aiming to preserve both network structure and node attribute proximity in the embedding space, thus effectively representing social networks for learning tasks. The different methods and techniques used in these approaches provide a varied perspective on node embedding generation and offer opportunities to explore their potential and effectiveness within various incorporation strategies in GNNs, especially GraphSAGE, for binary classification tasks on social network data.

Deep Embeddings in GNNs: The work by Sola et al. [6] delves into the integration of deep embeddings within Graph Neural Networks to enhance domain-independent predictions. This approach underscores the importance of leveraging contextual information in GNNs. An advantage of this method is its robustness in various domains due to the depth of embeddings, which capture complex patterns. However, the potential downside includes increased computational complexity and the risk of overfitting in scenarios with limited training data or overly complex graph structures.

3 Methods

GraphSAGE [1], known for its distinctive approach in aggregating neighborhood information, is well-suited for large-scale graphs. In our methodology, we focus on integrating a pre-calculated global node embedding into the GraphSAGE network, experimenting with placing it in different positions within the architecture. This modification enables the formation of a novel GraphSAGE architecture, tailored for a comparative study of four unique node embedding incorporation strategies. The subsequent sections will not only include a dedicated section on the pre-calculated global node embeddings, but will also elaborate on each GraphSAGE incorporation strategy, detailing their placement within the network. The schematic diagram of the incorporation methods is shown in Figure 1.

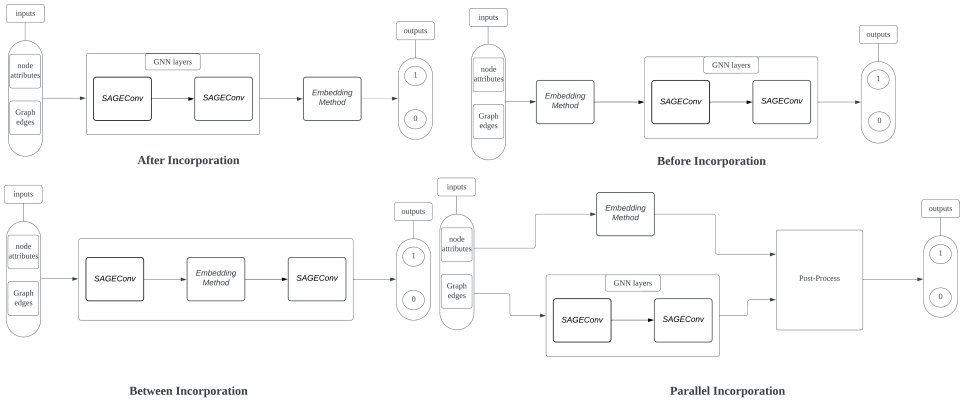

Figure 1: Schematic Diagram of 4 Incorporation Methods

3.1 Pre-Calculation of Global Node Embedding

In our study, three distinct global node embedding methods were pre-calculated and used consistently across all incorporation strategies in subsequent analyses. These methods include FeatherNode, ASNE, and BANE, each selected for their unique capabilities in node representation. FeatherNode focused on dimension reduction and spectral feature extraction, ASNE on capturing the network structure and node attributes, and BANE for its binarized embedding technique. The application of these diverse methods ensures a comprehensive evaluation of node embeddings across various GraphSAGE incorporation strategies, tailored to the specific requirements of our datasets.

3.2 After Incorporation–Weiyi You

After Incorporation involves appending the pre-calculated global node embeddings to the output of the final GraphSAGE convolutional layer using a concatenation operation. This concatenated output is then passed through a linear transformation (fully connected layer). This method not only enriches the final node representations with external information but also allows for an additional transformation that combines the strengths of GraphSAGE-learned embeddings with the global embeddings, potentially enhancing classification performance.

3.3 Before Incorporation–Shurui Wang

Before Incorporation involves merging the pre-calculated global node embeddings with the initial node features before they are input to the GraphSAGE network. Specifically, this concatenation occurs right at the start of the process, as the combined features are then passed through the first GraphSAGE convolutional layer. This method allows the network to leverage comprehensive node representations from the outset, potentially enhancing the learning efficiency and accuracy by utilizing both local and global node information from the beginning.

3.4 Between Incorporation–Weiyi You

Between Incorporation involves integrating pre-calculated global node embeddings directly after the first GraphSAGE convolutional layer. This is achieved by concatenating these global embeddings with the output of the first GraphSAGE layer. The concatenated output is then processed through a second GraphSAGE convolutional layer. This approach allows the GraphSAGE network to leverage both the local features learned in the first layer and the global context provided by the external embeddings, potentially enriching the node representations and enhancing the overall learning process.

3.5 Parallel Incorporation–Shurui Wang

Parallel Incorporation simultaneously operates the GraphSAGE model and processes pre-calculated global node embeddings through a separate linear transformation. The outputs of both the GraphSAGE layers and the linearly transformed global embeddings are then combined through an addition operation. This method capitalizes on the local learning strengths of GraphSAGE while simultaneously incorporating the broader context provided by the global embeddings, aiming for a synthesis that enriches node representation through a blend of both local and global perspectives.

4 Experiments

4.1 Dataset

In the quest to evaluate the performance of our Graph Neural Network (GNN) models, we strategically employed two datasets, comprising both publicly available and autonomously collected social network and web graph data, housed within the *torch_geometric.datasets* repository. Although each dataset is homogeneous within its own context, they present inherent heterogeneity when juxtaposed, exhibiting variances in critical parameters such as size, density, and feature quantity. Furthermore, they have been pre-labeled, rendering them aptly suited for binary node classification. A comprehensive summary of the descriptive statistics related to these datasets is meticulously tabulated in Table 1.

In our experiment, the two datasets were split into training, validation, and test sets using a ratio of 60%, 20%, and 20%, respectively. The split was performed on the entirety of the dataset, ensuring a representative distribution across these sets. A random seed of 100 was set using PyTorch's random number generator to ensure the reproducibility of the dataset partition. This approach guarantees

consistent and replicable conditions for training and evaluating models, crucial for the integrity of experimental results.

Dataset	Nodes	Edges	Density	Clustering Coefficient	Diameter	Node Feature	Classes
Deezer EUR	28,281	185,504	0.000463	0.141	21	128	2
Twitch DE	9,498	315,774	0.007	0.2008	7	128	2

Table 1: Descriptive Statistics of the Datasets

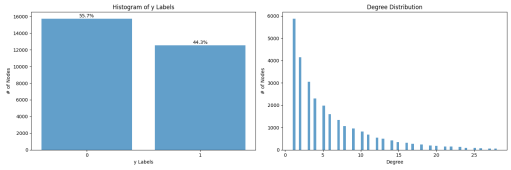

Figure 2: Twitch DE Dataset Visualization

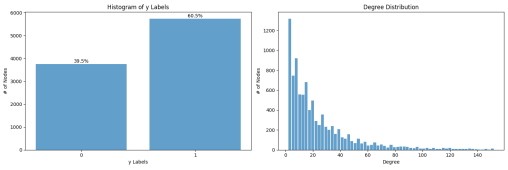

Figure 3: Deezer Europe Dataset Visualization

4.1.1 Deezer Europe Dataset[5]

The Deezer Europe Dataset, a user-user network from Deezer's European members (data from March 2020), consists of nodes representing users and links indicating mutual friendships. Node features include preferred artists. The binary classification task is to predict users' gender. Despite not being perfectly balanced, label distribution in Figure 2 suggests a reasonable balance, making accuracy a viable metric. Most users have 1 to 5 mutual friendships, indicating a low-density network.

4.1.2 Twitch Social Networks Dataset[4]

The Twitch Social Networks Dataset, derived from Twitch, features user-user networks with nodes as Twitch users and links as mutual friendships. Node features encompass liked games and streaming habits, with the binary classification task focusing on identifying users streaming mature content. We focus on Twitch DE, the largest of 6 regional datasets. Figure 3 shows a mild imbalance, with the minority label 0 at 40%. Most users have 1 to 30 mutual friendships, suggesting stronger connections than in the Deezer Europe Dataset, despite Twitch's dataset being roughly $\frac{1}{3}$ the size of Deezer's.

4.2 Baseline Model-GraphSAGE only

In our study, we established a baseline model using only the GraphSAGE architecture, without integrating any global node embedding methods. This baseline model consists of two GraphSAGE convolution layers followed by ReLU activation functions. Its performance was evaluated on two datasets: TwitchDE and DeezerEurope. After 5 training runs, each with 100 epochs, the baseline model achieved a mean test accuracy of 0.539 and a standard error of 0.021 for TwitchDE, and a mean test accuracy of 0.505 with a standard error of 0.016 for DeezerEurope. Figures 4 and 5 show the plots of accuracy over epochs for each run and dataset.

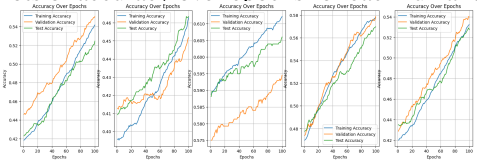

Figure 4: Base Model Accuracy over Epochs Plots for TwitchDE

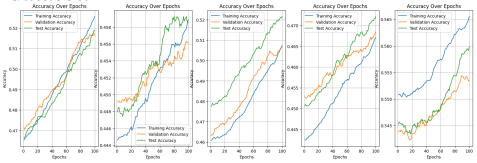

Figure 5: Base Model Accuracy over Epochs Plots for DeezerEurope

4.3 Global Node Embedding Parameters

In our experiments, global node embeddings were computed using FeatherNode, ASNE, and BANE, with specific attention to their dimensional settings. FeatherNode was configured with 16 dimensions for reduction, diverging from its default of 64, to balance dimensionality and feature representation. ASNE, typically set to 128 dimensions, was adjusted for our analysis. BANE was utilized with 16 dimensions, focusing on compact and efficient binary embeddings. These dimensionally-tuned

embeddings, derived from methods with distinct characteristics, were consistently utilized across all four GraphSAGE incorporation strategies. Details on parameters used in each embedding method are illustrated in Table 2.

Method	Parameter	Description
FeatherNode	Reduction dimensions: 16	SVD reduction dimensions
	SVD iterations: 20	Iterations for SVD computation
	Theta max: 2.5	Max evaluation point in characteristic function
	Eval points: 25	Number of evaluation points
	Order: 5	Scale of adjacency matrix powers
	Seed: 42	Random seed value
ASNE	Dimensions: 128	Dimensionality of embedding
	Workers: 4	Number of processing cores
	Epochs: 100	Training epochs
	Down sampling: 0.0001	Frequency of down sampling
	Learning rate: 0.05	Learning rate for training
	Min count: 1	Minimum count of node occurrences
	Seed: 42	Random seed value
BANE	Dimensions: 16	Size of embedding space
	SVD iterations: 20	Iterations for SVD computation
	Alpha: 0.3	Kernel matrix inversion parameter
	Iterations: 100	Matrix decomposition iterations
	Binarization iterations: 20	Iterations for binarization process
	Seed: 42	Random seed value

Table 2: Hyperparameters Used for Embedding Methods

4.4 After Incorporation–Weiyi You

The After Incorporation embedding-GraphSAGE model was trained in five runs with 100 epochs for each run. The model test 3 embedding methods for after incorporation in 2 datasets. With a 64-channel hidden layer, Cross Entropy loss function, Adam optimizer, and learning rate equal to 2×10^{-5}, the accuracy result of each run along the training epochs is visually represented in the below figures 6 7 8 9 10 11.

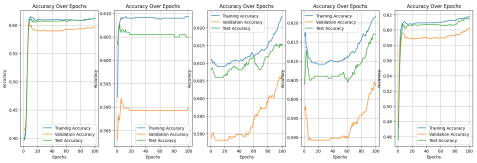

Figure 6: After Incorporation Model Accuracy over Epochs Plots for TwitchDE with FeatherNode

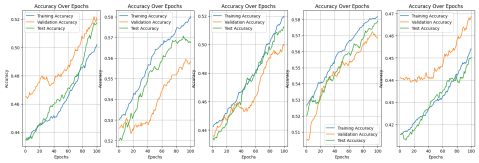

Figure 7: After Incorporation Model Accuracy over Epochs Plots for TwitchDE with ASNE

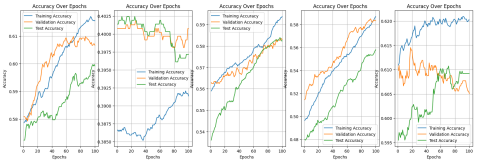

Figure 8: After Incorporation Model Accuracy over Epochs Plots for TwitchDE with BANE

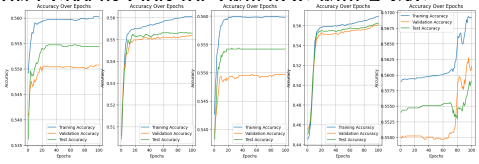

Figure 9: After Incorporation Model Accuracy over Epochs Plots for DeezerEurope with FeatherNode

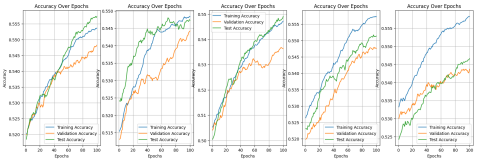

Figure 10: After Incorporation Model Accuracy over Epochs Plots for DeezerEurope with ASNE

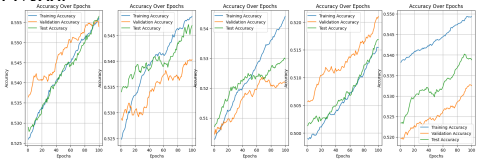

Figure 11: After Incorporation Model Accuracy over Epochs Plots for DeezerEurope with BANE

Embedding	TwitchDE		Deezer Europe	
	Results	Baseline Comparison	Results	Baseline Comparison
FeatherNode	61.12% ± 0.25%	+7.26%	55.63% ± 0.15%	+5.08%
ASNE	52.49% ± 2.02%	-1.37%	54.98% ± 0.19%	+4.43%
BANE	54.68% ± 3.32%	+0.82%	53.62% ± 0.62%	+3.07%
Baseline	53.86% ± 2.14%		50.55% ± 1.60%	

Table 3: After Incorporation Results and Baseline Comparison

In Table 3, we present the mean test accuracy and standard error over 5 runs for the After Incorporation Model. FeatherNode Embedding notably improves baseline models, showing increases of 7.26% and 5.09% for TwitchDE and Deezer Europe datasets, respectively. However, ASNE Embedding slightly declines by 1.37% in the TwitchDE dataset and improves by 4.44% in Deezer Europe. BANE Embedding performs similarly to the baseline, with a small 3.07% improvement. Apart from BANE and ASNE Embedding in the TwitchDE dataset, other results show relatively small standard errors.

In summary, After Incorporation excels when incorporating FeatherNode Embedding, revealing significant enhancements with small standard errors in both datasets. Conversely, BANE and ASNE embeddings do not exhibit clear improvements, especially in the TwitchDE dataset, where they have high standard errors.

4.5 Before Incorporation–Shurui Wang

The BeforeGraphSAGE model was consistently trained in five runs, each with 100 epochs, using a 64-channel hidden layer. The Adam optimizer, with a learning rate of 2×10^{-5}, was used alongside the CrossEntropyLoss function. Class numbers matched the dataset's requirements. Each run's accuracy progression is visually represented in the related figures below12 13 14 15 16 17.

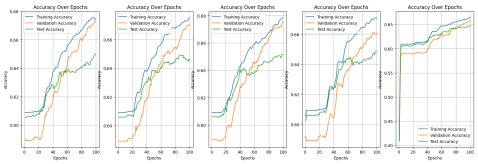

Figure 12: Before Incorporation Model Accuracy over Epochs Plots for TwitchDE with FeatherNode

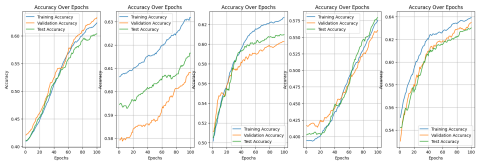

Figure 13: Before Incorporation Model Accuracy over Epochs Plots for TwitchDE with ASNE

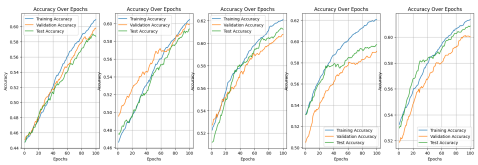

Figure 14: Before Incorporation Model Accuracy over Epochs Plots for TwitchDE with BANE

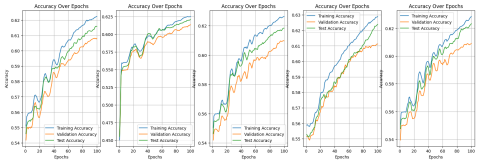

Figure 15: Before Incorporation Model Accuracy over Epochs Plots for DeezerEurope with FeatherNode

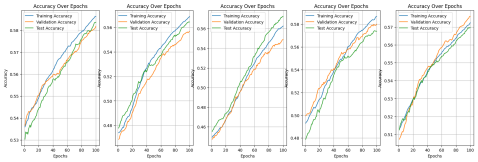

Figure 16: Before Incorporation Model Accuracy over Epochs Plots for DeezerEurope with ASNE

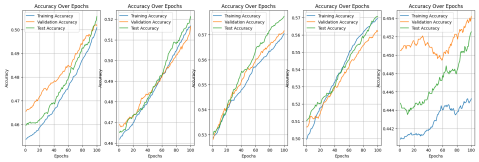

Figure 17: Before Incorporation Model Accuracy over Epochs Plots for DeezerEurope with BANE

Embedding	**TwitchDE**		**Deezer Europe**	
	Results	**Baseline Comparison**	**Results**	**Baseline Comparison**
FeatherNode	64.82% $\pm$ 0.10%	+10.96%	62.02% $\pm$ 0.14%	+11.47%
ASNE	60.76% $\pm$ 0.74%	+6.90%	57.27% $\pm$ 0.29%	+6.72%
BANE	59.87% $\pm$ 0.43%	+6.02%	52.51% $\pm$ 2.05%	+1.96%
Baseline	53.86% $\pm$ 2.14%		50.55% $\pm$ 1.60%	

Table 4: Before Incorporation Results and Baseline Comparison

Table 4 presents the mean test accuracy and standard error for Before Incorporation. FeatherNode embedding showed a substantial improvement over the baseline, with increases of 10.96% for TwitchDE dataset and 11.47% for Deezer Europe. The ASNE embeddings also improved approximately 6-7% over the baseline for both datasets. The BANE embedding, however, improved 6.02% for TwitchDE dataset but only 1.96% for Deezer Europe dataset. Except for BANE for the Deezer Europe dataset, all models have a small standard error, indicating consistent performance across runs.

In summary, the Before Incorporation approach saw notable performance gains with FeatherNode and ASNE embeddings, showcasing enhanced accuracy and lower variability, whereas BANE's performance exhibited more variation across datasets.

4.6 Between Incorporation–Weiyi You

The Between Incorporation embedding-GraphSAGE model was trained with 100 epochs in 5 runs, using 64+size of embedding features as the size of feature input of second GraphSAGE layer. Using the Adam optimizer, with a learning rate of 2×10^{-5}, and the cross entropy loss as loss function, the visually result of 5 runs is shown in below figures181920212223.

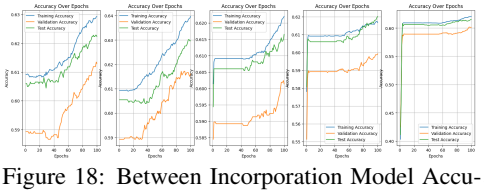

Figure 18: Between Incorporation Model Accuracy over Epochs Plots for TwitchDE with FeatherNode

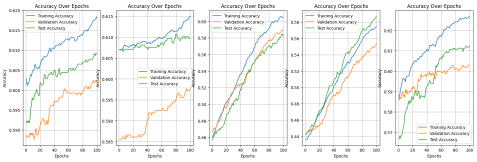

Figure 19: Between Incorporation Model Accuracy over Epochs Plots for TwitchDE with ASNE

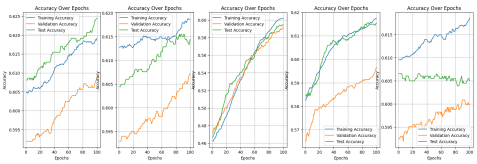

Figure 20: Between Incorporation Model Accuracy over Epochs Plots for TwitchDE with BANE

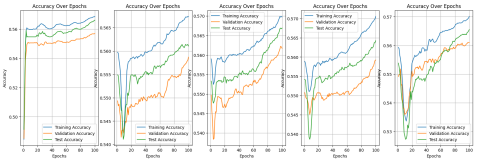

Figure 21: Between Incorporation Model Accuracy over Epochs Plots for DeezerEurope with FeatherNode

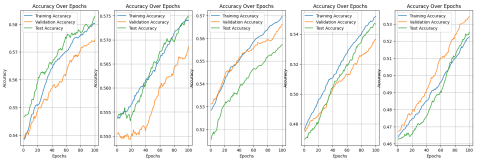

Figure 22: Between Incorporation Model Accuracy over Epochs Plots for DeezerEurope with ASNE

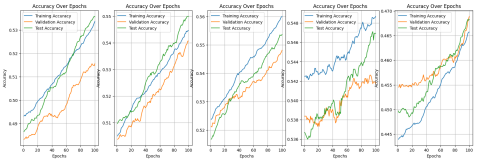

Figure 23: Between Incorporation Model Accuracy over Epochs Plots for DeezerEurope with BANE

Embedding	**TwitchDE**		**Deezer Europe**	
	Results	**Baseline Comparison**	**Results**	**Baseline Comparison**
FeatherNode	61.91% $\pm$ 0.19%	+8.06%	56.46% $\pm$ 0.09%	+5.91%
ASNE	60.01% $\pm$ 0.53%	+6.15%	55.74% $\pm$ 0.91%	+5.19%
BANE	61.05% $\pm$ 0.45%	+7.20%	53.04% $\pm$ 1.42%	+2.49%
Baseline	53.86% $\pm$ 2.14%		50.55% $\pm$ 1.60%	

Table 5: Between Incorporation Results and Baseline Comparison

Table 5 presents the accuracy results and standard errors of Between Incorporation. All Embedding shows outstanding performance in enchanting performance of baseline model with with 6-8% increase and low standard errors in TwitchDE dataset. FeatherNode and ASNE embedding give notable improvement around 5% increase and BANE embedding provides a small 2.49% increase.

In conclusion, all embedding provides significant improvement and low standard erro in TwitchDE dataset with between incorporation approach and FeatherNode embedding provides best improvement in both datasets. The between incorporation approach with featherNode embedding is a relatively stable way to improve the performance of the baseline model.

4.7 Parallel Incorporation–Shurui Wang

Our ParallelGraphSAGE model was trained over five runs with 100 epochs each.A hidden layer with 64 channels was used, and the Adam optimizer with a learning rate of 2×10^{-5} was employed to optimize the CrossEntropyLoss function. The number of classes for the two datasets was defined to tailor the model's output dimensionality accordingly. The trace plot of each run can be found in the figures below242526272829.

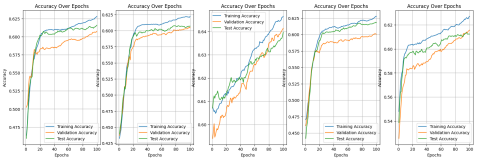

Figure 24: Parallel Incorporation Model Accuracy over Epochs Plots for TwitchDE with FeatherNode

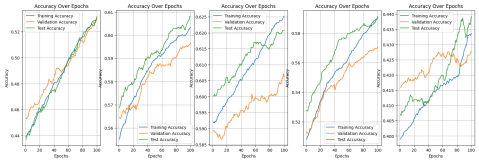

Figure 25: Parallel Incorporation Model Accuracy over Epochs Plots for TwitchDE with ASNE

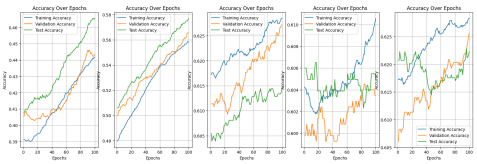

Figure 26: Parallel Incorporation Model Accuracy over Epochs Plots for TwitchDE with BANE

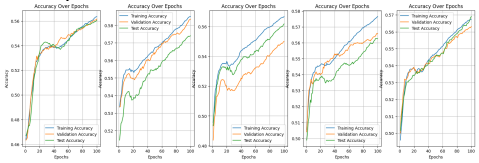

Figure 27: Parallel Incorporation Model Accuracy over Epochs Plots for DeezerEurope with FeatherNode

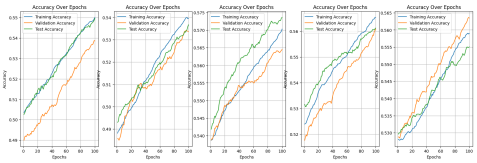

Figure 28: Parallel Incorporation Model Accuracy over Epochs Plots for DeezerEurope with ASNE

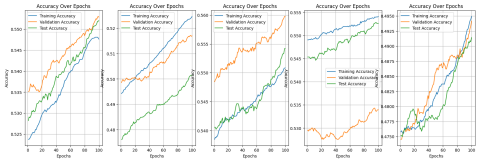

Figure 29: Parallel Incorporation Model Accuracy over Epochs Plots for DeezerEurope with BANE

Embedding	**TwitchDE**		**Deezer Europe**	
	Results	**Baseline Comparison**	**Results**	**Baseline Comparison**
FeatherNode	61.85% $\pm$ 0.50%	+8.00%	56.56% $\pm$ 0.23%	+6.01%
ASNE	55.39% $\pm$ 3.34%	+1.54%	55.45% $\pm$ 0.59%	+4.90%
BANE	57.51% $\pm$ 2.65%	+3.65%	53.03% $\pm$ 1.25%	+2.48%
Baseline	53.86% $\pm$ 2.14%		50.55% $\pm$ 1.60%	

Table 6: Parallel Incorporation Results and Baseline Comparison

Performance was measured by mean test accuracy and standard error, as shown in Table 6. FeatherNode embedding outperformed the baseline, with an 8% increase for TwitchDE and 6.01% for Deezer Europe. ASNE and BANE embeddings resulted in smaller gains. ASNE performs better for Deezer Europe and BANE performs better for TwitchDE. FeatherNode also showed lower standard error rates, indicating more consistent results across runs than ASNE and BANE. In contrast, ASNE and BANE showed higher variability in results across datasets.

In conclusion, FeatherNode provided the most significant performance boost and consistency in parallel incorporation for both datasets.

5 Conclusion

Incorporating global node embeddings, especially FeatherNode Embedding, demonstrated consistent and significant improvements over the baseline in various incorporation strategies. In general, all approaches enhance the accuracy of the baseline model to a certain extent, since the global node embedding extends the features of original nodes. With different embedding methods used in the incorporation strategies, both structural features and node attributes would be extracted by global node embeddings and integrated into the incorporating models. Notice that there are certain differences in the performance of different embedding methods and incorporation strategies in different databases, therefore, the choice of embedding method and incorporation strategy should be carefully considered based on the specific dataset characteristics and classification task. There are still many improvements that can be made in our models, incorporating global node embeddings does not fully utilize the structural features of the nodes in GraphSAGE layers. To improve our models, it is a good idea to make incorporated node embedding methods as optimizable neurons in our GNN.

6 Reproducibility

Source code of this project can be found on Github: https://github.com/yywwwyy/CPSC583.

References

[1] Hamilton, W., Ying, Z. and Leskovec, J. [2017], 'Inductive representation learning on large graphs', *Advances in neural information processing systems* **30**.

[2] Li, Q., Han, Z. and Wu, X.-M. [2018], Deeper insights into graph convolutional networks for semi-supervised learning, *in* 'Thirty-Second AAAI Conference on Artificial Intelligence'.

[3] Liao, L., He, X., Zhang, H. and Chua, T.-S. [2018], 'Attributed social network embedding', *IEEE Transactions on Knowledge and Data Engineering* **30**(12), 2257–2270.

[4] Rozemberczki, B., Allen, C. and Sarkar, R. [2021], 'Multi-scale attributed node embedding', *Journal of Complex Networks* **9**(2), cnab014.

[5] Rozemberczki, B. and Sarkar, R. [2020], Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models, *in* 'Proceedings of the 29th ACM international conference on information & knowledge management', pp. 1325–1334.

[6] Sola, F., A. D. H. I. e. a. [2023], 'Deep embeddings and graph neural networks: using context to improve domain-independent predictions', *Appl Intell(2023)* .

[7] Yang, C., Liu, M., He, X., Wang, C., Sun, M. and Zhang, Y. [2018], Binarized attributed network embedding, *in* '2018 IEEE International Conference on Data Mining (ICDM)', IEEE, pp. 1286–1291.