

# Improving Graph Machine Learning Performance Through Feature Augmentation Based on Network Control Theory

Anwar Said<sup>1</sup>, Obaid Ullah Ahmad<sup>2</sup>, Waseem Abbas<sup>3</sup>, Mudassir Shabbir<sup>4</sup>, Xenofon Koutsoukos<sup>1</sup>

**Abstract**—Network control theory (NCT) offers a robust analytical framework for understanding the influence of network topology on dynamic behaviors, enabling researchers to decipher how certain patterns of external control measures can steer system dynamics towards desired states. Distinguished from other structure-function methodologies, NCT's predictive capabilities can be coupled with deploying Graph Neural Networks (GNNs), which have demonstrated exceptional utility in various network-based learning tasks. However, the performance of GNNs heavily relies on the expressiveness of node features, and the lack of node features can greatly degrade their performance. Furthermore, many real-world systems may lack node-level information, posing a challenge for GNNs. To tackle this challenge, we introduce a novel approach, NCT-based Enhanced Feature Augmentation (NCT-EFA), that assimilates average controllability, along with other centrality indices, into the feature augmentation pipeline to enhance GNNs performance. Our evaluation of NCT-EFA, on six benchmark GNN models across two experimental setting—solely employing average controllability and in combination with additional centrality metrics—showcases an improved performance reaching as high as 11%. Our results demonstrate that incorporating NCT into feature enrichment can substantively extend the applicability and heighten the performance of GNNs in scenarios where node-level information is unavailable.

## I. INTRODUCTION

Network Control theory (NCT), with its rigorous mathematical foundation and practical applicability, has been instrumental in crafting systems that respond predictably to inputs, ensuring stability, efficiency, and desired performance [1]. Its principles permeate modern engineering, deftly guiding everything from simple home appliances to sophisticated aerospace vehicles [2]. Enter graph machine learning, an emerging field that thrives on the abstraction and analysis of relational data, inherently capturing the intricacies of complex systems as networks [3]. By intertwining the predictive prowess of graph-based models with the robust framework of control theory, there lies a profound opportu-

nity to revolutionize how we understand, design, and manage dynamic systems [4].

Average controllability is a critical metric in systems theory that quantifies the ability of a node within a network to influence the system's overall behavior through the introduction of external inputs [5]. It captures the essence of a node's role in controlling the dynamical landscape of a network by measuring the system's response to impulses, effectively gauging a node's intrinsic capability to steer the system's states.

As such, this metric is invaluable for discerning the underlying structural properties that dictate how control energy proliferates through the network, offering a window into the architecture's innate information flow.

Building on the insights provided by average controllability within the purview of NCT, network science on the other hand offers an expansive framework to further dissect and enhance our understanding of systemic influence and regulation [6]. Centrality measures within network science have emerged as pivotal tools in this venture, quantifying the roles and significance of individual nodes amid the complex web of interactions that typify controlled systems. Closeness centrality reveals those nodes that efficiently affect the entire network, due to their minimized path lengths to all other nodes, underpinning their strategic value for swift and comprehensive system-wide influence [7]. Eigenvector centrality takes into account the notion that connections to highly influential nodes contribute more significantly to a node's importance, emphasizing the network's inherent hierarchy and influence distribution. Meanwhile, betweenness centrality captures the nodes that frequently serve as bridges in the shortest paths between others, highlighting their critical role in facilitating or bottling the flow of control through the network [8]. These centrality metrics, thus, encapsulate vital information about the network structure, and play vital role in many modeling tasks [9].

Graph Neural Networks (GNNs) have emerged as a sophisticated approach for performing machine learning on graph-structured data, harnessing the framework of message passing to learn refined node representations that are critical for a variety of downstream machine learning tasks [10]. The message passing mechanism - central to the functionality of GNNs - iteratively updates node representations by aggregating features from their respective neighborhoods, thereby encapsulating both local and global structural information within the graph [11]. Nevertheless, the performance of GNNs is intrinsically linked to the initial availability and quality of the node features; deficient feature sets can sig-

\*This material is based upon work supported by the National Science Foundation Grant Nos. 2325416, and 2325417.

<sup>1</sup>Anwar Said, and Xenofon Koutsoukos are with the Computer Science Department at the Vanderbilt University, Nashville, TN. Email: anwar.said@vanderbilt.edu, xenofon.koutsoukos@vanderbilt.edu

<sup>2</sup>Obaid Ullah Ahmad is with the Electrical Engineering Department at the University of Texas at Dallas, Richardson, TX 75080. Email: Obaidullah.Ahmad@utdallas.edu

<sup>3</sup>Waseem Abbas is with the Systems Engineering Department at the University of Texas at Dallas, Richardson, TX. Email: waseem.abbas@utdallas.edu

<sup>4</sup>Mudassir Shabbir is with the Computer Science Department at Information Technology University, Lahore, Pakistan Email: mudassir.shabbir@itu.edu.pk

nificantly curtail the representational capacity of the learned embeddings [12]. A prevalent challenge arises in circumstances where node features are sparse or entirely missing. Under such constraints, the customary practice involves the utilization of one-hot encoding of node degrees as surrogate features, injecting a rudimentary form of structural information into the model [13]. Despite this adaptation, the absence of rich, discriminative features often results in GNNs that struggle to achieve the depth of understanding necessary for complex inference tasks, thereby revealing a limitation in their ability to thoroughly exploit the intricate relational patterns present within graph data [14].

To address the challenge of feature impoverishment in GNNs, our study proposes leveraging network controllability and centrality measures to enrich node feature sets, potentially improving GNN performance. Our main contributions are as follows:

- **Feature augmentation:** We propose integrating metrics like average controllability, betweenness centrality, closeness centrality, and eigenvector centrality as node features within the GNN framework.
- **Unique representation of average controllability:** We use a new node representation method to encode average controllability as a node feature. Through an ablation study, we demonstrate the effectiveness of the proposed scheme in enhancing classification performance.
- **Evaluation and comparison:** In social network classification domain, where node features are missing, we evaluate several GNN models using our proposed approach. We compare these models with baseline techniques such as one-hot-degree encoding. Our results demonstrate the superior performance of our proposed method.

By incorporating these additional metrics and exploring novel representations, our study aims to enhance the capabilities of GNNs and address the challenge of missing feature in network analysis tasks.

## II. RELATED WORK

The field of integrating NCT with graph machine learning is still in its early stages, with few studies exploring this intersection. To the best of our knowledge, only one study has been proposed that uses NCT metrics to derive graph representations for downstream graph classification tasks [4]. However, no existing study has integrated NCT with GNNs to advance the field. In the following, we briefly present [4] and provide references for further reading to interested readers. Due to space limitations, we keep the related work section concise.

The authors in [4] introduce an intriguing approach to integrate NCT with graph machine learning. The core idea of the paper involves using various controllability metrics to derive expressive graph representations. Specifically, this study explores the controllability gramian as a source of controllability information. It considers metrics such as trace, rank, and the first and last three eigenvalues of the controllability gramian as potential features for graph representations.

Moreover, it uses different numbers of leaders and repeats the process to derive an expressive set of features for the final graph representation. This study serves as a catalyst for further research to explore the potential of integrating NCT with GNNs, opening new avenues for advancing the field of graph machine learning.

In the realm of graph machine learning, the field is characterized by a rapid pace of research, primarily focusing on two main approaches: graph embedding methods and GNNs. Graph embedding methods employ various graph-theoretic and statistical techniques to derive representations of graphs, which are then utilized independently to train machine learning models such as support vector machines and random forests [15], [16]. A few notable and recent graph embedding methods include [17], [18], [19]. On the other hand, GNNs are deep learning-based approaches that are trained in an end-to-end fashion. They utilize techniques such as message passing, spectral methods, and recurrent neural networks to learn representations at the node and graph levels. GNNs offer an advantage over embedding methods by effectively leveraging both the graph's topology and its node and edge features [20], [13]. For further exploration of these concepts, interested readers are encouraged to refer to [3], [10], [21], [22].

## III. METHODOLOGY

NCT is a powerful paradigm originally rooted in the domain of systems and control engineering, which has found its footing in the study of social networks by offering a sophisticated mathematical framework for understanding how influence and information propagate through complex relational systems. When applied to social networks, NCT can provide a nuanced perspective on the capacity of individual nodes, or users within the network, to affect overall network behavior, based on the topology and dynamics of social interactions [23].

### A. Network Controllability Metrics

By identifying key control nodes within a social network—akin to the influential nodes in a social network—NCT enables us to tailor node representations such that they encapsulate not only the structural attributes of these entities but also their potential to exert control over the network state. This could include their capability to disseminate information effectively or to reconfigure connections for desired social outcomes strategically. Employing NCT to parameterize and analyze control nodes allows for the enhancement of node features, which can, in turn, improve the performance of models on downstream tasks such as social influence prediction, node classification, and graph classification, among others.

The structural composition of NCT encapsulates a network defined by the adjacency matrix  $A \in \mathbb{R}^{N \times N}$  of  $N = |V|$  nodes and a control set  $B \in \mathbb{R}^{N \times m}$ . Within this framework, it posits that the temporal evolution of the state of any given node, denoted  $x_i(t)$ , is governed by a composite function. This function represents the cumulative influence of all

preceding nodes, expressed as  $x_j(t)$ , in conjunction with any external inputs,  $u(t)$ , each modulated by appropriate weights within the network's topology. When the progression of node states is conceptualized through the lens of rates of change, such that the activity from preceding nodes influences the ongoing rate of state alteration in subsequent nodes, the model adopts the construct of a differential equation presented as follows:

$$\frac{d}{dt}\mathbf{x}(t) = A\mathbf{x} + B\mathbf{u}(t) \quad (1)$$

where  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^\top$  is the vector of node states,  $A$  is the adjacency matrix, and  $\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_n(t)]^\top$  is the vector of control signals.  $B^{N \times m}$  quantifies the effects of inputs on each node. In our experiments, the control set matrix  $B^{N \times N}$  is defined as the identity matrix, representing a uniform full control set.

The controllability Gramian is a powerful mathematical concept that plays a crucial role in understanding the control behavior of a network [24], [25]. By utilizing the controllability Gramian, we can measure the ease with which we can transition from one state to another in terms of the necessary control energy. In the context of the system defined in Equation 1, the infinite horizon controllability Gramian can be formally expressed as follows:

$$\mathcal{W} = \int_0^\infty e^{-A\tau}(-B)(-B)^\top e^{-A^\top\tau} d\tau \in \mathbb{R}^{N_f \times N_f}. \quad (2)$$

When the system is stable, implying that all eigenvalues of  $-A$  possess negative real parts,  $\mathcal{W}$  converges asymptotically and can be determined through the Lyapunov equation.

$$(-A)\mathcal{W} + \mathcal{W}(-A)^\top + (-B)(-B)^\top = 0, \quad (3)$$

The formulation presented in Equation 1 provides us with a framework to derive various NCT metrics from the observed system—among these, a key parameter of interest is average controllability, which is the focal point of the current study.

**Average Controllability:** Average controllability quantifies the extent to which the state vector,  $\mathbf{x}(t)$ , can be modulated in response to input stimuli administered to an individual node within the network [9]. A node that exhibits a higher degree of average controllability possesses an enhanced capacity to utilize the underlying graph structure to disseminate an impetus across the entire network. Average controllability is defined as the trace of the controllability Gramian, expressed as:

$$\mathcal{C}_a = \text{tr}(\mathcal{W})$$

Next, we detail GNNs, that will be used as a learning architecture for social network classification task.

## B. Graph Neural Networks

GNNs represent an innovative class of neural networks specifically designed to operate on graph-structured data [3]. GNNs are particularly adept at learning from graph-based data due to their ability to capture the inherent relationships and interdependencies between nodes through a process of iterative information aggregation [10]. Unlike traditional neural networks that assume independent and identical distribution of data, GNNs embrace the irregularities and topological characteristics of graphs, enabling them to learn and make predictions on data that exhibits rich relational context. This capability makes GNNs the preferred choice for a multitude of applications ranging from social network analysis, brain networks, and recommendation systems to traffic network flow optimization, where relational patterns are of paramount importance [26], [3].

Building upon the foundation of GNNs, Message Passing Neural Networks (MPNNs) extend this concept with a specific focus on node features and their interactions. MPNNs operate on the principle that each node's feature representation is updated by recursively aggregating features from its neighbors, thereby encapsulating both local structures and global context within the graph. This process is formalized by the message passing equation:

$$\mathbf{h}_v^{(t+1)} = \text{UPDATE} \left( \mathbf{h}_v^{(t)}, \text{AGGREGATE} \left( \mathbf{h}_u^{(t)} : u \in \mathcal{N}(v) \right) \right)$$

Here,  $\mathbf{h}_v^{(t+1)}$  represents the updated feature vector of node  $v$  at iteration  $(t+1)$ ; UPDATE and AGGREGATE are functions that define the updating and aggregation mechanisms, respectively;  $\mathbf{h}_u^{(t)}$  is the feature vector of a neighboring node  $u$ ; and  $\mathcal{N}(v)$  denotes the set of neighbors of node  $v$ . The power of MPNNs lies in the expressiveness of learned node representations, which can significantly enhance model performance on downstream tasks. Consequently, integrating better features through improved UPDATE and AGGREGATE functions can result in more expressive GNN models, capable of capturing complex patterns and ultimately leading to more accurate predictions.

## C. Constructing Node Features

In many real-world networks, the absence of node features is commonplace, attributable to a variety of factors including privacy constraints, data collection errors, proprietary information protection, and instances where features may be unobserved or inherently non-existent. In situations where node attributes are lacking, strategies like one-hot-degree encoding or the employment of randomly assigned features are often utilized to facilitate the training of MPNNs. However, the performance of these models can be compromised due to the paucity of informative features, a limitation corroborated by numerous preceding studies [12]. To mitigate this challenge and endow graphs devoid of node features with enriched representations, we advocate for the integration of NCT which has been described in section III-A. In the following section, we detail the rest of the network science measures that we use for enriching the node features.

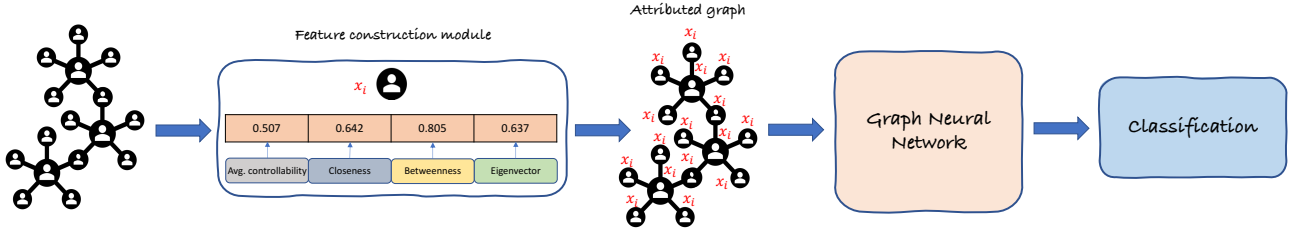


Fig. 1. Illustration of the proposed NCT-EFA framework. Starting with a simple graph, NCT-EFA utilizes network control theory and network science metrics to enhance the graph by node features. These graphs with enriched features are subsequently fed into GNN module for training and the downstream classification task.

**Closeness Centrality:** Closeness centrality is a measure used in network analysis to identify the relative importance of a node within a graph, based on its proximity to all other nodes. It is calculated as the reciprocal of the sum of the shortest path distances from a given node to all other nodes in the network. Nodes with higher closeness centrality are seen as having better potential to quickly interact with all others due to their central positioning in the graph’s structure.

**Betweenness Centrality:** Betweenness centrality quantifies the significance of a node within a network by measuring the extent to which the node acts as a bridge along the shortest paths between other nodes. It is computed as the fraction of all-pairs shortest paths that pass through a given node, indicating its role as an intermediary in facilitating communication or connectivity within the network. Nodes with high betweenness centrality are often crucial for the flow of information, resources, or influence across the network, as they can control and influence the interactions between different parts of the graph.

**Eigenvector Centrality:** This is a measure of influence within a network that assigns relative scores to all nodes based on the principle that connections to high-scoring nodes contribute more to the score of a node than equal connections to low-scoring nodes. This centrality considers not only the number of connections or edges a node has but also the quality of those connections in terms of the centrality of its neighbors. It is calculated by finding the eigenvector corresponding to the largest eigenvalue of the network’s adjacency matrix, with the components of this principal eigenvector giving the centrality scores of the nodes.

Building upon the four central measures—we compute these metrics for each individual node within the network and collate them into a feature vector representative of the node’s structural characteristics. This vector of network attributes effectively captures the node’s connectivity profile, potential for information dissemination, and overall influence within the network structure. To aid in understanding and visualizing this methodology, an illustrative representation of how these features are computed and assembled for every node has been shown in Figure 1.

**Ranks one-hot encoding:** In various systems, there’s often a need to introduce external inputs or amplify the influence of key users to alter the system’s behavior. One example could be considering the influential users in a social

network. In such cases, traditional network science measures that focus solely on graph topology may not suffice. Instead, we may rely on average controllability as a node feature. However, using a single real number as a feature may not be optimal for the model’s performance. To address this, we propose encoding average controllability in the following manner.

Given the average controllability vector for the entire graph, computed as described in [23] where each value corresponds to a node in the graph, we create a histogram  $\mathcal{H}$  with  $k$  bins to represent the distribution. Each bin corresponds to a range of average controllability values, and the height of each bin represents the frequency of nodes with average controllability values falling within that range. This histogram provides a summary of the controllability distribution across nodes in the graph, highlighting the prevalence of certain controllability levels. To create a feature vector for a node  $v$  using  $\mathcal{H}$ , we one-hot encode the feature vector based on the index  $\mathcal{H}(i)$  where the average controllability of node  $v$  falls. Formally, the one-hot encoding for the feature vector  $\mathbf{h}_v^0$  can be expressed as:

$$\mathbf{h}_v^0(i) = \begin{cases} 1 & \text{if } C_a(v) \in \mathcal{H}(i) \\ 0 & \text{otherwise} \end{cases}$$

The final component of our framework is dedicated to learning over the attributed graph. Within this context, we may employ any message passing-based GNN variant and adhere to conventional training protocols to fine-tune the model. In training our model for the graph classification task, we employ a binary cross-entropy loss to optimize the learning process. After training, the model is then applied to the binary classification task, using its learned features to distinguish between two separate classes.

#### IV. NUMERICAL EVALUATION

In this section, we evaluate the efficacy of our proposed methodology under two distinct experimental settings. Initially, we undertake a comparative analysis between the comprehensive feature set—encompassing average controllability, closeness centrality, betweenness centrality, and eigenvector centrality—and the baseline one-hot-degree encoding method. This comparison aims to establish the superiority of our enriched feature set in capturing the intricate topological nuances of the graph data. Subsequently, we show the

TABLE I  
DATASET STATS

Dataset	Graphs	Nodes		Density		Diameter		Classes	Task
		Min	Max	Min	Max	Min	Max		
Reddit Threads	203,088	11	97	0.021	0.382	2	27	2	Graph Classification
GitHub Stargazers	12,725	10	957	0.003	0.561	2	18	2	Graph Classification

TABLE II  
COMPARISON OF ROC AUC SCORES OF THE PROPOSED METHOD AGAINST ONE-HOT-DEGREE ENCODING (DEG)

Datasets	k-GNN		SAGE		GCN		UniMP		ResGatedGCN		GAT	
	deg	NCT-EFA	deg	NCT-EFA	deg	NCT-EFA	deg	NCT-EFA	deg	NCT-EFA	deg	NCT-EFA
Reddit Threads	83.72	<b>84.06</b>	83.64	<b>83.90</b>	82.83	<b>83.79</b>	83.83	<b>83.96</b>	84.01	<b>84.12</b>	<b>83.88</b>	83.86
GitHub Stargazers	71.21	<b>79.03</b>	66.27	<b>75.82</b>	68.34	<b>74.51</b>	68.39	<b>77.48</b>	74.97	<b>79.37</b>	64.21	<b>75.90</b>

effectiveness of employing one-hot-encoded ranks derived from average controllability measures. This second setting allows us to assess the impact of prioritizing nodes based on their average controllability and to ascertain the resultant performance gains in our graph neural network model.

#### A. Datasets

We consider the following two social network datasets in our experimental setup.

**Reddit Threads:** This dataset comprises an assortment of threads extracted from the Reddit platform, all of which were gathered during the month of May 2018. It encompasses both discussion and non-discussion based threads, presenting a diverse range of community interactions. The task is to distinguish between threads that facilitate discussion and those that do not, thereby classifying the content based on its conversational nature and potential for user engagement [27].

**GitHub Stargazers:** This is a social network dataset of developers who have starred notable machine learning and web development repositories on GitHub. The task involves classifying these networks to ascertain if they correspond to web development or machine learning repositories based on their stargazing activities [27].

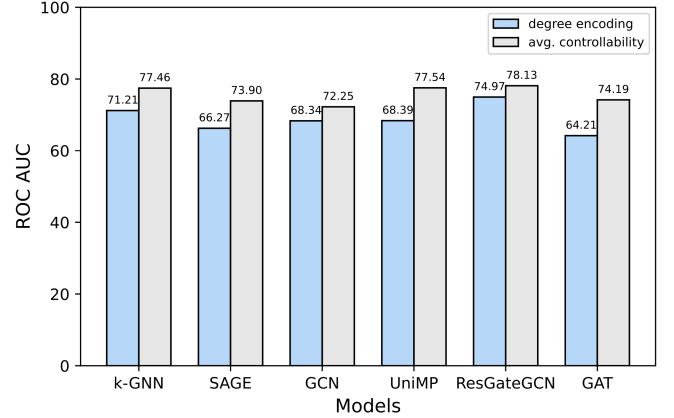
Both of these datasets correspond to a graph classification task. We present the statistics of these datasets in Table I.

#### B. Experimental Setup and Results

We consider six well-known baselines graph convolution methods that include  $k$ -GNN [28], GraphSAGE [11], GCN [10], Transformer Convolution (UniMP) [29], Residual Gated Graph ConvNets (ResGatedGCN) [20] and Graph Attention Network (GAT) [21].

The proposed learning framework comprises three layers of GNNs, each containing 64 hidden units. Post these layers, Sort Aggregation [30] is applied. Following the aggregation step, we employ two layers of 1D convolution complemented by Max Pooling. This is succeeded by a multi-layer perceptron with two layers, each having 32 hidden neurons. For model evaluation, we resort to 10-fold cross-validation, training each model for a duration of 100 epochs. We set the learning rate at  $1e^{-4}$  and the weight decay at  $5e^{-2}$ . All experiments are conducted on a Lambda machine equipped with an AMD Ryzen Threadripper PRO 5995WX64-Core

Fig. 2. Comparison of ROC AUC scores of the encoded ranks (average controllability) against degree one-hot-encoding.



CPU, 512 GB RAM, and an NVIDIA RTX 6000 GPU with 48 GB of memory.

#### C. Results with Full Set of Features

We present the ROC AUC (Receiver Operating Characteristic Area Under the Curve) classification results for both datasets using the full set of features in Table II. We maintained consistent architecture and experimental settings for evaluation, comparing the performance of both one-hot-degree encoding (deg) and the proposed method (NCT-EFA). Overall, we observed improved performance with NCT-EFA on both datasets, except for the GAT results on the Reddit dataset, which exhibited a slight (0.02%) decrease. Notably, NCT-EFA yielded an 11.69% improvement with GAT on the GitHub Stargazers dataset, a 9.55% improvement with GraphSAGE, and a 9.09% improvement with UniMP. These results underscore the effectiveness of the proposed NCT-EFA approach for downstream classification tasks. It is worth noting that the relatively minor difference between one-hot-degree encoding and NCT-EFA on Reddit Threads dataset can be attributed to the dataset's graph topology playing a more significant role in classification than the node features.

#### D. Numerical Results with Average Controllability

To illustrate the effectiveness of using only the average controllability, we conducted additional experiments. In this setup, we maintained the same architecture and experimental configuration but utilized one-hot encoded average controllability values, as discussed in Section III-C. We computed

the average controllability for each node and then encoded these values to create the corresponding node features. Subsequently, we trained the same models using this encoded dataset and compared the results with those using one-hot-degree encoding. The results are presented in Figure 2.

Our results (ROC AUC) indicate that encoding average controllability significantly enhances the performance of all the GNN models. Specifically, we observed a 9.98% improvement with GAT, 9.15% with UniMP, and 7.63% with GraphSAGE. Notably, the improvements seen with the full set of features in the previous section were 11.69%, 9.55%, and 9.05%, respectively. This suggests that average controllability contributes significantly more compared to the other features in these datasets.

## V. CONCLUSIONS AND FUTURE WORK

In this work, we have introduced a novel feature augmentation approach based on NCT, aimed at enhancing node features to boost the performance of GNNs in graph classification task within social networks. Our method leverages average controllability, closeness centrality, betweenness centrality, and eigenvector centrality as node features, which we evaluated with six different GNN architectures. Additionally, we introduced a novel rank encoding scheme that constructs expressive node features from average controllability, further enhancing the approach's effectiveness.

Our experimental results across various GNNs and a diverse social network dataset demonstrate consistent performance improvements, underscoring the efficacy of our proposed approach. These promising results motivate several key avenues for future exploration. For instance, combining various controllability metrics to augment GNNs could enhance their capabilities further. Additionally, integrating controllability metrics directly into the message passing mechanism of GNNs presents an intriguing research direction. Lastly, leveraging controllability metrics for constructing expressive graph embeddings represents another exciting path for future investigation.

The field of integrating controllability metrics with GNNs is relatively young but shows immense potential for developing more expressive graph machine learning methods. We believe that our work lays a foundation for exploring the integration of these two fields, paving the way for groundbreaking advancements in graph machine learning.

## REFERENCES

- [1] Y. Yazıcıoğlu, M. Shabbir, W. Abbas, and X. Koutsoukos, "Strong structural controllability of networks: Comparison of bounds using distances and zero forcing," *Automatica*, vol. 146, p. 110562, 2022.
- [2] B. Friedland, *Control system design: an introduction to state-space methods*. Courier Corporation, 2012.
- [3] W. L. Hamilton, *Graph representation learning*. Morgan & Claypool Publishers, 2020.
- [4] A. Said, O. U. Ahmad, W. Abbas, M. Shabbir, and X. Koutsoukos, "Network controllability perspectives on graph representation," *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [5] R. E. Kalman *et al.*, "Contributions to the theory of optimal control," *Bol. soc. mat. mexicana*, vol. 5, no. 2, pp. 102–119, 1960.
- [6] A.-L. Barabási, "Network science," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1987, p. 20120375, 2013.
- [7] A. Said, R. A. Abbasi, O. Maqbool, A. Daud, and N. R. Aljohani, "Cc-ga: A clustering coefficient based genetic algorithm for detecting communities in social networks," *Applied Soft Computing*, 2018.
- [8] L. C. Freeman *et al.*, "Centrality in social networks: Conceptual clarification," *Social network: critical concepts in sociology. Londres: Routledge*, vol. 1, pp. 238–263, 2002.
- [9] L. Parkes, T. M. Moore, M. E. Calkins, M. Cieslak, D. R. Roalf, D. H. Wolf, R. C. Gur, R. E. Gur, T. D. Satterthwaite, and D. S. Bassett, "Network controllability in transmodal cortex predicts positive psychosis spectrum symptoms," *Biological Psychiatry*, vol. 90, no. 6, pp. 409–418, 2021.
- [10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [11] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.
- [12] A. Said, M. Shabbir, T. Derr, W. Abbas, and X. Koutsoukos, "Enhanced graph neural networks with ego-centric spectral subgraph embeddings augmentation," *arXiv preprint arXiv:2310.12169*, 2023.
- [13] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.
- [14] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *International conference on machine learning*. PMLR, 2017, pp. 1263–1272.
- [15] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels," *Applied Network Science*, vol. 5, no. 1, pp. 1–42, 2020.
- [16] A. Said, T. Derr, M. Shabbir, W. Abbas, and X. Koutsoukos, "A survey of graph unlearning," *arXiv preprint arXiv:2310.02164*, 2023.
- [17] M. Togninalli, E. Ghisu, F. Llinas-López, B. Rieck, and K. Borgwardt, "Wasserstein weisfeiler-lehman graph kernels," *Advances in neural information processing systems*, vol. 32, 2019.
- [18] R. Kondor and H. Pan, "The multiscale laplacian graph kernel," *Advances in neural information processing systems*, vol. 29, 2016.
- [19] A. Said, M. Shabbir, S.-U. Hassan, Z. R. Hassan, A. Ahmed, and X. Koutsoukos, "On augmenting topological graph representations for attributed graphs," *Applied Soft Computing*, vol. 136, p. 110104, 2023.
- [20] X. Bresson and T. Laurent, "Residual gated graph convnets," *arXiv preprint arXiv:1711.07553*, 2017.
- [21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
- [22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, no. 1, 2020.
- [23] L. Parkes, J. Z. Kim, J. Stiso, J. K. Brynildsen, M. Cieslak, S. Covitz, R. E. Gur, R. C. Gur, F. Pasqualetti, R. T. Shinohara *et al.*, "Using network control theory to study the dynamics of the structural connectome," *bioRxiv*, 2023.
- [24] F. Pasqualetti, S. Zampieri, and F. Bullo, "Controllability metrics, limitations and algorithms for complex networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 40–52, 2014.
- [25] O. U. Ahmad, A. Said, M. Shabbir, W. Abbas, and X. Koutsoukos, "Control-based graph embeddings with data augmentation for contrastive learning," *arXiv preprint arXiv:2403.04923*, 2024.
- [26] A. Said, R. G. Bayrak, T. Derr, M. Shabbir, D. Moyer, C. Chang, and X. Koutsoukos, "Neurograph: Benchmarks for graph machine learning in brain connectomics," *arXiv preprint arXiv:2306.06202*, 2023.
- [27] B. Rozenberczki, O. Kiss, and R. Sarkar, "Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs," in *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20)*. ACM, 2020, p. 3125–3132.
- [28] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4602–4609.
- [29] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: Unified message passing model for semi-supervised classification," *arXiv preprint arXiv:2009.03509*, 2020.
- [30] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, "An end-to-end deep learning architecture for graph classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.