

AN OPTIMIZATION BASED FRAMEWORK FOR GRAPH COARSENING

MANOJ KUMAR



**DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY DELHI
OCTOBER 2024**

© Indian Institute of Technology Delhi (IITD), New Delhi, 2024

AN OPTIMIZATION BASED FRAMEWORK FOR GRAPH COARSENING

by

MANOJ KUMAR

DEPARTMENT OF ELECTRICAL ENGINEERING

Submitted

In fulfilment of the requirements of the degree of doctor of philosophy

to the



INDIAN INSTITUTE OF TECHNOLOGY DELHI

OCTOBER 2024

CERTIFICATE

This is to certify that the thesis titled **An Optimization Based Framework for Graph Coarsening**, submitted by **Manoj Kumar**, to the Indian Institute of Technology, Delhi, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under my supervision.

The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Sandeep Kumar
Thesis Supervisor
Department of Electrical Engineering
Indian Institute of Technology, Delhi
Hauz Khas New Delhi 110016

Place: New Delhi

ACKNOWLEDGEMENTS

My sincere gratitude goes to everyone who has contributed to this thesis in both direct and indirect manners. First and foremost, I would like to express my deepest thanks to my supervisor, Dr. Sandeep Kumar, for his continuous support, inspiration, motivation, scrutiny, and impetus. He guided me not only through my academic journey but also expanded my view of research and aided my personal evolution. During my doctoral studies, I greatly benefited from collaborating with very enthusiastic master's and undergraduate students—Anurag Sharma, Kumar Arjun, Prakash Pal, Samarh Bhatia, Yukti Makheeja, and Shashwat Saxena. Appropriate credit goes to them for some of the works reported here. I am also privileged to have been taught and mentored by inspiring professors at IIT Delhi, IIT BHU and Stony Brook University USA. My sincere thanks go to Prof. Surendra Prasad, Dr. Deepak Umakant Patil, Dr. Arpan Chattopadhyay, Dr. Yifan Sun, Prof. Ruchir Gupta and Dr. Shaurya Sriyam. My due veneration also goes to the reviewers of journals and conferences, whose rigorous scrutiny of my work has significantly shaped this research.

During my time here, I have been fortunate to be surrounded by wonderful friends. My heartfelt gratitude and best wishes go to Subhanu Halder, Gourab Panda, Abhishek Gupta, Sumit Kumar, Prakash Pal, Mohit kataria, Vipul kumar singh, Nikita Malik, Jyotismita Burman, and Ekta Srivastava for all the fun moments and challenging times we shared. I could not have imagined spending a single day without you all. Special thanks go to my family for supporting me in all my adventures and persistently believing in me. To my wife, Chandini Sengar, and daughter, Rhythm, who always stood behind me, impressing upon me that 'You can do it.' And to my parents, for always believing in me and pushing me to go beyond my fears.

Thank you all.

Manoj Kumar

ABSTRACT

Graph-based approaches play a pivotal role in advancing research across various domains within big data and machine learning. However, the surge in large-scale graph data presents significant computational challenges. Graph coarsening, a technique to summarize large graphs while preserving essential properties, emerges as a crucial solution. Existing methods often overlook node features, which impacts their applicability in real-world scenarios where both the graph matrix and node features are important.

In this thesis, we have developed a family of graph coarsening algorithms that aim to learn a coarsened graph by considering either the Laplacian matrix alone, the feature matrix alone, or both. The work introduces the novel optimization-based Featured Graph Coarsening (FGC) and its variants, which demonstrate superior performance compared to state-of-the-art methods. FGC aims to learn a smaller, tractable graph along with its feature matrix by considering the original graph and its feature matrix. This multi-block non-convex optimization problem simultaneously learns the coarsened graph matrix and feature matrix while ensuring desired properties. The algorithm uses majorization-minimization, log determinant, Dirichlet energy, and regularization frameworks, making it provably convergent and enforcing properties such as spectral similarity and ϵ -similarity in the learned coarsened graph. FGC is extended with Feature Dimensionality Reduction (FGCR), where we simultaneously reduce the dimension of features for each node of the graph while coarsening, unlike in FGC where the feature dimensions of the original and coarsened graphs remain the same.

The scalability of graph neural networks is addressed through an optimization framework for learning sparse featured coarsened graphs, resulting in improved generalization and computational efficiency. We have also developed a convex framework for feature-based graph coarsening that outperforms FGC in terms of speed while maintaining comparable performance. Furthermore, Structured Graph Coarsening algorithms are proposed to address the limitations of existing methods in coarsening graphs with specific properties. The unified optimization framework efficiently enforces structures like sparsity,

scale-free characteristics, bipartite, and multi-component structures.

Lastly, the coarsened graph learning algorithm, which represents the first method that learns a coarsened graph using raw data only, is introduced with provable convergence. Extensive experiments across various real-world applications demonstrate the efficacy of the proposed frameworks. This work contributes to advancing graph-based machine learning by addressing critical challenges in large-scale graph data analysis and coarsening techniques.

सार

ग्राफ-आधारित दृष्टिकोण बड़े डेटा और मशीन लर्निंग के विभिन्न क्षेत्रों में अनुसंधान को आगे बढ़ाने में एक महत्वपूर्ण भूमिका निभाते हैं। हालांकि, बड़े पैमाने पर ग्राफ डेटा में वृद्धि से महत्वपूर्ण गणनात्मक चुनौतियाँ उत्पन्न होती हैं। ग्राफ कोर्सनिंग, एक तकनीक है जो बड़े ग्राफ को संक्षिप्त करने के साथ उनके आवश्यक गुणों को बनाए रखने में सहायक होती है। मौजूदा तरीकों में अक्सर नोड फीचर्स को नज़रअंदाज़ कर दिया जाता है, जिससे उन तरीकों की वास्तविक दुनिया के परिदृश्यों में उपयोगिता प्रभावित होती है जहाँ ग्राफ मैट्रिक्स और नोड फीचर्स दोनों महत्वपूर्ण होते हैं।

इस शोध में, हमने ग्राफ कोर्सनिंग एल्गोरिदम का एक समूह विकसित किया है, जिसका उद्देश्य या तो केवल लैप्लासियन मैट्रिक्स, केवल फीचर मैट्रिक्स, या दोनों को ध्यान में रखते हुए एक संक्षिप्त ग्राफ सीखना है। इस कार्य में एक नया अनुकूलन-आधारित फीचर्ड ग्राफ कोर्सनिंग और इसके विभिन्न संस्करण प्रस्तुत किए गए हैं, जो आधुनिक तरीकों की तुलना में बेहतर प्रदर्शन दिखाते हैं। फीचर्ड ग्राफ कोर्सनिंग का उद्देश्य मूल ग्राफ और उसके फीचर मैट्रिक्स को ध्यान में रखते हुए एक छोटा, व्यावहारिक ग्राफ और उसका फीचर मैट्रिक्स सीखना है। यह एक बहु-ब्लॉक गैर-अवलोकनीय अनुकूलन समस्या है जो एक साथ संक्षिप्त ग्राफ मैट्रिक्स और फीचर मैट्रिक्स को सीखती है, जबकि आवश्यक गुणों को सुनिश्चित करती है। यह एल्गोरिदम मेजराइज़ेशन-मिनिमाइज़ेशन, लॉग डिटरमिनेंट, डिक्ले ऊर्जा, और नियमितीकरण फ्रेमवर्क का उपयोग करता है, जो इसे सिद्ध रूप से अभिसरण करने योग्य बनाता है और सीखे गए संक्षिप्त ग्राफ में स्पेक्ट्रल समानता और ϵ -समानता जैसी गुणों को लागू करता है। फीचर्ड ग्राफ कोर्सनिंग को फीचर डाइमेंशनैलिटी रिडक्शन के साथ विस्तारित किया गया है, जहाँ हम ग्राफ को संक्षिप्त करते समय प्रत्येक नोड के फीचर्स के आयामों को भी कम करते हैं, जबकि फीचर्ड ग्राफ कोर्सनिंग में मूल और संक्षिप्त ग्राफ के फीचर आयाम समान रहते हैं।

ग्राफ न्यूरल नेटवर्क्स की स्केलेबिलिटी को एक अनुकूलन फ्रेमवर्क के माध्यम से संबोधित किया गया है, जो कि विरल फीचर्ड कोर्सन किए गए ग्राफ को सीखने पर आधारित है, जिससे सामान्यीकरण में सुधार और गणनात्मक दक्षता प्राप्त होती है। हमने फीचर-आधारित ग्राफ कोर्सनिंग के लिए एक उत्तल फ्रेमवर्क भी विकसित किया है, जो गति के मामले में फीचर्ड ग्राफ कोर्सनिंग से बेहतर प्रदर्शन करता है, जबकि तुलनात्मक प्रदर्शन बनाए रखता है। इसके अलावा, स्ट्रक्चर्ड ग्राफ कोर्सनिंग एल्गोरिदम का प्रस्ताव किया गया है, जो विशेष गुणों वाले ग्राफ कोर्सनिंग के मौजूदा तरीकों की सीमाओं को दूर करता है। एकीकृत अनुकूलन फ्रेमवर्क कुशलता से विरलता, स्केल-फ्री गुणधर्म, द्विदलीय और बहु-घटक संरचनाओं जैसी संरचनाओं को लागू करता है।

अंततः, कोर्सन किए गए ग्राफ लर्निंग एल्गोरिदम, जो केवल कच्चे डेटा का उपयोग करके एक संक्षिप्त ग्राफ सीखने की पहली विधि है, को सिद्ध रूप से अभिसरण के साथ पेश किया गया है। विभिन्न वास्तविक अनुप्रयोगों में किए गए व्यापक प्रयोग इन प्रस्तावित फ्रेमवर्क की प्रभावशीलता को प्रदर्शित करते हैं। यह कार्य बड़े पैमाने पर ग्राफ डेटा विश्लेषण और कोर्सनिंग तकनीकों में महत्वपूर्ण चुनौतियों को संबोधित करते हुए ग्राफ-आधारित मशीन लर्निंग को आगे बढ़ाने में योगदान करता है।

Contents

CERTIFICATE	i
ACKNOWLEDGEMENTS	ii
ABSTRACT	iii
LIST OF FIGURES	xi
LIST OF TABLES	xvii
ABBREVIATIONS	xviii
1 Introduction	1
1.1 Chapter Schematization	4
1.1.1 Featured Graph Coarsening and Its Variants	4
1.1.2 Sparse Featured Graph Coarsening for Enhancing Graph-Based Machine Learning	5
1.1.3 Convex Formulation of FGC algorithm	5
1.1.4 Structured Graph Coarsening	6
1.1.5 Coarsened Graph Learning via Raw Data Only	6
1.1.6 Publication	6
2 Background of Graph and Graph Coarsening	7
2.1 Graph	7
2.2 Graph Learning from Data	8
2.2.1 Similarity Function Based Methods for Graph Learning	8
2.2.2 Model Based Methods for Graph Learning	9
2.2.3 Smooth Signal Based Methods for Graph Learning	10
2.3 Graph Dimensionality Reduction Techniques	11
2.3.1 Kron Graph Reduction	11
2.3.2 Heavy Edge Matching Algorithm for Graph Reduction	12
2.3.3 Local Variation Graph Coarsening	13
2.3.4 Graph Condensation	13
2.3.5 Differential Pooling	14
2.3.6 Faster Graph Embeddings via Coarsening	14
2.3.7 3D Hodge Decompositions	15
2.4 Summation	15
3 Featured Graph Coarsening and its Variants	16
3.1 Introduction	16

3.1.1	Graph Coarsening:	17
3.1.2	Lifted Laplacian(Θ_{lift})	19
3.1.3	Preserving properties of \mathcal{G} in \mathcal{G}_c	19
3.1.4	Graph Learning from Data	21
3.1.5	A General Framework for Graph Coarsening with Features	22
3.1.6	Graph Coarsening with Feature Dimensionality Reduction	23
3.1.7	Some properties of $C^\top \Theta C$ matrix	24
3.1.8	The $\ell_{1,2}^2$ norm regularizer for balanced mapping	27
3.1.9	Block Majorization-Minimization Framework	27
3.1.10	Majorization Function for \mathbf{L} -smooth and Differentiable Function	28
3.2	Proposed Featured Graph Coarsening (FGC) Algorithm	29
3.2.1	Update of C	30
3.2.2	Update of \tilde{X}	34
3.2.3	Similarity Guarantee	36
3.2.4	Interpretation of the proposed formulation and the FGC Algorithm	38
3.3	Proposed Featured Graph Coarsening with Feature Dimensionality Reduction(FGCR)	40
3.3.1	Update of C	41
3.3.2	Update of W	42
3.3.3	Update of H	42
3.4	Connection of graph coarsening with clustering and community detection	44
3.5	Experiments	45
3.5.1	Performance Evaluation for the FGC algorithm	47
3.5.2	Performance Evaluation for the FGCR Algorithm	50
3.5.3	Graph Coarsening under Adversarial Attack	51
3.5.4	Application of FGC in Classification	53
3.5.5	Effect of Hyperparameters	55
3.5.6	Affect of features on Coarsening: Toy Example	55
3.6	Summation	57
4	Sparse Featured Graph Coarsening for Enhancing Graph-Based Machine Learning	59
4.1	Introduction	59
4.1.1	Graph Neural Networks	61
4.2	Sparse Featured Graph Coarsening(FGCS)	62
4.2.1	Update of C	62
4.2.2	Update of \tilde{X}	63
4.3	FGCS for Scaling GNN: Node Classification and Graph Classification	64
4.3.1	Node Classification	64
4.3.2	Graph Classification	65
4.4	Experiments	66
4.4.1	Node Classification	69
4.4.2	Generalizability of FGCS algorithm	70
4.4.3	Graph Classification	71
4.4.4	FGCS with Adversarial attack	72
4.4.5	FGCS with missing sampled dataset	72
4.4.6	Visualization of FGCS	73
4.4.7	Time complexity	73

4.5	Summation	74
5	Convex Framework of Featured Graph Coarsening	79
5.1	Introduction	79
5.2	Proposed Formulation	79
5.3	Experiments	83
5.3.1	Node Classification	83
5.3.2	Time complexity	84
5.3.3	Ablation Study	85
5.4	Summation	85
6	Structured Graph Coarsening	87
6.1	Introduction	87
6.1.1	Graph Coarsening for Scaling up GNN	90
6.2	Proposed Framework For Structured Graph Coarsening	92
6.3	Sparse and Scale Free Coarsened Graph Learning	94
6.4	Proposed Multi-Component & Bipartite Coarsened graph learning	96
6.4.1	Multi-component Coarsened Graph Learning(MGC)	96
6.4.2	Bipartite Graph Coarsening (BI-GC)	98
6.5	Two Stage-Structured Coarsened Graph Learning	102
6.6	Experiments	104
6.6.1	Node Classification	104
6.7	Summation	108
7	Coarsened Graph Learning via Raw Data Only	110
7.1	Introduction	110
7.2	Proposed Formulation	111
7.2.1	Graph Laplacian and Adjoint Operator	111
7.3	Coarsened Graph Learning via Orthogonality Constraint(CGL1)	114
7.3.1	Update of \mathbf{w}	115
7.3.2	Update of \mathbf{P}	116
7.3.3	Update of $\tilde{\mathbf{X}}$	117
7.4	Coarsened Graph Learning via $\ell_{1,2}$ regularizer (CGL2)	117
7.4.1	Update of \mathbf{w}	118
7.4.2	Update of \mathbf{C}	118
7.4.3	Update of $\tilde{\mathbf{X}}$	118
7.5	Experiments	119
7.5.1	Node Classification	119
7.5.2	Generalizability of CGL algorithms	121
7.5.3	Graph Classification	122
7.6	Summation	122
8	Conclusion and Future Work	123
8.1	Conclusion	123
8.2	Future Work	124
8.2.1	Active Learning for Labelled Graph Coarsening and Online Prediction	125
8.2.2	Semi-supervised Coarsened Graph Learning	126

BIBLIOGRAPHY	127
LIST OF PAPERS BASED ON THESIS	137
TECHNICAL BIOGRAPHY OF AUTHOR	138

List of Figures

2.1	Graph Reduction Techniques	11
3.1	Toy example: graph coarsening.	18
3.2	Some possible realisations of the loading matrix for the toy example: $C1, C2, C3$, and $C4$ are not balanced mapping, while $C5$ is an example of balanced mapping.	40
3.3	This figure plots the top-100 eigenvalues of the coarsened Laplacian matrix against the original Laplacian matrix for the Cora, Citeseer, Polblogs, and ACM data sets with coarsening ratio $r=0.3$. The eigenvalues for the coarsened matrix obtained by the FGC algorithm are almost similar to the original graph Laplacian matrix, highlighting that FGC is superior in preserving the spectral properties in the coarsened graph matrix in comparison to the existing state-of-the-art.	49
3.4	This figure plots the ϵ values for a variety of real and synthetic data sets. The ϵ values lying between $(0, 1)$ indicate that the coarsened graph G_c learned by the proposed FGC method and \mathcal{G} are similar.	50
3.5	This figure plots the heat maps of the loading matrices $C^T C$ obtained by FGC algorithm for Cora, Citeseer, Polblogs and ACM data sets for the coarsening ratio $r = 0.01$. Even for the extreme coarsening, where the size of the graph is reduced by 100, the $C^T C$ is almost diagonal, which indicates that the C matrix is also almost orthogonal. For moderate coarsening ($r = 0.3, 0.5$) we are observing C to be perfectly orthogonal. The strength of the values of the diagonal entries signifies the number of nodes from the set V mapped to a super-node. As indicated from the vertical color bar the mapping is balanced, such that not all or the majority of nodes are mapped to one single supernode in the coarsened graph. These observations also validate that the $\ C^T\ _{1,2}^2$ norm penalty is effective in enforcing desired grouped sparsity structure. Finally, the good results for the experiments with extreme coarsening also suggest that the proposed method can be utilized for doing clustering and stochastic block model identification.	51
3.6	This figure shows the loss curves of FGC for (a) Cora, (b) Citeseer, (c) Polblogs and (d) ACM on different coarsening ratios.	51
3.7	This figure plots the top-100 eigenvalues of the coarsened graph obtained by FGCR against the original graph for Cora, Citeseer, Polblogs, and ACM data sets.	52
3.8	Top-100 eigenvalues plots of original graph and coarsened graph obtained by FGC(proposed) algorithm and it is evident that FGC outperforms state of the art algorithm on 10% and 5% perturbation attack.	54

3.9	This figure evaluates the classification performance of the FGC algorithm on the classic Zachary’s karate club data set [1] into 2 classes: (a) Ground truth, (b) Graclus[2], (c) spectral clustering ratio cut [3] (d) spectral clustering normalized cut [3] (e) LVN [4] and (f) FGC (Proposed). Orange nodes indicate misclassified points, FGC demonstrates a better performance, it resulted in only 1 misclassified point, while the number of misclassified points for (b), (c), (d), and (e) are 11, 7, 2 and 5, respectively.	55
3.10	This figure evaluates the classification performance of the FGC algorithm on the classic Zachary’s karate club data set [1] into 4 classes: (a) Ground truth, (b) spectral clustering ratio cut [3] (c) spectral clustering normalized cut [3] (d) LVN [4] and (e) FGC (Proposed). It is evident that FGC demonstrates a better performance, it resulted in 4 misclassified point, while the number of misclassified points for (b), (c) and (d) are 7, 11, and 24 respectively.	56
3.11	Fig.(a-e) shows the eigenvalue plot of original graph and coarsened graph obtained by FGC using hyperparameters $\alpha = 500$, $\lambda = 1000$ and varying γ in between (200-5000). It is evident that for a moderate γ , i.e., between 200 to 2000, the REE is almost similar, and our algorithm is consistent.	57
3.12	Fig.(a-e) shows the eigen value plot of original graph and coarsened graph obtained by FGC using hyperparameters $\lambda = 1000$, $\gamma = 500$ and varying α in between 200-5000). It is evident that for a moderate α , i.e., between 200 to 2000, the REE is almost similar, and our algorithm is consistent.	57
3.13	Fig.(a-e) shows the eigenvalue plot of original graph and coarsened graph obtained by FGC using hyperparameters $\alpha = 500$, $\gamma = 1000$ and varying λ in between (200-5000). It is evident that for a moderate λ , i.e., between 200 to 2000, the REE is almost similar, and our algorithm is consistent.	57
3.14	FGC on toy example having feature matrix $X1$	58
3.15	FGC on toy example having feature matrix $X2$	58
4.1	Graph Coarsening with FGC and FGCS.	60
4.2	This figure depicts the GNN architecture used for performing various downstream tasks, such as node classification, graph classification, and edge prediction.	61
4.3	The figure shows an illustration of node classification of original graph \mathcal{G} using coarsened graph \mathcal{G}_c . In step (1), we have coarsened a large original graph \mathcal{G} into \mathcal{G}_c and learned a loading matrix $C = P^\dagger$. In step (2), we have trained GNN and learned the weight matrix using coarsened feature matrix $X' = PX$ and coarsened labels $Y' = \text{argmax}(PY)$. In step (3), we have predicted labels to each node of the original graph using the trained GNN model which was trained on coarsened graph.	65
4.4	The figure shows an illustration of graph classification of original graph \mathcal{G} using coarsened graph \mathcal{G}_c . In step (1), we have coarsened original graphs $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_g\}$ into coarsened graphs $\mathcal{G}_c = \{\mathcal{G}_{c1}, \mathcal{G}_{c2}, \dots, \mathcal{G}_{cg}\}$ and learned loading matrices $C_i = P_i^\dagger \forall i \in \{1, 2, \dots, g\}$ corresponding to each graph and g is the total number of graphs in each dataset. In step (2), a 2-layer multi-layer perceptron (MLP) based classification model is trained on the coarsened graphs. Next, we predict the class of graphs in test sets.	66

4.5	This figure visualizes the coarsened graph learned from FGC [5] and proposed FGCS1 for a coarsening ratio of 0.3 and 0.5 respectively. The original graph is a stochastic block model graph of 100 nodes. It is evident that the coarsened graph learned from FGCS1 is sparse as compared to FGC.	76
4.6	Figures (a)-(d) show the visualization of the learned coarsened graph using proposed FGCS1 algorithm. It can be observed that the learned coarsened graph is sparse, which indicates that it is capable of performing well in downstream tasks such as node classification and graph classification.	77
4.7	Figure 4.7a and 4.7d show the distribution of the original labels within the Cornell and Texas datasets, respectively. Figures 4.7b and 4.7e illustrate the confusion matrix for labels assigned to the coarsened graph during GNN training. Figures 4.7c and 4.7f display the confusion matrix for labels assigned by the GNN to the original graph during testing.	78
5.1	The figure illustrates the time required for graph coarsening using both the proposed CFGC algorithm and the FGC [5] algorithm on a synthetic Erdos Renyi graph having number of nodes from 1000 to 50,000 for a coarsening ratio of 0.1. Clearly, the CFGC algorithm is more than two times faster than the FGC algorithm.	85
6.1	This figure illustrates that multiple structures can be imposed on the resulting coarsened graph for a given original graph \mathcal{G}	90
6.2	The figure shows an illustration of the proposed pipeline for node classification of an original graph \mathcal{G} using coarsened graph \mathcal{G}_c	92
6.3	This figure depicts the time required in sec. to learn the coarsened graph with various structures using the set of proposed algorithms, specifically SCG1, SCG2, SCG3, BI-GC, MGC, and TGC, when applied to a stochastic block model dataset. The objective of this figure is to present an in-depth analysis of the time complexity of algorithms 7 and 9, as discussed earlier.	108
6.4	The figures presented here depict the ϕ matrices of different graph representations: the bipartite coarsened graph, and the multi-component coarsened graph. Additionally, a histogram showcases the sparsity levels within each row of the coarsened graphs, focusing on data from the Citeseer dataset. Notably, these visualizations illustrate a critical point: the sparser the heat map of the coarsened graph, the more informative it becomes. To be specific, the node classification accuracy achieved using the bipartite coarsened graph is 77.11%. Meanwhile, the multi-component coarsened graph yields a node classification accuracy of 74.68% for a coarsening ratio of 0.3.	109

List of Tables

3.1	This table summarizes the REE and DE values obtained by FGC (proposed), LVN, LVE, Kron and HEM on different coarsening ratios (r) for standard real graph data sets. It is evident that FGC (proposed) outperforms state-of-the-art methods significantly.	47
3.2	This table summarizes the HE and RE values obtained by FGC (proposed), LVN and LVE on different coarsening ratios (r) for standard real graph data sets. It is evident that FGC (proposed) outperforms state-of-the-art methods significantly.	48
3.3	This table summarizes the REE values obtained by FGC (proposed), GOREN(HEM), GOREN(LVN), and GOREN(LVE) on different coarsening ratios (r) for real graph data sets. It is evident that FGC (proposed) outperforms state-of-the-art methods significantly.	48
3.4	REE, HE, and RE values for FGCR (proposed) algorithm on Cora, Citeseer, Polblogs, and ACM.	52
3.5	REE and DE results for Cora, Citeseer, and ACM for 10% and 5% perturbation by FGC (proposed), LVN, and LVE.	53
4.1	Homophilic datasets used in node classification.	67
4.2	Heterophilic datasets used in node classification.	67
4.3	Datasets used in graph classification task where Avg_p and Avg_d means an average number of nodes and average node degree in the graph. . . .	67

4.4	The table summarizes the node classification accuracy on both small and large datasets for the proposed FGCS1 and FGCS2 algorithms against the GCOND and SCAL. It is evident that FGCS1 and FGCS2 outperform state-of-the-art methods. However, we have compared only with GCOND and SCAL as these are the most recent techniques for graph node classification using the coarsened graph. Moreover, the performance of FGCS1 and FGCS2 are comparable. Also, GCOND and SCAL codes show memory error on a 32GB RAM GPU (NVIDIA P100) for the NELL dataset.	68
4.5	This table summarizes node classification performance on heterophilic graphs. It is evident that FGCS1 outperforms state-of-the-art algorithms. Moreover, the node classification accuracy obtained by FGCS1 is better than node classification accuracy of GCN trained on the whole dataset. Also, When applying the GCOND algorithm on heterophilic graphs using the code provided by [6], we faced some run-time errors for the Texas, Squirrel and Deezereurope datasets and could not compute their results.	70
4.6	Node classification performance using FGCS1 algorithm on different GNN architectures for a coarsening ratio of $r=0.3$. It is evident that FGCS1 is compatible with different widely used GNN architecture like GCN [7], APPNP [8], and GAT [9].	71
4.7	This table shows the graph classification results, i.e., graph classification accuracy (ACC) and combined time for coarsening and training MLP model in seconds as (ACC, τ) , using proposed FGCS1, OT [10], and DosCond [11]. It is evident that the FGCS1 algorithm outperforms in both accuracy and time complexity as compared to the state-of-the-art algorithms.	71
4.8	This table shows the node classification accuracy performed on the perturbed datasets having perturbation rate (pr) of (i) 10% (ii) 20%. This perturbing is done by adding extra fake edges to the dataset. It is evident that the effect of perturbation doesn't effect both algorithms, FGCS1 and FGCS2, in the node classification accuracy significantly.	72

4.9	The table presents the accuracy of node classification achieved by FGCS1 and FGCS2 algorithms on a dataset with missing samples. The missing rate(mr) of the dataset is (i) 10% (ii) 20%, and the missing values are generated randomly by setting the values of X_{ij} to zero. The results show that setting samples to zero does not significantly affect the accuracy of node classification for both FGCS1 and FGCS2 algorithms.	73
4.10	This table summarizes the time (τ) in sec. required to perform coarsening and node classification, both for a coarsening ratio of 0.05. It is evident that the proposed FGCS1 is very fast as compared to GCOND.	74
4.11	This table shows the comparison between the original heterophilic graph and coarsened graph. It is evident that the learned coarsened graph has less number of nodes and edges as compared to the original graph. Furthermore, it is evident that as the number of nodes is reducing while coarsening, the corresponding number of edges also reduces.	74
5.1	Datasets used in node classification.	83
5.2	The table summarizes the node classification accuracy on real benchmark datasets for the proposed CFGC algorithm against the GCOND, SCAL, FGC, and using whole dataset(Wh.da.). We have taken coarsening ratios $r = 0.5, 0.3,$ and 0.1 for small dataset and $r = 0.05, 0.03,$ and 0.01 for large dataset. It is evident that the proposed CFGC outperforms the state-of-the-art methods significantly for high coarsening ratios and is comparable for low coarsening ratios.	84
5.3	This table summarizes the time (τ) in sec. required to perform coarsening and node classification for a coarsening ratio of 0.05. It is evident that the proposed CFGC is faster than state-of-the-art algorithms. Moreover, the time required to perform coarsening and node classification using the proposed methods is less than that required to perform node classification using the given original graph.	85
5.4	The table presents a comparison between the proposed CFGC and its ablated counterpart on the Cora and Citeseer datasets. Evidently, omitting the convex constraint $\ \tilde{X} - C^T X\ _F^2$ results in a significant reduction in accuracy.	86

6.1	Datasets used in node classification.	104
6.2	The table summarizes the node classification accuracy on real benchmark datasets for the proposed SCG1, MGC and BI-GC algorithms against the GCOND and SCAL. For small datasets, we have taken coarsening ratios $r = 0.5, 0.3,$ and 0.1 ; for large datasets, we have taken $r = 0.1, 0.05$ and 0.03 . It is evident that proposed MGC and BI-GC outperform state-of-the-art methods by a significant amount. Furthermore, across a wide range of datasets, the MGC algorithm consistently outperforms or demonstrates comparable performance to the BI-GC algorithm.	105
6.3	The table summarizes the node classification accuracy on real datasets for the proposed structured graph coarsening algorithms SCG1, SCG2 and SCG3 for a coarsening ratio of 0.1 . Our results unequivocally demonstrate that the SCG2 and SCG3 algorithms consistently outperform the SCG1 algorithm. This remarkable improvement can be attributed to the incorporation of non-convex regularizers, specifically MCP (SCG2) and SCAD (SCG3), in the learning process of creating a sparse coarsened graph. These non-convex regularizers surpass the performance of the convex regularizer used in SCG1. This enhanced sparsity in the coarsened graph, brought about by the non-convex regularizers, yields a significant boost in the accuracy of node classification.	106
6.4	The table summarizes node classification accuracy achieved on real benchmark datasets using both the proposed SCG1 and the two-stage tree coarsened graph learning (TGC) algorithm. These evaluations were conducted across different coarsening ratios, specifically, $0.1, 0.3, 0.5,$ and 0.7 . It is noteworthy that when dealing with higher coarsening ratios of 0.3 and 0.7 , TGC demonstrates a substantial performance advantage over the SCG1 algorithm. In contrast, for coarsening ratios of 0.3 and 0.1 , TGC show comparable performance to SCG1 algorithm.	106

6.5	The table summarizes the node classification accuracy on real datasets for the proposed multi-component graph coarsening (MGC) algorithm for different values of component n and coarsening ratio 0.1. It is evident that enforcing multi-component structures on the coarsened graph for different values of component n improves the accuracy.	107
6.6	This table summarizes the time (τ) in sec. required to perform coarsening and node classification for a coarsening ratio of 0.05. It is evident that the proposed SCG1, SCG2, SCG3, MGC, and BI-GC are faster than state-of-the-art algorithms. Moreover, the time required to perform coarsening and node classification using the proposed methods is less than that required to perform node classification using the given original graph.	107
6.7	The table summarizes node classification accuracy for both single-stage and two-stage structured coarsened graph learning methods. Notably, the single-stage structured coarsened graph learning algorithms consistently outperform their two-stage counterparts. In the two-stage multicomponent coarsened graph learning approaches, which is denoted as TSM-1 and TSM-2, the initial stage entails the construction of the graph using the SCG algorithm. Subsequently, in the second stage, the multi-component structure is imposed through the Louvain method(TSM-1)[12] and the MGAE (Multi-Graph Autoencoder)(TSM-2) [13] technique. On the other hand, TSBI represents the two-stage bipartite graph learning approach. The coarsened graph is constructed using the SCG algorithm in the first stage. Then, to enforce the bipartite structure, the SGL [14](Sparse Graph Learning) algorithm is employed.	107
6.8	Node classification accuracy (%) obtained using different GNN structures like GCN, GAT, and APPNP on different datasets using the proposed SCG1 algorithm for a coarsening ratio of 0.1. It is evident that the proposed SCG method is suitable for all GNN architecture.	108
7.1	Node Classification Datasets	120
7.2	Graph Classification Datasets	120

7.3	This table shows the node classification performance of the proposed algorithms CGL1 and CGL2 compared to state-of-the-art methods GCOND, SCAL, and FGC on a Homophilic Graph. It is evident that for coarsening, we have utilized only the feature matrix X of the graph, and yet our framework outperforms the existing state-of-the-art graph coarsening methods, which consider both the Laplacian and the feature matrix for coarsening.	120
7.4	This table shows the node classification performance on the heterophilic datasets. It is evident that for coarsening, we have utilized only the feature matrix X of the graph, and yet our framework outperforms or comparable against the existing state-of-the-art graph coarsening methods, which consider both the Laplacian and the feature matrix for coarsening. . . .	121
7.5	Node classification performance using CGL1 algorithm on different GNN architectures for a coarsening ratio of $r=0.05$. It is evident that proposed CGL1 is compatible with different widely used GNN architecture like GCN [7], APPNP [8], and GAT [9].	121
7.6	This table depicts the graph classification performance on real datasets. It is evident that the proposed algorithm outperforms the state of the art methods.	122

ABBREVIATIONS

FGC	Featured Graph Coarsening
FGCR	Featured Graph coarsening with Feature Dimensionality Reduction
BSUM	Block Successive Upper Bound Minimization
GNN	Graph Neural Networks
FGCS	Sparse Featured Graph coarsening
CFGC	Convex Framework of Featured Graph Coarsening
REE	Relative Eigen Error
DE	Dirichlet Energy
HE	Hyperbolic Error
SCG	Sparse Coarsened Graph
SFCG	Scale Free Coarsened Graph
MGC	Multicomponent Coarsened Graph
TSBI	Two Stage Bipartite Graph
TSM	Two Stage Multi-Component
GCN	Graph Convolution Network
GAT	Graph Attention Network
APPNP	Approximation of Personalized Propagation of Neural Predictions
CGL	Coarsened Graph Learning
LASSO	Least Absolute Shrinkage and Selection Operator
MLP	Multi-Layer Perceptron