UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Graph Representation Learning with Edge Heterophily Discriminating

Diploma Thesis

Syrago Akrivousi

Supervisor: Dimitrios Rafailidis

September 2024

UNIVERSITY OF THESSALY

SCHOOL OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Graph Representation Learning with Edge Heterophily Discriminating

Diploma Thesis

Syrago Akrivousi

Supervisor: Dimitrios Rafailidis

September 2024

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εκμάθηση Αναπαράστασης Γράφων με Διαχωρισμό Ετερογενών Ακμών

Διπλωματική Εργασία

Συραγώ Ακριβούση

Επιβλέπων/πουσα: Δημήτριος Ραφαηλίδης

Σεπτέμβριος 2024

Approved by the Examination Committee:

Supervisor **Dimitrios Rafailidis**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **Dimitrios Katsaros**
Associate Professor, Department of Electrical and Computer Engineering, University of Thessaly

Member **George Thanos**
Laboratory Teaching Staff, Department of Electrical and Computer Engineering, University of Thessaly

Acknowledgements

First and foremost, I would like to thank my supervisor, Prof. Dimitrios Rafailidis, whose support and guidance were invaluable at every phase of my research. His expertise and unwavering belief were instrumental in the development of this thesis.

I would like to express my appreciation to the members of my thesis committee, Prof. Dimitrios Katsaros and Prof. George Thanos, for their time and effort in evaluating my work.

Additionally, I would like to thank Professor Emeritus, Mr. Elias Houstis, for his guidance and valuable advice, which inspired me throughout my academic studies and earned my utmost respect.

Lastly, I owe a profound debt of gratitude to my family and close friends for their unreserved support, understanding and constant encouragement throughout my academic years.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

«Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I also declare that the results of the work have not been used to obtain another degree. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism».

The declarant

Syrago Akrivousi

Diploma Thesis

Graph Representation Learning with Edge Heterophily Discriminating

Syrago Akrivousi

Abstract

Graph Representation Learning (GRL) has gained a lot of attention from researchers. Modern GRL methods have been observed to face challenges on graphs with heterophilic edges, which exist in real data. However, in order to build an effective model for this purpose, complex neural architectures are designed, which require a large number of parameters for training, making them inaccessible for use in real-life applications where resources are limited. In this thesis, a method for graph representation learning with edge discrimination, called GREET, is proposed to address this issue. Moreover, a knowledge distillation strategy is proposed to overcome the weaknesses of existing models due to the volume of parameters while preserving the performance and decision-making capabilities of the model. In particular, first a teacher model is trained on offline data, and then a distillation loss function is applied to a student model in order to transfer the knowledge from the teacher to a smaller model, which requires fewer parameters. According to the experiments that are performed on six datasets, two homophilic and four heterophilic, they demonstrate that the GREET model performs better than the state-of-the-art GRL models, especially on heterophilic graphs. The implementation of the knowledge distillation strategy, based on the experimental results, proves to be an effective approach, as it remarkably reduces the computational cost while preserving the model performance in great measure. The proposed approach in this study aims to contribute to the use of complicated neural architectures in applications with limited resources to handle graphs with both homophilic and heterophilic edges since real-world data contain both types of edges.

Keywords:

Graph representation learning, Heterophic graphs, Knowledge distillation, Edge discrimination, Model compression

Διπλωματική Εργασία

Εκμάθηση Αναπαράστασης Γράφων με Διαχωρισμό Ετερογενών Ακμών

Συραγώ Ακριβούση

Περίληψη

Η μάθηση αναπαράστασης γράφων έχει κερδίσει μεγάλη προσοχή από τους ερευνητές. Έχει παρατηρηθεί ότι οι σύγχρονες μέθοδοι αντιμετωπίζουν προκλήσεις σε γράφους με ετερογενείς ακμές, οι οποίες υπάρχουν στα πραγματικά δεδομένα. Ωστόσο, για να δημιουργηθεί ένα αποτελεσματικό μοντέλο για το σκοπό αυτό, σχεδιάζονται πολύπλοκες νευρωνικές αρχιτεκτονικές, οι οποίες απαιτούν μεγάλο αριθμό παραμέτρων κατά την εκπαίδευση, γεγονός που τις καθιστά απρόσιτες για χρήση σε πραγματικές εφαρμογές όπου οι πόροι είναι περιορισμένοι. Στην παρούσα διπλωματική εργασία, προτείνεται μια μέθοδος εκμάθησης αναπαράστασης γράφων με διάκριση ακμών, η οποία ονομάζεται GREET, για την αντιμετώπιση αυτού του ζητήματος. Επιπλέον, προτείνεται μια στρατηγική απόσταξης γνώσης για να ξεπεραστούν οι αδυναμίες των υφιστάμενων μοντέλων εξαιτίας του όγκου των παραμέτρων, διατηρώντας παράλληλα τις δυνατότητες λήψης αποφάσεων του μοντέλου. Συγκεκριμένα, πρώτα εκπαιδεύεται ένα μοντέλο δασκάλου σε δεδομένα εκτός σύνδεσης και στη συνέχεια εφαρμόζεται μια συνάρτηση απώλειας απόσταξης σε ένα μοντέλο μαθητή, προκειμένου να μεταφερθεί η γνώση από τον δάσκαλο σε ένα μικρότερο μοντέλο, που απαιτεί λιγότερες παραμέτρους. Σύμφωνα με τα πειράματα που πραγματοποιήθηκαν σε έξι σύνολα δεδομένων, δύο ομογενή και τέσσερα ετερογενή, αποδεικνύεται ότι το GREET έχει καλύτερη απόδοση από τα σύγχρονα μοντέλα, ιδίως σε ετερονενείς γράφους. Η εφαρμογή της στρατηγικής απόσταξης γνώσης, με βάση τα πειραματικά αποτελέσματα, αποδεικνύεται αποτελεσματική προσέγγιση, καθώς μειώνει σημαντικά το υπολογιστικό κόστος, διατηρώντας παράλληλα σε μεγάλο βαθμό την απόδοση του μοντέλου. Στην παρούσα μελέτη, η προτεινόμενη προσέγγιση συμβάλει στη χρήση πολύπλοκων νευρωνικών αρχιτεκτονικών σε εφαρμογές με περιορισμένους πόρους για τον χειρισμό γράφων με ομογενείς και ετερογενείς ακμές, εφόσον τα δεδομένα του πραγματικού κόσμου περιέχουν και τους δύο τύπους ακμών.

Λέξεις-κλειδιά:

Εκμάθηση αναπαράστασης γράφου, Ετερόφιλοι γράφοι, Απόσταξη γνώσης, Διαχωρισμός ακμών, Συμπίεση μοντέλου

Table of contents

Acknowledgements ix

Abstract xii

Περίληψη xiii

Table of contents xv

List of figures xvii

List of tables xix

Abbreviations xxi

1 Introduction 1

2 Related Work 5

2.1 Introduction to Graph Representation Learning 5

2.1.1 Graph Neural Networks 5

2.1.2 Types of Nodes in a Node Classification in GNNs 6

2.1.3 Graph Convolution Networks 6

2.1.4 Graph Representation Learning 7

2.1.5 Heterophilic Graphs: Importance of Developing GNNs 8

2.2 Knowledge Distillation 8

3 Proposed Model 11

3.1 The GREET Model 11

3.1.1 Edge Discriminating Module 12

3.1.2 Dual-Channel Encoding Module 14
3.1.3 Model Training 15
3.2 Knowledge Distillation Strategy 17
3.2.1 Variations of the Model 18

4 Experiments **19**
4.1 Datasets and Evaluation Protocol 19
4.2 Models 20
4.2.1 Examined Models 20
4.2.2 Experimental Models 21
4.3 Performance Evaluation 22
4.4 Parameter Tuning 25

5 Conclusion **27**

Bibliography **29**

List of figures

1.1 Examples of homophilic and heterophilic graphs (Left (a): a citation network; Right(b): an online transaction network) [1]. 2

3.1 The general structural design of the GREET model [2]. 12

4.1 Impact of hyperparameter γ on the prediction accuracy of student model GREET-KD*. 25

4.2 Impact of hyperparameter γ on the prediction accuracy of student model GREET-KD**. 26

List of tables

4.1 Statistics of benchmarking datasets 20

4.2 Results in terms of classification accuracies (in percent ± standard deviation) on homophilic benchmarks. 22

4.3 Results in terms of classification accuracies (in percent ± standard deviation) on heterophilic benchmarks. 23

4.4 Model parameters on homophilic datasets (in millions), when comparing the proposed GREET student model with the other non-distillation strategies. . 24

4.5 Model parameters on heterophilic datasets (in millions), when comparing the proposed GREET student model with the other non-distillation strategies. . 24

Abbreviations

GNN	Graph Neural Network
GCN	Graph Convolutional Network
GRL	Graph Representation Learning
UGRL	Unsupervised Graph Representation Learning
KD	Knowledge Distillation

Chapter 1

Introduction

Researchers have increasingly used graphs as a fundamental format for representing and analyzing complex data in recent years. Many real-life applications, such as social networks [3], biological networks [4], and recommender systems [5], commonly use them. Recently, graph neural networks (GNNs) have achieved remarkable progress and have become widespread in tasks such as link prediction, node classification, graph classification, and graph representation learning [6, 7]. The rapid evolution of deep learning and the flexible structure of graph data have played a key role in this development. GNNs exploit the topology of the graph to be able to extract important patterns and features from it. In this way, GNNs handle graph-based data with more accuracy and efficiency [8, 9]. However, despite the large number of GNN architectures designed, a critical challenge in this area is processing heterophilic data, that is, graphs of nodes with dissimilar attributes or with different labels that are more likely to connect.

Conventional GNNs mainly use homophilic data, which are graphs where nodes with similar labels are likely to be connected, as shown in the Figure 1.1 [1]. Researchers have acknowledged the drawbacks of GNNs, which usually make them inefficient on heterophilic data due to their natural bias towards homophilic structures [10]. This limitation has led to the development of advanced GNN models, which have robust performance in various data situations. The development of GNNs specifically for heterophilic graphs marks a significant improvement in the field of graph representation learning.

Furthermore, traditional GNNs face some additional challenges when applied to unsupervised graphs, as node labels are not available to guide the learning process. The labels provide direct evidence, so edge discrimination can be incorporated into supervised node classifica-

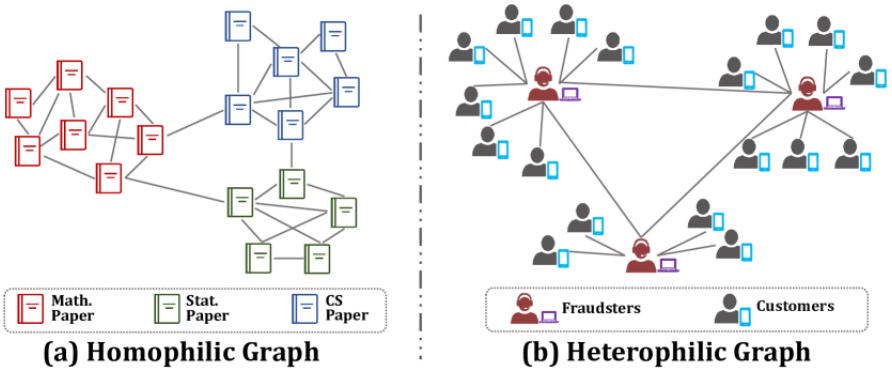

Figure 1.1: Examples of homophilic and heterophilic graphs (Left (a): a citation network; Right(b): an online transaction network) [1].

tion models [11]. Nevertheless, it is difficult to distinguish edge types based exclusively on node characteristics and graph structure in unsupervised scenarios, especially for edges connecting nodes with moderate similarities. Consequently, novel GNN models currently being designed for heterophilic and unsupervised graphs aim to address these challenges, enabling more accurate and efficient representation learning in a more extensive range of graph-based applications.

These advanced GNN models for heterophilic graphs are highly complicated and computationally expensive, which makes them challenging to implement in real-world environments with limited resources. Knowledge distillation has emerged as a promising solution to address this problem, as a simpler student model is trained to replicate the performance of a more computationally complicated model, which is the teacher. This technique has been successfully applied to a variety of domains, enabling smaller models in Natural Language Processing [12], efficient models in Speech Recognition [13] and accurate models in Medical Diagnosis [14]. The transfer of knowledge from the teacher model to the student model dramatically reduces computational resource requirements and achieves high performance. This methodology provides multiple advantages for practical implementations, such as the ability of robust GNNs to be integrated into devices with limited computational and power capabilities.

The contribution of the thesis diploma is summarised as follows:

- We study a number of state-of-the-art methods for representing heterophilic graphs in an unsupervised scenario and consider the GREET model [2], which integrates a module for edge discrimination with a representation learning module.
- In addition, we propose a knowledge distillation framework for the reduction of training parameters in the domain of graph representation learning. In doing so, it facilitates knowledge transfer between a large model and a variant of it, which has lessened computational requirements.

The thesis is structured in five chapters, starting with the introduction of the thesis, presented in chapter 1. In chapter 2, the material of related works is reviewed, while chapter 3 presents the GREET model and discusses the importance of developing an efficient knowledge distillation framework in the domain of graph representation learning to reduce the computational requirements of the model. In chapter 4, the models examined in the thesis, the datasets used, and the experimental setup adopted for their evaluation with the results produced are presented. Finally, this thesis ends with the conclusions in Chapter 5.

Chapter 2

Related Work

2.1 Introduction to Graph Representation Learning

The field of representation learning for heterophilic graphs has become increasingly significant for dealing with complex real-world challenges such as social network analysis, recommendation systems, and fraud detection [15]. We can find interesting patterns and enhance decision-making procedures in a variety of applications by utilizing the inherent connections and heterogeneity inside these graphs.

2.1.1 Graph Neural Networks

A Graph Neural Network (GNN) is a specialized deep learning model designed to work with graph-structured data, where nodes represent entities and edges represent relationships between them. GNNs generate embeddings for each node by iteratively combining the embeddings of its neighbors, effectively capturing the node's local and global context within the graph. This process enables the model to learn how a node's status is influenced by its interactions with its neighbors. GNNs can be trained in various frameworks, including supervised learning with labeled data, semi-supervised learning with a mix of labeled and unlabeled data, and unsupervised learning without any labels, making them versatile for numerous applications like social network analysis, recommendation systems, and biological network modeling. Furthermore, GNNs are well suited for tasks such as node classification, link prediction, and graph classification.

2.1.2 Types of Nodes in a Node Classification in GNNs

In the classification of nodes in GNNs there are three categories of nodes, training nodes, transductive test nodes and inductive test nodes [16]. **Training nodes** are labeled nodes that are used during the training phase. They have their embeddings, which are computed at the last level of the GNN and are used to compute the loss function during training. **Transductive test nodes** are nodes whose labels are not available but their features are accessible during the training phase. They also have their embeddings computed by the GNN but do not contribute to the computation of the loss function. On the other hand, **inductive test nodes** are unseen nodes that the model encounters after training. They are totally excluded from the computation of the GNN embedding and the loss function during the training.

2.1.3 Graph Convolution Networks

Graph convolutional networks (GCNs) are a popular category of GNNs and generalize traditional convolutional neural networks to the graph domain [6]. GCNs have two types that can overlap: spectral-based and spatial-based methods.

1. Spectral-based GNNs: A spectral representation of the graphs is applied in spectral techniques. The methods are defined theoretically from the perspective of graph signal processing. The initial recognized study on spectral-based GNNs was published by Bruna et al.(2013) [17]. Subsequently, spectral-based GNNs have seen a rising number of advancements, extensions, and approaches. ChebyNet, as proposed by Defferrard et al. (2016) [18], is a spectral-free filter that eliminates the need for eigenvector analysis and achieves a large reduction in complexity. It approximates the spectral filter using Chebyshev polynomials. A similar technique is used by Cayley [19]. Until recently, a large number of spectral-based GNNs had emerged [20, 21].

2. Spatial-based GNNs: Based on the structure of the graph, spatial techniques define convolutions directly on it. The spatial-based convolution method aggregates nodes' surrounding information and is an extension of the traditional Euclidean convolution technique. Because of their ability to be transferred, confined nature, and reduced computational cost, these convolutions have proven to be quite appealing. The initial spatial techniques were developed using CNN [22] and random walks [23, 24, 25]. Following that, Gilmer et al. (2017) proposed the MPNN model [26], which solidified

the message-passing mechanism in spatial techniques to unify several variants. The GraphSAGE model [7] generates embeddings by uniformly sampling a fixed-size set of neighbors and aggregating their features instead of using the entire neighbor set. Xu, Hu et al. (2018) found that GNNs' message passing can't distinguish certain graph structures, so they created GIN [8], which improves representation by adjusting the central node's feature weight. More recently, there have been efforts to enhance the representational power of GNNs [27, 28].

2.1.4 Graph Representation Learning

Graph representation learning (GRL), also known as graph embedding, is a subfield of machine learning that focuses on embedding graph structures into continuous vector spaces, facilitating tasks such as node classification, link prediction, and graph clustering.

- **Conventional GRL** methods typically employ deterministic approaches to directly encode graph topology and node attributes into low-dimensional representations. Some examples of these methods use random walks to construct the nearest neighbors, such as Node2vec, LINE, and DeepWalk [29, 25, 24]. In addition, techniques like VGAE [30] and ARGA [31] leverage autoencoder-like design to maximize the agreement between neighboring nodes.

- **Contrastive GRL** methods emphasize learning representations by maximizing agreement between different views or augmentations of the same graph while discriminating them from other graphs. These approaches frequently rely on strategies for self-supervised learning. Examples are GraphCL [32], which investigates the implementation of contrastive learning in graph-level learning challenges, and GRACE [33], which generates augmented views by edge dropping and feature masking. Furthermore, contrastive methods are more robust and perform better in unsupervised scenarios.

2.1.5 Heterophilic Graphs: Importance of Developing GNNs

The graph learning community usually divides graphs into homophily and heterophily categories based on node labels in order to facilitate learning. Homophilic graphs are characterized by nodes that tend to connect with similar nodes, often resulting in clusters of nodes with similar features or labels. This is a common scenario in social networks, where people with similar interests or demographics form tight-knit communities. This type of network has been extensively studied, partly due to its straightforward nature. Conversely, heterophilic graphs have nodes that are more likely to form connections with other nodes that are not similar to them, resulting in a more varied collection of connections. This kind of graph is prevalent in scenarios such as recommendation systems or collaborative networks, where diverse interactions are beneficial. Nevertheless, learning about heterophilic graphs presents greater challenges. Numerous GNNs have been developed to improve performance in environments characterized by low homophily. Geom-GCN [34] introduces an innovative geometric aggregation method to address the issues of losing neighborhood structural information and lacking long-range dependencies. Furthermore, H2GCN [10] aggregates information from higher-order neighborhoods, separating ego and neighbor embeddings and combining intermediate representations. More recently, GGCN [35] presented a strong and more comprehensive method that tackles the differences in features and degrees between surrounding nodes by combining learned degree corrections and signed messages, thereby reducing the homophily assumption included in GCNs.

2.2 Knowledge Distillation

Knowledge distillation is a machine learning technique used to build effective models without appreciably compromising performance. It accomplishes this by training a compact model (student) to mimic the behavior of a larger, more complex model (teacher), hence addressing the need to deploy models in environments with limited resources [36, 37]. This technique transfers knowledge to the student model by utilizing the insights that the teacher model received during training. Knowledge distillation aims to maintain the teacher model's excellent performance while lowering computing requirements, making the student model appropriate for real-world scenarios where speed and resource efficiency are crucial [38, 39].

GNNs have benefited significantly from the application of knowledge distillation, which has helped to minimize their size and computing requirements. This method allows effective GNNs to be used in real-life situations without greatly sacrificing their functionality. For distillation on graphs, output logits [40], graph structures [41], and embeddings [42] are the three types of data that can be thought of as transferable knowledge.

The application of knowledge distillation is essential in graph-based models in the area of GRL to enable them to be used despite their high computational requirements. Large GNNs that are designed for this task achieve high accuracy, but they are unsuitable for real-time or resource-constrained applications due to their extensive processing requirements. By using knowledge distillation, a smaller and more efficient GNN can be trained to mimic the performance of a larger model. In this manner, the computational load is significantly reduced while preserving the production of high-quality graph representations, since it is affordable to use efficient GNNs in more practical applications.

A few efforts have been made to decrease the model sizes of the fundamental neural networks developed by GRL using knowledge distillation techniques. Ma and Mei [43] presented a multi-task knowledge distillation technique that improves GRL, especially in situations where training data is limited, by utilizing network-theory-based graph metrics as auxiliary tasks. More recently, Joshi et al. [44] proposed G-CRD, a unique contrastive learning method, to improve the performance of lightweight GNNs by aligning their node embeddings with those of more expressive teacher models, maintaining both global topology and latent interactions.

Chapter 3

Proposed Model

3.1 The GREET Model

The existence of two types of edges, homophilic and heterophilic, leads to some fundamental questions, such as whether it is possible for a model to discriminate among these two categories in an unsupervised process. The absence of labels is a significant difficulty in distinguishing edges. To this problem is added the need to combine edge discrimination with graph representation learning in an efficient way in an integrated model. In response to these challenges, Liu Y. et al. [2] suggested a innovative unsupervised method for Graph Representation learning with Edge hEterophily discriminaTing (GREET).

In this chapter, we cover the suggested GREET method in depth. As the model architecture is shown in Figure 3.1, GREET comprises two parts in particular: an edge discriminating unit that distinguishes between homophilic and heterophilic edges and a dual-channel representation learning unit that utilizes both types of edges to produce informative node representations. The edge discriminating unit implements an edge discriminator that is trained unsupervised using node characteristics and structural encodings to distinguish edges into homophilic and heterophilic edges. The dual-channel representation learning unit generates node representations by implementing graph convolution on the two discretized sides using low-pass and high-pass filters, respectively, using a dual-channel coding component. The generated node representations are used to measure similarities between nodes in order to train the edge discriminator, thereby integrating representation learning with edge discrimination.

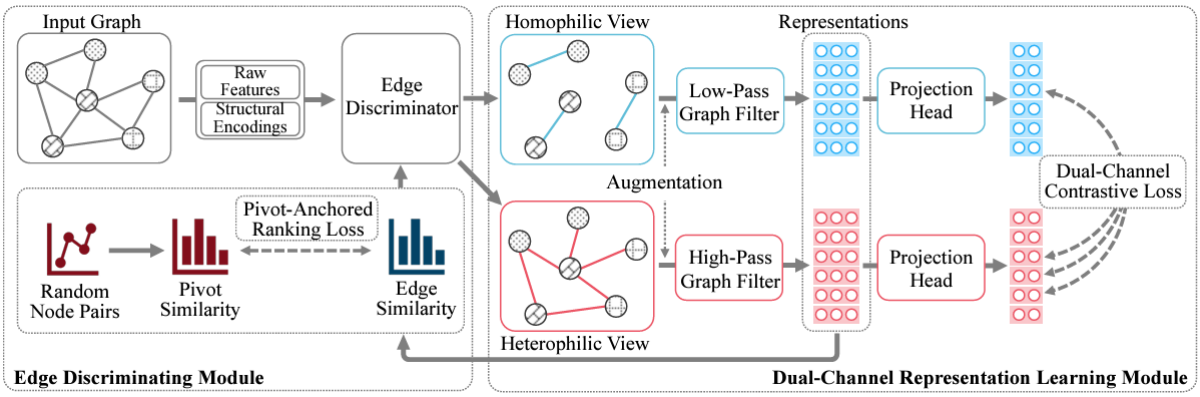

Figure 3.1: The general structural design of the GREET model [2].

3.1.1 Edge Discriminating Module

The edge discriminator is designed to estimate the homophily probability of each edge and, based on this, to distinguish between homophilic and heterophilic edges. The probability of homophily between two neighboring nodes is mainly influenced by two main factors, the structural features where the topological role of the nodes is revealed and the features in which their semantic content is contained. A common solution for integrating information about the structure and characteristics of a node is the use of GNN. Nevertheless, the employment of GNN leads to a smoothing of the node's characteristics, resulting in the loss of the necessary information needed by the model about heterophily.

For this reason, the model examines an alternate approach. It attempts to model the structural features with a vectorial structural encoding, or, as it is otherwise known, positional encoding [45, 46, 47]. By applying positional coding, the combination of node structural encodings with raw features results in the preservation of effective evidence. More precisely, to apply position coding and extract structural information, the model uses a structural coding based on a diffusion process with random walk [46]. Using d_s-step random walk-based graph diffusion, the d_s-dimensional structural encoding s_i of node v_i can possibly be calculated:

$$\mathbf{s}_i = \left[\mathbf{T}_{ii}, \quad \mathbf{T}_{ii}^2, \quad ..., \quad \mathbf{T}_{ii}^{d_s}\right] \in \mathbb{R}^{d_s} \tag{3.1}$$

where $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$ is the random walk transition matrix.

Applying two MLP layers, the edge discriminator calculates the homophily probability given the input of the generated structural encodings and raw features in the following way:

$$\mathbf{h}'_i = MLP_1([\mathbf{x}_i||\mathbf{s}_i]), \mathbf{h}'_j = MLP_1([\mathbf{x}_j||\mathbf{s}_j]),$$
$$\theta_{i,j} = (MLP_2([\mathbf{h}'_i||\mathbf{h}'_j]) + MLP_2([\mathbf{h}'_j||\mathbf{h}'_i]))/2 \tag{3.2}$$

where $[\cdot||\cdot]$ represents the concatenation operation, $\mathbf{h}'_i$ is an intermediate embedding of node v_i, and $\theta_{i,j}$ is the estimated homophily probability for $e_{i,j}$. In the embedding concatenations, we use the second MLP layer for their different orders to prevent the estimation from being influenced by the edges' direction.

The computation of the probability $\theta_{i,j}$ aims to derive a sample of a binary homophily indicator $w_{i,j}$ for edge $e_{i,j}$ from the Bernoulli distribution $w_{i,j} \sim$ Bernoulli($\theta_{i,j}$), where when $w_{i,j} = 1$, homophily is implied and when $w_{i,j} = 0$, heterophily is implied. The discriminator is difficult to train since the sampling is not differentiable, so to overcome this challenge, the model approximates the binary indicator $w_{i,j}$ with the Gumbel-Max reparameterization trick [48, 49]. More precisely, the discrete homophily indicator $w_{i,j}$ is loosened to a continuous homophily weight $\hat{w}_{i,j}$, which is calculated as follows:

$$\hat{w}_{i,j} = Sigmoid((\theta_{i,j} + log\delta - log(1-\delta))/\tau_g) \tag{3.3}$$

where the temperature hyperparameter is $\tau_g > 0$ and the selected Gumbel random variate is $\delta \sim$ Uniform(0, 1). When τ_g is closer to 0, then $\hat{w}_{i,j}$ is inclined to be more precise, approaching 0 or 1.

By computing the edge homophily indicators, it is possible to distinguish the original graph $G = (\mathbf{A}, \mathbf{X})$ into homophilic and heterophilic graph views, $G^{(hm)} = (\mathbf{A}^{(hm)}, \mathbf{X})$ and $G^{(ht)} = (\mathbf{A}^{(ht)}, \mathbf{X})$. Furthermore, in order to enable training of the edge discriminator, replacing the explicit distinction of edges between homophilic and heterophilic types, it assigns a soft weight to every edge when it operates on the homophilic or heterophilic view:

$$\mathbf{A}^{(hm)}_{i,j} = \hat{w}_{i,j}, \mathbf{A}^{(ht)}_{i,j} = 1 - \hat{w}_{i,j}, \text{ for } e_{i,j} \in \mathcal{E} \tag{3.4}$$

3.1.2 Dual-Channel Encoding Module

Based on homophilic and heterophilic graph views, in order to produce informative representations from them, it is necessary to extract the common information between similar nodes along homophilic edges and filter out unrelated information arising from dissimilar neighbors along heterophilic edges. To accomplish this, the model consists of two different encoders that apply to the homophilic and heterophilic graph views, low-pass and high-pass graph filters, respectively.

In the homophilic view, where similar nodes are linked together, the low-pass graph filter is applied to smooth the characteristics of the nodes along the homophilic structure. The application of the low-pass filter is chosen as it can extract the low-frequency information in the graph signals and subsequently maintain the common knowledge of similar nodes. SGC [50] is used as the low-pass GNN filter in this model, which is analyzed as follows:

$$\mathbf{H}_0^{(hm)} = MLP^{(hm)}(\mathbf{X}), \mathbf{H}_l^{(hm)} = \tilde{\mathbf{A}}^{(hm)}\mathbf{H}_{l-1}^{(hm)} \tag{3.5}$$

where $\tilde{\mathbf{A}}^{(hm)}$ is the symmetric normalized homophilic adjacency array, $l \in \{1, ..., L\}$ is the index of the layer and $\mathbf{H}_L^{(hm)}$ is the final output that forms the homophilic representation array $\mathbf{H}^{(hm)}$.

In the heterophilic view, where dissimilar nodes are connected together, it is not efficient to apply a low-pass filter since it would normalize the characteristics of the nodes. As a result, the information of different nodes would be confused, which would lead to a loss of the node's discriminative properties. According to these observations, the model applies high-pass graph filtering, which is the reverse function of the low-pass filter, in order to utilize the heterophilic edges in a more efficient way. The high-pass filter sharpens the characteristics of the nodes along the edges and preserves the high-frequency graph signals, so it is possible to distinguish the representations of dissimilar connected nodes from one another. Additionally, it has been proven that the Laplacian graph operator, i.e., the multiplication of the Laplacian graph matrix $\mathbf{L}$, is effective for extracting high-frequency components. For this purpose, the model, in order to apply high-pass filtering to the heterophilic view, uses Lap-SGC, a simple GNN, which is explained below:

$$\mathbf{H}_0^{(ht)} = MLP^{(ht)}(\mathbf{X}), \mathbf{H}_l^{(ht)} = \tilde{\mathbf{L}}^{(ht)}\mathbf{H}_{l-1}^{(ht)} \tag{3.6}$$

where $\tilde{\mathbf{L}}^{(ht)} = \mathbf{I} - \alpha\tilde{\mathbf{A}}^{(ht)}$ is the symmetric normalized Laplacian array of the heterophilic

view, where α is a hyperparameter that regulates the intensity of high-pass filtering, $l \in \{1, ..., L\}$ and $\mathbf{H}_L^{(ht)}$ is the final output of Lap-SGC that forms the heterophilic representation array $\mathbf{H}^{(ht)}$.

Therefore, two different sets of representations capturing low- and high-frequency information, respectively, are generated by applying two distinct encoders, an SGC encoder and a Lap-SGC encoder. The final node representations are produced by concatenating the representations generated from both views: $\mathbf{H} = [\mathbf{H}^{(hm)}||\mathbf{H}^{(ht)}] \in \mathbb{R}^{n \times d_r}$ with its i-th row $\mathbf{h}_i \in \mathbb{R}^{d_r}$ being the representation vector of node v_i.

3.1.3 Model Training

In this method, the training strategies of the edge discrimination unit and the representation learning unit have been specifically designed. The edge discrimination unit training is based on node representations to measure node similarity, and a pivot-anchored ranking loss has been designed. The representation learning unit then produces representations of the two views produced by the edge discriminator unit and creates a strong dual-channel contrast loss for training the representation encoders. Finally, an alternating training strategy is introduced to repeatedly optimize the two components so that each component increasingly enhances the other, so that the total optimization objective can be theoretically defined as the sum of the losses of the two components.

Pivot-Anchored Ranking Loss

The edge discriminator is designed to distinguish between homophilic edges, where similar nodes are connected to each other, and heterophilic edges, where dissimilar nodes are linked together. However, in this model, a major limitation is that the edge discriminator must efficiently perform and find the boundary between ”similar” and ”dissimilar” nodes in an unsupervised scenario. For this purpose, it is proposed to compute the similarity using randomly sampled pairs of nodes as ” pivots”. In addition, a pivot-anchored ranking loss is designed to ensure that pairs of nodes connected by homophilic edges are notably more similar than pairs of pivot nodes and, correspondingly, pairs of nodes linked by heterophilic edges are significantly more dissimilar than pairs of pivot nodes. More specifically, for each homophilic or heterophilic edge $e_{i,j} \in \mathcal{E}$, the homophilic pivot-anchored ranking loss $R^{(hm)}(e_{i,j})$ and the heterophilic pivot-anchored ranking loss $R^{(ht)}(e_{i,j})$ are defined, correspondingly, as follows:

$$R^{(hm)}(e_{i,j}) = [s_{v_{i'},v_{j'}} - s_{e_{i,j}} + \gamma^{(hm)}]_+,$$
$$R^{(ht)}(e_{i,j}) = [s_{e_{i,j}} - s_{v_{i'},v_{j'}} + \gamma^{(ht)}]_+ \tag{3.7}$$

where $[x]_+ = max(x, 0)$ indicates that exclusively the positive case of x is taken into consideration, $s_{e_{i,j}} = cos(\mathbf{h}_i, \mathbf{h}_j)$ is the cosine similarity at the representation level between the node pair connected by the edge $e_{i,j}$, $s_{v_{i'},v_{j'}} = cos(\mathbf{h}_{i'}, \mathbf{h}_{j'})$ is the cosine similarity between two randomly selected nodes $v_{i'}$ and $v_{j'}$, $\gamma^{(hm)}$ and $\gamma^{(ht)}$ are correspondingly the margin parameters for the homophilic and heterophilic pivot-anchored ranking losses, which have the ability to influence the similarity difference to a considerable degree.

The homophilic and heterophilic ranking losses $L_r^{(hm)}$ and $L_r^{(ht)}$ are obtained by applying the expectation to all homophilic and heterophilic edges, respectively, as shown in the following:

$$L_r^{(hm)} = \mathop{\mathbb{E}}_{e_{i,j}\sim\prod(\hat{w}_{i,j})} R^{(hm)}(e_{i,j}) = \frac{1}{\hat{W}^{(hm)}} \sum_{e_{i,j}\in\mathcal{E}} \hat{w}_{i,j} R^{(hm)}(e_{i,j}),$$
$$L_r^{(ht)} = \mathop{\mathbb{E}}_{e_{i,j}\sim\prod(1-\hat{w}_{i,j})} R^{(ht)}(e_{i,j}) = \frac{1}{\hat{W}^{(ht)}} \sum_{e_{i,j}\in\mathcal{E}} (1-\hat{w}_{i,j}) R^{(ht)}(e_{i,j}), \tag{3.8}$$

where the multinomial distributions are $\prod(\hat{w}_{i,j})$ and $\prod(1-\hat{w}_{i,j})$, and $\hat{w}_{i,j}$ and $1-\hat{w}_{i,j}$ are the homophilic and heterophilic edge weights that parameterize the multinomial distributions, $\hat{W}^{(hm)}$ and $\hat{W}^{(ht)}$ are the normalization conditions that are derived correspondingly from the sum of the total edge weights $\hat{w}_{i,j}$ and $1-\hat{w}_{i,j}$. Therefore, the total pivot-anchored ranking loss to be minimized is defined as follows:

$$L_r = L_r^{(hm)} + L_r^{(ht)} \tag{3.9}$$

Robust Dual-Channel Contrastive Loss

A strong contrasting mechanism is employed to train the node representation learning module, which forces the model from two distinct graph projections to produce semantically consistent representations. In this way, through the indirect use of the produced representations for feature-level similarity reconstruction, the negative effects caused by small amounts of falsely distinguished edges are effectively combated.

In particular, first for each node $v_i \in \mathcal{V}$, it detects another node v_j within the top-k similar subset of nodes of v_i that is measured by the attributes $\mathcal{N}_i$=kNN(v_i,k), and then compares

their representations across homophilic and heterophilic views. With the aim of providing a fair measure of similarity, the node representations resulting from the two different views are projected into a latent space with two MLP learners, correspondingly, and subsequently the similarity between the projections in the latent space is used as a metric to measure the node representation similarity. The robust cross-channel contrastive loss, which is based on InfoNCE contrastive loss [51, 33], is written as follows:

$$L_c = -\frac{1}{n}\sum_{v_i \in \mathcal{V}}[\frac{1}{2|\mathcal{N}_i|}\sum_{v_j \in \mathcal{N}_i}(log\frac{e^{cos(z_i^{(hm)}, z_j^{(ht)})/\tau_c}}{\sum_{v_k \in \mathcal{V}\setminus v_i} e^{cos(z_i^{(hm)}, z_k^{(ht)})/\tau_c}} + log\frac{e^{cos(z_i^{(ht)}, z_j^{(hm)})/\tau_c}}{\sum_{v_k \in \mathcal{V}\setminus v_i} e^{cos(z_i^{(ht)}, z_k^{(hm)})/\tau_c}})] \tag{3.10}$$

where $\mathbf{z}_i^{(hm)}$ and $\mathbf{z}_i^{(ht)}$ are correspondingly the projections of the representations of the homophilic and heterophilic views of v_i in latent space, τ_c is a hyperparameter named temperature for contrastive learning, and $cos(\cdot, \cdot)$ is cosine similarity. Technically, the contrastive loss is computed using a mini-batch method for large graph datasets.

3.2 Knowledge Distillation Strategy

The fundamental concept of knowledge distillation is the pre-training of a large model, called the teacher, as an offline training process and then computing a smaller model with fewer parameters, called the student, which is suitable for deployment in production where available resources are limited. The complicated neural network structure of the teacher model requires the training of a high number of parameters to find out the offline data pattern. More specifically, the training of the student model is performed according to a distillation loss function to transfer knowledge from the pre-trained teacher model. The student model, therefore, imitates the performance of the teacher model while maintaining excellent prediction accuracy while reducing the requirements for student model training, as the training costs of the model are decreased due to its compressed size.

In this study, the aim is to use the GREET model and achieve a remarkable reduction in requirements by transferring knowledge to a student model. First, the teacher model is pre-trained according to the instructions of the original paper. Then, the student model, which is a variant of the original GREET model with reduced parameters, is generated and trained according to a distillation loss function in order to transfer the knowledge from the teacher model that was obtained during its training [52, 53].

The knowledge obtained from the teacher GREET model was passed to the student GREET model through the distillation loss function L^D, whereby the student model is alternately trained for the iterative optimization of the two components [52]. The distillation loss function is defined as a minimization problem in the following way:

$$minL^D = \gamma L^S + (1 - \gamma)L^T \tag{3.11}$$

where L^S is the ranking loss and the contrastive loss produced by the student model and L^T is the inference error of the two modules of the teacher model, in Equations 3.9 and 3.10. The hyperparameter $\gamma \in [0, 1]$ during the training of the student model regulates the relative influence provided by the two losses of the models, the student and the teacher. When we set a larger value for parameter γ, it indicates that more importance is given to minimizing the student's model losses, and thus, it does not highly consider the knowledge of the teacher model. The L^D distillation loss function allows knowledge transfer with reduced training requirements and also enables the accuracy of the model to be preserved at similar levels or even improved.

3.2.1 Variations of the Model

The objective of knowledge distillation is to decrease the number of trainable parameters in the student model in order to limit the resources required to use the model. In order to examine the efficiency of the knowledge distillation framework in the recommended GREET model, two variants of the original model were generated with a greatly compressed size. According to the original model, which has the most extended number of parameters and was proposed in the initial paper, a model with a smaller number of parameters was created first, followed by a second model with an even more restricted number of parameters. The development of these variants is intended to evaluate the model based on the effect of model compression during the knowledge transfer process. Furthermore, the experimental evaluation of these variants aims to provide information on the implementation of knowledge distillation in the field of GRL and its effect on the accuracy of the model.

Chapter 4

Experiments

4.1 Datasets and Evaluation Protocol

Dataset Description

Cora and CiteSeer [54] are two homophilic citation network datasets, which are standards for evaluating graph representation algorithms. Each node denotes a paper, each edge denotes a citation relationship between two papers, and the features are bag-of-word representations of documents. The labels correspond to the research topic of the documents, which are categorized into seven and six categories for each dataset, respectively.

Cornell, Texas and Wisconsin [34] are three heterophilic datasets, which are more specifically three web networks from the computer science departments of different universities. Nodes are web pages and edges are hyperlinks between two of them. The features are representations of the corresponding page in the form of a bag of words and the labels correspond to web page types that are classified into five categories.

Chameleon [34] is also a heterophilic dataset, which is a Wikipedia network, where nodes indicate web pages in Wikipedia and edges indicate connections between two of them. The attributes include the informative nouns on Wikipedia pages and the labels represent the average web page visitation, which is categorized into five categories.

All datasets are accessed by importing the torch_geometric.datasets library. In addition, for homophilic datasets, public separations were adopted and for heterophilic datasets, the commonly used 48%/32%/20% training/validation/testing splits, which have been used in previous works, were adopted. The statistics of the datasets are presented in the Table4.1.

Table 4.1: Statistics of benchmarking datasets

Dataset	Nodes	Edges	Features	Classes	Train/Val/Test
Cora	2,708	10,556	1,433	7	140/500/1000
CiteSeer	3,327	9,104	3,703	6	120/500/1000
Chameleon	2,277	36,051	2,325	5	1,092/729/456
Cornell	183	295	1,703	5	87/59/37
Texas	183	309	1,703	5	87/59/37
Wisconsin	251	499	1,703	5	120/80/51

Training and Evaluation Protocol

To evaluate the effectiveness of the representations produced by the model, the transductive node classification is adopted as a task. In the training phase, the models are trained in an unsupervised manner. In the evaluation phase, the generated representations are frozen and a logistic regression classifier is applied to train and test them. For all datasets, the evaluation metric is reported as the mean test accuracy and standard deviation for ten runs of experiments. For each trial, a teacher model and a corresponding student model are created.

Accuracy

A fundamental metric for evaluating the performance of a model that has classification as its task is accuracy, providing a brief insight into how well the model performs in terms of correct predictions. It is estimated as the ratio of correct predictions to the total number of input samples.

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{All Predictions}} \tag{4.1}$$

4.2 Models

4.2.1 Examined Models

To evaluate the performance and effectiveness of the experiments, a number of recent implementations of models related to GRL are selected, which belong to the category of contrastive UGRL methods. These methods are as follows:

- **DGI [55]:** maximizes the mutual information between the input graphs and their high-level hidden representations and captures both node characteristics and topological structure in an unsupervised manner.
- **GMI [56]:** maximizes mutual information between local surface representations and high-level graph summaries, but without relying on random walk methods.
- **GCA [57]:** uses adaptive augmentation systems based on topological and semantic biases to preserve the inherent structures and features of graphs.
- **BGRL [58]:** provides alternative augmentations of the input graph without relying on negative examples, so as to reduce memory overhead and allow application to extremely large graphs.

The publicly accessible version was used for each baseline. In addition, the tuning of the parameters was maintained as stated in the original publication of each, on condition that the same datasets were used.

4.2.2 Experimental Models

The experimental models include the GREET model [2], the original model, which is also considered the teacher model, and the GREET-KD* and GREET-KD** models, which are used as the student models in the knowledge distillation process.

- **GREET:** the proposed method.
- **GREET-KD*:** is a variant of the proposed model with a reduced number of parameters
- **GREET-KD**:** is a variant of the proposed model with an even further reduced number of parameters

For the proposed method and its variants, the initial tuning of the parameters as stated in the original paper was used.

More specifically, for the first GREET-KD* student model, it adapts the same hyper-parameters as the original work, except for the value of the dimension of intermediate embedding d_i, the dimension of representations d_r and the dimension of projected embedding d_p, where $d_i = d_r = d_p = 32$ are used. This achieves a highly reduced set of parameters compared to the teacher model, where $d_i = d_r = d_p = 128$ is used for these parameters.

Then, for the second student model GREET-KD**, which wants to achieve even more parameter reduction, the above hyperparameters are set to $d_i = d_r = d_p = 16$.

4.3 Performance Evaluation

In the tables 4.2 and 4.3, we present the experimental results when comparing the examined models on two homophilic and four heterophilic datasets, respectively. First, without applying the knowledge distillation strategy and comparing their performances, we can observe that the GREET model significantly outperforms all the examined models. Being designed to improve representation learning performance on heterophilic graphs, it is important to place particular emphasis on its superiority of performance on heterophilic datasets over the baselines, in which case the minimum improvement is by 12% over the GMI model for the Cornell dataset. Given that baselines are typical and contrastive UGRL methods, it is evident that the GREET method clearly outperforms them. The main drawback of these UGRL methods is that, following their approach, they continuously smooth the representations along heterophilic edges, losing a significant amount of information. On the other hand, the higher performance of GREET is due to the discrimination and exploitation of both edge types. Therefore, it is concluded that this approach can benefit GRL in general, knowing that data derived from real applications contains both homophilic and heterophilic edges.

Table 4.2: Results in terms of classification accuracies (in percent ± standard deviation) on homophilic benchmarks.

Model	Dataset	
	Cora	CiteSeer
DGI	81.96± 0.81	72.52±0.048
GMI	82.77± 0.85	69.47±1.17
GCA	81.77± 1.023	69.95±1.005
BGRL	79.14± 1.71	69.41±1.10
GREET	**83.21± 0.49**	**72.77±0.99**
GREET-KD*	82.57±0.89	71.69±1.33
GREET-KD**	78.76±1.42	71.09±0.86

Table 4.3: Results in terms of classification accuracies (in percent ± standard deviation) on heterophilic benchmarks.

Model	Dataset			
	Chameleon	Cornell	Wisconsin	Texas
DGI	39.21±1.34	60.00±4.15	57.06±2.23	66.49±2.76
GMI	41.54±1.95	61.62±6.02	64.31±2.60	57.03±5.33
GCA	49.46±2.14	55.38±6.30	53.08±6.33	58.21±4.59
BGRL	43.60±1.09	54.05±2.70	62.94±2.84	66.76±1.73
GREET	**62.50±2.74**	**74.05±3.01**	**84.31±2.32**	**85.68±3.64**
GREET-KD*	58.60±1.76	73.24±5.62	80.59±5.50	82.62±7.28
GREET-KD**	54.08±2.37	70.54±5.17	79.22±6.55	81.08±5.41

According to the experimental results, the outperformance of the GREET model is the main reason for its use as a teacher model. Nevertheless, due to its intricate architecture, which consists of two different modules to allow GREET to show higher performance, it is a model with considerable computational cost, as evidenced by the number of parameters presented in the tables 4.4 and 4.5. As a consequence, the proposed knowledge distillation approach focuses on reducing these training parameters, which means that fewer resources are required to utilize the model while keeping the performance within a similar range.

In tables 4.2 and 4.3, the performance of the variants of the proposed method, which are the GREET-KD* and GREET-KD** student models, is also presented. According to the results, comparing the variants with the original model and baselines, it is noticed that on heterophilic datasets, the student models show a slight drop in performance compared to the teacher model but still outperform the other methods, indicating that the knowledge and decision-making capabilities of the teacher model have been distilled to a sufficiently high degree of accuracy. On homophilic datasets, the student models do not outperform either the other methods nor the original model but still preserve extremely high performance, remarking that the maximum deviation from the lowest performance of the other models is less than 1%. Furthermore, it is important to consider that the students' models have dramatically reduced the number of parameters compared to the teacher model, showing a compression rate of 76% and 88%, respectively, in the two experimental variants. These experimental findings demonstrate the effectiveness of the proposed knowledge distillation approach in the

field of GRL, as the student models are able to attain quite high performance considering the remarkable reduction in computational resources.

Table 4.4: Model parameters on homophilic datasets (in millions), when comparing the proposed GREET student model with the other non-distillation strategies.

Model	Dataset		Compression Rate
	Cora	CiteSeer	
DGI	0.996	2.159	
GMI	1.993	4.317	
GCA	0.882	2.045	
BGRL	1.130	2.292	
GREET	0.586	1.491	
GREET-KD*	0.140	0.360	0.76
GREET-KD**	0.070	0.179	0.88

Table 4.5: Model parameters on heterophilic datasets (in millions), when comparing the proposed GREET student model with the other non-distillation strategies.

Model	Dataset				Compression Rate
	Chameleon	Cornell	Wisconsin	Texas	
DGI	1.453	0.056	0.056	0.502	
GMI	2.906	1.004	1.004	1.004	
GCA	1.355	0.055	0.055	0.112	
BGRL	1.587	0.602	0.602	0.116	
GREET	0.929	0.723	0.723	0.723	
GREET-KD*	0.226	0.168	0.168	0.168	0.76
GREET-KD**	0.113	0.083	0.083	0.083	0.88

4.4 Parameter Tuning

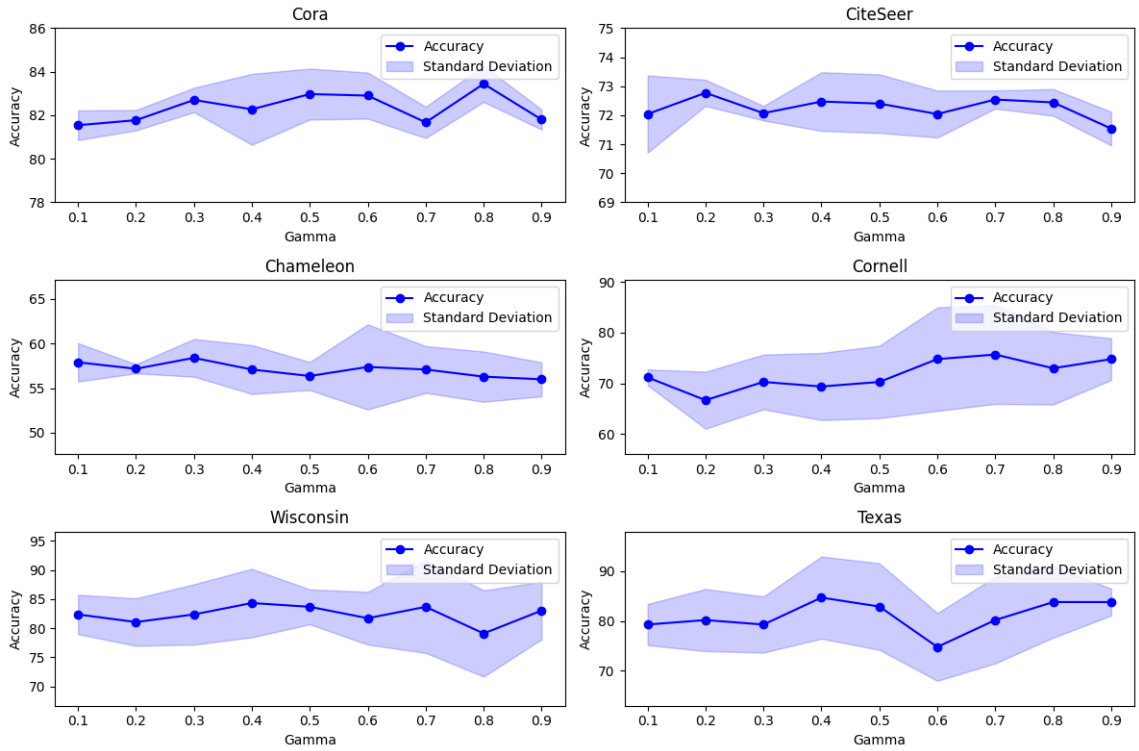

Figure 4.1: Impact of hyperparameter γ on the prediction accuracy of student model GREET-KD*.

In Figures 4.1 and 4.2, we evaluate the effect of the hyperparameter γ of equation 3.11 on the performance of the student models for each of the datasets used for experimental evaluation. The value of the hyperparameter γ is varied from 0.1 to 0.9, with a step of 0.1. For each value of the hyperparameter γ, the average accuracy resulting from three runs is reported. This observation contributes to understanding how the value of γ affects the distillation process and how the teacher influences the student. In line with the above, it is observed that for all datasets and in both variants of the model, the best performance is reported for values of γ ranging between 0.2 and 0.8. In particular, when γ takes the value 0.1, the knowledge of the teacher model is mainly distilled, which means that the student's training is restricted to the teacher's knowledge, and in this way, the prediction accuracy is reduced. Conversely, when the γ takes the value of 0.9, the distillation process is focused on the student, which means that the student model derives less knowledge from the teacher model, which does not contribute to the accuracy improvement.

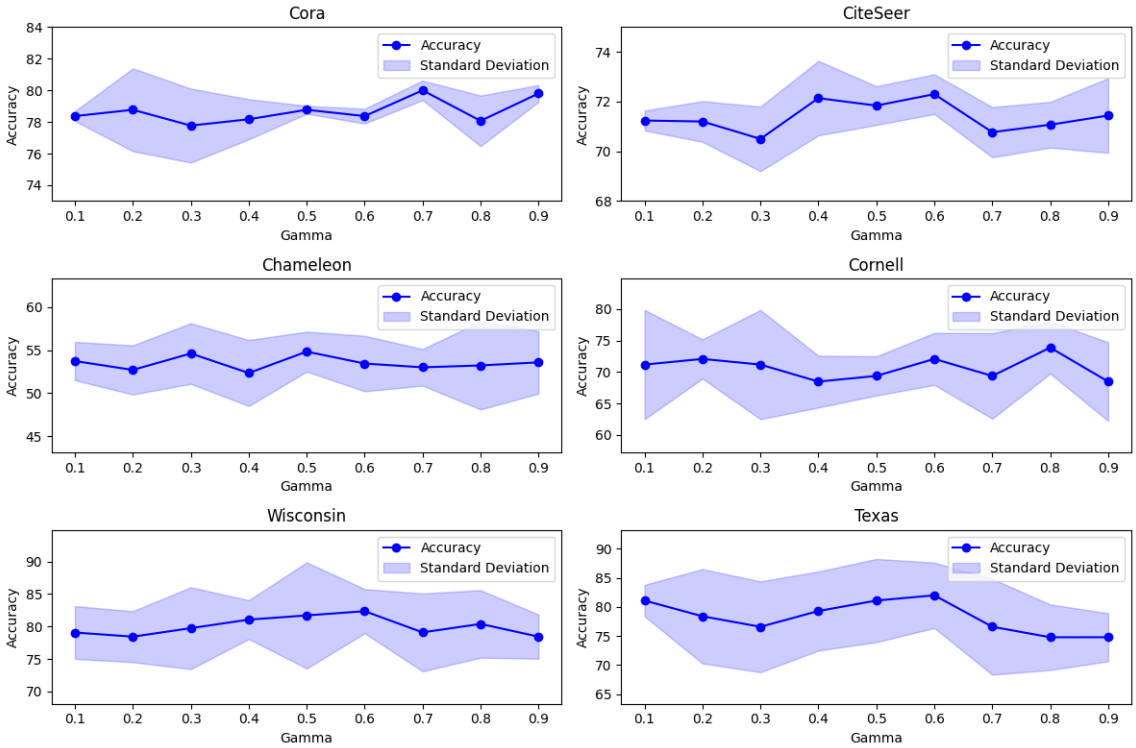

Figure 4.2: Impact of hyperparameter γ on the prediction accuracy of student model GREET-KD**.

Chapter 5

Conclusion

In this thesis, the implementation of a knowledge distillation framework for model compression in the field of graph representation learning (GRL) was presented. The motivation behind this was to reduce the computational cost without significantly affecting the performance of the model. In order to draw conclusions on the effectiveness of the proposed method, extensive experiments are performed using different state-of-the-art methods belonging to the category of contrastive UGRL techniques, and their performance is compared with the performance of the proposed model, which is GREET. While observing the results, the outperformance of the GREET model is demonstrated compared to the other GRL methods. The design of the model, in which it uses an edge discriminator to distinguish the two types of edges and a dual-channel encoding component to generate representations according to the edge discrimination, makes the model significantly more efficient, especially on heterophilic graphs, in comparison to the baseline GRL methods.

Furthermore, observing the number of parameters required to implement the model, it is evident that GREET is computationally expensive, as it requires a high number of parameters because of its complicated architecture. For this reason, a knowledge distillation approach is proposed, where knowledge from a large teacher model is transferred to smaller student models. In particular, first a teacher model is pre-trained on offline data, and then to transfer the knowledge to a smaller student model, which requires a lower number of parameters, a distillation loss function is applied. The experimental evaluation showed that the application of knowledge distillation is effective, as the results of the smaller models were quite satisfactory despite the high reduction in the number of parameters. Hence, this study demonstrates that knowledge distillation is an effective approach that can be successfully applied in the field of

GRL and is able to significantly compress the model. The application of this approach, therefore, makes the model reachable in real-world applications where resources are restricted.

Future Work

In the future, this research could be expanded by broadening the experiments with more datasets from different sources and searching for methods to optimize the efficiency of the method for reduced computational requirements, e.g., by examining other forms of loss functions to train the student model. Furthermore, a number of transfer learning techniques could be studied. In this way, the model will be able to exploit the knowledge gained from different graph datasets to train on a new one with higher efficiency, a very important factor for the utilization of the model in real-life applications in which the model is required to train on new data.

Bibliography

[1] Xin Zheng, Yi Wang, Yixin Liu, Ming Li, Miao Zhang, Di Jin, Philip S. Yu, and Shirui Pan. Graph neural networks for graphs with heterophily: A survey, 2024.

[2] Yixin Liu, Yizhen Zheng, Daokun Zhang, Vincent CS Lee, and Shirui Pan. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating, 2022.

[3] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. KDD '09, page 807–816, New York, NY, USA, 2009. Association for Computing Machinery.

[4] Tianyu Liu, Yuge Wang, Rex Ying, and Hongyu Zhao. Muse-gnn: Learning unified gene representation from multimodal biological graph data. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 24661–24677. Curran Associates, Inc., 2023.

[5] Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. Multi-modal knowledge graphs for recommender systems. CIKM '20, page 1405–1414, New York, NY, USA, 2020. Association for Computing Machinery.

[6] T. N. Kipf and M Welling. Semi-supervised classification with graph convolutional networks. *In ICLR*, 2017.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[8] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks?, 2019.

[9] Xin Zheng, Miao Zhang, Chunyang Chen, Soheila Molaei, Chuan Zhou, and Shirui Pan. Gnnevaluator: Evaluating gnn performance on unseen graphs without labels. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 33606–33623. Curran Associates, Inc., 2023.

[10] Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7793–7804. Curran Associates, Inc., 2020.

[11] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks, 2020.

[12] Sangchul Hahn and Heeyoul Choi. Self-knowledge distillation in natural language processing, 2019.

[13] Yan Gao, Titouan Parcollet, and Nicholas Lane. Distilling knowledge from ensembles of acoustic models for joint ctc-attention end-to-end speech recognition, 2021.

[14] Xihua Li and Qikun Shen. A hybrid framework based on knowledge distillation for explainable disease diagnosis. *Expert Systems with Applications*, 238:121844, 2024.

[15] Rafaël Van Belle, Charles Van Damme, Hendrik Tytgat, and Jochen De Weerdt. Inductive graph representation learning for fraud detection. *Expert Systems with Applications*, 193:116463, 2022.

[16] William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.

[17] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *in Proc. of ICLR*.

[18] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. 06 2016.

[19] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2019.

[20] Mingguo He, Zhewei Wei, zengfeng Huang, and Hongteng Xu. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 14239–14251. Curran Associates, Inc., 2021.

[21] Mingqi Yang, Yanming Shen, Rui Li, Heng Qi, Qiang Zhang, and Baocai Yin. A new perspective on the effects of spectrum in graph neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 25261–25279. PMLR, 17–23 Jul 2022.

[22] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[23] Xiao-ming Wu, Zhenguo Li, Anthony So, John Wright, and Shih-fu Chang. Learning with partially absorbing random walks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[24] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. KDD '14, page 701–710, New York, NY, USA, 2014. Association for Computing Machinery.

[25] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. WWW '15, page 1067–1077, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee.

[26] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272. PMLR, 06–11 Aug 2017.

[27] Zemin Liu, Yuan Fang, Chenghao Liu, and Steven C.H. Hoi. Node-wise localization of graph neural networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, August 2021.

[28] Zhongyu Huang, Yingheng Wang, Chaozhuo Li, and Huiguang He. Going deeper into permutation-sensitive graph neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 9377–9409. PMLR, 17–23 Jul 2022.

[29] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks, 2016.

[30] Thomas N. Kipf and Max Welling. Variational graph auto-encoders, 2016.

[31] Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, and Chengqi Zhang. Adversarially regularized graph autoencoder for graph embedding, 2019.

[32] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations, 2021.

[33] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning, 2020.

[34] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks, 2020.

[35] Yujun Yan, Milad Hashemi, Kevin Swersky, Yaoqing Yang, and Danai Koutra. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks, 2022.

[36] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021.

[37] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3048–3068, June 2022.

[38] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[39] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015.

[40] Kai Wang, Yu Liu, Qian Ma, and Quan Z. Sheng. Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings. In *Proceedings of the Web Conference 2021*. ACM, April 2021.

[41] Kaituo Feng, Changsheng Li, Ye Yuan, and Guoren Wang. Freekd: Free-direction knowledge distillation for graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, August 2022.

[42] Cuiying Huo, Di Jin, Yawen Li, Dongxiao He, Yu-Bin Yang, and Lingfei Wu. T2-gnn: Graph neural networks for graphs with incomplete features and structure via teacher-student distillation, 2022.

[43] Jiaqi Ma and Qiaozhu Mei. Graph representation learning via multi-task knowledge distillation, 2019.

[44] Chaitanya K. Joshi, Fayao Liu, Xu Xun, Jie Lin, and Chuan Sheng Foo. On representation knowledge distillation for graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4656–4667, April 2024.

[45] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning, 2020.

[46] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations, 2022.

[47] Yue Tan, Yixin Liu, Guodong Long, Jing Jiang, Qinghua Lu, and Chengqi Zhang. Federated learning on non-iid graphs via structural knowledge sharing. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023.

[48] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017.

[49] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2017.

[50] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr. au2, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying graph convolutional networks, 2019.

[51] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[52] Stefanos Antaris, Dimitrios Rafailidis, and Sarunas Girdzijauskas. Egad: Evolving graph representation learning with self-attention and knowledge distillation for live video streaming events, 2020.

[53] Stefanos Antaris and Dimitrios Rafailidis. Distill2vec: Dynamic graph representation learning with knowledge distillation, 2020.

[54] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi□Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, sep 2008.

[55] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax, 2018.

[56] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. Graph representation learning via graphical mutual information maximization, 2020.

[57] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*. ACM, April 2021.

[58] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L. Dyer, Rémi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping, 2023.